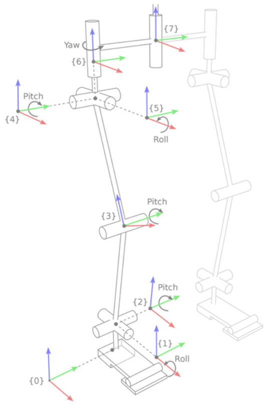
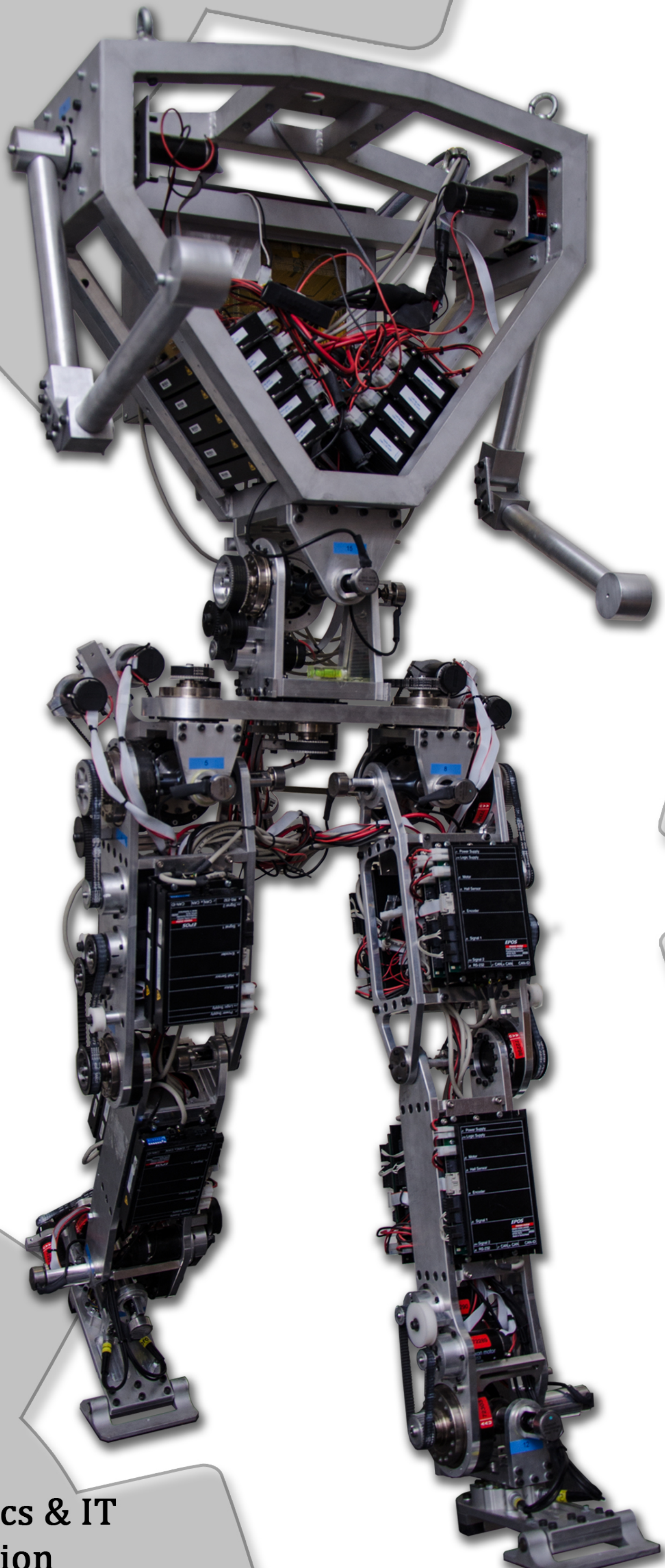
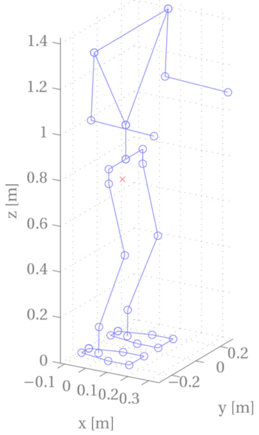
Approaching Dynamic Gait with AAU-BOT1

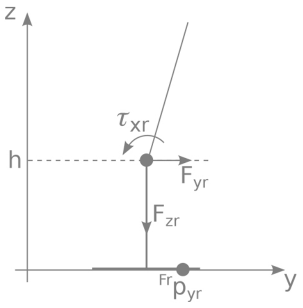






Master Thesis

06-06-2013
Aalborg University
13gr1034 - Electronics & IT
Control and Automation



AALBORG UNIVERSITY
STUDENT REPORT

Department of Electronic systems
Electronics & IT
Fredrik Bajers Vej 7
9220 Aalborg Ø
www.es.aau.dk

Title:
Approaching Dynamic Gait with AAU-BOT1

Theme:
Control of Complex Systems

Project period:
September 3rd 2012 - June 6th 2013

Group:
Group 1034

Group members:
Niels H. Andersen
Rune Madsen

Supervisors:
Jan Helbo
Morten Juelsgaard
Jakob Stoustrup

Copies: 6

Page count: 186

Attachments: DVD

Appendix: 6

Completion: 6th June 2013

Synopsis:

This thesis concerns the topic of obtaining dynamic gait with the AAU-BOT1. Previous work on the AAU-BOT1 has resulted in a functional robot, where each joint can be individually controlled. The main topic has therefore been to design and implement a controller strategy capable of obtaining dynamic gait with the robot.
Through a study of the human gait, including recording human gait in a motion tracking room, knowledge about human gait has been obtained. Kinematic and dynamic models are derived for the robot. From these the overall movements to obtain gait is deduced.
Two levels of controllers are used to control the robot. Low level controllers are used to control each separate joint. A high level gait and balance controller is used to generate joint references and keep the robot in balance. The perception of balance is gained by measuring the angle of the torso and the interaction between foot and ground. The foot ground impact is given special attention. A shock absorption strategy is used to reduce harmful effects of hard impacts. With the implemented controller strategy the AAU-BOT1 has been shown to be capable of obtaining gait with a velocity up to 0.15 km/h.

The content of this thesis is freely available, but publication (with reference) may only be pursued due to agreement with the authors.

Preface

This thesis is the result of the project produced by group 13gr1034 during the 3rd and 4th semester of the Master programme in Control and Automation at Aalborg University. The theme of this project is to work with control of complex systems. In this scope, the project conducted is enabling AAU-BOT1 to walk autonomously.

The thesis is divided in chapters, sections, and subsections which are numbered. For example, chapter 7, section 1, subsection 2 is numbered 7.1.2. A list of acronyms can be found at the end of the thesis. In the end of the thesis several appendixes can be found which includes material not strictly necessary to read to understand the concepts. These appendixes are denoted with capital letters A, B, C, etc. and will be referred to throughout the thesis.

Some content of the thesis refers to datasheets of used hardware in the project, or implementation code, which can be found on the DVD attached to this thesis. The path to these files is given in the following way: 💿`/folder_name/file_name.ext`. Several videos are captured and generated during this project to document performance of the implemented system. These videos are also found on the DVD 💿`/videos/`.

Niels Hyltoft Andersen

Rune Madsen

CONTENTS

Chapter 1

Introduction

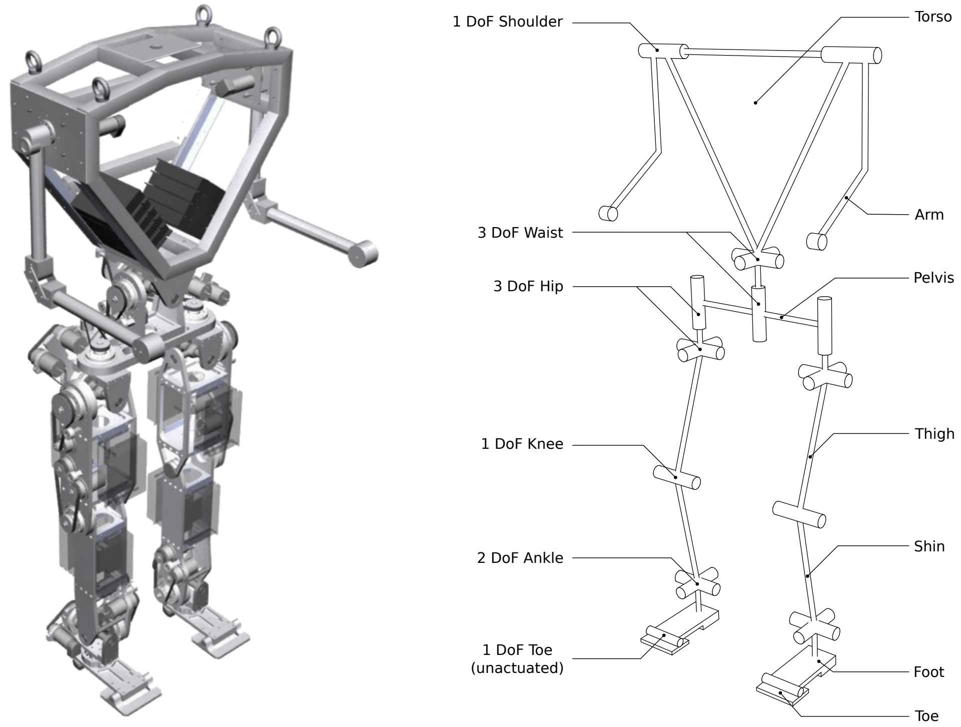
1.1 Background

This thesis treats the topic of obtaining gait with a biped robot. As subject of this research the AAU-BOT1 depicted in figure 1.1(a) will be used. The AAU-BOT1 is a biped humanoid robot which is the first and so far only robot constructed as a part of the AAU-BOT project. The AAU-BOT project serves the purpose of increasing cooperation between health science and robotics at Aalborg University [1]. This is attempted by cooperating on developing a humanoid biped robot which can imitate limb dysfunctionalities, which causes limping similar to those of humans. The added strain that limping puts on other parts of the body can through this project be studied in great detail without involving human test subjects.

The AAU-BOT1 has been designed to resemble the human body closely in terms of proportion and joint flexibility. The actuators placed in the joints of the robot have been chosen to be strong enough to recreate human gait, but have been chosen such that they can provide only slightly more torque than what is needed to walk. This leaves only a small margin for suppressing disturbances and reduces the amount of possible approaches to walking. The strategy for walking with the AAU-BOT1 must be chosen energy efficient by imitating human gait as it was meant for, or at least choose a strategy that minimises the torques needed to walk.

To enable the AAU-BOT1 to walk, the robot has 19 Degrees of Freedom (DoF). These have been carefully selected to imitate the flexibility of a human body, with special focus on the lower half of the body. 17 of the 19 DoF are actuated by electrical motors while the toe joint in each foot is left unactuated. This foot design is made to imitate the flexibility of a human foot which can bend throughout a step. The different DoF's are depicted in figure 1.1(b).

The AAU-BOT1 has been developed by different student groups since its birth in 2006.



(a) 3D drawing [2].

(b) Degrees of freedom.

Figure 1.1. Figure (a) shows a 3D drawing of the AAU-BOT1 robot standing upright, while (b) is a sketch showing the degrees of freedom of the robot.

A total of five different groups have contributed to the development of the robot. At the time of this projects beginning, the hardware part of the AAU-BOT1 has been fully developed and interfaces for motor and sensor units have been designed and implemented.

While previous groups have mainly been focusing on completing the construction of the robot, this project will focus on designing and implementing a gait strategy on the AAU-BOT1 that enables it to perform dynamic gait [2, 3, 4, 5]. The definition of dynamic gait will be introduced in section 2.2.2. Obtaining dynamic gait with the AAU-BOT1 is a major step towards the overall goal of the AAU-BOT1 project. As the AAU-BOT1 is capable of performing gait, limb dysfunctionalities can be simulated by putting constraints on separate joints.

1.2 Thesis Objective

From the work on the AAU-BOT1 by the previous project groups, a fully assembled robot with well defined interface is turned over. The next step towards the overall goal of the AAU-BOT project is to make the AAU-BOT1 perform gait. The thesis of this project is:

Is it possible to obtain dynamic gait with the AAU-BOT1?

As a part of obtaining dynamic gait, the AAU-BOT1 must also be able to initiate walk from standstill and come to a standstill from walking state. The ability to initiate and terminate gait is considered essential abilities to the robot. Without these the robot will have to be hoisted on and off the floor during gait.

The robot was designed by the first project group from a goal of achieving a gait velocity of 1 m/s [3]. During this project focus will however be to obtain dynamic gait and no requirement to the gait velocity is specified. For the AAU-BOT project as a whole, other objectives such as turning and limping could also be of interest, but it has been chosen to limit the scope of the project to obtain gait in a straight line.

1.3 Thesis Outline

In order to obtain the thesis objectives described above, the development is planned into different separate steps. The following list describes in which order the main tasks of the project are carried out. In the list references are given to the chapters where the topics will be treated.

- **Preliminary analysis:** Human gait will be investigated to identify essential properties that can be used in the control of the AAU-BOT1. In order to assess stability of gait different measures of balance will be identified from literature concerning biped locomotion. Finally an assessment of the current status of the AAU-BOT1 will be carried out by revising the documentation from the previous groups and inspecting the instrumentation and implementation of the interfaces. The inspection of the sensors will include calibration and configuration. The outcome of the preliminary analysis is found in chapter 2.

- **Choosing the overall approach:** From the preliminary analysis, overall approaches to the aspects of balancing, control, and trajectory generation for biped robots will be identified. This will give the basis for the choice of an overall controller structure that divides the responsibility of obtaining gait with the AAU-BOT1 into smaller parts. In chapter 3 different approaches will be discussed and the overall controller structure will be settled. Having settled on an overall structure an experiment will be carried where the human gait is recorded to identify parameters and trajectories for the human gait which are relevant for the development of the different parts in the controller. The results from this experiment finalises chapter 3.

- **Modelling:** When the general strategies have been settled, models of the AAU-BOT1 will be derived. Modelling of both kinematic and dynamic behaviour will be needed for controller design. The necessary complexity of the dynamic model of the whole robot is not straight forward to determine, hence several iterations are assumed necessary before a suitable model is found. Some models will be derived as gray or black box models hence experiments will be conducted to determine model parameters. Previous groups have described challenges due to high friction in the joints. This topic will therefore be given special attention. The derived models will be presented in chapter 4.

- **Controller design:** When the above tasks have been completed the controllers can be designed. These will be designed from the models developed and will have the task of enabling the AAU-BOT1 to walk. Different control approaches will be attempted and compared to see which one performs the best. The derived joint controllers responsible of letting the separate joints follow their references are described in chapter 5. The references for the joint controllers are generated by a high level controller from a specification of the desired gait. The high level controller is described in chapter 7. For the high level controller to make joint trajectories that make the robot keep balance the robot will be given a perception of balance based on the measures of balance identified in the preliminary analysis. In chapter 6 the estimation of the measures of balance will be presented.

- **Performance testing:** When a promising controller scheme has been designed and implemented, the performance of this will be tested. At this point it will be verified if the proposed strategy allows the AAU-BOT1 to comply with the thesis objective. If the thesis objectives can not be complied with, the controller design is attempted improved using the test results. As a final controller scheme is settled an acceptance test will be carried out. The result of the acceptance test is presented in chapter 8.

- **Final evaluation:** The results from the acceptance test will finally be compared to the thesis objective and accessed. Proposals will be made on how to enhance the performance of the gait even further in future projects. In chapter 9 the final conclusions of this thesis are drawn and the perspectives for the AAU-BOT1 are presented.

CHAPTER 2

Preliminary Analysis

This chapter introduces the conceptual knowledge needed to analyse human gait and some of the main concepts used in the field of walking humanoids. Additionally the AAU-BOT1 and previous work conducted on the robot will be introduced.

In section 2.1 human gait is analysed and section 2.2 introduces balance measures used in control of bipeds and how these measures is used to distinguish static and dynamic gait. Section 2.3 introduces the actuators and sensors mounted in the AAU-BOT1. In section 2.4 the products from previous projects are introduced.

Chapter Content

2.1 Humanoid Locomotion

In recent years development of humanoid robotics has received a great deal of attention however literature in the field does not always agree on the notation and terminology, and sometimes this leads to confusion. In this section some of the main terms in the field of robotics will be defined to make our interpretation clear. In addition to the definition of the terminology, this section will cover some of the basic theory about the human gait and some of the parameters that describes gait for separate subjects.

The terminology that will be used in this thesis is based on [6] that identifies the misunderstandings in use of terminology and try to unify the terminology and basic notation in the field of humanoid robotics. First the term walk will be defined.

Walk: *Movement by putting forward each foot in turn, not having both feet off the ground at once.*

From this definition is seen that the result of walking is to move the whole body by moving the legs in a way such that only one foot at the time is separated from the ground. The way walking is performed is called gait.

Gait: *Manner of walking or running.*

Looking at humans walking is clear that not two persons have identical ways of walking, and the way the humans are walking (or running) is described by the gait. One representation of a gait could be to record all the joint angles in a subject's body since this will contain information of how the walk is performed.

Throughout the rest of this thesis translations and rotations will be specified as shown in figure 2.1. Furthermore it should be noted that the x-axis in some cases will be denoted the *sagittal axis*, while the y-axis will be denoted the *transverse axis*.

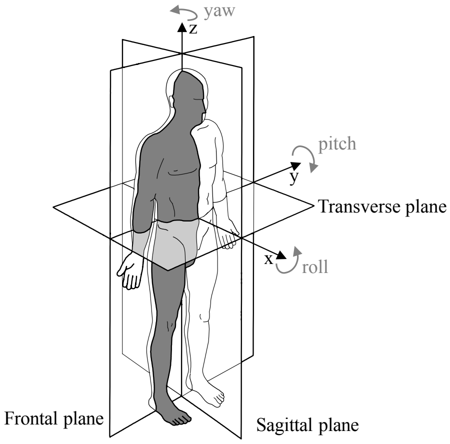


Figure 2.1. Sketch of plane and axis definition with respect to orientation [5].

Having defined the term gait, a deeper analysis of the human gait will reveal which parameters can be used to define the gait of a human.

2.1.1 The human gait

From the definition of walk a cycle is identified, since the feet are placed in front of each other in turn. This cycle will be referred to as the *gait cycle*. Figure 2.2 illustrates the different phases of the gait cycle. The gait cycle is described for the right leg and is for symmetric gait identical, but delayed a half cycle, for the left leg [7].

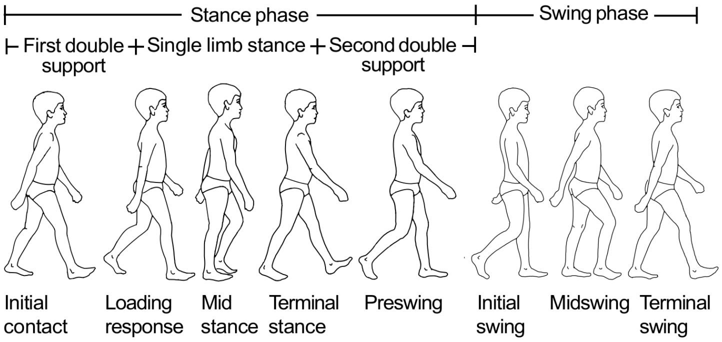


Figure 2.2. The main phases of the human gait cycle. This illustration is borrowed from [7].

The gait cycle consists of two main phases the *stance phase* and the *swing phase*. During the stance phase the right foot is on the ground whereas it is lifted from the ground in the swing phase. The stance phases for the two legs are overlapping which cause a division of the stance phase into sub phases according to figure 2.2:

1. *First double support*, when the right foot touches the ground while the left foot is still in contact with the ground.
2. *Single limb stance*, when the left leg is in the swing phase and only the right foot touches the ground.
3. *Second double support*, when the left foot enters stance phase (as the foot get in contact with the ground) and both legs are in stance phase.

As indicated previously the gait cycle for a healthy human is normally symmetric i.e. equal length of stance and swing phases for both legs where the cycle for the left leg is half a cycle behind (or ahead) of the right leg. For a normal human the swing phase takes about 40 % of the cycle where the stance phase take up the rest 60 %.

Events

The two main phases (stance and swing) of the gait cycle can be divided into distinct events. These events are shown both in figure 2.2 and in figure 2.3. As seen from the figures there exists different names for the same events. In this thesis the names from figure 2.3 will be used. Figure 2.3 illustrates the cyclic nature of gait, and from that no first event (or phase) is apparent.
For the stance phase the events are as follows:

1. *Heel-strike* starts the stance phase and is the point at which the heel gets in contact with the ground. At this time the centre of gravity is at its lowest position.

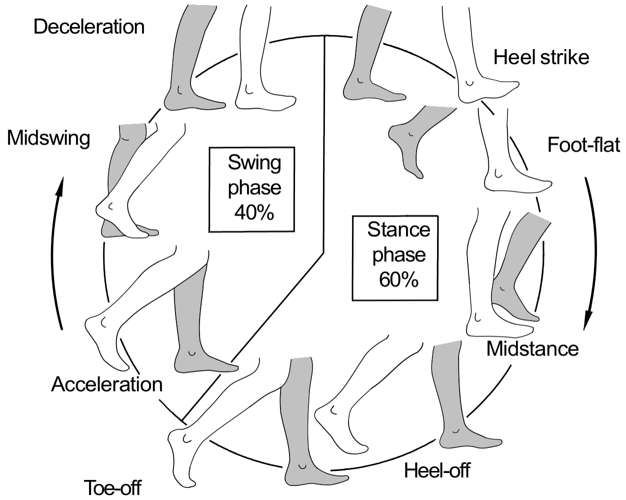


Figure 2.3. The eight events in the swing and stance phases of the human gait cycle. The figure illustrates the events for the right leg (white on figure). This illustration is borrowed from [7] and modified slightly to enhance visibility.

2. *Foot-flat* occurs at the time when the front of the foot reaches the ground and the foot is flat on the ground.
3. *Midstance* occurs when the swinging leg passes the standing leg. At midstance the centre of gravity is at its highest position.
4. *Heel-off* occurs when the heel leaves the ground and only the front of the foot remains in contact with the ground. At this time push-off is initiated to create forward momentum.
5. *Toe-off* ends the stance phase as the foot leaves the ground.

For the swing phase the events are as follows:

7. *Acceleration* begins as the foot leaves the ground and the leg is accelerated forward.
8. *Midswing* occurs when the swinging leg passes the leg in stance phase. This event will occur at the same time of *Midstance* on the other leg.
9. *Deceleration* ends the swing phase. The leg is slowed down and stabilised and thereby prepared for the next cycle initiated by the next heel strike.

Support Areas and Phases

A very essential component in making a humanoid walk and keep balance while during so is the foot contact to the ground. This contact between the foot and the ground is called the support area and is defined by [8] as:

> **Support area:** *The supporting area is formed by the convex hull about the ground support points.*

This area is also referred to as the Polygon of Support (PoS). The size and shape of this area does according to the definition depend on the number of feet having contact to the ground and how much of the feet have contact with the ground. It should be noted that the PoS is not defined when no foot ground contact exists e.g. when running.

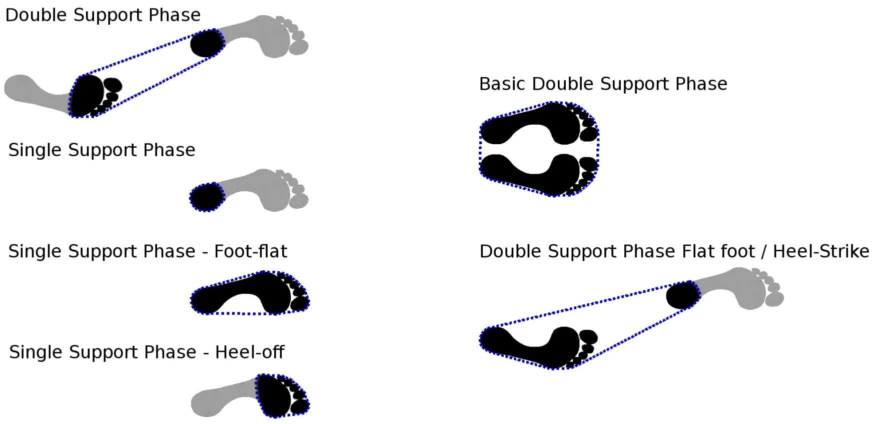

Double Support Phase
Single Support Phase
Single Support Phase - Foot-flat
Single Support Phase - Heel-off
Basic Double Support Phase
Double Support Phase Flat foot / Heel-Strike

Figure 2.4. PoS in the different support phases. The black part of the foot is the part in contact with the ground and the areas encircled by the blue dashed line is the PoS.

Figure 2.4 shows the PoS for both the *single support phase* where only one foot is in contact with the ground and for the *double support phase* where both feet are in contact with the ground. During the single support phase the area of the supporting foot that touches the ground changes from being only the heel, to be the whole foot, and lastly to only be the front foot. This change of area having ground contact yields a subdivision of the single support phase as illustrated in figure 2.4.

1. *Double Support Phase:* the heel strike of the left foot has just occurred and both the heel of the left foot and the forefoot of the right foot has contact with the ground. The support area is in this phase the convex hull including both the heel of the left foot and the toe of the right foot.
2. *Single Support Phase:* as the right foot loses ground contact just after toe off the heel is the only part of the left foot that has contact with the ground. The support area reduces to be the convex area containing the left heel.
3. *Single Support Phase - Foot-flat:* as the left foot rolls over and the fore foot of the left foot reaches the ground the support area expands to the convex area of the total foot. During this phase the right leg passes the left leg.
4. *Single Support Phase - Heel-off:* as the heel of the left foot looses contact to the ground during the heel off event the support area decreases to the convex area of the toes of the left foot.

As the heel strike of the right foot occurs, the cycle is repeated where left and right are interchanged. In addition to those support phases other special support phases are experienced e.g. when standing still with parallel feet and when initialising or ending walk.

- *Basic Double Support Phase:* the feet are parallel and they are both in contact with the ground. This phase is experienced while standing still.

- *Double Support Phase - Flat foot / Heel-strike:* In this phase one foot has ground contact with the whole foot. The other foot only has contact with the heel. This phase is experienced when the heel-strike occurs for the swinging leg before heel-off has occurred on the standing leg. During initiation of walk this phase could be experienced.

Distance Measures

Until now the main focus has been to investigate how the humans move during walk and how the contact between the subject and the ground is described. In addition valuable information about the human gait can be identified by analysing the footprint of a human.

From figure 2.5 several distance measures are identified. The *Stride length* is the distance travelled by the human during one gait cycle and is measured as the distance (in the direction of movement) between two successive heel strikes of the same feet. The *Left step length* is the distance between a heel strike on the right foot and the following heel strike on the left foot (and the other way around for a *Right step length*. Two step lengths (left + right) therefore make up one stride length. For normal healthy persons the left and right step length is approximately the same, but can become asymmetric if one leg is injured or has a stiff joint.

The *step width* is as shown in the figure the transverse distance between the left and right foot. For normal gait this distance is a few centimetres but for people with balancing problems it is up to 20 cm. The last measure is the angle of the foot relative to the direction of movement.

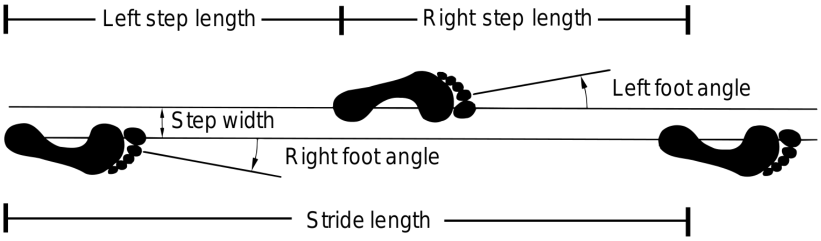


Figure 2.5. Distance parameters identified from a person's footprint. This illustration is borrowed from [7].

In addition to the distances identified from figure 2.5 there is a *step height* parameter. This parameter is also referred to as the ground clearance and describes the distance between the ground and the lowest part of the swinging foot.

Limb and Torso Movement During Gait

During gait not only the legs are used, but also most of the upper body and extremities. According to [9, Chap 4] these movements are necessary to compensate for the up and down, sideways and rotational oscillation such that the energy required to walk is reduced. The up and down oscillations results in some high and low points. The high points occur as previously described during single support near mid-stance and the low points occur when the distance between the feet are greatest e.g. in double support. To reduce the amplitude of the oscillation three different actions occur:

- *Pelvic tilt:* In stance phase the pelvis tilts 4 °-6 ° downwards away from the leg in stance phase towards the swinging leg. The angle is at maximum during midstance. The displacement occurs in the hip joint and has the effect of flattening out the high points of the oscillation. As an effect of the tilt of the pelvis the knee of the swinging leg must flex to make ground clearance. For females the pelvis is tilting more than for males. This tilt is apparent to a greater extent for females mainly due to a wider pelvis.
- *Pelvic rotation:* Around the vertical axis the pelvis rotates 3 °-5 ° to left and right from the centre position during normal gait and is at the maximum at heel-strike. The rotation of the pelvis reduces the vertical oscillation of the centre of mass. In addition this rotation effects in an increased stride length. The amplitude of the rotation increases with an increase of walking speed due to the increased stride length.
- *Knee flexion:* During mid-stance the supporting legs knee is flexed to flatten out the high points on the oscillation. At heel-strike the knee joint is almost fully extended and begins slowly to flex until midstance occurs where the knee is flexed about 15 °. After midstance the knee is extended until push-off and swing where the knee is flexed to avoid contact with the ground during the swing phase.

In addition to those, several other actions can be identified that helps creating a more fluid gait.

- *Lateral sway:* During gait the upper body sways to move mass closer to the weight bearing leg. During a whole stride the sway result in a movement of the centre of mass of about 5 cm, and with an increased step with the lateral sway is increased and vice versa.
- *Arm swing:* To counteract the angular momentum created by the movements of the legs and rotation of the pelvis the arms are swung. These rotations are such that the maximum forward arm swing occurs in the opposite side of the swinging leg and backward arm swing occurs in opposite to the leg in stance phase.

2.2 Balancing Points

In the field of humanoid robotics, one of the main objectives is to keep the robot in an upright position keeping balance i.e. not falling over the edge of the foot. In the literature several different points have been defined and by inspecting where they are located with respect to the PoS it can be determined if the robot is balancing or falling. In this section the points of interest to the method proposed in this thesis will be described.

2.2.1 Centre of Mass

The first point considered is the Centre of Mass (CoM). For a static system the CoM can be used to see if an object or collection of objects is in balance. For an object $\mathcal{O}$ consisting of N

particles, the centre of mass is defined as the mapping $\mathbf{c}:\mathscr{O}\to\mathbb{R}^3$, given by [10, p. 271]:

$$\mathbf{c} = \frac{\sum_{p_i\in\mathscr{O}} m_i\mathbf{r}_i}{\sum_{p_i\in\mathscr{O}} m_i} \qquad [\text{m}] \quad (2.1)$$

Where:

$p_i = (m_i, \mathbf{r}_i)$ is the i'th particle given by its mass and position

$m_i \in \mathbb{R}$ is the mass of the i'th particle $[\text{kg}]$

$\mathbf{r}_i = \left[x_i\ y_i\ z_i\right]^T \in \mathbb{R}^3$ is the position of the i'th particle [m]

Knowing the mass and position of the CoM of each individual part, the CoM of the robot can be calculated assuming each part as a particle.

Projecting the CoM to the ground generates another balancing point; the so called Ground Projected Centre of Mass (GCoM). The GCoM point is used to examine the CoM position in the transverse plane only. The GCoM is calculated in exactly the same way as the CoM, ignoring the z-component. From a know CoM the GCoM is found as the mapping $\mathbf{gc}:\mathbb{R}^3\to\mathbb{R}^3$, given by:

$$\mathbf{gc} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{c} = \begin{bmatrix} c_x \\ c_y \\ 0 \end{bmatrix} \qquad [\text{m}] \quad (2.2)$$

As for the CoM the GCoM is only telling about balance if the system is static. If the ground projection of the CoM is within the PoS then the system is under static balance [6]. A gait where the GCoM is within the PoS at all time is called *static gait* [11].

2.2.2 Zero Moment Point

To handle balance of moving objects other balance points are identified. The criterion for keeping balance with a moving system is called dynamic balance and is specified in the following way [6]: The humanoid gait is dynamically balanced *if there is no rotation of the supporting foot (or feet) about its (or their common) edge during walking.* An often used way of determining if dynamic balance is obtained is by calculating the ZMP, which is defined by [6] in the following way:

> **ZMP:** *the point on the support area (excluding the edge) for which* $\sum M_x = 0$ *and* $\sum M_y = 0$

Where $\sum M_x$ and $\sum M_y$ denotes the sum of all contributions to the moments about the x-axis and y-axis respectively. If this point is kept within the PoS no rotation of the supporting foot will be experienced hence dynamic balance is obtained. From the definition it is seen that the sum of the moments about the z-axis does not necessarily need to be zero. This is the case since the moment is compensated for by the friction between the foot and the ground.

From the definition of Zero Moment Point (ZMP) it is seen that it only exists inside the support area. If the ZMP enters the edge of the support area the humanoid will start falling about that edge and by definition ZMP does not exist anymore. It is however convenient to allow the ZMP to leave the support area to be able to analyse balance even when falling and therefore the Fictitious Zero Moment Point (FZMP) is defined. FZMP is a point outside the support area where the conditions for the ZMP holds, namely: $\sum M_x = 0$ and $\sum M_y = 0$. Besides fictitious this point is in literature also called imaginary or virtual ZMP.

For a gait where the ZMP is within the PoS at all times but the GCoM in some periods leaves the PoS is called *dynamic gait* [11].

2.2.3 Centre of Pressure

Another point of interest in terms of dynamic balance is the Centre of Pressure (CoP). It relates to the force between the foot and the ground, and does therefore only exist when ground contact exists. It is defined as the point where the resultant of the ground reaction force acts [12]. For a set $\mathscr{P}$ consisting of N contact points between the foot and the ground, the centre of pressure is defined as the mapping $\mathbf{cp} : \mathscr{P} \to \mathbb{R}^3$, given by:

$$\mathbf{cp} = \frac{\sum_{p_i \in \mathscr{P}} f_i \mathbf{r}_i}{\sum_{p_i \in \mathscr{P}} f_i} \qquad [\mathrm{m}] \quad (2.3)$$

Where:

$p_i = (f_i, \mathbf{r}_i)$ is the i'th ground contact point

$f_i \in \mathbb{R}$ is the ground normal force at the i'th contact point [N]

$\mathbf{r}_i = \left[x_i \; y_i \; z_i\right]^T \in \mathbb{R}^3$ is the position of the i'th contact point [m]

It is shown by [13] that the CoP is equal to the ZMP when the points are inside the PoS. Being able to measure the ground normal forces at all contact points with the ground creates an easy way to measure the position of ZMP/CoP.

2.3 The AAU-BOT1

This section introduces the instrumentation of the AAU-BOT1. A summary of the technical specifications mentioned in this section can be found in appendix A.

2.3.1 Gears, Motors, and Motor Drivers

To make the 17 actuated joints move, electrical motors apply torque to each joint through a gear system shown in figure 2.6. As seen in the figure the torque is transferred from the electrical motors to the harmonic drive gear via a belt. The harmonic drive gear transfers the torque to the joint to make it move. The construction with different size of the cog wheels on the motor shaft and on the shaft to the harmonic drive gear implies a gear ratio, this construction will be referred to as the spur gear.

The type of the harmonic drive gear varies from joint to joint depending on the gear ratio needed and the available space. Due to the varying need of torque in the different joints three different sizes of motors are used: 60 W, 90 W, and 120 W. In places where a large amount of torque is needed, two electric motors are used to drive the same joint as shown in figure 2.6. Using two motors instead of one doubles the available torque. This is used in the hip roll, knee pitch, and ankle pitch joint. In situations where the torque needed exceeds the nominal maximum torque of the motors, the motors may according to the manufacturer be overloaded repeatedly up to five times the nominal current for shorter periods [14]. In table A.1 in appendix A the motor and gear type is shown alongside with the gear ratio of the Harmonic Drive gear, the gear ratio due to different sizes of cog wheels, and the total gear ratio. The motors are from the *Maxon RE series* and the harmonic drive gears are from the *Harmonic Drive AG CPU-S series*.

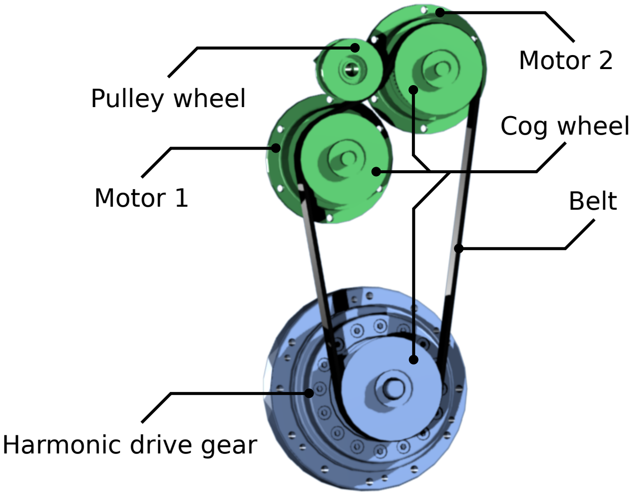


Figure 2.6. Sketch showing the assembly of the gears in a joint of the robot.

The current to each motor is provided through an EPOS of type *Maxon EPOS 70/10*. These motor controllers have an RS-232 and Controller Area Network (CAN) interface. While the EPOS has several different control modes (position, velocity and current), only the current control will be used. This is due to the fact that if position or velocity control is used, this causes problems for the joints that are double actuated. When angle readings on the two motors are not identical, they can start working against each other. Instead joints that are double actuated will only utilise sensor readings from one of the motors, and controlled as though it was one. The second motor will run in a slave configuration where the same current will be applied to the two motors.

2.3.2 Posture Determination

In order to be able to determine the posture of the robot, the angle of all the joints must be known at all times. This is measured through the EPOS's that have the possibility to receive information about angle position in two ways. Either via the *Encoder* input or *Hall Sensor* input. On the AAU-BOT1 both are used to gain precise information about joint angles. To achieve absolute position of the joints, a hall sensor is attached to each joint and returns a voltage that indicates the absolute angle. Besides from these, an encoder is attached to all motors that through the EPOS can return a precise, but not absolute, angle.

The hall sensor is of the type *Honeywell HRS100-series* and is placed in all joints of the robot except the two shoulder joints and the yaw joint of the pelvis and the hips. The voltage from the hall sensor is sampled in the 10 bit ADC in the EPOS. The absolute angle measurement is used to limit the operating range of each joint. By adjusting two potentiometers, a maximum and minimum trigger level can be adjusted that shuts off the current to the motors through the EPOS's at a specified minimum and maximum angle. This introduces a safety feature that partly protects the robot from destroying itself. Assuming that the ADC in the EPOS uses its full 10 bit resolution to sample the output voltage and maximum operating range of the Hall effect sensor of 90 °, the theoretically possible resolution of the angle can be calculated as:

$$Res_{Hall} = \frac{90}{2^{10}} = 0.0879\,^{\circ}/\text{bit}$$

The encoders placed on the drive shaft of each motor, offers a much higher total joint resolution, but this angle is only relative to the start position. Additionally it should be noted that since the encoder can only return joint angle by dividing motor rotation with gear ratio, some uncertainties arise in terms of elasticity and backlash in the gears. The encoders are of the type *Maxon motor encoder MR Type L, 512 CPT 3 Channel* which have a resolution of 2048 counts per revolution. Due to gear ratios that vary from 125:1 to 360:1 depending on the joints, the relative precision varies from joint to joint. The minimum resolution can be stated as:

$$Res_{\text{enc-min}} = \frac{360}{n_{\text{cnt}}} N_{\text{min}} = \frac{360}{2048}\frac{1}{125} = 0.0014\,^{\circ}/\text{tick}$$

where:

$n_{\text{cnt}} = 2048\,\text{ticks/rev}$ is number of counts per revolution

$N_{\text{min}} = \frac{1}{125}$ is minimum gear ratio

The ticks from the encoders are sampled in the EPOS's, which also keeps track of the relative angles in ticks from a defined zero position. In addition to the angle measurement, the EPOS provides a measurement of the angular velocity of the shaft. Since the minimum change in angle is a tick, the precision of the angular velocity provided by the EPOS can be calculated as:

$$Res_{\text{vel-min}} = \frac{Res_{\text{enc-min}}}{Ts} = \frac{0.0014}{0.001} = 1.4\,^{\circ}/\text{s}$$

where:

$Ts = 0.001\,\text{s}$ is the sampling time used in the EPOS

2.3.3 Inertial Measurement Unit

In order to give the AAU-BOT1 a sense of balance, the robot is fitted with an Inertial Measurement Unit (IMU) of the type *Xsens MTi - miniature gyro-enhanced Attitude and Heading Reference*. This unit can measure accelerations, rate of turn, and magnetic field on all three axis. An internal sensor fusion algorithm combines all the sensor data to improve the performance of the outputs e.g. by removing drift caused by temperature changes. The final filtered output of the IMU is an orientation, acceleration, rate of turn and magnetic field all in three dimensions. The readings from the IMU are transferred to the Onboard Computer (OBC) via a RS232 connection. The key specifications of the IMU are given in table 2.1.

Reading	**Spec.**
Static accuracy (roll/pitch)	< 0.5°
Static accuracy (heading)	<1°
Dynamic accuracy	2° RMS
Range (pitch)	±90°
Range (roll/heading)	±180°
Rate of turn (scale)	±300°/s

Table 2.1. Singled out specifications of the IMU.

2.3.4 Force Torque Sensor

Another important skill to adapt from humans is the ability to feel CoP with our feet. From this skill humans are able to determine if movement is necessary in order to maintain balance. This skill is very important and has been implemented on the robot with a Force Torque Sensor (FTS) in each ankle. By measuring forces and torques acting on the feet it is possible to determine how each foot supports the robot. The placement of the FTS in the ankle can be seen in the left side of figure 2.7.

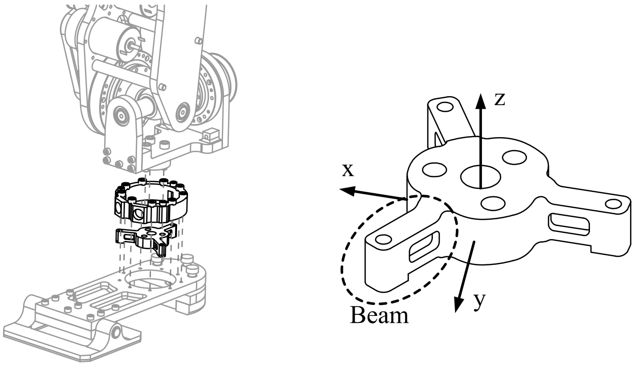


Figure 2.7. The placement and construction of the FTS [5].

To allow the FTS to measure forces and moments, a design with three beams attaching the AAU-BOT1 leg to the foot is used. A sketch of this can be seen to the right in figure 2.7. From strain gauges placed on the three beams, shearing and bending forces are recorded and used to calculate forces and moments. A more detailed description of the FTS design can be found in [3].

The sensor readings of shearing and bending forces can be converted through a linear transformation into forces and moments in a local FTS frame with origin in the centre of the

FTS, x-axis pointing towards the toe of the foot, and z-axis pointing upwards:

$$\mathbf{s} = \mathbf{C}\,\mathbf{v}$$

$$\begin{aligned} &\text{where:} \\ \mathbf{s} = [F_x\ F_y\ F_z\ M_x\ M_y\ M_z]^T\ &\text{is a output vector with forces } \mathbf{F} = [F_x\ F_y\ F_z]^T \\ &\text{and moments } \mathbf{M} = [M_x\ M_y\ M_z]^T \\ \mathbf{v} = [v_{s1}\ v_{b1}\ v_{s2}\ v_{b2}\ v_{s3}\ v_{b3}]^T\ &\text{is a measurement vector with shear forces } \mathbf{v}_s = [v_{s1}\ v_{s2}\ v_{s3}]^T \\ &\text{and bending forces } \mathbf{v}_b = [v_{b1}\ v_{b2}\ v_{b3}]^T \\ \mathbf{C} \in \mathbb{R}^{6\times6}\ &\text{is a linear transformation matrix} \end{aligned}$$

The linear transformation matrix **C**, which will be denoted the calibration matrix, is determined by applying different loads to the FTS and recording the resulting sensor readings. The theory and approach behind this is described in appendix B.

2.3.5 Onboard Computer

The OBC is the brain of the AAU-BOT1. This computer is responsible of running the controllers used to make the robot walk and is mounted on the back of the robot. All the EPOS's are connected to this computer through their CAN interface. The IMU and FTS sensor data is also received at this computer to provide information about the current state of the robot. The OBC is accessible through an Ethernet connection, which is used to upload new files and remote control the system.

2.3.6 AAU-BOT1 Setup

In figure 2.8 the whole setup around the AAU-BOT1 is shown. At this stage the robot does not carry its own power supply. Instead power to the OBC, sensors, and EPOS's is provided by three different external power supplies. At this time the AAU-BOT1 is therefore always connected to its surrounding setup by four cables. One Ethernet cable and three power cables. The laptop is used to send orders to the OBC while the AAU-BOT1 is active. The desktop computer is used to upload new software to the OBC and give order to start up and listen for commands send from the laptop. The winch is operated by a completely separate circuit and is used to hoist the robot off the ground. The emergency stop can be used to manually cut off power to the robot in case of any unwanted or unexpected behaviour. More detailed information about the surrounding setup can be found in the previous thesis [2] appendix K *Users Manual for AAU-BOT1 Laboratory*.

2.4 Products of Previous Work on the AAU-BOT1

At the time of this thesis beginning the AAU-BOT1 is fully assembled and instrumentation is completed. In addition drivers have been written and a MATLAB® Simulink interface created. Using this interface, controllers can be implemented directly in Simulink and converted to a program that is executed on the on board computer to run the implemented controllers.

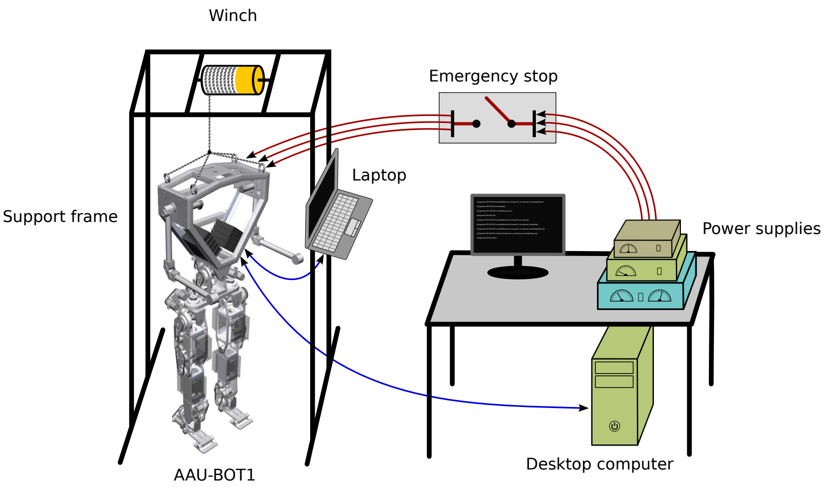


Figure 2.8. Sketch of the whole AAU-BOT1 setup including peripheral equipment. Blue lines indicate Ethernet connections. Red lines indicate power connections.

During operation it has been observed that some of the hardware drivers returns an error and stops working. These breakdowns does however not happen very often and it has been chosen to ignore these. Despite the driver flaws the robot is ready for operation when calibration of sensors has been carried out. The following products of previous works are available at the beginning of this thesis:

Robot and instrumentation: The AAU-BOT1 with fully instrumented with actuators, sensors, and on-board computer. The On-board computer is installed with Debian 5.0 patched with RTAI to obtain real time capabilities in user space [15].

Hardware drivers: Drivers for all interfaces available in the on-board computer. The main drivers are for the CAN interface to connect to the EPOS's and for the serial I/O card to interface the IMU and FTS.

Simulink interface: Provides Simulink blocks that interfaces to the hardware drivers. The programs implemented in Simulink using this interface are converted into C source code using the Real Time Workshop library[1]. This source code is transferred to the on board computer, compiled, and executed to use the implemented controller.

Operator console: Interface for the operator to control the AAU-BOT1. This includes starting capture of data from the IMU and FTS and controlling the EPOS to move the robot joints to the initial position. Additionally the operator console provides functionality to start a program to control the robot that has been sent to the on board computer from an external computer.

Simulation model: A simulation model is set up in Simulink that uses the SimMechanics library. Using this library, the dynamic effects is modelled using only the physical dimensions of the robot and the knowledge about the mass and moment of inertia of the individual body-parts. The simulation model includes the foot-ground interaction and the joint friction.

Visualisation tool: The SimMechanics library used to simulate the AAU-BOT1 provides a visualisation tool. Using this visualisation tool however slows down the simulation significantly. As an alternative a visualisation tool called *Palantir* has been developed that

[1] The predecessor for the now available Simulink Coder

that can be executed on an external computer and get information about the posture of the robot via either TCP or UDP.

For a deeper knowledge about the whole system, the reader is encouraged to study previous reports written on the AAU-BOT1. For a deeper insight in the design of the robot refer to [3]. More information about final preparation and changes to the original hardware refer to [4]. This thesis also explains the initial software structure and driver implementation.

The first version of the Simulink interface is described in [5]. This thesis also describes the operator console, a simulation model, and a visualisation tool. These tools where however been redesigned by the next group working with the AAU-BOT1. For information about the newest implementation of these tools refer to [2]. The simulation model created by [2] has been revised and updated during the fall semester 2012, by an exchange student who worked with modelling, control, and simulation of the AAU-BOT1. The updates made during this project are described in [16].

In addition to those tools the previous groups working on the AAU-BOT1 has developed different control strategies for controlling the AAU-BOT1. The following report describes an attempt towards making the AAU-BOT1 perform static gait [5]. The first and so far only report in the AAU-BOT project working with achieving dynamic gait is [2].

CHAPTER 3

OVERALL APPROACH

This chapter introduces the overall approach to obtain dynamic gait with the AAU-BOT1. Various approaches are discussed and the overall controller structure developed in this project is presented. Finally the results of an experiment concerning recording of human gait in a motion tracking room is presented.

Section 3.1 introduces the idea of splitting the responsibility of walking into a two different levels of controllers. A low level controller ensuring desired posture and a higher level controller to obtain gait and balance by generating the postures. In section 3.2 different approaches for generating gait trajectories will be discussed. Different controller structures for the low level controller are presented in section 3.3. The chosen overall controller structure for the AAU-BOT1 is presented in section 3.4. Section 3.5 presents the results of the experiment analysing the human gait.

Chapter Content

3.1 Proposed Controller structure

To obtain gait with the AAU-BOT1, an overall structure as shown in figure 3.1 is proposed. The input to the system is a specification of the desired gait velocity and the output is the AAU-BOT1 performing gait. In this thesis only gait in a straight line will be treated, but at a later stage the input should also include a direction of movement.

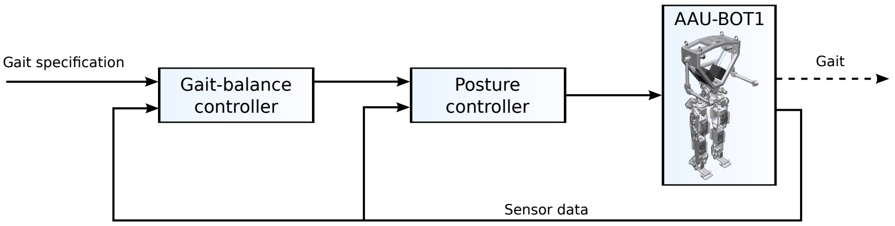


Figure 3.1. The proposed overall controller structure.

The task of controlling the robot is divided into two overall tasks. These two tasks are illustrated as the two controllers shown in figure 3.1. The inner controller, called posture controller, has responsibility of controlling each joint of the robot to a reference. It receives joint angles measured in the AAU-BOT1 as feedback. The outer controller is the gait-balance controller and serves the purpose of generating the joint references to the posture controller that makes the AAU-BOT1 walk while maintaining balance. This controller uses joint angles, IMU, and FTS sensor data as feedback to determine where ZMP and CoM is located with respect to the robot.

Different strategies for the gait-balance controller and posture controller will be discussed next. The work of previous project groups on the AAU-BOT1 within these topics has been studied. Earlier groups have proposed various solutions for both gait, balance, and posture controllers. The gait controllers have been ranging from guessing on multiple trajectories and selecting the one that is most stable, to generating trajectories from inverted pendulum models. Posture controllers ranging from LQR controllers controlling all joints with one controller, to simpler PI controllers with one controller for each joint have been proposed.

Due to the limited success in terms of achieving gait with earlier proposed solutions, it has been chosen to look with fresh eyes on the topic and investigate new approaches. The previous group on the AAU-BOT1 project have put effort in implementing and using a SimMechanics simulation model. This project will however focus more on testing strategies directly on the robot. The SimMechanics model is very computational heavy to execute and simulating a few seconds of motion takes hours of simulation time. Furthermore the model has shown behaviour different from the AAU-BOT1 even during simple experiments. For these reasons the usage of this model has been very limited.

3.2 Gait-balance Controller

The main functionality of the gait-balance controller is to generate joint references to the posture controller. These references should be generated in a way such that the robot is maintained in an upright position and the gait specifications are fulfilled. This task can be fulfilled in several ways. In this section some different strategies will be presented and discussed.

The joint trajectories can in general be grouped into either online or offline generated. All joint trajectories do however not necessarily need to be defined in the same way. Some of the joint trajectories could be generated offline while others could be generated online. It is also possible to create a reference from a combination of online and offline generated trajectories.

For offline generated trajectories the trajectories are calculated beforehand and stored such that they merely should be played back in runtime. This has the major advantage that it does not require much computational power in runtime. The main disadvantage of this method is that this approach is very inflexible. An example of this is that changing walking velocity requires different trajectories for different velocities. Furthermore with this approach external disturbances can not be taken into consideration and robustness of the final system is low.

For the online generated trajectories the case is the opposite. Since they are generated online they are very flexible as they can be changed in runtime. On the other hand they require more processing power which could be an issue. Revisiting the example of changing walking velocity, the online generated trajectories could be generated as a function of the walking velocity and thereby easily adapted to different velocities. Four different approaches that have been found in literature about walking robots will be discussed next.

Mimic the Human Gait

A simple approach suggested by a previous AAU-BOT group [2] is to generate joint trajectories by replaying motion captured during human gait. By recording the movement of human joints during at least one gait cycle, a static gait cycle can be found. By following trajectories defined in this way, the walk will become anthropomorphic, but the gait will not necessarily be stable nor energy efficient since the gait is optimized for the recorded human and not for the robot. Since the AAU-BOT1 does not have the exact same dimension as the human, some extra energy must be added to compensate for the differences.

Even though the recorded human maintains balance during the gait, this does not ensure that the robot will do so as well. Important parameters such as shock absorption, limp length and mass distribution will be different for the robot and the human. To maintain balance, the reference to some of the joints must be modified by a balance controller, that tries to keep the robot in an upright position for example by controlling the ZMP to be within the PoS.

Mimicking human gait is a simple approach but it has several inconvenient properties. The main disadvantage is that it is very inflexible. Creating different gait velocities could require many different recordings of human gait. Since the balance controller only has control over a limited set of the joints, it can be difficult to achieve robust gait that can handle unforeseen disturbances.

Passive Dynamics

The concept of passive dynamic walkers was invented in 1888 by George T. Fallis [17]. He invented a two legged toy that was able to walk down a slope by rocking from side to side without any actuation. Even though this is not very human like it has inspired several scientists to develop passive walkers. In 2001 Collins et al. [18] developed a passive dynamic walker with knees. The single most interesting thing about passive dynamic walkers is that they are very energy efficient. They preserve most of their forward momentum from one step into the next. Placed on a slope they are capable of walking down along it without any actuation. The addition of knee joints make them walk in a way similar to humans.

From models of passive walkers, trajectories for some of the joints in the AAU-BOT1 can be generated. Generating trajectories in this way the gait becomes anthropomorphic and from the nature of passive dynamic walkers also energy efficient. As for the recorded human gait, the walk will be less efficient for the AAU-BOT1, since the structure of the AAU-BOT1 is not as the passive dynamic walkers. One big difference is the construction of the feet. For the passive dynamic walkers the feet are rounded on the bottom side such that most of the forward energy is maintained which is not at all the case for the AAU-BOT1.

Using this way of generating references for the joints ensures the cyclic movement, but as for the approach where the human gait is mimicked, a balance controller must be added to ensure that the body is maintained in a upright position.

ZMP control

An approach for creating online generated trajectories is to move PoS with respect to the ZMP. This approach is often executed by defining a trajectory for the ZMP or CoM of the biped robot and keeping balance by keeping ZMP inside PoS [19]. By knowing the position of the ZMP, the feet can be controlled such that the ZMP follows a trajectory while balance is ensured. This is a relatively dynamic approach that allows the robot to cope with large external perturbations. If external forces are suddenly applied to the robot, the ZMP will move and a new placement of the feet will be attempted by the controller, to catch the robot before it falls over. How well this approach works comes down speed and excess torque of the joints.

In order to make this method work efficiently different strategies have been imposed. One is to predict placement of the feet some steps ahead of time to improve stability [20]. It is important to note that while in single support phase, only placement of one of the feet can be selected. For large perturbations it might be necessary to perform several steps in order to regain balance. The foot placement can either be found by trying to maximise the stability of the robot or trying to minimise the energy consumption used to walk.

Central Pattern Generator

The Central Pattern Generator (CPG) of a human is the part of the nervous system generating patterns for movement of the body. The pattern generator is capable of generating rhythmic patterns without need of any input from sensors. Having such rhythmic patterns, higher level centres are capable of generating rhythmic movements of the body parts by modulating the signals from the CPGs. Even though the rhythmic patterns are generated without any sensory feedback, the patterns are shaped by using feedback and in this way adapting the patterns to the environment [21].

The idea of having CPGs and modulating their output has inspired several scientists to develop something similar to generate trajectories for humanoid locomotion [22, 23, 24]. The main idea is to use a number of oscillators, which properties are described by a set of parameters. The outputs of the oscillators are then used to synthesize references for the joints in the human body. The task in using this strategy is to find the parameters that make the final output most anthropomorphic. One way of finding the proper parameters could be to use some training data recorded from a human walking, and choosing the oscillator parameters such that the synthesized output is as close as possible to the training data.

The main disadvantage in using this strategy it the relative high complexity. The main advantage is the ease of changing the gait by changing the parameters of the oscillators and the mechanism used to synthesize the references from the oscillator output.

3.3 Posture Controller

The goal of the posture controller is to track references given by the gait-balance controller. It is vital to the final performance of the robot that the posture controller performs well. It has been observed by previous groups that a large amount of friction is present in the joints of the AAU-BOT1 [2]. The posture controller should therefore be designed such that it can cope with disturbances introduced by friction.

Friction is however not the only issue of achieving good performance in the posture controller. Since all the joints are mechanically linked together any movement of one joint will introduce a load on all the other joints. Considering 17 individual joints moving at the same time, these cross couplings becomes computational heavy to calculate. A natural way of work around this problem is to disregard the cross couplings and other disturbances such as gravity. Instead the posture controller must be designed in such a way that it can move accurately even under the influence of these disturbances.

A vast amount of different more or less complicated controllers have been proposed for position control of robot manipulators [25]. Previous groups on the AAU-BOT1 project have worked with both simple PI-controllers, as well more advanced strategies like computed torque controllers, and LQR. The posture controller implemented by the previous group working with the AAU-BOT1 was available and implemented at beginning of this project. This posture controller consists of individual PI controllers in each joint of the robot. Initial tests have however shown that the performance is not satisfactory. More on this topic will be presented in chapter 5.

Other controller structures such as a cascade controllers often results in good performance. By letting one controller provide the reference for another controller in the inner loop, a faster response to changes e.g. caused by disturbances can be achieved. This reduces the influence of disturbances and could be an effective simple solution that will be considered.

Computed torque has shown very good performance within the area of robotic manipulators. Ideally this type of controller can completely remove the influence of known disturbances such as friction and cross couplings. A disadvantage is however that these torques can be difficult to calculate for a complex system.

From the area of non-linear controllers a sliding mode controller could also be of interest. With this type of controller stability and performance can be guaranteed under any condition as long as the magnitude of the disturbances are known and bounded. When the disturbances have great magnitude as it is the case with the AAU-BOT1, the sliding mode controller needs to be designed aggressively. This causes high amplitude and rapidly shifting control signals that are not ideal for being applied directly to an actuator on the robot. If the output signal generated by the non-linear controller can not be applied by the actuator, the design specifications, which the controller is designed from, can not be guaranteed.

The dynamics of the robot are greatly influenced by which support phase it is in. For that reason it might be necessary to switch between a range of controllers to perform satisfactory in both single and double support phase. This option is also considered if one controller can not perform accurately enough for all cases. In the design phase the simplest controllers should be attempted first. If performance turns out to be poor, gradually more complicated controllers will be attempted until a satisfactory result is achieved. Both simulations and tests on the robot will be used to determine the performance of each controller. Parameters like precision, disturbance rejection, and computational load will be evaluated.

3.4 Chosen Controller Structure

From the discussion of the approaches for generating trajectories the overall strategy for the gait-balance controller is settled. Due to the large flexibility of the ZMP control approach it is chosen to use this approach as starting point. Using this approach a ZMP reference is determined that allows for gait with the desired specification. From the ZMP reference the body movements are then generated to obtain the desired gait. In figure 3.2 the responsibilities of the gait-balance controller have been divided in to several different tasks. A short description of each function block will be presented next.

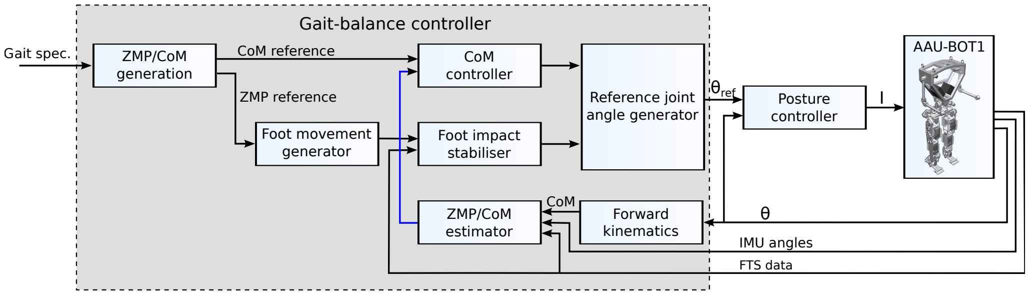


Figure 3.2. The controller structure devised to enable gait with the AAU-BOT1.

ZMP/CoM generation: From the gait specification, a ZMP and CoM reference is created. These references specify where the ZMP and CoM of the robot should be located, in order to make the AAU-BOT1 walk with the specified gait velocity, while maintaining balance.

Foot movement generator: In order to obtain gait with the robot, foot movements must be specified. This block specifies exactly how each foot should be lifted and put in front of the other to walk.

Foot impact stabiliser: The legs of the AAU-BOT1 are almost completely rigid when the legs are held in a specific posture by the posture controller alone. When a foot impacts the ground high force transients are experienced with a leg controlled solely by the posture controller. To reduce these shock absorption is implemented in the legs by modifying the foot position references before they are given to the joint angle generator.

CoM controller: To keep the robot in balance, it is important that the ZMP and CoM trajectory generated from the gait specification are followed accurately. This task is taken care of by the CoM controller.

Reference joint angle generator: This block collects inputs from the CoM controller and the foot impact stabiliser, and calculates the necessary joint angles through inverse kinematics to obtain the desired CoM and foot motion.

Forward kinematics: From the measured joint angles on the robot, the forward kinematics is used to calculate where the CoM is located.

ZMP/CoM estimator: To achieve a sense of balance the ZMP and CoM is estimated from measurements from the FTS's and IMU, and from the calculated CoM via the kinematics.

Posture controller: To make sure that all joint references are followed closely, the posture controllers provide current references to the individual joint motors.

The basic controller structure has been settled, but the exact movements of the robot can still be performed in numerous ways. To gain insight in how humans walk at different gait velocities, a test where human gait is recorded in a motion tracking room is performed. This test and its results will be discussed next.

3.5 Human Gait Parameters

To gain insight in the movement of humans during gait, three test subjects are recorded in a motion tracking room, walking on a treadmill. The objective of this test is to deduce main pattern of movement during gait, along with key parameters of the gait. This among other things includes step time, step length, and double support ratio. The gait is recorded at different gait velocities to study how the gait pattern changes.

From only three test subjects, it will not be possible to deduce any certain results. But having raw recordings of human motion gives a vast amount of possibilities in terms of analysing different parts of the gait. This will be used for generating human inspired gait with the AAU-BOT1. The parameters extracted from the data has been compared to other studies of human gait and shown a reasonable resemblance [7].

The three tests subjects consist of two female and one male. A description of the experiment is given in appendix F. In table 3.1 the extracted gait parameters are shown. From the recorded tests it is chosen to extract parameters for gait velocity of 0.5 km/h, 1 km/h, and 3.6 km/h. Video recordings from the experiments can be found on the enclosed DVD 💿 `/videos/mt_*.mp4`. In the following sections elaboration are made on the extracted parameters. In some of the elaborations the relationship between CoM movement and ZMP is considered. If this relationship is not clear for the reader refer to section 4.3.3 where correspondence is shown through the cart-table model.

Parameter		**Female 1 (A)**			**Female 2 (N)**			**Male**		
Gait velocity	[m/s]	0.14	0.28	1.00	0.14	0.28	1.00	0.14	0.28	1.00
Step time	[s]	1.53	1.22	0.64	1.99	1.31	0.60	1.82	1.24	0.59
Step length	[mm]	212.36	338.51	640.61	275.83	365.05	598.73	252.46	345.01	590.67
Step width	[mm]	147.58	128.72	111.09	95.35	95.03	82.10	207.13	221.63	189.80
Double support ratio	[%]	58.20	57.82	43.69	66.83	56.19	28.31	56.45	54.79	30.79
Transverse pelvis amp	[mm]	54.55	39.39	19.75	38.27	30.96	11.66	91.07	87.46	39.94
Vertical pelvis amp	[mm]	3.86	5.12	10.50	2.17	3.97	9.66	3.41	4.63	8.25
Max heel lift	[mm]	103.88	142.06	194.89	114.94	154.93	195.88	116.63	149.39	196.98
Max toe lift	[mm]	29.87	35.17	60.39	25.98	33.55	32.91	26.66	34.96	43.37

Table 3.1. Gait parameters for two female and one male test subject.

Step time and length: The step time parameter is determined by measuring a characteristic point in the gait. In this case when the heel reaches the maximum elevation. The time between the maximum heel lift for one foot is determined and thereby directly the stride time. From the stride time, the step time and the step length is calculated. The step length can be calculated this way since the gait velocity is known from the velocity of the treadmill.

In table 3.1 it can be observed that the gait time, and thereby also step length is approximately the same for the three test subjects. The only big difference is that for the first female test subject the step time is lower and the step length thereby shorter. The main reason for this is assumed to be the height of the first test subject. The first test subject is shorter than the others, hence it is natural to take more shorter steps rather than longer steps.

Step width: This parameter is determined by measuring the position of the feet in the periods where the feet have ground contact. From this the mean distance between the feet is calculated.

From table 3.1 it is seen that for all test subjects increasing the velocity to 1.0 m/s reduces the step width. For the step width at the two low velocities no clear tendency exists. What is not visible from these numbers is that the test subjects move the swinging foot towards the supporting one, this behaviour is presumably exists to move the mass closer to supporting leg.

Double support ratio: The double support ratio is determined by determining the time where both feet are on the ground. Since only height measurement is available, the foot is assumed to be supporting if at least one of the foot markers are lifted no more than 4 mm from its initial height. One problem with this way of determining the support ratio is that the toe marker is actually placed on the instep of the foot rather than on the toe, this implies that the marker is elevated from the initial position without the foot is losing support.

From table 3.1 it is seen that the double support ratio clearly reduces as the gait velocity increases. This indicates that for low gait velocities the human tries to use shorter time to move the leg and then stay in double support longer until the other leg must be moved. This is supported by statements from the test subjects that in general think that walking with the two low gait velocities is quite unnatural. They had to be more aware of their gait to follow the velocity of the treadmill.

Amplitude of pelvis displacement: The displacement of the pelvis in the vertical and transverse axis is identified from the pelvis centre calculated as the mean of the two hip marker positions. The amplitude of the pelvis displacement d_{amp} in the transverse axis is calculated as:

$$d_{\text{amp}} = \frac{\sum_{i=1}^{N}(p_i^{\text{max}} - p_i^{\text{min}})}{2N} \qquad [\text{m}]$$

Where:

N is the number of strides [−]

p_i^{min} is the minimum position of the pelvis in the transverse axis in stride i [m]

p_i^{max} is the maximum position of the pelvis in the transverse axis in stride i [m]

The pelvis displacement in the vertical axis is calculated in a similar way. For the transverse movement of the pelvis it is observed from table 3.1 that increasing the velocity reduces the amplitude for all test subjects. With the faster step time the acceleration of the CoM increases hence the amplitude for the CoM movement must be reduced to keep the ZMP under the supporting foot. Another aspect is that by increasing the gait velocity the step width reduces and the amplitude of the CoM movement must therefore also be reduced.

From the table it is seen that for the second female and the male test subject their pelvis movement nearly matches half of the step width with gait velocity of 0.14 m/s. Hence the centre of the pelvis is moved close to the centre of the supporting foot. This indicates that the CoM is moved close to the supporting foot and the gait is close to static. For the first female test subject this is not the case, and the pelvis is only moved approximately 1/3 of the step width. This agrees with the observation that this test subject has a higher step frequency at low velocities compared to the other two test subject. This higher step frequency makes the gait more dynamic and the total mass should therefore not be moved as much to place the ZMP under the supporting foot.

For the vertical movement of the pelvis the general tendency is that the amplitude increases with the velocity. This is a consequence of the increased step length since moving the leg further ahead of the pelvis the height of the pelvis must be reduced.

Max heel and toe lift: These parameters are determined by taking the mean of the maximum lift of the heel and toe markers during each step. From table 3.1 it is seen that with an increasing gait velocity the lift of the heel and toe increases.

Taking the amount of test subjects into account it is difficult to conclude much from these experiments in terms of differences in the gait of male and female, since many other aspects could have made impact on the gait performed by the test subjects. The largest difference observed between the female and the male test subjects is the step width and the amplitude of the transverse pelvis movements. For the female test subjects the step width was smaller which implies that the transverse pelvis movement was smaller.

Foot Movements During Gait

To visualise how humans move their feet in the sagittal plane during gait, the position of the heel and toe marker for one foot is extracted and the position and orientation of the foot is calculated. In figure 3.3 the position and orientation of one foot is illustrated for one stride with gait velocity of 1.00 m/s.

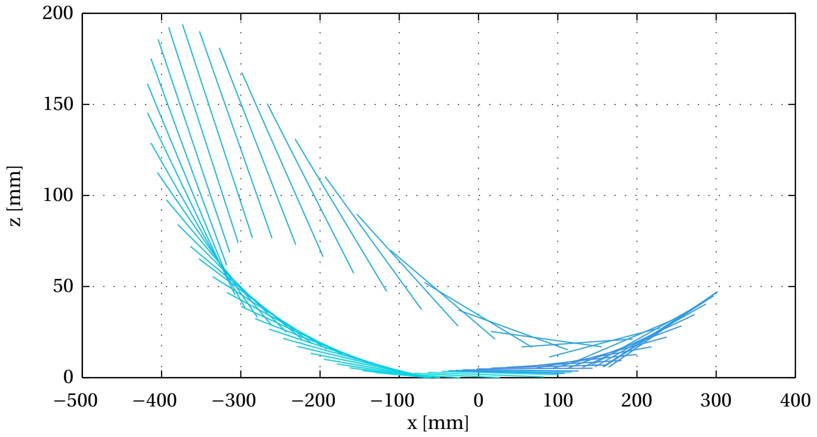


Figure 3.3. Right foot orientation and position during one stride for the male test subject at 1.00 m/s. The left most point of each line is the position of the heel. The length of each line is 150 mm. The time between each line is 0.02 s and the colour gradient from light to dark blue indicates the time.

From a flat foot on the ground the step is initiated by a heel lift. The heel lift is succeeded by a toe off whereupon the foot is further lifted and the forward foot movement is initiated. As the heel reaches its maximum height the direction of the rotation is reversed such that the heel impacts the ground first when the foot is lowered. After heel impact the foot is moved backwards and rotated to make the foot flat on the ground and the stride cycle is completed.

Transverse Pelvis and Torso Movements

To get more insight into how the CoM is moved by humans, the movements of pelvis and torso is studied in the transverse axis. For small step lengths the most dominant motion of the CoM will be in this axis. To move one foot in front of the other the weight must be shifted in the transverse axis to allow foot lift. For this reason it is chosen only to study the CoM movement in the transverse axis at this point.

By identifying the centre point of the two hip markers and the centre point between the two shoulder markers, the movement of the pelvis and torso is identified. In figure 3.4, 3.5, and 3.6 these two points are plotted for the three test subjects for gait velocities of 0.14 m/s, 0.28 m/s, and 1.00 m/s.

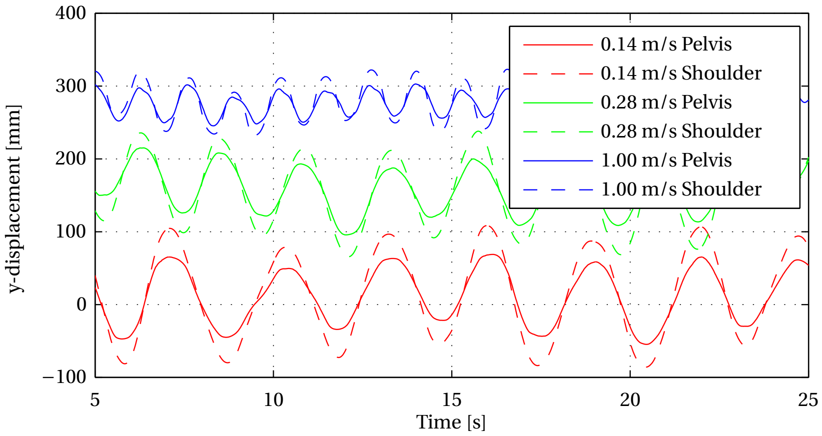


Figure 3.4. Transverse displacement of the pelvis and shoulders during gait for the first female test subject. Note that the data is offset for the two faster gait velocities to enhance visibility.

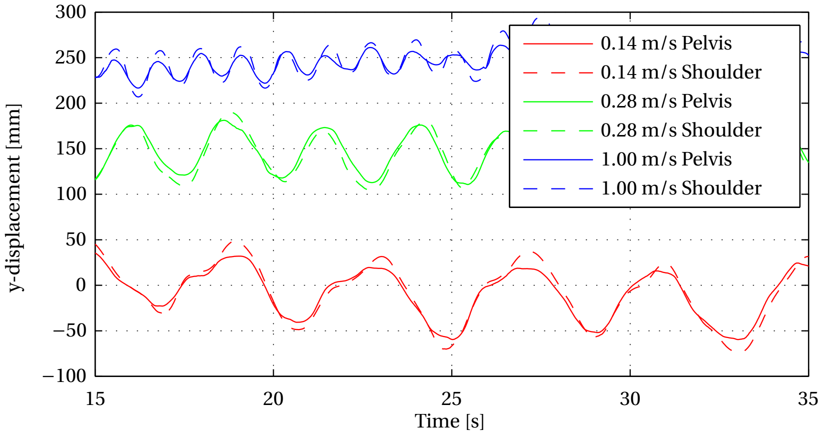


Figure 3.5. Transverse displacement of the pelvis and shoulders during gait for the second female test subject. Note that the data is offset for the two faster gait velocities to enhance visibility.

From all figures a reduction of pelvis movement at higher gait velocity is clearly visible which corresponds with the data identified in table 3.1. By comparing the figures it is observed that the two female test subjects utilise torso movements to a greater extent than the male test subject. Additionally it is observed that the usage of the torso increases with decreasing gait velocity for all three test subjects.

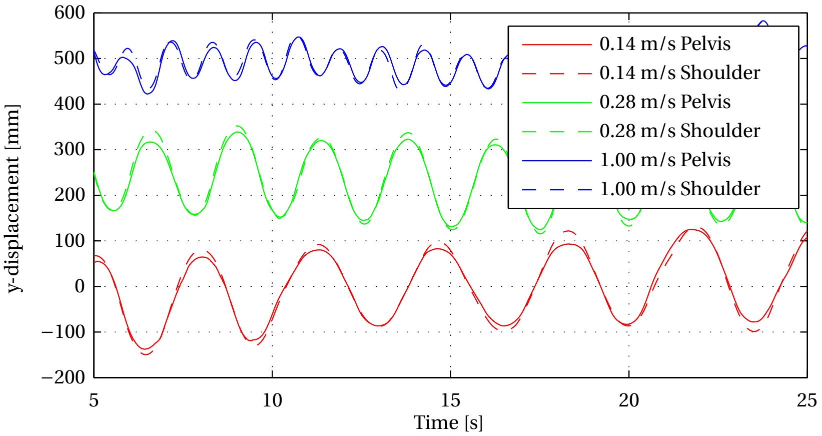


Figure 3.6. Transverse displacement of the pelvis and shoulders during gait for the male test subject. Note that the data is offset for the two faster gait velocities to enhance visibility.

In this section parameters and characteristic movements are extracted from measurement of human gait. These will be used as general guidelines in the design of the gait. In addition the raw data from the experiments will be inspected further in situations where additional knowledge about a specific movement is needed.

Chapter 4

Modelling

This chapter contains derivations of the models used in this project. Dynamic models for the joints of the robot are presented in section 4.1. In section 4.2 the forward and inverse kinematics of the robot posture is deduced. Finally section 4.3 introduces the dynamic models used to describe behaviour of ZMP, CoM, and the connection between these.

Chapter Content

4.1 Joint Model

This section describes the deduction of the model used to model the behaviour of all the joints in the AAU-BOT1. All joints consist of one or two motors, a spur gear, and a harmonic drive gear. For more information about the mechanical construction refer to section 2.3. In figure 4.1 the general model for the transmission in the joint is seen.

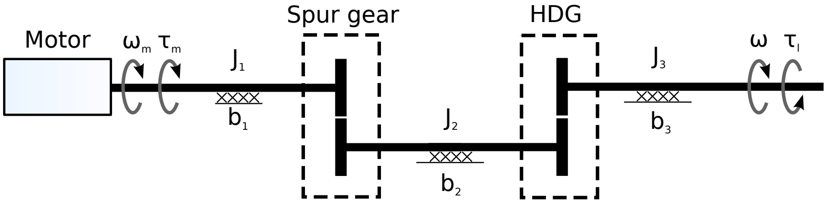


Figure 4.1. Model of the joint. HDG refers to the harmonic drive gear. ω is the shaft angular velocity, τ denotes torque, J is a moment of inertia, and b indicates friction.

In [5], it was proposed to include elasticity in the model to due to backlash and elasticity in the belts. The reason for this was that measurements from the sensors mounted on the joint diverged from the ones measured by the encoder on the motor. This behaviour could however not be reproduced which comply with the observations made by [2] for which reason elasticity is not included in the joint model.

By collecting all moments of inertia of the high velocity side on the harmonic drive gear into one moment of inertia J_m and assuming that all friction acts on the low velocity output shaft the model is simplified as shown in figure 4.2 where the gear ratio is the product of the gear ratio of the spur gear and the harmonic drive gear. The assumption that all friction acts on the output shaft is made since a friction model of the whole gear is determined experimentally. More information about the friction model will be presented in section 4.1.1.

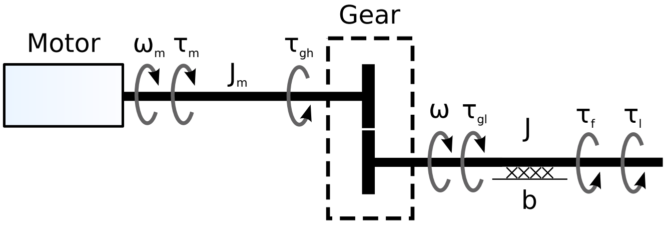


Figure 4.2. Simplified model of the joint. ω is the shaft angular velocity, τ denotes torque, J is a moment of inertia, and b indicates friction.

For the high velocity side of the gear, the dynamics are as follows:

$$J_m \dot{\omega}_m = \tau_m - \tau_{gh} \qquad [\text{Nm}] \quad (4.1)$$

Where:

J_m is the moment of inertia of the motor and the gear (see table 4.1) $\left[\text{kgm}^2\right]$

ω_m is the angular velocity of the motor shaft [rad/s]

τ_m is the torque delivered by the motor [Nm]

τ_{gh} is the torque load from the other side of the gear [Nm]

In the same way the dynamics for the low velocity side is found as:

$$J\dot{\omega} = \tau_{gl} - \tau_f - \tau_l \qquad [\text{Nm}] \quad (4.2)$$

Where:

J is the moment of inertia of the part of the robot that the joint moves $\left[\text{kgm}^2\right]$

ω is the angular velocity of the low velocity shaft [rad/s]

τ_{gl} is the torque delivered from the high velocity side of the gear [Nm]

τ_f is the torque due to friction (see section 4.1.1) [Nm]

τ_l is the load torque in the joint [Nm]

The difference in angular velocity on the two sides of the gear is given directly by the gear ratio. The case is not that simple for the transmission of torque due to loss in the gear. The equations for the velocity and torque transmission in the gear is as follows:

$$\omega = \frac{\omega_m}{G} \qquad [\text{rad/s}] \quad (4.3)$$

$$\tau_{gl} = \eta\, G\, \tau_{gh} \qquad [\text{Nm}] \quad (4.4)$$

Where:

G is the gear ratio according to table 4.1 [−]

$\eta = 0.82$ is the efficiency of the gear [1] [−]

Combining equations (4.1) to (4.4) the following dynamics for the low velocity side of the gear is found:

$$(J + J_m\, G^2\, \eta)\dot{\omega} = \eta\, G\, \tau_m - \tau_f - \tau_l \qquad [\text{Nm}] \quad (4.5)$$

Since the motors are driven by the current controller in the EPOS, the torque generated by the motor τ_m is directly calculated from the input current disregarding all the dynamics of the electrical circuits in the motor.

$$\tau_m = N_m\, k_t\, i \qquad [\text{Nm}] \quad (4.6)$$

Where:

N_m is the number of motors in the joint [−]

k_t in the motor constant [Nm/A]

i is the applied current to the motor [A]

[1] The maximum efficiency at 25°C identified in the datasheet for the gears by [5]

For the separate joints J_m, G, k_t, and N_m are shown in table 4.1. The value of J_m is calculated in [5] from the moment of inertia of the motor, spur gear, and high velocity side of the HDG under the assumption that the spur gear does not have any loss. In table 4.1 the value of $J_m\ G^2\ \eta$ is also shown to compare the size of this to the moment of inertia of the load connected to the output shaft.

Joint(s)	J_m [kgm^2]	$J_m\ G^2\ \eta$ [kgm^2]	G [−]	k_t [Nm/A]	N_m [−]
Shoulder	55.6e-7	0.071	125	53.8 e-3	1
Waist Roll	358.6e-7	2.647	300	62.2 e-3	1
Waist Pitch	461.7e-7	4.907	360	60.3 e-3	1
Pelvis Yaw	149.6e-7	1.018	288	53.8 e-3	1
Hip Roll	859.5e-7	5.369	276	60.3 e-3	2
Hip Pitch	464.8e-7	2.269	244	60.3 e-3	1
Hip Yaw	98.8e-7	0.361	211	62.2 e-3	1
Knee	520.3e-7	0.755	133	60.3 e-3	2
Ankle Roll	149.8e-7	0.491	200	53.8 e-3	1
Ankle Pitch	354.3e-7	2.810	311	60.3 e-3	2

Table 4.1. Parameters for the joints in the robot.

4.1.1 Friction modelling

Due to the construction of the joints with motors, harmonic gear drives, and spur gears some of the joints in the robot suffer from substantial amount of friction. A good model of the friction in the joints of the robot is usable both in compensating for the friction when controlling the angle of the joint but also for creating a descriptive simulation model of the robot.

The friction model that will be derived here should model the τ_f term in equation (4.5). Friction modelling is an old discipline and several different models have been proposed, in [25] a summary of different friction models is given. An often used friction model is the LuGre model, which is a dynamic friction model that includes effects such as pre-sliding displacements and hysteresis effects. The LuGre model is shown useful to model and compensate for the friction in systems with harmonic drive gears [26]. The LuGre model is therefore the obvious choice for modelling the friction of the joints in the robot. This model is however not chosen since the static friction of the critical joints is high and dependent on the angle of the joint, hence the dynamic friction is both assumed to be small compared to the static friction but also hard to identify. This assumption complies with observations made by [27]. As a first choice the simpler static friction model including coulomb and viscous friction, stiction and the Stribeck effect is used [28]:

$$\tau_f = \left(\tau_c + (\tau_s - \tau_c) e^{-\left(\frac{\omega}{\omega_s}\right)^2}\right) \operatorname{sgn}(\omega) + \mu\omega \qquad [\text{Nm}] \quad (4.7)$$

where:

τ_c is the coulomb friction coefficient [Nm]

μ is the viscous friction coefficient [Nm · s/rad]

τ_s is the stiction coefficient [Nm]

ω_s is the stribeck velocity [rad/s]

ω is the angular velocity of the motor shaft [rad/s]

To determine the parameters of (4.7) two different experiments are carried out. To determine the viscous friction the motors are made to let the joint angle follow a position ramp with a constant velocity. Two different controllers are used to control the input currents to the motors. A high gain PI controller and a PID controller with lower gains. The high gain controller is better at keeping a constant velocity at low velocities, while it is not able to obtain velocities above 15 °/s due to stability issues. Since it is desired to map the friction up to 40 °/s the PID controller with lower gains is used for velocities above this velocity.

Under the assumption that the velocity is approximately constant, equation (4.5) reduces to $0 = \eta\ G\ \tau_m - \tau_f - \tau_l$. When the torque generated by the motor τ_m and the load τ_l is known, the friction τ_f is easily calculated. The torque generated by the motor τ_m is calculated from measurements of the applied current to the motor. In the following tests the load torque will be assumed only to consist of gravity, which can be calculated from the mass and CoM of the swinging limb. To make a better estimate of the Stribeck effect, the density of measurements at low velocity is increased.

To find the stiction an open loop experiment is carried out, where the current input is slowly increased until the joint breaks free of stiction and the breakaway current i recorded. For a more thorough description of the tests and controllers, refer to appendix E.

The model parameters are found using the `lsqcurvefit()` function in MATLAB® which by default uses the *trust region reflective* optimisation algorithm to solve the following nonlinear least squares problem:

$$\min_{\mathbf{x}} \sum_{i=0}^{N-1} (\tau_f(x, \omega_i) - \tau_i)^2 \qquad (4.8)$$

Where:

$\tau_f(x, \omega_i)$ is the friction model given in equation (4.7)

$\mathbf{x} = [\tau_c\ \tau_s\ \mu\ \omega_s]$ is the friction model parameters

N is the number data points

ω_i is the angular velocity at data point i

τ_i is the torque at data point i

The resulting friction model for the left knee is shown in figure 4.3 along with the data points collected using both the high gain and the low gain controllers. Only the data points for the low gain controller is used to fit the model for reasons described later in this section.

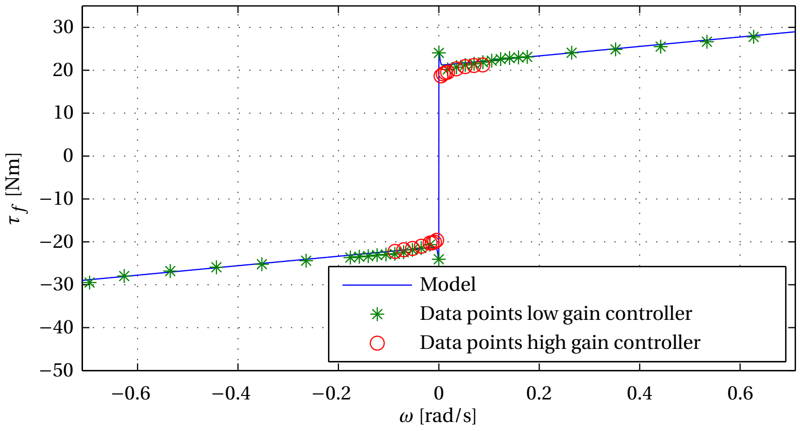


Figure 4.3. Friction model for the left knee including data points used to estimate model parameters.

From the figure it seems reasonable to use a friction model with stiction, coulomb, and viscous friction since the overall shape comply with the fitted model including those components. It does however seem like the friction drops at low velocities rather than increase which was expected. Due to the stiction it is however chosen to use to the model including the Stribeck effect since the transition from stiction to viscous and coulomb is smooth when the Stribeck effect is included while it is discontinuous if the stribeck velocity is left out. A more smooth friction model is desired since it is more suitable for friction compensation purposes.

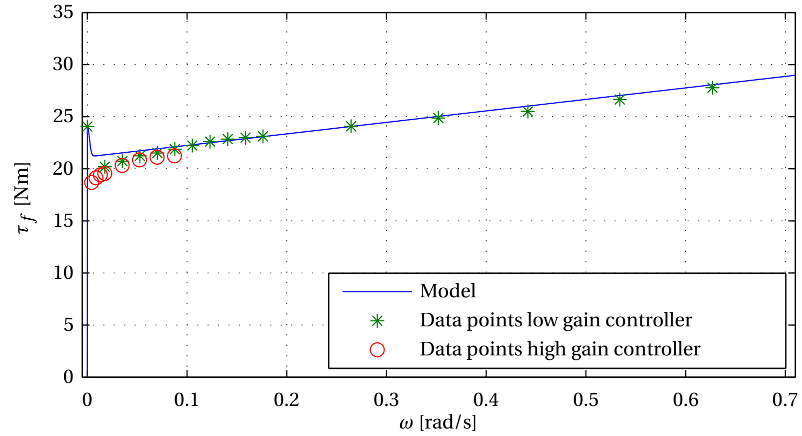


Figure 4.4. Friction model for the left knee including data points used to estimate model parameters (zoom).

In figure 4.4 a zoom of figure 4.3 is shown. As seen in the figure, the friction is mapped with both controllers for different angular velocities. For those angular velocities it seems that the measured friction for the high gain controller is a little less than for the low gain, but the overall shape is consistent. For the high gain controller the movements were observed to be shakier compared to the one with lower gains. The reduced friction observed while using the controllers with high gain could be explained by the theory that shaky movements reduce the effect of friction and oscillators have previously been attached to mechanical systems to compensate for friction [28]. From this it is seen that the choice of controller gains does not influence the overall shape of the observed friction, but shaky movements of the robot reduces the observed friction. It is therefore chosen only to use data from the low gain controller since this has a wider operating range and the high gain controller does not give any additional information about the friction.

The friction model for the right knee is determined in the same way as for the left, and the resulting model is shown in figure 4.5 where the model for the left knee is shown for comparison. In table 4.2 the resulting model parameters are shown. Due to an observation of highly angle dependent friction, the belts in the spur gears are changed. In figure 4.5 the models of the friction is shown with both the new and the old belt. From the figure it is seen that the viscous friction is almost unchanged while the coulomb friction and stiction is reduced between the two models of the right knee. The main reason for the reduced friction is most likely that the new belt is not tightened as hard as the old and not the reduced dependence on the angle. From figure 4.5 and table 4.2 it is seen that the viscous friction is almost twice as high as for the left knee compared to the right even though the knees are identical except that they are mirrored. From this it is seen that it is necessary to make experiments on all joints to obtain precise friction models.

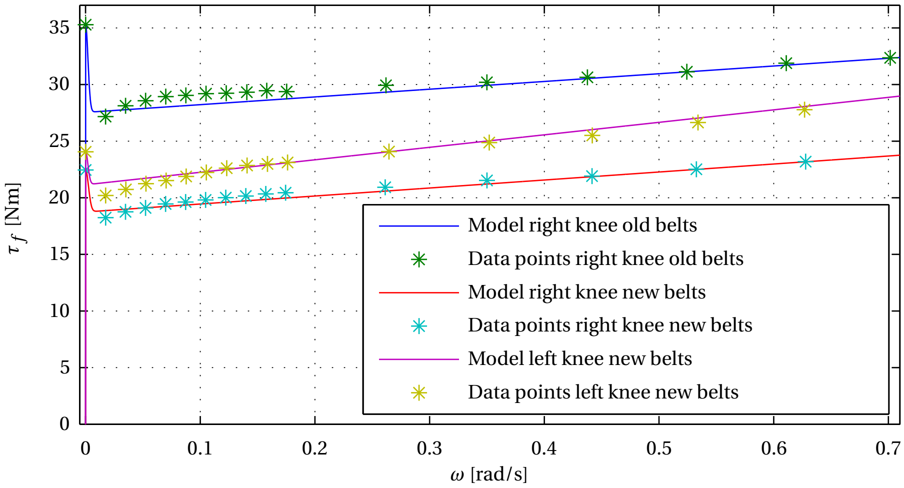


Figure 4.5. Friction model for both knees including data points used to estimate model parameters.

Friction model	τ_s [Nm]	τ_c [Nm]	μ [Nm/(rad/s)]	ω_s [rad/s]
Left Knee	24.01	21.15	11.03	0.0029
Right Knee (old)	35.30	27.54	6.82	0.0030
Right Knee (new)	22.45	18.74	7.07	0.0036

Table 4.2. Friction model parameters for the knee joints.

Angle dependent friction

As previously mentioned the friction was observed to be angle dependent by visual inspection and the belts in the spur gears were therefore changed. In figure 4.6 the torque caused by friction is shown when tracking an angle ramp reference with the low gain controller on the right knee with the old and the new belt.

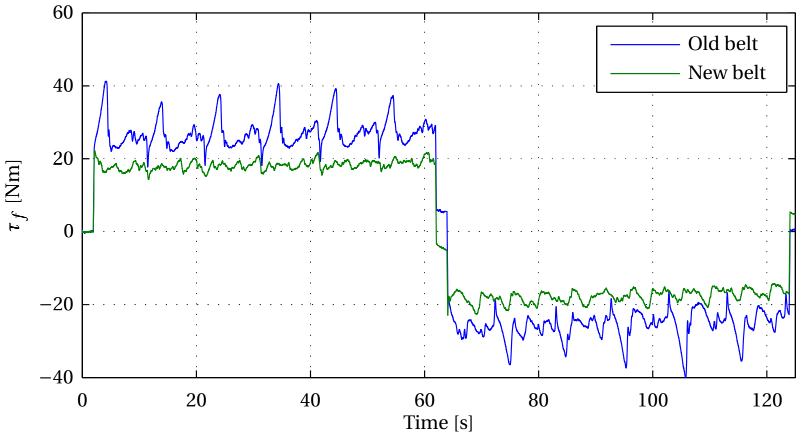


Figure 4.6. Measured torque caused by friction when turning with ±1 °/s.

From the figure it is easily seen that with the old belt, the friction is increased at some points. Within the first 60 s when running with positive velocity six spikes are identified while only five spikes are identified when running with negative velocity. The reason for this is that the joint starts in an angle with high friction and the high friction in the initial position is not observed when returning to the initial position. The spikes identified on the plot comply with the visual observation that the velocity of the joint decreased when the belt in the spur gear has rotated 360 °, which happens six times from the start position 0 ° to the max bend at 60 °. From figure 4.6 it is seen that changing the belt reduces the angle dependency of the friction. Additionally it is seen that the friction in general is reduced which comply with the results from the friction model shown in figure 4.5 and table 4.2.

Even though the angle dependency is reduced it seems that it is not completely removed. To analyse the angle dependency of the friction, another experiment is carried out. 20 runs are made running with constant velocity of ±5 °/s and ±10 °/s. The procedure for this test is described in appendix E. The results from the test are shown in figure 4.7. For the torques shown in the figure, the absolute value of the torque is shown to visualise where an increase of friction is present.

From the figure it is seen that with the same velocity the experienced torque is almost the same in all the 20 runs. In figure 4.8 mean value of the torque for each velocity is shown. In the figure an offset is added to the torques measured when running with positive velocities to be able to identify dependence on direction of movement.

From the figure it is seen that running in the same direction, the overall shape of the experienced torque is the same, and spikes are identified at the same angles. Running in the opposite direction it is however not possible to identify many similarities. Running at positive

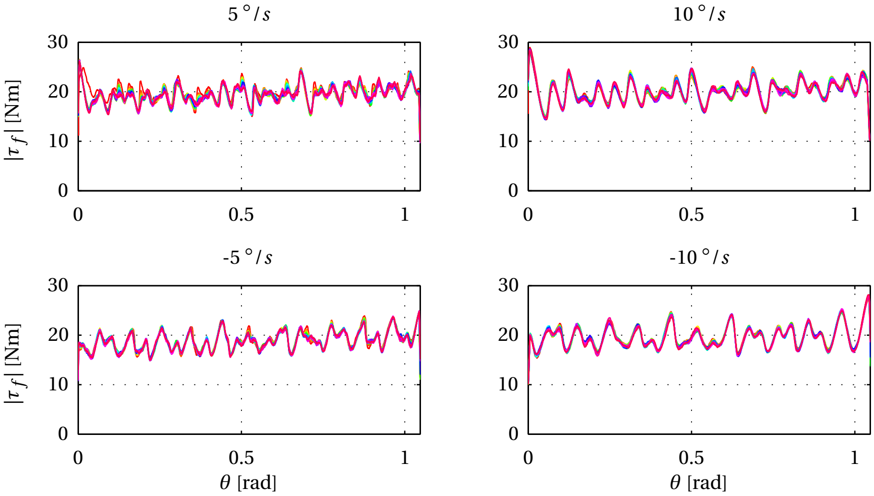


Figure 4.7. Torque caused by friction in the right knee when turning the same 60° with the four different velocities.

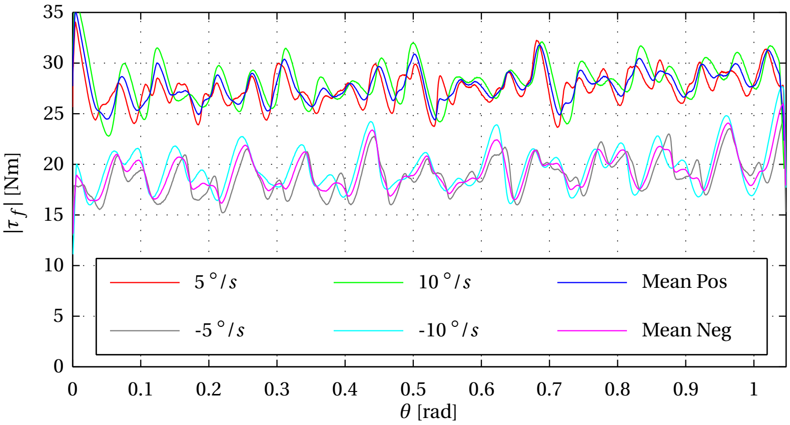


Figure 4.8. Mean torque caused by friction in the right knee when turning the same 60° with four different velocities. The graphs for positive velocities are offset by 10 Nm to enhance visibility.

velocities the current will start increasing when the friction increases because the error will increase. In this case the increased friction will show up as a higher torque at the angles above the angle where the friction i increased. And the same applies for the negative velocities that the friction at an angle will show up as an increased torque at a slight lower angle.

From figure 4.8 some periodicity is observed. To analyse this, a frequency analysis is carried out on the two mean graphs from figure 4.8 and the result is shown in figure 4.9. From this figure it is seen that for negative velocities there is high frequency content about 11 rad^{-1} while this in not seen for the positive velocities. Additionally it is observed that independent of the direction of movement, the observed torque has some content about the frequencies: 1 rad^{-1}, 6 rad^{-1}, 16 rad^{-1}, and 21 rad^{-1}.

The content at 1 rad^{-1} is assumed to be caused by gravity that is not compensated for perfectly. This is also seen from figure 4.8 where the torque in general is a little larger at 1 rad compared to 0 rad. The frequency content at 6 rad^{-1}, 16 rad^{-1}, and 21 rad^{-1} fits perfectly with

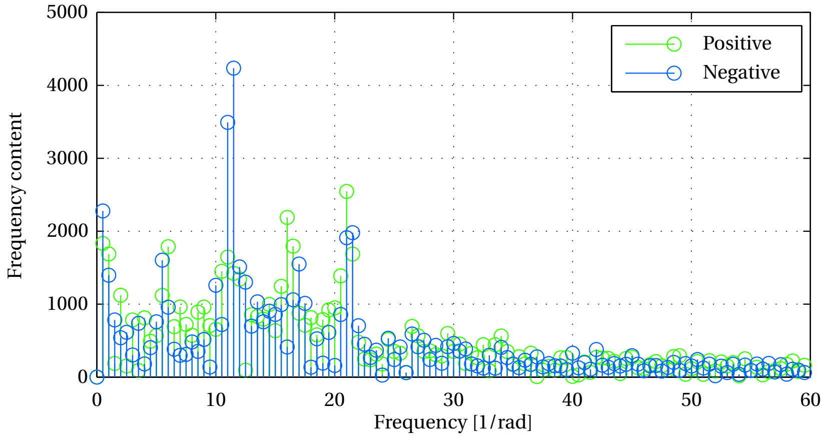


Figure 4.9. Frequency content of the mean torque for positive and negative velocities shown in figure 4.8. The DC value is subtracted before the frequency content is calculated.

the fact that the belt traverses about 6 revolutions when the knee traverses 1 rad, the large cog wheel traverses about 16 revolutions, and the small cog wheel traverses about 21 revolutions. The cause of the frequency content at about 11 rad^{-1}, has been identified to be mechanical oscillations. It is therefore not caused by friction in the gears.

During the experiments where friction model parameters are determined, the joints with highly angle dependent friction are identified. In these joints the belts are replaced to reduce the angle dependency.

In table 4.3 the model parameters are shown for all joints. The parameters shown in the table is generated from a new set of experiments where the posture controllers designed in chapter 5 is used since more constant velocity is obtained with these controllers.

It should be stressed that the parameters τ_c, μ, and ω_s presented in table 4.3 are captured from average values, and do therefore not include any information about the position dependent friction. For the stiction parameter τ_c only one experiment is conducted for each joint, hence this parameter could vary a lot if the parameter was identified at an angle with either a high or low friction compared to the average. From the table it is seen that the friction model parameters for two similar joints is not necessarily close to each other. An example is the two knee joints where the left knee friction model in general has larger parameters compared to the right knee. The main cause for this is assumed to be an higher tension of the belt in the spur gear in the left knee compared to the right knee.

4.2 Kinematics

In the kinematic models of the AAU-BOT1, the task is to mathematically link all the joints of the robot together. The forward kinematics is deduced to calculate positions of all the joints given the joint angles. In addition, by knowing the masses and positions of the CoM of each individual body part, the CoM of the whole robot can be calculated. An inverse kinematic model is deduced for the legs, to be able to calculate the joint angles needed to reach a specific position with the feet.

Joint no.	Joint name	τ_s [Nm]	τ_c [Nm]	μ [Nm/(rad/s)]	ω_s [rad/s]
1	Right Foot Roll	11.58	13.71	3.70	178.76e-3
2	Right Foot Pitch	25.32	18.91	14.28	2.75e-3
3	Right Knee	24.39	17.95	7.77	2.96e-3
4	Right Hip Pitch	21.83	16.52	14.03	3.27e-3
5	Right Hip Roll	22.69	15.67	16.47	3.75e-3
6	Right Hip Yaw	8.70	6.32	8.00	601.63e-3
7	Left Hip Yaw	10.72	8.52	5.48	2.99e-3
8	Left Hip Roll	19.51	15.23	16.91	3.38e-3
9	Left Hip Pitch	18.38	12.97	14.78	3.57e-3
10	Left Knee	27.67	20.33	9.42	3.15e-3
11	Left Foot Pitch	23.39	20.50	15.24	3.13e-3
12	Left Foot Roll	14.57	11.07	5.09	3.22e-3
13	Pelvis Yaw	12.00	9.61	4.46	3.10e-3
14	Pelvis Pitch	20.48	15.40	15.27	3.78e-3
15	Pelvis Roll	15.57	8.71	22.89	15.84e-3
16	Right Arm	5.87	4.54	3.55	3.30e-3
17	Left Arm	3.75	3.18	1.56	2.88e-3

Table 4.3. Friction model parameters for all joints.

4.2.1 Forward Kinematics

The idea in the forward kinematics is to place coordinate frames in all joints, and use the orientation and position of one frame with respect to the other frame to convert positions in one frame into positions in another. By making succeeding transformations, all points of interest in the robot can be represented in a common frame, the pelvis frame. Appendix D contains a description and an example of how these transformations between coordinate frames are carried out. To introduce the notation, an example is included here. A point ${}^{A}\mathbf{P}_1 \in \mathbb{R}^3$ in frame {A}, indicated by the superscript A, can be represented in frame {B} by applying the following transformation:

$$\begin{bmatrix} {}^{B}\mathbf{P}_1 \\ 1 \end{bmatrix} = {}^{B}_{A}\mathbf{T} \begin{bmatrix} {}^{A}\mathbf{P}_1 \\ 1 \end{bmatrix}$$

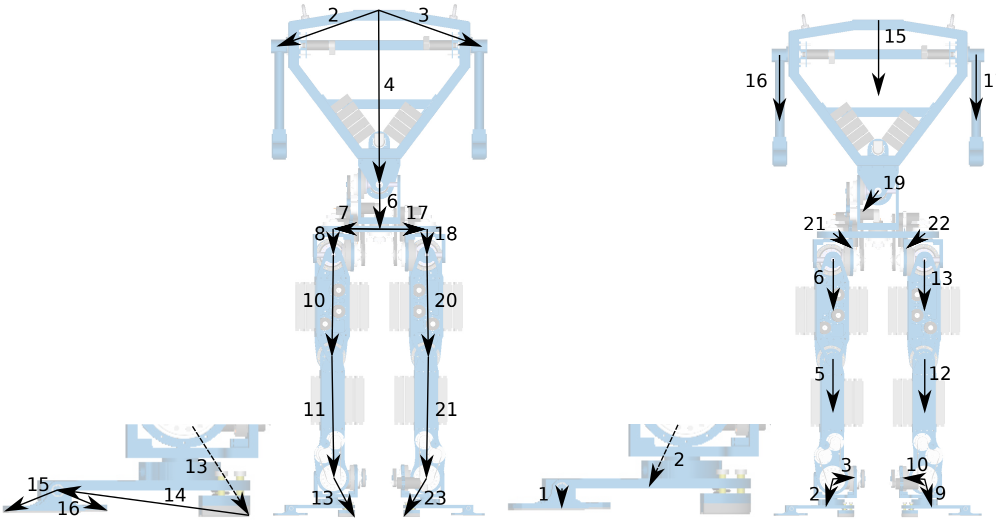
Where:

${}^{B}\mathbf{P}_1 \in \mathbb{R}^3$ is a representation of point $\mathbf{P}_1$ in frame {B}

${}^{B}_{A}\mathbf{T} \in \mathbb{R}^{4\times4}$ is the transformation matrix applying a rotation and translation

A coordinate frame is placed in every joint of the robot. If one joint has several DoF, one frame for each DoF is placed in the joint, such that each transformation only rotates about a single axis. In the initial position of the robot, where the robot is in upright position as seen in figure 1.1(a), all frames in the robot are aligned with the pelvis frame.

The positions of the joints, and thereby frames, are specified with respect to a parent joint frame by its position in the parent joint frame. The position of the frames with respect to the parent frame is extracted from the solidworks drawing of the robot by the previous group working with the AAU-BOT1 [2]. The extracted measures of the joint positions are shown in table A.3 in appendix A, and in figure 4.10(a) the positions are illustrated. It should be noted that a small error was detected in the order of rotation in the pelvis pitch-roll joint compared to [2]. The correct order is roll, pitch, yaw seen from the upper body and not pitch, roll, yaw as stated in [2].



(a) Joint Positions.

(b) CoM Positions.

Figure 4.10. Positions of the joints and CoM of links. The numbers on the vectors in figure (a) and (b) are the indexes shown in table A.3 and A.4 respectively. In the figure short vectors are not shown. It should be noted that the legs are rotated 90 deg to enhance visibility. [2]

For the joint positions shown in table A.3 and the known angle of the joints, the joint positions in the pelvis frame are found by forming a chain of transformations that transforms the position of a joint given in a frame to a position in the pelvis frame. For example the points in the right foot are transformed into the frame placed in the Right Ankle Roll. This is done purely by translation since no actuated joints are in between any of the points and the Right Ankle Roll frame. The points are then transformed into Right Ankle Pitch frame by rotating the points in the roll direction and no translation since the two joints are in the same position. The next transformation is into the Right Knee frame. Here the points are rotated in the pitch direction (Right Ankle Pitch) and translated by the position of the Right Ankle Pitch in the Right Knee frame. In this way the transformations are build to transform any point of interest specified in any of the joint frames in the robot into the pelvis frame. The pelvis frame is chosen to coincide with the pelvis yaw frame. In figure 4.11 these transformations are shown as blue arrows.

None of the instruments on the AAU-BOT1 can provide a global position of the robot. To be able to keep track of the global position of the robot, the forward kinematics is used. The approach taken is to assume that the most supporting foot always keeps a constant position with respect to the global frame. Knowing the transformation from pelvis frame to supporting foot frame (shown as red arrows in figure 4.11), the points of interest specified in the pelvis frame can be converted into the supporting foot frame. Having all points of interest in the supporting foot frame these can be converted into the global frame using the knowledge that the position and orientation of the supporting foot in the global frame is the same as for the previous iteration. For multiple steps the integrated global position will diverge more and

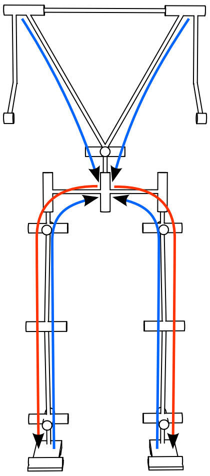


Figure 4.11. A sketch showing the kinematic chains defined in the forward kinematics. All joint positions are transformed into the common pelvis frame (indicated by blue arrows). Then to the supporting foot (one of the red arrows), and finally to the global frame.

more from the actual global position due to e.g. small errors in the length of limbs, errors in the determined joint angles, and flexibility in the structure. However for a limited amount of steps the global position has been determined to be sufficiently accurate using this method.

Calculation of CoM

The position of the CoM for each link is found in the pelvis frame in the same way as for the joint positions. In table A.4 the position and mass of each link in the robot is specified with respect to its specified parent frame. In figure 4.10(b) the positions are illustrated. From the position of the CoM and the mass of each link the CoM of the entire robot is calculated using equation (2.1), defining each link as a particle. The table A.4 and figure 4.10(b) are inspired from [2] the figure and data is however reviewed and minor errors have been corrected.

4.2.2 Inverse Kinematics

This section describes the deduction of the inverse kinematics. During gait it is necessary to specify and place the feet of the robot, at target positions and orientations with respect to the robot. In order to do this, all the joint angles in the legs need to be calculated.

In case of the AAU-BOT1, it is chosen to define the position of the feet, by defining the two heel positions with respect to the pelvis frame. For the right leg this is defined as the transformation that transforms a point given in frame {0} to frame {7} in figure 4.12.

The desired heel position and orientation of the right foot with respect to the pelvis frame is defined as the transformation ${}^{7}_{0}\tilde{\mathbf{T}}$ and has a structure as seen in expression 4.9.

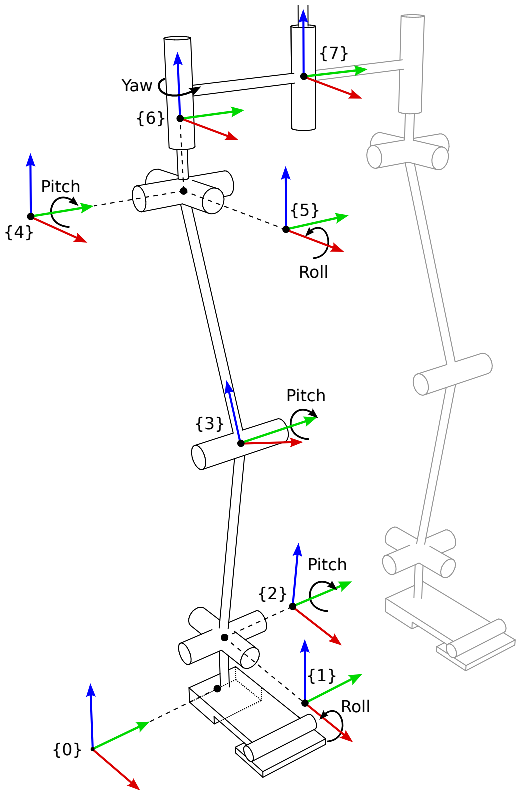


Figure 4.12. Definition of the different frame in the right leg of the AAU-BOT1. A red arrow denotes x-axis, green y-axis, and blue z-axis. It should be noted that the parent/child relationship is swapped compared to the forward kinematics, such that the frames in the joint follow the upper limb. E.g. the frame in the knee follows the thigh.

$$
{}^{7}_{0}\tilde{\mathbf{T}} = \begin{bmatrix} i_x & j_x & k_x & p_x \\ i_y & j_y & k_y & p_y \\ i_z & j_z & k_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.9}
$$

where:

$\mathbf{i} \in \mathbb{R}^3$ is the unit direction vector of the x-axis in frame {0} seen from {7}

$\mathbf{j} \in \mathbb{R}^3$ is the unit direction vector of the y-axis in frame {0} seen from {7}

$\mathbf{k} \in \mathbb{R}^3$ is the unit direction vector of the z-axis in frame {0} seen from {7}

$\mathbf{p} \in \mathbb{R}^3$ is the position of the origin in frame {0} seen from {7}

Considering each leg from pelvis frame to heel frame, the legs both have 6 DoF. Generally two different approaches can be taken to solve the inverse kinematic, *closed-form* or *numerical* solutions [29]. Since numerical solutions, due to their iterative nature, requires much more time to solve, only a closed-form solution is considered in this project.

To guarantee that a closed-form solution can be found for a manipulator of 6 DoF, Pieper's work is studied [29]. Pieper have deduced that all serially connected manipulators with 6 DoF have a closed-form solution, if the last three joint axes intersect. Looking at figure

4.12 it is clear that this is indeed not the case for the leg of the AAU-BOT1 seen from the pelvis to the heel, since the only place that three axes intersect is in the hip joint. However if the problem is reformulated, such that the task is to find a transformation ${}^0_7\tilde{\mathbf{T}}$ that transforms a point in frame {7} to frame {0}, the three hip joint axes are the last three axes, and a closed form solution can be found. From the specification of transformation ${}^7_0\tilde{\mathbf{T}}$ the transformation ${}^0_7\tilde{\mathbf{T}}$ is easily found from the property of inverse transformations shown in appendix D:

$$ {}^0_7\tilde{\mathbf{T}} = {}^7_0\tilde{\mathbf{T}}^{-1} $$

To simplify the complexity of deducing the inverse kinematics, the origin of frame {6} is chosen to be the same as frames {4} and {5} even though it should be translated in the z-direction according to table A.3. This simplification removes an additional length in the calculations but the solution for the joint angles will be the same. Another simplification is that the transformation from where the three hip joints intersect to the centre of pelvis ${}^7_6\mathbf{T}$ is omitted by redefining the target position of the heel with respect to the hip yaw frame {6}, rather than frame {7}. This simplification makes the calculations simpler but the result will be the same since frame {6} and {7} are always aligned and only translated with respect to each other.

Remembering that we have reversed the perspective from pelvis to heel frame, the final transformation from frame {6} to frame {0} is now found as:

$$ {}^0_6\tilde{\mathbf{T}} = ({}^6_7\mathbf{T}\,{}^7_0\tilde{\mathbf{T}})^{-1} = \begin{bmatrix} i'_x & j'_x & k'_x & p'_x \\ i'_y & j'_y & k'_y & p'_y \\ i'_z & j'_z & k'_z & p'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.10} $$

where:

$\mathbf{i}' \in \mathbb{R}^3$ is the unit direction vector of the x-axis in {6} seen from {0}

$\mathbf{j}' \in \mathbb{R}^3$ is the unit direction vector of the y-axis in {6} seen from {0}

$\mathbf{k}' \in \mathbb{R}^3$ is the unit direction vector of the z-axis in {6} seen from {0}

$\mathbf{p}' \in \mathbb{R}^3$ is the position of the origin in frame {6} seen from {0}

The goal is now to find a transformation ${}^0_6\mathbf{T} = {}^0_6\tilde{\mathbf{T}}$ by combining the transformations between the frames in the joints defined by the joint angles and length of limbs:

$$ {}^0_6\mathbf{T} = {}^0_1\mathbf{T}\,{}^1_2\mathbf{T}\,{}^2_3\mathbf{T}\,{}^3_4\mathbf{T}\,{}^4_5\mathbf{T}\,{}^5_6\mathbf{T} \tag{4.11} $$

In the following expressions, the notation C_x and S_y will denote $\cos(\theta_x)$ and $\sin(\theta_y)$ respectively and C_{xy} denotes $\cos(\theta_x + \theta_y)$. The lengths indicated by l_x, l_y, and l_z indicates the displacement of the child frame with respect to its parent frame. These lengths are found from table A.3. The signs of the position vectors are changed to swap the parent/child relationship. By swapping the parent/child relationship, the six transformations used in (4.11) becomes much simpler compared to keeping the original relationship. The reason for this is that keeping the original relationship the position of the child frame in the parent will depend on the angle of the parent frame. This is a consequence of the changed direction of the kinematic chain, starting from the hip and moving to the foot. The only drawback in changing the relationship is that the resulting angles have opposite sign of the angles used in the forward kinematics. This is compensated for after the angles have been calculated.

The transformation between the different frames used in equation (4.11) are specified by the joint angles and lengths between the joints in the following way:

$$
{}^0_1\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & l_{x1} \\ 0 & C_1 & -S_1 & l_{y1} \\ 0 & S_1 & C_1 & l_{z1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}^1_2\mathbf{T} = \begin{bmatrix} C_2 & 0 & S_2 & 0 \\ 0 & 1 & 0 & 0 \\ -S_2 & 0 & C_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}^2_3\mathbf{T} = \begin{bmatrix} C_3 & 0 & S_3 & 0 \\ 0 & 1 & 0 & 0 \\ -S_3 & 0 & C_3 & l_{z3} \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^3_4\mathbf{T} = \begin{bmatrix} C_4 & 0 & S_4 & 0 \\ 0 & 1 & 0 & l_{y4} \\ -S_4 & 0 & C_4 & l_{z4} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}^4_5\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_5 & -S_5 & 0 \\ 0 & S_5 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}^5_6\mathbf{T} = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

The proposed solution to deduce the joint angels will base on an algebraic approach, where equations will be deduced from the link parameters in each joint. The proposed solution is inspired by [30]. The final set of equations that must be solved can be found by equating equation (4.10) and (4.11).

$$
\begin{gathered}
{}^0_6\mathbf{T} = {}^0_6\tilde{\mathbf{T}} \\
\Downarrow \\
{}^0_1\mathbf{T}{}^1_2\mathbf{T}{}^2_3\mathbf{T}{}^3_4\mathbf{T}{}^4_5\mathbf{T}{}^5_6\mathbf{T} = \begin{bmatrix} i'_x & j'_x & k'_x & p'_x \\ i'_y & j'_y & k'_y & p'_y \\ i'_z & j'_z & k'_z & p'_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{gathered}
\tag{4.12}
$$

In the next part of this section, the deduced joint angles will be stated. More detailed information about how these expressions are deduced from equation (4.12), can be found in appendix C. First, the three angles $\theta_1, \theta_2, \theta_3$ that determines the position of the hip joint are found:

$$
C_3 = \frac{-l_{z4}^2 - l_{z3}^2 - l_{y4}^2 + p_x'^2 - 2p'_x l_{x1} + l_{x1}^2 + p_y'^2 - 2p'_y l_{y1} + l_{y1}^2 + p_z'^2 - 2p'_z l_{z1} + l_{z1}^2}{2 l_{z4} l_{z3}} \tag{4.13}
$$

$$
\theta_3 = \operatorname{atan2}\left(\pm\sqrt{1 - C_3^2}, C_3\right) \tag{4.14}
$$

Notice that to solve the $\cos(\theta_3)$ in equation 4.13, the function $\operatorname{atan2}(y, x)$ is used. This function works in almost the same way as $\operatorname{atan}\left(\frac{y}{x}\right)$, but instead of returning an angle between $[-\frac{\pi}{2}, \frac{\pi}{2}]$ it returns an angle within $[-\pi, \pi]$, where the angle has been adjusted to the right quadrant based on the sign of x and y. The two next angles are found in equation 4.15 and 4.16.

$$\theta_2 = \operatorname{atan2}\left(a_2 d_2 - b_2 c_2, a_2 c_2 + b_2 d_2\right) \tag{4.15}$$

Where:

$$\begin{aligned}
a_2 &= l_{z4} C_3 + l_{z3} \\
b_2 &= l_{z4} S_3 \\
c_2 &= \pm\sqrt{(p'_y - l_{y1})^2 + (p'_z - l_{z1})^2 - l_{y4}^2} \\
d_2 &= p'_x - l_{x1}
\end{aligned}$$

$$\theta_1 = \operatorname{atan2}\left(a_1 d_1 - b_1 c_1, a_1 c_1 + b_1 d_1\right) \tag{4.16}$$

Where:

$$\begin{aligned}
a_1 &= l_{z4} \\
b_1 &= l_{z4} C_{23} + C_2 l_{z3} \\
c_1 &= p'_y - l_{y1} \\
d_1 &= p'_z - l_{z1}
\end{aligned}$$

The three last joint angles $(\theta_4, \theta_5, \theta_6)$ that determines the orientation of the hip joint are now stated:

$$\theta_5 = \pm\operatorname{atan2}\left(-C_1 k'_y - S_1 k'_z, \sqrt{1 - (-C_1 k'_y - S_1 k'_z)^2}\right) \tag{4.17}$$

$$\theta_6 = \operatorname{atan2}\left(C_1 i'_y + S_1 i'_z, C_1 j'_y + S_1 j'_z\right) \tag{4.18}$$

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \tag{4.19}$$

Where:

$$\theta_{234} = \operatorname{atan2}\left(k'_x, C_1 k'_y - S_1 k'_y\right)$$

This concludes the deduction of the six joint angles in the right leg. Some of the above equations have two solutions. In case of multiple solutions the solution that does not violate the limitations on the robot is chosen e.g. the solution for the knee joint angle θ_3 that does not hyperextend the knee is chosen. Next, everything must be deduced for the left leg as well. Fortunately the procedure is exactly the same with only some minor changes in some of the distances because the two legs are not completely symmetrically. Otherwise the only difference is that the hip-pelvis transformation needs to transform to the left hip instead of the right. For these reasons the deduction of the left leg will not be evaluated here.

The final topic of this section will be the existence of a solution to the joint angles. First of all the reachable workspace of the heel is limited by the length of the leg when it is fully extended. Any desired target frame has to lie within the sphere with centre in the right/left hip joint and a radius of the extended leg. Inside this sphere the physical construction of the leg limits the reachable workspace even further. This especially concerns the knee joints, which due to limitations can not hyperextend or bend more than approximately 73 ° from full extension. This creates a smaller sphere close to the hip joint inside which the heel can never reach.

Similarly the joint limitations for all the other joints limits the reachable workspace of the heels. These will however not be discussed here.

Some of these limitations will become apparent when trying to solve a problem that is outside the workspace. If a target frame is attempted which is too far away from the robot, equation (4.14) will result in a negative square root, and no solutions will be found. However the gait trajectories are not assumed to be influenced by these limitations. This is due to the fact that the legs have been designed such that human flexibility should be achieved. Hence following a trajectory which is almost similar to a human should be possible.

The workspace is also influenced by the individual placements of the legs, since collision must be avoided. This will however not be considered in the inverse kinematics. Instead the gait-balance controller should make sure that collision is avoided.

4.2.3 Kinematic Model Assessments

The together the forward and inverse kinematics forms a powerful tool for creating robot motions and calculating CoM. The accuracy of the kinematic models is however vital to the final performance, when implemented on the AAU-BOT1. If positions of joints and CoM's are not accurate the actual position of the feet and CoM will diverge from what is desired. If the errors are too great the robot will be impossible to control. It is chosen to assume that positions specified by earlier project groups are accurate. Measuring the actual positions would require a near total disassembly of the robot.

It is as well difficult to measure the exact position and orientation of the joints with manual measuring equipment. The forward and inverse kinematics have been tested to be inverse if each other. The actual precision of the two models will not be evaluated. Tests using both forward and inverse kinematics have however shown that they correspond reasonably well with the actual robot. An example of this shown in figure 7.20 where snapshots of a simulation are shown. In the simulation the inverse kinematics is used to determine the needed joint angles to perform gait and the forward kinematics is used to visualise the actual robot movements from the derived joint angles. A video showing the output of the entire simulation can be found on the DVD 💿`/videos/kinematic_gait_simulation.avi`.

4.3 Dynamic Models

The following section introduces the dynamic models used to describe the behaviour of the CoM and ZMP for the AAU-BOT1. These models will be used in the Kalman filter described in section 6.3 and for generating CoM references described in section 7.3. To make the models usable in online computations, it is desired to find models of a relatively low complexity.

Several different ways of modelling biped robots have been proposed in literature about the subject. Some methods rely on a total dynamic model of the robot, whereas others rely on simplified models. The majority of the simplified models use inverted pendulums to model movement of the CoM of the entire robot. Most often either the 2D or 3D linear inverted pendulum model is used. These are often referred to as the 2D-LIPM and 3D-LIPM models [31, 32]. In these models the entire mass of the robot is considered concentrated in the CoM that is located in the bob of a pendulum, and the pivot point is the supporting ankle. In double support the single inverted pendulum models perform poorly if the pivot point is kept at one of the two ankles. To overcome this, [33] derived a Linear Biped Model (LiBM), where two 2D-LIPM are superimposed, one for each leg.

In section 4.3.1 we will present the model for the 3D-LIPM which models the behaviour of the CoM in single support. In the end of section 4.3.1 the similarity between the 3D-LIPM and 2D-LIPM is shown. By combining two LIPMs the behaviour of the CoM in double support is modelled by the LiBM presented in section 4.3.2. To model the correspondence between the CoM movement and ZMP of the AAU-BOT1 the cart-table model will be used [19]. This model will be presented in section 4.3.3.

4.3.1 The 3D Linear Inverted Pendulum Mode

Observing a humanoid in single support as an inverted pendulum seems intuitive. Using a telescopic pendulum rather than a fixed length pendulum includes the humanoids capability of shortening or extending the leg to raise or lower the CoM.

From this idea Kajita et. al. derived a model for the movements of the CoM in 3D [32]. In this section the derivation of the 3D-LIPM will be carried out with more intermediate steps compared to the original derivation made in [32].

In figure 4.13 a telescopic inverted pendulum is shown in three dimensions. The position of the point mass $\mathbf{p} = [x\,y\,z]^T$ is described by two angles and the length of the pendulum $\mathbf{q} = [\theta_r\,\theta_p\,r]^T$. The angle θ_r is the angle between the pendulum and the xz-plane, and the angle θ_p is the angle between the pendulum and the yz-plane. The positive direction of these angles is defined according to a right handed frame such that $\theta_r < 0$ and $\theta_p > 0$ for the case shown in figure 4.13.

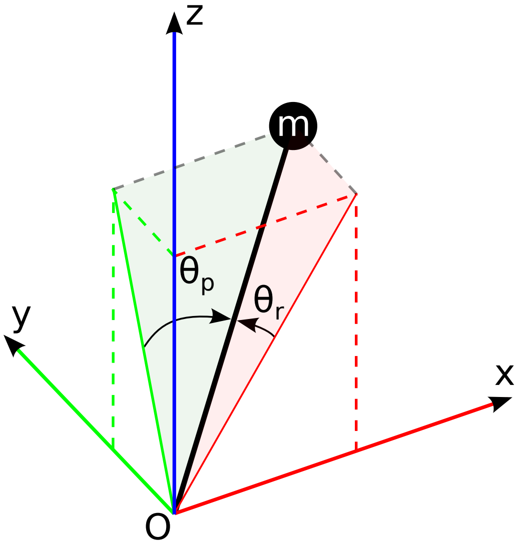


Figure 4.13. Sketch of 3D-inverted pendulum.

The relationship between the state variables **q** and the position of the point mass **p** is:

$$x = rS_p \qquad [\mathrm{m}] \quad (4.20)$$

$$y = -rS_r \qquad [\mathrm{m}] \quad (4.21)$$

$$z = \sqrt{r^2 - x^2 - y^2} = r\sqrt{1 - S_p^2 - S_r^2} = rD \qquad [\mathrm{m}] \quad (4.22)$$

Where:

$$S_p = \sin\theta_p$$

$$S_r = \sin\theta_r$$

$$D = \sqrt{1 - S_p^2 - S_r^2}$$

Using (τ_r, τ_p, f) as inputs to apply torques and force in the direction of the state variables **q**, equations for the work performed on the point mass with respect to the state variables is found. Equating these to the work performed on the mass in Cartesian coordinates the following relationship is found:

$$\tau_r\Delta\theta_r + \tau_p\Delta\theta_p + f\Delta r = F_x\Delta x + F_y\Delta y + F_z\Delta z \qquad [\mathrm{J}]$$

Where:

$$\begin{bmatrix} F_x & F_y & F_z \end{bmatrix} = \begin{bmatrix} m\ddot{x} & m\ddot{y} & m\ddot{z} + mg \end{bmatrix} \qquad [\mathrm{N}]$$

Δ is a change in the corresponding variable

m is the mass of the point mass [kg]

g is the acceleration due to gravity [m/s^2]

Writing as dot products and rearranging yields:

$$[\tau_r\, \tau_p\, f] = [F_x\, F_y\, F_z]\frac{\Delta\mathbf{p}}{\Delta\mathbf{q}}$$

The last term of the latter equation can be identified as the Jacobian, $\frac{\Delta\mathbf{p}}{\Delta\mathbf{q}} \rightarrow \frac{\partial\mathbf{p}}{\partial\mathbf{q}} = \mathbf{J}$, when step size Δ is made infinitesimally small. Inserting this and rearranging the expression, following dynamics are found:

$$m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = (\mathbf{J}^T)^{-1}\begin{bmatrix} \tau_r \\ \tau_p \\ f \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \qquad [\mathrm{N}] \quad (4.23)$$

The Jacobian **J** is calculated as:

$$\mathbf{J} = \frac{\partial\mathbf{p}}{\partial\mathbf{q}} = \begin{bmatrix} 0 & rC_p & S_p \\ -rC_r & 0 & -S_r \\ -rC_rS_r/D & -rC_pS_p/D & D \end{bmatrix}$$

Where:

$$C_p = \cos\theta_p$$

$$C_r = \cos\theta_r$$

Left multiplying the transpose of the Jacobian $\mathbf{J}^T$ on all terms in (4.23) removes the task of calculating the inverse of the Jacobian:

$$m\begin{bmatrix} 0 & -rC_r & -rC_rS_r/D \\ rC_p & 0 & -rC_pS_p/D \\ S_p & -S_r & D \end{bmatrix}\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \tau_r \\ \tau_p \\ f \end{bmatrix} - mg\begin{bmatrix} -rC_rS_r/D \\ -rC_pS_p/D \\ D \end{bmatrix} \tag{4.24}$$

Extracting the first row from (4.24) and using the kinematic relationships given in equations (4.20) to (4.22) the dynamics for movements in the yz-plane is identified:

$$m(-z\ddot{y}+y\ddot{z}) = \frac{D}{C_r}\tau_r - mgy \qquad [\text{Nm}] \tag{4.25}$$

In the same way dynamics for movement in the xz-plane is identified from the second row in (4.24)

$$m(z\ddot{x}-x\ddot{z}) = \frac{D}{C_p}\tau_p + mgx \qquad [\text{Nm}] \tag{4.26}$$

These two equations for the movements of the pendulum, allows for a wide range of movements. As specified in [32] it is desired to limit the motion of the pendulum to a class motion that would be relevant for walking. The first constraint put on the motion is that the CoM must move in a plane:

$$z = k_x x + k_y y + z_c \qquad [\text{m}] \tag{4.27}$$

In [32] it is specified that the constraint plane should match the slope of the ground where the robot is walking. The scope in this thesis is to make the AAU-BOT1 walk on flat ground, hence the movements of the CoM is constrained to the flat plane ($k_x = 0, k_y = 0$). The intersection with the z-axis z_c is chosen as the average distance from the CoM of the robot to the ground. The constraint in equation (4.27) and its double derivative with respect to time reduces to:

$$z = z_c \qquad [\text{m}] \tag{4.28}$$

$$\ddot{z} = 0 \qquad \left[\text{m/s}^2\right] \tag{4.29}$$

Inserting this in (4.25) and (4.26) the resulting dynamics are found:

$$\ddot{y} = \frac{g}{z_c}y - \frac{1}{mz_c}\frac{D}{C_r}\tau_r \qquad \left[\text{m/s}^2\right] \tag{4.30}$$

$$\ddot{x} = \frac{g}{z_c}x + \frac{1}{mz_c}\frac{D}{C_p}\tau_p \qquad \left[\text{m/s}^2\right] \tag{4.31}$$

A special case of these pendulum equations is when the movement is additionally constrained to be in either the xz- or yz-plane. In this case the dynamics reduces to the dynamics of the 2D-LIPM [31]. Constraining the movements to be in the yz-plane by specifying $\theta_p = 0\,\text{rad}$ the equation for the yz-movements (4.30) reduces to:

$$\begin{aligned} \ddot{y} &= \frac{g}{z_c}y - \frac{1}{mz_c}\frac{\sqrt{1-S_r^2}}{C_r}\tau_r && \left[\text{m/s}^2\right] \\ &= \frac{g}{z_c}y - \frac{1}{mz_c}\tau_r && \left[\text{m/s}^2\right] \quad (4.32) \end{aligned}$$

For the movements constrained to the xz-plane ($\theta_r = 0\,\text{rad}$) equation (4.31) reduces to:

$$\ddot{x} = \frac{g}{z_c}x + \frac{1}{mz_c}\tau_p \qquad \left[\text{m/s}^2\right] \quad (4.33)$$

As seen in equations (4.30) and (4.31) the dynamics for the telescopic inverted pendulum becomes linear apart from the input non-linearity. It should be noted that in the derivation it is assumed that the torque is applied in the direction of the angles θ_r and θ_p. This is not exactly the situation on the robot since roll torque τ_r is applied about the x-axis and pitch torque τ_p is applied about the y-axis in the frame that is rotated with angle θ_r about the x-axis in the original frame. For situations where the angles θ_r and θ_p are small the modelling error becomes neglectable. In the 2D-case this issue is not apparent since the cross couplings leave the equations.

4.3.2 Linear Biped Model

The Linear Biped Model (LiBM) superimposes two 2D-LIPM to create a model which takes the two stance legs of a biped robot in double support into account. This approach was originally proposed by [33] and the following deduction takes the same approach. The authors of [33] however leave out an important fact about the assumption of the definition of the position of the CoM with respect to the centre point between the two feet. This will be clarified in this deduction.

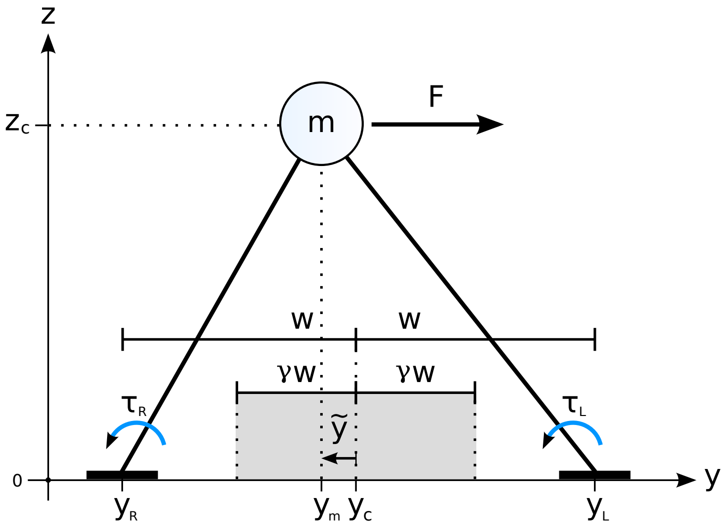


Figure 4.14. Sketch of the linear biped model. F indicates the forces acting on the mass m. y_R and y_L represents the global position of the right and left foot. τ_R and τ_L are the two torques acting in the two pivot points. y_c is the centre position between the two legs. y_m is the position of the shared mass m. w is half the stance width between the two feet. γw is the double support region. z_c is a constant height of m above the ground. $\tilde{y}$ is the relative position of y_m with respect to y_c.

The point of origin of the LiBM is to assume that a biped robot can be seen as two inverted pendulums sharing the same mass m as shown in figure 4.14. The forces acting in the horizontal plane on this mass can be deduced by superimposing the forces from two 2D-LIPMs F_R and F_L. To take into account that m represents a constant mass that is distributed between the two feet, two weighting factors ω_R and ω_L are introduced. These factors indicates how much of the total weight that is put on each foot. The two force contributions are deduced from the 2D-LIPM model found in (4.32). This results in the following expression for the force F:

$$F = m\ddot{y}_m = \omega_L F_L + \omega_R F_R \qquad [\text{N}] \quad (4.34)$$

where:

$F_R = \dfrac{mg}{z_c}(y_m - y_R) - \dfrac{1}{z_c}\tau_R$ is the force contribution from the right leg [N]

$F_L = \dfrac{mg}{z_c}(y_m - y_L) - \dfrac{1}{z_L}\tau_L$ is the force contribution from the left leg [N]

Introducing the total torque $\tau = \omega_R\tau_R + \omega_L\tau_L$ in equation 4.34, yields equation 4.35.

$$m\ddot{y}_m = -\frac{\tau}{z_c} + \omega_L\left(\frac{mg}{z_c}(y_m - y_L)\right) + \omega_R\left(\frac{mg}{z_c}(y_m - y_R)\right) \qquad [\text{N}] \quad (4.35)$$

The next step is to define how to select ω_R and ω_L. The solution proposed in [33], which will be adopted in this report, is to make the weights a function of the position of the CoM y_m. To make the LiBM valid for both single and double support, the weights are defined for three cases. Right single support, left single support and double support. To determine a condition that must be satisfied for the LiBM to be in double support, a relative position $\tilde{y}$ is defined, and double support is assumed when $\tilde{y}$ holds the following condition:

$$|\tilde{y}| < \gamma w$$

where:

$\tilde{y} = y_m - y_c$ is the position of y_m with respect to y_c [m]

$w = \frac{1}{2}(y_L - y_R)$ is half the stance width of the feet [m]

γ is a constant named the double support ratio

In figure 4.14 the double support region for the LiBM is indicated by the grey area. The weight of the left leg (ω_L) is defined for the three different support cases as seen in expression (4.36).

$$\omega_L = \begin{cases} 1 & \text{for} \quad \tilde{y} \geq \gamma w \quad \textit{(left support)} \\ \frac{\gamma w + \tilde{y}}{2\gamma w} & \text{for} \quad |\tilde{y}| < \gamma w \quad \textit{(double support)} \\ 0 & \text{for} \quad \tilde{y} \leq \gamma w \quad \textit{(right support)} \end{cases} \qquad (4.36)$$

Introducing the constraint that $\omega_L + \omega_R = 1$ we can deduce an expression for ω_R.

$$\omega_R = \begin{cases} 0 & \text{for} \quad \tilde{y} \geq \gamma w \quad \textit{(left support)} \\ \frac{\gamma w - \tilde{y}}{2\gamma w} & \text{for} \quad |\tilde{y}| < \gamma w \quad \textit{(double support)} \\ 1 & \text{for} \quad \tilde{y} \leq \gamma w \quad \textit{(right support)} \end{cases} \qquad (4.37)$$

The development of the two weightings is shown in figure 4.15 as a function of $\tilde{y}$. The rate of change of the two functions is defined by the choice of the double support ratio γ.

Using the weighting defined in (4.36) and (4.37), the biped model equation (4.35) describes an approximate behaviour of $\ddot{y}_m$ for the three different support cases. For right single support the model reduces to equation (4.38) for right support and (4.39) for left support.

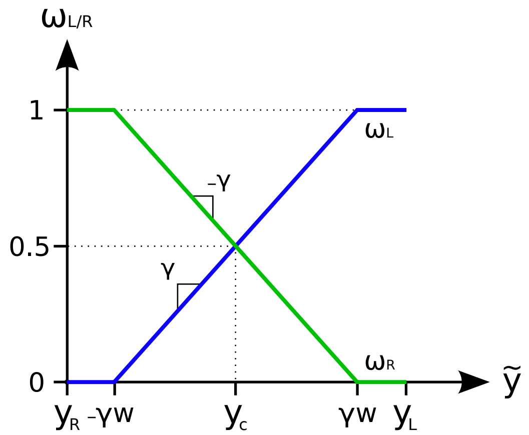


Figure 4.15. Illustration of the two weighting functions plotted.

$$\ddot{y}_m = \frac{g}{z_c}(y_m - y_R) - \frac{1}{m\,z_c}\tau_R \qquad [\mathrm{m/s^2}] \quad (4.38)$$

$$\ddot{y}_m = \frac{g}{z_c}(y_m - y_L) - \frac{1}{m\,z_c}\tau_L \qquad [\mathrm{m/s^2}] \quad (4.39)$$

Introducing the expressions for the two weights in double support reduces the LiBM to the linear equation shown in (4.40).

$$\ddot{y}_m = -\frac{g}{z_c}\frac{1-\gamma}{\gamma}\left(y_m - \frac{1}{2}\left(y_L + y_R\right)\right) - \frac{1}{m\,z_c}\tau \qquad [\mathrm{m/s^2}] \quad (4.40)$$

The final LiBM model diverges slightly from the original proposed in [33], since the introduction of $\tilde{y}$ in the final expression of the model slightly complicates the double support model. Using (4.40) however creates a more general expression that does not require y_c to be placed in Origin of the coordinate system.

For motion of CoM in the x-axis, the model is almost the same. The only difference is that the sign on the input torques is switched due to the orientation of the global frame. The right and left single support models in the x-axis are respectively (4.41) and (4.42). The double support model becomes (4.43).

$$\ddot{x}_m = \frac{g}{z_c}(x_m - x_R) + \frac{1}{m\,z_c}\tau_R \qquad [\mathrm{m/s^2}] \quad (4.41)$$

$$\ddot{x}_m = \frac{g}{z_c}(x_m - x_L) + \frac{1}{m\,z_c}\tau_L \qquad [\mathrm{m/s^2}] \quad (4.42)$$

$$\ddot{x}_m = -\frac{g}{z_c}\frac{1-\gamma}{\gamma}\left(x_m - \frac{1}{2}\left(x_L + x_R\right)\right) + \frac{1}{m\,z_c}\tau \qquad [\mathrm{m/s^2}] \quad (4.43)$$

4.3.3 Cart-table Model

To identify the point on the ground where the net moment is zero, a cart-table model is used [19]. The model is illustrated in figure 4.16 for the yz-plane. An identical model can be set up for the motion in the xz-plane. The model consists of a table with no mass that has ground support area identical to the robots, and a height corresponding to the height of the CoM of the robot. On this table a cart is placed which has the same mass as the entire robot and the same position as the CoM of the robot.

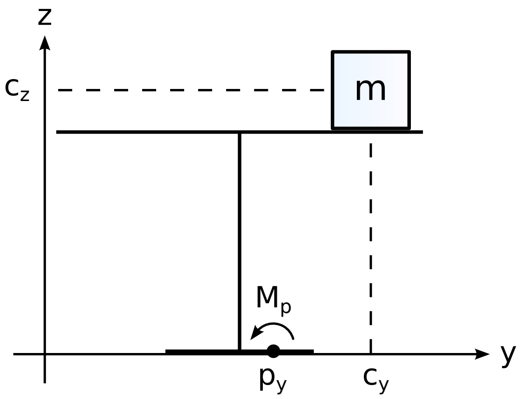


Figure 4.16. The cart-table model for ZMP movements in the frontal plane.

From the figure the torque about a point p_y on the ground surface is calculated from the resulting force on the mass in the y-direction, and the gravitational force.

$$M_p = -mg(c_y - p_y) + m\ddot{c}_y c_z \qquad \text{[Nm]}$$

Where:

M_p is the torque around point p_y	[Nm]
m is the mass of the cart	[kg]
c_y, c_z is the position of the mass in the yz-plane	[m]
g is the gravitational acceleration	[m/s^2]

Equating M_p to zero, the point where the net torque is zero (the ZMP) can be identified:

$$p_y = c_y - \frac{c_z}{g}\ddot{c}_y \qquad \text{[m]} \quad (4.44)$$

From this equation the relationship between the movement of the CoM and the position of the ZMP is given in the yz-plane. For the xz-plane the following equation is derived:

$$p_x = c_x - \frac{c_z}{g}\ddot{c}_x \qquad \text{[m]} \quad (4.45)$$

As explained in [34, p. 374] the cart-table model is only valid as long as the ZMP is kept within the support area. If the ZMP leaves the support area both the table and the robot will turn over the edge of the support area. The cart-table model is deduced from an assumption that the mass moves strictly in the transverse plane. Therefore this model will be most accurate when the CoM is moved with constant height above the ground.

Extension of the Cart-table Model

To verify to what degree the deduced cart-table model fits the AAU-BOT1, an experiment is performed. The robot is given a CoM reference in the transverse plane and the measured ZMP is compared to the one given by the cart-table model. The result of this experiment is seen in figure 4.17. The ZMP calculated from the cart-table model has been filtered using a moving average, to suppress noise created by differentiating the measured CoM.

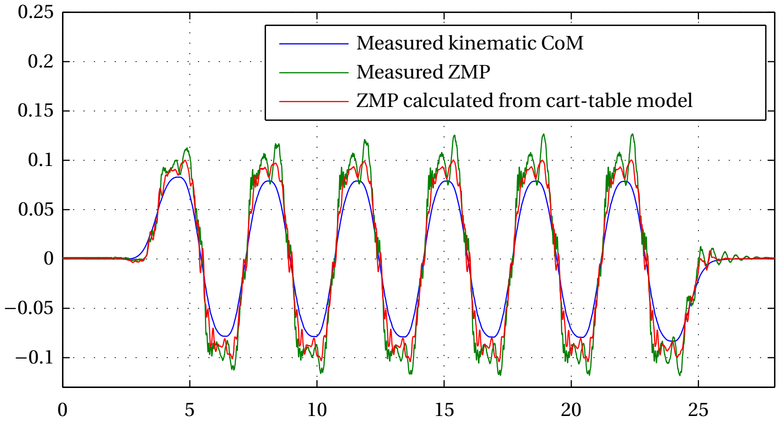


Figure 4.17. Comparison of ZMP measured FTS data and ZMP calculated from cart-table model.

In this figure the measured ZMP has a slightly higher amplitude than the ZMP calculated from the cart-table model. In the cart-table model only acceleration of the CoM is considered. The exact approach for moving the CoM will be explained in chapter 7, but the CoM is moved sideways by moving the pelvis in the transverse axis. For this to happen the legs must rotate with pivot about the ankle joints as shown in figure 4.18. This rotation introduces an additional moment which is not taken into consideration in the cart-table model.

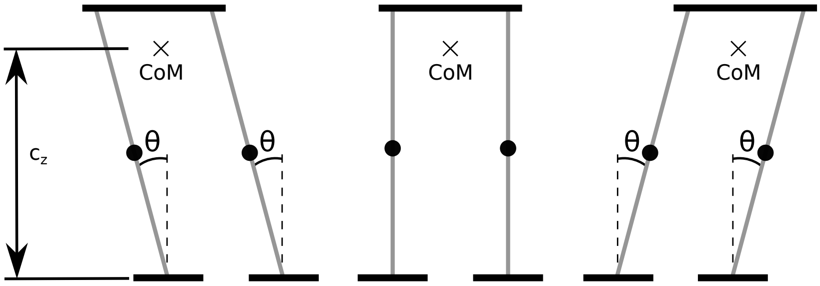


Figure 4.18. Posture of the legs for three different CoM positions.

By including the moment of inertia of the rotating legs, this moment can be introduced. It is suggested to consider both legs as a single rod with length c_z and uniformly distributed mass rotating about the end. The additional torque about the ankles can then be calculated:

$$\tau_L = J\ddot{\theta} = \frac{1}{3}m_L c_z^2\ddot{\theta} \qquad [\text{Nm}]$$

where:

J	is the moment of inertia of the legs	$[\text{kg}\,\text{m}^2]$
m_L	is the mass of the rotating part of the legs	$[\text{kg}]$
$\ddot{\theta}$	is the angular accelerations of the legs.	$[\text{rad}/\text{s}^2]$

Introducing this additional moment into equation (4.44) and assuming $\ddot{\theta} = \frac{\ddot{c_y}}{c_z}$ yields the following equation for the ZMP:

$$p_y = c_y - \left(1 + \frac{1}{3}\frac{m_L}{m}\right)\frac{c_z}{g}\ddot{c}_y \qquad \text{[m]} \quad (4.46)$$
$$= c_y - 1.13\,\frac{c_z}{g}\ddot{c}_y \qquad \text{[m]}$$

where:

$$m_L = 26.8\,\text{kg}$$
$$m = 67\,\text{kg}$$

Comparing equation (4.44) and equation (4.46) it can be observed that the contribution from the rotating legs can be seen as a linear contribution from the acceleration of the CoM to the ZMP. A factor of 1.13 does however not correct the difference seen in figure 4.17. By performing a least square curve fit in MATLAB® , a constant factor of 1.79 has been determined to give the best correspondence between measured ZMP and ZMP found through the cart-table model from the measured CoM. In figure 4.19 the results from a test with the adjusted cart-table model can be seen. With the increased factor the cart-table model and measured ZMP almost corresponds perfectly. The cart-table model with a factor of 1.79 multiplied to the CoM acceleration will be used as the model that correlates CoM and ZMP.

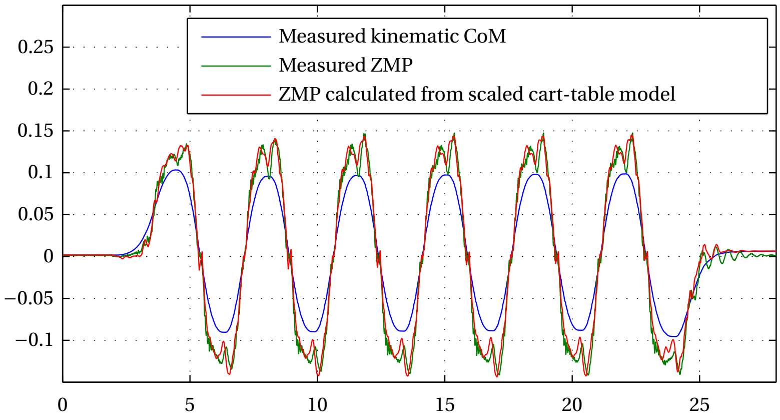


Figure 4.19. Comparison of ZMP measured FTS data and ZMP calculated from extended cart-table model.

In the sagittal axis a similar experiment has been conducted and the correction factor that gives the best correspondence between measured ZMP and ZMP found through the cart-table model is determined to be 1.72 . Using these constants the cart-table model for x and y-direction ends up as:

$$p_x = c_x - 1.72\,\frac{c_z}{g}\ddot{c}_x \qquad \text{[m]} \quad (4.47)$$

$$p_y = c_y - 1.79\,\frac{c_z}{g}\ddot{c}_y \qquad \text{[m]} \quad (4.48)$$

CHAPTER 5

POSTURE CONTROLLER

This chapter describes the design, implementation and test of the posture controller. Section 5.1 presents some of the considerations made before the actual design of the controller is done. This section also introduces the specifications of the posture controller. Section 5.2 describes how the gains of two different controllers are determined from simulations. Section 5.3 compares the performance of the two controller structures by performing tests on the robot. In section 5.4 the controller structure performing the best is selected and controller gains for all the joints are presented. Finally the design of a filter to estimate the states of the joints is presented in section 5.5.

Chapter Content

5.1 Overall Design of Posture Controllers

The task of the posture controller is to put all joints in positions specified by the gait-balance controller as described in chapter 3. Throughout this section all torques, inertia and other effects will be considered in the joints after the gears as proposed section 4.1.

Considering only one joint, the system can be seen as an apparently simple Single Input Single Output (SISO) system with an angle reference as input and a resulting angle as output. This seemingly simple system is complicated by a number of different factors. One of the problems is the cross coupling between the individual joints. Each time a joint is rotated this creates loads in many of the surrounding joints as well. A solution to this problem could be to calculate the influences between the joints and compensate for this through a feed forward signal. This would however be a very computational heavy task to do online for all 17 joints and is not considered a feasible solution. Instead it is chosen to see the cross couplings as disturbances in each joint. The controllers should then be designed such that they have a low sensitivity to external disturbances.

Another problem is the position dependent friction that was discovered in section 4.1.1. Although the amplitude of the position dependent friction seems low after the belts have been changed, it has still been observed to have a great effect. Using the PI-controller designed by the previous group on the knee joint, the joint velocity is changing in a way that causes the rest of the robot to shake. Hence this effect should also be suppressed by the designed controller.

The last but also greatest factor that makes the controller design more difficult is the fact that the robot is going through different support phases. This greatly influences the load on the motors in the legs. In single support phase a part of the controllers only have to cope with the weight and inertia of the free hanging leg. When the free hanging leg becomes the supporting leg, the load changes to be the whole robot instead of the leg. This causes the whole dynamic of the joints to change rapidly in time.

The previous project group working on the AAU-BOT1 tried to solve these problems with a simple PI-controllers [2]. To compensate for the change of load in the different support phases, they designed two different controllers for some of the joints and switched between them during gait. Initial tests of the knee joint controllers generally showed poor performance. They do not handle the varying friction very well, which caused shaky movements and they become unstable at joint angular velocities above 15 °/s. Some of the reason why this controller performs badly can be explained by the fact that the varying friction was most likely not a big issue at that time, and the varying friction has increased due to the two year standstill of the robot. Studying the results obtained by this group their results are not considered satisfactory. It is therefore desired to improve this performance of the posture controllers before any attempt of walking can be done.

5.1.1 Specifications

In order for the controllers to be useable, some general specifications are defined before the design begins. In order to specify the minimum bandwidth of the system the gait of a human being is studied. Considering the original target gait velocity of 1 m/s the step frequency is identified from the gait experiment described in section 3.5 to be approximately 1.67 Hz. The gait cycle frequency can not be used as a definitive indication of the maximum frequency content of the joint reference angle during gait. To have some extra bandwidth, the closed loop bandwidth (f_{min}) is chosen to be at least 4 Hz.

$$BW_{\text{min}} = f_{\text{min}}\,2\pi = 4 \cdot 2\pi = 25.1\,\text{rad/s}$$

The sampling frequency of the system has been chosen by a previous group [4]. The network inside the AAU-BOT1 has been designed to run at 250 Hz and it is therefore chosen to keep this rate. The maximum system bandwidth (BW_{max}) is chosen ten times lower:

$$BW_{\text{max}} = f_{\text{max}}\,2\pi = \frac{250}{10} \cdot 2\pi = 157\,\text{rad/s}$$

To suppress the external disturbances the disturbance rejection should be as high as possible. It is equally important to suppress both low frequency disturbances such as the influence of gravity and higher frequency disturbances such as the position dependent friction. Since it can be difficult to state a specific required disturbance rejection ratio, gains and controller structure should be chosen such that this remains high.

To ensure that the system is stable with model discrepancies and other unknown variations, the controllers will be designed with at least 45 ° of phase margin.

5.1.2 Controller Structure

A vast variety of different controller types exist. Since the designed controller needs to be run in 17 joints simultaneously, both computational effort and development time are important factors. As a first proposal two simple controller types will be designed, implemented and tested to examine performance. If these do not provide satisfactory results, other controller strategies must be investigated. The two simple controllers that will be attempted are a standard PID controller shown in figure 5.1 and a cascade controller with a structure as shown in figure 5.2. To enhance precision of the state measurements a Kalman filter is introduced in the feedback loop indicated by $H_\theta(s)$ and $H_\omega(s)$. It should be noted that this filter does not use angular velocity measurement from the EPOS although it is available. Instead backward difference is made on the angle readings. This is the reason why the angular velocity feedback arrow is indicated with dashes in figure 5.2. Further information of how this filter is designed and implemented can be found in section 5.5. During the controller design perfect state estimation is assumed and no dynamic of the sensors and filter is considered: $H_\theta(s) = 1$ and $H_\omega(s) = 1$. For the actual state estimation approach see section 5.5.

Throughout this design section, detailed information of the design will only be given on the right knee joint. The rest of the joints are designed from the same principles and stated in the end of this chapter.

First a model of the plant (the right knee joint) will be found. From (4.5) in the Joint Model section we have the following expression:

$$(J + J_m\, G^2\, \eta)\dot{\omega} = \eta\, G\, \tau_m - \tau_f - \tau_l \qquad \text{[Nm]} \quad (5.1)$$

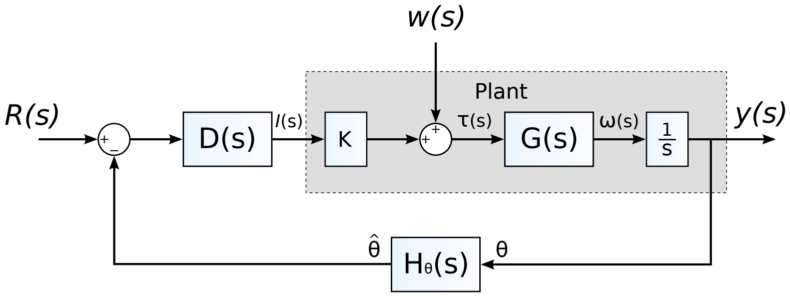


Figure 5.1. Block diagram of the PID controller. $R(s)$ is the reference angle, $y(s)$ is the resulting joint angle, $w(s)$ is a disturbance term collecting all unknown influences, $D(s)$ is the controller, K is a constant combining the motor constant with the gear ratio, the transfer function $G(s)$ specifies the dynamics of the plant, and $H_\theta(s)$ is the filter and sensor dynamic.

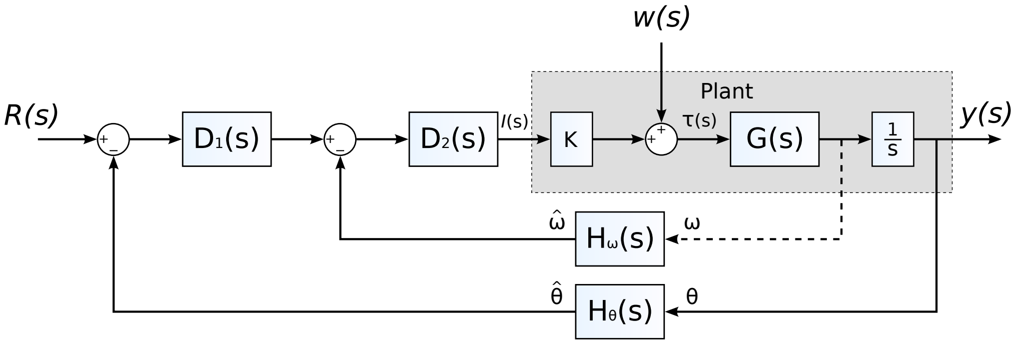


Figure 5.2. Block diagram of the cascade controller. $R(s)$ is the reference angle, $y(s)$ is the resulting joint angle, $w(s)$ is a disturbance term collecting all unknown influences, $D_1(s)$ and $D_2(s)$ denotes two different controllers, K is a constant combining the motor constant with the gear ratio, the transfer function $G(s)$ specifies the dynamics of the plant, H_θ, and $H_\omega(s)$ are the filter and sensor dynamics.

This model is simplified by considering only the viscous friction part of the friction model ($\tau_f = \omega\mu$) and considering the load torque and unmodelled non-linear friction as a disturbance $w(t)$. Introducing these simplifications and the expression for the motor torque given in equation (5.1) reduces to:

$$(J + J_m\, G^2\, \eta)\dot{\omega} = \eta\, G\, N_m\, k_t\, i(t) - \omega\, \mu + w(t) \qquad \text{[Nm]} \quad (5.2)$$

Collecting the total torque applied to the joint as the sum of the torque from the motor and the disturbance ($\tau(t) = \eta\, G\, N_m\, k_t\, i(t) + w(t)$) we finally get:

$$(J + J_m\, G^2\, \eta)\dot{\omega} = \tau(t) - \omega\, \mu \qquad \text{[Nm]}$$

Where:

$$\tau(t) = K i(t) + w(t) \qquad \text{[Nm]}$$

$$K = \eta\, G\, N_m\, k_t \qquad \text{[Nm/A]}$$

By Laplace transforming this expression, the transfer function $G(s)$ is identified:

$$G(s) = \frac{\omega(s)}{\tau(s)} = \frac{1}{s\left(J + J_m\, G^2\, \eta\right) + \mu} \qquad \left[\frac{\text{rad/s}}{\text{Nm}}\right] \quad (5.3)$$

It should be noted that the parameter G is describing the gear ratio of the joint and should not be confused with the transfer function $G(s)$.

All the terms in the plant model are known with the exception of J. This is the moment of inertia in the joint. When the leg is not supporting the robot, the contribution to inertia is only from the lower leg and the motor gear configuration. When the leg is fully supporting the robot, everything else but the lower leg contributes to the inertia and J naturally increases significantly.

In the design of the controllers both situations are considered. An estimate of the inertia will be calculated by simplifying the robot to a number of cuboids with uniformly distributed mass. The whole lower leg will be considered as one cuboid. Inserting the masses and dimensions[1] from appendix A, the inertia of the lower leg in pitch orientation can be calculated as:

$$J_y = \frac{1}{12}\, M \,\left(d_x^2 + d_z^2\right) = \frac{1}{12}\, 8.234 \,\left(0.150^2 + 0.484^2\right) = 0.176\,\mathrm{kg\,m^2} \tag{5.4}$$

where:

J_y is the moment of inertia of lower leg rotated about its centre $\left[\mathrm{kg\,m^2}\right]$

$M = 8.234\,\mathrm{kg}$ is the total mass of the lower leg

$d_x = 0.150\,\mathrm{m}$ is the depth of the lower leg

$d_z = 0.484\,\mathrm{m}$ is the height of the lower leg

To calculate the actual inertia in the knee joint, the inertia of the lower leg can be moved to the knee joint using the *Parallel Axis Theorem* [35]. The final moment about the knee is calculated in equation (5.5).

$$J_{\text{knee-small}} = J_y + M\,d^2 = 0.176 + 8.234 \cdot 0.242^2 = 0.658\,\mathrm{kg\,m^2} \tag{5.5}$$

where:

$d = 0.242\,\mathrm{m}$ is the distance between the two parallel axis

The large moment of inertia can be calculated by considering the rest of the body as three different cuboids. One for the right thigh, one for the left leg and one for the upper body including arms. These moment of inertia can be added together to find their total contribution to the moment of inertia about the right knee. The calculations will not be presented here, but the principle is the same as above. The moment of inertia is calculated to be: $J_{\text{knee-large}} = 16.016\,\mathrm{kg\,m^2}$.

[1] Only the lengths of the limbs are available from the appendix as the distance between the joints. The other dimensions needed are measured on the robot.

Inserting the known values into K and $G(s)$ gives us the equations of the plant model.

$$K = k_t \eta G N_m = 0.0603 \cdot 0.82 \cdot 133 \cdot 2 = 13.152\,\mathrm{Nm/A} \tag{5.6}$$

$$G_{\text{small}}(s) = \frac{1}{s\left(J_{\text{knee-small}} + J_m\, G^2\, \eta\right) + \mu} = \frac{1}{1.413\,s + 7.07}\,\frac{\mathrm{rad/s}}{\mathrm{Nm}} \tag{5.7}$$

$$G_{\text{large}}(s) = \frac{1}{s\left(J_{\text{knee-large}} + J_m\, G^2\, \eta\right) + \mu} = \frac{1}{16.771\,s + 7.07}\,\frac{\mathrm{rad/s}}{\mathrm{Nm}} \tag{5.8}$$

Where:

μ is found for the right knee in table 4.2 [Nm/rad]

$J_m\, G^2\, \eta$ is found for the right knee in table 4.1 $\left[\mathrm{kgm}^2\right]$

To be able to compare the disturbance rejection of the two controllers, transfer functions expressing the output $y(s)$ with respect to the disturbance $w(s)$ must be found. For the PID controller this is straight forward, from figure 5.1 the disturbance rejection (DR) can directly be stated as:

$$DR_{\text{PID}}(s) = \frac{y(s)}{w(s)} = \frac{G(s)}{s + K\,D(s)\,G(s)} \quad \left[\frac{\mathrm{rad}}{\mathrm{Nm}}\right] \tag{5.9}$$

In the cascade controller the influence of the disturbance is slightly more complicated. To clarify the deduction of disturbance rejection given in equation 5.10, a block diagram of the system seen with the disturbance as the input, is seen in figure 5.3. Notice that this figure does not show the feedback transfer function $H(s)$, but the states are the estimates as indicated by the notation.

$$DR_{\text{casc}}(s) = \frac{y(s)}{w(s)} = \frac{G(s)}{s\,(1 + K\,D_2(s)\,G(s)) + K\,D_1(s)\,D_2(s)\,G(s)} \quad \left[\frac{\mathrm{rad}}{\mathrm{Nm}}\right] \tag{5.10}$$

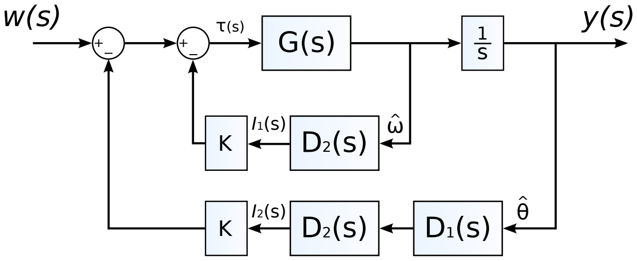


Figure 5.3. Block diagram of disturbance rejection in the cascade controller.

A structure of the controller $D(s)$ in figure 5.1 must be chosen. To avoid any steady state errors in the position and suppress other low frequency disturbances, an integral term is introduced. Since the plant already contains one pole in zero and another one close to zero (exact pole position depends on inertia) a phase of 180 ° is present even at low frequencies. It is therefore needed to introduce a differential term to the controller in order to obtain stability. Hence $D(s)$ will be designed as PID controller.

In the cascade controller the inner loop controller $D_2(s)$ is chosen to be a P controller. The main objective of this controller is to suppress disturbances. The outer controller $D_1(s)$

is chosen to be a PI controller to ensure zero steady state error. The controllers are specified as follows:

$$\begin{aligned}
D(s) &= K_P + K_I\,\frac{1}{s} + K_D\, s \\
D_1(s) &= K_{P1} + K_{I1}\,\frac{1}{s} \\
D_2(s) &= K_{P2}
\end{aligned}$$

5.2 Controller Design

For the two different controllers, two sets of gains will be found. One for each plant model designed from different inertia. To design the PID controller the MATLAB tool `pidtool()` is used. Through this tool the three parameters of the PID controllers are found by specifying a desired phase margin of 45 ° and a crossover frequency of 40 rad/s. It should be noted that the crossover frequency could have been chosen much higher and thereby get a higher bandwidth. But 40 rad/s was chosen such that the found PID gains were generally smaller than the ones found by the previous group of $K_\text{P} = 550$ and $K_\text{I} = 55$, since this controller showed to be only marginally stable.

$$\text{Gains for } G_\text{small}(s): \begin{cases} K_\text{P} = 112 \\ K_\text{I} = 77 \\ K_\text{D} = 2.2 \end{cases} \rightarrow \begin{cases} \text{PM} & = 45\,^\circ \\ \text{Min. } DR & = -63\,\text{dB} \\ \text{CLBW} & = 58\,\text{rad/s} \end{cases}$$

$$\text{Gains for } G_\text{large}(s): \begin{cases} K_\text{P} = 416 \\ K_\text{I} = 183 \\ K_\text{D} = 17.3 \end{cases} \rightarrow \begin{cases} \text{PM} & = 45\,^\circ \\ \text{Min. } DR & = -74\,\text{dB} \\ \text{CLBW} & = 34\,\text{rad/s} \end{cases}$$

In the cascade controller structure, the inner loop controller $D_2(s)$ is designed first. Since the plant only has one pole, K_P2 can be set to any value and still stabilise the system. To create the highest possible disturbance rejection the inner loop gain should be as high as possible. At the same time it must also be made sure that the closed loop bandwidth of inner loop does not exceed 157 rad/s as specified in the controller specifications. The maximum bandwidth limitation sets a constraint on $K_\text{P2} < 14$ when the low inertia plant is present, and $K_\text{P2} < 165$ when high inertia is present.

Initial tests however showed that for the low inertia it is not possible to increase the inner gain to more than 7 before instability becomes an issue. For this reason a $K_\text{P2} = 5$ is chosen for the knee joint which results in the following closed loop properties of the inner loop.

$$\text{for } K_\text{P2} = 5: \begin{cases} \text{PM} & = 95\,^\circ \\ \text{Min. } DR & = -39\,\text{dB} \\ \text{CLBW} & = 61.5\,\text{rad/s} \end{cases}$$

In the outer loop the two controller gains are K_P1 and K_I1. The proportional gain is desired to be as big as possible to give a fast system response and suppress disturbances. $K_\text{P1} = 100$ has been chosen since this is the largest possible value of the proportional gain maintaining a phase margin at 45 °. The magnitude of $K_{\mathrm{I1}} = 50$ is chosen such that memory of previous error does not dominate the current input references. With these gains the closed loop properties of the cascade controller with low inertia plant becomes:

$$\text{for } K_{\mathrm{P1}} = 100,\ K_{\mathrm{I1}} = 50,\ K_{\mathrm{p2}} = 5: \begin{cases} \text{PM} & = 44\,° \\ \text{Min. } DR & = -76\,\text{dB} \\ \text{CLBW} & = 103\,\text{rad/s} \end{cases}$$

The approach for designing the cascade controller with high inertia is the same and results in the following controller and closed loop properties.

$$\text{for } K_{\mathrm{P1}} = 100,\ K_{\mathrm{I1}} = 50,\ K_{\mathrm{P2}} = 75: \begin{cases} \text{PM} & = 45\,° \\ \text{Min. } DR & = -99\,\text{dB} \\ \text{CLBW} & = 114\,\text{rad/s} \end{cases}$$

To find out which of the controller structures that is the best for the knee joint, a comparison of the theoretical performance is carried out. Since both controllers were designed with the objective of 45 ° phase margin the stability properties are the same, not considering disturbances. From the step response of the four controllers shown in figure 5.4 it is clear that the larger closed loop bandwidth of the cascade controllers leads to better rise times.

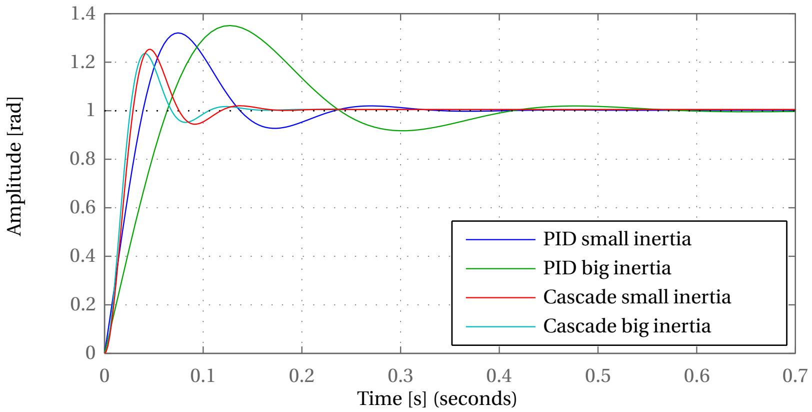


Figure 5.4. Simulated step response of the four different controllers.

What is even more important in terms of choosing the right controller is the disturbance rejection. Looking at figure 5.5 it is seen that the cascade controllers generally have a better suppression of disturbance. Judging from the results shown in the two figures the cascade controller seems to be the best choice.

5.3 Controller Test

To test which, if any, of the controllers that is suitable as posture controller in the knee, the controllers are implemented on the robot and tested by exciting the joint with different

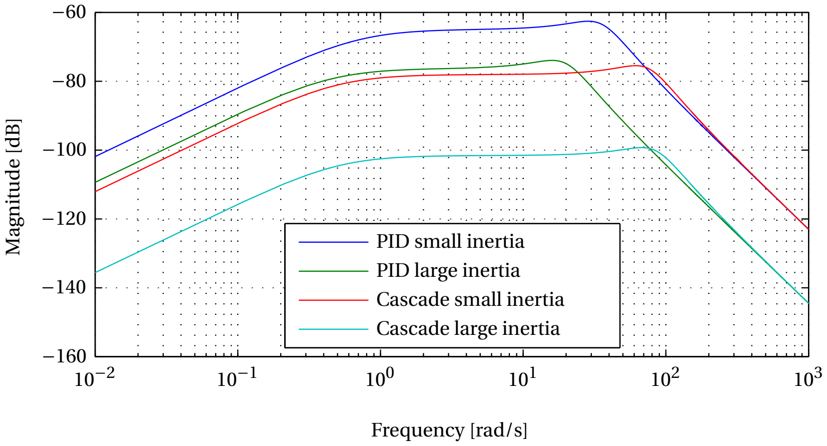


Figure 5.5. Disturbance rejection for the four different controllers.

references. In the following plots the measured data are the recorded estimated states. Errors are found by subtracting the reference from the estimated state.

The first test that is performed is a step response on the low inertia system with the right lower leg hanging off the ground. Figure 5.6 shows the result of the test where the system is excited with a step with an amplitude of 0.175 rad. To ease comparison with figure 5.4 the amplitude of the angle has been scaled to one.

Comparing the simulated step response in figure 5.4 with the measured in 5.6 it is clear that the rise time is longer in practice. The fact that both controllers show exactly the same response in the first part of the climb points towards the fact that limitations of the motor/EPOS configuration is to blame for this. Hence no controller will be able to decrease the rise more than what is shown in figure 5.6. The cascade controller shows a settling characteristic as expected by a linear dynamic system swinging over and under the reference in a sinusoidal way. The PID controller on the other hand looks misshapen in the first overshoot phase. A normal linear system would not be able to show this characteristic. The most likely cause for this behaviour is friction. In the dynamic model of the joint only the linear viscous friction has been included. The coulomb friction will help stop the movement faster than in the simulated response. This influence should also be apparent in the cascade controller, but since this controller is better at rejecting disturbances, the influence of friction is reduced. From the step response it can be concluded that both controllers are stable. Furthermore it seems to indicate that the cascade controller is generally less influenced by disturbances such as friction, but has an overshoot greater than the PID controller.

To further analyse the performance concerning disturbance rejection, an angle reference introducing constant angular velocity is applied. The reference has the same shape as shown in figure E.1. The velocities applied are 1 °/s, 5 °/s, 20 °/s, and 40 °/s starting with the lowest one first. The angle error using the two different controllers is shown in figure 5.7 and 5.8. Figure 5.7 shows the first 105 s of the experiment where the angular velocity is 1 °/s. For the PID controller it is clearly seen that after steady state has been reached, the position dependent friction identified section 4.1.1 influences the error significantly. Figure 5.8 shows the angular error for the last 40 s of the experiment where the velocity is greater. It

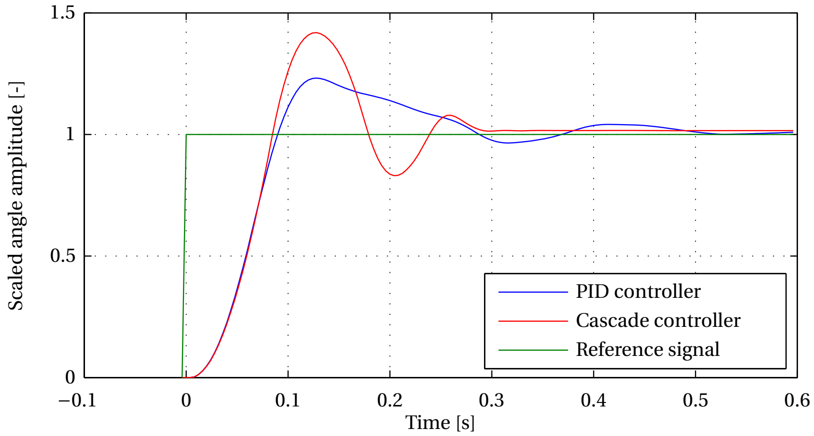


Figure 5.6. Comparison of the step response using the PID and cascade controller respectively. The reference has an amplitude of 0.175 rad, all recorded data has been scaled to one.

is seen that the difference between the two controllers becomes less pronounced for higher velocities where the inertia of the lower leg helps keeping a constant velocity.

Again the cascade controller shows better rejection of disturbances. The better performance generally becomes less apparent at higher angular velocities, but the angular error remains less than for the PID controller at all times. Although the angular errors seen with the PID controller seems small, the varying velocity propagates vibrations into the rest of the robot and is therefore undesirable.

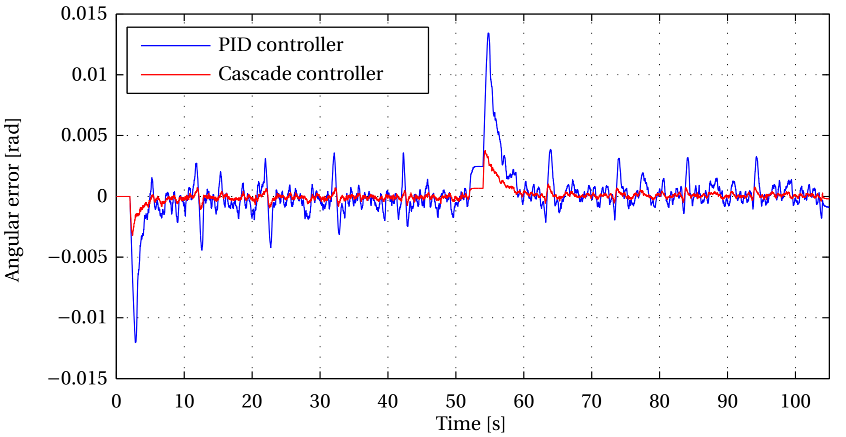


Figure 5.7. The first 105 seconds of joint angle error applying ramp input.

An additional test is supposed to analyse the performance of the controllers at higher angular velocities. A smoother sinusoidal angle reference is used to protect the setup. The sinusoidal input has a constant amplitude of 25 °, but the frequency is increased over time until a joint angular velocity of approximately 200 °/s is reached. The angle error of this experiment is shown in figure 5.9. The overall angle error with the two controllers is seen to have approximately the same amplitude and the system remains stable. The cascade controller

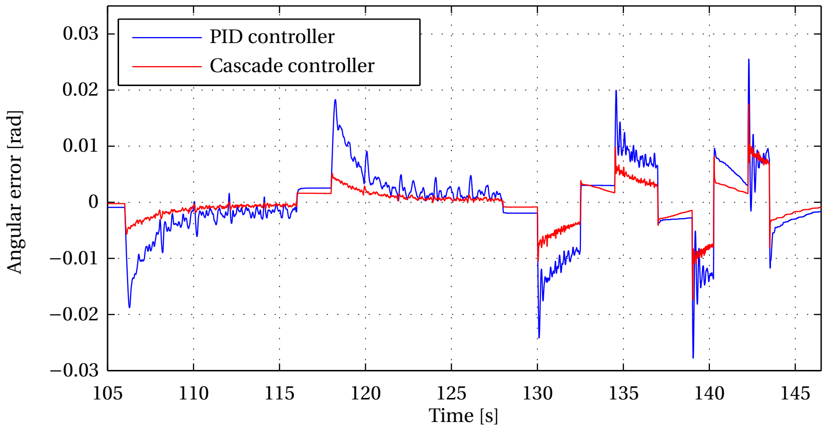


Figure 5.8. The last 40 seconds of joint angle error applying ramp input.

generally performs slightly better, but not significantly. The shape of the error does however reveal a difference between the two controllers. Where the cascade controller shows an almost linear dependence between error and angular velocity, the PID controller shows a more distorted relation. Again it seems like the PID controller is a more influenced by disturbances.

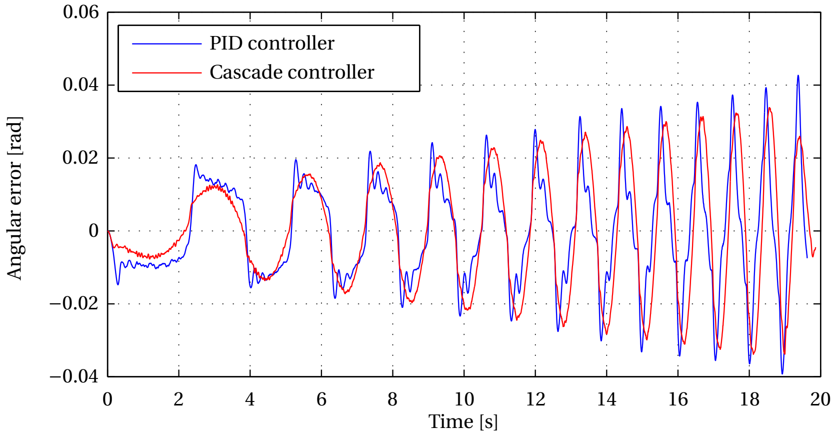


Figure 5.9. Step response of the four different controllers.

All the tests described above should be performed for the high inertia case as well. But since keeping balance standing on one leg while trajectories are applied to the knee joint is practically impossible this can not be done. Instead some modification of the sinusoidal test was made. Instead of trying to balance on one foot, the AAU-BOT1 was placed with both feet planted on the floor and the input reference was applied to both knees. This influences the inertia calculated earlier which reduces to less than half of the original calculated. It is however the only reasonable way to test the knee joint under load. Simulations showed that the designed high gain controllers should still remain stable under the increased inertia. However in practice the PID controller showed to be only marginally stable, while the cascade controller was unstable. Through a trial and error approach, different controller gains were attempted until a reasonable result was achieved. For the PID controller $K_\text{P} = 200$, $K_\text{I} = 100$, and $K_\text{D} = 8$ showed reasonable results. For the cascade controller increasing the inner loop gain K_P2 to more than 7 was not possible even with the increased inertia. Changing K_P1 and K_I1 from their original designed values of 100 and 50 only gave worse results. In the end the best gains turned out to be the same as for the low inertia case. The angle errors with these two controllers are shown in figure 5.10. It should be noted that the amplitude of motion was limited to 5 ° and maximum angular velocity was around 15 °/s. Otherwise the motion was too violent to be controlled.

Comparing the two controllers, the cascade controller still performs the best although it has been left unchanged. The shape of the error is now closer to being the same, but the cascade controller results in only half the error.

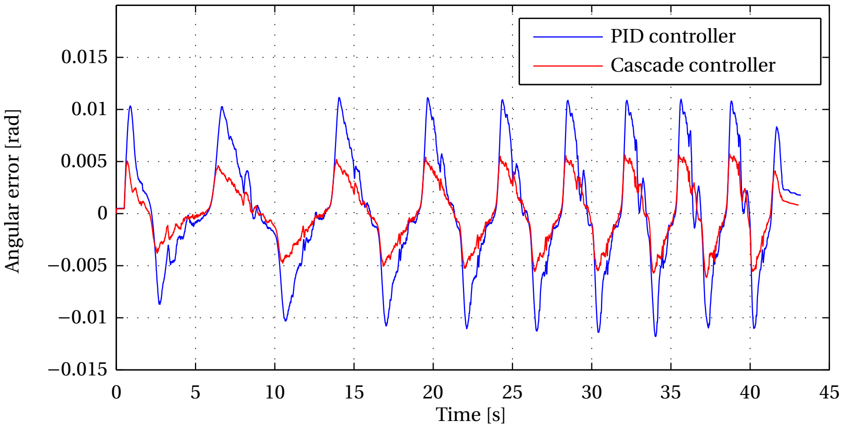


Figure 5.10. Angle errors for the two controllers when standing on both feet and moving the body by rotating the knees with a sinusoidal reference.

The final test which is carried out is a performance comparison of the designed cascade controller compared to the PI-controller designed by the previous project group. This test is meant to show if the designed controllers are stable when they are used in multiple joints and if performance is better than what has been achieved earlier. The gains used for the cascade controllers are shown in table 5.1.

The trajectory chosen for this test is an ellipse shaped motion of the CoM, which can be seen as the blue curve in figure 5.11. This motion requires that all joints in the legs rotates, which tests the combined performance. The joint trajectories are created from the kinematics deduced in section 4.2.

In figure 5.11 the resulting CoM of the new and old controller is seen. From this figure it can be seen that both controllers are capable tracking the joint references such that the desired CoM is obtained. It is however apparent that the old PI-controller in some places has less tracking error than the new cascade controller. This become more visible in figure 5.12, where the CoM movement in the y-axis has been plotted as a function of time. At around 3 s and 13 s where

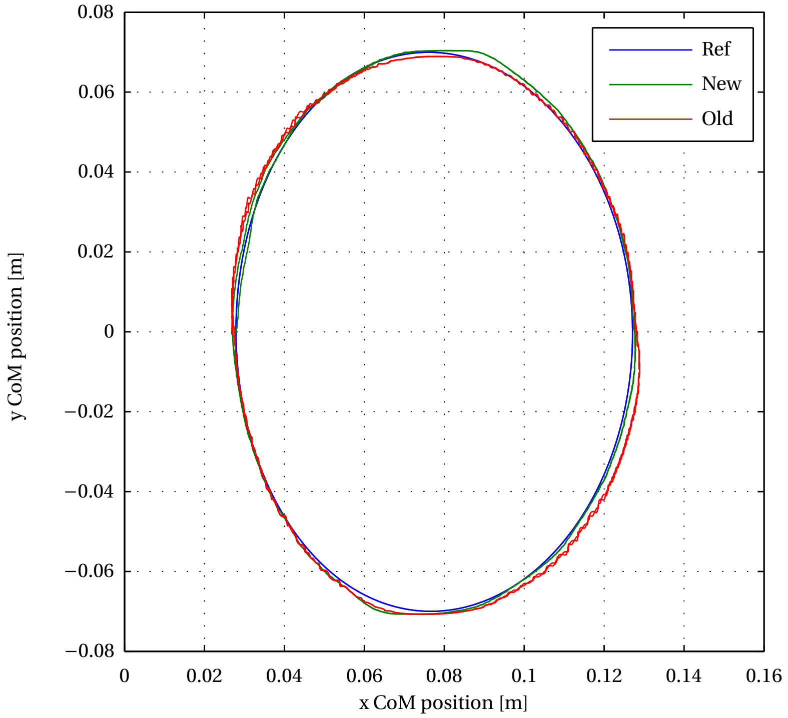


Figure 5.11. Comparison of the CoM movement using previous group PI controller and the new cascade controller.

the CoM position is at its maximum the old controller tracks the reference slightly better. At around 8 s and 18 s where CoM reaches its minimum y-position the performance of the two controllers seems to be equal, and with both controllers the robot gets a little stuck and starts to move later than desired.

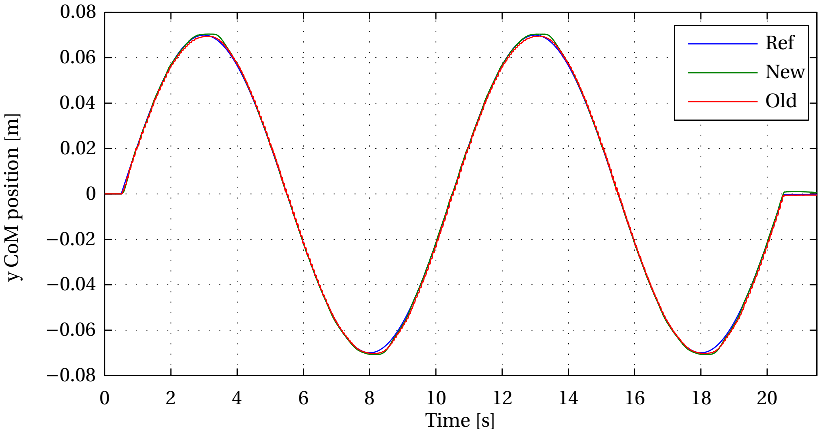


Figure 5.12. CoM movement on the y-axis plotted as a function of time.

An important difference between the two controllers can be seen in figure 5.13 where a zoom of figure 5.12 is shown. From this it is seen that the new cascade controller nicely follows the reference until it reaches the maximum value where it gets stuck for a while before the movement in the opposite direction begins. For the PI-controller the overall obtained motion is more desired since the robot does not get stuck in the same way. What should be noted in figure 5.13 is the shaky motion obtained when using the PI-controller. In contrast the cascade controller shows smother movement. During gait it is vital that the movement is kept as smooth as possible to avoid vibrations in the entire robot. For this reason the new developed controllers are significantly better than the old PI-controllers although the trajectory following capabilities does not seem much improved.

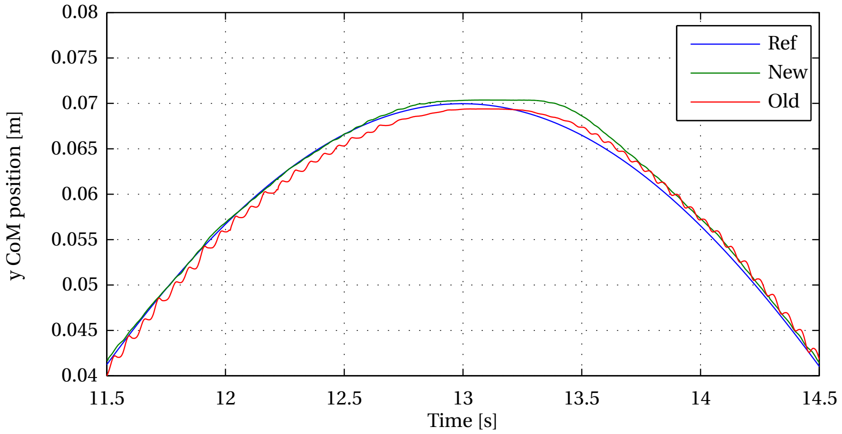


Figure 5.13. A zoom of the CoM movement on the y-axis plotted as a function of time.

5.4 Final Controller Choice

From all the above tests the cascade controller shows the best performance overall. It is therefore chosen to implement this type of controller in all the joints. Generally the joint angle errors have been shown to be small. Initially it was thought that it would be necessary to change controller during gait, to compensate for the shifting inertia. From figure 5.10 and tests performed on the ankle pitch joints, it has however been chosen only to use one controller for each joint. As mentioned earlier an alternative way of compensating for the shifting inertia could also be to compute a feed forward current to help the controller. At this point it does however not seem necessary to do this. Similarly compensation for the friction could also be done through feed forward, but it is chosen not to do so because of the good results obtained by the cascade controller.

For the rest of the joints in the AAU-BOT1 the controllers are designed from the same considerations as presented in this chapter. All the controllers have been tested in practice and the gains have been adjusted if deemed necessary. The final controller gains for all the joints can be found in table 5.1.

Joint number	Joint name	K_{P1} (outer)	K_{I1} (outer)	K_{P2} (inner)
1	Right ankle roll	100	50	3
2	Right ankle pitch	100	100	5
3	Right knee	100	100	5
4	Right hip pitch	100	100	5
5	Right hip roll	100	200	5
6	Right hip yaw	100	50	3
7	Left hip yaw	100	50	3
8	Left hip roll	100	200	5
9	Left hip pitch	100	100	5
10	Left knee	100	100	5
11	Left ankle pitch	100	100	5
12	Left ankle roll	100	50	3
13	Pelvis yaw	40	20	3
14	Waist pitch	50	25	10
15	Waist roll	50	25	5
16	Left shoulder	50	25	1
17	Right shoulder	50	25	1

Table 5.1. Controller gains for all cascade controllers.

Anti-windup

The DC-motors in each of the joints have limitation to how much current they can handle. To avoid overloading the motors a current saturation is inserted before the currents are applied to the robot. Since this could cause windup errors in the outer controller, an anti-windup circuit is implemented in the cascade controller as seen in figure 5.14. The chosen structure assumes that current saturation will only occur when the joint experiences a high load that causes the angular velocity to be near zero. In this case the feedback in the inner controller will only have small effect and the inner controller can be seen as a multiplication of the output from the outer controller. Under this assumption the feedback gains in this circuit are determined. The error occurring in case of a saturation is first divided by the inner controller gain K_{P2} to reduce the influence of this controller. The other feedback gain K_a is design by

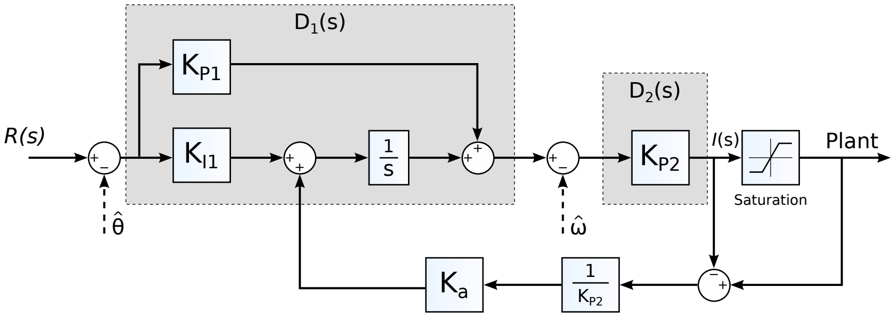


Figure 5.14. Block diagram of the anti-windup circuit used to remove windup problems in the controller.

calculated from the integral time:

$$K_a = \frac{1}{T_i}$$

where:

$$T_i = \frac{K_{I1}}{K_{P1}} \text{ is the integral time}$$

With the chosen design of the anti-windup loop, problems can occur if the saturation is caused by an error between the reference and measured angular velocity in the inner loop. This situation is not likely to happen and will not be addressed.

5.5 Angle and Angular Velocity Estimation

As described in section 5.2 the controllers assumes perfect knowledge of the states. To estimate the angle and angular velocity of the joints a Kalman filter is used. Three different sensor measurements are available from the joints as described in section 2.3. Two measurements of the angle with a resolution of 0.0879 ° and 0.0014 ° respectively and an angular velocity measurement with a resolution of at least 1.4 °/s.

In addition to the measurements, a model of the dynamics in the joint must be used. Due to the angle dependent friction in the joints identified in section 4.1.1 the derived joint model in section 4.1 will not add much information to the Kalman filter. By utilising the high sampling frequency and the precise angle measurements it is observed that a simpler model assuming zero acceleration performs well, hence this will be used. In the Kalman filter only angle measurements from the motor encoders are used to estimate the angular velocity. In this way the velocity is calculated by backward difference of the angle and is lowpass filtered. The velocity measurement from the EPOS is not used since the resolution of the angular velocity calculated by backward difference is higher. For a single joint the discrete time linear system model is chosen as follows:

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{w}_k$$

$$y_k = \mathbf{C}_k \mathbf{x}_k + v_k$$

Where:

$$\mathbf{x}_k = \begin{bmatrix} \theta_k & \omega_k \end{bmatrix}^T \text{ is the joint states}$$

$$\mathbf{A}_k = \begin{bmatrix} 1 & Ts \\ 0 & 1 \end{bmatrix} \text{ is the simplified joint system dynamics}$$

$$\mathbf{C}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \text{ is the system output matrix}$$

y_k is the system output

$\mathbf{w}_k$ is white process noise with covariance $\mathbf{Q}_k$

v_k is white measurement noise with covariance R_k

Ts is the sampling time of the discrete time system (and filter)

The filter equations used to estimate the angles are now presented [36]. The first step is to predict the new states:

$$\begin{aligned}
\hat{\mathbf{x}}_k^- &= \mathbf{A}_k \hat{\mathbf{x}}_{k-1}^+ \\
\mathbf{P}_k^- &= \mathbf{C}_k \mathbf{P}_{k-1}^+ \mathbf{C}_k^T + \mathbf{Q}_k
\end{aligned}$$

Followed by an update step:

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + R_k)^{-1} \\
\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (z_k - \mathbf{C}_k \hat{\mathbf{x}}_k^-) \\
\mathbf{P}_k^+ &= (I - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_k^-
\end{aligned}$$

Where:

$+/-$ indicates if estimate is after / before update step

$\mathbf{P}_k$ is the covariance of the state estimate

$\mathbf{K}_k$ is the Kalman gains

z_k is the sensor measurement

To tune the lowpass effect of the filter, the variances for the filters are adjusted. Since the best knowledge of the angle is the measurement, the variance for the measurements is chosen low. Choosing a low variance for the model of the velocity implies a smooth velocity estimate, which however is delayed due to the constant velocity assumption in the model. Since a delay is not desired in the control loop, the variance for the model of the velocity is chosen high which results in only a minor lowpass filtering effect. The resulting variance matrices for the Kalman filter is chosen as follows:

$$\begin{aligned}
\mathbf{Q}_k &= \mathrm{diag}(1, 1\mathrm{e}^6) \\
R_k &= 1
\end{aligned}$$

Missing Sensor Data

If sensor data from the angle measurement is missing (two subsequent samples are the same) the estimated angular velocity will be close zero at the time instance when the sensor data is missing and be close the double of the true velocity at the next time instance. This is a result of the model structure and the fact that the measurements are trusted much more than the model. An example of this is seen in figure 5.15.

To overcome this undesired behaviour and obtain the result shown with red in the figure, the reading from the sensor is disregarded by specifying a high variance for the measurement at the sample instance where the data is detected as missing. The measurement variance is updated according to the following rule:

$$R_k = \begin{cases} 1\mathrm{e}^{16} & \text{if } |\omega_{k-1}| > \omega_{th} \text{ and } |z_k - z_{k-1}| < \Delta z_{th} \\ R_k & \text{otherwise} \end{cases}$$

At low velocities the missing sensor data is not as harmful to the filter as for high velocities and due to the fact that a simplified model of the system is used it is chosen that the sensory

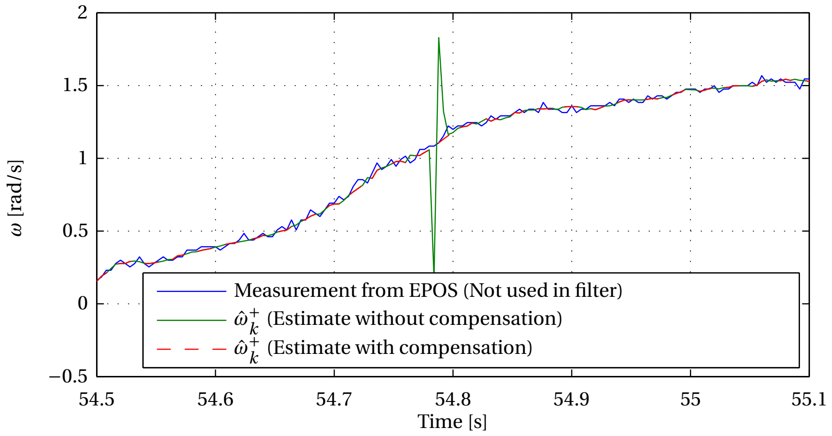


Figure 5.15. Figure showing measured and estimated joint angular velocities when a sample is missing.

information is not disregarded if the estimated velocity is below a threshold $\omega_{th} = 0.01\,\text{rad/s}$ to be sure that the filter does not diverge. If the estimated velocity above ω_{th} the sensor data is disregarded if the difference between the last and current measurement corresponds to an angular velocity below 0.001 rad/s $\Rightarrow \Delta z_{th} = 0.001 \cdot Ts$.

Initial State and Covariance

To initiate the filter the initial state vector and covariance matrix must be defined. Since we know that the system is in zero position and having zero velocity the initial states are chosen as zero and as covariance the identity is used:

$$\hat{\mathbf{x}}_0^+ = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$

$$\mathbf{P}_0^+ = I_{2\times 2}$$

The Resulting Filter

To estimate the angular velocity of all seventeen joints in the robot, one large filter is used, where all angles and angular velocities are collected in one state vector. The dynamics, process, and sensor noise for all joints are assumed to be the same, and the states and the filter matrices for the total filter become:

$$\mathbf{x}_k = \begin{bmatrix} \theta_k^1 & \theta_k^2 & \dots & \theta_k^{17} & \omega_k^1 & \omega_k^2 & \dots & \omega_k^{17} \end{bmatrix}^T$$

$$\mathbf{P}_k \in \mathbb{R}_{34\times 34}$$

$$\mathbf{Q}_k = \text{diag}(I_{17\times 17}, I_{17\times 17} \cdot 1\text{e}^6)$$

$$\mathbf{R}_k = I_{17\times 17}$$

In the resulting filter the update of the sensor noise due to missing sensor data is made for each individual joint by updating the corresponding entry on the diagonal in $\mathbf{R}_k$ and thereby not disregard the total measurement vector when sensor data from a single sensor is missing.

For a test on a single joint the estimate of the angular velocity if shown in figure 5.16 alongside with the measurement of the angular velocity from the EPOS and the angular velocity calculated by backward difference directly from the angle measurements. In figure 5.17 the estimated angle of the joint is shown alongside with the angle measurement from the EPOS.

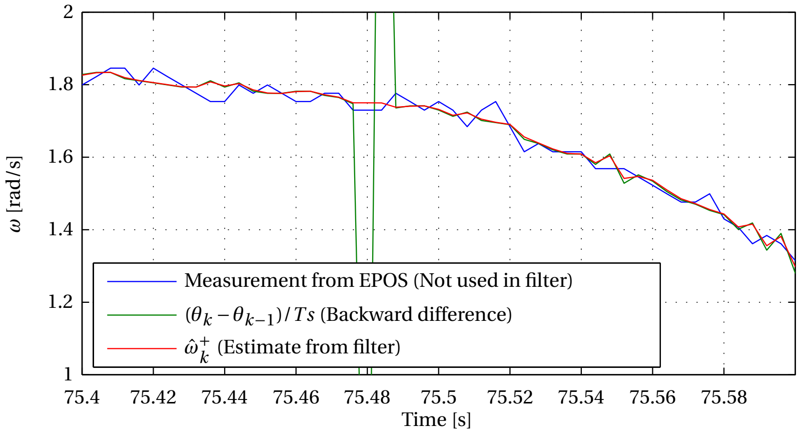


Figure 5.16. Measured, calculated, and estimated angular velocity of left knee joint on a test run.

From figure 5.16 it is seen that the overall shape of the estimated angular velocity is the same as the velocity measurement from the EPOS. Additionally it is seen that the estimate is smoother that the measurement. Compared to the backward difference method it is seen that there is not a that big difference despite of the compensation for missing sensor data which is seen at time 75.48 s where the spike observed when using the backward difference method is not present in the estimate form the filter.

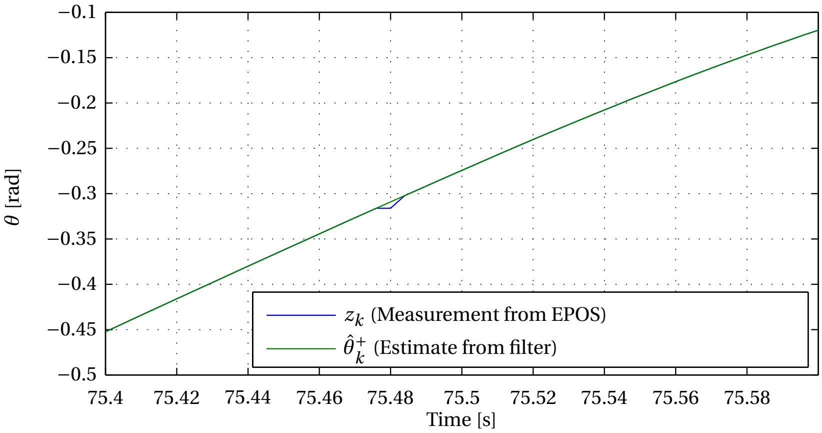


Figure 5.17. Measured and estimated angle of left knee joint on a test run.

From figure 5.17 another benefit of using the Kalman filter to estimate the states is seen. Even though a measurement is lost the angle is precisely estimated from the assumed constant velocity.

Assessments

As seen in figure 5.16 and 5.17 the resulting filter seems to fulfil the task of estimating the angular velocity. With the chosen setup of the filter, it does not show a much better estimate of the angular velocity compared to the result obtained by backward difference despite of the compensation for the missing samples. This compensation could have been implemented manually without the use of a Kalman filter and thereby reduce the computational burden even more. It is however chosen to use the Kalman filter since it is fairly easy to set up and it is easy to implement the desired features such as compensating for the missing samples. In addition it is straight forward to extend the filter later e.g. by including some of the dynamics in the joint.

CHAPTER

6

Balance Perception and Stability

This chapter contains a description different measures of balance and how to access balance and stability for the biped robots. In section 6.1 the way of accessing stability and balance is discussed. In section 6.2 the Polygon of Support (PoS) for the AAU-BOT1s feet is analysed. Discrepancies between the measured CoM through the kinematics, the measured ZMP and the size of the PoS are identified and corrections in the calculation of the CoM are proposed. Finally in section 6.3 a Kalman filter to estimate CoM and ZMP is designed.

Chapter Content

6.1 Balance and Stability

As introduced in section 2.2 *Balancing Points*, different approaches can be taken to assess stability of gait. Besides from studying CoM, ZMP, or CoP, as introduced in this section, other methods exist, such as orbital stability based Poincaré first-return map [37]. Some methods rely on multiple recorded steps in order to assess stability of the gait. This makes them mainly usable for offline assessments. Others rely only on current measurements and can be used online.

For control purposes the methods using only current measurements are the most obvious choice. The method that will be used in this project is the criterion of dynamic stability [11]. This criterion specifies a biped robot to be dynamically stable if the ZMP stays within PoS at all times. If the CoM is within the PoS at all times the biped is statically stable. When the biped is statically stable during a gait the gait will be referred to as a static gait. If the biped is dynamically stable but not statically stable the gait will be referred to as a dynamic gait. To access the quality of the gait i.e. how stable the gait is the stability margin will be defined as:

> **Stability margin:** *The minimum distance from the balancing point (CoM or ZMP) to the edge of the PoS.*

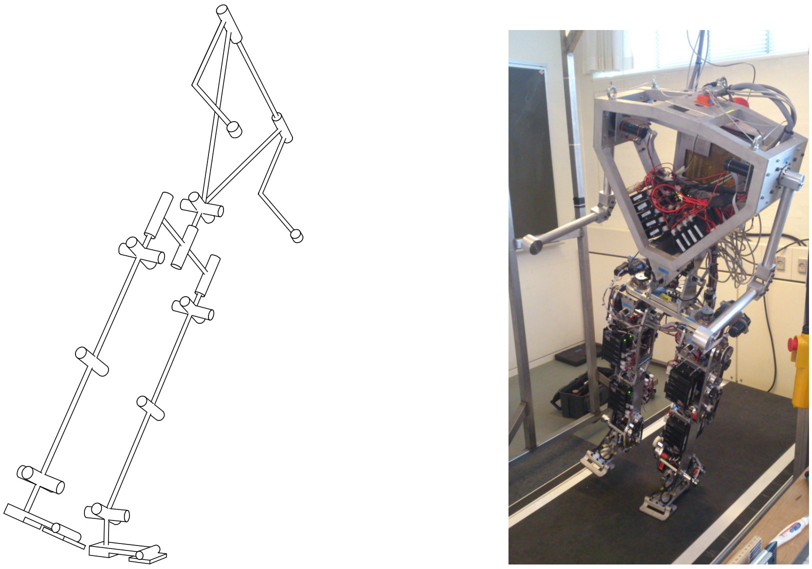
In the next section the PoS of the robot will be studied and the size of the PoS determined such that the stability margin can be accessed for a gait.

6.2 Study of PoS

While the PoS for a single foot should be the actual footprint experiments have shown that only a part of the foot can be regarded as supporting. To determine how much of the foot that can be regarded in determination of the PoS an experiment has been conducted. As described in section 2.2 CoM and ZMP are two points that can be used to determine if the robot is in balance. For the static case or for very slow movements the ZMP and the GCoM will coincide. This can be shown from the definition of the ZMP stated in section 2.2.2 in the case where only moment contributions from gravitational forces exist. If the ZMP and the GCoM are within the PoS the robot will balance otherwise it will fall. This is used in the experiment where it is desired to determine the PoS by measuring the CoM and ZMP in both double and single support, in the case where the robot is just about to fall. To determine the CoM of the robot, the forward kinematics described in section 4.2.1 is used. To determine ZMP, measurements from the FTS are used as described in section 2.3.

In the first part of the test it is desired to measure the foremost and the rearmost point of the PoS. The robot is placed on the ground with all limbs fully extended and both feet flat on the ground. The pitch angle of the ankles is slowly increased to move the CoM forward until the robot falls. The time where the robot falls is registered, to be able to identify joint angles measured by the motor encoders, forces and torques measured by the FTS's, and angles measured by the IMU. The experiment is repeated with different configurations also including hip pitch and waist pitch. In the same way tests are carried out to determine the rear most point of the PoS.

In the second part of the test, the width of the PoS of the feet is determined. In this test the robot is placed in an initial position with the CoM centred one of the feet according to the forward kinematics. The initial position is obtained using only roll and pitch angles in the corresponding ankle. For one test the initial posture is shown in figure 6.1. The roll angle is then either increased or decreased slowly to determine either the left or right side of the PoS of the foot. At the time where the robot falls over the side of the foot, the time is captured. The experiment is repeated to obtain four measurements for each side of each foot.



(a)

(b)

Figure 6.1. Initial posture for one of the experiments where the width of the left foot PoS is determined.

The results from the experiments are shown in figure 6.2. On this figure the calculated CoM, at the time where the robot falls, is indicated by a cross, and the corresponding measurement of the ZMP is indicated by a circle. Two different PoS are indicated for the two feet. *PoS 1* is the complete area given by the entire heel and the toe joint placed in the centre of the toe. *PoS 2* is the area in the centre of the heel that is made of alloy and the toe joint. The part outside the alloy heel is made of hard rubber to be able to absorb heel impacts. Assessments on the two different PoS are carried out later in this section.

From the figure some discrepancies are apparent between the calculated CoM, measured ZMP, and the physical construction of the foot:

- Neither CoM nor ZMP reaches the edge of the foot.
- The CoM's are generally determined to be closer to the centre of the foot, where the ZMP's are more symmetric located within the PoS of the feet.

Since the movements are very slow, it is known that the CoM must be at the edge of PoS when the fall is detected. Since the feet are symmetric about their centre line, the PoS must be

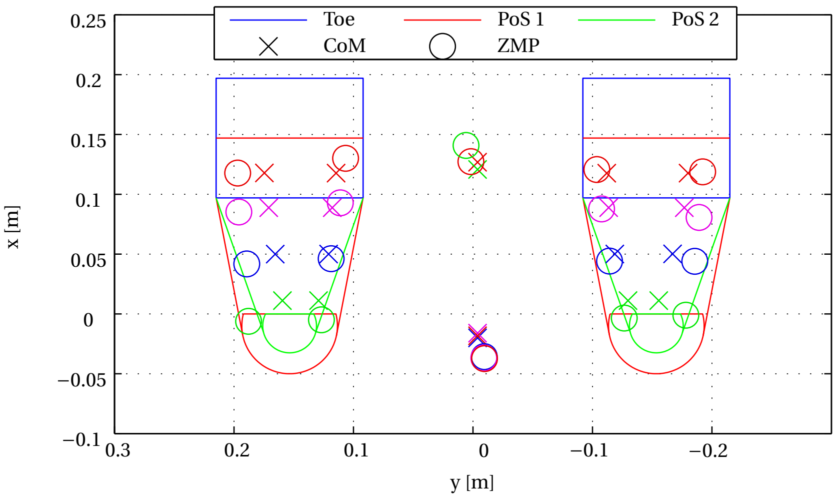


Figure 6.2. Top view of the PoS of the feet with computed CoM and measured ZMP just when the robot starts to fall over the foot.

symmetric about this line, hence the measured CoM should also be located symmetric about the centre line of the feet.

The posture of the robot is generated by only changing the angles of the ankles. The overall location of the detected CoM's is too close to the centre of the PoS. Ideally these should be detected exactly on the edge of the PoS or at least symmetrically about the centre line of the PoS. This indicates that the mass of the upper body of the robot is too small in the calculation of the CoM from the kinematics. More mass in the upper body in the calculation of the CoM would imply that the CoM is determined closer to the outside of the foot because the upper body is on the outer side of the foot (see figure 6.1). This observation agrees with the fact that the mass of the robot is 67 kg, while the masses used in the calculation of the CoM only sums to 58.2 kg hence 8.8 kg are missing in the calculation.

To obtain a more precise determination of the CoM from the forward kinematics, it is decided to include the additional mass in the calculation of the CoM. By consulting technical staff Simon Jensen and our supervisors it was concluded that the most likely cause of this discrepancy is the addition of cables, FTS amplifiers and the OBC mounted on the back of the robot. These items are likely to have been added after the original masses of the limbs were defined. From this it is chosen to add mass in the kinematics according to table 6.2.

Index	**Link name**	**Additional mass** [kg]	**Change in CoM position** [m]
5	Right Shin	0.5	
6	Right Thigh	0.5	
12	Left Shin	0.5	
13	Left Thigh	0.5	
15	Torso	4.8	$[-0.04\ 0.00\ 0.02]^T$
20	Pelvis	2.0	$[-0.01\ 0.00\ 0.00]^T$

Table 6.1. Corrections of the CoM positions and Masses for the limbs given in table A.4.

From the table it is seen that for the shins and thighs it is chosen only to add mass since it is assumed that the additional mass does not change the overall position of the CoM. For the pelvis, the mass is moved 1 cm back since the additional mass consists of cables that are placed on the rear of the robot. The position of the mass of the Torso is moved back and slightly up due to the position of the OBC and cables on the back.

With this new mass distribution and placement, the height of the CoM when all limbs of the robot is fully extended is 0.755 m where it was 0.71 m before the correction. A test to determine the position the CoM for the AAU-BOT1 is performed by laying the robot on the ground and lifting it in two support points. The force is measured in the two support points. From these measurements the height of the CoM on the real robot is determined to be 0.75 m, which complies with the theoretical CoM after the corrections.

In figure 6.3 the content from figure 6.2 is redrawn with the corrections given in table 6.2 implemented in the calculation of the CoM. From the figure it is seen that the CoM determined from the kinematics is more symmetric with respect to the PoS. With the knowledge that the measured CoM should be exactly symmetrical, the corrections seem to improve the quality of the model.

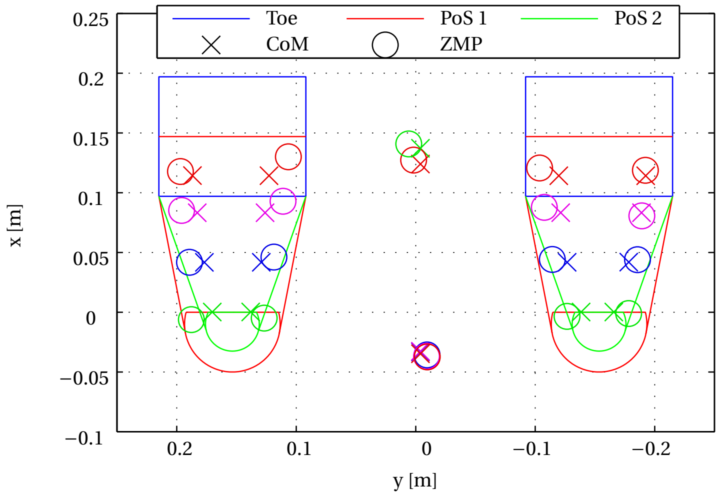


Figure 6.3. Top view of the PoS of the feet with computed CoM and measured ZMP just when the robot start to fall over the foot with updated calculation of the CoM.

Narrow Detection of Foot PoS

From figure 6.2 and 6.3 it is seen that both the ZMP and CoM at the time where the robot falls have some margin to the edge of the foot. Additionally it is seen that the ZMP measurement in general is further apart from the centre of the foot than the CoM is.

An initial hypothesis for this observation is that when the actual CoM of the robot leaves the part of the PoS given by the alloy in the heel (marked by green in the figures) and enters the rubber part (marked with red in the figure), the rubber begin to compresses. This implies that the robot will obtain an additional angle, and the actual CoM will move further towards the edge of the PoS. The robot will therefore fall over before the calculated CoM reaches the edge of the PoS.

To test if this hypothesis is true, the angle measured in the IMU is compared with the angle of the IMU calculated from the forward kinematics. In figure 6.4 the calculated and measured angle of the IMU is plotted for the test on the right foot towards the inside marked with blue in figure 6.3. The angles until time 50 s should not be considered since this is the time where the robot is put into the initial position with assistance from human hand. From time 50 s and onward the robot is balancing itself until time 97 s where it falls over the edge of the foot.

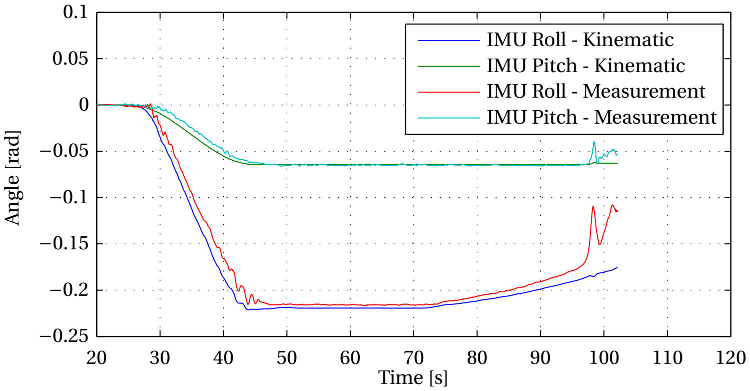


Figure 6.4. The angle of the IMU calculated through the kinematics and extracted directly from the IMU measurement. Is recorded that the robot falls at 97 s.

From the figure it is seen that the pitch angle is kept constant while the roll angle increases after time 70 s. Additionally it is seen that the measured angle increases more than the calculated one. This could indicate that the hypothesis is true to some degree. A closer study of the difference between the calculated and measured angle however shows that the difference between the measured angle and calculated angle shows up very early and seems to be a linear function of the angle. This does not comply with the hypothesis where the difference first should occur when the CoM enters the PoS given by the rubber part of the heel.

Even though the initial hypothesis is rejected a discrepancy between the measured and calculated IMU angle is identified. Since the discrepancy first occurs when the CoM leaves the centre of the foot and some torque about the ankle is present, it is considered that deformation in the FTS could be the reason. For the FTS to be able to measure strains, it needs a deform slightly. This deformation changes the angle of the robot with respect to the foot frame. From figure 6.4 the difference between the calculation and the measurement of the IMU angle is about 0.02 rad which is assumed to be a plausible deformation in the FTS.

Assuming that the only deformation causing the discrepancy is the one in the FTS, the calculation of the CoM could be corrected by adding the difference between the measured and the calculated IMU pitch and roll angle in the ankles. Doing so an IMU corrected version of the calculated CoM is found for the points where the robot falls. These points are shown in figure 6.5.

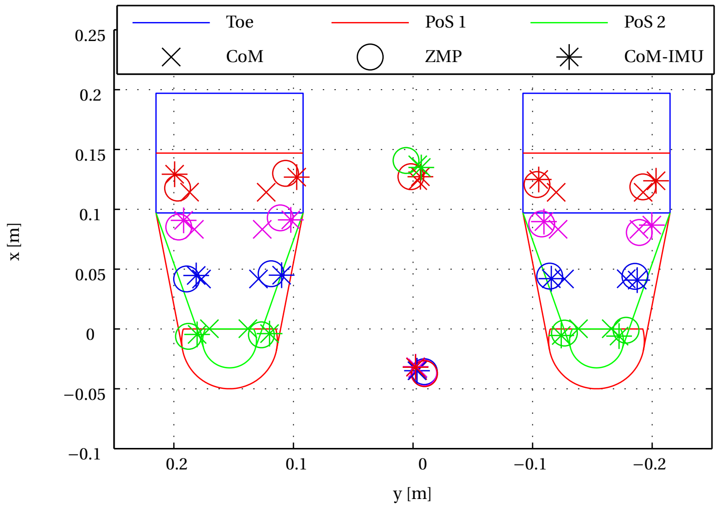


Figure 6.5. Top view of the PoS of the feet with computed CoM only from kinematics and a corrected version from the IMU when the robot starts falling. Additionally the measured ZMP is shown.

From the figure it is seen that the IMU corrected CoM is moved closer to the edge of the foot and coincides in general better with the ZMP measurement. From the CoM determined with the IMU correction the mean with and length of the PoS of the feet is found. The width is determined by measuring the distance between equal coloured pairs for one foot, and the mean of these widths for both feet are used as the width of the PoS at that colour. In table 6.2 the determined width and length of the PoS is shown. The mean width of the foot at the blue markers (72 mm) will be used in this thesis as the witdh of the feet because the CoM and ZMP will be placed approximately at a position in x-direction corresponding to the blue markers.

	Distance [mm]
Length	160
Width - Green	54
Width - Blue	72
Width - Purple	90
Width - Red	100

Table 6.2. Width and length of the PoS of the feet. The colours refer to the position in the x-direction according to the markers in figure 6.5.

Influence of Correction when not on Edge of PoS

In the corrections made to the kinematic model, only the points where the robot falls were analysed. It is however of interest to know how the changes influence the determination of the CoM when it is not on the edge of the foot. To do this the calculated CoM before and after correction is plotted alongside with the CoM corrected with the IMU measurement and the ZMP in figure 6.6. The plotted data is the same as used to generate figure 6.4.

From the figure it is seen that the CoM in general is moved further to the right on the

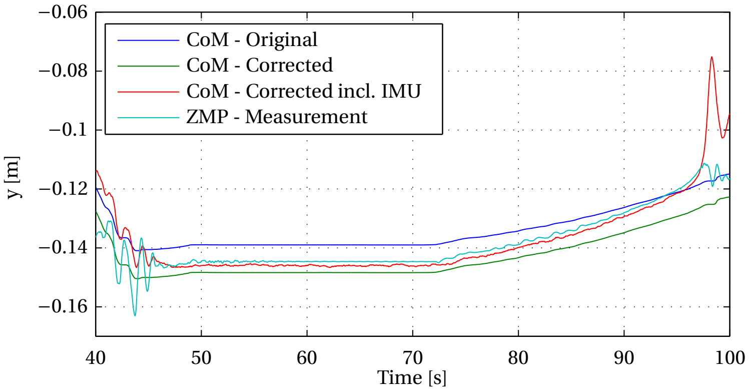


Figure 6.6. CoM calculated with the new mass distribution and corrected by the IMU measurement. For comparison the CoM calculated with the original mass distribution and the ZMP measurement is also shown.

right foot which was expected. It could seem like it is moved too far but with the correction from the IMU angle, the ZMP, and CoM are within 2 mm. Additionally it is seen that the shape of the development of the CoM is much alike the ZMP, this is a very important since the CoM and ZMP should coincide in the static case and when the robot is moving very slowly.

The results shown in figure 6.6 indicate that the corrections made to the model improve its quality. This leads to a better determination of the CoM from the kinematic model. Further introducing a correction based on the IMU measured angle seems to result in the best CoM determination. From figure 6.2 and 6.5 it is seen that the corrections makes the model identify the PoS as being more symmetric and closer to the actual known footprint.

6.3 Estimation of CoM and ZMP

In order to give the AAU-BOT1 an accurate perception of balance, it is essential to obtain an accurate determination of the position of the CoM and the ZMP. In order to achieve this a Kalman filter is used to combine readings from sensors with the models deduced in section 4.3. In order to make this filter run, data given from the different sensors needs to be combined and converted in various ways. This results in a system that can be seen in figure 6.7. In this figure each conversion is shown as a separate block with a specific input and output, connections to other function blocks are indicated by arrows. Notice that some of the arrows are coloured, this is done solely with the purpose of enhancing visibility of connections. The Kalman filter is implemented in the *CoM and ZMP estimator* function block. The inputs to this system are the sensor data given from the EPOS's and FTS's. The output is an estimated position of CoM and ZMP of the robot, together with a variable called support phase, which indicates if left, right, both, or no feet have ground contact. This is used to indicate current state of the robot and detect if the robot is hoisted off the ground. Next, the functionality of the function blocks shown in figure 6.7 will be introduced.

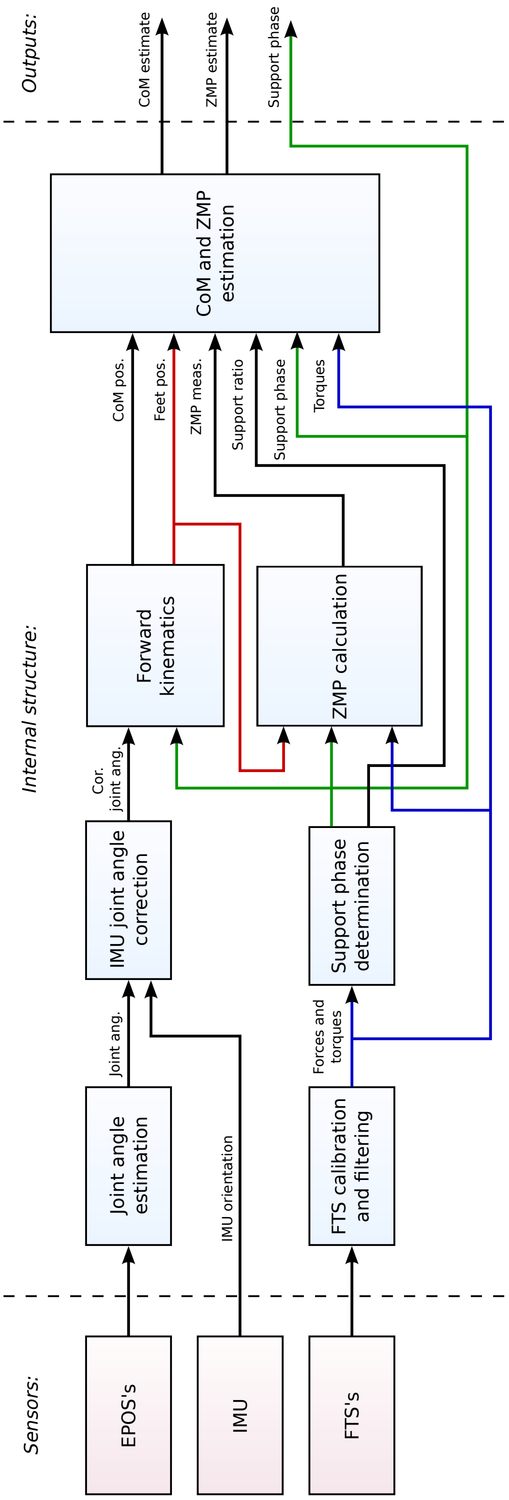


Figure 6.7. Internal structure of the CoM/ZMP estimator.

Joint angle estimation

In this block all the joint angles need to be determined from EPOS data. Since the state estimator used in the posture controller already estimates the joint angles, it is chosen to reuse the output of this block.

IMU Joint Ankle Correction

In section 6.2 it was identified that the FTS in the ankles of the AAU-BOT1 deforms as the load increases. This block uses the measured angles from the IMU to determine the deformation in the ankles. By assuming that all deformation occurs in the ankles, the angles measured from the IMU can via the measured joint angles and the kinematics be transformed such that the orientation of the shin with respect to the foot is measured. The idea is sketched in figure 6.8.

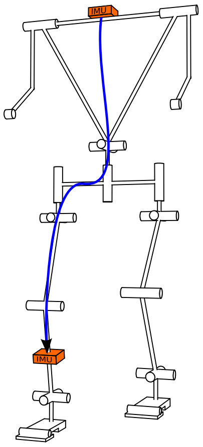


Figure 6.8. Sketch illustrating the idea of using the IMU to calculate the orientation of the shin with respect to the foot, and thereby roll and pitch angles of the ankles. The orientation of the IMU is converted to measure the orientation of the shin.

From the IMU the orientation output can be converted into a rotation matrix ${}^{Gs}_{S}\mathbf{R}$ describing the orientation of the sensor frame $\{S\}$ with respect to a global sensor reference frame $\{Gs\}$. Using the kinematics the orientation of the shin with respect to the foot ${}^{F}_{Sh}\mathbf{R}$ is found as:

$$ {}^{F}_{Sh}\mathbf{R} = {}^{F}_{G}\mathbf{R}\; {}^{G}_{Gs}\mathbf{R}\; {}^{Gs}_{S}\mathbf{R}\; {}^{S}_{Sh}\mathbf{R} $$

Where:

${}^{F}_{G}\mathbf{R}$ is the orientation of the global frame $\{G\}$ seen from the foot frame $\{F\}$

${}^{G}_{Gs}\mathbf{R}$ is a rotation matrix aligning the global sensor frame $\{Gs\}$ and the global frame $\{G\}$

${}^{Gs}_{S}\mathbf{R}$ is the sensor orientation measurement

${}^{S}_{Sh}\mathbf{R}$ is the orientation of the shin $\{Sh\}$ seen from the sensor frame $\{S\}$

The two rotations ${}^{F}_{G}\mathbf{R}$ and ${}^{S}_{Sh}\mathbf{R}$ are determined directly using the forward kinematics. The rotation ${}^{G}_{Gs}\mathbf{R}$ is inserted to align the global sensor frame $\{Gs\}$ and the global robot frame $\{G\}$. This rotation is needed because these two global frames can be rotated with respect to each other about their common z-axis. The robot is defined to have the frontal plane aligned with the yz-plane of the global frame at the initial position, while the global sensor frame has its x-axis pointing towards north.

From the kinematics it is known that seen from the foot to the shin the order of rotation is roll (x) and then pitch (y). The pitch θ_p and roll θ_r can therefore be determined:

$$\begin{aligned}
{}^{F}_{Sh}\mathbf{R} &= \mathbf{R}_\mathrm{x}\,\mathbf{R}_\mathrm{y} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_\mathrm{r} & -\sin\theta_\mathrm{r} \\ 0 & \sin\theta_\mathrm{r} & \cos\theta_\mathrm{r} \end{bmatrix} \begin{bmatrix} \cos\theta_\mathrm{p} & 0 & \sin\theta_\mathrm{p} \\ 0 & 1 & 0 \\ -\sin\theta_\mathrm{p} & 0 & \cos\theta_\mathrm{p} \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta_\mathrm{p} & 0 & \sin\theta_\mathrm{p} \\ \sin\theta_\mathrm{p}\sin\theta_\mathrm{r} & \cos\theta_\mathrm{r} & -\cos\theta_\mathrm{p}\sin\theta_\mathrm{r} \\ -\cos\theta_\mathrm{r}\sin\theta_\mathrm{p} & \sin\theta_\mathrm{r} & \cos\theta_\mathrm{p}\cos\theta_\mathrm{r} \end{bmatrix}
\end{aligned}$$

From this it is seen that pitch θ_p can be determined from the first row of the orientation matrix ${}^{F}_{Sh}\mathbf{R}$, and roll θ_r can be determined from the second row of the orientation matrix ${}^{F}_{Sh}\mathbf{R}$.

The roll and pitch angles determined in this way are then used instead of the roll and pitch angles of the ankles measured by the encoders in these joints.

FTS calibration and filtering

This block receives the raw FTS measurements from both FTS's and converts the readings into torques and moments in the two ankles. An illustration of the structure of this block is shown in figure 6.9.

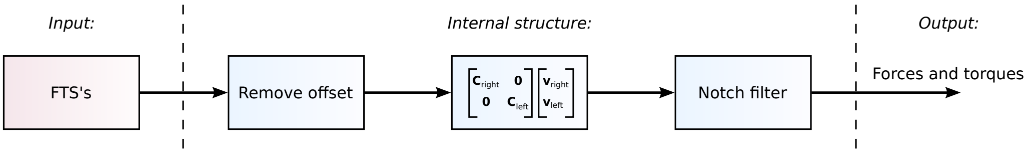

Input:
Internal structure:
Output:
FTS's
Remove offset
$\begin{bmatrix} \mathbf{C}_\mathrm{right} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_\mathrm{left} \end{bmatrix} \begin{bmatrix} \mathbf{v}_\mathrm{right} \\ \mathbf{v}_\mathrm{left} \end{bmatrix}$
Notch filter
Forces and torques

Figure 6.9. Internal structure of the *FTS calibration and filtering* block. $\mathbf{C}_\mathrm{right}$ and $\mathbf{C}_\mathrm{left}$ are the two calibration matrices for the right and left foot respectively. $\mathbf{v}_\mathrm{right}$ and $\mathbf{v}_\mathrm{left}$ are the sensor data from the right and left foot respectively.

The FTS readings are influenced by an offset without any load on the feet. To compensate for this raw data is sampled for two seconds while the AAU-BOT1 is hoisted off the ground and the mean value of each sensor is calculated and subtracted for every following sample. After the offset has been subtracted the calibration matrices $\mathbf{C}_\mathrm{right}$ and $\mathbf{C}_\mathrm{left}$ from appendix B are multiplied onto the measurement vectors $\mathbf{v}_\mathrm{right}$ and $\mathbf{v}_\mathrm{left}$ to give the forces and moments.

While performing various tests on the robot, readings from the FTS's generally showed an overlaying disturbance of a sinusoidal signal with constant frequency of approximately 2.25 Hz. This disturbance was initially thought to be caused by oscillations in one or more of the joint controllers. Additional tests however showed that the disturbance was caused by mechanical flexibility in the robot. Applying a force impulse on one foot while the AAU-BOT1 is hanging off the ground, clearly shows an oscillation propagating through the pelvis to the other leg, and both legs are left oscillating for several seconds.

In figure 6.10 an example of the measured force in the z-direction of the right foot is shown as the blue graph. The data was recorded during a test where both feet were placed firmly on the ground and the pelvis was moved from side to side to shift load between the two feet. From this measurement it is clear that the oscillations have an impact on the balance of the robot. Two approaches to solving this problem were discussed. Either the robot should try to counteract this actively, or it should ignore the oscillations.

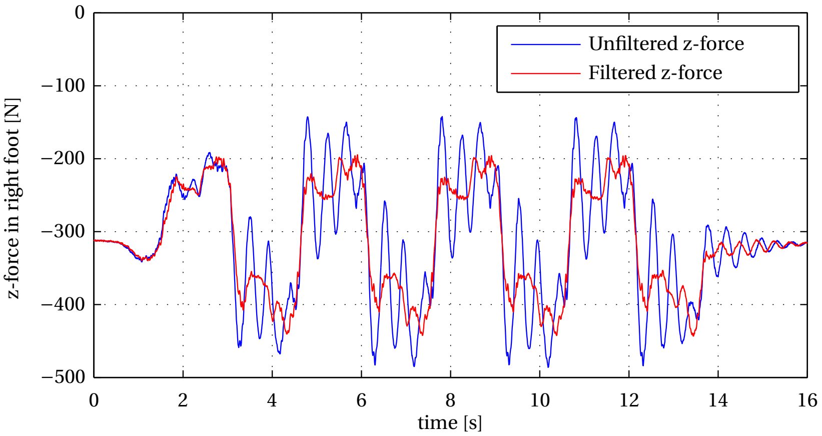


Figure 6.10. Unfiltered and notch filtered z-force readings.

In the end it was chosen to ignore the oscillations, since the disturbance appears to have zero mean and therefore assumed to have low impact on the balance of the robot. To ignore the oscillations it is chosen to filter out this frequency. Different combinations of notch and low pass filters have been tested, but in the end a single notch filter has shown to be a simple effective solution. A standard single notch filter with a transfer function as seen 6.1 is used [38].

$$H_{\text{notch}}(s) = \frac{s^2 + 2\frac{d}{c}\omega_0 s + \omega_0^2}{s^2 + 2\frac{1}{c}\omega_0 s + \omega_0^2} \tag{6.1}$$

where:

ω_0 is the notch frequency

d and c are tuning parameters that determines maximum damping and width of the damping region

In the design of the filter ω_0 is naturally selected as 2.25 Hz since this is the approximate frequency at which the disturbance influences. The tuning parameters d and c are designed through hand tuning by trying to achieve an effective damping of the disturbance without influencing the original measurement too much. In the end a $d = 0.01$ and $c = 3$ have been chosen. In figure 6.11 a magnitude plot of the zero order hold discretised notch filter is shown. With the selected filter parameters the maximum damping at 2.25 Hz is −39.5 dB and the −3 dB cutoff frequencies are placed at 1.6 Hz and 3 Hz. The result of the filtering can be seen in figure 6.10 as the red graph, where it is clearly seen that the influence of the oscillation has been damped significantly.

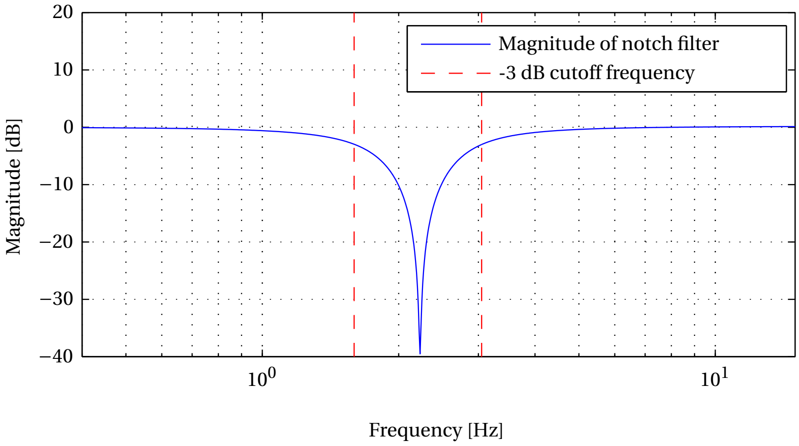


Figure 6.11. Magnitude plot of notch filter.

Support phase Determination

In the Linear Biped Model deduced in section 4.3.2 three different models were found. One for right, left, and double support. To switch between these models at the right time, the current support phase must be known. This information is extracted from the z-forces measured in the feet (Fz_{right}, Fz_{left}). First a *support ratio* is calculated for right foot (srr) and left foot(srl). This indicates how much of the total weight is placed on each foot. For the right foot the support ratio is calculated as:

$$srr = \frac{Fz_{\text{right}}}{Fz_{\text{right}} + Fz_{\text{left}}} \tag{6.2}$$

Using the support ratio for the right foot, the support phase is determined.

$$\text{Support phase } = \begin{cases} \text{0 (no)} & \textit{for} \quad Fz_{\text{right}} + Fz_{\text{left}} \geq -500\,\text{N} \\ \text{1 (right)} & \textit{for} \quad Fz_{\text{right}} + Fz_{\text{left}} < -500\,\text{N} \;\; \textit{and} \;\; srr \geq 0.99 \\ \text{2 (double)} & \textit{for} \quad Fz_{\text{right}} + Fz_{\text{left}} < -500\,\text{N} \;\; \textit{and} \;\; 0.03 \leq srr \leq 0.97 \\ \text{3 (left)} & \textit{for} \quad Fz_{\text{right}} + Fz_{\text{left}} < -500\,\text{N} \;\; \textit{and} \;\; srr \leq 0.01 \\ \text{prev. support} & \textit{otherwise} \end{cases}$$

The last condition specifies the support phase to be equal to the previous determined if none of the other conditions hold. Notice that the gab between the thresholds in the conditions for right, left, and double support phase. This introduces a hysteresis effect to eliminate registration of multiple switches during one transition from one support phase to another. The output named *support ratio* indicated with an arrow in figure 6.7 is simply defined as the *srr*.

Forward kinematics

This function block calculates the position of the CoM and the two feet in a global coordinate frame. The inputs to the forward kinematics are the estimated joint angles and current support phase. The positions of all the limbs and CoM are calculated directly as specified in section 4.2.1.

ZMP calculation

The purpose of this function block is to calculate the position of the ZMP. In single support an expression for the ZMP in the y-direction, can be deduced from figure 6.12 by calculating the moments about point ${}^{Fr}p_{yr}$ from the FTS readings and equating this to zero since no moments exists at the zero moment point.

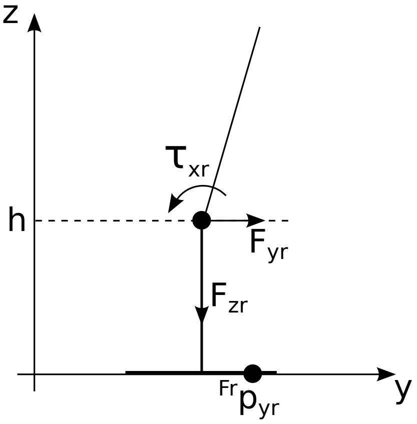


Figure 6.12. Sketch illustrating the relevant forces and torques in the calculation of the y-component of the ZMP for the right foot. The forces and torques are measured by the FTS placed in height h above the ground.

$$0 = \tau_{xr} - hF_{yr} + F_{zr}(0 - {}^{Fr}p_{yr})$$

$$\Downarrow$$

$${}^{Fr}p_{yr} = \frac{\tau_{xr}}{F_{zr}} - \frac{hF_{yr}}{F_{zr}} \tag{6.3}$$

Where:

${}^{Fr}p_{yr}$ is the y-component of the ZMP in the right foot FTS frame {Fr}

τ_{xr} is the measured torque about the x-axis in the right foot FTS

F_{yr}, F_{zr} is the measured force in the y and z direction in the right foot FTS

$h = 0.04$ is the height of the right foot FTS over ground level

In a similar way the x-component of the ZMP in the right foot frame can be calculated:

$${}^{Fr}p_{xr} = -\frac{\tau_{yr}}{F_{zr}} - \frac{hF_{xr}}{F_{zr}} \tag{6.4}$$

Knowing that the z-component is zero, the ZMP in the right foot frame is ${}^{Fr}\mathbf{p}_r = [{}^{Fr}p_{xr}\ {}^{Fr}p_{yr}\ 0]^T$. Using the knowledge of the feet position and orientation through the kinematics the ZMP for the right foot is then converted to the global frame: $\mathbf{p}_r = \mathbf{T}({}^{Fr}\mathbf{p}_r)$, where $\mathbf{T}(\cdot)$ is a transformation given by the forward kinematics transforming points in the right foot FTS frame to the global frame.

In the same way the ZMP is calculated for the left foot. In double support the ZMP is found by combining the two identified ZMP's. From figure 6.13 the total ZMP is calculated by summing all moments about the x and y-axis acting at point $\mathbf{p}$ on the ground surface $(z = 0)$ and equating to zero.

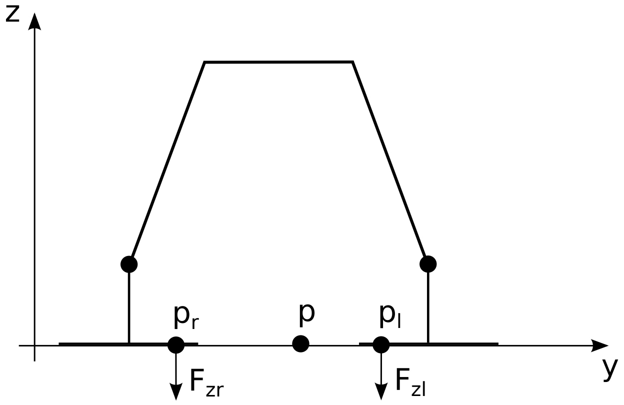


Figure 6.13. Sketch illustrating the relevant forces and torques in the calculation total ZMP for the robot from the identified ZMP's from the two feet.

$$\mathbf{0} = (\mathbf{p}_r - \mathbf{p})F_{zr} + (\mathbf{p}_l - \mathbf{p})F_{zl} \qquad \text{[Nm]}$$

$$\Downarrow$$

$$\mathbf{p} = \frac{\mathbf{p}_r F_{zr} + \mathbf{p}_l F_{zl}}{F_{zr} + F_{zl}} \qquad \text{[m]} \quad (6.5)$$

In the implemented version of equation (6.3), (6.4), and the corresponding for the left foot the last term is omitted. From an experiment the last term is determined to be up to 6 mm. This maximum value is however obtained in the case where almost no force is acting on the foot in the z-direction hence the error would not have much influence on the global ZMP determined by equation (6.5).

CoM and ZMP estimation

Finally with all the inputs to the *CoM and ZMP estimation* block generated, the estimator can be designed. To merge these different inputs into a CoM and ZMP estimate, a Kalman filter is utilised. The models needed are the once deduced in section 4.3. The CoM and ZMP estimator is split into two parallel filters. One estimates the x-component of the CoM and ZMP, while the other estimates the y-component.

The dynamic model of the CoM acceleration in the y-axis switches between three different models namely; (4.38), (4.39), and (4.40). To handle this model switching it is chosen to implement a Kalman filter which estimates all the states of the three different models simultaneously. To model the correspondence between the CoM and ZMP the cart-table model given in equation (4.48) is used. This results in a system with 15 states, 5 for each model. The following system matrices will assume the following structure of the state space system:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$$
$$\mathbf{y} = \mathbf{Cx}$$

The three models have the same 5 states namely: $[c_y\ \dot{c}_\mathrm{y}\ \ddot{c}_\mathrm{y}\ \dddot{c}_\mathrm{y}\ p_\mathrm{y}]^T$. c_y denotes the position of the CoM, while p_y denotes the position of the ZMP. When the robot is in right single support the inputs to the model are $\mathbf{u}_\mathrm{right} = [P_{yr}\ \tau_{xr}]^T$. P_{yr} is the position of the right foot, while τ_{xr} is the torque about the x-axis measured in the right foot. For left single support $\mathbf{u}_\mathrm{left} = [P_{yl}\ \tau_{xl}]^T$ and for double support $\mathbf{u}_\mathrm{double} = [P_{yr}\ P_{yl}\ \tau_{xr}\ \tau_{xl}]^T$. The continuous state space model for right support looks as follows:

$$\mathbf{A}_\mathrm{right} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{g}{z_c} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\frac{1.79 z_c}{g} & 0 \end{bmatrix} \qquad \mathbf{B}_\mathrm{right} = \begin{bmatrix} 0 & 0 \\ -\frac{g}{z_c} & -\frac{1}{m z_c} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

For the double support case the state space model looks as follows:

$$\mathbf{A}_\mathrm{double} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{g}{z_c}\frac{1-\gamma}{\gamma} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\frac{1.79 z_c}{g} & 0 \end{bmatrix} \qquad \mathbf{B}_\mathrm{double} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{g}{2 z_c}\frac{1-\gamma}{\gamma} & \frac{g}{2 z_c}\frac{1-\gamma}{\gamma} & -\frac{\omega_R}{m z_c} & -\frac{\omega_L}{m z_c} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

And finally the state space model for left single support:

$$\mathbf{A}_\mathrm{left} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{g}{z_c} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\frac{1.79 z_c}{g} & 0 \end{bmatrix} \qquad \mathbf{B}_\mathrm{left} = \begin{bmatrix} 0 & 0 \\ -\frac{g}{z_c} & -\frac{1}{m z_c} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The output model of the three support phases is exactly the same namely:

$$\mathbf{C}_\mathrm{right} = \mathbf{C}_\mathrm{double} = \mathbf{C}_\mathrm{left} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The three models are combined into one big state space system:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{A}_\mathrm{right} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_\mathrm{double} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_\mathrm{left} \end{bmatrix}_{15\times15} \begin{bmatrix} \mathbf{x}_\mathrm{right} \\ \mathbf{x}_\mathrm{double} \\ \mathbf{x}_\mathrm{left} \end{bmatrix}_{15\times1} + \begin{bmatrix} \mathbf{B}_\mathrm{right} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_\mathrm{double} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_\mathrm{left} \end{bmatrix}_{15\times8} \begin{bmatrix} \mathbf{u}_\mathrm{right} \\ \mathbf{u}_\mathrm{double} \\ \mathbf{u}_\mathrm{left} \end{bmatrix}_{8\times1}$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{C}_{\text{right}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\text{double}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_{\text{left}} \end{bmatrix}_{6\times 15} \begin{bmatrix} \mathbf{x}_{\text{right}} \\ \mathbf{x}_{\text{double}} \\ \mathbf{x}_{\text{left}} \end{bmatrix}_{15\times 1}$$

Constants:

$g = 9.82\,\text{m/s}^2$ is the gravitational constant

$m = 67\,\text{kg}$ is the total mass of the robot

$z_c = 0.72\,\text{m}$ is the approximate height of CoM over ground

$\gamma = 0.98$ is the double support ratio introduced in Linear Biped Model

This combined system is then Euler discretised and implemented in the same way as the state estimator described in section 5.5. All initial states are defined as zero and initial covariance is defined as a $I_{15\times 15}$ identity matrix. The model variances are chosen the same for the three models and are as follows: $\mathbf{Q} = \text{diag}([10 \;\; 50 \;\; 10 \;\; 10^5 \;\; 10^{-3}])$. From the various sensors the measurement vector $\mathbf{z} = [c_y \; p_y]^T$ is weighted with the following variances; $\mathbf{R} = \text{diag}([0.1 \; 10])$.

The structure of the other filter estimating the x-direction is exactly the same. Only the dynamic models have small differences. When the CoM and ZMP has been estimated, the support phase input is used to choose which of the outputs that is considered the most accurate and should be used.

CHAPTER 7

GAIT-BALANCE CONTROLLER

This chapter introduces the concepts used to generate motions for the AAU-BOT1 that enables it to obtain dynamic gait and maintain balance. Section 7.1 gives an overview of the gait-balance controller and shows how the individual parts are connected. The first part of creating gait is to generate ZMP and CoM trajectories. These topics will be treated in section 7.2 and 7.3. The ZMP and CoM references need to be combined with foot lifts and converted into joint angles. This work is described in section 7.4. When a foot impacts the ground a high transient force is experienced. In section 7.5 the precautions taken to soften impact and make sure that the robot lands with its foot parallel to the ground is described. A CoM controller is described in section 7.6 which is used to maintain balance of the robot during gait. During this project ZMP control will not be implemented, but some initial considerations are discussed in section 7.7.

Chapter Content

7.1 Introduction to Gait-balance Controller

The following section will introduce the overall structure of the gait-balance controller. Different ways of generating gait were presented in section 3.2. The approach referred to as ZMP control in section 3.2 has been chosen since it is a relatively flexible approach that can be implemented in various ways. A possible implementation is to generate ZMP, and from this the CoM, offline and use these trajectories to form the basic motions of gait. The same approach can also be implemented with online generation of the ZMP and CoM to be able to change the gait in runtime. This opens for the possibility to create more advanced balance controllers that changes the placement of the feet based on measurements.

An initial attempt was made where the human recorded gait was replayed on the robot. This did however not show any promising result and this approach was quickly abandoned. In this project it has been chosen first to generate a ZMP trajectory directly from the gait specification, and then deduce a CoM that makes the robot obtain the desired ZMP reference. Foot movement and balance control is then build on top of this to make the AAU-BOT1 walk while remaining in balance. A block diagram of the different functionalities in the gait-balance controller can be seen in figure 7.1.

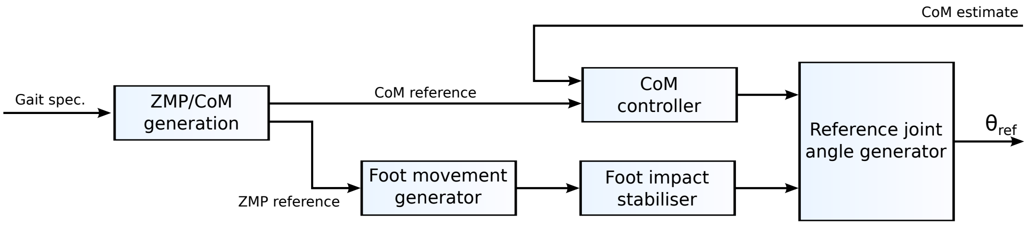


Figure 7.1. Controller structure of the gait-balance controller.

The ZMP and CoM trajectories are generated from a specification of the gait to create the overall motions of the robot. This is done offline before the execution of the gait in the *ZMP/CoM generation* block. The output of this block is a CoM trajectory that the robot must follow. It has been chosen only to implement a balance controller which utilises position of CoM. It is assumed that by controlling the CoM close enough to its reference, the specified ZMP will be achieved. This is implemented in the *CoM controller* which also receives the estimated CoM from the Kalman filter.

To achieve gait trajectories must also be specified for the feet. This task is handled by the *Foot movement generator*. The timing, height, and length of each step is determined in this block from the ZMP trajectory. It has been chosen to generate the foot trajectories offline as well.

To help stabilise the robot during gait, shock absorption is implemented in the legs of the robot. This reduces the influence of the forces experienced when the robot impacts the ground with a foot. The shock absorption is performed online by changing the foot height of the robot in the *Foot impact stabiliser*. Finally the *Reference joint angle generator* receives information about desired CoM and foot placement and converts this into reference joint angles θ_{ref} that the posture controller must follow.

In the following sections more detailed information about how each functionality is implemented will be given.

7.2 ZMP Reference Generation

From the movements recorded in the motion tracking room, important information about human gait has been obtained. The study of pelvis and torso movement gave some insight in the movement of the CoM. In order to determine the exact movement of the CoM during gait, detailed information about distribution of mass of the test subjects is needed. To gain additional information about CoM movement additional information is found in literature concerning human gait [13].

Movement of the ZMP can not be extracted from the measurements made in the motion tracking room. To determine the ZMP, recordings of forces in the feet of the test subjects would be required. To gain knowledge about this, external literature is consulted [13, 39].

The movement of the CoM in the frontal plane shifts between right and left foot in an almost sinusoidal way. The frequency and amplitude of this motion is determined by the step time and width. For long step times the CoM is placed closer to the centre of the supporting foot. In the sagittal axis the CoM moves with a more constant speed, with a mean determined by the gait velocity.

To keep balance, the ZMP of a human during gait, stays within the PoS at all times. As soon as the heel of the swinging foot impacts the ground and the double support phase begins, the ZMP is begun shifted from the previously supporting foot to the new supporting foot. This movement happens in an almost straight line from the toes of the previously supporting foot to the heel of the new supporting foot. In the end of the double support phase the ZMP has been moved under the new supporting foot and the previously supporting foot is lifted from the ground. During single support phase the ZMP is moved from the heel of the supporting foot towards the toes.

For this project it is chosen to define the ZMP trajectory separately for the transverse and sagittal axis. Instead of moving the ZMP from the heel towards the toe as humans do, it is chosen to keep the ZMP centred under the foot in single support.

The stride time of the robot needs to be defined before the ZMP reference can generated. From the cart-table model it can be seen that if a fast stride time is chosen the CoM does not have to move as much, since the accelerations become greater. On the other hand a fast stride time means that the joints of the robot have to move faster. Limitations exist in both maximum movement of CoM and joint velocity. A stride time of 3.5 s has proven a good compromise between these two constraints. Comparing this to the human recorded data, this approximately corresponds to a gait velocity of 0.14 m/s (0.5 km/h). The actual gait velocity of the robot can be adjusted with the step length.

The ZMP trajectory in the frontal plane is furthermore defined by the step width. From the recorded human data this distance varies from 81 mm to 222 mm. The further apart the feet are placed from each other, the more stability is gained in sideways motion. A large step width however also requires the CoM to be moved further away from centre stance, which again causes problems with joint angle limitations. Due to the placement of the spur gears in the ankles, the step width can not be less than 240 mm. To make the gait as close to human as possible the minimum step width is chosen. The resulting ZMP trajectory in the frontal plane can be seen in figure 7.2. To create a smooth transition from one foot to the other, a fifth order polynomial is fitted in the double support phase. The endpoints of the fifth order polynomial are defined such that the desired position, zero velocity, and zero acceleration is obtained.

From section 2.1 it is known that for human gait, the duration of the double support phase correspond to approximately 20 % of the stride time. In section 3.5 it is additionally shown from the recordings of the human gait that for low gait velocities the duration of the double support increases. The duration of the double support cannot directly be specified through the ZMP since both feet can be placed on the ground even though the ZMP is beneath one foot. It however puts a lower limit on the duration of double support phase since both feet must be on the ground when the ZMP is between the inside of the feet. It is chosen to use 80 % of the stride time to move the ZMP between the centres of the feet (40 % for each transition). This causes the ZMP to be between the feet approximately 40 % of the stride time hence the duration of the double support phase must be at least 40 %. Gait with shorter double support phase have been tested but resulted in more wobbly gait. Choosing the double support even longer makes the single support phases shorter and thereby less time to swing the feet. By using 80 % of the stride time to move the ZMP between the feet a good compromise is obtained.

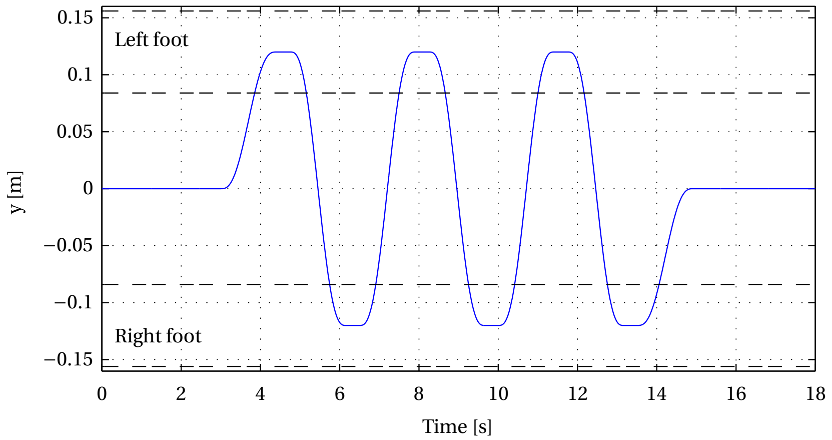


Figure 7.2. ZMP reference trajectory for movements in the transverse axis for gait with step width of 240 mm and stride time of 3.5 s. The dashed lines indicate the width of the feet.

To define the ZMP reference in the sagittal axis, a step length must be chosen. From the selected stride time of 3.5 s the step length can be calculated as a function of the gait velocity. As an example the ZMP reference is shown in figure 7.3 for a gait velocity of 0.1 m/s. This results in a step length of 174 mm.

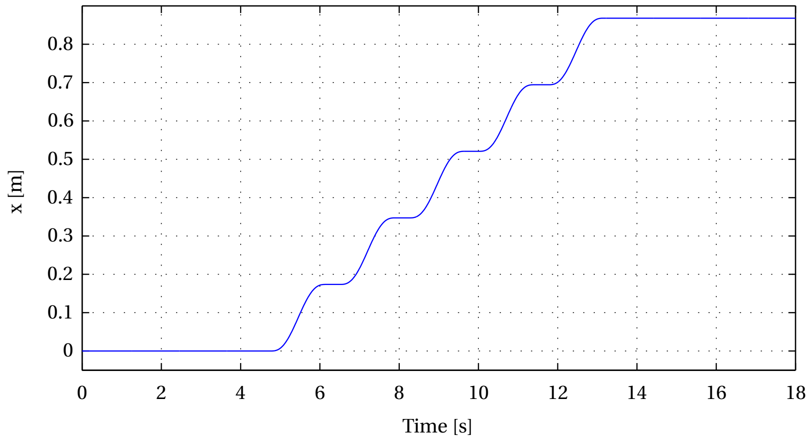


Figure 7.3. ZMP reference trajectory for movements in the sagittal axis for gait with step length of 174 mm and stride time of 3.5 s. In double support the ZMP is moved from the rear foot to the front foot identified by the change in position on the x-axis.

A plot of the ZMP reference in an xy-plot for a gait velocity of 0.1 m/s is shown in figure 7.4. The outline of the feet are plotted to show stability margin.

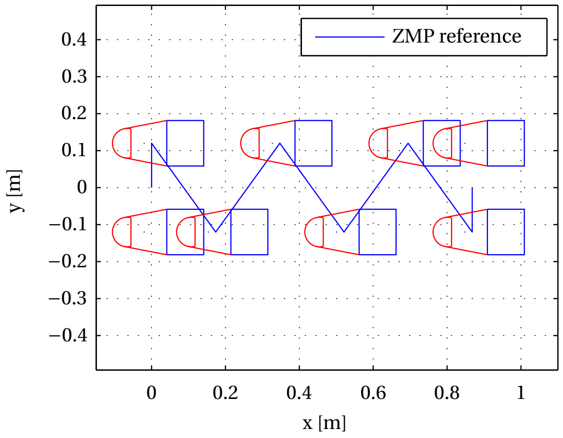


Figure 7.4. ZMP reference trajectory for movements in the transverse plane for gait with step width of 240 mm, step length of 174 mm, and stride time of 3.5 s.

7.3 CoM Reference Generation

To obtain gait with a biped robot, different ways of moving the CoM have been proposed by other research groups. Two approaches that both have been successfully implemented and tested on other biped robots are considered. One approach described by [19] is to use feedback control with a ZMP as reference and CoM as input to the system to determine the CoM that minimises ZMP tracking error. The cart-table model is used to describe the connection between ZMP and CoM movement.

Another approach described in [40], is to use the natural motion of an inverted pendulum to generate the CoM. The motion of the robot in single support is assumed to be that of an inverted pendulum with pivot about the supporting foot and no moments about the pivot point. By solving the second order homogeneous differential equations (4.32) and (4.33) from the 2D-LIPM with chosen initial conditions, the ballistic motion of the CoM during the swing can be calculated. To generate the CoM in the transition between the two single support cases the authors of [40] proposes a linear transition of the CoM between the two feet with constant velocity.

The end result of the two proposed approaches is very similar. An advantage of the first solution is that the conversion from a reference ZMP trajectory to CoM trajectory can be performed in almost the same way independent of the ZMP trajectory. In the inverted pendulum conversion, choices have to be taken concerning initial and end conditions dependent on the desired ZMP trajectory, but this method is less computational heavy.

In this project the first solution is chosen and the following part concerns designing the controller that minimises the ZMP tracking error. In the control problem the cart-table model, described in section 4.3.3, is used to describe the movements of the ZMP. In this section only movement in transverse axis will be discussed, but the same method is used for the movement in the sagittal axis. Using equation (4.44) and assuming that the movements of the CoM can be controlled by specifying the time derivative of the horizontal acceleration ($u_y = \frac{d}{dt}\ddot{c}_y$) the discretised dynamic model (Using zero-order hold and sampling time Ts) for the ZMP is:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \tag{7.1}$$

$$y(k) = \mathbf{C}\mathbf{x}(k) \tag{7.2}$$

Where:

$$\mathbf{A} = \begin{bmatrix} 1 & T_s & T_s^2/2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} T_s^3/6 & T_s^2/2 & T_s \end{bmatrix}^T$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & -c_z/g \end{bmatrix}$$

$$\mathbf{x}(k) = \begin{bmatrix} c_y(kT_s) & \dot{c}_y(kT_s) & \ddot{c}_y(kT_s) \end{bmatrix}^T$$

$$u(k) = u_y(kT_s)$$

$$y(k) = p_y(kT_s)$$

The structure of the controller is shown in figure 7.5 where the reference is the desired ZMP reference and the output of the system is the obtained ZMP and the corresponding CoM trajectory.

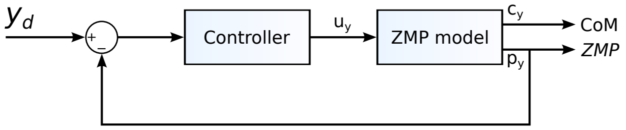


Figure 7.5. Controller structure for the ZMP reference tracking.

To obtain zero steady state tracking error integral action is included by introducing an integral state $x_i(k)$ that accumulates the error between the system output $y(k)$ and the desired output $y_d(k)$: $x_i(k+1) = x_i(k) + y(k) - y_d(k) = x_i(k) + Cx(k) - y_d(k)$. Including the integral state in an augmented state vector $\tilde{\mathbf{x}}(k) = [x_i(k)\ \mathbf{x}(k)^T]^T$ the system dynamics is identified:

$$\tilde{\mathbf{x}}(k+1) = \begin{bmatrix} 1 & C \\ 0 & A \end{bmatrix} \tilde{\mathbf{x}}(k) + \begin{bmatrix} 0 \\ B \end{bmatrix} u(k) + \begin{bmatrix} -1 \\ 0 \end{bmatrix} y_d(k) \tag{7.3}$$

By using the `dlqr` function in MATLAB® the feedback gain $\mathbf{K}$ to generate the control signal $u(k) = -\mathbf{K}\tilde{\mathbf{x}}(k)$ is found by minimising the cost function:

$$J = \sum_{k=0}^{\infty} \left(\tilde{\mathbf{x}}(k)^T \mathbf{Q} \tilde{\mathbf{x}}(k) + u(k)^T R u(k) \right) \tag{7.4}$$

Where:

$$\mathbf{Q} = \mathrm{diag}(1,0,0,0)$$
$$R = 10^{-4}$$

Using the derived gains in a simulation with the model given in equation (7.3) and the ZMP reference shown in figure 7.2 results in a CoM and ZMP as shown in figure 7.6

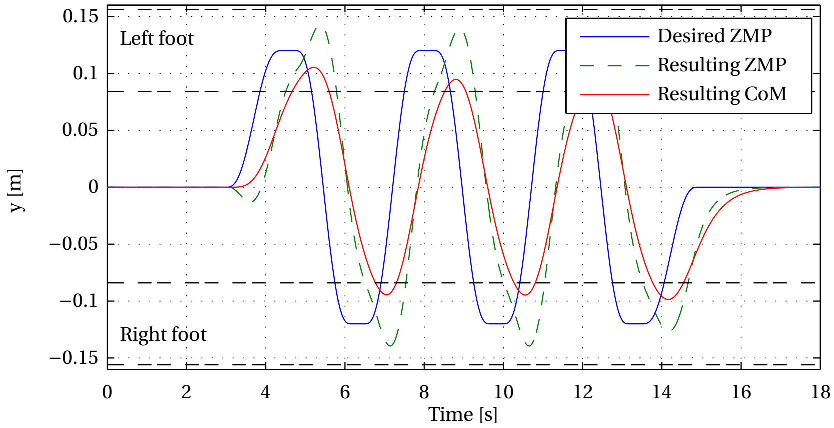


Figure 7.6. Obtained ZMP and CoM with state feedback controller that includes integral control. The black dashed lines indicate the width of the feet.

From the figure it is seen that the controller is capable of reducing the tracking error in steady state, but in general is the controller not capable of generating a CoM that creates the desired ZMP. A problem with the poor reference tracking is that the ZMP does not reache the support area of one foot before the other foot eventually is lifted from the ground. This is seen just after 3.5 s where the reference ZMP is in the centre of the left foot, and the right foot could be lifted if the reference was tracked, but the actual ZMP is closer to the right foot than to the left.

The obtained result shown in figure 7.6 is not capable generating a reference for the CoM that results in the desired ZMP. It should be noted that the weights $\mathbf{Q}$ and R used in the optimisation problem have been modified to obtain better performance, but no better result can be obtained even though the cost of using control signal is reduced and the penalty of having tracking error is increased.

One way to obtain better tracking of the ZMP is to use information about how the ZMP reference changes in the future. This information is obviously available since the knowledge of where the feet must be placed at the next step must be available. This information of how the ZMP changes in the future can be included in the control problem in several ways. In Model Predictive Control (MPC) the system model is used in an optimisation problem where the control signal, that minimizes the tracking error between the model output and the reference, is identified by solving the optimisation problem at each sample interval. Another approach is to incorporate the knowledge about how the reference changes in the system model and solving the problem using LQR. The latter way will be referred to as preview control.

In biped projects the preview control approach has already been used for generating CoM trajectories [19]. Initial tests with the preview control method showed good results hence this method will be used to generate the CoM trajectory in this project.

Introducing Preview Control

The idea of introducing preview control to obtain a better ZMP reference tracking is inspired from [19]. The way preview control is implemented is inspired from [41].

For the reference it is assumed that it is known to each time step N_L time steps ahead $y_d(k+1), \ldots, y_d(k+N_l)$, this is referred to as the preview horizon. Secondly it is assumed that the reference beyond sample $k+N_L$ can be approximated by the last sample in the preview horizon $y_d(k+N_L)$:

$$y_d(k+i) = y_d(k+N_L) \quad \text{for } i = N_l+1, \ldots \tag{7.5}$$

First the system given in equation (7.1) is written into incremental states $\Delta\mathbf{x}(k) = \mathbf{x}(k) - \mathbf{x}(k-1)$ using an incremental control input $\Delta u(k) = u(k) - u(k-1)$:

$$\Delta\mathbf{x}(k+1) = \mathbf{A}\Delta\mathbf{x}(k) + \mathbf{B}\Delta u(k) \tag{7.6}$$

By specifying the incremental demand as $\Delta y_d(k) = y_d(k) - y_d(k-1)$, and the error state as $e(k) = y(k) - y_d(k)$, the dynamics for the error can be derived:

$$\begin{aligned}
e(k+1) &= y(k+1) - y_d(k+1) \\
&= \mathbf{CA}\mathbf{x}(k) + \mathbf{CB}u(k) - y_d(k+1) \\
&= \mathbf{CA}(\mathbf{x}(k-1) + \Delta\mathbf{x}(k)) + \mathbf{CB}(u(k-1) + \Delta u(k)) - (y_d(k) + \Delta y_d(k+1)) \\
&= \mathbf{CA}\mathbf{x}(k-1) + \mathbf{CB}u(k-1) + \mathbf{CA}\Delta\mathbf{x}(k) + \mathbf{CB}\Delta u(k) - y_d(k) - \Delta y_d(k+1) \\
&= e(k) + \mathbf{CA}\Delta\mathbf{x}(k) + \mathbf{CB}\Delta u(k) - \Delta y_d(k+1)
\end{aligned}$$

The dynamics for the augmented system with states $e(k)$ and $\Delta\mathbf{x}(k)$ can now be written as:

$$\begin{bmatrix} e(k+1) \\ \Delta\mathbf{x}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{CA} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \begin{bmatrix} e(k) \\ \Delta\mathbf{x}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{CB} \\ \mathbf{B} \end{bmatrix} \Delta u(k) + \begin{bmatrix} -1 \\ 0 \end{bmatrix} \Delta y_d(k+1) \tag{7.7}$$

Since the demand is known N_L samples ahead, the incremental demands can be collected into a $N_L \times 1$ vector:

$$\mathbf{x}_d(k) = [\Delta y_d(k+1) \ldots \Delta y_d(k+N_L)]^T \tag{7.8}$$

The assumption given in equation (7.5) that the demand is constant beyond N_L time steps ahead, implies that the incremental demand is zero when the preview horizon is exceeded:

$$\Delta y_d(k+i) = 0 \quad \text{for } i = N_L+1, \ldots \tag{7.9}$$

This implies that the dynamics for the demand state $\mathbf{x}_d(k)$ can be written as:

$$\mathbf{x}_d(k+1) = \mathbf{A}_d \mathbf{x}_d(k) \tag{7.10}$$

where:

$$\mathbf{A}_d = \begin{bmatrix} 0 & 1 & & 0 \\ & 0 & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{bmatrix}_{N_L \times N_L}$$

The augmented state vector $\bar{\mathbf{x}}(k) = [e(k)\, \Delta\mathbf{x}(k)^T\, \mathbf{x}_d(k)^T]^T$ is now used. The dynamics for the states $\bar{\mathbf{x}}(k)$ is now identified from equations (7.7) and (7.10)

$$\bar{\mathbf{x}}(k+1) = \left[\begin{array}{cc|cc} 1 & \mathbf{CA} & -1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \hline \multicolumn{2}{c|}{\mathbf{0}} & \multicolumn{2}{c}{\mathbf{A}_d} \end{array}\right] \bar{\mathbf{x}}(k) + \left[\begin{array}{c} \mathbf{CB} \\ \mathbf{B} \\ \hline \mathbf{0} \end{array}\right] \Delta u(k) \tag{7.11}$$

For this system we can write a quadratic cost function

$$J = \sum_{k=0}^{\infty} \left(\bar{\mathbf{x}}(k)^T \mathbf{Q} \bar{\mathbf{x}}(k) + \Delta u(k)^T R \Delta u(k) \right) \tag{7.12}$$

Where:

$$\mathbf{Q} = \mathrm{diag}(Q_e, \mathbf{Q}_x, \mathbf{0})$$

Q_e is the cost associated with the tracing error $e(k)$, $\mathbf{Q}_x$ is the cost associated with the incremental states $\Delta\mathbf{x}(k)$. From the equation it is seen that no cost is associated with the demand. Putting a cost on these states would not make sense since these cannot be reached from the system input. Solving the optimization problem using `dlqr` in MATLAB® the state feedback gains $\mathbf{K}$ to generate the incremental control $\Delta u(k) = -\mathbf{K}\bar{\mathbf{x}}$ is found. The gains in $\mathbf{K}$ can be split into a part regarding the tracking error $K_e \in \mathbb{R}$, a part regarding the original incremental states $\mathbf{K}_x \in \mathbb{R}^3$, and finally a part regarding the future incremental demands

$\mathbf{K}_d \in \mathbb{R}^{N_L}$. These are identified as: $\mathbf{K} = [K_e\,\mathbf{K}_x\,\mathbf{K}_d]$. Introducing these gains the incremental control can be written as:

$$\begin{aligned}\Delta u(k) &= -K_e e(k) - \mathbf{K}_x \Delta \mathbf{x}(k) - \mathbf{K}_d \mathbf{x}_d(k)\\ &= -K_e e(k) - \mathbf{K}_x \Delta \mathbf{x}(k) - \sum_{l=1}^{N_L} \mathbf{K}_d(l) \Delta y_d(k+l)\end{aligned}$$

Assuming that $y(k) = 0, y_d(k) = 0, \mathbf{x}(k) = \mathbf{0}$ for $k = 0, -1, \ldots$ the optimal controller for the original system given in equation (7.1) and (7.2) can be identified as:

$$u(k) = -K_e \sum_{i=0}^{k} e(i) - \mathbf{K}_x \mathbf{x}(k) - \sum_{l=1}^{N_L} \mathbf{K}_d(l) y_d(k+l) \tag{7.13}$$

The optimal control problem is solved with parameters $Q_e = 1$, $\mathbf{Q}_x = \text{diag}(0,0,0)$, $R = 10^{-4}$, and $N_L = 2/Ts$, and the resulting gains are used to control the original system to follow the ZMP reference shown in figure 7.2. This results in a CoM and ZMP as shown in figure 7.7. It is chosen not to put any cost on the incremental states because the main objective is to make the ZMP follow the reference very close no matter how the CoM moves.

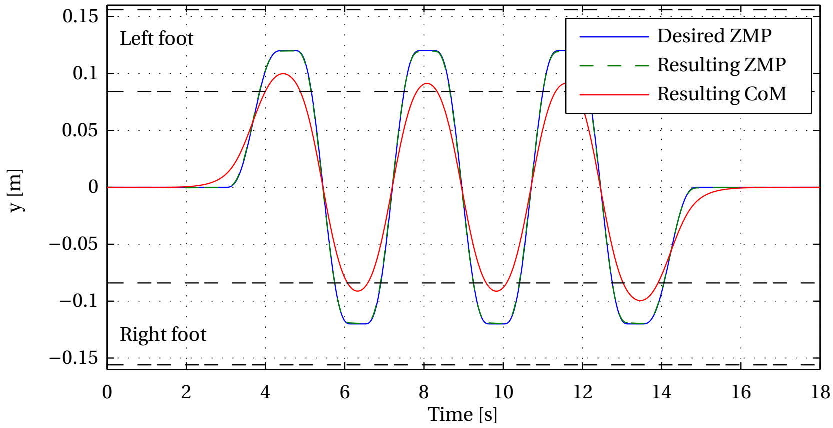


Figure 7.7. Obtained ZMP and CoM using preview control with parameters $Q_e = 1$, $\mathbf{Q}_x = \text{diag}(0,0,0)$, $R = 10^{-4}$, and $N_L = 2/Ts$. The black dashed lines indicate the width of the feet.

From the figure it is seen that the controller is capable of controlling the ZMP to follow the desired reference. It is also clear from the figure that the effect of the preview control makes the CoM move before the ZMP reference moves. Comparing figure 7.6 and figure 7.7 it is clear to identify the positive effect of including preview control. Increasing the cost associated with the tracking error Q_e or decreasing the cost associated with changing control input R does not affect the result much. On the other hand changing the preview horizon N_L influences the results greatly. Decreasing the horizon some of the phenomenon observed with the controller without preview action, i.e. the resulting ZMP moves in the opposite direction of the reference when the reference changes rapidly. Increasing the horizon makes the tracking error even smaller. A preview horizon of 2 s is assessed as a good compromise between tracking and computational burden.

In figure 7.8 and 7.9 the resulting CoM trajectories are shown for the x-direction and in the xy-plane respectively. From the figures it is seen that the ZMP trajectory is tracked and a CoM reference is generated.

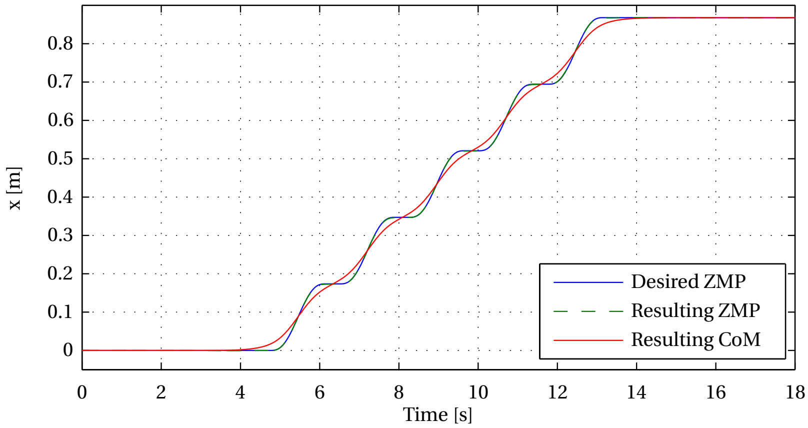


Figure 7.8. Obtained ZMP and CoM using preview control in the sagittal axis with parameters $Q_e = 1$, $\mathbf{Q}_x = \text{diag}(0,0,0)$, $R = 10^{-4}$, and $N_L = 2/Ts$.

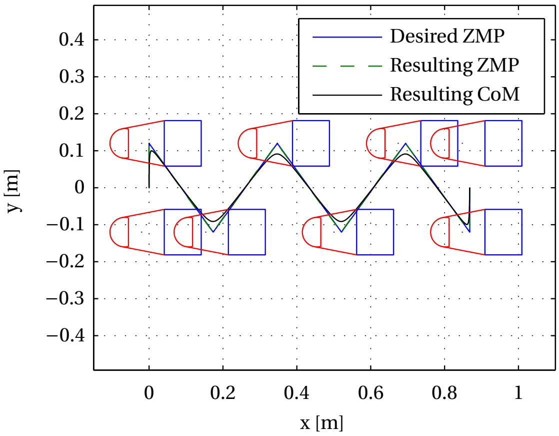


Figure 7.9. Obtained ZMP and CoM using preview control in the transverse plane.

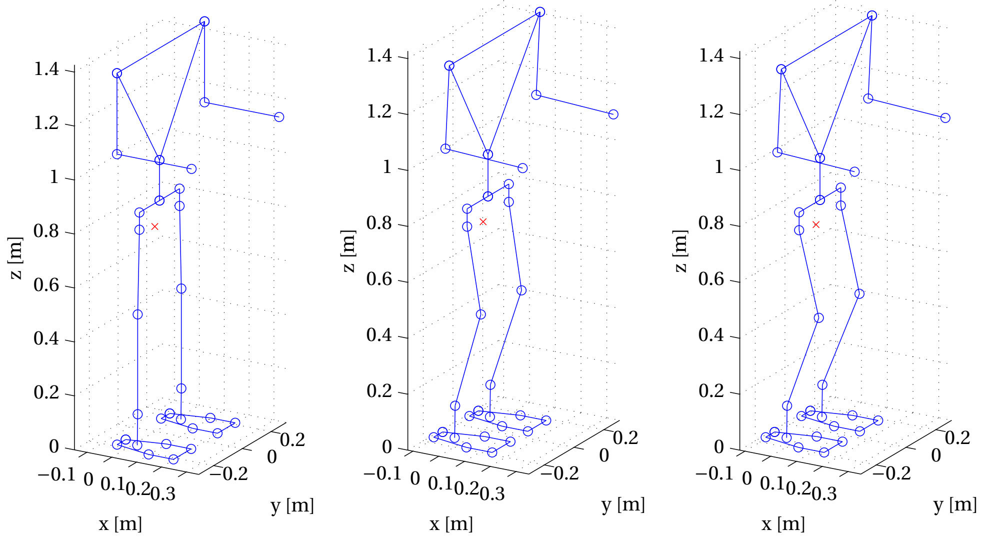
7.4 Joint Trajectory Generation

From the ZMP and CoM trajectories defined in the two previous sections joint angle references must be generated such that these trajectories are tracked. In addition to ZMP and CoM trajectories the desired way of moving the feet must be taken into consideration.

The main idea in generating the joint angles is to define an initial posture where the position of the feet with respect to the pelvis is defined. To obtain the joint angles during gait, this initial posture is modified by adding the feet trajectory to the initial position. By using the inverse kinematics described in section 4.2.2 the joint angles are found.

Initial Posture

In this section the choice of initial posture is discussed. In figure 7.10 three initial positions are shown. In figure 7.10(a) a posture is shown where all limbs are fully extended i.e. all joint angles are zero. This posture is not a feasible initial posture since the feet are separated in the transverse axis by 30.7 cm where the desired stance width is 24 cm. Additionally the CoM is located close to the rear of the foot such that the robot is almost falling backwards.



(a) Straight posture

(b) 97.5% pelvis height

(c) 96% pelvis height

Figure 7.10. Three different initial postures. The red cross is the CoM of the entire robot.

In figure 7.10(b) and 7.10(c) two feasible initial postures are shown. These postures are generated by specifying the feet position in the transverse axis to be ±12 cm from the pelvis centre. Additionally both feet are moved −6 cm in the sagittal axis compared to the straight posture to move the pelvis, and thereby CoM, closer to the toe to obtain better stability. To move the CoM even further forward in the initial posture, the torso is pitched 0.05 rad resulting in a 4 mm CoM displacement.

The only way the two feasible initial postures differ is in the height of the pelvis i.e. how much the legs are bend. For the two shown feasible initial postures the height of the pelvis is 97.5 % and 96 % of the pelvis height obtained in the straight posture. The choice of pelvis height is a compromise between the amplitude of the pelvis movement during gait and the margin to the separate joint angle limitations. An initial posture where the pelvis height is close to the pelvis height in the straight posture is desired since the joint angles are far away from their limitations. This however limits the extent that the pelvis can be moved in the transverse plane without changing the pelvis height.

For the straight posture the pelvis can not be moved since the legs are already fully stretched. Reducing the pelvis height to 97.5 % the pelvis can be moved 14 cm in the transverse axis, 11 cm forward, or 23 cm backward before one of the legs is fully extended. The difference in possible forward and backward movement of the pelvis is caused by the forward movement of the pelvis in the initial posture. Reducing the pelvis height to 96 % the pelvis can be moved 18 cm in the transverse axis, 15 cm forward, or 27 cm backward before one of the legs is fully extended. The forward movement can however not be obtained since the pitch angle limitations is almost violated in the initial position. For the posture with 97.5 % pelvis height, no pitch angles are violated.

To obtain the necessary pelvis movement and have some margin to the joint limitations, the pelvis height is chosen to be 97.5 % of the pelvis height in the straight posture. The initial posture is described by the transformation from the heel frames to the pelvis frame:

$$\mathbf{TRi} = \begin{bmatrix} 1 & 0 & 0 & -0.124 \\ 0 & 1 & 0 & -0.120 \\ 0 & 0 & 1 & -0.836 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{TLi} = \begin{bmatrix} 1 & 0 & 0 & -0.124 \\ 0 & 1 & 0 & 0.120 \\ 0 & 0 & 1 & -0.836 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.14}$$

Where:

TRi is the initial right heel frame seen from pelvis frame

TLi is the initial left heel frame seen from pelvis frame

In figure 7.11 the initial position of the feet are shown in the global frame. From the figure it is seen that the feet are placed symmetrically with respect to the sagittal axis indicating that the pelvis centre is placed on the sagittal axis in the initial position. In the figure two CoM's are shown indicated by crosses. The red cross is the obtained CoM when the pelvis is moved forward and the torso is pitched with respect to the straight posture. The blue cross indicates the position of the CoM if only the pelvis height and stance width were reduced. From the two crosses it is seen that the resulting CoM is moved closer to the toe as desired.

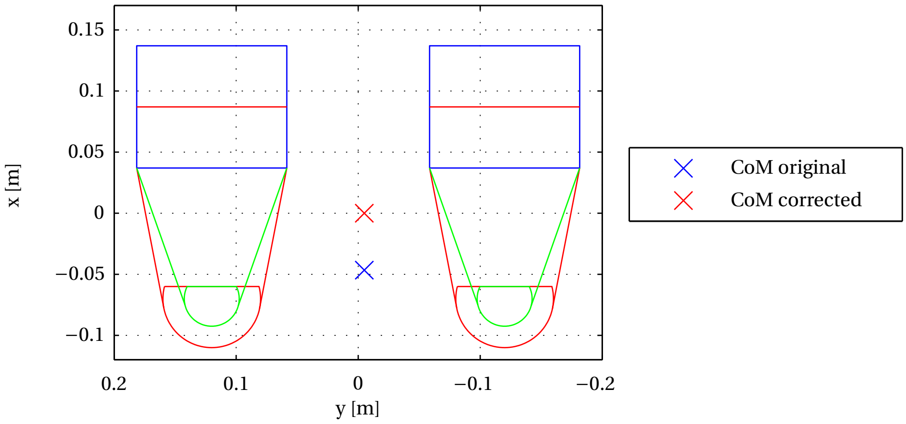


Figure 7.11. Initial position of the feet in the global frame. CoM position with respect to the feet is indicated with crosses. The CoM indicated with a red cross indicates the final CoM obtained with the chosen initial posture. The CoM indicated with a blue cross is obtained with the posture where only pelvis height and stance width is reduced.

In the figure it is seen that the final CoM in the initial position is located on the transverse axis but slightly moved towards the right foot. This displacement is caused by unsymmetrical distribution of the mass in the robot in the transverse axis.

Feet Movements

From the ZMP trajectory, the trajectories for foot lift and foot movements can be defined since the position of the ZMP tells when the feet can be lifted and where they should be moved to. In figure 7.12 the foot height trajectories are shown for both feet alongside with the ZMP trajectory.

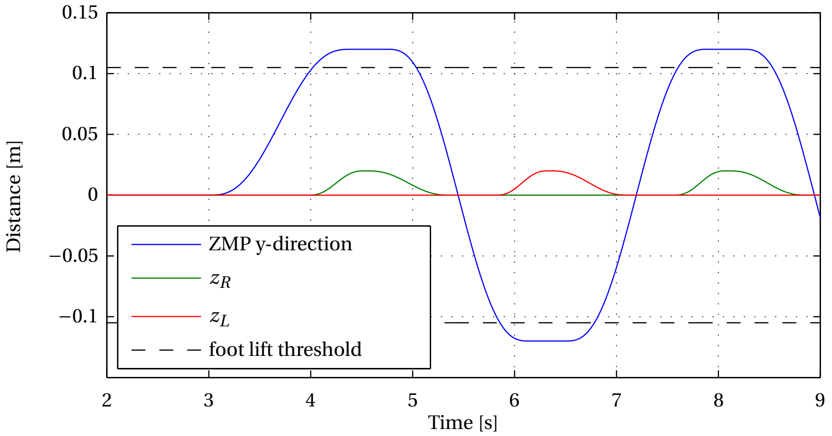


Figure 7.12. Foot height trajectories plotted alongside with the ZMP in the transverse axis.

The trajectories are designed such that the lift is initiated when the ZMP trajectory is within 15 mm from its maximum value. This threshold is indicated with the black dashed line in the figure. The foot lift is ended when the ZMP trajectory crosses zero. The late ending of the foot lift is chosen to be sure that the robot lands and stays on the foot when it impacts the ground and not just bounces off the ground again. For a single lift the foot height trajectory consists of a lift and a lowering phase. These two parts of the foot lift trajectory are generated from fifth order polynomials to be able to specify the position, velocity, and acceleration in both endpoints of the transition. From the figure it is clear that the end and start position of the transition is either zero or the maximum foot height. To make the transition as smooth as possible in the ends, the velocities and accelerations are chosen to be zero.

In general the timing for a complete lift is chosen such that the lift takes 40 % of the time and the lowering takes 60 % of the time. The lowering is made longer than the lift since this makes a softer impact. From figure 7.12 it is seen that the feet are lifted 20 mm above ground level. This is chosen as a compromise between reducing unnecessary movements and having enough height to reduce the risk of striking the ground during swing.

Having defined the foot height trajectories, the trajectories for the foot movement in the sagittal axis is defined. The position of the feet in the global frame are shown in figure 7.13. The trajectories are defined such that the position of the supporting foot complies with the position of the ZMP.

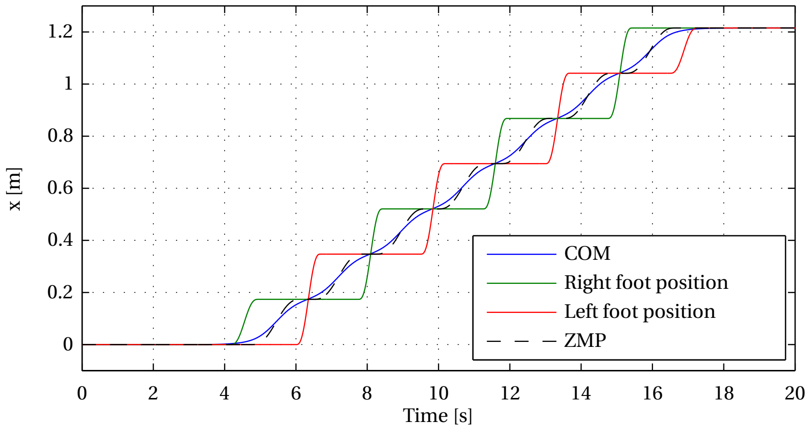


Figure 7.13. Foot position in the sagittal axis in the global frame.

The transition between two positions of a foot is chosen to be a fifth order polynomial, where the desired position is specified for the endpoints and the velocity and acceleration for the endpoints are chosen to be zero. This transition was chosen since the change from standstill to movement become smooth as desired. During the analysis of the recording of the human gait described in section 3.5 it was observed that this is actually very close to the way the humans in our experiment moved their feet. The timing in the sagittal foot movement is designed such that the transition occurs in the period when the foot is lifted more than 10 mm from the ground.

To be able to obtain the desired foot movements the position of the feet must be known in the pelvis frame. Since the position of the pelvis is not specified for gait, it is assumed that the pelvis position and the CoM coincide in the sagittal axis. The position of the feet in the pelvis frame can then be calculated as the difference between the CoM reference and the desired position of the feet in the global frame. The resulting position of the feet in the pelvis frame is shown in figure 7.14. The assumption that the pelvis position and CoM in the sagittal axis coincide may not be accurate hence the CoM reference may not be tracked correctly. Using this assumption, the correct foot position in the global frame can however be obtained. CoM reference tracking can be obtained by moving the pelvis with respect to both feet in the sagittal axis. This approach is described later in this section.

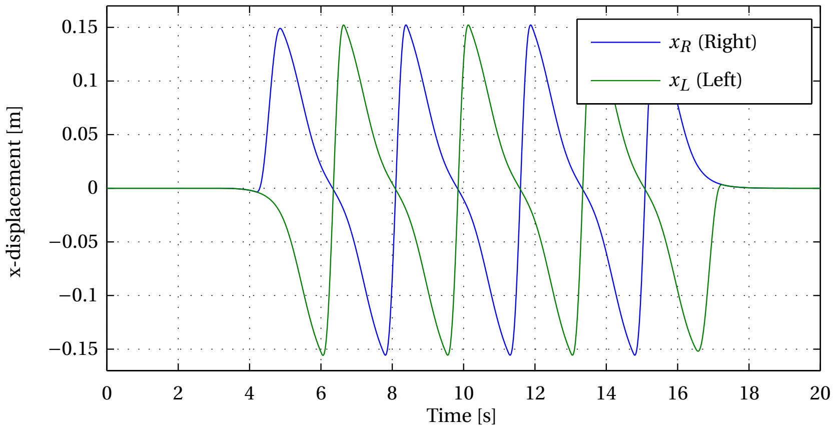


Figure 7.14. Foot position in the sagittal axis in the pelvis frame.

From the initial right foot position **TRi** defined in (7.14) and the derived foot height trajectories, the right foot frame as a function of time can be derived:

$$\begin{aligned}
\mathbf{TR}(t) &= \mathbf{TRm}(t)\cdot\mathbf{TRi}\\
&= \begin{bmatrix} 1 & 0 & 0 & x_R(t)\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & z_R(t)\\ 0 & 0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix} 1 & 0 & 0 & -0.124\\ 0 & 1 & 0 & -0.120\\ 0 & 0 & 1 & -0.836\\ 0 & 0 & 0 & 1\end{bmatrix}\\
&= \begin{bmatrix} 1 & 0 & 0 & -0.124+x_R(t)\\ 0 & 1 & 0 & -0.120\\ 0 & 0 & 1 & -0.836+z_R(t)\\ 0 & 0 & 0 & 1\end{bmatrix}
\end{aligned} \tag{7.15}$$

Where:

$x_R(t)$ is the displacement in the sagittal axis of the right foot shown in figure 7.14

$z_R(t)$ is the vertical displacement of the right foot shown in figure 7.12

In (7.15) the transformation containing the foot movements **TRm** is pre-multiplied to the initial foot frame **TRi** since the movements of the feet are defined in the pelvis frame. In the same way the transformation describing the left foot frame is derived:

$$\mathbf{TL}(t) = \begin{bmatrix} 1 & 0 & 0 & -0.124+x_L(t)\\ 0 & 1 & 0 & 0.120\\ 0 & 0 & 1 & -0.836+z_L(t)\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{7.16}$$

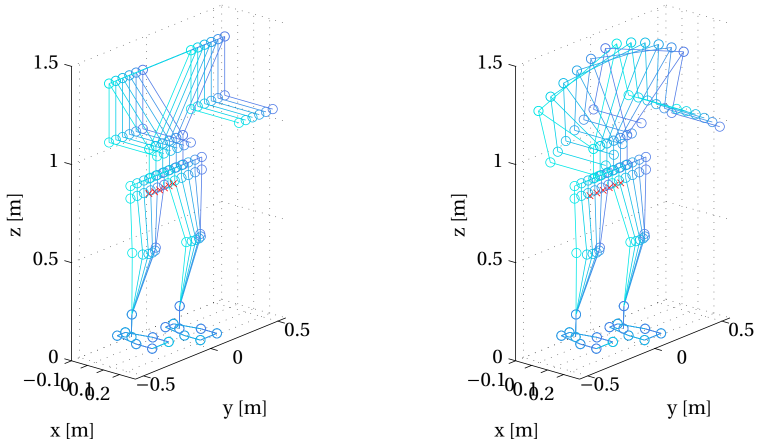
Where:

$x_L(t)$ is the displacement in the sagittal axis of the left foot shown in figure 7.14

$z_L(t)$ is the vertical displacement of the left foot shown in figure 7.12

CoM tracking

To obtain the desired CoM, and thereby ZMP, it is chosen to move the pelvis in the transverse plane in both the sagittal and the transverse axis. To obtain the desired CoM in the transverse axis it is additionally chosen to use the torso, by rolling it, if the desired CoM movement cannot be obtained only using the pelvis. The two ways of obtaining the CoM in the transverse axis is sketched in figure 7.15 where the range from maximum negative to maximum positive pelvis, and thereby CoM, displacement is shown as six separate postures. The idea of using torso to move the CoM further is inspired from the analysis of the human gait carried out in section 3.5 where it was observed that the torso movement was used by some of the subjects and to a greater extend at lower gait velocities.



(a) Only pelvis.

(b) Pelvis and torso.

Figure 7.15. CoM movements in transverse axis.

Through a simulation the relation between pelvis movement, torso roll, and CoM displacement is identified. The result is shown in figure 7.16. From the figure it is seen that by moving the pelvis the maximum distance the CoM can be moved approximately 110 mm from the initial position. By rolling the torso 0.3 rad the CoM is moved 24 mm from the initial position. The torso could be rolled up to 0.38 rad it is however chosen to limit this to 0.3 rad to have margin to the joint angle limit.

From figure 7.16 the following relationship is derived:

$$c_\mathrm{y}(q_\mathrm{y},\theta_\mathrm{r}) = 0.725\,q_\mathrm{y} + 0.078\,\theta_\mathrm{r} - 0.005 \qquad [\mathrm{m}] \quad (7.17)$$

Where:

c_y is position of CoM in the transverse axis [m]

q_y is position of pelvis in the transverse axis [m]

θ_r is the roll angle of the torso [rad]

From equation (7.17) it is clear that a desired CoM can be obtained using different combinations of pelvis movement and torso roll. To obtain the desired CoM it is chosen to

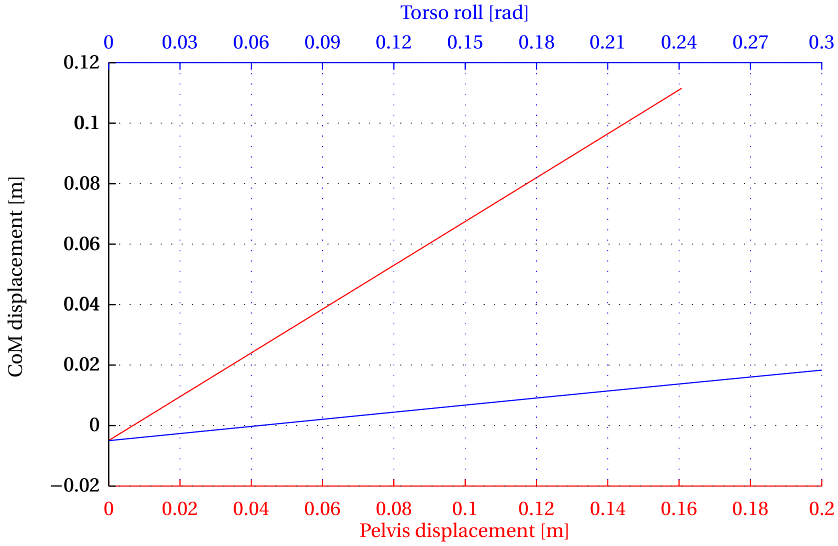


Figure 7.16. Relationship between pelvis and torso movements and the CoM movements in transverse axis.

add a constraint relating the pelvis and torso movements:

$$\frac{q_{\text{y}}}{q_{\text{y-max}}} = \frac{\theta_{\text{r}}}{\theta_{\text{r-max}}} \qquad [-] \quad (7.18)$$

Where:

$q_{\text{y-max}}$ is the maximum desired movement of the pelvis [m]

$\theta_{\text{r-max}}$ is the maximum desired roll of the torso [rad]

Inserting this constraint in equation (7.17) by isolating θ_{r} a closed form solution can be found for the pelvis movement and torso roll:

$$\theta_{\text{r}}(q_{\text{y}}) = q_{\text{y}}\frac{\theta_{\text{r-max}}}{q_{\text{y-max}}} \qquad [\text{rad}] \quad (7.19)$$

$$q_{\text{y}}(c_{\text{y}}) = \frac{c_{\text{y}} + 0.005}{0.725 + 0.078\,\frac{\theta_{\text{r-max}}}{q_{\text{y-max}}}} \qquad [\text{m}] \quad (7.20)$$

The parameters $\theta_{\text{r-max}}$ and $q_{\text{y-max}}$ are tuning parameters used to determine how much the torso is used. By fixing $q_{\text{y-max}}$ to the actual maximum pelvis displacement $\theta_{\text{r-max}}$ can be used to define the roll that is obtained when the maximum pelvis displacement is obtained. Setting $\theta_{\text{r-max}} = 0\,\text{rad}$ the behaviour shown in figure 7.15(a) is obtained while specifying $\theta_{\text{r-max}} = 0.3\,\text{rad}$ the behaviour shown in figure 7.15(b) is obtained. When the reference CoM has larger amplitude than the obtainable CoM displacement by pure pelvis displacement $\theta_{\text{r-max}}$ must be chosen greater than zero otherwise it can be set to zero to avoid the usage of torso roll.

For the CoM movement in the sagittal axis only the pelvis translation is considered. In figure 7.17 the maximum negative to maximum positive pelvis displacement is shown as six separate postures.

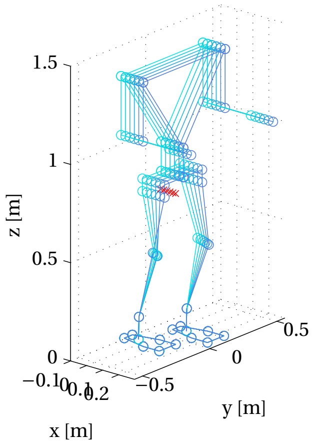


Figure 7.17. CoM movements in the sagittal axis.

In the same way as for the transverse movements the relationship between the pelvis movement in the sagittal axis and the change in CoM is identified:

$$\Delta c_\mathrm{x}(\Delta q_\mathrm{x}) = 0.741\ \Delta q_\mathrm{x} \qquad [\mathrm{m}] \quad (7.21)$$

Where:

Δc_x is a change in CoM position in the sagittal axis with respect to the feet [m]

Δq_x is a change in pelvis position in the sagittal axis with respect to the feet [m]

Since the CoM is only changed by pelvis movement, the expression for the pelvis movement as a function of desired CoM movement is identified directly from equation (7.21):

$$\Delta q_\mathrm{x}(\Delta c_\mathrm{x}) = \frac{1}{0.741}\ \Delta c_\mathrm{x} \qquad [\mathrm{m}] \quad (7.22)$$

Comparing equation (7.17) and equation (7.21) it is seen that for the movements in the sagittal axis a change in position is used rather than fixed positions. This is caused by the fact that the pelvis movement is defined with respect to the feet, and as robot moves along the sagittal axis the position of the feet is changed in sagittal axis while they are fixed in the transverse axis.

Having defined the conversion from CoM movement to pelvis movement the left and right foot frames defined in equation (7.15) and (7.16) can be updated to include pelvis movement using the fact that moving the pelvis in one direction is equivalent to moving both feet in the opposite direction. For the right foot frame equation (7.15) can be extended to:

$$
\begin{aligned}
\mathbf{TR'}(t) &= \mathbf{TRc}(t) \cdot \mathbf{TR}(t) \\
&= \begin{bmatrix} 1 & 0 & 0 & -\Delta q_\text{x}(\Delta c_\text{x}) \\ 0 & 1 & 0 & -q_\text{y}(c_\text{y}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -0.124 + x_R(t) \\ 0 & 1 & 0 & -0.120 \\ 0 & 0 & 1 & -0.836 + z_R(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & -0.124 + x_R(t) - \frac{1}{0.741}\,\Delta c_\text{x} \\ 0 & 1 & 0 & -0.120 - \frac{c_\text{y}+0.005}{0.725+0.078\,\frac{\theta_\text{r-max}}{q_\text{y-max}}} \\ 0 & 0 & 1 & -0.836 + z_R(t) \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{7.23}
$$

In the same way the left foot frame given in equation (7.16) is updated to:

$$
\mathbf{TL'}(t) = \begin{bmatrix} 1 & 0 & 0 & -0.124 + x_R(t) - \frac{1}{0.741}\,\Delta c_\text{x} \\ 0 & 1 & 0 & 0.120 - \frac{c_\text{y}+0.005}{0.725+0.078\,\frac{\theta_\text{r-max}}{q_\text{y-max}}} \\ 0 & 0 & 1 & -0.836 + z_R(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.24}
$$

In the foot frames shown in equation (7.23) and (7.24) everything but Δc_x is known directly from the reference CoM and ZMP trajectories and from the desired initial posture. The needed changed CoM position Δc_x is determined by calculating the difference between the CoM reference in the sagittal axis and the CoM obtained in the sagittal axis. This can be calculated in real time and the difference can be reduced using a controller. Knowing the trajectory before hand, the gait can be simulated by setting $\Delta c_\text{x} = 0$ and the actual Δc_x can be determined by comparing the simulated CoM to the reference.

From the defined foot frames in (7.23) and (7.24) the joint angle trajectories for the legs are calculated using the inverse kinematics described in section 4.2.2. The torso roll joint angle reference calculated in equation (7.19) is directly used as joint angle reference.

Simulating a Gait

Using the above mentioned approach the joint trajectories are calculated for the gait with the ZMP and CoM trajectories specified in section 7.2 and 7.3 without using the torso roll ($\theta_\text{r-max} = 0\,\text{rad}$) and without any CoM correction in the sagittal axis ($\Delta c_\text{x} = 0\,\text{m}$). From the calculated joint trajectories the forward kinematics are used to calculate the position of the resulting CoM in the global frame. The obtained CoM is shown in figure 7.18.

From this figure it is seen that with the derived joint angles the CoM of the robot follows the reference. From the figure the precision is however not visible. In figure 7.19 the tracking error is shown.

From this figure it is seen that the CoM in the sagittal axis fluctuates around the reference and diverges from the reference more than 20 mm during gait. For the CoM in

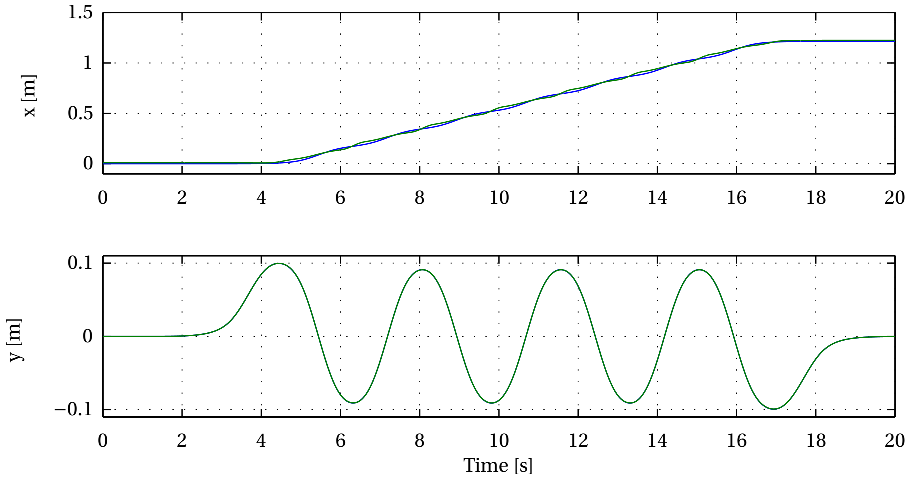


Figure 7.18. CoM obtained in simulation. Blue is reference and green is simulated CoM.

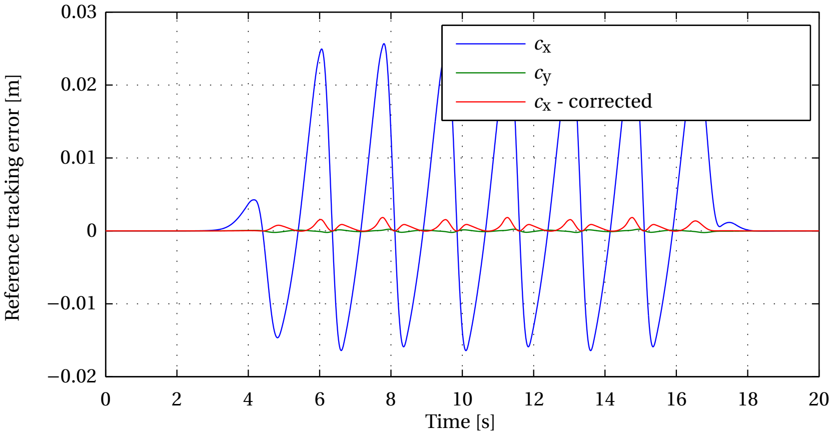


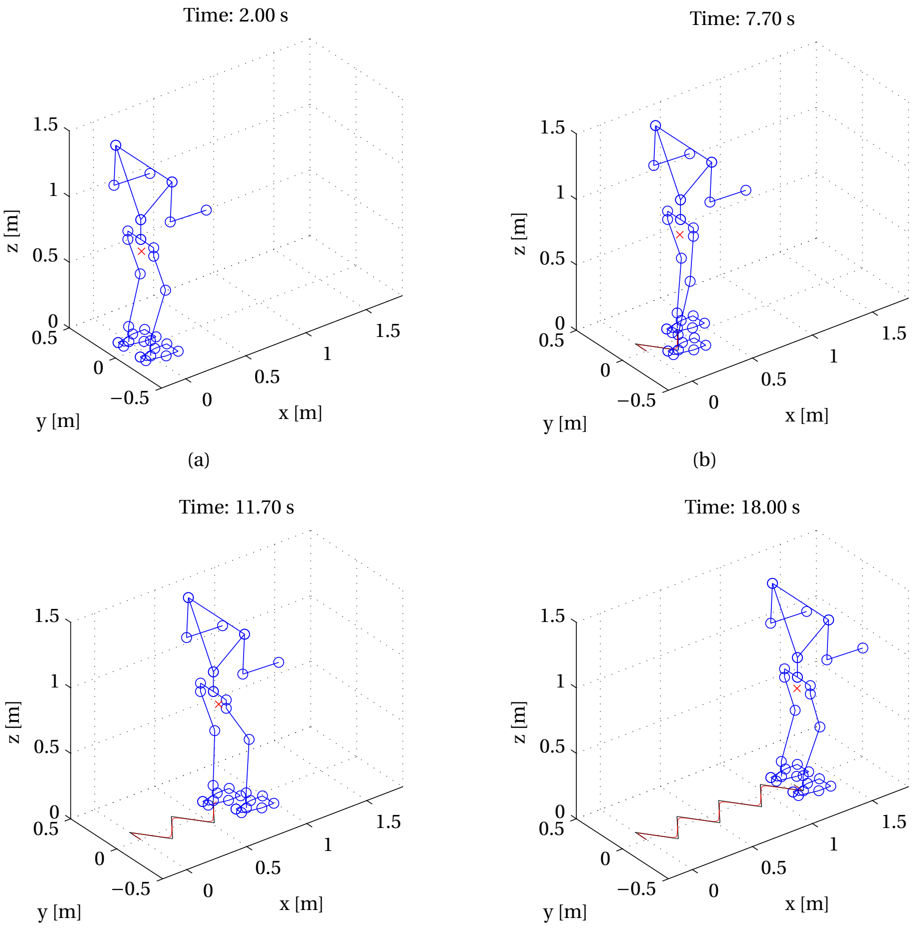
Figure 7.19. CoM tracking error in the transverse plane.

the transverse axis it is seen that the simulated output barely diverges and from a zoom is identified that the deviation is less than 0.2 mm. As earlier described the obtained error in the sagittal axis is expected since no sagittal CoM correction is used in this simulation. Using the simulated tracking error as CoM correction in the sagittal axis Δc_x and rerunning simulation the error in the sagittal axis shown with red in figure 7.19 is obtained.

From the figure it is seen that by using the correction the tracking error is reduced significantly, and a closer study of data shows that the maximum deviation is 1.8 mm. The tracking error is not completely eliminated due to the way the correction is implemented where the pelvis is moved proportional to the tracking error with a constant factor. Due to the movement of the legs in the sagittal axis the factor between CoM displacement and pelvis movement is not constant which is assumed in the correction. The tracking performance obtained with this correction is considered sufficient, and no better relationship will be derived.

For the movements in the transverse axis it could seem strange that the tracking error is small when the overall approach in tracking the CoM is the same as in the sagittal axis. The reason for this is that the factor between pelvis and CoM movement is close to constant and is barely affected by foot lift and movements in the sagittal axis.

A video showing the above described simulation is placed on the enclosed DVD `/videos/kinematic_gait_simulation.avi`. In figure 7.20 four snapshots from the video are shown.



(a)

(b)

(c)

(d)

Figure 7.20. Four snapshots of the posture during the simulated gait using the kinematics. The black line on the ground is the trace of the ZMP reference and the red is the trace of the obtained CoM from the simulation.

What is not evident from the above description and figures is that the gait performed in this simulation is not feasible for the robot due to angle limitations in the joints. The rear foot pitch angle is violated. Due to the choice of keeping the foot aligned with the ground at all times rather that making a heel off, the pitch angle of the ankle becomes much larger compared to the pitch angle obtained by humans during a similar gait. For a feasible gait the step length must be reduced.

7.5 Foot Impact Stabiliser

At the end of single support the swinging foot is placed on the ground at the desired position and the desired time if everything is perfectly timed. Perfect timing is however not obtainable in practice and in the most cases either early or late impact occurs. If early impact occurs, the swinging leg hits the ground before it has been fully extended. If no precaution is taken, the leg will continue the extension after ground contact and thereby make an undesired push off. Another effect of early impact is that the foot is not aligned with the ground, hence the robot will try to stand on the outer side of the foot.

A late impact will in general be harder to deal with than an early impact. While precaution can be made to handle the negative effects of an early impact there is not much to do in the case of a late impact other than wait for the impact to occur. The main reason for a late impact is that the mass of the robot has been moved too far towards the outside of the supporting foot.

In this section the arrangements made to handle the negative effects of early impacts are described.

7.5.1 Shock Absorbing Legs

As the foot of the robot hits the ground the impact is in many cases harmful to the stability of the gait since the impact often generates an undesired push off just after the impact. This undesired push off is boosted if the impact happens early since the leg is being extended when the impact occurs. To avoid this push off and soften the impact the legs are used as shock absorbers instead of keeping them locked. By using the forces measured in the FTS the length of the legs is modified as if they where shock absorbers.

The idea of using the legs as shock absorbers is inspired from [42], where the authors have implemented and tested the concept with success. In figure 7.21(a) the idea is illustrated, and in figure 7.21(b) the equivalent mass-spring-damper system for one leg is shown.

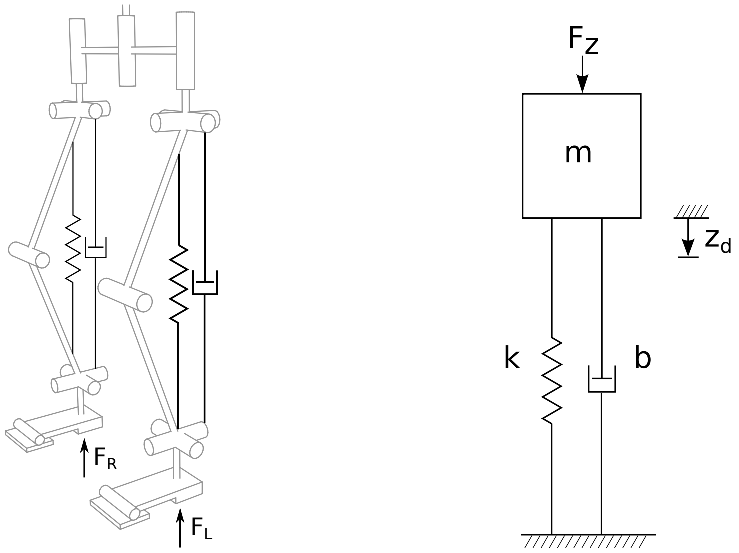
From figure 7.21(b) the equation for the displacement z_d for one leg in Laplace domain is found:

$$z_d(s) = \frac{1}{ms^2 + bs + k} F_z(s) \qquad [\mathrm{m}] \quad (7.25)$$

Where:

m is the mass on one leg	[kg]
b is the damping coefficient	[Ns/m]
k is the spring constant	[N/m]
$F_z(s)$ is the recorded ground reaction force from the FTS	[N]

The three parameters m, b, and k are determined from gait experiments, and the final choice



(a) Sketch illustrating the concept.

(b) Mass-spring-damper equivalence.

Figure 7.21. Virtual shock absorber in the legs of the AAU-BOT1.

of parameter values are described later. The length of the leg $\bar{z}$ is now determined as:

$$\begin{aligned}\bar{z}(s) &= z(s) + z_d(s) \\ &= z(s) + \frac{1}{ms^2 + bs + k} F_z(s)\end{aligned} \qquad [\mathrm{m}] \quad (7.26)$$

Where:

$\bar{z}(s)$ is the resulting length of the leg [m]

$z(s)$ is the reference length of the leg [m]

Two Shock Absorber Configurations

From experiments it was identified that the shock absorbing legs is an effective way to damp oscillations caused by flexibility in the mechanical structure. It is therefore chosen to always use the shock absorbers in the legs and not only during ground impact. To soften the impact in a satisfactory way the shock absorber needs to be relatively soft. A too soft shock absorber is however not desired since this would cause too much subsidence in the legs. To achieve a soft landing while avoiding too much subsidence in the legs during gait, a solution of combining two different shock absorbers is proposed. A soft shock absorber configuration is used while the leg is swinging and during impact, but after impact the shock absorber is gradually switched to a harder configuration that has less subsidence.

In table 7.1 the parameters for the two shock absorber configurations are shown. The mass m is chosen to be the half of the entire robot mass, the spring constant k is chosen to obtain a desired maximum displacement z_d, and finally is the damping coefficient b chosen from experiments to obtain the desired dynamical properties. In figure 7.22 a step response for the two configurations is shown. The amplitude of the step is −658 N which is the maximum steady state force.

	m [kg]	b [Ns/m]	k [N/m]
Hard	33.5	51200	256000
Soft	33.5	7600	41600

Table 7.1. Parameters for the shock absorbers in the legs.

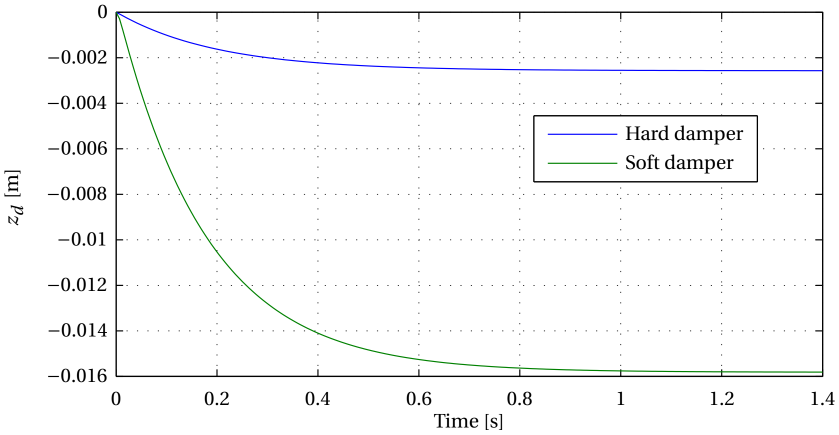


Figure 7.22. Step response of the two shock absorbers with a step amplitude of −658 N.

From figure 7.22 it is seen that the hard damper has a slight more damped response and a lower amplitude. This complies with the specifications where the soft damper must soften the landing and a large displacement is necessary, whereas a large displacement is not desired for the hard damper.

To combine the effects of the hard and soft shock absorbers, the output of equation (7.26) is calculated for both the hard damper $\bar{z}_{hard}$ and for the soft damper $\bar{z}_{soft}$ in parallel and their outputs are combined:

$$\bar{z} = w \cdot \bar{z}_{hard} + (1 - w) \cdot \bar{z}_{soft} \tag{7.27}$$

The weight w is designed as shown in figure 7.23. The two vertical lines indicate discrete events that are used to synchronize the development of the weight w.

The transition after the impact is chosen to be a cosine since this shows better results in experiments compared to a linear transition. The duration of the transition is chosen as long as possible to make the transition smooth, but short enough to ensured that the hard damper is used ($w = 1$) before the double support ends. The transition time is chosen to be 0.5 s. It should be noted that the state of the soft damper is forced to be equal to the hard damper in the transition from 1 to 0 to avoid a transient in the final output.

To avoid push off with the extending leg when ground impact occurs, it is chosen to stop the extension of the leg. This is done by holding the reference to the soft damper when impact is detected until $w = 1$. In figure 7.24 the effects of the shock absorbers are shown for a gait experiment.

From the figure it is seen in the beginning that both feet are lifted from their initial position by approximately 1.3 mm, which complies with the specification of the hard damper (see figure 7.22) where the total force is divided on the two feet. At time 35.5 s the mass of

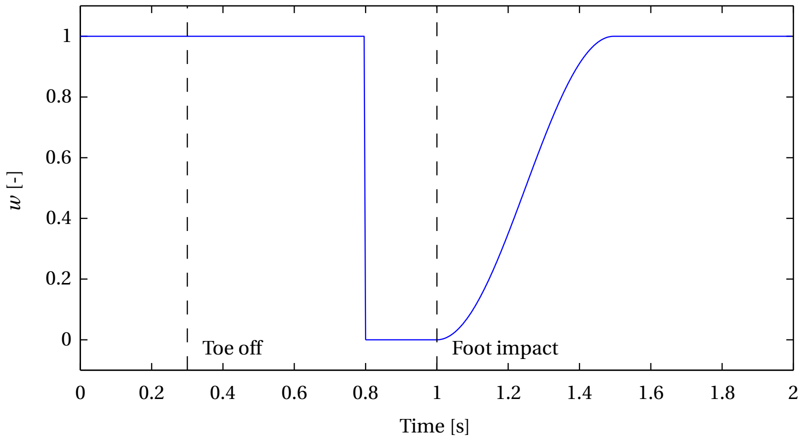


Figure 7.23. Example of how the weight w changes for one leg during a step.

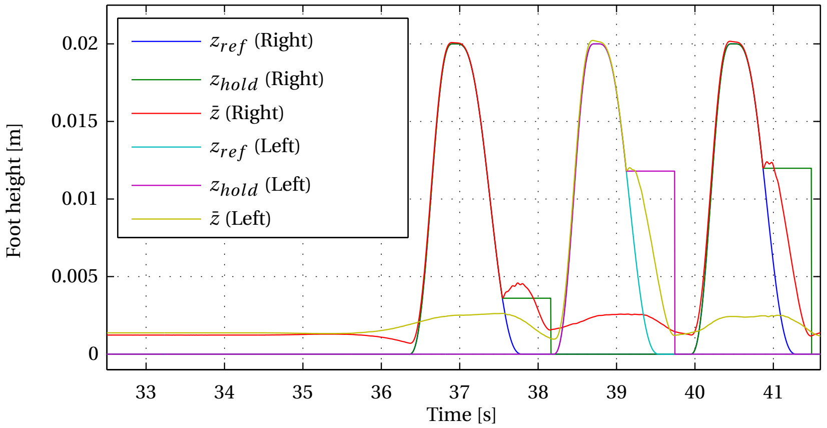


Figure 7.24. Effects of the spring dampers recorded during a gait experiment. The height shown is the lift of the feet from their initial positions rather than the actual length of the legs.

the robot is moved towards left and the compression of the left damper increases to the full compression of 2.6 mm for the hard damper. In the same period the right leg is lifted and about time 36.5 s the height of the foot converges to the reference height indicating that the foot has lost ground contact ($F_z = 0$).

At time 37.5 s the right foot has impact with the ground. From the figure it is seen that the reference is held constant (only for the soft damper) and due to the softness of the soft damper the foot is raised again to soften the landing ensuring that the right foot not leaves the ground again. After impact both legs are extended since some mass is moved from the left leg to the right. The cycle is now repeated for lifting the left foot.

From the figure it is seen that the transition time for switching from the soft to the hard damper cannot be any longer since the transition is just completed when the left foot starts to lift. This is observed just after time 38 s where the held value jumps to the reference again indicating that the transition is completed and the hard damper is used.

Align Feet with Ground Surface

To make sure that the feet are aligned with the ground surface when the length of one leg diverge from the other due to compression of one leg, the roll angles of the ankles are compensated. The compensation angle is shown in figure 7.25. To the left the situation is sketched for the uncompensated case, where the robot is supporting on the edge of the feet. To the right the roll angles are compensated by adding the correction angle θ_{cor} to the original roll angle reference. On the figure the corrected feet orientations are illustrated with blue.

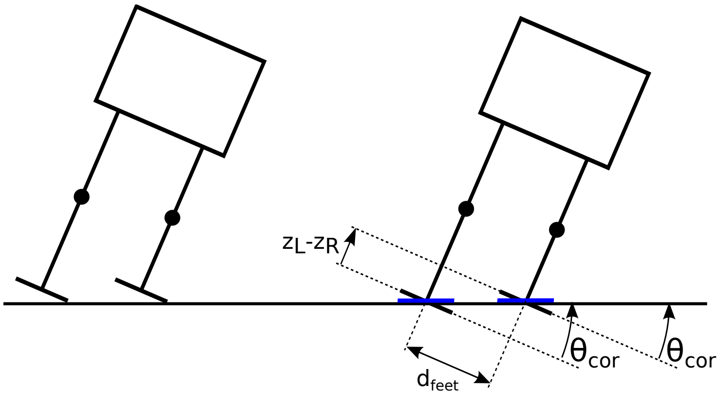


Figure 7.25. Foot roll angle compensation to ensure the feet are aligned with the ground surface.

The correction angle is calculated as:

$$\theta_{cor} = \text{atan}\left(\frac{z_L - z_R}{d_{\text{feet}}}\right) \qquad \text{[rad]} \quad (7.28)$$

Where:

z_L is the reduction in left leg length due to compression [m]

z_R is the reduction in right leg length due to compression [m]

d_{feet} is the distance between the centre of the feet [m]

It should be stressed that the reduction in length of the legs used in equation (7.28) is only due to compression of the shock absorbers and not due to change in reference to lift the feet from the ground.

Shock Absorber Tests

To verify the effects of the shock absorbers an experiment is carried out. The robot is put in its initial position described in section 7.4 and manually held in an angle as shown in figure 7.26 such that the right foot is raised 20 mm above ground level.

The robot is released and the length of the legs is recorded alongside with the force measurements from the FTS's. The experiment is conducted without dampers, only with the hard dampers, and finally with a combination of the hard and the soft dampers. On the DVD a video is placed that shows the behaviour of the robot during the test with the different configurations 💿`/videos/shock_absorber_drop_comparison.avi`.

In figure 7.27 the measured forces are shown and in figure 7.28 the reduction in leg length is shown for the two tests with shock absorbers activated. In both figures the right foot impact with the ground occurs at time 0.2 s.

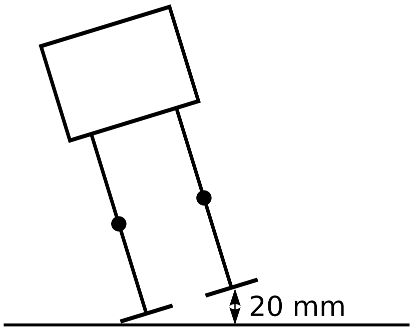


Figure 7.26. Shock absorber experiment setup.

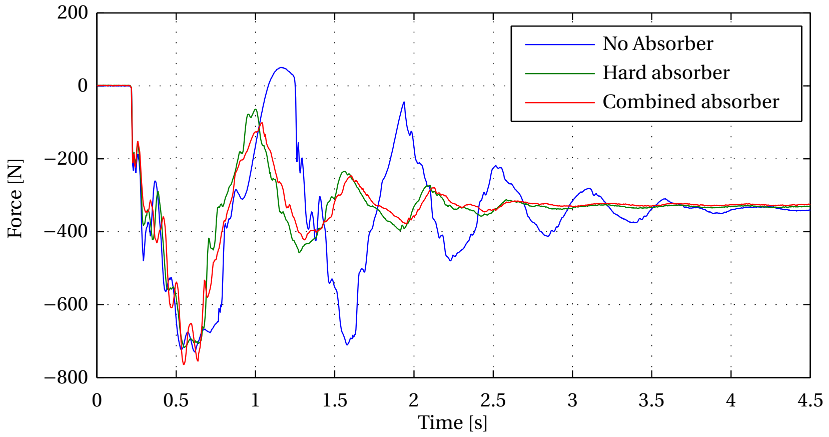


Figure 7.27. Ground reaction force (F_z) measured by right foot FTS during the experiment.

From figure 7.27 the effect of the shock absorbers are clearly visible. Comparing the test with dampers to the one without dampers it is seen that the oscillations dies out much faster with the dampers. Additionally it is seen that for the experiment without the dampers that the right foot leaves the ground again at about time 1.2 s. The fact that the force becomes positive is caused by dynamics in the FTS's when no force is applied to the foot. In the damped cases the foot stays on the ground after impact for both experiments with absorbers. From the figure no significant difference is present in forces when using the combined damper compared to only the hard damper.

In figure 7.28 the compression of the legs is shown. From the figure it is seen that the combined absorber compresses more as expected. Even though figure 7.28 shows different behaviour between the two different shock absorber setups this is not visible in the ground reaction forces, hence the advantage of using the combination of shock absorbers can not be verified directly from the figures.

To prove the advantage in using the combined damper a gait experiment has been carried out. One test with only the hard dampers and one test with the combination. A video containing both tests can be found on the enclosed DVD 💿 `/videos/shock_absorber_gait_comparison.avi`. From the video it is seen that when only using the hard dampers the gait become more wobbly at the end and at the last step the robot falls. On the other hand using the combination of the dampers makes the gait steadier and the robot completes all the planned steps.

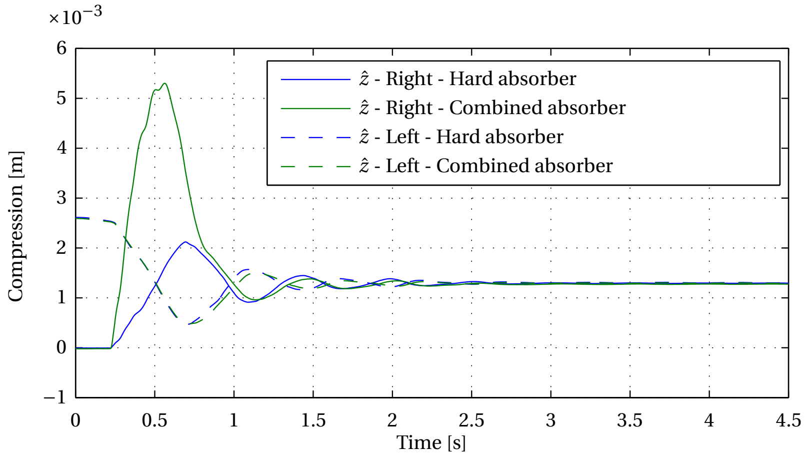


Figure 7.28. Reduction of leg length during the experiment.

7.5.2 Foot Angle Correction

The second precaution taken to reduce the harmful effects of an early impact is that the swing foot is rotated such that it will land flat on the ground. In figure 7.29 this is illustrated. To the left the left foot is lifted and the robot is standing on the right foot. In the figure the compensated foot orientation is illustrated with blue while the reference orientation is illustrated with black. To the right in the figure it is seen that the left foot lands flat on the ground if the robot falls over the edge of the right foot.

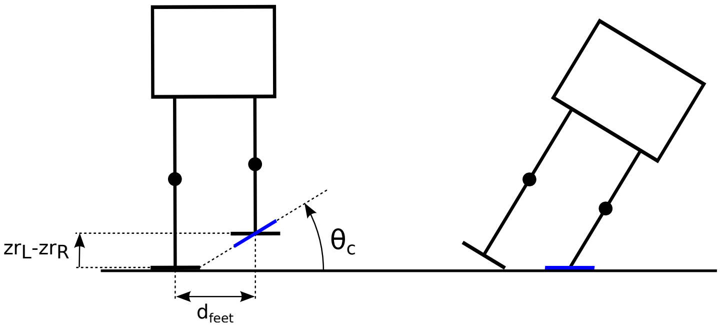


Figure 7.29. Swinging foot roll angle compensation to ensure the feet are aligned with the ground surface in case of early impact. To the left the distances for calculating the compensation angle θ_c is shown. To the right it is shown that the left foot is flat on the ground even if the robot falls over the edge of the right foot in right support.

The compensation angle is calculated as:

$$\theta_c = \text{atan}\left(\frac{zr_L - zr_R}{d_{\text{feet}} - w_{\text{foot}}/2}\right) \qquad \text{[rad]} \quad (7.29)$$

Where:

zr_L is the reduction in left leg length due to foot lift [m]

zr_R is the reduction in right leg length due to foot lift [m]

d_{feet} is the distance between the centre of the feet [m]

w_{foot} is the width of one foot [m]

It should be noted that the reduction in leg length zr_L and zr_R is only the contribution from the foot lift reference. It does not include the reduction in leg length due to the shock absorbing legs described in section 7.5.1. The correction angle is only added to the swinging foot and the angle of the supporting foot is not affected. If $\theta_c > 0$ it is added to the left ankle roll angle while it is added to the right ankle roll angle if $\theta_c < 0$.

7.6 CoM Controller

To keep the ZMP within the PoS of the robot, it is desired to follow the CoM trajectory found in section 7.3, since this will make the ZMP follow its reference. Only control of the CoM in the transverse direction is considered since this is the most critical. Ideally the CoM should also be controlled in the sagittal axis, but transverse axis CoM tracking is considered more critical because the PoS is narrower in single support. Additionally the amplitude of the movement of the CoM in the sagittal axis can be reduced by reducing the step size while the amplitude of the CoM movement in the transverse axis cannot be reduced due to the needed separation of the feet.

In chapter 5 it was shown that in double support the posture controller was able to make the joints follow a reference, which implies that a desired CoM can approximately be obtained directly by specifying the joint angles to the posture controller. In figure 7.30 the CoM and ZMP is shown for an open loop experiment where the CoM reference shown in figure 7.7 is tracked by calculating joint angles using the inverse kinematics and following them using the posture controller. In the experiment both feet are on the ground at all times.

From the figure it is seen that the overall shape of the CoM and ZMP reference is followed. When the ZMP is between the feet the reference is tracked with a small error. On the other hand when the ZMP is within one foot larger errors are recorded. At 30.5 s and just before 34 s it can be observed that the ZMP moves towards the inside of the foot. By calculating the velocity of the resulting CoM it is observed that the velocity stays close to zero for a short period of time. As a result the ZMP moves towards the CoM as expected from the cart-table model described in section 4.3.3, since the acceleration of the CoM will be close to zero. Observing this tendency yields for a controller that could reduce the tendency of getting stuck when changing direction of the CoM.

From figure 7.30 it is seen that when the ZMP enters within the width of the feet, it stays within this area until it is desired to leave according to the reference. This indicates that a gait

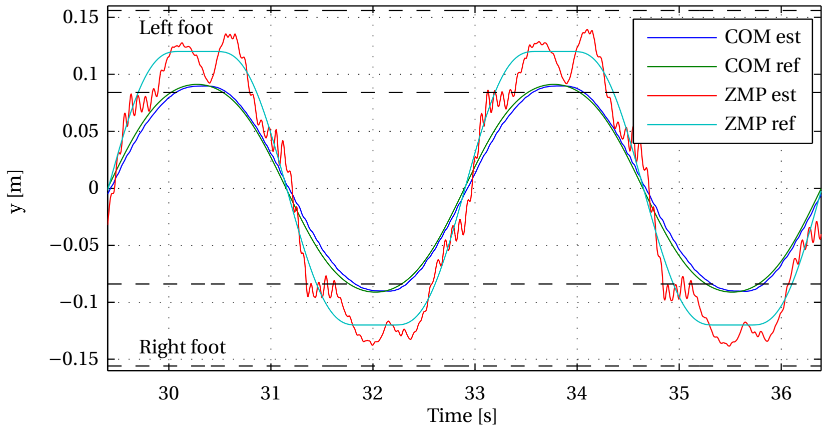


Figure 7.30. CoM and ZMP output of the system in open loop with CoM and ZMP reference for the transverse movement according to figure 7.7. No foot lift and no movement in the sagittal axis is performed. The black dashed lines indicate the width of the PoS of the feet.

could be performed by lifting and swinging the non-supporting foot. In figure 7.31 the output CoM is shown for a gait with a step length of 50 mm. In addition to reducing the step length compared to what is presented in section 7.2 it is also necessary to move the ZMP additionally 6 mm towards the outside of the foot to keep the robot in balance.

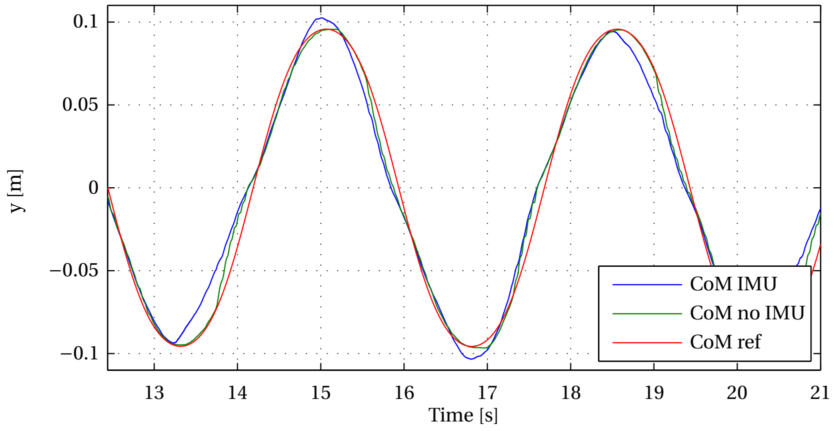


Figure 7.31. CoM output with and without correction from the IMU from a gait with step length 50 mm performed in open loop.

In figure 7.31 the calculated CoM is shown both with and without the additional information from the IMU about deformation in the ankles described in section 6.2. From the figure it is observed that at the change in direction about time 13.5 s and 18.5 s the CoM calculated without the correction seems to track the reference while the deformation makes the CoM move towards the inside of the foot. For the change in direction at time 15 s and 17 s the opposite situation is observed where the IMU corrected CoM moves towards the outside of the supporting foot. In this situation it is seen that the CoM calculated directly from the joint angles seems to get stuck. This correspond nicely with the fact that the CoM calculated with the IMU correction overshoots the reference hence the joint need more power to pull the entire mass of the robot towards the inside of the supporting foot.

From figure 7.31 it is seen that the deformation measured by the IMU when the foot is lifted causes much larger deviation from the CoM reference than what is introduced by the posture controller alone. The main objective of the CoM controller must therefore be to reduce the influence of the disturbance from the deformation.

To reduce the effect of the disturbance two different controller strategies were considered. In the first strategy the CoM error was reduced by controlling the CoM input to the system from which the joint angles are calculated. In this way almost all joints in the robot are used to reduce the CoM error. The effort used to reduce the error is therefore distributed among several joints. In the second strategy the CoM error is reduced only by modifying the joint angle reference to the ankle roll joint of the supporting foot. This idea reduces the error close to where the disturbance enters, but puts a lot of strain on one specific joint.

Preliminary tests of both strategies were performed. A CoM controller using the posture of the robot to control the CoM was successfully implemented. It was however not possible to achieve sufficiently bandwidth with this type of controller, without causing too many vibrations in the robot. A better performance was achieved using the ankle roll angle to move the CoM. In the rest of this section the design of the CoM controller using this strategy will be described.

Ankle Correction CoM Controller

In figure 7.32 the proposed controller structure for controlling the CoM by modifying the roll angle reference to the supporting ankle is shown.

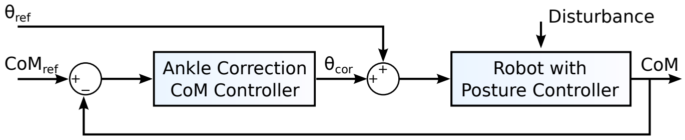

θref
Disturbance
CoMref
+
−
Ankle Correction CoM Controller
θcor
+
+
Robot with Posture Controller
CoM

Figure 7.32. Block diagram of the ankle correction CoM controller.

In figure 7.32 θ_{ref} is the joint angle reference generated in section 7.4, and the CoM reference is the one generated in section 7.3. The idea in the controller is to only modify the roll angle of the supporting ankle by adding a correction angle θ_{cor} to the prescribed joint angle reference. This correction angle is calculated from the error between the CoM calculated with the information about the deformation in the structure from the IMU and the CoM reference.

To be able to design a controller that reduces the effect of the disturbance, a model from ankle roll angle to transverse movement of CoM is deduced. From the designed gait it is known that the feet are parallel with the sagittal axis such that a change in the roll angle of the ankle only implies movement of the CoM in the transverse axis. The model is derived from the joint model given in section 4.1 and the posture controller described in section 5. The joint model is derived in the same way as the knee model was derived in section 5.1.2. An estimate of the total inertia of the robot, seen from the ankle roll joint, is used as load inertia in the model. The estimate of the inertia is calculated assuming the robot is a cuboid rotated about its corner with a height of 1.54 m, width of 0.44 m, and mass of 64 kg which is uniformly distributed. The inertia is calculated to be: $J = 54.98\,\text{kgm}^2$.

Using the joint model with this inertia and the posture controllers for the ankle roll joints specified in table 5.1 the model from angle reference input to angle output is found. A linearised relation between torso roll angle (θ_{roll}) and CoM movement in transverse axis (c_y) is utilised:

$$c_y = c_z\theta_{\text{roll}}$$

where:

$c_z = 0.741\,\text{m}$ is the constant height of the CoM

The resulting model is of third order with one zero. The bode plot for the model is shown in figure 7.33. The dynamic is dominated by a complex pole pair at 6.92 rad/s. The third pole and the zero are both placed near 0.5 rad/s and therefore not visible from the bode plot.

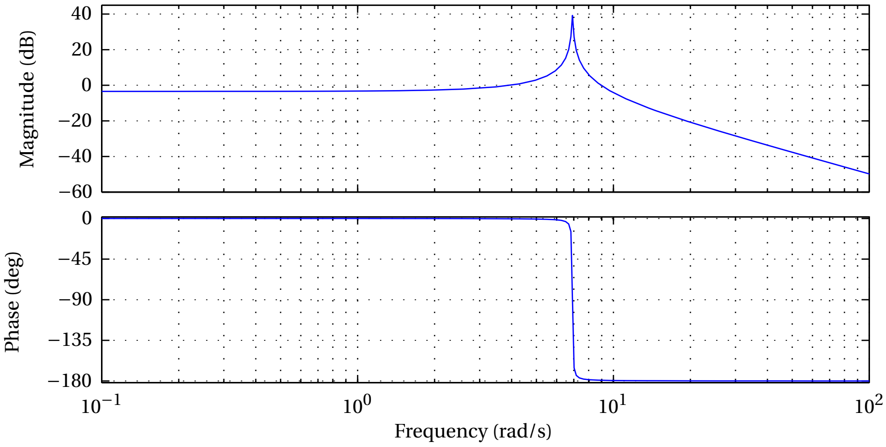


Figure 7.33. Bode plot for the model from angle reference input to CoM output.

From the bode plot it is seen that this model of the system is only marginally stable and the design requirement from chapter 5 of phase margin of 45 ° is not obtained. The gains for the posture controller were initially obtained using the same model for the joint, but the gains were adjusted to make the controllers perform well in practice. The gain in the inner loop controller were reduced which apart from making the inner controller stable also makes the final model less damped. The underdamped behaviour seen from the bode plot is however not observed in reality. A plausible reason for this is that the friction in the ankle roll joint is higher than assumed in the joint model. The friction in the joint was determined from an experiment where the joint only was loaded with the mass of the foot where the load when in operation is the entire mass of the robot. Additionally the coulomb friction was not included in the joint model. From table 4.3 it is seen that for low angular velocities the coulomb friction contribution is much larger than the viscous friction included in the model.

Since the behaviour seen in the bode plot in figure 7.33 does not comply with what observed in reality, it does not make any sense to design a CoM controller from this. To obtain a more descriptive model of the system an experiment could be conducted. It is however hard to conduct for the robot in single support since balance must be maintained during the experiment.

Without a descriptive model of the system a PI controller with hand-tuned gains is attempted. During a series of gait experiments the proportional gain is chosen to be $k_\mathrm{p} = 1$ and the integral gain is chosen to be $k_\mathrm{i} = 0.25$. Choosing the gains larger than this, the supporting foot starts to rotate during gait. If the torque applied in the ankle could be limited when the ZMP is close to the edge of the PoS, the rotation of the foot would not be experienced even if the gains are increased. To implement this, a deeper study of a torque limitation mechanism for the posture controller must be carried out. Initially the PI controller with the above gains will be tested.

As previously described the above controller only makes corrections to the joint angle reference for the supporting foot. By using the controller only for the supporting foot, the rotation of the swinging foot described in section 7.5.2 is not affected and a flat foot landing, in case of early impact, is still ensured. The joint reference to the ankle joints must be made smooth in the transition from uncontrolled to controlled case and vice versa. To create this it is chosen to gradually fade the control signal in and out when the robot enters and leaves single support. To make sure that an integral error is not carried on from one support phase into another, the integral term is reset between each single supports phase. In figure 7.34 the controller weights used to fade the controllers in and out are shown as the blue and green curve. The red curve shows when the integrators are active. When it is disabled the integrated error is reset to zero. The support phase shown as the two fainted blue and yellow areas is determined from the *support ratio* defined in equation 6.2.

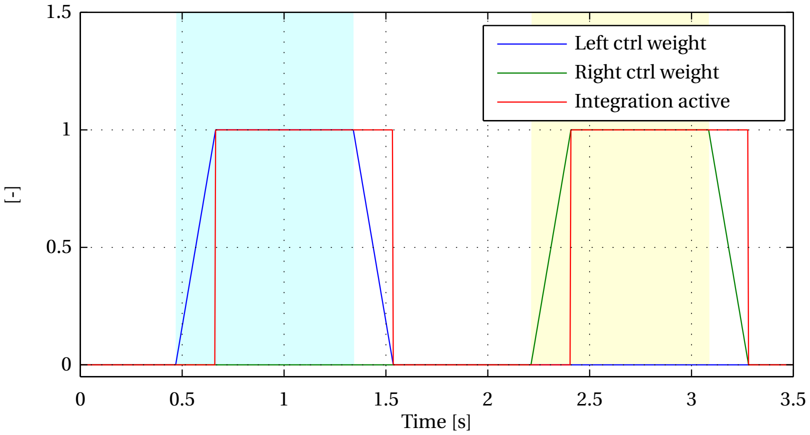


Figure 7.34. The blue and green curve shows how the controller outputs are faded in and out between single and double support. Red curve indicates when the integrator is active. When it is one, the integration is active. When it is zero, it is disabled and reset to zero. The faint blue area indicates left single support. The faint yellow area indicates right single support.

For the CoM controller anti-windup is considered in two cases. First the current limitation in the motors is considered. It has been observed that the foot starts to rotate with respect to the ground before the current limitation in the motor is reached. This renders the current anti-windup unnecessary.

Secondly, if the foot rotates with respect to the ground, the ankle correction will not have any effect on the CoM. This means that any error in the CoM position will not be reduced by a controller input. This will cause the integral term to diverge. However if this happens, either the robot will fall away from the supporting foot and nothing can be done to prevent this, or it will land on the other foot and enter double support where the integral term is reset. In both cases anti-windup will not help.

With the above described CoM controller a gait experiment has been conducted to verify the performance of the controller. The gait is similar to the one performed in open loop with the only difference that the ZMP is centred on the foot which should give the best stability margin. The CoM tracking error is shown in figure 7.35 for both the open loop experiment and the experiment with the CoM controller.

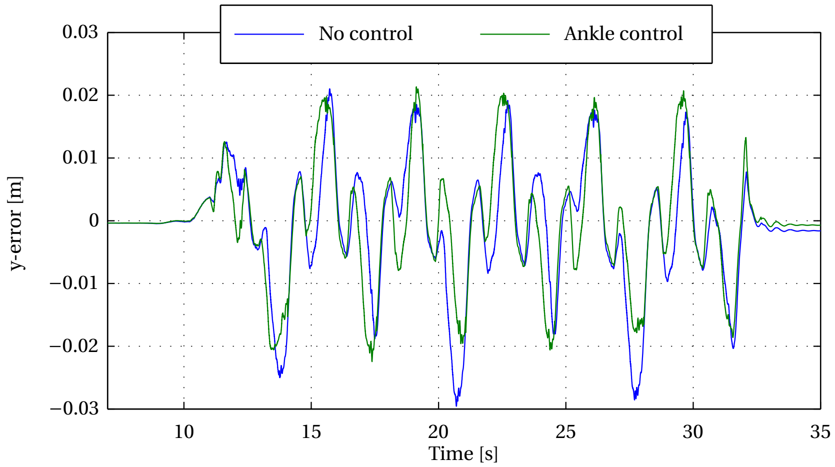


Figure 7.35. Error between CoM reference and obtained CoM in the transverse axis.

The error plotted in the figure shows the same overall behaviour with or without the controller. The 12 steps performed in the gait are easily identified from the large error when the ankle deforms and the robot falls towards the inside of the supporting foot. Even though the overall shape is the same, it is seen that the maximum error is reduced using the CoM controller. By further inspection of the data the maximum error experienced in open loop is 29 mm while it with CoM controller is reduced to 22 mm.

From the figure it is also seen that the observed error is more uniform for the gait with controller. The 11 last spikes have an amplitude of about 20 mm, where the errors for the uncontrolled system vary much more in amplitude. While the reduction in the amplitude of the error might seem insignificant, it has proven to be a vital improvement. The resulting steps become more constant because the time of ground impact with the foot in each step becomes more constant. This has shown to reduce the risk of falling significantly.

From this result it is seen that the CoM controller is capable of reducing the CoM tracking error. It could however seem like higher gains could benefit the reduction of error even more. This is however not possible since the feet are almost rotating when the largest error is experienced.

7.7 Proposals to ZMP Control

This section will not introduce a ZMP controller but describe some initial considerations that can be used for further research. For ZMP control it is suggested to include the use of torso rotation to move the ZMP. In this way the ZMP can be controlled by either or both using torso and pelvis movement. First the cart-table model is extended to introduce the torso roll angle as an input variable. Figure 7.36 shows the principle behind this deduction, where a pendulum with mass m_2 is placed on top of a mass m_1 which is assumed to only move on the y-axis with constant height c_z. The mass m_2 represents the mass of the upper body, while m_1 represents the mass of the rest of the AAU-BOT1. The distance r shown in figure 7.36 is the distance from torso roll joint to CoM of the upper body. The angle θ is the angle of the upper body introduced by the torso roll joint.

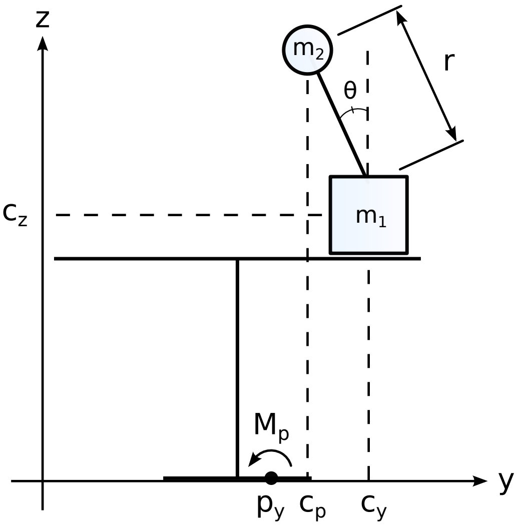


Figure 7.36. The cart-table model for ZMP movements in the frontal plane extended with torso rotation.

The same approach as in the deduction of the cart-table model in section 7.36 will be used. All the contributions to the moment M_p about the zero moment point p_y is found. The pendulum is linearised about $\theta = 0$, hence the height of m_2 becomes constant $(c_z + r)$. The following equation describes the moments about p_y:

$$M_p = m_1\, g(p_y - c_y) + c_z\, m_1\, \ddot{c}_y + m_2(c_z + r)\ddot{c}_y + m_2\, g(p_y - c_p) + J_p\, \ddot{\theta} + m_2\, r\, c_z\, \ddot{\theta} \qquad \text{[Nm]}$$

The position of the pendulum in the y-axis (c_p) can be stated as a combination of CoM position c_y and angle of the pendulum θ: $c_p = c_y - r\,\theta$. The moment of inertia of the upper body J_p is calculated by considering the upper body as a point mass m_2 rotating about a point at a distance of r: $J_p = m_2 r^2$. By introducing the upper conditions and zero moment about p_y a new expression can be deduced for the position of the zero moment point.

$$0 = m_1 g(p_y - c_y) + (c_z m_1 + m_2(c_z + r))\ddot{c}_y + m_2 g(p_y - c_y + r\theta) + m_2 r(z_c + r)\ddot{\theta} \qquad \text{[Nm]}$$

$$p_y = c_y - \frac{c_z m_1 + m_2(c_z + r)}{(m_1 + m_2)g}\ddot{c}_y + \frac{m_2 r}{m_1 + m_2}\theta - \frac{m_2 r(z_c + r)}{(m_1 + m_2)g}\ddot{\theta} \qquad \text{[m]} \quad (7.30)$$

From equation 7.30 it can be seen that movement of the lower body and torso roll contributes as separate terms that can be superimposed to move the ZMP. Since the position of the lower body and angle of torso can be used as two completely separate inputs to the system, two separate transfer functions can describe their influence on the ZMP.

$$\frac{p_y}{c_y} = 1 - \frac{c_z m_1 + m_2(c_z + r)}{(m_1 + m_2)g}s^2 = 1 - 0.143s^2 \qquad (7.31)$$

$$\frac{p_y}{\theta} = \frac{m_2 r}{m_1 + m_2} - \frac{m_2 r(z_c + r)}{(m_1 + m_2)g}s^2 = 0.0817 - 0.0135s^2 \qquad (7.32)$$

where:

$$m_1 = 48.76 \qquad [\text{kg}]$$

$$m_2 = 18.24 \qquad [\text{kg}]$$

$$r = 0.30 \qquad [\text{m}]$$

$$c_z = 0.741 \cdot 1.79 = 1.33 \qquad [\text{m}]$$

The models deduced above gives conversion from resulting CoM and torso angle to ZMP. For control purposes the dynamics for CoM and torso movements must therefore be included to obtain the final dynamic ZMP models.

The movement of the CoM is a complex combination of all the joint controllers designed in chapter 5. This makes it difficult to deduce a CoM model directly from this work. Instead an experimental approach is taken, where a step input is applied to the CoM in the transverse axis with both feet on the ground. The joint angles during the step response are recorded and converted into a CoM through the forward kinematics.

In figure 7.37 the recorded CoM is shown as the green curve. To fit a model to the recorded data the *System Identification Toolbox*™ in MATLAB® is used to create a black box model. Different model orders have been attempted and the best compromise between simplicity and performance has shown to be the fourth order model with one zero seen in equation (7.33).

$$\frac{c_y(s)}{c_{\text{y-ref}}(s)} = \frac{40256s + 892165}{s^4 + 142s^3 + 8123s^2 + 110898s + 880354} \qquad (7.33)$$

Figure 7.37 shows the simulated step response of the model as a red curve. It might seem that only a small amount of the dynamics have been captured with this model. However in order to improve the model significantly, models of order eight or higher is necessary.

The dynamic model of the torso has been determined in a similar way and can be seen in equation (7.34).

$$\frac{\theta}{\theta_{\text{ref}}} = \frac{23954}{s^3 + 52.66s^2 + 1633s + 23375} \qquad (7.34)$$

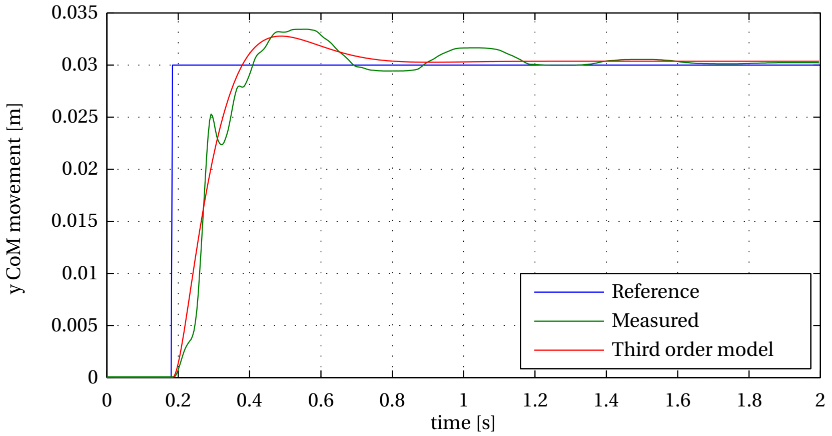


Figure 7.37. Measured and simulated step response on the CoM.

By multiplying the transfer functions for the CoM movement given in equation (7.31) and (7.33), and multiplying the transfer functions for the torso movement given in equation (7.32) and (7.34) we get the two dynamic transfer functions seen in equations(7.35) and (7.36). A bode plot of the two can be seen in figure 7.38. Both of the two transfer functions have a zero in the RHP introduced by the ZMP model, which renders them non-minimum phase. This was also observed in the CoM generation in section 7.3.

$$\frac{p_y}{c_{\text{y-ref}}} = \frac{-5772s^3 - 127929s^2 + 40256s + 892165}{s^4 + 142s^3 + 8123s^2 + 110898s + 880354} \tag{7.35}$$

$$\frac{p_y}{\theta_{\text{ref}}} = \frac{-324s^2 + 1957}{s^3 + 52.66s^2 + 1633s + 23375} \tag{7.36}$$

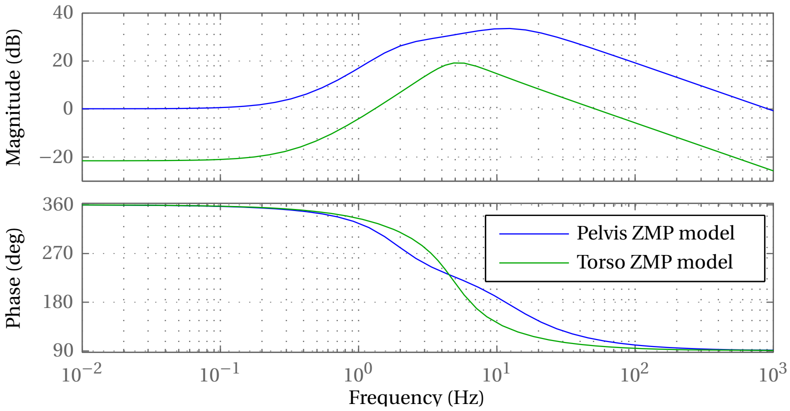


Figure 7.38. Bode plot of the dynamic ZMP model with movement of CoM.

The fact that the system is non-minimum phase complicates the control of the ZMP. Different solutions to this problem have been proposed in literature. The easiest way to deal with non-minimum phase systems is to reduce the closed loop bandwidth of the system to be slower than half the frequency of the RHP zero [43]. For the two transfer functions the RHP zero is located at 0.42 Hz and 0.39 Hz which means that this approach is not usable. An alternative approach is to reverse the sign of the controller gain and limit to controller to only handle frequencies higher than two times the RHP zero [43]. This approach is used in [42] where the ZMP tracking error was reduced. Another approach suggests utilising that the two inputs to the system can be controlled individually. The underswing created by movement of the CoM can be counteracted using torso roll in the opposite direction [44].

The controller strategy to control the ZMP will be left for future project groups to design. At this point stability is ensured only through CoM control.

Chapter 8

Acceptance Test

In this acceptance test the obtained gait will be analysed and compared to the thesis objective specified in section 1.2. Four different combinations of stride time and step lengths will be presented, and in each of the tests the AAU-BOT1 walks 6 strides in a straight line. The four different configurations are shown in table 8.1. Each test is performed ten times to assess robustness of the gait. The success rate shown in this table indicates how many times out of ten that the robot kept its balance during the entire test. The plots shown in this chapter are created from data recorded during the acceptance test which can be found on the DVD in the following folder 💿`/data/acceptance_test/`. For the first three tests only pelvis movement in the transverse plane is used to move the CoM. In the test where 5 s stride time is used the CoM needs to be moved further away from centre position and this is not possible with the chosen initial posture. To compensate for this the torso roll is used with a maximum roll angle $\theta_{\text{r-max}} = 0.2\,\text{rad}$. Videos from the acceptance test is placed on the enclosed DVD with descriptive titles and following prefix 💿`/videos/acc_test_*`.

No.	**Stride time** [s]	**Step length** [mm]	**Velocity** [m/s]	**Success rate**
1	3.5	25	0.014	10/10
2	3.5	50	0.029	10/10
3	3.5	75	0.043	9/10
4	5.0	50	0.020	10/10

Table 8.1. Configuration of tests performed in the acceptance test.

Different step lengths have been attempted but step lengths above 75 mm makes the robot fall often. During the tests, the estimated position of the ZMP and CoM is recorded along with the position of the feet and the support phase. The position of the feet and support phase is used to determine the position and size of the PoS at all time during gait. In figure 8.1 the

ZMP and CoM is plotted in the sagittal axis as a function of time for a stride time of 5 s and step length of 50 mm. The white area indicates the boundary of the PoS.

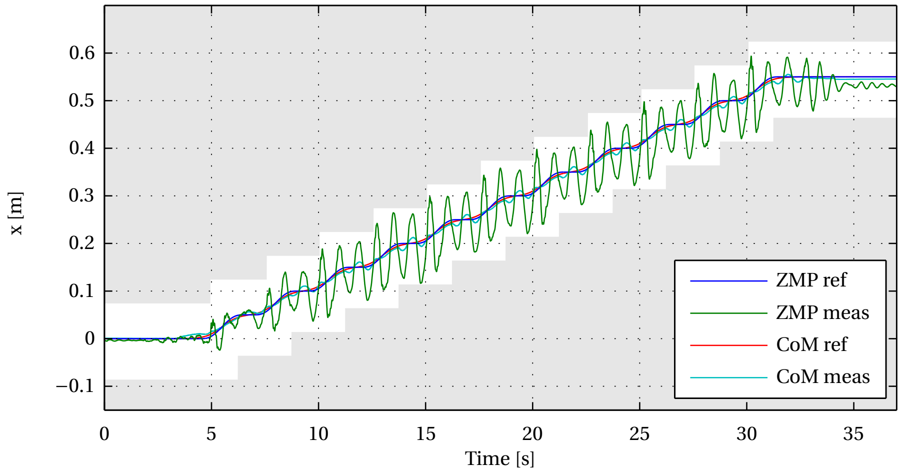


Figure 8.1. Plot of ZMP and CoM in sagittal axis during gait with 5 s stride time and 50 mm step length. The white area indicates PoS.

To ease reading of the plot a zoom of the area of the time between 13 s and 23 s is shown in figure 8.2. In both figures the ZMP can be seen to fluctuate significantly back and forth, it however stays within PoS, hence it is dynamically stable in the sagittal axis. The fluctuations in the ZMP do not seem to be caused by the controllers since the oscillating behaviour is not found in the recorded joint angles. Instead this seems to be a result of the flexibility in the robot, which results in a behaviour similar to an underdamped spring-damper system. Each time a limp is moved the resulting forces causes this system start oscillating. The problems gradually increase as the movements of the robot becomes faster. The CoM is held much more constant with a large margin to the edges of the PoS. This shows that the gait is statically stable in the sagittal axis.

In figure 8.3 and 8.4 the ZMP and CoM is plotted for the transverse axis. Where the ZMP in the sagittal axis stays strictly inside the PoS, the ZMP in the transverse axis leaves PoS for short periods of time. This happens in right support, where the ZMP leaves the PoS in direction towards the opposite foot. This is seen in every right single support, and in 8.4 it is seen at 17 s and 22 s. This should indicate that the robot is actually falling, this is however not the case since the robot recovers in every case. In the AAU-BOT1 FTS's are used to determine where the ZMP is located. Using this measurement the ZMP, it should not be possible to measure a ZMP that is outside PoS. The fact that this happens anyway indicates that either the measurement of the ZMP is not completely accurate or the PoS is in fact slightly bigger than what has been found in section 6.2. From the measurements of the ZMP, the robot can not be shown to be dynamically stable in the transverse axis. Taking possible deviations in the measurement and PoS into consideration the measurement at least indicates that the robot is only marginally stable.

Studying the CoM in figure 8.4 it can be seen that the main reason for the poor

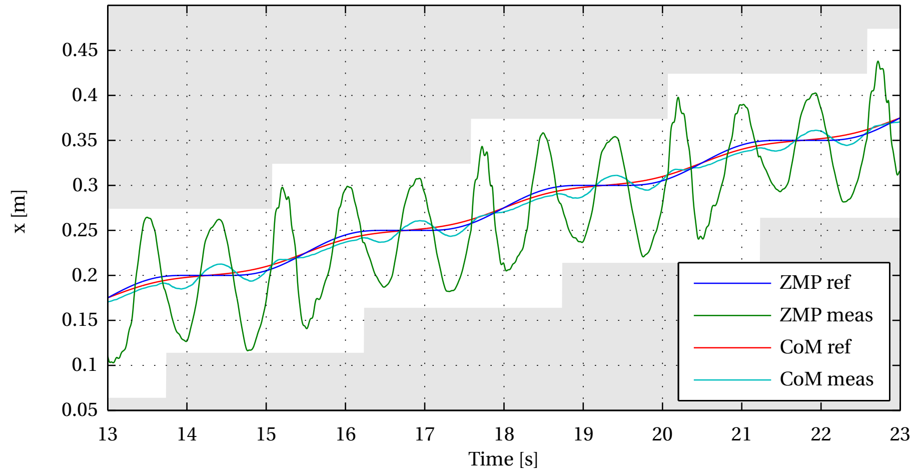


Figure 8.2. Magnified plot of ZMP and CoM in sagittal axis during gait with 5 s stride time and 50 mm step length. The white area indicates PoS.

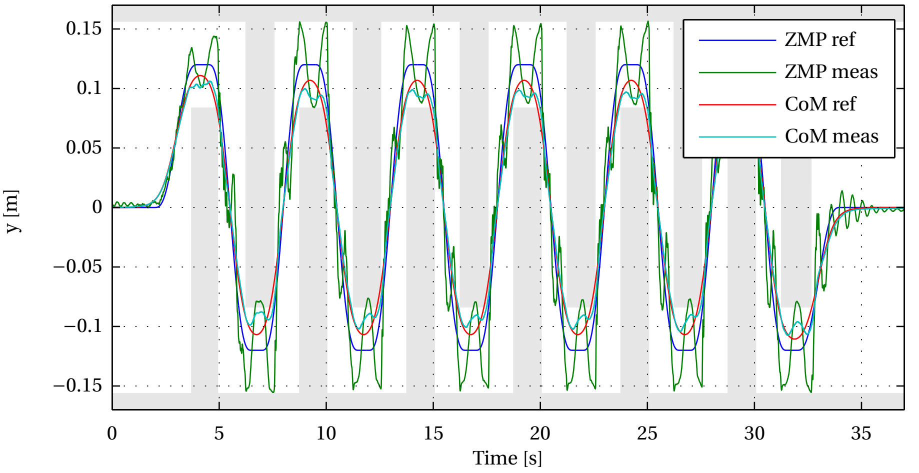


Figure 8.3. Plot of ZMP and CoM in transverse axis during gait with 5 s stride time and 50 mm step length. The white area indicates PoS.

ZMP tracking seems to be caused by poor CoM tracking in single support. The ankle CoM controller can not force the CoM to its reference and the robot falls slightly in over its foot. This causes the ZMP to drop as well. It should be noted that the shown CoM is compensated by the measured IMU angle such that the deflection of the FTS's in the ankles is taken into consideration. When the CoM starts to accelerate towards the opposite foot the ZMP is once again forced towards the outside of the foot. For this relatively slow stride time of 5 s, it is seen that the CoM almost stays within PoS and the gait remains almost static.

In figure 8.5 and 8.6 the ZMP and CoM has been plotted for a stride time of 3.5 s and a step length of 50 mm. In the sagittal axis the overall shape for the balance points is the same as for the slow stride time. There is however an important difference since the ZMP now leaves the PoS. As for the transverse axis in the previous test this indicates that the robot is only marginally dynamically stable. The main cause of this is assumed to be the reduced

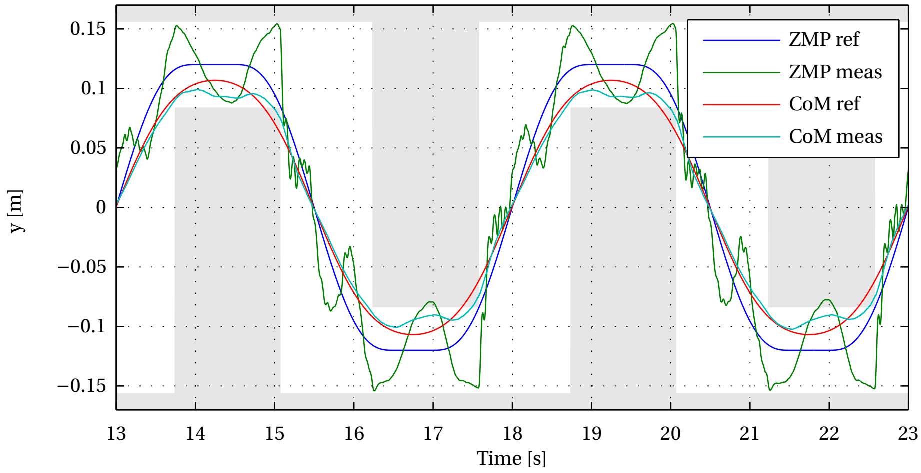


Figure 8.4. Magnified plot of ZMP and CoM in transverse axis during gait with 5 s stride time and 50 mm step length. The white area indicates PoS.

time in double support, where the oscillations can be reduced, and the increased velocity of the swinging foot. The CoM tracking is almost identical in the two tests and the gait remains static in the sagittal axis.

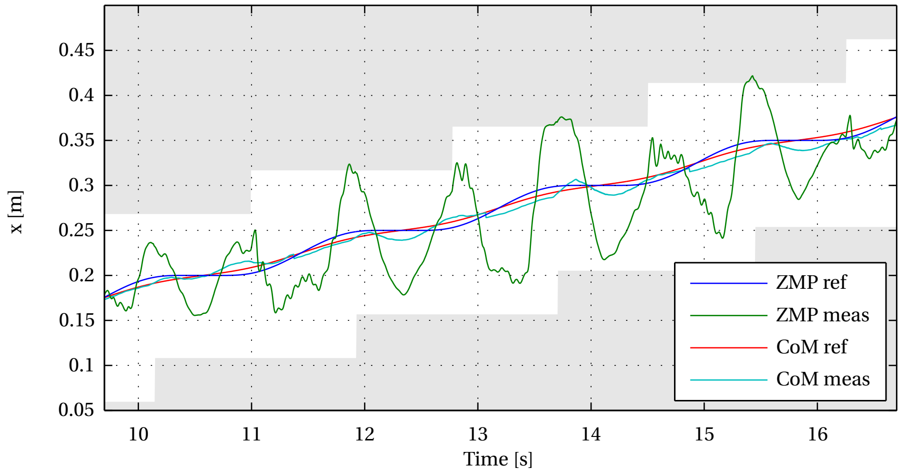


Figure 8.5. Magnified plot of ZMP and CoM in sagittal axis during gait with 3.5 s stride time and 50 mm step length. The white area indicates PoS.

For the transverse axis shown in figure 8.6 the difference in the two tests is more pronounced. The reduced single support time and amplitude of the reference means that the ankle controller has an easier job moving the CoM. In the two first steps the CoM actually overshoots its reference and moves the CoM further towards the outer edge of the foot. In the two last steps the CoM still falls towards the lifted foot. The changed behaviour of CoM results in a different ZMP during gait. In the two first steps the ZMP does not fall towards the inside of the foot as for the slow gait, but the increased CoM results in a ZMP closer to the outside of the foot. In contrast the two last steps show that the ZMP falls towards the inside of the foot too early. This is especially pronounced in the last step where it leads to an early impact of the foot, but the robot recovers and stays in balance.

In double support it is observed that the ZMP moves back an fourth between the feet rather than a pure transition from the rear to the front foot. This indicates that the effects of the foot impact have not been completely taken care of by the shock absorption.

The faster stride time results in less amplitude of the CoM. This leads to a bigger percentage of the gait cycle where the AAU-BOT1 is only dynamically stable, but not statically stable.

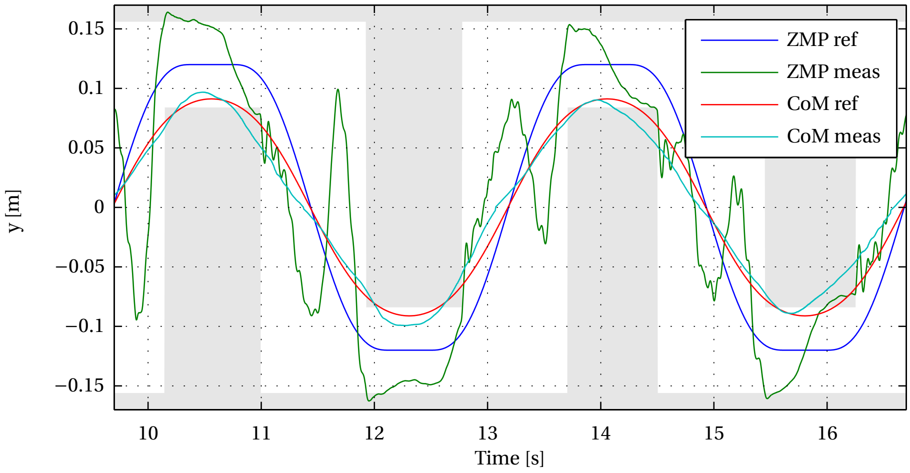


Figure 8.6. Magnified plot of ZMP and CoM in transverse axis during gait with 3.5 s stride time and 50 mm step length. The white area indicates PoS.

Generally it can be concluded that for a faster stride frequency the behaviour of each step varies more. For the slow gait the balance points move more consistently but create problems in terms of achieving the desired CoM in single support. It would be preferable if the fast gait could be performed more consistently and the stability margin could be increased.

In some of the experiments it was observed that the measured CoM gradually accumulated an error to the reference in the sagittal axis. The movement of the CoM was slightly less than specified by the reference. The reason for this is that sometimes the lifted foot impacts the ground earlier than anticipated. If this early impact happens while the foot is still moving forward, the robot will not take a full step. Over time this error accumulates and repeatedly creates a forward motion of the robot that is slightly less than desired. Since the length of each step is constant the robot can not catch up to its reference. A more dynamic definition of the step length would have to be implemented to cope with this problem.

The results of the two other step lengths with stride time 3.5 s will not be discussed in detail. As seen in table 8.1 the robot lost its balance during one of the experiments. This shows that 75 mm step length is the approximate maximum step length with the current controller strategy. The data of the two tests show much the same behaviour as the ones presented here.

The biggest difference is that for 75 mm step length, the steps vary more. In some of the single support phases the ZMP is placed very close to the outer edge of the foot, while in others the robot falls in over the foot resulting in an early impact of the swinging leg. Especially early impact creates problems when the swinging leg is still moving forward when it hits the ground.

The acceptance test has shown that the AAU-BOT1 is capable of walking. The gait is dynamic in the transverse axis but static in the sagittal axis. The tests have shown that while it is possible to walk with an arbitrary short step length, it is not possible to walk consistently with a step length of 75 mm or more.

CHAPTER 9

EPILOGUE

9.1 Conclusion

This thesis has treated the topic of making the AAU-BOT1 walk dynamically. Initially an analysis of human locomotion has been performed to gain insight in movement of limbs during gait and important gait measures. Different balance points were studied to be able to assess balance of the robot during gait. The construction of the AAU-BOT1 and the work of previous project groups were studied to identify the starting point of this project.

From this an overall approach to make the AAU-BOT1 walk was settled. A low level controller should make the joints follow a desired reference. A higher level controller should then have the responsibility of generating the joint references from a gait specification. Additionally it should modify the joint angle references online to maintain balance during gait. An experiment was carried out in a motion tracking room, to record characteristics of real human gait. This was later used as inspiration for design of the controller strategy.

To design the low level controller, a joint model for each separate joint was derived. The friction in the joints turned out to be the biggest challenge in the controller design. A significant amount of position dependent friction was identified to be caused by old stiff drive belts in the spur gears. A PID and cascade controller were designed and compared. In the end the cascade controller performed the best and this was selected as the structure of the low level controller.

To design the higher level controller different models were needed, both forward and inverse kinematics of the robot were derived. Using the forward kinematics the position of the AAU-BOT1 and its CoM in the global frame can be calculated from measured joint angles. Using the derived inverse kinematics the joint angles can be calculated for the legs by specifying a target location and orientation of the feet.

To determine the dynamic behaviour of the CoM of the AAU-BOT, the robot was modelled as a telescopic inverted pendulum in single support using the Linear Inverted Pendulum Mode (LiPM). In double support the dynamics of the AAU-BOT1 were modelled by combining two LiPM into a so called Linear Biped Model (LiBM). To model the relation between movement of CoM and ZMP a cart-table model was deduced. An extension to this model was made by including the moment of inertia of the legs. During experiments this model showed better correspondence between measured ZMP and CoM compared to the original cart-table model.

Using measurement of the ZMP from the FTS's placed in the ankles of the AAU-BOT1, the measurement of the CoM from the forward kinematics, and the dynamic models, an estimate of the ZMP and CoM were calculated using a Kalman filter.

To create the overall movement patterns to generate gait, it was chosen to specify a desired ZMP from a gait specification. It was then shown that preview control could be used to convert this ZMP reference into a CoM reference. This approach is used in both transverse and sagittal axis. To move the CoM to the reference it was suggested to use movement of the pelvis. For the transverse axis the upper body was used as well to move the CoM to avoid joint constraints.

When the weight of the robot had been shifted to one foot, the inverse kinematics were then used to put one foot in front of the other to obtain gait. It was observed that when the swinging foot landed on the ground a high force transient was experienced. This caused problems with stability. To reduce this problem a shock absorption strategy was implemented in the legs of the robot.

Finally a controller was used to close the loop around the system to obtain better CoM reference tracking. It was chosen to implement this by changing the ankle roll joint reference. Deflection in the FTS's in the ankles was observed in single support and the CoM was therefore not determined accurately by the forward kinematics alone. Instead the angle reading of the IMU was utilised to obtain a better estimate of the CoM.

From the acceptance test it is clear that the thesis objective stated in chapter 1.2 has been achieved using the designed controller strategy. It is possible to obtain dynamic gait with the AAU-BOT1 including gait initiation and termination. With the chosen approach gait initiation and termination is automatically obtained from the way that the CoM is calculated from the ZMP trajectory through preview control.

The results of the acceptance test however show that there is almost no stability margin during the gait. Especially for gait with step length of 75 mm the low stability margin is seen as the robot falls during one of the ten tests. By reducing the step lengths and thereby the motion of the swinging leg the mechanical oscillations are reduced hence a more stable gait is obtained and for step lengths of 50 mm and 25 mm the AAU-BOT1 were capable of completing all tests. From the acceptance test it was also shown that an increased stride time implies that the separate steps become more consistent and the stability margin increased, the gait however also becomes almost static.

The resemblance to human gait might seem vague at this point, but the principles at which this thesis is based can be extended to form more anthropomorphic gait. It has shown difficult to take recordings of human gait and directly translate this into gait with the AAU-BOT1. Instead the approach has been to make the robot walk using models of the robot and add the knowledge about human gait to define joint trajectories.

At this time the gait of the robot is almost static. If faster stride time can be obtained, the CoM will never enter PoS in the transverse axis during single support and the gait will become more dynamic. In the sagittal axis the gait is completely static due to the short step length, but if the stride length can be extended it will also be possible to make the CoM leave PoS for shorter periods of time hence make the gait dynamic in both axis.

While the gait velocity is still low compared to that of humans, this project has still been a big leap forward for the AAU-BOT1 project. What started out as a robot project which had only taken a single slowly step, has been turned into a walking biped robot.

9.2 Perspectives

The applications of walking robots are numerous. Even though the initial idea with the AAU-BOT1 was to be able to simulate strain on other joints in case of limb dysfunctionality, the potential of a walking biped robot is huge. Humans have shaped their surroundings such that most places can be reached by bipeds. Creating robots moving with biped locomotion is therefore more ideal than wheeled robots. Obstacles found everywhere around us such as stairs, doorsteps, and gaps are easier to pass for a biped than for a wheeled robot. Being able to move in an environment made for humans, the robots can start to take over some of the physical workload from the humans. This could be monotonous or heavy lifting tasks in the industry. Another possibility is to use the humanoid robots for tasks that should be executed in an unsafe environment e.g. rescue missions after earth quakes or radioactive areas.

For the AAU-BOT1 the near future is more focused towards a better gait implementation. Before the AAU-BOT1 can be used to anything different from walking, the gait must be more stable and more robust to disturbances. In the current implementation of the gait-balance controller, most of robot movement is calculated beforehand. This makes the gait more vulnerable to external disturbances. It should however be noted that the CoM trajectory generation could easily be implemented to run online. The preview controller only dependents on knowing the ZMP reference two seconds ahead of time. The movement of the feet, which is also generated offline, could as well be generated online. By measuring when the weight has been shifted to one foot, the movement of the opposite foot can be initiated at the appropriate time. This could make the gait less vulnerable to small deviations in the timing of the gait cycle. By adapting the step length to measured placement of the foot, it should be possible to reduce the problem that the CoM gets slightly behind its reference in the sagittal axis.

The main limitation of the current implementation is that the AAU-BOT1 is not able to perform consistent steps when the step length becomes too long. More work should be put into designing CoM and ZMP control. The biggest problem at this time is that the mechanical oscillation of the robot it not reduced sufficiently. The first step should therefore be to reduce these. This could be done through active damping in the joints. A possible implementation of this could be to insert an active spring-damper system similar to what has been implemented in the legs. This damper should act on moments measured in the FTS's and try to damp the oscillations. By implementing this in the ankles a small amount of additional flexibility will introduced in these joints. This could be beneficial to the ground contact since the ankle would have the possibility to give in and let the foot stand flat on the ground. This will however be left for coming project groups to look into.

If tight ZMP tracking can be obtained by further development of the ZMP controller, the gait velocity can be increased. Good tracking of the ZMP allows for more advanced ZMP trajectories, where the ZMP moves from the heel to the toe during single support. In this way it is possible to make a human-like heel off when the ZMP is under the toe of the supporting foot. This allows longer steps and faster gait. Longer steps can also be achieved by looking more into anthropomorphic movements where e.g. pelvis rotation, about the vertical axis is be utilised. Rotation of the pelvis allows for longer step lengths without decreasing the pelvis height in the initial position.

From the previous group working on the AAU-BOT1 a hint was given though a bug report, that errors in some of the drivers exists [2]. In the beginning of this project these errors were not observed and no development time was put into updating the drivers. During the last three months of the project period, the errors have started to show as more experiments of gait were conducted. In the current driver implementation an error in the driver makes the system go into error state and the robot stops all motion. This is very critical, hence driver update is deemed necessary for future success. The entire system could benefit from a complete system update since old versions of MATLAB® and Linux is running on the machines. The outdated MATLAB® means that some functions are not available and a number of bugs exist.

With the current state of the controller and driver implementation the AAU-BOT1 it is not capable of performing tasks away from the controlled environment in the laboratory. However by further development of the controllers, update of the hardware drivers, and by installing a battery pack, the AAU-BOT1 should be ready to detach from its support frame and walk freely.

ACRONYMS

API	Application Programming Interface
CAN	Controller Area Network
CoM	Centre of Mass
CoP	Centre of Pressure
CPG	Central Pattern Generator
DGM	Dynamic Gait Measure
DoF	Degrees of Freedom
FTS	Force Torque Sensor
FZMP	Fictitious Zero Moment Point
GCoM	Ground Projected Centre of Mass
IMU	Inertial Measurement Unit
MPC	Model Predictive Control
LiBM	Linear Biped Model
LiPM	Linear Inverted Pendulum Mode
OBC	Onboard Computer
PoS	Polygon of Support
RMSD	Root Mean Square Deviation
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
ssh	Secure Shell
ZMP	Zero Moment Point

Bibliography

[1] (2010, aug) Aalborg University. Accessed: Dec 2012. [Online]. Available: http://www.aaubot.aau.dk/

[2] J. Andersen and H. D. Christensen, "One step closer; towards dynamic walking on aau-bot1," Institute of Electronic Systems at Aalborg University, Tech. Rep., jun 2010.

[3] M. M. Pedersen, A. A. Nielsen, and L. F. Christiansen, "Design of biped robot aau-bot1," Institute of Mechanical Engineering at Aalborg University, Tech. Rep., jun 2007.

[4] P. K. Jensen, M. Garbus, and J. V. Knudsen, "Instrumentation, modeling and control of aau-bot1," Institute of Electronic Systems at Aalborg University, Tech. Rep., jun 2008.

[5] B. T. Jensen and M. O. K. Niss, "Modeling, simulation, and control of biped robot aau-bot1," Institute of Electronic Systems at Aalborg University, Tech. Rep., jun 2009.

[6] M. Vukobratovic, B. Borovac, and V. Potkonjak, "Towards a unified understanding of basic notions and terms in humanoid robotics," *Robotica*, vol. 25, no. 1, pp. 87–102, 2007.

[7] C. Vaughan, B. Davis, and J. O'connor, *Dynamics of human gait.* Human Kinetics Publishers Champaign, Illinois, 1992.

[8] D. Wollherr, *Design and control aspects of humanoid walking robots.* VDI-Verlag, 2005.

[9] R. Schafer, *Clinical biomechanics: musculoskeletal actions and reactions.* Williams & Wilkins, 1987.

[10] R. Serway and J. Jewett, *Physics for Scientists and Engineers, with Modern Physics.* Thomson-Brooks/Cole, 2004.

[11] C. Mummolo and J. H. Kim, "How dynamic is dynamic walking? human vs. robotic gait," *ASME Conference Proceedings*, vol. 2011, no. 54839, pp. 1159–1165, 2011.

[12] A. Goswami, "Postural stability of biped robots and the foot-rotation indicator (fri) point," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 523–533, 1999.

[13] M. Popovic, A. Goswami, and H. Herr, "Ground reference points in legged locomotion: Definitions, biological trajectories and control implications," *The International Journal of Robotics Research*, vol. 24, no. 12, pp. 1013–1032, 2005.

[14] maxon motor. (2010) maxon motor data and operating ranges. Accessed: May 2013. [Online]. Available: http://www.maxonmotor.com/medias/sys_master/8798985748510.pdf

[15] Rtai - the realtime application interface for linux from diapm. Accessed: May 2013. [Online]. Available: https://www.rtai.org/

[16] B. Moris, "Modeling for control of a biped robot - aau-bot1, another step closer towards dynamic walking," Department of Mechanical Engineering at Eindhoven University of Technology, Tech. Rep., dec 2012.

[17] G. T. Fallis. (1888) Walking toy. [Online]. Available: http://ruina.tam.cornell.edu/research/history/1888_fallis.pdf"

[18] S. Collins, M. Wisse, and A. Ruina, "A three-dimensional passive-dynamic walking robot with two legs and knees," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 607–615, 2001.

[19] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Proceedings of the 2003 IEEE Internation Conference on Robotics & Automation.* IEEE, 2003.

[20] B. Stephens and C. Atkeson, "Modeling and control of periodic humanoid balance using the linear biped model," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, dec. 2009, pp. 379 –384.

[21] A. Ijspeert, "2008 special issue: Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.

[22] S. Mondal, A. Nandy, P. Chakraborty, and G. Nandi, "A central pattern generator based nonlinear controller to simulate biped locomotion with a stable human gait oscillation," *International Journal of Robotics and Automation (IJRA)*, vol. 2, no. 2, pp. 93–106, 2011.

[23] J. Reblo, "Study and simulation of humanoid robot walking algorithms," Master's thesis, university of porto, 2010.

[24] W. Yang, N. Chong, C. Kim, and B. You, "Self-adapting humanoid locomotion using a neural oscillator network," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on.* IEEE, 2007, pp. 309–316.

[25] B. Bona and M. Indri, "Friction compensation in robotics: an overview," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on.* IEEE, 2005, pp. 4360–4367.

[26] P. Gandhi, F. Ghorbel, and J. Dabney, "Modeling, identification, and compensation of friction in harmonic drives," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 1. IEEE, 2002, pp. 160–166.

[27] J. Hauschild, G. Heppler, and J. McPhee, “Friction compensation of harmonic drive actuators,” in *Proceedings of 6th International Conference on Dynamics and Control of Systems and Structures in Space*, 2004, pp. 683–692.

[28] H. Olsson, K. Åström, C. Canudas de Wit, M. Gäfvert, and P. Lischinsky, “Friction models and friction compensation,” *European journal of control*, vol. 4, pp. 176–195, 1998.

[29] J. Craig, *Introduction to Robotics: Mechanics and Control*, ser. Addison-Wesley series in electrical and computer engineering: Control engineering. Pearson/Prentice Hall, 2005.

[30] M. Ali, H. Park, and C. Lee, “Closed-form inverse kinematic joint solution for humanoid robots,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 704–709.

[31] S. Kajita and K. Tani, “Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode,” in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, 1991, pp. 1405–1411.

[32] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, “The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation,” in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2001, pp. 239–246.

[33] B. Stephens and C. Atkeson, “Modeling and control of periodic humanoid balance using the linear biped model,” in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*. IEEE, 2009, pp. 379–384.

[34] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2008.

[35] C. M. Close, D. K. Frederick, and J. C. Newell, *Modelling and Analysis of Dynamic Systems*. Wiley, 2002.

[36] G. Welch and G. Bishop, “An introduction to the kalman filter,” 1995.

[37] I. R. Manchester, U. Mettin, F. Iida, and R. Tedrake, “Stable dynamic walking over uneven terrain,” *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 265–279, 2011.

[38] W. East and B. Lantz. (2005, aug) Notch filter design. Accessed: Mar 2013. [Online]. Available: http://www.ligo.caltech.edu/docs/T/T050162-00.pdf

[39] D. A. Winter, “Human balance and posture control during standing and walking,” *Gait & Posture*, vol. 3, no. 4, pp. 193–214, 1995.

[40] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, “A realtime pattern generator for biped walking,” in *Robotics and Automation, 2002. Proceedings. ICRA’02. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 31–37.

[41] T. Katayama, T. Ohki, T. Inoue, and T. Kato, “Design of an optimal controller for a discrete-time system subject to previewable demand,” *International Journal of Control*, vol. 41, no. 3, pp. 677–699, 1985.

[42] J.-Y. Kim, J. Lee, and J.-H. Ho, “Experimental realization of dynamic walking for a human-riding biped robot, hubo fx-1,” *Advanced Robotics*, vol. 21, no. 3-4, pp. 461–484, 2007.

[43] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*. John Wiley, 2005.

[44] H. Dallali, M. Brown, and B. Vanderborght, “Using the torso to compensate for non-minimum phase behaviour in zmp bipedal walking,” in *Advances in Robotics Research*, T. Kröger and F. Wahl, Eds. Springer Berlin Heidelberg, 2009, pp. 191–202. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01213-6_18

[45] J. Awrejcewicz, *Classical Mechanics*. Springer, 2012. [Online]. Available: http://books.google.dk/books?id=W-ljhVWpImoC

Appendix

Appendix A

AAU-BOT1 - Technical information

This chapter contains a collection of tables and figures containing essential technical information about the AAU-BOT1. This appendix is meant to be used as reference sheet for looking up specification concerning the AAU-BOT1.

A.1 Instrumentation

Gears, Motors, Motor Drivers, and Sensors in the Joints

In this section the make and model is specified of the hardware used in the joints. First the model of all hardware is specified. For the Motor and HDG the type is different from joint to joint, and the type used in each joint is specified in table A.1.

- Motor: *Maxon RE series*
- Motor Driver: *Maxon EPOS 70/10*
- HDG: *Harmonic Drive AG CPU-S series*
- Hall Sensor: *Honeywell HRS100SSAB090*
- Encoder:*Maxon motor encoder MR Type L, 512 CPT 3 Channel*

Joint(s)	Motor	HDG	HDG ratio	Spur gear ratio	Total gear ratio
Shoulder	60 W	CPU-S 14-100	100	20/16	125
Waist Roll	90 W	CPU-S 14-100	100	57/19	300
Waist Pitch	150 W	CPU-S 17-120	120	57/19	360
Pelvis Yaw	60 W	CPU-S 17-100	100	46/16	288
Hip Roll	2x150 W	CPU-S 20-120	120	69/30	276
Hip Pitch	150 W	CPU-S 20-120	120	57/28	244
Hip yaw	90 W	CPU-S 17-100	100	40/19	211
Knee	2x150 W	CPU-S 17-100	100	52/39	133
Ankle Roll	60 W	CPU-S 14-100	100	44/22	200
Ankle Pitch	2x150 W	CPU-S 20-160	160	35/18	311

Table A.1. Specification of motors and gears for the different joints in the robot.

IMU

The IMU attached to the AAU-BOT1 is: *Xsens MTi - miniature gyro-enhanced Attitude and Heading Reference.*

FTS and FTS Amplifier

The FTS is special made for the AAU-BOT1 by [3]. The amplifier used to amplify the signals from the strain-gauges in the FTS is of type: Lorenz Messtechnik SI-RS4852-1K (with two channels)[1].

[1] Refer to the datasheet on the enclosed DVD

A.2 Joint Limitations and Orientation

This section describes the angle limitation of all the individual joints in the AAU-BOT1. Most of the joints have a limited operating range. The most common cause is collision of different parts of the robot if these limitations are exceeded. In table A.2 the physical limitations are stated for both positive and negative direction of turn, together with the cause of the limitation. The notation (Foot bracket, HDG) means that the foot bracket collides with the harmonic drive gear. The arms can not collide with anything and are therefore able to rotate freely.

To protect the robot an angle limitation is implemented in the posture controller. No matter what angle is input to the posture controller the joint reference for each joint will never exceed the limitations specified in table A.2, defined in the two software limitation columns. It should however be noted that this limitation does not prevent the individual limbs of the robot to collide during gait.

The orientation column specifies whether the reference angle is inverted in the posture controller. This is done to give a consistent definition of what a positive and negative angle reference on a joint corresponds to. After the change of signs a positive angle reference corresponds to a counter clockwise rotation about each joints local coordinate system.

Joint number & name		Joint pos.	Joint neg.	SW pos.	SW neg.	Pos. limit reason	Neg. limit reason	Orientation
1	R. Ankle Roll	12.5°	−46.6°	10.8°	−31.9°	Foot bracket, HDG	FTS, Potmeter	−1
2	R. Ankle Pitch	34.5°	−28.6°	32.8°	−26.0°	Foot bracket, Shin	Potmeter, Shin	−1
3	R. Knee Pitch	74.9°	−9.1°	60.0°	−6.0°	Epos, Epos	Internal Potmeter	1
4	R. Hip Pitch	17.6°	−45.2°	16.4°	−40.2°	Thigh, HDG	Potmeter, Bracket	−1
5	R. Hip Roll	26.9°	−16.4°	25.5°	−14.7°	Potmeter, Pelvis	HDG, Bracket	1
6	R. Hip Yaw	147.6°	−28.0°	139.2°	−26.6°	Motor, Pelvis	Belt, Belt	−1
7	L. Hip Yaw	29.1°	−146.6°	26.6°	−139.2°	Belt, Belt	Motor, Pelvis	−1
8	L. Hip Roll	15.5°	−28.5°	14.7°	−25.5°	HDG, Bracket	Potmeter, Pelvis	1
9	L. Hip Pitch	17.3°	−42.3°	16.4°	−40.2°	Thigh, HDG	Potmeter, Bracket	1
10	L. Knee Pitch	73.2°	−17.3°	60.0°	−6.0°	Epos, Epos	Internal Potmeter	−1
11	L. Ankle Pitch	36.8°	−28.5°	32.8°	−26.0°	Foot bracket, Shin	Potmeter, Shin	1
12	L. Ankle Roll	33.6°	−11.4°	31.9°	−10.8°	FTS, Potmeter	Foot bracket, HDG	−1
13	Pelvis Yaw	59.7°	−82.3°	56.7°	−78.1°	Motor, Bracket	Coord length	−1
14	Waist Pitch	24.3°	−35.9°	23.1°	−34.0°	Max. motor torque	Max. motor torque	−1
15	Waist Roll	31.1°	−22.2°	29.6°	−21.1°	Max. motor torque	Bracket, Torso	1
16	R. Arm Pitch	∞	−∞	∞	−∞	–	–	1
17	L. Arm Pitch	∞	−∞	∞	−∞	–	–	−1

Table A.2. Angular limitation of all the joints in the AAU-BOT1.

A.3 Joint Position and COM Vectors

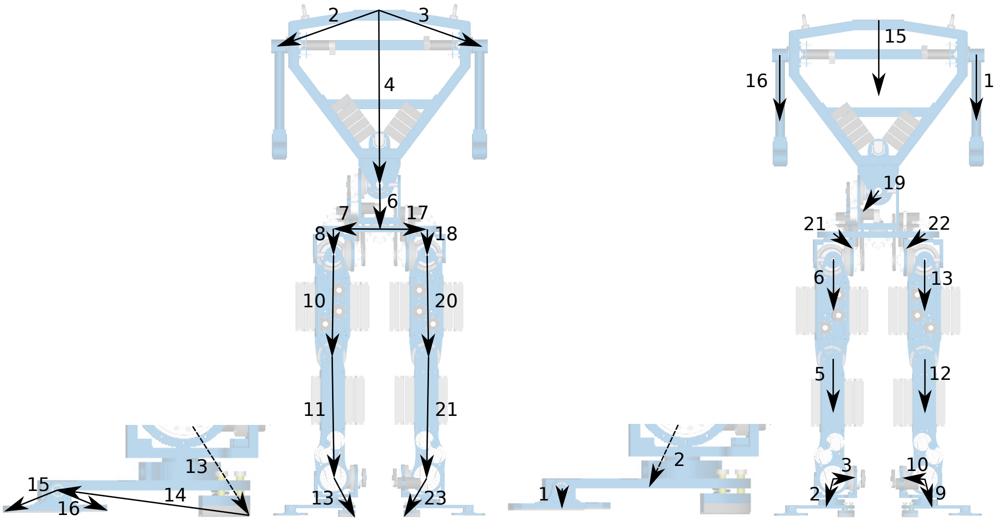
This section includes the vectors describing the dimensions of the AAU-BOT1 which are used in the forward and inverse kinematics.

Index	Joint position [m]	Joint name	Parent joint	Joint number
1	$[- \quad - \quad -]^T$	IMU	*World*	(Pitch, Yaw, Roll)
2	$[0.048 \quad -0.293 \quad -0.136]^T$	Right Shoulder	IMU	16 (Pitch)
3	$[0.048 \quad 0.317 \quad -0.136]^T$	Left Shoulder	IMU	17 (Pitch)
4	$[0.043 \quad 0.012 \quad -0.555]^T$	Waist Roll	IMU	15 (Roll)
5	$[0.000 \quad 0.000 \quad 0.000]^T$	Waist Pitch	Waist Roll	14 (Pitch)
6	$[0.000 \quad 0.000 \quad -0.150]^T$	Pelvis Yaw	Waist Pitch	13 (Yaw)
7	$[0.000 \quad -0.140 \quad 0.000]^T$	Right Hip Yaw	Pelvis Yaw	6 (Yaw)
8	$[0.000 \quad 0.000 \quad -0.064]^T$	Right Hip Roll	Right Hip Yaw	5 (Roll)
9	$[0.000 \quad 0.000 \quad 0.000]^T$	Right Hip Pitch	Right Hip Roll	4 (Pitch)
10	$[0.000 \quad -0.012 \quad -0.310]^T$	Right Knee	Right Hip Pitch	3 (Pitch)
11	$[0.000 \quad 0.000 \quad -0.370]^T$	Right Ankle Pitch	Right Knee	2 (Pitch)
12	$[0.000 \quad 0.000 \quad 0.000]^T$	Right Ankle Roll	Right Ankle Pitch	1 (Roll)
13	$[-0.064 \quad -0.003 \quad -0.114]^T$	Right Heel	Right Ankle Roll	-
14	$[0.192 \quad 0.003 \quad 0.021]^T$	Right Toe Joint	Right Heel	-
15	$[0.053 \quad -0.003 \quad -0.021]^T$	Right Toe Tip	Right Toe Joint	-
16	$[-0.047 \quad -0.003 \quad -0.021]^T$	Right Meta Phalanx	Right Toe Joint	-
17	$[0.000 \quad 0.140 \quad 0.000]^T$	Left Hip Yaw	Pelvis Yaw	7 (Yaw)
18	$[0.000 \quad 0.000 \quad -0.064]^T$	Left Hip Roll	Left Hip Yaw	8 (Roll)
19	$[0.000 \quad 0.000 \quad 0.000]^T$	Left Hip Pitch	Left Hip Roll	9 (Pitch)
20	$[0.000 \quad 0.012 \quad -0.310]^T$	Left Knee	Left Hip Pitch	10 (Pitch)
21	$[0.000 \quad 0.000 \quad -0.370]^T$	Left Ankle Pitch	Left Knee	11 (Pitch)
22	$[0.000 \quad 0.000 \quad 0.000]^T$	Left Ankle Roll	Left Ankle Pitch	12 (Roll)
23	$[-0.064 \quad -0.000 \quad -0.114]^T$	Left Heel	Left Ankle Roll	-
24	$[0.192 \quad 0.000 \quad 0.021]^T$	Left Toe Joint	Left Heel	-
25	$[0.053 \quad -0.000 \quad -0.021]^T$	Left Toe Tip	Left Toe Joint	-
26	$[-0.047 \quad -0.000 \quad -0.021]^T$	Left Meta Phalanx	Left Toe Joint	-

Table A.3. Positions of joint frames in the AAU-BOT1 with respect to the parent joint frame. The horizontal lines separate the joints into upper body joints, right leg joints, and left leg joints. The *World frame* is included to illustrate that the robot is rotated and translated with respect to a frame fixed to the ground surface. The rotation is determined by the IMU while no measurement is available for determining the translation.

Index	CoM position [m]	mass [kg]	Link name	Parent joint
1	$[0.003\ \ -0.003\ \ -0.015]^T$	0.172	Right Toe	Right Toe
2	$[0.031\ \ -0.003\ \ -0.090]^T$	0.802	Right Heel	Right Ankle Roll
3	$[-0.048\ \ -0.018\ \ -0.014]^T$	1.515	Right Ankle Bracket	Right Ankle Roll
4	$[-0.003\ \ 0.005\ \ -0.000]^T$	0.470	Right Ankle Cross Axle	Right Ankle Pitch
5	$[0.001\ \ 0.031\ \ -0.246]^T$	5.275	Right Shin	Right Knee
6	$[0.002\ \ -0.047\ \ -0.144]^T$	6.656	Right Thigh	Right Hip Pitch
7	$[0.001\ \ -0.008\ \ -0.000]^T$	0.640	Right Hip Cross Axle	Right Hip Roll
8	$[-0.000\ \ 0.000\ \ -0.015]^T$	0.172	Left Toe	Left Toe
9	$[0.028\ \ -0.000\ \ -0.090]^T$	0.802	Left Heel	Left Ankle Roll
10	$[-0.052\ \ -0.017\ \ -0.014]^T$	1.500	Left Ankle Bracket	Left Ankle Roll)
11	$[-0.006\ \ -0.006\ \ -0.000]^T$	0.470	Left Ankle Cross Axle	Left Ankle Pitch
12	$[-0.002\ \ -0.032\ \ -0.246]^T$	5.279	Left Shin	Left Knee
13	$[0.000\ \ 0.045\ \ -0.144]^T$	6.656	Left Thigh	Left Hip Pitch
14	$[0.001\ \ 0.007\ \ -0.000]^T$	0.640	Left Hip Cross Axle	Left Hip Roll
15	$[0.042\ \ 0.001\ \ -0.275]^T$	11.618	Torso	IMU
16	$[0.104\ \ -0.005\ \ -0.245]^T$	0.906	Right Arm	Right Shoulder
17	$[0.104\ \ -0.005\ \ -0.231]^T$	0.906	Left Arm	Left Shoulder
18	$[-0.005\ \ -0.004\ \ -0.000]^T$	0.754	Waist Cross Axle	Waist Roll
19	$[-0.004\ \ -0.039\ \ -0.041]^T$	2.167	Waist Bracket	Waist Pitch
20	$[-0.022\ \ -0.001\ \ 0.016]^T$	4.689	Pelvis	Pelvis Yaw
21	$[-0.047\ \ -0.029\ \ -0.021]^T$	3.052	Right Hip Bracket	Right Hip Yaw
22	$[-0.047\ \ 0.027\ \ -0.021]^T$	3.052	Left Hip Bracket	Left Hip Yaw

Table A.4. CoM positions in a specified parent joint frame and mass of the links in the AAU-BOT1.



(a) Joint Positions.

(b) CoM Positions.

Figure A.1. Positions of the joints and CoM of links. The numbers on the vectors in figure (a) and figure (b) are the indexes shown in table A.3 and A.4 respectively. In the figure short vectors are not shown. It should be noted that the legs are rotated 90 deg to enhance visibility, and the positive x-direction of the legs are pointing in the same way as the toes. (A copy of figure 4.10 in the main report)

Appendix B

FTS Calibration

This appendix describes the test conducted to calibrate the FTS sensors placed in each foot of the AAU-BOT1. The goal is to determine a linear calibration that can be used to transform the raw sensor readings into forces and moments acting in origin of the FTS coordinate system. This system has origin in the middle of the FTS. For a further information about the design and sensors of the FTS refer to [3].

The overall goal of this calibration is to find the linear calibration matrix **C** that maps the measurement vector $\mathbf{v} = [v_{s1}\ v_{b1}\ v_{s2}\ v_{b2}\ v_{s3}\ v_{b3}]^T \in \mathbb{R}^6$ into calibrated forces and moment data $\mathbf{s} = [F_x\ F_y\ F_z\ M_x\ M_y\ M_z]^T \in \mathbb{R}^6$. Where v_{s1}, v_{s2} and v_{s3} are the outputs from shearing force sensors in the FTS and v_{b1}, v_{b2} and v_{b3} are sensor readings from three bending force sensors in the FTS. F_x, F_y, and F_z together with M_x, M_y, and M_z represents forces and moments respectively in the FTS frame. Formally the calibration can be written as:

$$\mathbf{s} = \mathbf{C}\mathbf{v} \tag{B.1}$$

where:

$\mathbf{C} \in \mathbb{R}^{6\times 6}$ is the calibration matrix

To find the calibration matrix **C** a number of different forces are applied to the FTS and the resulting sensor readings are saved. In theory six different readings are sufficient to determine a calibration. But to minimise the influence of measurement uncertainties and other disturbances, multiple forces are applied to determine a calibration matrix which will minimise the mean square error of all the readings. This is done by imposing the property of the pseudo inverse of a matrix and equation B.1 can be written into equation B.2.

$$\mathbf{C} = \mathbf{s}\mathbf{v}^T(\mathbf{v}\mathbf{v}^T)^{-1} \tag{B.2}$$

where:

$\mathbf{s} \in \mathbb{R}^{6\times N}$ is a matrix with N calculated data sets of forces and moments

$\mathbf{v} \in \mathbb{R}^{6\times N}$ is a matrix with N sensor data sets

Calculation of applied forces and moments

To apply the different forces needed to calibrate the FTS, a test rig designed by [4] is used. This rig can be seen in figure B.1. The FTS is unmounted from the foot of the AAU-BOT1 and instead mounted in the test frame as indicated on the drawing as a red unit. The FTS is fastened tightly to both the frame and the beam such that it can not move or turn during the experiments. By tightening or loosening the cable tensioner a wide range of forces can be applied to the pull point (**pp**) and through the beam affect the sensors of the FTS.

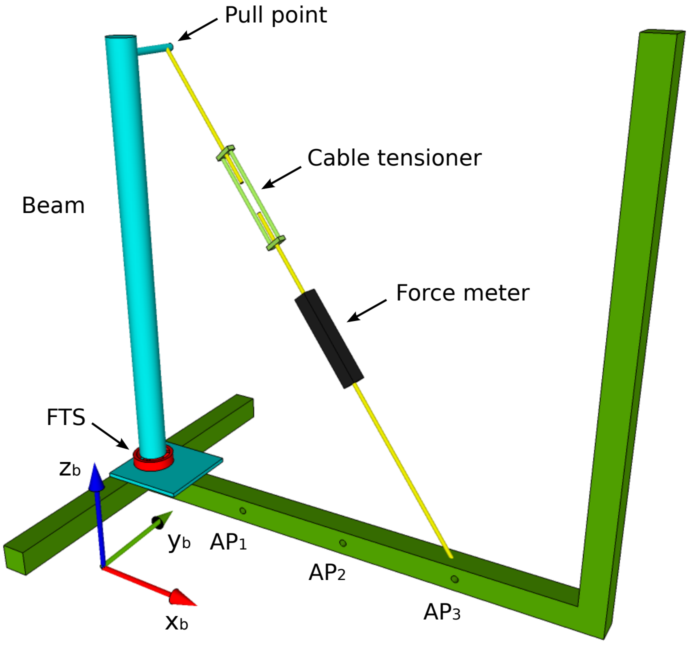


Figure B.1. A drawing of the test frame used to apply different loads to the FTS.

In order to ensure a good calibration, forces should be applied in different ways to test different combinations of forces and moments. In practice this is done in two ways. The first way is to pull with different angles from **pp** by anchoring the other end of the wire in different anchor points ($\mathbf{AP}_1$, $\mathbf{AP}_2$, and $\mathbf{AP}_3$). The other way is to turn the orientation of the FTS with respect to the base frame (x_b; y_b; z_b) that always points as indicated in figure B.1. It should be noted that origin of the base frame has been offset in the figure to enhance visibility. The origin of this coordinate system is in the middle of the FTS as indicated in figure B.2. The rotation of the FTS with respect to the base frame is indicated by θ_z.

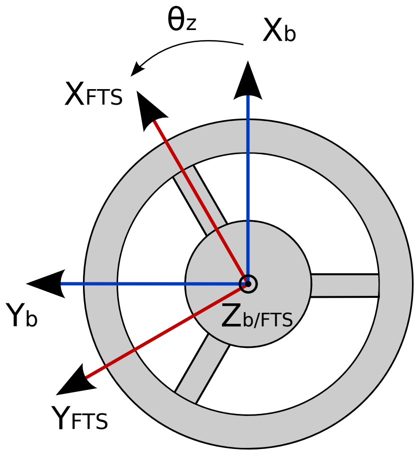


Figure B.2. A drawing of the FTS frame $(x_{FTS}, y_{FTS}, z_{FTS})$ with respect to the base frame (x_b, y_b, z_b).

Since the desired output of **Cv** in equation B.1 is a vector with forces and moments in the FTS frame, the applied forces to the beam must be transformed into forces and moments in this frame. This procedure for doing this is described next.

The overall approach is to divide the applied force into two separate contributions. One which is translated into forces in the FTS frame and one that is translated into moments. The force contribution is found simply by translating the force acting on **pp** to origin of the FTS frame [45]. This is done by multiplying the magnitude of the force given by the force meter with the unit direction vector from **pp** towards one of the three anchor points as seen in equation B.3.

$$\mathbf{F}_P = |\mathbf{F}_P| \frac{\mathbf{AP} - \mathbf{pp}}{|\mathbf{AP} - \mathbf{pp}|} \qquad [\mathrm{N}] \quad (\mathrm{B.3})$$

where:

$\mathbf{F}_P \in \mathbb{R}^3$ is the resulting forces in the base frame [N]

$\mathbf{AP} \in \mathbb{R}^3$ is the respective anchor point [m]

In a 3-dimensional space the cross product between a force vector and a vector denoted $\mathbf{b} \in \mathbb{R}^3$ pointing from a point $\mathbf{p} \in \mathbb{R}^3$ towards a the point of attack of the force vector, returns the moments about the point **p** [45]. This is exploited in equation B.4 to calculate the moments created by $\mathbf{F}_P$ about origin in the base frame.

$$\mathbf{M} = \mathbf{b} \times \mathbf{F}_P \qquad [\mathrm{Nm}] \quad (\mathrm{B.4})$$

where:

$\mathbf{M} \in \mathbb{R}^3$ is a vector that contains the moments around its three axis [Nm]

As described earlier the FTS can be rotated in different angles with respect to the base frame. To change the forces and moments from the base to the FTS frame, the vectors are rotated around the z-axis with the following rotation matrix:

$$\mathbf{R}_z(\theta_z) = \begin{bmatrix} cos\,\theta_z & -sin\,\theta_z & 0 \\ sin\,\theta_z & cos\,\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.5}$$

where:

θ_z is the rotation angle about the z-axis (B.6)

The vector **s** containing forces and moments with respect to the FTS frame can now be defined as follows:

$$\mathbf{s} = \begin{bmatrix} \mathbf{F}_{FTS} \\ \mathbf{M}_{FTS} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_z(\theta_z)\mathbf{F}_P \\ \mathbf{R}_z(\theta_z)\mathbf{M} \end{bmatrix} \tag{B.7}$$

where:

$\mathbf{F}_{FTS} \in \mathbb{R}^3$is forces in the FTS frame

$\mathbf{M}_{FTS} \in \mathbb{R}^3$is moments in the FTS frame

As described earlier the **s** matrix in equation B.2 is expanded to dimension $6 \times N$ for N different applied forces to find the calibration matrix **C** from B.2.

B.1 Test Procedure

For each of the following tests, the recorded data is logged by utilising the existing FTS to OBC interface. Recorded data is streamed from the OBC to a local desktop computer which saves the data in a file. For each test the six sensor data readings are sampled for 10 s in order to allow meaning of the data. Meaning the data over time reduces the influence of noise on the readings.

After initial tests with high strain on the FTS, it was observed that the readings from the FTS did not return to the same steady state value as before the strain. In order to compensate for this, a zero reading without any load was performed before each test with a new orientation of either beam or FTS, to be able to subtract this offset. This are the readings denoted with 0 kg.

In table B.1 all the different tests that were performed are listed. Each test is denoted with a number, an orientation of the FTS with respect to the frame, an orientation of the beam with respect to the FTS, a number that indicates which anchor point was pulled from, and finally the weight measured by the force meter on the right and left FTS.

Test no.	FTS angle (θ_z) [deg]	Beam angle [deg]	Anchor point [1,2,3]	Weight (right) [kg]	Weight (left) [kg]
1	120	148	1	0.00	0.00
2	120	148	1	4.90	5.15
3	120	148	1	10.60	9.95
4	120	148	1	20.40	20.70
5	120	148	1	40.40	39.70
6	120	148	3	0.00	0.00
7	120	148	3	5.00	4.85
8	120	148	3	10.40	10.15
9	120	148	3	19.90	19.90
10	120	148	3	40.05	40.20
11	240	60	1	0.00	0.00
12	240	60	1	5.00	4.90
13	240	60	1	10.10	10.50
14	240	60	1	20.10	19.90
15	240	60	1	40.50	40.10
16	240	60	3	0.00	0.00
17	240	60	3	5.00	5.00
18	240	60	3	10.30	9.95
19	240	60	3	20.30	20.30
20	240	60	3	40.20	40.20
21	0	60	1	0.00	0.00
22	0	60	1	5.10	5.00
23	0	60	1	10.00	10.15
24	0	60	1	20.10	20.20
25	0	60	1	40.00	40.10
26	0	60	3	0.00	0.00
27	0	60	3	5.20	5.05
28	0	60	3	10.05	10.05
29	0	60	3	20.00	20.10
30	0	60	3	40.30	39.70

Table B.1. Test procedures for collecting calibration data.

In order to verify that the calibration results in a reasonable result, a verification is performed as well. The verification data is collected at different angles and weights compared to the calibration, to check that the calibration is indeed valid at other angles.

Test no.	FTS angle (θ_z) [deg]	Beam angle [deg]	Anchor point [1,2,3]	Weight (right) [kg]	Weight (left) [kg]
31	320	60	2	0.00	0.00
32	320	60	2	4.90	4.90
33	320	60	2	15.50	14.90
34	320	60	2	25.10	25.20
35	320	60	2	34.80	35.20

Table B.2. Test procedures for collecting verification data.

B.2 Remarks

After tension on the cable tensioner had been adjusted to a specific load, the reading on the digital weight was observed to decrease slowly over time. This was mostly observed during large tension. The cause of this is assumed to be that the beams in the FTS slowly deforms over time when exposed to high strain. In order to reduce this influence, the calibration rig was left untouched for a short while before each reading was made. This gave more steady readings, since most of the deformation happened shortly after the force had been applied.

B.3 Results and assessments

Using the above mentioned method the two calibration matrices have been identified to be as defined in (B.8) and (B.9).

$$\mathbf{C}_{\text{right}} = \begin{bmatrix} -0.0006 & -0.0052 & 0.0125 & 0.0060 & -0.0081 & -0.0069 \\ 0.0080 & -0.0035 & 0.0172 & 0.0029 & 0.0252 & -0.0018 \\ 0.2031 & -0.0033 & 0.2037 & -0.0012 & 0.2014 & 0.0085 \\ 0.0066 & 0.0011 & 0.0004 & -0.0009 & -0.0052 & 0.0006 \\ -0.0039 & 0.0003 & 0.0067 & -0.0002 & -0.0035 & -0.0004 \\ -0.0004 & -0.0006 & -0.0013 & -0.0009 & -0.0005 & -0.0016 \end{bmatrix} \tag{B.8}$$

$$\mathbf{C}_{\text{left}} = \begin{bmatrix} 0.0001 & 0.0030 & 0.0141 & -0.0027 & -0.0018 & -0.0112 \\ 0.0078 & 0.0002 & 0.0163 & 0.0015 & 0.0266 & -0.0005 \\ 0.2053 & 0.0119 & 0.1946 & 0.0109 & 0.2000 & -0.0038 \\ 0.0068 & 0.0009 & 0.0005 & -0.0006 & -0.0052 & -0.0014 \\ -0.0037 & 0.0000 & 0.0073 & -0.0003 & -0.0034 & -0.0010 \\ 0.0002 & -0.0008 & -0.0005 & -0.0011 & 0.0003 & -0.0017 \end{bmatrix} \tag{B.9}$$

To test the quality of the calibration the validation data sets are used. Six plots with the verification data from the right foot are shown in figure B.3. The red asterixes show the resulting output forces and moments (y-axis) with respect to what should theoretically be applied in the setup (x-axis). This calculation is performed as defined in (B.3) and (B.4). Ideally the theoretically applied force and moment should be exactly the same as the output of the calibrated senor data using matrix (B.8). The ideal calibration is indicated with a blue line in the plots.

From figure B.3 it is immediately clear that for some sensors the calibration is not very good. While the data indeed seem to behave linearly with respect to the applied forces and moments, the performance varies greatly, depending on what force or moment is measured. While F_z, M_x, and M_y seem be to read correctly, F_x, F_y, and M_z are less accurate. The validation plots for the left FTS will not be presented here.

Calculating the Root Mean Square Deviation (RMSD) with respect to the ideal curves shown in figure B.3 results in deviations as shown in table B.3. While large RMSD in the M_z readings can be caused by the fact that only low torques are applied around the z-axis, it

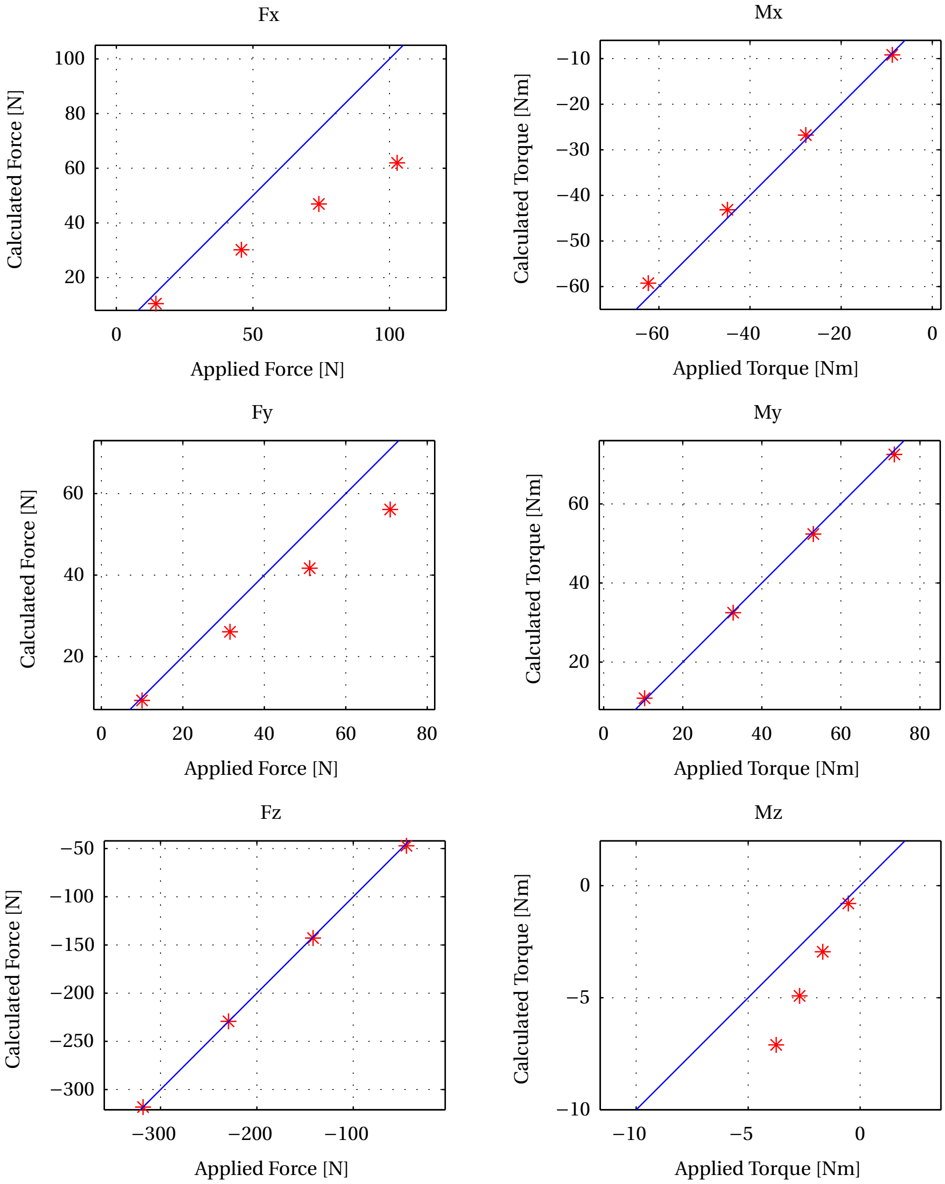


Figure B.3. Verification data (red asterix) and ideal calibration (blue line) plotted for the three forces and moments.

remains unclear why F_x and F_y give such bad results for the right FTS. Other project groups have been able to achieve maximum deviations of approximately 5 % on all sensors [5]. It should however be noted that these results were achieved by performing verification in the same direction as some of the calibration data were collected.

The individual data points have been searched, looking for the cause of the poor performance, but no clear cause was found. Only F_z, M_x, and M_y will be used in the calculation of the ZMP hence no further investigation will be performed.

Output:	**RMSD right FTS [%]:**	**RMSD left FTS [%]:**
F_x	34.80	2.18
F_y	16.85	5.12
F_z	2.64	8.75
M_x	4.31	2.87
M_y	2.76	2.71
M_z	76.02	55.11

Table B.3. RMSD of verification data.

Appendix C

Deduction of Inverse Kinematics

This appendix contains the deduction of the inverse kinematics of the right leg. Compared to the inverse kinematics section 4.2.2, this appendix will concentrate on the algebraic operations performed during the deduction and start out from the specification of the problem given in equation (4.12).

In the following, the notation C_x and S_y will denote $\cos(\theta_x)$ and $\sin(\theta_y)$ respectively and C_{xy} denotes $\cos(\theta_x + \theta_y)$.

From equation (4.12) in section 4.2.2 the inverse kinematic problem is identified to be:

$$
\begin{aligned}
{}^0_6\mathbf{T} &= {}^0_6\tilde{\mathbf{T}} \\
&\Downarrow \\
{}^0_1\mathbf{T}\,{}^1_2\mathbf{T}\,{}^2_3\mathbf{T}\,{}^3_4\mathbf{T}\,{}^4_5\mathbf{T}\,{}^5_6\mathbf{T} &= \begin{bmatrix} i'_x & j'_x & k'_x & p'_x \\ i'_y & j'_y & k'_y & p'_y \\ i'_z & j'_z & k'_z & p'_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{C.1}
$$

The transformation matrices are as defined below in correspondence with figure 4.12 as they were defined in section 4.2.2:

$$
{}^0_1\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & l_{x1} \\ 0 & C_1 & -S_1 & l_{y1} \\ 0 & S_1 & C_1 & l_{z1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}^1_2\mathbf{T} = \begin{bmatrix} C_2 & 0 & S_2 & 0 \\ 0 & 1 & 0 & 0 \\ -S_2 & 0 & C_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}^2_3\mathbf{T} = \begin{bmatrix} C_3 & 0 & S_3 & 0 \\ 0 & 1 & 0 & 0 \\ -S_3 & 0 & C_3 & l_{z3} \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^3_4\mathbf{T} = \begin{bmatrix} C_4 & 0 & S_4 & 0 \\ 0 & 1 & 0 & l_{y4} \\ -S_4 & 0 & C_4 & l_{z4} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}^4_5\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_5 & -S_5 & 0 \\ 0 & S_5 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}^5_6\mathbf{T} = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

The first three joint angles that will be deduced are θ_1, θ_2, and θ_3. The position of the hip frame can be expressed from these three angles. From ${}^0_6\tilde{\mathbf{T}}$ the position of the hip seen from the foot is defined by $\mathbf{p}' = [p'_x p'_y p'_z]^T$. Extracting the corresponding entries in ${}^0_6\mathbf{T}$, the following three equations can be stated:

$$
\begin{aligned}
p'_x &= l_{z4}(C_2S_3 + S_2C_3) + l_{z3}S_2 + l_{x1} \\
p'_y &= l_{y4}C_1 + l_{z4}(S_1S_2S_3 - S_1C_2C_3) - l_{z3}S_1C_2 + l_{y1} \\
p'_z &= l_{y4}S_1 + l_{z4}(C_1C_2C_3 - C_1S_2S_3) + l_{z3}C_1C_2 + l_{z1}
\end{aligned}
$$

Obviously this is three equations with three unknowns which from rules of calculation with cos and sin can be stated as:

$$p'_x - l_{x1} = l_{z4}S_{23} + l_{z3}S_2 \tag{C.2}$$

$$p'_y - l_{y1} = l_{y4}C_1 - l_{z4}S_1C_{23} - l_{z3}S_1C_2 \tag{C.3}$$

$$p'_z - l_{z1} = l_{y4}S_1 + l_{z4}C_1C_{23} + l_{z3}C_1C_2 \tag{C.4}$$

From these three expression we will derive θ_1, θ_2, and θ_3.

Solution to θ_3

First the knee angle θ_3 will be found. By squaring C.2, C.3, and C.4 and adding these together the left hand site (LHS) can be expressed as:

$$LHS = p'^2_x - 2p'_x l_{x1} + l^2_{x1} + p'^2_y - 2p'_y l_{y1} + l^2_{y1} + p'^2_z - 2p'_z l_{z1} + l^2_{z1}$$

And the right hand side (RHS) as:

$$
\begin{aligned}
RHS &= l^2_{z4} + l^2_{z3} + 2l_{z4}l_{z3}(S_{23}S_2 + C_{23}C_2) + l^2_{y4} \\
&= l^2_{z3} + l^2_{y4} + l^2_{z4} + 2l_{z4}l_{z3}C_3
\end{aligned}
$$

Assembling the left and right hand side again, C_3 and from this θ_3 can be isolated:

$$C_3 = \frac{-l^2_{z4} - l^2_{z3} - l^2_{y4} + p'^2_x - 2p'_x l_{x1} + l^2_{x1} + p'^2_y - 2p'_y l_{y1} + l^2_{y1} + p'^2_z - 2p'_z l_{z1} + l^2_{z1}}{2l_{z4}l_{z3}}$$

$$\theta_3 = \operatorname{atan2}\left(\pm\sqrt{1 - C^2_3}, C_3\right) \tag{C.5}$$

The function atan2$\left(y,x\right)$ is almost the same as the normal arcus tangens function $\tan^{-1}\left(\frac{y}{x}\right)$, but instead of returning an angle within $[-\pi/2;\pi/2]$, atan2 returns an angle within $[-\pi;\pi]$ adjusted to the right quadrant from the signs of x and y.

Solving for θ_2

The next joint angle that will be found is θ_2. To find this angle, equation C.3 and C.4 is squared and added together which yields:

$$(p'_y - l_{y1})^2 + (p'_z - l_{z1})^2 = (l_{z4}C_{23} + l_{z3}C_2)^2 + l_{y4}^2$$

The C_{23} term can now be split up to give the equation the following form:

$$\begin{aligned}
\pm\sqrt{(p'_y - l_{y1})^2 + (p'_z - l_{z1})^2 - l_{y4}^2} &= l_{z4}C_{23} + l_{z3}C_2 \\
&= l_{z4}C_2C_3 + l_{z3}C_2 - l_{z4}S_2S_3 \\
&= C_2(l_{z4}C_3 + l_{z3}) - l_{z4}S_2S_3 \qquad \text{(C.6)}
\end{aligned}$$

In equation C.2 the S_{23} can similarly be expanded to be given a very similar structure on the right hand side:

$$\begin{aligned}
p'_x - l_{x1} &= l_{z4}\left(C_2S_3 + S_2C_3\right) + S_2 l_{z3} \\
&= S_2\left(l_{z4}C_3 + l_{z3}\right) + l_{z4}C_2S_3 \qquad \text{(C.7)}
\end{aligned}$$

From [29] it is known that equations on a form as seen in equation C.8 and C.9, can be solved for θ as defined in equation C.10.

$$a \cdot \cos(\theta) - b \cdot \sin(\theta) = c \qquad \text{(C.8)}$$

$$a \cdot \sin(\theta) + b \cdot \cos(\theta) = d \qquad \text{(C.9)}$$

$$\theta = \text{atan2}(ad - bc, ac + bd) \qquad \text{(C.10)}$$

Since equation C.6 and C.7 have exactly this structure, an expression for θ_2 can be calculated as:

$$\theta_2 = \text{atan2}\left(a_2d_2 - b_2c_2, a_2c_2 + b_2d_2\right) \qquad \text{(C.11)}$$

Where:

$$\begin{aligned}
a_2 &= l_{z4}C_3 + l_{z3} \\
b_2 &= l_{z4}S_3 \\
c_2 &= \pm\sqrt{(p'_y - l_{y1})^2 + (p'_z - l_{z1})^2 - l_{y4}^2} \\
d_2 &= p'_x - l_{x1}
\end{aligned}$$

Solving for θ_1

In a similar fashion an expression of θ_1 can be found. It can be observed that since θ_2 and θ_3 is now known, the structure of C.3 and C.4 is exactly the same as defined in C.8 and C.9. This yields the following solution to the angle:

$$\theta_1 = atan2\left(a_1 d_1 - b_1 c_1, a_1 c_1 + b_1 d_1\right) \tag{C.12}$$

Where:

$$a_1 = l_{z4}$$
$$b_1 = l_{z4} C_{23} + C_2 l_{z3}$$
$$c_1 = p'_y - l_{y1}$$
$$d_1 = p'_z - l_{z1}$$

Solving for θ_5

Next the three angles θ_4, θ_5, and θ_6 that determines the orientation of the leg in the hip joint are found. In order to find these we use the known angle θ_1 to calculate ${}^1_0\mathbf{T} ={}^0_1 \mathbf{T}^{-1}$ and left multiply this on both sides of equation (C.1) to obtain:

$${}^1_2\mathbf{T}{}^2_3\mathbf{T}{}^3_4\mathbf{T}{}^4_5\mathbf{T}{}^5_6\mathbf{T} ={}^1_0 \mathbf{T} \begin{bmatrix} i'_x & j'_x & k'_x & p'_x \\ i'_y & j'_y & k'_y & p'_y \\ i'_z & j'_z & k'_z & p'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.13}$$

In these two new matrices we equal entry $(2,3)$ in each of them and get the following:

$$-S_5 = C_1 k'_y + S_1 k'_z$$

From this equation we can directly find the solution as:

$$\theta_5 = \pm\text{atan2}\left(-C_1 k'_y - S_1 k'_z, \sqrt{1-(-C_1 k'_y - S_1 k'_z)^2}\right) \tag{C.14}$$

Solving for θ_6

To find an expression for θ_6 the same principle is used, but this time we set up two equations. Equation C.15 we get by comparing entry $(2,1)$ in equation (C.13) and equation C.16 we get from entry $(2,2)$ in equation (C.13).

$$C_5 S_6 = C_1 i'_y + S_1 i'_z \tag{C.15}$$
$$C_5 C_6 = C_1 j'_y + S_1 j'_z \tag{C.16}$$

By dividing these two equations tangent of θ_6 is found.

$$\frac{S_6}{C_6} = \tan(\theta_6) = \frac{C_1 i'_y + S_1 i'_z}{C_1 j'_y + S_1 j'_z}$$

Which can be solved using the atan2 function:

$$\theta_6 = \text{atan2}\left(C_1 i'_y + S_1 i'_z, C_1 j'_y + S_1 j'_z\right) \tag{C.17}$$

Solving for θ_4

The final and last joint angle that needs to be determined is θ_4. We first compare entry (1,3) in equation (C.13):

$$\begin{aligned} k'_x &= ((C_2C3 - S_2S_3)S_4 + (C_2S_3 + S_2C_3)C_4)\,C_5 \\ &= S_{234}C_5 \end{aligned}$$

Then from entry (3,3) in equation (C.13) we get:

$$\begin{aligned} C_1k'_y - S_1k'_y &= (-S_4S_2C_3 - S_4C_2S_3 + C_4C_2C_3 - C_4S_2S_3)\,C_5 \\ &= C_{234}C_5 \end{aligned}$$

By dividing these two equations a tangent function can be obtained.

$$\frac{S_{234}}{C_{234}} = \tan(\theta_{234}) = \frac{k'_x}{C_1k'_y - S_1k'_y}$$

Which we can solve in the usual way using the atan2 function:

$$\theta_{234} = \text{atan2}\left(k'_x, C_1k'_y - S_1k'_y\right)$$

Angle θ_{234} its self is of no interest to the inverse kinematics, but since θ_2 and θ_3 is known at this point, θ_4 can easily be found.

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \tag{C.18}$$

This concludes the deduction of the joint angles in the right leg of the robot. For the left leg the procedure is exactly the same, only some of the values in the transformation matrices are different.

Appendix D

Coordinate transformations

This appendix introduces how points in one coordinate frame can be transformed into another coordinate frame when the relation between the frames are known. A small example is presented that also introduces the used notation. The notation is inspired from [29].

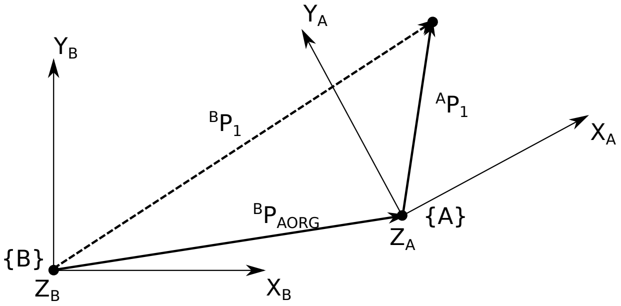


Figure D.1. Position of point P_1 in frame {A} and {B}.

In figure D.1 two frames; {A} and {B} are shown. A point ${}^A\mathbf{P}_1 = \begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^T$ is defined in frame {A} indicated by the superscript A. Frame {A} is described with respect to frame {B} by the orientation with respect to frame {B} and the translation of frame {A} in frame {B}. The orientation of frame {A} with respect to frame {B} is given as a rotation matrix ${}^B_A\mathbf{R} \in \mathbb{R}^{3\times3}$. The translation of frame {A} with respect to frame {B} is given as the position of the origin of frame {A} in frame {B}: ${}^B\mathbf{P}_{AORG} = \begin{bmatrix} x_{AORG} & y_{AORG} & z_{AORG} \end{bmatrix}^T$. From the relation between the two frames and the known position of point ${}^A\mathbf{P}_1$ in frame {A} the position of point ${}^B\mathbf{P}_1$ in frame {B} is calculated as:

$${}^B\mathbf{P}_1 = {}^B_A\mathbf{R}\;{}^A\mathbf{P}_1 + {}^B\mathbf{P}_{AORG} \tag{D.1}$$

To simplify this notation both the rotation and translation will be combined into a single transformation given by one transformation matrix, transforming points in frame {A} to frame

{B}. This transformation is given as follows:

$$\begin{bmatrix} {}^{B}\mathbf{P}_1 \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} {}^{B}_{A}\mathbf{R} & {}^{B}\mathbf{P}_{AORG} \\ \hline \mathbf{0} & 1 \end{array}\right] \begin{bmatrix} {}^{A}\mathbf{P}_1 \\ 1 \end{bmatrix} \tag{D.2}$$

Being able to transform positions between two frames makes it possible to transform into any number of frames, when the orientation and translation that links the frames is known. Continuing the previous example having the following chain of frames: {A}→{B}→{C}→… →{L}→{M}, the position of point ${}^{M}\mathbf{P}_1$ in frame {M} is determined by using the following transformation:

$${}^{M}_{A}\mathbf{T} = {}^{M}_{L}\mathbf{T} \dots {}^{C}_{B}\mathbf{T}\,{}^{B}_{A}\mathbf{T} \tag{D.3}$$

D.1 Rotation of frames

The rotation part (${}^{B}_{A}\mathbf{R}$) of the transformation shown in equation (D.2) describes how frame $\{A\}$ is rotated with respect to frame $\{B\}$ and is in general constructed by identifying the unit direction vectors of the axis of frame $\{A\}$ seen from frame $\{B\}$:

$${}^{B}_{A}\mathbf{R} = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix} \tag{D.4}$$

where:

$\mathbf{i} \in \mathbb{R}^3$ is the unit direction vector of the x-axis in {A} seen from {B}

$\mathbf{j} \in \mathbb{R}^3$ is the unit direction vector of the y-axis in {A} seen from {B}

$\mathbf{k} \in \mathbb{R}^3$ is the unit direction vector of the z-axis in {A} seen from {B}

Three commonly used rotations are the rotations about one of the three axis in the frames i.e. Pitch (X), Roll (Y), and Yaw (Z). These rotation matrices are as follows:

$${}^{B}_{A}\mathbf{R}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C & -S \\ 0 & S & C \end{bmatrix} \quad {}^{B}_{A}\mathbf{R}_Y = \begin{bmatrix} C & 0 & S \\ 0 & 1 & 0 \\ -S & 0 & C \end{bmatrix} \quad {}^{B}_{A}\mathbf{R}_Z = \begin{bmatrix} C & -S & 0 \\ S & C & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where:

S is sine of the angle of frame {A} seen from {B}

C is cosine of the angle of frame {A} seen from {B}

D.2 Inverting transformations

In can be shown that transformations on the above form have the property that inverting them makes a transformation in the opposite direction:

$${}^{A}_{B}\mathbf{T} = {}^{B}_{A}\mathbf{T}^{-1}$$

APPENDIX E

Analysis of Friction

This appendix contains a description of a collection of tests conducted to analyse the friction in some of the critical joints in the AAU-BOT1, with main focus on the knee joints. Initial tests on the knee joints have shown that the friction in the knees is high and angle dependent. To able to cope with friction in the controller, a model must be created. This appendix contains a description of the tests carried out to identify the behaviour of the friction and to collect the data needed to make a friction model.

E.1 Identification of the source to the angle dependent friction

The main source to the angle dependent friction was early identified to be pre-shaped belts in the spur gears in the knee joints. The belts are shaped because the robot has been standing still for two years. The belt in the left knee was changed before any thorough test was carried out. For the right knee two datasets for model parameter identification are collected. One data set before the belt was changed and one after the change of the belt to be able to verify the influence on the pre-shaped belts.

Even though changing the belts removes a part of the angle dependency on the friction some angles do however seem to suffer more from friction than others. An experiment is carried out to be able to analyse the position dependency.

E.2 Friction experiments

Two different types of friction experiments are carried out to analyse and model the friction in the joints. The general idea in the tests is to control the joint to move with constant velocity such that the moment of inertia of the moving part does not have any influence and the measured applied current is caused by the friction when the effect of the gravity is compensated for. The constant angular velocity is obtained by following an angle ramp reference with a PI(D) controller.

In the experiments two different controllers are used. A high gain PI controller tuned project group 10gr1036 [2] which is not stable for angular velocities above 15 °/s. To be able to run with higher velocities a PID controller with lower k_p and k_i gains and with a differential term is designed. This controller shows smoother movements at higher velocities and is shown stable at velocities up to 40 °/s. In table E.1 the gains for the two controllers are shown.

Controller	$k_p/1e3$	$k_i/1e3$	$k_d/1e3$
Original	550.0	55.0	0.0
New	16.5	16.5	5.0

Table E.1. Controller gains for the used controllers.

To improve performance of the PID controller, a feed forward signal is generated. This feed forward signal is comprised by the sum of an estimate of the effect of the gravity and an initial estimate of the coulomb friction.

In the test the leg is straight in the start position 0° and is then bend to 60° and thereafter straightened again. In the experiment carried out to collect data for the friction model the joint is controlled to bend and stretch at 16 different velocities ranging from 0.25 °/s to 40 °/s. Even though it would be preferable to test at higher velocities is not possible to follow the reference sufficiently well at higher velocities with the used controller. An example of the reference signal is shown in figure E.1.

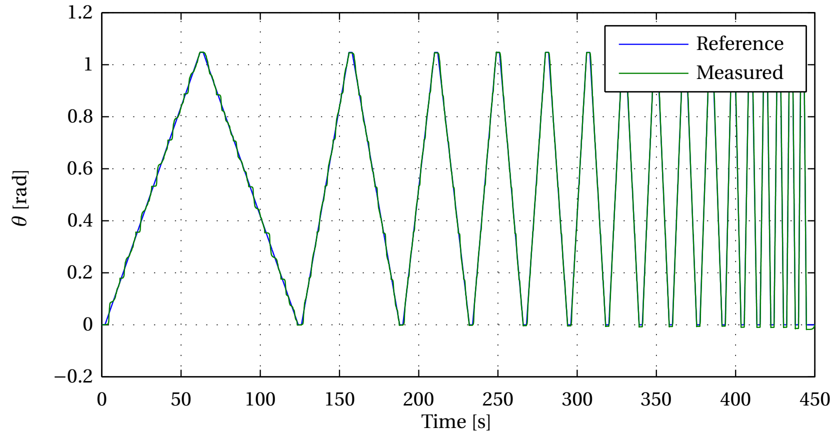


Figure E.1. Reference angle tracked by the controller, and measured angle.

For the experiment carried out to collect data for the analysis of the angle dependent friction the joint is controlled to bend and stretch at 5 °/s and 10 °/s. For both velocities the joint is bended and stretched 20 times to test if the observed phenomenon is an angle dependence and not due to other disturbances.

Appendix F

Recording of the human gait

To gather data for analysis on the human gait, the gait of two female and one male test subject is recorded for different velocities. The gait is recorded by measuring the position of some relevant points on the test subject body while the test subject is walking on a threadmill. The gait is recorded using a motion tracking system where reflecting balls are placed on the points of interest. In figure F.1 the placement of the 14 markers are shown for the male test subject. The markers are placed symmetrically on the test subject. The markers on the left side of the test subject are highlighted with a green circle

Marker position:

- Shoulder
- Elbow
- Hand
- Hip
- Knee
- Heel
- Toe

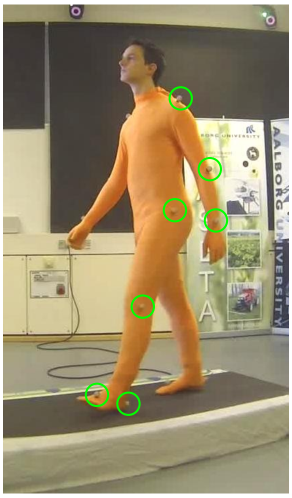


Figure F.1. The male test subject during a gait test. Markers on the left side of the test subject are highlighted with a green circle.

The output of the motion tracking system is a .mat file where the positions of all observed markers are saved. The datasets are placed on the DVD:

F1: 💿`/data/mt_gait/2013_02_01-anja/data/anja_test_XXX.mat`

F2: 💿`/data/mt_gait/2013_02_04-nanna/data/nanna_test_XXX.mat`

M : 💿`/data/mt_gait/2013_01_31-niels/data/niels_test_XXX.mat`

Where XXX is the test number according to table F.1. The gait is recorded for the three test subjects at velocities 0.5, 1.0, 2.0, 3.6, 4.0, 5.0. For each velocity eight recordings are made. Three recordings with a duration of 60 seconds and five recordings with a duration of 10 seconds. The long recordings are recorded to be able to make a mean step, and the short recordings are carried out to obtain additional recordings of the initiation of the gait. In addition to these tests an experiment is carried out where the test subject is asked to walk on the treadmill while it is halted to identify the effect of walking on a treadmill compared to walking on the ground.

The last test carried out for all test subjects is a calibration test. The test subject is asked to stand still on the treadmill while it is running. This test is used to identify the orientation of the treadmill in the motion tracking room. In the test the test subjects is asked to have parallel feet such that this test also can be used to identify bias on the feet angle.

For the male test subject two tests are carried out where he runs with 10 km/h and 15 km/h. These tests are conducted to be able to identify some parameters of a human running. These experiments will however not be of much use in this project.

Test Description	**Female 1 (A)**	**Female 2 (N)**	**Male**
0.5 km/h Stand still (Calibration)	1	1	4
0.5 km/h for 60 s	2-4	2-4	1-3
0.5 km/h for 10 s	5-9	5-9	5-9
1.0 km/h for 60 s	10-12	10-12	10-12
1.0 km/h for 10 s	13-17	13-17	13-17
2.0 km/h for 60 s	18-20	18-20	18-20
2.0 km/h for 10 s	21-25	21-25	21-25
3.6 km/h for 60 s	26-28	26-28	26-28
3.6 km/h for 10 s	29-33	29-33	29-33
4.0 km/h for 60 s	34-36	34-36	34-36
4.0 km/h for 10 s	37-41	37-41	37-41
5.0 km/h for 60 s	42-44	42-44	42-44
5.0 km/h for 10 s	45-49	45-49	45-49
Walk on non-moving treadmill	50-54	50-54	50-54
10 km/h (run)			55
15 km/h (run)			56

Table F.1. Test number for the recorded gaits.

F.1 Extracting Marker Positions

In the data files from the tests, the position is saved for all recognized markers. Since the markers are identical the motion tracking system can not distinguish the markers and the markers are therefore stored in the file in a random[1] way. To compensate for this a MATLAB® script is written that loads a data set and sorts the markers by comparing the distances between markers at the current sample and the markers at the previous sample. The sample code is found on the DVD 💿`/data/mt_gait/motion_extraction_example.m`

In addition to extraction of the marker positions, the script shows a representation of how the markers move in the space. In figure F.1 three snapshots of the representation is shown for the male test subject walking at 3.6 km/h.

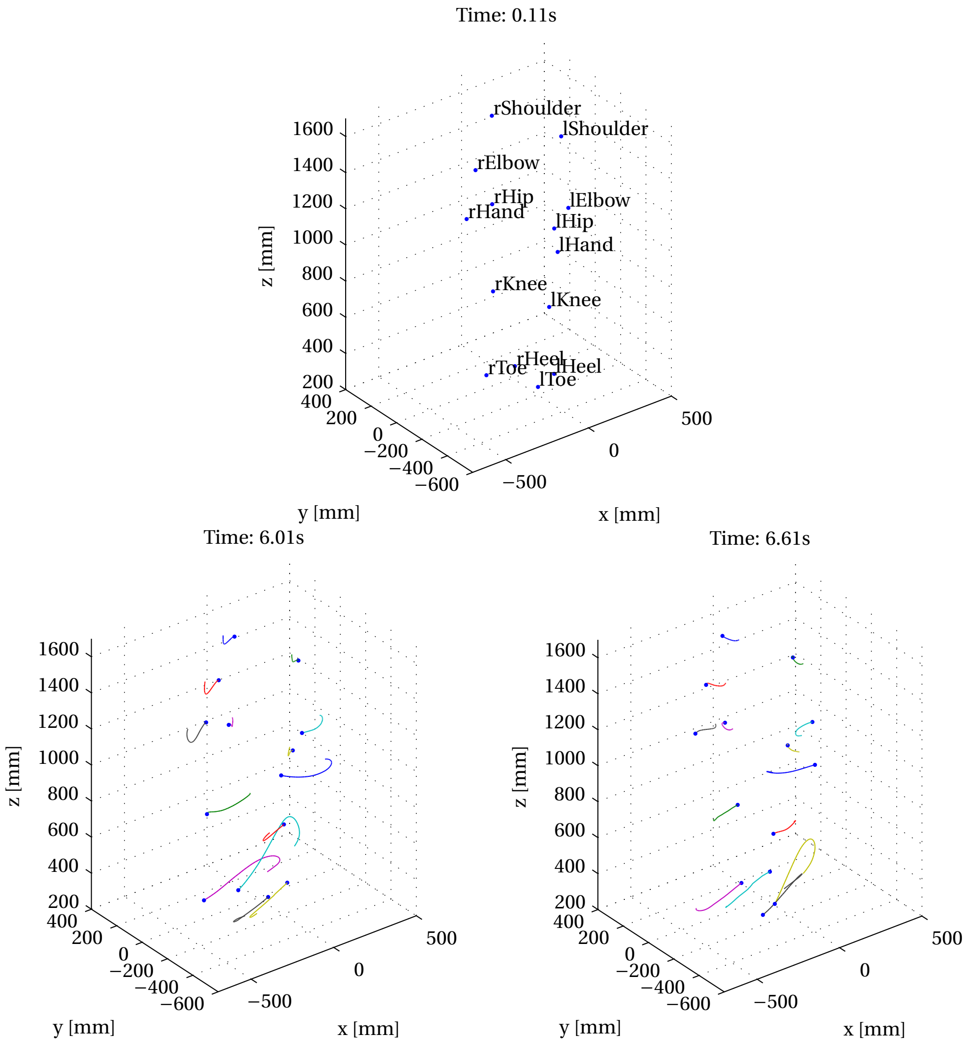


Figure F.2. Three snapshots showing the marker positions for test 26 for the male test subject. The traces are 0.5 s. In the first snapshot the gait has not been initiated and no traces are therefore visible. In addition marker names are shown here to be able to identify the markers.

[1] Not completely random. If all markers are correctly recognised at every sample the markers are stored in the same order at every sample. But if one marker is not recognized or a non-existing marker is recognized the order at that sample and the following samples will be shuffled.