

Sensor Fusion on a Humanoid Robot

Improving the estimation of the center of mass

Bart Vissers

Master of Science Thesis



Sensor Fusion on a Humanoid Robot

Improving the estimation of the center of mass

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Mechanical Engineering at Delft
University of Technology

Bart Vissers

August 20, 2012

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
BIO MECHANICAL DESIGN

The undersigned hereby certify that they have read and recommend to the Faculty of
Mechanical, Maritime and Materials Engineering for acceptance a thesis entitled

SENSOR FUSION ON A HUMANOID ROBOT

by

BART VISSERS

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE MECHANICAL ENGINEERING

Dated: August 20, 2012

Supervisor(s):

dr.ir. M. Wisse

Reader(s):

prof.dr.ir. P.P. Jonker

ir. E.J.H de Vries

Abstract

When a humanoid robot is performing a certain task and is simultaneously maintaining balance, it is important that it can accurately measure its own state. The state is described by a number of variables, such as the center of mass (CoM) and the location of the feet. This information is used by the motion controllers to properly execute the task and control the posture. The accuracy of these measurements is crucial, since incorrect state information leads to an incorrect control action, which may harm the stability on the robot.

With two experiments on the humanoid robot TULip, problems with the current state estimation were identified. Firstly, in an experiment with a motion capture system, it was shown that due to backlash and a constant bias, errors of over 80 mm occur in the perception of the feet. Secondly, using a force plate it was found that the CoM position and velocity estimation also contain errors. Here an average error of 23 mm was observed for the CoM position. The average error in the CoM velocity was 2 mm/s but had a standard deviation of 21 mm/s. On TULip, the prediction of the step location heavily relies on the accuracy of the CoM. Thus, the error in the CoM results in an erroneous and unsteady step location prediction. The focus for this thesis therefore lies on improving the CoM estimation, and the main goal is *improving the estimation of the center of mass position and velocity by applying sensor fusion on the humanoid robot TULip*.

Subsequently, a new CoM estimator is designed. First, an analysis of the current CoM position and velocity calculations is made. The calculations are then improved since these serve as an input to the CoM estimator. The new estimator is based on the Kalman filter. It uses the Linear Inverted Pendulum Model to make a forward prediction of the CoM dynamics. For the state correction the Kalman filter uses the calculations of the CoM position and velocity, and the CoM velocity measured by the inertial measurement unit.

Finally, the new estimator tested using a force plate. It was found that the CoM position estimation is not significantly improved. This is caused by errors in the kinematic model. For the CoM velocity estimation however, a large improvement was observed. The average error is now 4 mm/s with a standard deviation of just 3 mm/s. This makes the prediction of the step location more steady by a factor of 7. The CoM velocity estimation is now sufficiently accurate for a proper prediction of the step location.

Table of Contents

1	Introduction	1
1-1	Problem Definition	1
1-2	Research Goal	2
1-3	Thesis Outline	2
2	Background Information	3
2-1	The Humanoid Robot Tulip	3
2-2	The Instantaneous Capture Point	6
2-3	The Kalman Filter	8
3	Accuracy of the Current Measurements	11
3-1	Accuracy of the Foot Location	12
3-2	Accuracy of the CoM Calculation	23
3-3	Goal for this thesis	30
4	Current Calculation of the Center of Mass	33
4-1	Calculation of the CoM Position	33
4-2	Calculation of the CoM Velocity	33
4-3	Analysis of the Current State Estimation	35
4-4	Conclusion	36
5	Simulation	37
5-1	Modelling	37
5-2	Results	39
5-3	Conclusion	40
6	The New Center of Mass Estimator	41
6-1	General Description	41
6-2	Calculation of the Measurement Signals	42
6-3	Modelling the Dynamics of the Robot	43
6-4	Conclusion	45

7	Experimental Results	47
7-1	Double Support	47
7-2	Single Support	50
7-3	Discussion of the Results	52
7-4	Tuning the Kalman Filter	52
7-5	Conclusion	54
8	Discussion and Future Work	55
8-1	Using a Software Based Solution	55
8-2	The Motion Capture Experiment	55
8-3	The Force Plate Experiment	56
8-4	Implementation of the Kalman Filter	56
8-5	Extensions to the Kalman Filter	57
8-6	Future Work	58
9	Conclusions	59
	Bibliography	61

List of Figures

2-1	The kinematic configuration(a) indicating the degrees of freedom where Series Elastic Actuation (SEA) is applied, and a photograph (b) of the humanoid robot TULip.	4
2-2	Schematic diagram of a series electric actuator. Between the motor and the load a spring is added. By controlling the elongation of the spring, the load is effectively controlled by force instead of position [1].	4
2-3	Detailed view of the left leg of TULip, which shows the positioning of the actuators and springs for applying series elastic actuation. The illustration is adopted from [2].	5
2-4	If the robot is pushed by an external force F , it must step onto the Capture Region in order to come to a full stop [3].	7
2-5	Picture of the Linear Inverted Pendulum Model (LIPM), which is used for the calculation of the ICP. It consists of a point mass at constant height z_0 and a massless telescopic leg. $\mathbf{P}\mathbf{x}_{com}$ denotes the projection of the CoM to the ground plane. The picture is adopted from [4].	8
2-6	Block diagram of the Kalman filter.	10
2-7	Visualization of the effect of the Kalman filter. Given two measurement signals P and V , each with a certain probability. The Kalman filter is able to make an estimate of state with a higher probability than the individual signals. Figure is adopted from [5].	10
3-1	Overview of the measurement setup, showing the robot hanging in the frame surrounded by six motion capture cameras.	12
3-2	A screen capture of the Qualisys Track Manager software, which is used to collect and process the motion capture data.	14
3-3	Schematic top view of the robot, showing the five movement directions of the right foot. Similarly, for the left foot the movements are mirrored.	15
3-4	A schematic graph of the expected behavior of backlash. It consists of a trajectory and a measured trajectory under the influence of ± 0.5 backlash. At $t = 1$ the trajectory moves upwards, where backlash causes a delay in the measured signal. The movement stops at $t = 2$, the error then remains at -0,5. When the trajectory moves downwards at $t = 3$, an error of 0.5 in opposite direction arises.	16

3-5	The trajectory of the right ankle is shown in x (top) and y direction (bottom), of both the motion capture (MC) measurement and the data from TULip (VTD). The trajectory is expressed relative to the left ankle, which is kept in place while the right foot successively moves outwards, forwards, forwards & outwards, backwards, backwards & outwards.	17
3-6	The trajectory of the left ankle is shown in x (top) and y direction (bottom), of both the motion capture (MC) measurement and the data from TULip (VTD). The right foot was kept in place and served as frame of reference.	17
3-7	Trajectory of the right foot and the histogram of the error, while being perturbed in y direction (outwards). A bimodal distribution consisting of two Gaussian functions is fitted to the histogram. The colors in the trajectory plot correspond with the colors of the peaks in the histogram.	18
3-8	Trajectory of the right foot while being perturbed in y direction, and the histogram of the error. A bimodal distribution consisting of two Gaussian functions is fitted to the histogram.	19
3-9	Top view of the feet before and after a left step (trial 1). In red the positions before the step are shown, the positions after the step are shown in blue. Furthermore, both the MC and VTD data are shown. A large error is observed both before and after the step.	21
3-10	Top view of a step with the right foot (trial 7). It shows that in this case, both before and after the step a rotational error is present.	21
3-11	Photograph of the experimental setup. The robot is standing on the force plate (square on the floor) at the VUMC. It is protected from falling by the frame behind the robot. During the experiments the cables are slack.	26
3-12	The plots show the difference between the unfiltered and filtered CoP data. After filtering the CoP position (a) with a second order Butterworth filter at 0.5 Hz, only minor differences are visible. However, when differentiating the filtered CoP position this results in a much smoother CoP velocity signal (b).	27
3-13	While the robot was in double support, a sinusoidal ICP trajectory was used to slowly oscillate the robot in x direction. In the graph the current position estimator is compared to the CoP position measured by the force plate. A constant error of 23 ± 1.1 mm is observed with respect to the CoP position.	28
3-14	Comparison of the current estimator with the differentiated CoP position, under the same conditions as with Figure 3-13. The CoM velocity error is currently 2 ± 21 mm/s.	28
3-15	Plot showing the Gaussian probability distributions of the error in the ICP prediction that arises due to an incorrect CoM position and CoM velocity.	29
5-1	Drawing of the Linear Inverted Pendulum Model (LIPM). The model consists of a point mass on a fixed height (z_0) and a massless telescopic leg. The ankle torque (τ_a) is applied to the base of the pendulum.	38
5-2	Result from the simulation of the Kalman filter that is applied on a LIPM. The graphs show the position (top) and velocity of the CoM (bottom). The Kalman filter used the noise signals and forward model to make an estimation of the real trajectory.	40
6-1	Illustration of the LIPM, indicating the height of the CoM and the location of the IMU. Due to the height difference, the acceleration does not correspond to the acceleration of the CoM.	43

7-1	While the robot was in double support, a sinusoidal ICP trajectory was used to slowly oscillate the robot in x direction. In the graph the old and new CoM position estimations are compared to the CoP position measured by the force plate. It shows that both estimators perform similar, with a large constant error with respect to the CoP position.	48
7-2	Comparison of the old and new CoM velocity estimations, and the differentiated CoP position, for the x direction in double support. It shows that there is a large error between the old estimator and the CoP velocity. However, the new estimator is much more accurate.	48
7-3	Probability distribution of the error on the predicted ICP in x direction, resulting from an inaccurate CoM position and velocity estimation. The new estimator shows only a minor improvement for the error due to the CoM position. However, the error due to the CoM velocity estimation has become much less. The corresponding averages and standard deviations are listed in Table 7-1.	49
7-4	Comparison of the old and new CoM position estimator with the measurements from the force plate, for the x direction. The robot was standing solely on the right foot. As with the experiment in double support, a large constant offset is present.	51
7-5	The CoM velocity in x direction of the CoM velocity estimators and the force plate measurement, while the robot was standing on its right foot. For clarity only the first 20 seconds of the experiment is shown. Again, the new estimator performs significantly better than the old one.	51

List of Tables

3-1	Overview of the backlash observed during the experiment, for the movement directions where a significant backlash could be observed. The backlash was determined by the distance between the two peaks in the bimodal distribution.	19
3-2	Errors of the ankle position in the left step experiments, relative to the swing foot. When combining the measurements before and after the step, the following averages and standard deviations are found: $\mu_x = 19 \pm 10.0$ mm, $\mu_y = 47 \pm 11$ mm.	22
3-3	Errors of the ankle position in the right step experiments relative to the swing foot. The averages and mean with the errors before and after the step combined: $\mu_x = 13 \pm 23$ mm, $\mu_y = -52 \pm 14$ mm.	22
3-4	Angular error in the left swing foot in VTD, relative to the swing foot in MC. It shows that for four trials not error is present, except for the last one.	23
3-5	Angular error in the right swing foot. Here in three of the trials a large angular error was observed.	23
3-6	Relevant properties of the robot TULip, used for the analysis of the predicted ICP error, resulting from a inaccurate CoM.	25
5-1	Parameters used in the simulation of the Kalman filter. These values resemble the robot TULip as much as possible.	39
7-1	Comparison of the old and new estimators, on the effect on the predicted step location. For the CoM position of little improvement is observed, but the CoM velocity has become much more accurate. Figure 7-3 shows the corresponding probability distributions.	49
7-2	The average errors and standard deviations of the CoM position and velocity estimators, for the experiment in single support. Again, it is shown that the improvement in the CoM position estimator is only minor. For the new CoM velocity estimation, the standard deviations are much smaller, which indicates a significant improvement.	50
7-3	The average errors and standard deviations in the initially predicted ICP due to an error in the CoM, for the experiment in single support on the right foot.	52
7-4	Noise variances that are found with the experiment in the double support.	53

Nomenclature

List of Abbreviations

DoF	Degrees of Freedom
ICP	Instantaneous Capture Point
IMU	Inertial Measurement Unit
LIPM	Linear Inverted Pendulum Model
MC	Motion Capture
SEA	Series Elastic Actuation
VTD	Variable Tuning Dialog. The application to register the internal data of TULip

List of Symbols

$\mathbf{x}_{icp}$	Location of the ICP
$\mathbf{x}_{icp}(\Delta t)$	Location of the predicted ICP for step time Δt
ω_0	Reciprocal time constant for the LIPM
τ_a	Torque applied at the ankle
g	Gravitational constant
m	Mass of the robot

List of Symbols of the Kalman filter

$\mathbf{A}$	State matrix of the forward model
$\mathbf{B}$	Input matrix of the forward model
$\mathbf{C}$	Sensor model
$\mathbf{K}_k$	Kalman gain
$\mathbf{P}_k^+$	A posteriori error covariance matrix
$\mathbf{P}_k^-$	A priori error covariance matrix
$\mathbf{Q}$	Process noise covariance matrix
$\mathbf{R}$	Measurement noise covariance matrix
$\mathbf{x}_k^+$	A posteriori state estimate
$\mathbf{x}_k^-$	A priori state estimate
$\mathbf{y}_k$	Vector of measured signals

Acknowledgements

First of all, I would like to thank my supervisor Martijn Wisse for his advice and guidance during the project. Not only during my MSc project, but also during my literature study you have been a great motivator.

Secondly, I would like to thank my direct colleagues Riewert van Doesburgh, Sebastiaan Kiemel and Tim van der Ven. Working together with you was a pleasure, and our discussions were of great value for my thesis. I would also like to thank the people who helped keeping TULip in one piece. Jan van Frankenhuyzen for his help on the mechanics, Guus Liqui Lung for solving electronical problems, and Gijs van der Hoorn for fixing software bugs.

During this thesis I got the opportunity to do two great experiments with TULip. I would like to thank Patrick Forbes for letting me borrow the motion capture system, and helping me setting up the cameras and markers. The force plate experiment was only possible thanks to Marjolein van der Krogt and Kim van Hutten at the VUMC in Amsterdam.

Finally, I would like to thank my family and girlfriend for their continuous support throughout my study.

Delft, University of Technology
August 20, 2012

Bart Vissers

Chapter 1

Introduction

Currently it is very challenging to let bipedal walking robots walk and maneuver in a human-like manner. Additionally, robots lack the ability to withstand large disturbances, while this comes natural for human beings. These challenges are also encountered with the bipedal walking robot TULip. The research on this robot currently focuses on robust balancing and walking. However, until now TULip is not able to perform stable walking. For a large part this is caused by an inaccurate state estimation.

A robot calculates several important variables to control the stability, such as the center of mass, the center of pressure underneath the feet, and the location of its limbs. In the event of a disturbance, this information is used to assess its current state and determine an appropriate control action. However, when these variables are inaccurately measured or calculated, the robot is unable to give a proper response, which may even lead to a fall.

The accuracy of the variables declines due to various mechanical deficiencies and sensor dynamics, which add a bias, noise and drift to the measurement signal. Additionally, errors arise due to modeling errors, which has two causes: First, parameters used in calculations may be inaccurate. This especially applies to parameters that are hard to determine in practice, such as the inertia of each limb. Second, some calculations are simplified, because for more realistic models no closed form solution exists, or are too computationally intensive to be performed in real time. However, this goes at the cost of accuracy.

Accurately estimating the state of the robot is thus a difficult task, since numerous errors arise with both sensing and processing information. This makes it harder to determine a suitable response, which ultimately limits the performance. Therefore improving the state estimation on the humanoid robot TULip is the main focus of this thesis.

1-1 Problem Definition

Several roboticists [6, 7] have identified issues with an incorrect estimation of the center of mass (CoM), and with the humanoid robot TULip we currently also face these problems. An

accurate estimation of the CoM is crucial, since it is used to control the balance and predict the step location during walking. The CoM is particularly difficult to determine, because it can not be directly measured, and depends on the posture of the entire robot.

For this thesis, it is intended to improve the CoM estimation by applying sensor fusion. With sensor fusion, sensory data from multiple sources is combined to improve the estimate of a state. The Kalman filter [8] is a well-established and lightweight method of realizing this, which will therefore be implemented on the robot. Thus, the focus of this research lies on improving the estimate of the center of mass position and velocity for the humanoid robot TULip.

1-2 Research Goal

The goal of this thesis is therefore:

Improving the estimation of the center of mass position and velocity by applying sensor fusion on the humanoid robot TULip.

1-3 Thesis Outline

This thesis is divided in the following chapters:

Chapter 2 presents background information on important concepts that are used throughout this thesis.

Chapter 3 presents the results of two experiments that were done on the robot TULip. This illustrates the error in the perception of the feet and the error in the current calculation of the center of mass.

Chapter 4 presents an explanation and analysis of the center of mass calculator that is currently implemented on TULip.

Chapter 5 presents a simulation of the Kalman filter that is used to design the improved CoM estimator.

Chapter 6 introduces the new estimator for the center of mass, based on the Kalman filter. Additionally, the CoM calculations are improved using the analysis of chapter 4.

Chapter 7 describes an experiment that is used to assess the performance of the new CoM estimator.

Chapter 8 presents a discussion on the results and the used methods throughout this thesis. Furthermore, possible future research is presented.

Chapter 9 presents the main conclusions of this thesis.

Background Information

In this chapter first the research platform is introduced, the robot TULip. Then an explanation of the Instantaneous Capture Point is given, which is used to predict the step location for the robot. Additionally, this will be used to assess the performance of the center of mass estimator. Finally, a brief introduction of the Kalman filter is presented, which is applied in the new estimator.

2-1 The Humanoid Robot TULip

The CoM estimator that is designed for this thesis will be implemented on the humanoid robot TULip. Therefore this section will give a brief introduction.

TULip [9, 2] is electrically powered and serves as a research platform on human-like walking and balancing. It is also intended to participate in the robotic soccer competition RoboCup [10]. Currently, the research focus mostly lies on walking. However, up to now, TULip is unable to produce a stable walking gait.

The robot has an onboard computer and can run on batteries, which enables it to operate fully autonomously. It runs Linux combined with the real-time framework Xenomai. The control loop runs at 1 KHz, and the software for controlling the robot is written in C++.

The humanoid robot Flame [9] is the predecessor of TULip, that utilized Limit Cycle Walking [11]. However, Flame was restricted to planar walking. TULip on the other hand, is more versatile due to an additional Degree of Freedom (DoF) at the hip. This allows it to make turns while walking.

The kinematic configuration and a photograph of the robot are shown in Figure 2-1. TULip has a height of about 1.3 m and weighs 16.6 kg. Similar to humans, it has six DoF per leg; a three DoF hip, a one DoF knee and a two DoF ankle.

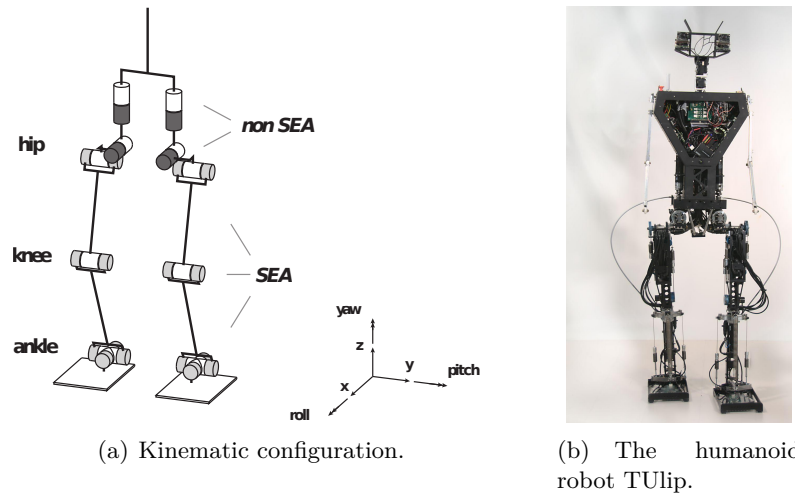


Figure 2-1: The kinematic configuration(a) indicating the degrees of freedom where Series Elastic Actuation (SEA) is applied, and a photograph (b) of the humanoid robot TULip.

2-1-1 Actuation

TULip makes use of Maxon brushed DC motors with planetary gearheads. This type of transmissions is very compact and light weight, but suffer from one to two degrees of backlash [12, 13]. The actuation was designed so that both position control and force control can be applied. The use of force control generally leads to smooth and natural motions, this is therefore extensively used in the current control algorithms.

For providing force control, Series Elastic Actuation (SEA) is applied. With SEA, an elastic element is placed between the actuator and the joint, as schematically shown in Figure 2-2.

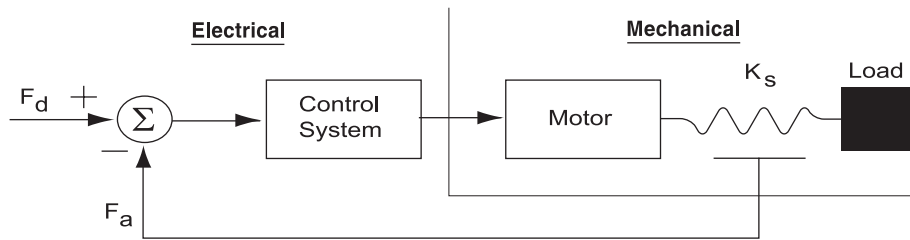


Figure 2-2: Schematic diagram of a series electric actuator. Between the motor and the load a spring is added. By controlling the elongation of the spring, the load is effectively controlled by force instead of position [1].

SEA provides several additional advantages to a robot, such as shock absorption and a high force fidelity [1]. However, a major drawback is the low control bandwidth, which is mostly dependant on the stiffness of the spring and the desired output force. For TULip it was found that the bandwidth is 10-15 Hz for torques up to 5 Nm and approximately 5-10 Hz for torques between 5-12 Nm [2].

As indicated in Figure 2-1(a), all joints except the hip x joint and the hip z joint are actuated using SEA. Moreover, Figure 2-3 shows the mechanics of left leg in detail. For all SEA joints,

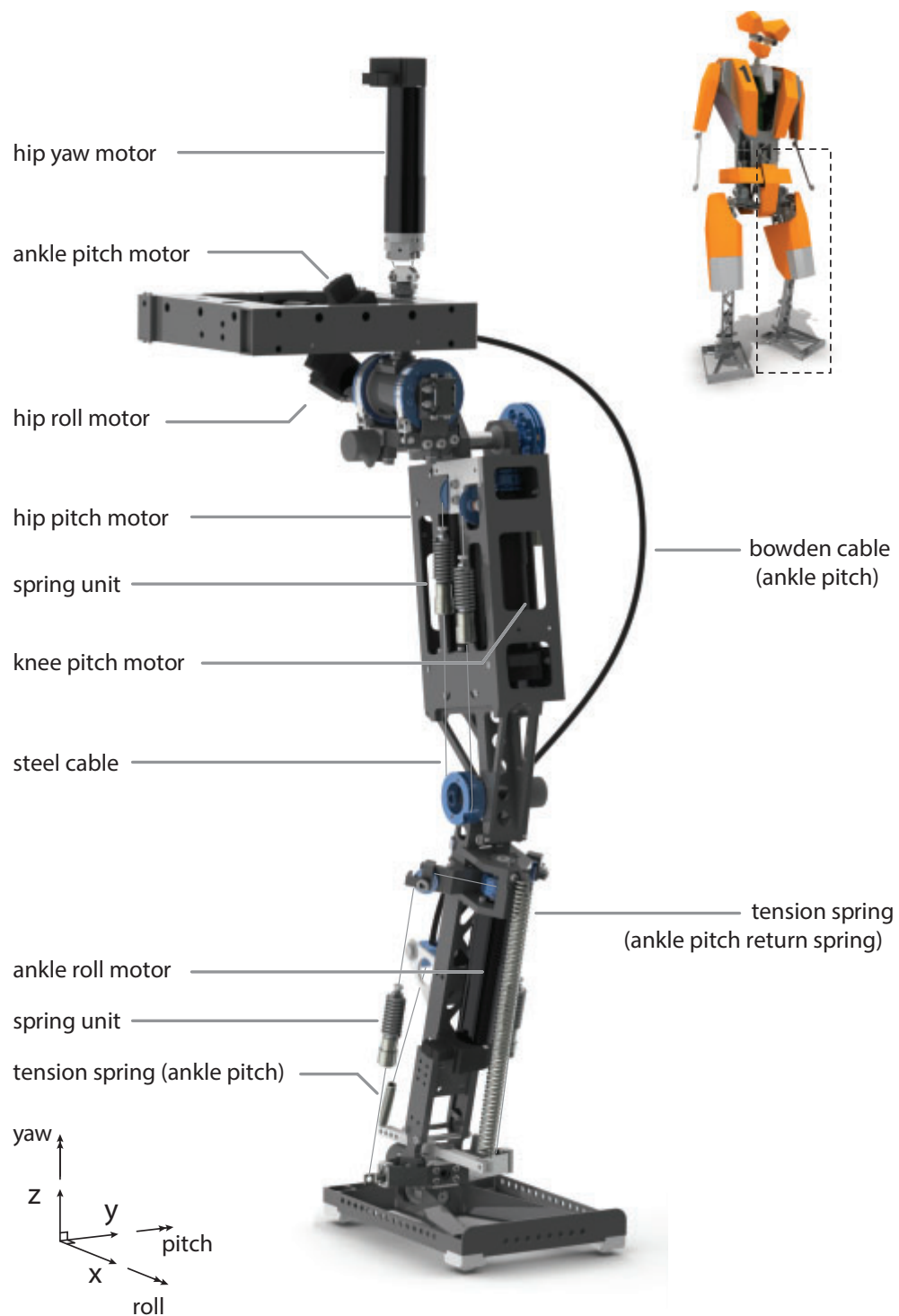


Figure 2-3: Detailed view of the left leg of TULip, which shows the positioning of the actuators and springs for applying series elastic actuation. The illustration is adopted from [2].

linear springs are used that are connected to the motor using steel cables and pulleys. Only the ankle pitch actuation is different. To reduce the moment of inertia of the legs about the hip joints, this motor is positioned in the trunk and a bowden cable is used to transmit the power to the joint. Additionally, the dorsiflexion is passively actuated with a return spring.

2-1-2 Sensor System

When using SEA, a sensor is required on both sides of the spring to be able to measure the elongation and thus the exerted force. On TULip rotary encoders are used, that are placed on the back of the motor and directly on the joint. Additionally, this results in an accurate measurement of the angle of the joints, despite the backlash in the planetary gearheads. However, the hip x joint and the hip z joint are not force controlled and thus lack this additional sensor. Since backlash is also present in these joints, the foot location is currently inaccurately measured by the robot. This is further investigated in Section 3-1.

Furthermore, TULip is equipped with an Inertial Measurement Unit (IMU) and contact sensors underneath the feet. The XSens MTi [14] IMU is attached to the trunk of the robot, and is able to measure the linear accelerations, the angular velocities, angular orientation and the Earth's magnetic field. On the corners of each foot Tekscan Flexiforce pressure sensors [15] are mounted, which are used to detect contact with the floor.

The head of the robot holds two wide angle cameras of type the Surveyor Stereo Vision System [16]. Image processing is done locally using onboard Blackfin processors. However, currently the information from the vision system is not used for controlling the robot.

2-2 The Instantaneous Capture Point

Capture Point theory [3, 4] is a new and promising technique to determine the step location while walking or during push recovery. It is extensively used in TULip. Since this theory heavily relies on a accurate CoM estimation, it is used in the problem definition in Section 3-2-1.

When a robot is recovering balance by stepping, it must determine the place where it should place its swing foot. This can be done using Capture Points. The Capture Point is a point on the ground where the robot should step to come to a complete stop. In practise this is rarely a single point, but rather a region on the ground where the robot should step in. Therefore the set of all Capture Points are called the Capture Region, as shown in Figure 2-4. So when a robot is pushed, it must bring its support polygon inside the Capture Region by making a step. Here the magnitude of the push determines the size and location of the Capture Region:

- If the disturbance is small and the Capture Region already intersects with the support polygon of the robot, it does not need to take a step. It can use another postural control strategy to prevent a fall.
- For a larger push the robot must make a step to let the support polygon intersect with the Capture Region. This will stop the robot from falling as long as the Capture Region is within the reach of the swing leg.

- When the robot is perturbed by a severe disturbance, the Capture Region might not be reachable with a single step. Then the robot has to take two or more steps to come to a stop, if it can stop at all.

The latter case, where the robot must take multiple steps, is described by the N -step capturability theory [4]. Here N denotes the number of steps, that forms the N -step Capture Region. This essentially describes how many steps are required to come to a stop.

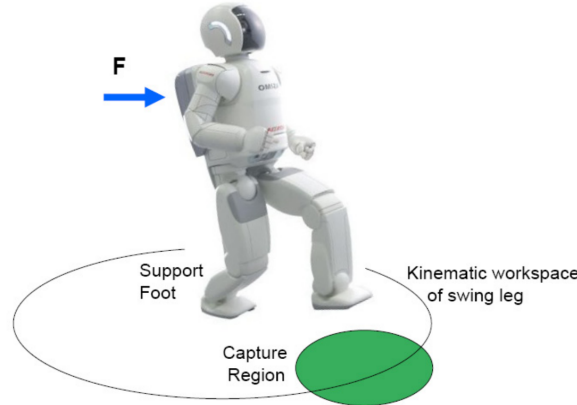


Figure 2-4: If the robot is pushed by an external force F , it must step onto the Capture Region in order to come to a full stop [3].

First we will discuss the Instantaneous Capture Point (ICP), which is the point where the robot should instantaneously step to come to a stop. The ICP is calculated using the Linear Inverted Pendulum Model (LIPM), which is shown in Figure 2-5. This model simplifies the dynamics of the robot to a single point mass. This is further explained in Section 6-3, since it is also used in the CoM estimator. Due to the simplification, the model has linear equations of motion:

$$\ddot{\mathbf{x}}_{com} = \omega_0^2(\mathbf{P}\mathbf{x}_{com} - \mathbf{x}_{ankle}). \quad (2-1)$$

Here $\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ is the projection to the ground plane. ω_0 is the reciprocal of the time constant, defined as $\omega_0 = \sqrt{\frac{g}{z_0}}$, with g as the gravitational constant and z_0 is the CoM height.

Using the LIPM, a closed form solution for the ICP is found:

$$\mathbf{x}_{icp} = \mathbf{P}\mathbf{x}_{com} + \frac{\dot{\mathbf{x}}_{com}}{\omega_0}. \quad (2-2)$$

So this means that if the robot is falling and the CoM moves away from the ankle joint, the robot should instantaneously step to $\mathbf{x}_{icp}$ to come to a stop.

In practise however, a robot can not take a step instantaneously, it takes a certain amount of time. So in order to control the swing leg trajectory, a prediction of the step location after a short amount of time has to be made. With predefined step time Δt , the Capture Point prediction is:

$$\mathbf{x}_{icp}(\Delta t) = (\mathbf{x}_{icp}(0) - \mathbf{x}_{ankle})e^{\omega_0\Delta t} + \mathbf{x}_{ankle}. \quad (2-3)$$

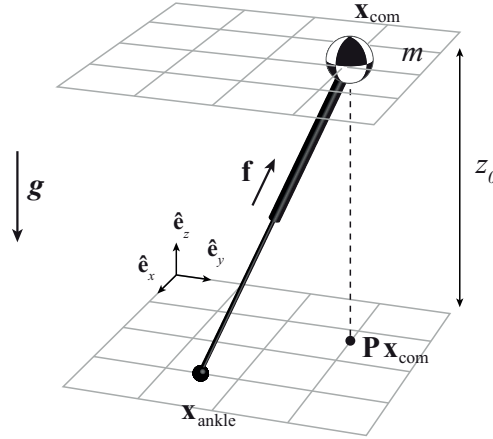


Figure 2-5: Picture of the Linear Inverted Pendulum Model (LIPM), which is used for the calculation of the ICP. It consists of a point mass at constant height z_0 and a massless telescopic leg. $\mathbf{P}\mathbf{x}_{\text{com}}$ denotes the projection of the CoM to the ground plane. The picture is adopted from [4].

When the robot is walking or handling pushes by making a step, this equation is used to calculate the step location, and control the swing leg accordingly. The step location is determined at the beginning of the step, but as the step progresses, the step location is updated so that its accuracy increases.

Thus even though the LIPM is a crude simplification of the dynamics of the robot, it can be used to make a prediction of the step location. And since the equations are analytically derived, it requires little computational effort, allowing it to be calculated in real time. It becomes clear that the ICP prediction heavily relies on the CoM estimation, therefore it is vital that it is accurately measured. However, in the next chapter it will be shown that this is currently not the case.

2-3 The Kalman Filter

For improving the CoM estimation of TULip, a Kalman filter [8, 17] is implemented. In this section gives a brief introduction.

The Kalman filter is an algorithm that can combine multiple measurement signals that contain noise to make an estimation of the system states. The choice for using the Kalman filter is made because it is a well-proven method for state estimation and sensor fusion. It is widely applied, including in many robotic applications. The main benefits are:

- It is computationally fast. This is crucial, because this allows it to run in real-time on the robot.
- Also the memory usage is minimal, as it does not require storing of information of all preceding states. It only calculates the properties of the previous state.

- The algorithm is relatively easy to understand, it is therefore less difficult to maintain and adapt the code later on.
- Many possible extensions for the Kalman filter exist to further improve the performance of the estimator.

Even though a continuous time version of the Kalman filter exists, here only the discrete version will be discussed, since this will also be used on the robot. The Kalman filter uses state space equations to describe the dynamics of the system:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}. \quad (2-4)$$

This is also called the forward model, since it is used by the Kalman filter to make a forward prediction of the states. Furthermore, the Kalman filter needs a sensor model, which describes the measurements ($\mathbf{y}_k$) as a function of the states ($\mathbf{x}_k$):

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k. \quad (2-5)$$

The equations are under the influence of process noise ($\mathbf{w}$) and measurement noise ($\mathbf{v}$), which are defined as zero-mean white noise, with variances $\mathbf{Q}$ and $\mathbf{R}$:

$$p(\mathbf{w}) \sim N(0, \mathbf{Q}), \quad (2-6)$$

$$p(\mathbf{v}) \sim N(0, \mathbf{R}). \quad (2-7)$$

Using the dynamics and the measurements, the Kalman filter is able to make an estimate of the states using,

$$\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k^-). \quad (2-8)$$

Here $\mathbf{K}$ is Kalman gain, which is updated each time step. An overview of the Kalman filter is shown as a block diagram in Figure 2-6. It first makes a prediction of the next state using the forward model:

$$\mathbf{x}_k^- = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}. \quad (2-9)$$

This is also known as the a priori state estimate. For making the final estimate the error covariance between the true state and the estimate is minimized. Therefore first the a priori error covariance is determined:

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}^+ \mathbf{A}^T + \mathbf{Q} \quad (2-10)$$

Subsequently the Kalman gain is calculated,

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}^T (\mathbf{C}\mathbf{P}_k^- \mathbf{C}^T + \mathbf{R})^{-1}, \quad (2-11)$$

which is then used to find the improved, or a posteriori state estimate $\mathbf{x}_k^+$:

$$\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k^-). \quad (2-12)$$

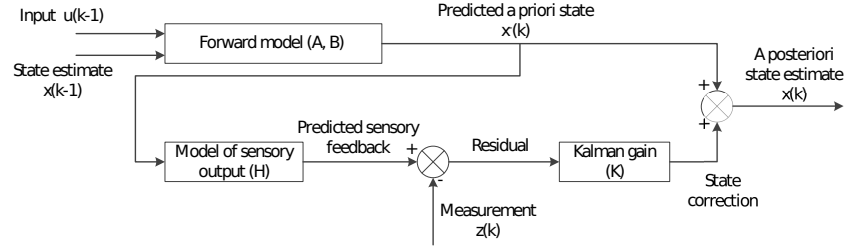


Figure 2-6: Block diagram of the Kalman filter.

Here $\mathbf{y}_k - \mathbf{C}\mathbf{x}_k^-$ is called the residual, which is the difference between the predicted measurement $\mathbf{C}\mathbf{x}_k^-$ and the actual measurement $\mathbf{y}_k$. Finally, the a posteriori error covariance is calculated, which will be used in the next update step. Here $\mathbf{I}$ is the identity matrix:

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{C}) \mathbf{P}_k^- . \quad (2-13)$$

The estimate made by the Kalman filter is optimal when the dynamics of the forward model match the real system, the noise is white and the covariances are exactly known. This is also visualised in Figure 2-7, where multiple measurement signals are combined it find an estimate of the states. This estimate has a higher probability than the individual measurement signals.

The parameters $\mathbf{Q}$ and $\mathbf{R}$ are the process and measurement noise covariances, respectively. These essentially describe the reliability of the a priori estimate and the actual measurements. If the measurements are really accurate, the measurement error covariance ($\mathbf{R}$) decreases to zero. The Kalman filter will then increase the Kalman gain ($\mathbf{K}$) and weights the residual more heavily than the prediction. If, on the other hand, the measurements are less reliable and the prediction is more accurate, the a priori error covariance ($\mathbf{P}_k^-$) decreases to zero. The Kalman gain will then decrease, weighing the prediction more heavily than the residual.

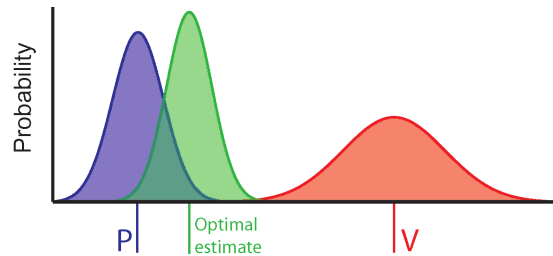


Figure 2-7: Visualization of the effect of the Kalman filter. Given two measurement signals P and V , each with a certain probability. The Kalman filter is able to make an estimate of state with a higher probability than the individual signals. Figure is adopted from [5].

Chapter 3

Accuracy of the Current Measurements

Currently, the main goal for the humanoid robot TULip is robust balancing and walking. The performance mostly depends on the applied control method, and for the controllers it is important that the state of the robot is accurately measured. More specifically, two measurements are of importance; an accurate perception of the location of the feet, and an accurate calculation of the CoM to be able to predict the step location.

Firstly, accurately knowing the foot location is important because it forms the support polygon. This is a convex hull that is described by the feet that are in contact with ground, and indicates the stability margin with respect to the Center of Pressure (CoP). When the robot is walking or maintaining balance, it basically controls the CoP position with respect to the support polygon. Therefore, an inaccurate perception of the foot location results in an incorrect control action, which may lead to a fall.

Secondly, an accurate calculation of the desired step location is important. During walking the speed and direction of the gait can be altered by changing the step location. But more importantly, making a step is a strategy for maintaining balance. This can be applied during walking, for instance by placing the foot further outwards when the robot is pushed sideways. Additionally, stepping serves as a balancing strategy for posture control, where it is applied in the event of severe disturbances. However, in order to maintain balance using the step location, it is crucial that it is accurately calculated and executed, else the performed step may be insufficient to maintain balance. The step location is predicted using the CoM position and velocity, therefore it is vital that these are accurately calculated.

Currently TULip is performing poorly, which is caused by one or both of these factors. This chapter discusses two experiments, one involves a motion capture system and one with a force plate. These are used to evaluate the current determination of the foot location and the CoM of the robot.

3-1 Accuracy of the Foot Location

In order to investigate the accuracy of the current measurements of the foot location, experiments were conducted using the Qualisys motion capture system. The data from the motion capture recording can then be compared to the data of the robot. Here the error between the calculated and observed foot location can be determined. Additionally, some of the mechanical deficiencies in the kinematic chain of the robot are identified. This section explains the setup, goal and experimental results.

3-1-1 Setup and Goal

In the measurement setup, the robot is surrounded by six Oqus cameras [18] in about a 180° angle and at a distance of about 2 meters from the robot. The cameras were placed at alternating heights so that the relative angle between each camera was as large as possible, as this improves the accuracy of the measurement. A frame and cables are used to prevent the robot from falling on the ground if it would lose its balance, but the cables are slack during the experiments.

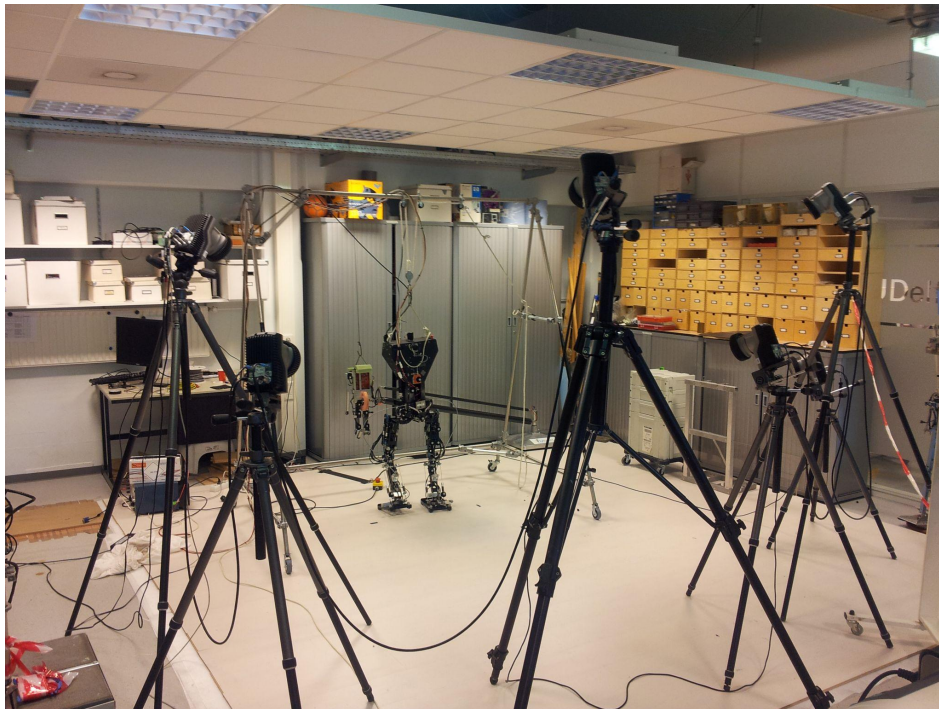


Figure 3-1: Overview of the measurement setup, showing the robot hanging in the frame surrounded by six motion capture cameras.

Passive markers with a diameter of 20 mm were attached to the robot. On each limb at least three markers were required. The accompanied software is then able to reconstruct the movement in six degrees of freedom of the entire robot. Therefore three markers were used for every limb except on the lower legs, where four markers were used. This brings to

total number of markers to 23. During the experiments all movements were recorded and simultaneously the robot logged all its internal data, so it can later be compared.

While the movements of the robot were being recorded two experiments were done:

- The first experiment was intended to measure the effect of backlash on the location of the feet. While TULip was standing, its feet were manually slid over the floor. By comparing the foot location of the motion capture recording and the internal data of the robot, backlash can be observed.
- The second experiment had a similar goal, but then in a more realistic situation. Now Tulip was standing on one foot, then a small push to the upper body was applied so that the robot is forced to make a step in order to maintain balance.

These experiments were conducted to find the measurement error in the kinematic chain of the robot. As explained in Section 2-1-2 it is known that there is backlash present in some joints. This specifically holds for the hip joint, which has a large effect on the location of the feet. Additionally, due to a lack of sensors this error can not be measured by the robot, therefore this error is unknown to the robot. By using the motion capture system this error can be quantified.

3-1-2 Data Processing

Before discussing the results of the experiments, first the manipulations that were applied on the data are briefly discussed. These were necessary to be able to compare the internal data of TULip with the data of the motion capture system.

The motion capture system came with the Qualisys Track Manager software (Figure 3-2), that recorded the video from the cameras and reconstructed a 3D image of all the markers. It was also used to process the data. During post-processing a rigid body was defined for each limb of the robot. This is a set of three or more markers that are combined to increase the accuracy of the motion trajectories of each limb. It is difficult to quantify the accuracy, as it is affected by many factors, such as camera positioning, marker placement and the calibration process which is done before each experiment. However, the software indicates the residual, which is an estimate of the position error. For the ankle joints the average residual is 0.13 mm with a standard deviation of 0.07 mm.

The internal data of TULip was recorded using the Variable Tuning Dialog (VTD). This is an application that is specifically written for TULip, and can be used to monitor and manipulate TULip's variables in real time. Both the motion capture (MC) data and the internal (VTD) data of TULip are imported into Matlab. Then a number of manipulations were done to be able to compare the data:

- With the VTD all relevant data of TULip could be recorded. In the MC data however, only the marker locations were available, not the physical locations of the robot. Therefore first the ankle joints, hip joints, foot corner points and the location of the nertial Measurement Unit (IMU) were created using the respective markers.

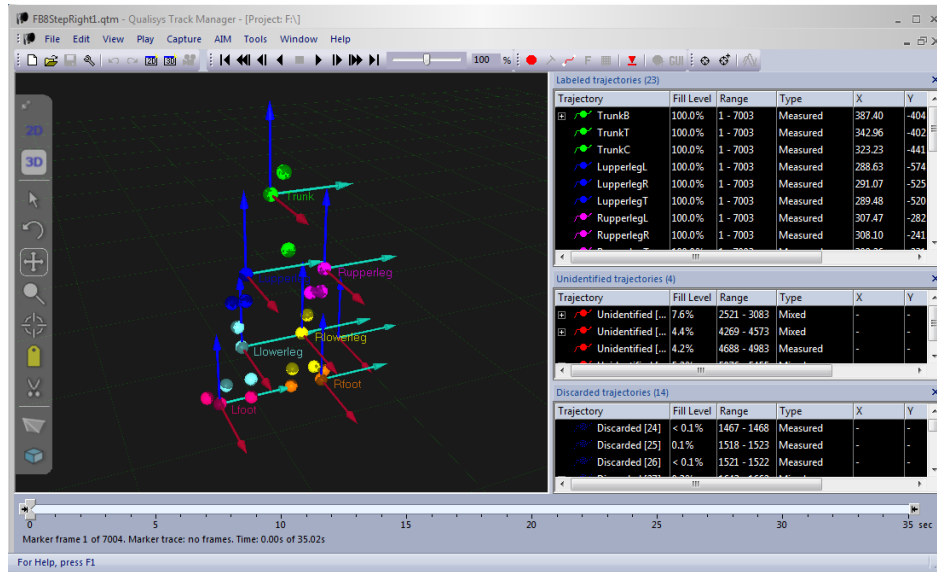


Figure 3-2: A screen capture of the Qualisys Track Manager software, which is used to collect and process the motion capture data.

- All data was then translated to the same frame of reference. For the following experiments the ankle joint of the stance leg was used.
- Next, both datasets had to be synchronised in both orientation and time. This was quite a difficult task since the time, the frame rate and the orientation in three angles differed. Finding a solution for all these variables at once was not possible, therefore it was done in several steps that are explained below. For all optimizations the Matlab command `fminsearch` was used.
 - First an initial estimate of the rotations was made based on the first frame of both data sets. This was done by minimizing the error between the foot corner points of the stance leg of both datasets. After the data was synchronised in time, the rotations were determined more accurately.
 - All VTD data was then rotated to the MC data according to the initial estimate.
 - For finding the time offset one of two methods was applied, depending on the experiment:
 - * The camera system is equipped with an external sync output, which sends a signal when the cameras start recording. This was connected to TULip and sent to the VTD, so it could be used to synchronize the datasets.
 - * Alternatively the time offset was found by minimizing the error of the top of the trunk and ankle joint of the swing leg, for the entire duration of the experiment. It proved to be more reliable than the external sync signal.

For experiment 1 the external sync output was used. This was done because backlash causes a delay between the real trajectory and the trajectory measured by the VTD (schematically illustrated in Figure 3-4). When an optimization would

be used to find the time offset, this delay would be removed, thereby diminishing the effects of backlash. In experiment 2 it was not intended to identify backlash, therefore the optimization was used for its reliability.

- Both datasets are interpolated using the right time window and frame rate.
- An improved estimate of the rotations was made. It was found that in some occasions the rotational error changes over time, therefore the angular offset was found for each frame in time.
- The rotational correction was applied to each frame.

Now that both are synchronised and expressed in the same frame of reference, they are further analyzed. The results are shown in the following sections.

3-1-3 Experiment 1: Backlash while Standing

In this experiment, the robot was standing on both feet. Then one foot was manually moved back and forth over the floor, while the other foot was kept in place. Throughout the experiment, each foot thus carried half of the weight of the robot. The foot position was measured with the ankle joint, using the ankle joint of the stance leg as frame of reference. The experiment was done in five movement directions for each foot, as shown in Figure 3-3.

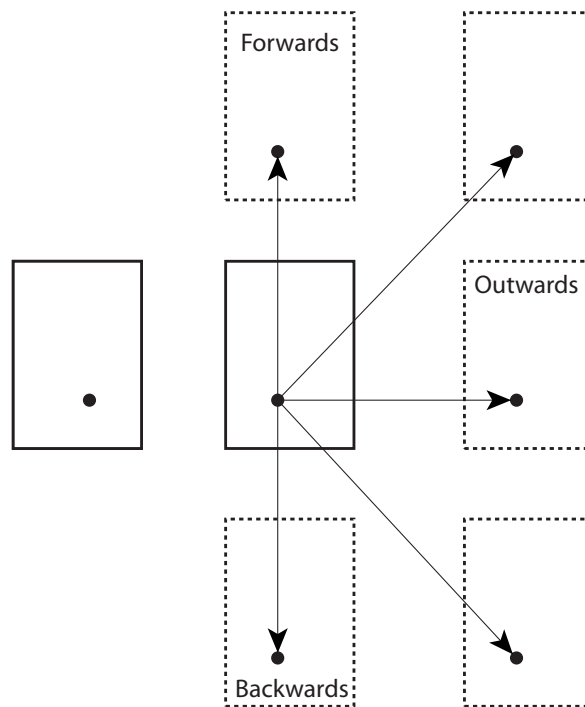


Figure 3-3: Schematic top view of the robot, showing the five movement directions of the right foot. Similarly, for the left foot the movements are mirrored.

By comparing the motion capture data with the data from the robot, the error in the foot location was used to identify the effects of backlash. This could then be used to calculate the backlash in the gearbox of the hip x joint.

The expected behavior of backlash is shown by an example in Figure 3-4. It shows that backlash causes both a delay and a measurement error. It also illustrates that the error is not constant, instead, the error is opposite of the movement direction. So if trajectory moves upwards the error is negative and when the trajectory moves downwards the error is positive. Thus in the experiment, where the foot is oscillated several times in different directions, the following error characteristics due to backlash are expected:

- The average error over multiple oscillations is close to zero.
- The error distribution displays two distinct peaks, one for each movement direction.

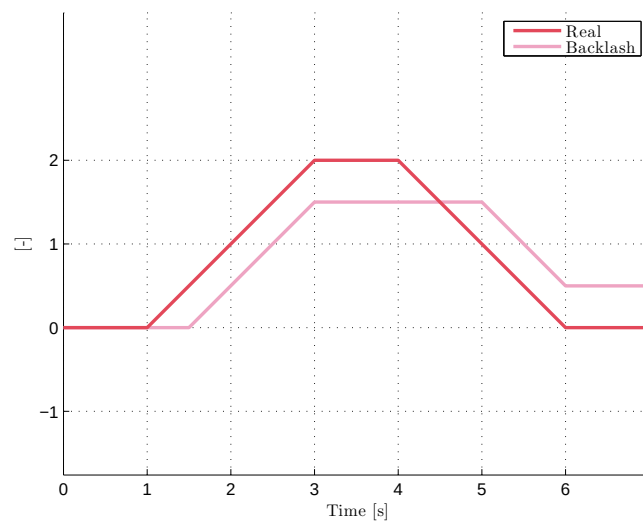


Figure 3-4: A schematic graph of the expected behavior of backlash. It consists of a trajectory and a measured trajectory under the influence of ± 0.5 backlash. At $t = 1$ the trajectory moves upwards, where backlash causes a delay in the measured signal. The movement stops at $t = 2$, the error then remains at -0.5 . When the trajectory moves downwards at $t = 3$, an error of 0.5 in opposite direction arises.

Results

The Figures 3-5 and 3-6 show the results for the right foot and the left foot, respectively. The data does not resemble the expected behavior of backlash, since a large constant offset is present. For the right foot there is no large error in x direction, but an average offset of -52 mm is observed in y direction. For the left foot an error in both directions is observed, with an average 44 mm in x direction and 54 mm in y direction. However, from backlash it was expected that the sign of the error depends on the movement direction, which means that for this experiment the average error should be close to zero. Therefore this constant error can not be caused by backlash in the joints, and must have another cause. It is expected that errors in the kinematic model are the source of this offset, however in this thesis this is not further investigated.

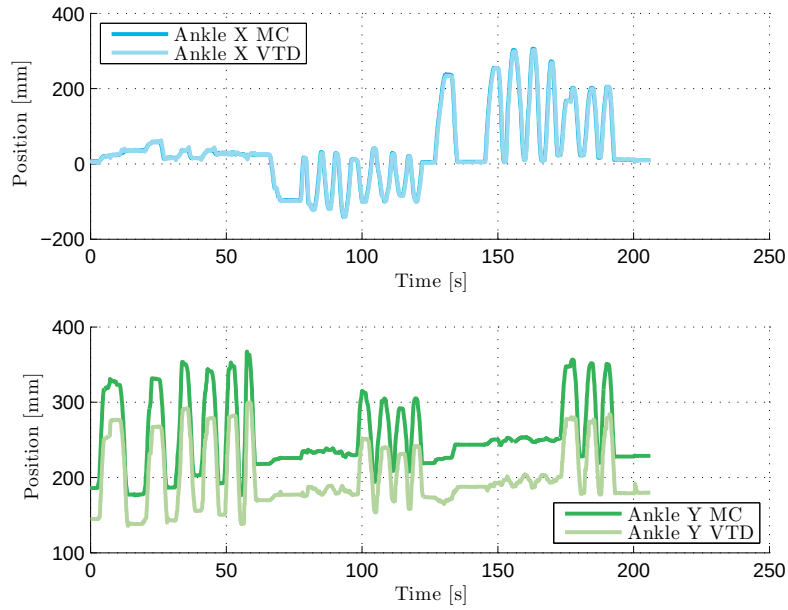


Figure 3-5: The trajectory of the right ankle is shown in x (top) and y direction (bottom), of both the motion capture (MC) measurement and the data from TULip (VTD). The trajectory is expressed relative to the left ankle, which is kept in place while the right foot successively moves outwards, forwards, forwards & outwards, backwards, backwards & outwards.

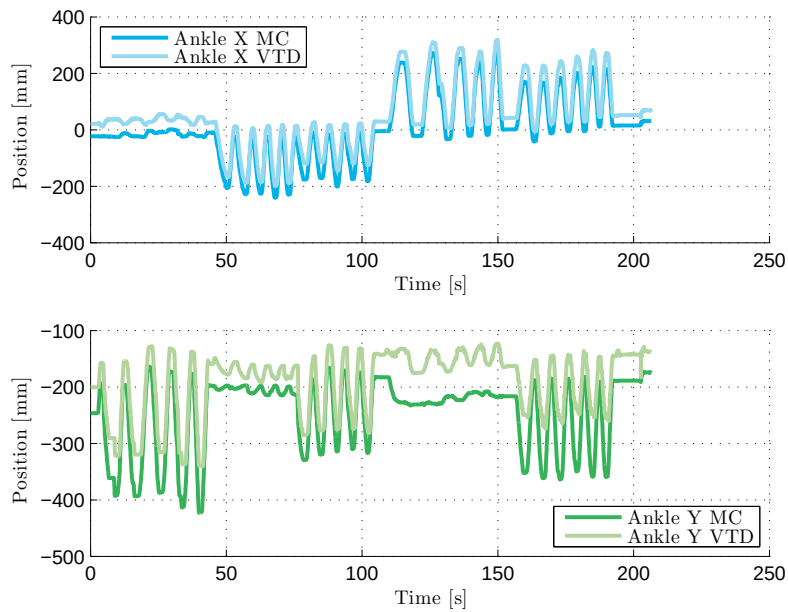


Figure 3-6: The trajectory of the left ankle is shown in x (top) and y direction (bottom), of both the motion capture (MC) measurement and the data from TULip (VTD). The right foot was kept in place and served as frame of reference.

The method of identifying backlash is explained using the first part of both datasets, where the foot was only perturbed outwards. The Figures 3-7 and 3-8 show the trajectory along with a normalized histogram of the error between the MC data and the VTD data. The histograms shows that the distribution of the error is bimodal. This corresponds to the behavior of backlash, that creates one distribution when moving away and another distribution when moving back, leaving a ‘dead zone’ in the middle. This problem was approached as a Gaussian mixture distribution. Two Gaussian functions were fit to each histogram, using the the Matlab function `gmdistribution.fit`. Each bar in the histogram was then colored according to the mode where it belongs to. Subsequently this color was added to the trajectory plot (Figure 3-7(a) and 3-8(a)), clearly showing the peaks in the distribution correspond with the movement direction of the foot.

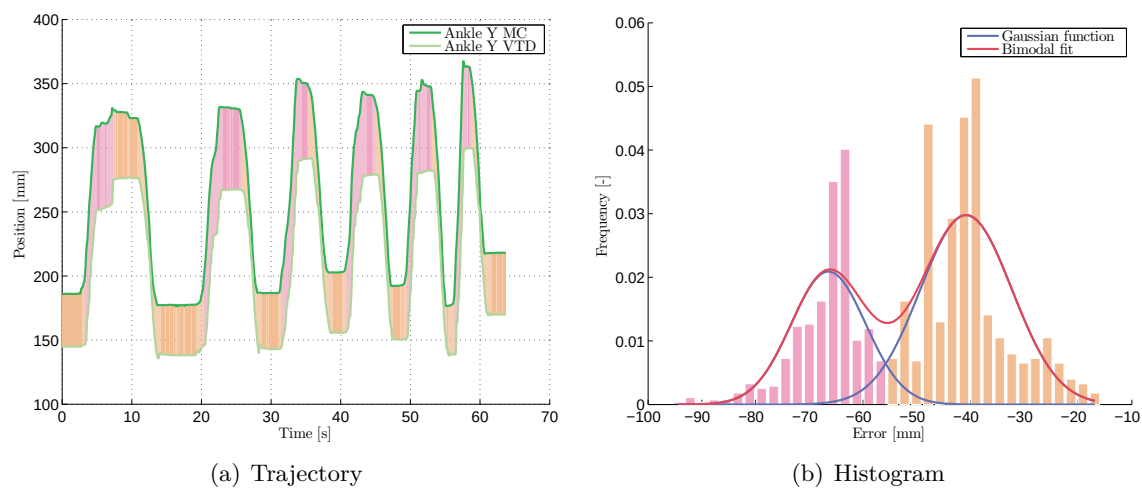


Figure 3-7: Trajectory of the right foot and the histogram of the error, while being perturbed in y direction (outwards). A bimodal distribution consisting of two Gaussian functions is fitted to the histogram. The colors in the trajectory plot correspond with the colors of the peaks in the histogram.

The distance between the two peaks in the histogram indicate the error caused by backlash. However, not for all directions the presence of backlash could be observed. The significance of backlash was determined using the mixture proportion and the distance between the two peaks. The mixture proportion is a property of the mixture distribution, that indicates how the data is distributed among the modes. All results are summarized in Table 3-1, where the blanks indicate that a significant error due to backlash was not observed. Additionally, for the backwards movement backlash could not be observed.

For the right foot no significant backlash was observed in x direction, which also corresponds with Figure 3-5. Unexpectedly, the measurement of the robot was very accurate in this case, although it is not clear why. For the left foot backlash in x direction was observed in three occasions, with an average distance of 26 mm between the peaks of the bimodal distribution.

In y direction backlash was observed more often. This also corresponds with the hardware of the robot, since it is known that x rotation of the hip joint is problematic. The backlash for the outwards movement is also shown in Figure 3-7(a) and 3-8(b) for the right and left foot, respectively. When considering the backlash in y direction for all movement directions and

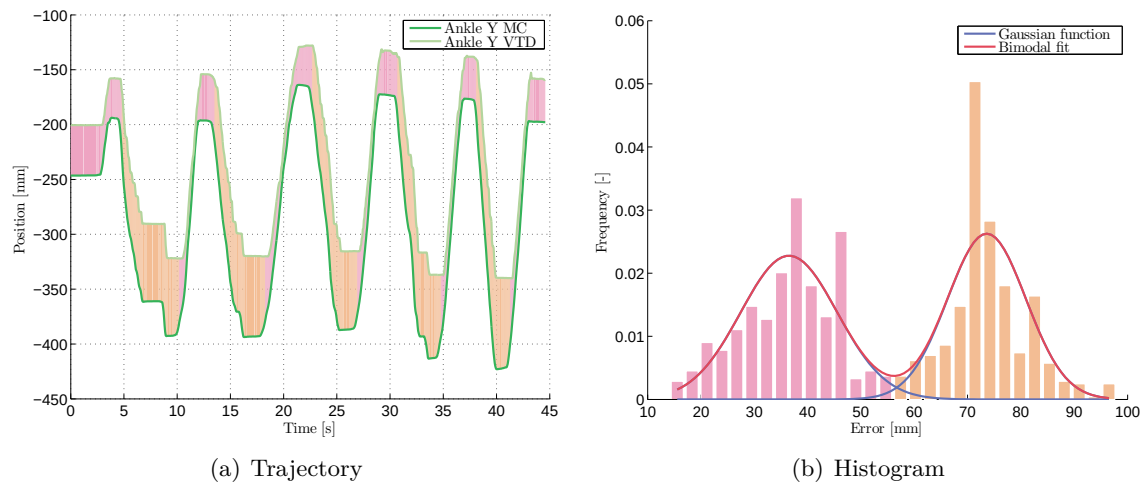


Figure 3-8: Trajectory of the right foot while being perturbed in y direction, and the histogram of the error. A bimodal distribution consisting of two Gaussian functions is fitted to the histogram.

both feet, it is found that the average backlash is with 32 mm, which is slightly larger than in x direction.

Due to the absence of an encoder on the hip x joint, the error due backlash in y direction is unknown to the robot. However, now that the error at the foot is known, the angular backlash of the hip x gearbox can be determined. Here the half of the measured error is used, since the error is measured from one foot to the other, and therefore two hip x joints contribute to the error. The leg length of TULip is 600 mm, with a 16 mm error the angular backlash of the gearbox at the hip is 1.5° . This corresponds with the backlash in planetary gearheads [12, 13].

Since the hip y actuator does have an encoder directly on the joint, the robot can sense the backlash in that gearbox. Therefore it is expected that the backlash that is measured at the foot in x direction (26 mm on average) is also caused by a combination of the hip x and z joint.

Direction of movement	Foot	Backlash x [mm]	Backlash y [mm]
Outwards	Right		26
	Left		37
Forwards	Right		
	Left	27	
Forwards & Outwards	Right		21
	Left	26	
Backwards & Outwards	Right		29
	Left	25	47

Table 3-1: Overview of the backlash observed during the experiment, for the movement directions where a significant backlash could be observed. The backlash was determined by the distance between the two peaks in the bimodal distribution.

Conclusion

With this experiment it was shown that there is a large error present in the perception of the feet. As shown in Figure 3-8(b), the error in the foot location can be over 80 mm. It was found that this error is caused by a combination of backlash and a constant bias.

Backlash was observed in some movement directions. It caused an average error at the feet of 26 mm and 32 mm for the x and y direction, respectively. Using these figures it was shown that the hip x gearbox has 1.5° of backlash.

Additionally, a constant bias was found. For the right foot there was no error in x direction, but in y direction this was -52 mm on average. For the left foot an error in both directions was observed, which was 45 mm in x direction and 55 mm in y direction.

Taking into account that the width of the feet is 127.5 mm, an error of over 80 mm is over half of the foot width. This has therefore a large effect on the accuracy of the support polygon, and is very harmful for stability of the robot.

3-1-4 Experiment 2: Backlash while Stepping

Now that the magnitude of the error in the perception of the foot location is known, it is interesting to see how it affects the performance in a more realistic experiment. Therefore a push recovery experiment was done where TULip was forced to make a step in order to maintain balance. The experiment consisted of the following events:

1. TULip was standing on both feet.
2. TULip was ordered to transition to single legged balancing.
3. When in single support, a small lateral push to upper body was applied, forcing TULip to make a step to maintain balance.

Again, the foot location are measurement between both ankle joints, using the stance leg as reference. The experiment was repeated ten times, five steps with the left foot and five with the right foot. However, due to the erroneous recording of a right step experiment, one dataset is omitted from the results.

Results

For each trial an instance before and after the step was picked where the robot was standing still. For these moments a comparison is made between the MC data and VTD data. Figure 3-9 shows such a result of a step with the left leg.

Subsequently, the error between the MC data and VTD data was determined, for the situation before the step and after the step. This error was then expressed relative to the orientation swing foot as measured by the motion capture system. This compensates for the error resulting from the rotational difference between the measurements. The resulting errors in x and y direction of all trials are listed in Table 3-2 and 3-3. Similar to the previous experiment a large error between the VTD data and the motion capture measurement is observed.

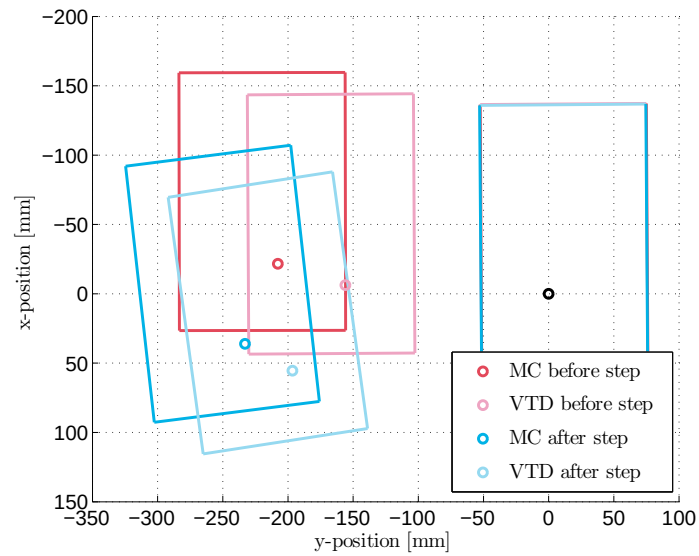


Figure 3-9: Top view of the feet before and after a left step (trial 1). In red the positions before the step are shown, the positions after the step are shown in blue. Furthermore, both the MC and VTD data are shown. A large error is observed both before and after the step.

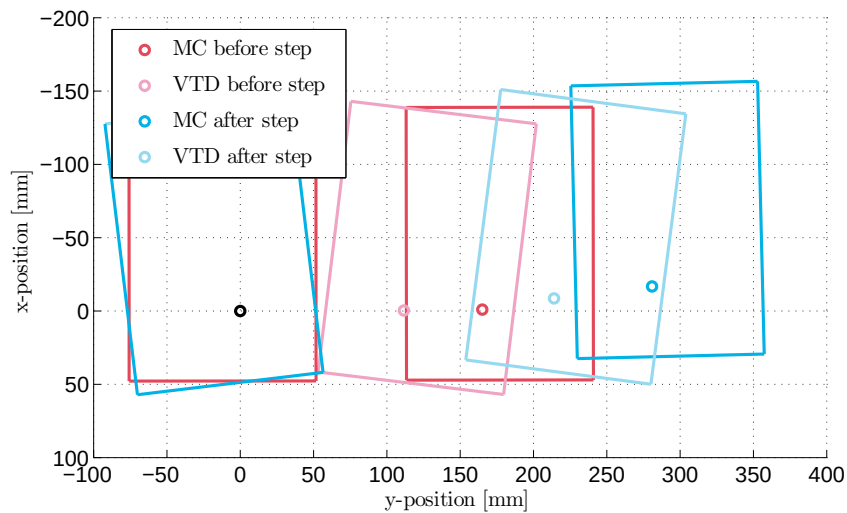


Figure 3-10: Top view of a step with the right foot (trial 7). It shows that in this case, both before and after the step a rotational error is present.

There is no significant difference observed between before and after the step. In x direction for the left step the error is 19 mm and is fairly consistent over multiple trials. For the right step however, a surprising result is found, since the robot is quite accurate in most attempts. Only in trial 8 a large error is observed, for the remaining trials the error is much smaller. This is striking, because in the previous experiment the robot also performed very well with the right foot in x direction. For the y direction again a very large error is observed for both feet, with an average of 47 mm for left steps, and -52 mm for right steps. This also corresponds with the previous experiment.

The tables also show that the measurement of the robot makes an underestimation of the actual foot location in y direction. This means that the foot is farther away from the stance foot than what the robot has measured. This may be caused by backlash, however the previous experiment showed that the backlash in y direction is 32 mm on average. Therefore there must be an additional constant bias present, as with the previous experiment.

Trial	Before Step		After Step	
	Error x [mm]	Error y [mm]	Error x [mm]	Error y [mm]
1	16	52	24	34
2	23	34	26	43
3	25	48	31	70
4	16	53	23	55
5	-4	38	10	39

Table 3-2: Errors of the ankle position in the left step experiments, relative to the swing foot. When combining the measurements before and after the step, the following averages and standard deviations are found: $\mu_x = 19 \pm 10.0$ mm, $\mu_y = 47 \pm 11$ mm.

Trial	Before Step		After Step	
	Error x [mm]	Error y [mm]	Error x [mm]	Error y [mm]
6	-16	-49	6	-28
7	1	-54	7	-67
8	50	-56	47	-37
9	2	-54	8	-69

Table 3-3: Errors of the ankle position in the right step experiments relative to the swing foot. The averages and mean with the errors before and after the step combined: $\mu_x = 13 \pm 23$ mm, $\mu_y = -52 \pm 14$ mm.

During some trials also a large rotational error was observed, as listed in Table 3-4 and 3-5 for the steps to the left and right. It appears that there are two distinct groups; the absolute error is either around 0° or more close to 10° . When splitting the results in these the groups, the it is found that the absolute averages are 0.9° and 8.7° . This large error might be caused by the hip z joint, which also lacks an encoder that can measure the actual leg orientation. However, the rotational difference is too large if it would be caused by backlash, therefore no clear cause has been found.

So, this experiment shows that the average error in the foot location is 17 mm in x direction,

Trial	Before Step Angular error [°]	After Step Angular error [°]
1	0.2	1.5
2	1.5	-0.3
3	1.7	1.8
4	-0.2	-0.7
5	8.6	8.7

Table 3-4: Angular error in the left swing foot in VTD, relative to the swing foot in MC. It shows that for four trials not error is present, except for the last one.

Trial	Before Step Angular error [°]	After Step Angular error [°]
6	-0.7	0.8
7	-7.0	-8.9
8	-11.0	-10.0
9	-8.0	-7.8

Table 3-5: Angular error in the right swing foot. Here in three of the trials a large angular error was observed.

and 49 mm in y direction. This is similar to the previous experiment. Additionally it is found that a rotational error of up to 11° can occur during stepping.

3-1-5 Conclusion

With the motion capture experiment it was shown that a large error occurs in the measurement of the foot location. In the first experiment a backlash of 26 mm and 32 mm was found for the x and y direction, respectively. This corresponds to a 1.5° of backlash in the gearbox of the hip x joint. Additionally, a large constant bias was observed. This was in the order of 50 mm for both the x and y direction. The source of this error could not be identified, but it is expected that it is caused by errors in the kinematic model of TULip. In the second experiment, errors of a similar magnitude were observed while the robot was making a step. Additionally, it was shown that there is also a rotational error present.

3-2 Accuracy of the CoM Calculation

The step location is calculation using Capture Points, which heavily rely on the center of mass estimation of the robot. First this Capture Point prediction is used to determine the maximum allowable error. Subsequently, an experiment with a force plate is done to measure the position and velocity of the CoM. By comparing the actual CoM and the CoM that is calculated by the robot, the error in the step location prediction can be determined.

3-2-1 Maximum Allowable Error

One of the most important uses of the CoM position and velocity, is the step location calculator. When the robot is walking or performing push recovery by stepping, the right step location has to be calculated in order to maintain balance. On Tulip this is done by using the Capture Point theory (Section 2-2). In this section this will be used to define the maximum error that is allowed on the CoM position and velocity.

The Instantaneous Capture Point (ICP) is calculated using both the CoM position and velocity. Furthermore, it is used to predict the step location at the beginning of the step. The equation used for this analysis is found by substituting (2-2) into (2-3). $\mathbf{x}_{ankle}$ is used as frame of reference and therefore set to zero. Focusing on the one dimensional case, this results in:

$$x_{icp}(\Delta t) = \left(x_{com}(0) + \frac{\dot{x}_{com}(0)}{\omega_0} \right) e^{\omega_0 \Delta t}. \quad (3-1)$$

So the capture point location at Δt depends on the CoM position and velocity at $t = 0$. To find the effect of an incorrect CoM measurement on the ICP prediction, a CoM position error $\delta_{com}(0)$ and velocity error $\dot{\delta}_{com}(0)$ is substituted, giving:

$$x_{icp}(\Delta t) + \epsilon_{icp}(\Delta t) = \left(x_{com}(0) + \delta_{com}(0) + \frac{\dot{x}_{com}(0) + \dot{\delta}_{com}(0)}{\omega_0} \right) e^{\omega_0 \Delta t}. \quad (3-2)$$

Here $\epsilon_{icp}^x(\Delta t)$ is the error in the ICP prediction at Δt , resulting from an incorrect CoM estimation. When solving, this results in:

$$\epsilon_{icp}(\Delta t) = \left(\delta_{com}(0) + \frac{\dot{\delta}_{com}(0)}{\omega_0} \right) e^{\omega_0 \Delta t}. \quad (3-3)$$

This is a single equation with two unknowns. Therefore, for finding the maximum allowable error in the CoM position, it is assumed that $\dot{\delta}_{com}(0) = 0$:

$$\epsilon_{icp}^x(\Delta t) = \delta_{com}(0) \cdot e^{\omega_0 \Delta t}. \quad (3-4)$$

Likewise, it is assumed that $\delta_{com}(0) = 0$ to find the effect of an erroneous CoM velocity:

$$\epsilon_{icp}^{\dot{x}}(\Delta t) = \frac{\dot{\delta}_{com}(0)}{\omega_0} \cdot e^{\omega_0 \Delta t}. \quad (3-5)$$

In the case of Tulip, the variables ω_0 and Δt are known constants, as listed in Table 3-6. ω_0 is easily calculated using the CoM height, which is fairly constant throughout a step. The step time is a parameter in the current step generator of the robot. Substituting these properties, it is found that:

$$\epsilon_{icp}^x(\Delta t) = 18.8 \cdot \delta_{com}(0), \quad (3-6)$$

$$\epsilon_{icp}^{\dot{x}}(\Delta t) = 4.8 \cdot \dot{\delta}_{com}(0). \quad (3-7)$$

Variable	Description	Value
z_0	CoM height	0.64 [m]
ω_0	Reciprocal time constant of the LIPM	$\sqrt{\frac{g}{z_0}} = 3.918 \text{ [s}^{-1}\text{]}$
Δt	Step time	0.75 [s]

Table 3-6: Relevant properties of the robot TULip, used for the analysis of the predicted ICP error, resulting from a inaccurate CoM.

So this means that at the start of the step, the error in the predicted ICP is 18.8 times larger than the error of the CoM position. Similarly, the error in the predicted ICP is 4.8 times larger than the error in the CoM velocity.

Ideally, the predicted ICP is sufficiently accurate so that if the robot steps towards the initially predicted ICP, it is still within its support polygon. For the feet of TULip, the minimal distance between the ankle joint and the edge of the foot is 48 mm, therefore

$$\max \epsilon_{icp}(\Delta t) = 48 \text{ [mm]}. \quad (3-8)$$

This means that the error of the CoM position and velocity must be:

$$|\delta_{com}| < 2.6 \text{ [mm]} \quad \text{if } \dot{\delta}_{com}(0) = 0, \quad (3-9)$$

$$|\dot{\delta}_{com}| < 10 \text{ [mm/s]} \quad \text{if } \delta_{com}(0) = 0. \quad (3-10)$$

Thus, the step location prediction of TULip is used to quantify the maximum allowable error. It is found that the error in CoM position must be less than 2.6 mm and the error in CoM velocity must be less than 10 mm/s. If this is not the case, the robot would fall if it would step towards the initially predicted Capture Point.

3-2-2 Force Plate Experiment

To measure the true CoM position and velocity, the robot was placed on a force plate at the VUMC in Amsterdam (Figure 3-11), while the robot was given several simple balancing tasks. The force plate is an AMTI OR6-5-1000 [19], which uses strain gauges as sensing elements and has the dimensions of 464 x 508 mm. With the measurements from the force plate, the location of the CoP can be determined. This is different from the CoM, but by keeping the inertial forces low it is assumed that the CoP location is equal to the floor projection of the CoM. With this experiment the accuracy of the current CoM measurement is assessed.

In the experiment, TULip was standing on the force plate in double support. Then a sinusoidal reference trajectory was used to move the robot back and forth while it was standing. This was done slowly, with a frequency of 0.1 Hz, to keep the accelerations at a minimum. The reference trajectory was applied using the ICP controller, since it was readily available. As the robot is moving, the force plate is able to measure the changing CoP position.

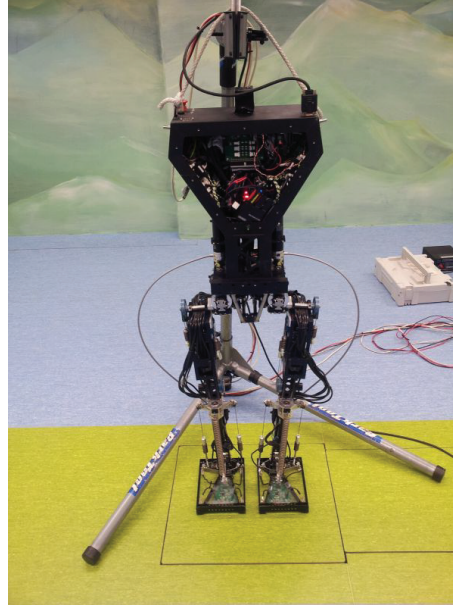


Figure 3-11: Photograph of the experimental setup. The robot is standing on the force plate (square on the floor) at the VUMC. It is protected from falling by the frame behind the robot. During the experiments the cables are slack.

Data Processing

The data from the force plate can not be directly compared to the data from the robot. As with the motion capture data, it first had to be synchronised with the VTD data. After importing both datasets into Matlab, the following operations were done.

The force plate measures the forces (F) and moments (τ) in x , y and z direction, that are exerted by the robot. These are recorded at 100 Hz and will be used to calculate the CoP of the robot with respect to its feet. The origin of the force plate is slightly below the floor level. This height offset, and thus z_{cop} , is therefore known in advance. Subsequently, the CoP position in x and y direction are calculated with

$$x_{cop} = \frac{-(\tau_y - F_x \cdot z_{cop})}{F_z}, \quad (3-11)$$

$$y_{cop} = \frac{(\tau_x - F_y \cdot z_{cop})}{F_z}. \quad (3-12)$$

The CoP is then transformed to the frame of the stance foot of the robot. During the experiments the location of the feet relative to the force plate were manually measured so that the exact offset between the origin of the force plate and the ankle joints is known.

Next the CoP is numerically differentiated to find the CoP velocity. Again, here it is assumed that this corresponds to the CoM velocity by keeping the inertial forces at a minimum. However, due to noise in the CoP position, large spikes arise in the differentiated signal. Therefore the CoP position is filtered before differentiating to remove high frequency noise. This is done with a second order Butterworth filter with the cut-off frequency at 0.5 Hz. This

is quite low, but since the robot is oscillated at 0.1 Hz, no relevant data is lost. A comparison of the filtered and unfiltered signals is made in figure 3-12, where it is shown that by filtering the CoP position, the CoP velocity contains much less noise. A drawback of filtering is that it creates a delay, which is clearly visible in the CoP position (figure 3-12(a)). However, the force plate data still has to be synchronised with the VTD data. By using the filtered signal for this, the delay is effectively removed.

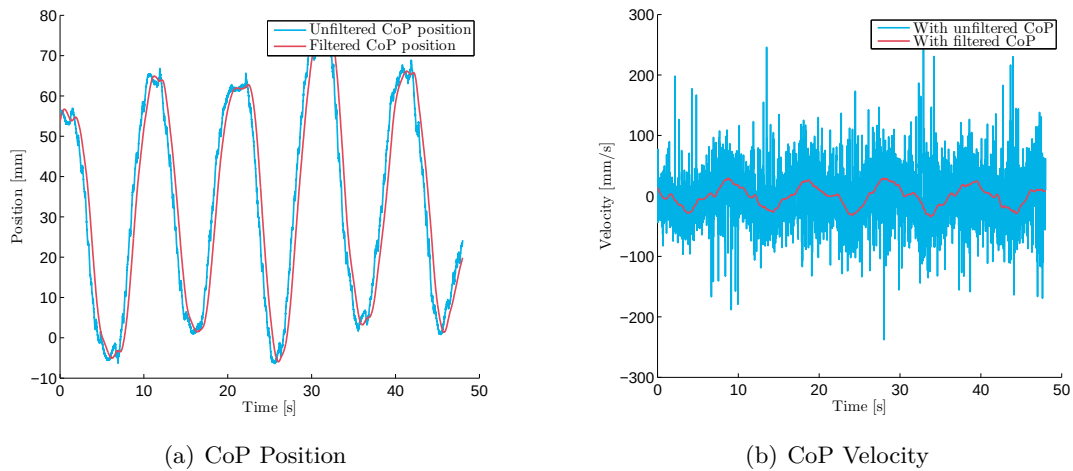


Figure 3-12: The plots show the difference between the unfiltered and filtered CoP data. After filtering the CoP position (a) with a second order Butterworth filter at 0.5 Hz, only minor differences are visible. However, when differentiating the filtered CoP position this results in a much smoother CoP velocity signal (b).

The last step was synchronising both datasets in time. This was done by minimizing the squared error between the filtered CoP position of the force plate and the CoM position of the VTD data, by shifting the time offset. After the time offset was found, the datasets were clipped accordingly.

Experimental results

During the experiment, the robot was standing in double support on the force plate, while being slowly oscillated back and forward with a frequency of 0.1 Hz. Figure 3-13 shows the CoP position measured by the force plate and CoM position that is calculated by the robot in x direction. The CoM trajectory follows the CoP trajectory, but shows a large constant bias. The average error is 23 mm, with a standard deviation of 1.1 mm.

The CoM velocity is compared to the differentiated CoP in Figure 3-14. It clearly shows that the current CoM velocity calculation is really inaccurate. It does not follow the actual CoP velocity at all, and it contains noise. The average error is 2 mm/s, with a standard deviation of 21 mm/s. During the experiment the CoM velocity was kept low so that the CoP would be similar to the CoM. However, it is expected that the error in CoM velocity would increase for higher velocities. Unfortunately, it is not possible to test this with a force plate.

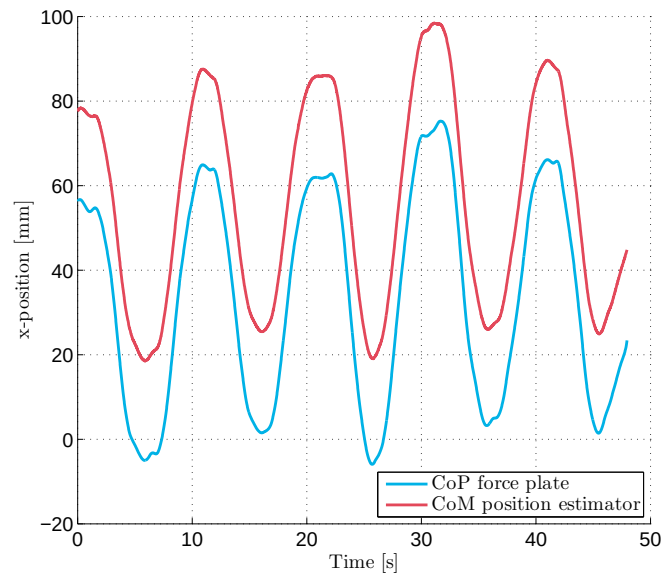


Figure 3-13: While the robot was in double support, a sinusoidal ICP trajectory was used to slowly oscillate the robot in x direction. In the graph the current position estimator is compared to the CoP position measured by the fore plate. A constant error of 23 ± 1.1 mm is observed with respect to the CoP position.

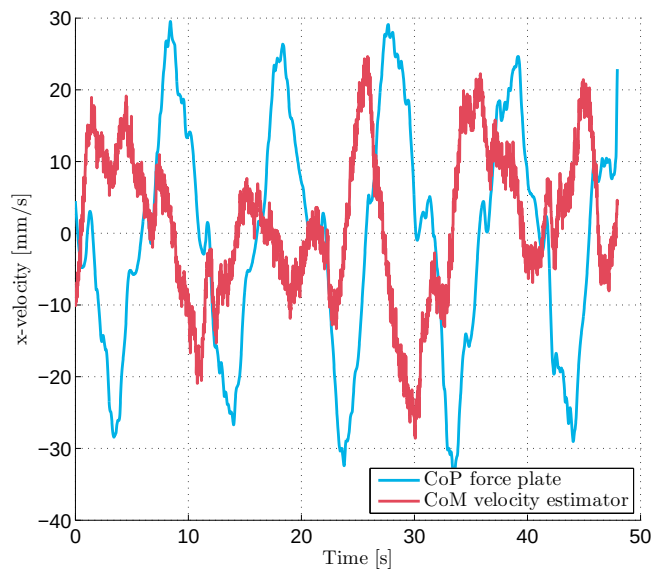


Figure 3-14: Comparison of the current estimator with the differentiated CoP position, under the same conditions as with Figure 3-13. The CoM velocity error is currently 2 ± 21 mm/s.

3-2-3 Effect of the CoM estimation on the ICP prediction

Previously it was shown that, in order to make an accurate prediction of the step location, the error on the CoM must be really small. Subsequently, with the force plate experiments it was shown that there is a large error present in the measurement of CoM position and velocity. By combining these, the effect of the current CoM calculation on the ICP prediction can be determined.

In Section 3-2-1 an error margin was established for the CoM position and velocity for making an accurate initial prediction of the step location. Using the properties of Tulip (Table 3-6), Equation (3-7) stated that:

$$\begin{aligned}\epsilon_{icp}^x(\Delta t) &= 18.8 \cdot \delta_{com}(0), \\ \epsilon_{icp}^{\dot{x}}(\Delta t) &= 4.8 \cdot \dot{\delta}_{com}(0).\end{aligned}$$

By substituting the CoM error found in the force plate experiment, the error in the ICP prediction is found. This is done separately for the position and velocity of the CoM, resulting in

$$\begin{aligned}\epsilon_{icp}^x(\Delta t) &= 436 \pm 21 \text{ [mm]}, \\ \epsilon_{icp}^{\dot{x}}(\Delta t) &= 10 \pm 102 \text{ [mm]}.\end{aligned}$$

Thus the error in the CoM position estimation result in a large, but fairly constant error in the ICP prediction. The CoM velocity error on the other hand, results in a small average error, but with a large standard deviation. This is also visualized in Figure 3-15.

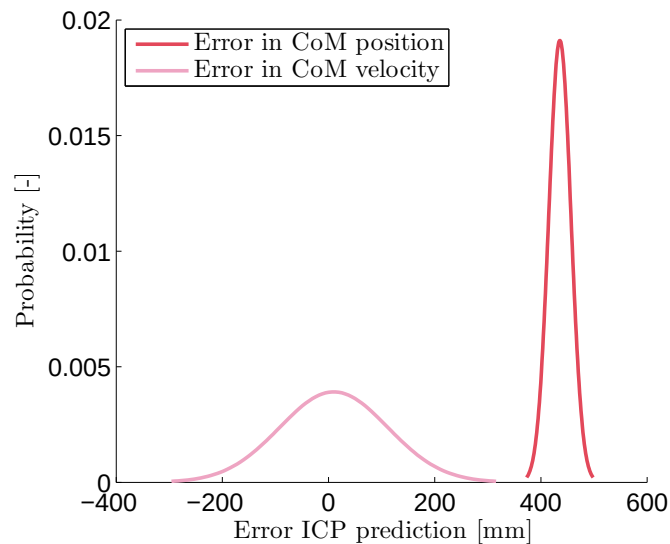


Figure 3-15: Plot showing the Gaussian probability distributions of the error in the ICP prediction that arises due to an incorrect CoM position and CoM velocity.

Both error sources have an undesirable effect on the robot while it is attempting to make a step. The CoM position error is causing a large initial error in the ICP prediction. But

the ICP is updated at each cycle, which makes the prediction more accurate as the step progresses. However, for an optimal swing leg trajectory, the optimal step location must be known as accurately as possible.

On average the CoM velocity error has little effect on the ICP prediction. However, the large deviation resulting from the noisy CoM velocity, causes the Capture Point to move around during the step. Again, this makes an optimal swing leg trajectory more difficult. Additionally, on foot impact this may cause the robot to start shaking, since the ICP is used to control the posture. Both of these phenomena are observed in experiments by other students.

Thus, it is shown how the initially predicted step location is affected by an incorrect CoM estimation. It was found that the average error is 436 mm resulting from a CoM position error. The CoM velocity error causes a relatively small average error, but with a standard deviation of 102 mm, which results in an unsteady ICP prediction. Therefore both errors have a negative effect on the ICP prediction.

3-2-4 Conclusion

For the accuracy of the step location, first an analysis was made of the maximum allowable error. Using the Capture Point prediction and the properties of TULip, it was found that the CoM position error must be smaller than 2.6 mm and the CoM velocity error must be smaller than 10 mm/s. This way the initially predicted Capture Point is sufficiently accurate.

Through an experiment where the robot was placed on a force plate, it was shown that current error in the CoM position and CoM velocity are 23 ± 1.1 mm and 2 ± 21 mm/s, respectively. Finally, it was shown that this results in an error in the predicted ICP of 436 ± 21 mm due to the CoM position error, and 10 ± 102 mm due to the CoM velocity error.

3-3 Goal for this thesis

In this chapter, two experiments were described. In the first experiment a motion capture system was used to analyse the perception of the foot location. In the second experiment, a force plate was used to measure the accuracy in the CoM position and velocity. With both experiments errors were found that hamper the performance for a walking robot.

First it was shown that an error of up to 80 mm is present in the perception of the feet. This is partially caused by backlash in the hip joint, which causes an error in the foot location of 26 mm and 32 mm for the x and y direction. Additionally, an unidentified bias in the order of 50 mm was observed for both the x and y direction.

With the second experiment, the accuracy of the CoM position and velocity calculation was investigated. Using a force plate it was found that there is a CoM position error of 23 ± 1.1 mm and a CoM velocity error of 2 ± 21 mm/s. It was shown that both errors cause large inaccuracies in the prediction of the step location.

Significantly improving the perception of the feet is difficult, because for a large part this is caused by mechanical deficiencies. Some performance gains could be gained by mounting

additional sensors, so that the backlash can be measured. However, backlash would then still be present, so for an optimal performance the hip joints would have to be redesigned.

Therefore, it was decided to focus on improving the CoM estimation for this thesis. This will result in a more accurate and steady ICP prediction, which contributes to a more robust walking and balancing robot.

Current Calculation of the Center of Mass

In the previous chapter it was shown that there is a large error in the calculation of the center of mass position and velocity. This chapter explains how the center of mass is currently calculated in the robot TULip. Subsequently the flaws in the current calculation are pointed out, so that a better estimator can be designed.

All variables throughout this thesis are denoted with subscripts and superscripts to indicate what it represents and what is the frame of reference. Furthermore, vectors and matrices are indicated with a boldface letter in lower case or upper case, respectively. For instance, $\dot{\mathbf{x}}_{com}^{trunk}$ is a vector containing the velocity of the center of mass relative to the trunk.

4-1 Calculation of the CoM Position

The calculation of the CoM position ($\mathbf{x}_{com}$) is quite straightforward. For each rigid body (b) it fetches the mass and CoM position with respect to the trunk. These are all added to the cumulative CoM vector. By dividing this vector by the total mass the CoM position with relation to the trunk is found:

$$\mathbf{x}_{com, mass}^{trunk} = \frac{\sum_{i=1}^b \mathbf{x}_i^{trunk}}{\sum_{i=1}^b m_i}. \quad (4-1)$$

Subsequently, a second order Butterworth filter with cross-over frequency of 20 Hz is applied to remove high frequency noise in the CoM position.

4-2 Calculation of the CoM Velocity

The current calculation of the CoM velocity consists of three parts, each separate from one another.

- The CoM position is differentiated, this gives the CoM velocity with relation to the trunk.
- The trunk velocity is found by using the kinematic chain from each foot.
- The effect of the angular velocity of the trunk on the CoM velocity is calculated.

Each part is described below, followed by an explanation of how the parts are combined to find the CoM velocity.

4-2-1 Differentiating the CoM Position

First, the CoM velocity ($\dot{\mathbf{x}}_{com, mass}^{trunk}$) is calculated by applying a first order differentiator on the CoM position with relation to the trunk,

$$\dot{\mathbf{x}}_{com, mass}^{trunk}(t) = \frac{\mathbf{x}_{com, mass}^{trunk}(t) - \mathbf{x}_{com, mass}^{trunk}(t-1)}{\Delta t}. \quad (4-2)$$

Again a second order Butterworth filter at 20 Hz is applied to remove high frequency noise.

4-2-2 Trunk Velocity from the Kinematic Chain

Next, the trunk velocity with relation to the feet ($\dot{\mathbf{x}}_{trunk, KC}^{foot}$) is determined. This is measurement from Inertial Measurement Unit (IMU).

All the differentiated joint angles in the kinematic chain from the foot to the trunk are considered to find the trunk velocity. Next, the trunk acceleration is measured with the IMU. Here a Kalman filter is then applied to estimate the velocity of the trunk. Here the **A** and **B** matrices of the forward dynamics are used to numerically integrate the acceleration of the IMU. An improved estimate is found by comparing this with the trunk velocity from the kinematic chain. This is done separately for the x and y direction, and for each foot.

4-2-3 Angular velocity trunk

In the last part of the CoM velocity calculation, the effect of the angular velocity of the trunk is considered. This is done using:

$$\dot{\mathbf{x}}_{com}^{trunk, rot} = \begin{bmatrix} -\omega_y c_{\omega_y, \hat{x}} \\ \omega_x c_{\omega_x, \hat{y}} + \omega_z c_{\omega_z, \hat{y}} \\ 0 \end{bmatrix}. \quad (4-3)$$

Here ω describes the angular velocity of the trunk about each axis, which is measured by the IMU. The c terms are estimated constants that represent the distance between the trunk and the CoM position.

In the current CoM calculation, all terms are combined to find the CoM velocity relative to a foot:

$$\dot{\mathbf{x}}_{com}^{foot} = \dot{\mathbf{x}}_{com, mass}^{trunk} + \dot{\mathbf{x}}_{trunk, KC}^{foot} + \dot{\mathbf{x}}_{com}^{trunkrot}. \quad (4-4)$$

4-3 Analysis of the Current State Estimation

Now that it is clear how the CoM position and velocity are calculated, an analysis is made of the current method. Here several improvements are proposed that will be used in designing an improved calculator in Chapter 6.

4-3-1 The Center of Mass Position

The CoM position is calculated by dividing the cumulative CoM vector by the total mass. The CoM is not a physical point on the robot, and it depends the orientation of each limb of the robot. Therefore this is the only way of determining the CoM position on TULip. However, this method is sensitive to both modelling errors and backlash. Modelling parameters, such as the mass and CoM of each limb, are tedious to determine as it requires that the robot is completely disassembled, with the copper wiring left intact. However, these parameters have a large effect on the CoM position. Backlash makes the measurement of the position of limbs inaccurate, which also causes an error in the CoM calculation. This especially holds for distant limbs such as the foot relative to the trunk, where a small angular offset at the hip results in a large error. It was also shown in Section 3-1-3, where it was found that 1.5° backlash in the hip joints results in an average error of 32 mm in the ankle position.

4-3-2 The Center of Mass Velocity

Also in the CoM velocity several errors may arise. It is currently calculated by the summation of three terms; the differentiation CoM position, the trunk velocity from the kinematic chain and the effect of the angular velocity in the trunk.

Using the kinematic chain for the velocity of the trunk relative to the feet, introduces needless errors. Since there is no link between the CoM and the trunk, the differentiated CoM position must be used nonetheless. Currently the CoM position with respect to the trunk is differentiated, but alternatively the CoM position with respect to the foot could be differentiated. This would give the same result, except without the inaccuracy introduced by the kinematic chain.

The angular velocity of the trunk (4-3), seems to be the implementation of

$$\dot{\mathbf{x}}_{com}^{trunk,rot} = \boldsymbol{\omega} \mathbf{x}_{com}^{trunk}. \quad (4-5)$$

Here $\boldsymbol{\omega}$ is a skew symmetric matrix containing the angular velocities of the trunk, and $\mathbf{x}_{com}^{trunk}$ is the distance between the trunk and the CoM. However, in the current calculator, this distance is an estimated constant, not the actual vector between the trunk and the CoM. Additionally, currently several terms in $\boldsymbol{\omega}$ are missing.

Regardless, the angular velocity also seems to be redundant. If the CoM position would change resulting from a rotation of the upper body, (4-1) changes over time. Since the CoM position is differentiated to find the CoM velocity relative to the trunk (4-2), angular velocity of the trunk is also accounted for.

Even though the CoM position calculation is sensitive to errors, it is expected that solely using the differentiated CoM position results in a more accurate calculation of the CoM

velocity. With the force plate experiment (Section 3-2-2) it was found that the current CoM position calculation follows the real CoM trajectory quite accurately, only with a constant bias. However, differentiating a constant error becomes zero, therefore the differentiated CoM position is expected to accurately follow the true CoM velocity.

4-4 Conclusion

This chapter explains how the CoM position and velocity are currently being calculated. Subsequently, an analysis is made which shows how the calculations can be improved. It was shown that the CoM velocity can be calculated by solely differentiating the CoM position with respect to the feet. Hence, the parts of the trunk velocity by the kinematic chain and angular velocity of the trunk can be omitted, resulting in a more accurate calculation of the CoM velocity. This will be used in the upcoming chapter to improve the estimation of the CoM.

Chapter 5

Simulation

For improving the CoM position and velocity on the robot, first several simulations were made. This was done to iteratively come to the design of the new CoM estimator for TULip. Additionally, this gave more insight to the performance and functioning of the Kalman filter. This chapter presents the final simulation and briefly discusses the results.

5-1 Modelling

Using Matlab, a simulation is made to estimate the CoM position and velocity of the LIPM. This is done for the one dimensional case, and the properties that are used resemble the robot as much as possible.

Similar to the ICP calculation (Section 2-2), the Linear Inverted Pendulum Model (LIPM) [20, 21] is used in the estimator. Additionally, this is used to determine the effect the ankle torque has on the dynamics of the CoM. The linearity of the LIPM is due to the assumption that the CoM height is constant and that there is no angular momentum present in the system. This is a requirement for the Kalman filter, since the model dynamics must be described in linear state space. Therefore the LIPM (Figure 5-1) is used to make the forward prediction in the estimator. For the one dimensional case the dynamics of the center of mass (x_{com}) are described with:

$$m\ddot{x}_{com} = \frac{mg}{z_0}x_{com} + \frac{\tau_a}{z_0}. \quad (5-1)$$

Here m denotes the total mass of the robot and z_0 is the constant CoM height. The ankle joint (x_{ankle}) is assumed to be fixed and used as frame of reference, and is therefore left out of the equation. Furthermore, τ_a represents the ankle torque, which is applied on the base of the LIPM.

All parameters of the simulation are listed in table 5-1, and are chosen similar to the properties of TULip. Rewriting the LIPM equation gives:

$$\ddot{x}_{com} = \omega_0^2 x_{com} + \frac{1}{mz_0}\tau_a, \quad (5-2)$$

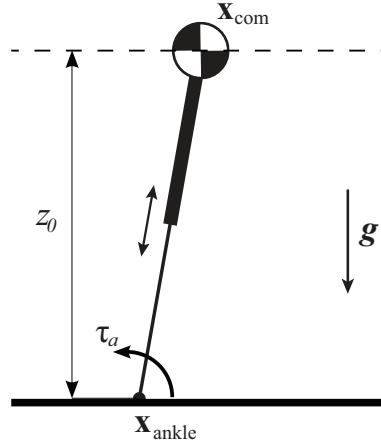


Figure 5-1: Drawing of the Linear Inverted Pendulum Model (LIPM). The model consists of a point mass on a fixed height (z_0) and a massless telescopic leg. The ankle torque (τ_a) is applied to the base of the pendulum.

where $\omega_0 = \sqrt{\frac{g}{z_0}}$. In simulation the discretized state space model is used. Alternatively the continuous version of the Kalman filter could have been used, but this better resembles the final implementation on TULip:

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

$$\begin{bmatrix} x_{com} \\ \dot{x}_{com} \end{bmatrix}_k = \begin{bmatrix} 1 & h \\ \omega_0^2 h & 1 \end{bmatrix} \begin{bmatrix} x_{com} \\ \dot{x}_{com} \end{bmatrix}_{k-1} + \begin{bmatrix} 0 \\ \frac{h}{mz_0} \end{bmatrix} \tau_{a,k-1}. \quad (5-3)$$

The sensor signals are mapped to the system states using a sensing model. For this simulation both the position and velocity of the CoM are used measurement signals:

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k$$

$$\begin{bmatrix} x \\ \dot{x} \end{bmatrix}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{com} \\ \dot{x}_{com} \end{bmatrix}_k. \quad (5-4)$$

To test the Kalman filter, three models are created using these equations of motion; a ‘real’ model that represents reality, a forward model that is used for the a priori prediction and a noise model that creates the noisy measurement signals. The Kalman filter will then use the forward model and the noisy measurement signals to approximate the real model.

First the real model is simulated, which was given a reference CoM position of 0.1 m for a simulation time larger than 0.2 seconds. Since the model is in an unstable equilibrium, a PD-controller generates an ankle torque (τ_a) to prevent it from falling. The gains are manually tuned and do not give an optimal control action. However, for this simulation this is not important. The ankle torque that is determined by the PD-controller will also be applied on the other models.

Next, the forward model is created. It is equal to the input matrix in (5-3), except it makes an overestimation of the ankle torque:

$$\mathbf{B}_{forward} = 1.5\mathbf{B}. \quad (5-5)$$

Parameter	Description	Value
T	Simulation time	3 [s]
h	Time step of discretization	0.0025 [s]
g	Gravitational constant	9.81 [m/s ²]
z_0	Constant height of the LIPM	0.64 [m]
m	Total mass of robot	16.6 [kg]
x_{ref}	Reference CoM position	0.1 [m] for $T > 0.2$ [s]
K_P	Proportional gain of controller	1000
K_D	Derivative gain of controller	75

Table 5-1: Parameters used in the simulation of the Kalman filter. These values resemble the robot TULip as much as possible.

The noise model has the same dynamics as the real model, with the addition of a normally distributed random signal:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + w_{k-1}, \quad (5-6)$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + v_k. \quad (5-7)$$

w_k and v_k represent the process noise and measurement noise respectively, and are generated with the Matlab command `randn`, using:

$$p(w) \sim N(0, \mathbf{Q}), \quad (5-8)$$

$$p(v) \sim N(0, \mathbf{R}). \quad (5-9)$$

Here $\mathbf{Q}$ and $\mathbf{R}$ denote the covariance matrix for each signal, which are chosen as:

$$\mathbf{Q} = \begin{bmatrix} 0.8583 & 0 \\ 0 & 0.8583 \end{bmatrix} \cdot 10^{-6} [\text{m}^2], \quad \mathbf{R} = \begin{bmatrix} 0.95 & 0 \\ 0 & 0.95 \end{bmatrix} \cdot 10^{-4} [\text{m}^2]. \quad (5-10)$$

These are chosen the same as the covariances of the Kalman filter that is currently used to estimate the trunk velocity on TULip.

5-2 Results

The Kalman filter uses only the forward model with the overestimation of the torque and measurement signals containing noise. It then attempts to find the CoM position and velocity as determined by the real model. Even though Matlab has the built-in functions `kalman` and `kalmd` to create a Kalman filter, it was decided to write a custom function based on the explanation in Section 2-3.

The result of the simulation is shown in figure 5-2. The overestimation of the forward model and the noise in the measurement signals are clearly visible. However, using this information, the Kalman filter is able to make an accurate prediction of the CoM position and velocity.

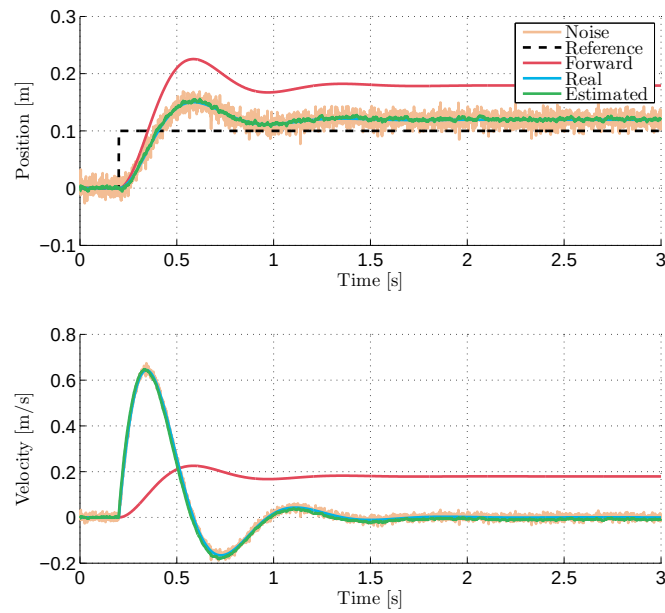


Figure 5-2: Result from the simulation of the Kalman filter that is applied on a LIPM. The graphs show the position (top) and velocity of the CoM (bottom). The Kalman filter used the noise signals and forward model to make an estimation of the real trajectory.

5-3 Conclusion

In simulation the Kalman filter is tested in combination with the Linear Inverted Pendulum Model. It uses fictitious measurement data of the position and velocity of the CoM, and the ankle torque to estimate the system states. This model will serve as the basis for the CoM estimator that will be implemented on TULip.

The New Center of Mass Estimator

This chapter describes the design of the new estimator, that predicts the CoM position and velocity for the robot TULip. First, a general description is given for improving the CoM estimation. Subsequently, the analysis of Chapter 4 is used to improve the calculations of the CoM, because these are used as input to the estimator. Finally, the forward model and sensor model that are used in the Kalman filter are presented.

6-1 General Description

On the robot TULip, the Kalman filter (Section 2-3) will be used to make a better estimate of the CoM position and velocity. This will be done only in x and y direction, because these are most important for balance control and the step location prediction.

The Kalman filter is able to find an estimate of the system states by combining multiple measurement signals that are under the influence of noise. The following four measurement signals will be used:

- $x_{com,mass}$, the CoM position calculated with the sum of all masses.
- $\dot{x}_{com,mass}$, the CoM velocity found by differentiating the CoM position.
- $\ddot{x}_{com,imu}$, the acceleration of the CoM measured by the IMU.
- τ_a , the ankle torque of the both feet.

The CoM position, velocity and acceleration will serve as a measurement input ($\mathbf{y}_k$). The contribution of the torque is incorporated in the forward model of the Kalman filter.

The measurement inputs of the Kalman filter will all be expressed relative to a foot. This is contrast to the current calculator, which estimates the CoM with respect to the trunk. However, when the CoM is determined relative to the foot, it better resembles the LIPM. But

more importantly, other algorithms that use the estimated CoM, such as the step location predictor, need CoM to be expressed relative to the stance foot. For the current estimator this means that the CoM must be transformed from the trunk to the stance foot, which may introduce additional inaccuracies. However, for the new estimator this is no longer required.

Additionally, the states of the CoM in x direction are independent of the variables in y direction and vice versa. Therefore the estimator is implemented for just one direction, which keeps the code better readable and adaptable. Thus, when the filter is implemented on the robot it will be instantiated four times, for the x and y direction and for both feet.

6-2 Calculation of the Measurement Signals

In Chapter 4 the current CoM calculator was analysed. It was found that in several calculations unnecessary errors were introduced. So for the new estimator these calculations are rewritten. Furthermore, two additional signals are used in the new CoM estimator, being the acceleration of the CoM and the torque applied at the ankles. This section will also show how these are determined.

6-2-1 The CoM position

The calculation of the CoM position is done the same as before (Section 4-1). However, the new estimator uses the LIPM where the states are expressed relative to the foot. Therefore the measurement signals must also be expressed in this frame. The equation now becomes:

$$\mathbf{x}_{com,mass}^{foot} = \frac{\sum_{i=1}^b \mathbf{x}_i^{foot}}{\sum_{i=1}^b m_i}. \quad (6-1)$$

The CoM position in z direction, and thus the CoM height, is also determined with this equation. However, since it is less important for the controllers of the robot, no improved estimation will be made.

6-2-2 The CoM velocity

In the current calculator, the CoM velocity consisted of three parts; the differentiated CoM position, the trunk velocity from the kinematic chain and the angular velocity of the trunk. In the analysis of the current calculator (Section 4-3), it was concluded that the last two parts can be omitted, as these introduce needless errors. So the new estimator will solely rely on the differentiated CoM position, which is also done relative to the foot:

$$\dot{\mathbf{x}}_{com,mass(t)}^{foot} = \frac{\mathbf{x}_{com,mass(t)}^{foot} - \mathbf{x}_{com,mass(t-1)}^{foot}}{\Delta t}. \quad (6-2)$$

6-2-3 The CoM acceleration

The acceleration of the CoM can be determined by using the measurement of the IMU. The location of the CoM is generally below the position of the IMU, as shown in Figure 6-1. The IMU measurement therefore makes an overestimation of the CoM acceleration. This is compensated for by using the LIPM and the fraction of heights:

$$\ddot{x}_{com,imu}^{foot} = \frac{z_{com}}{z_{imu}} \cdot \ddot{x}_{imu}^{foot}. \quad (6-3)$$

By using the LIPM, $\ddot{x}_{com,imu}$ is calculated under the assumption that there is a rigid connection between the foot, the CoM and the IMU. Due to the hip and knee joints of the robot this is obviously not the case. However, trunk has the largest contribution to the CoM position, since it weighs 10.8 kg [2] while total mass of the robot is 16.6 kg. Thus, if the trunk accelerates, it is likely that the CoM is also accelerating.

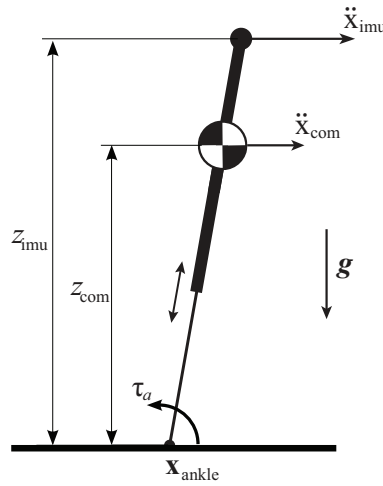


Figure 6-1: Illustration of the LIPM, indicating the height of the CoM and the location of the IMU. Due to the height difference, the acceleration does not correspond to the acceleration of the CoM.

6-2-4 The ankle torques

As shown in Section 2-1-1, the ankles are actuated using Series Elastic Actuation. This allows the use of force control, and as a result torque is also accurately measured. The ankle torques can thus be directly read out from the ankle controllers.

6-3 Modelling the Dynamics of the Robot

This section presents the forward model that is used in the Kalman filter to make a prediction of the next state. Subsequently, the sensor model is presented, based on the measurement signals of the previous section. Earlier it was explained that the estimator is implemented for

a single direction, and then instantiated multiple times. Therefore in this section the filter is described for the right foot in x direction, the filters for the left foot and the y direction work in a similar fashion.

6-3-1 The Forward Model

The Linear Inverted Pendulum Model (LIPM) [21, 20] is used to describe the dynamics of the robot. With the simulation (Chapter 5) it was already shown that the equations of motions of the LIPM (Figure 5-1) are linear. The equation of motion of the one dimensional LIPM is:

$$\ddot{x}_{com} = \omega_0^2 x_{com} + \frac{1}{mz_0} \tau_a, \quad (6-4)$$

where $\omega_0 = \sqrt{\frac{g}{z_0}}$ is the reciprocal time constant of the LIPM. τ_a represents the torque that is applied on the base of the LIPM.

In the LIPM only the ankle torque of one leg can be modelled. However, when the robot is standing with two feet on the ground, both ankles are able to exert a torque that can alternate the CoM. Thus for the estimator belonging to the right foot, both ankle torques are incorporated by using:

$$\ddot{x}_{com,R} = \omega_{0,R}^2 \cdot x_{com,R} + \frac{1}{mz_{0,L}} \tau_{a,L} + \frac{1}{mz_{0,R}} \tau_{a,R}. \quad (6-5)$$

Here the additional subscripts L and R denote the variables with respect to the left and right foot. When the robot would stand on an uneven surface, the CoM height with respect to each foot is different. Therefore $z_{0,L}$ and $z_{0,R}$ are calculated separately for each foot. If the robot would transition to right single support mode, and left foot loses contact with the ground, its ankle torque $\tau_{a,L}$ simply becomes zero.

Since a discrete Kalman filter will be used on TULip, the equation of motion is written as discrete state space equations. For clarity the subscript R is omitted in the state vector ($\mathbf{x}_k$) and the state matrix ($\mathbf{A}$). With h as the time step:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}u_{k-1} \\ \begin{bmatrix} x_{com} \\ \dot{x}_{com} \end{bmatrix}_k &= \begin{bmatrix} 1 & h \\ \omega_0^2 h & 1 \end{bmatrix} \begin{bmatrix} x_{com} \\ \dot{x}_{com} \end{bmatrix}_{k-1} + \begin{bmatrix} 0 & 0 \\ \frac{h}{mz_{0,L}} & \frac{h}{mz_{0,R}} \end{bmatrix} \begin{bmatrix} \tau_{a,L} \\ \tau_{a,R} \end{bmatrix}_{k-1}. \end{aligned} \quad (6-6)$$

Note that the parameters used in the $\mathbf{A}$ and $\mathbf{B}$ are not constant, instead, these are updated each time step. If, for instance, TULip is standing in double support and lifts up one leg, the CoM position will slightly move upwards. This will change z_0 and ω_0 in the dynamic model. Thus, to ensure that the LIPM always matches the actual dynamics of the robot as accurately as possible, these variables are calculated each time step and update in the model. This is analogous to the ICP calculation in Section 2-2.

6-3-2 The Sensor Model

The sensor model describes how the measurement signals correspond to the states in the forward model. In Section 6-2 it was already shown how the measurement signals are calculated

for the new estimator. However, the acceleration of the CoM is not available as state information. As a result, the acceleration measurement of the IMU can not be directly added to the sensor model. To solve this, the acceleration measurement is integrated. Subsequently, the sensor model (**C**) is presented.

The acceleration measurement signal, which is calculated as shown in Section 6-2-3, is integrated using the CoM velocity estimate (2-12) of the previous time step:

$$\dot{x}_{com,imu,k} = \dot{x}_{com,k-1}^+ + \ddot{x}_{com,imu,k} \cdot h. \quad (6-7)$$

Alternatively it would be possible to solely use the integrated acceleration, for instance with

$$\begin{aligned} \dot{x}_{com,imu}(0) &= 0, \\ \dot{x}_{com,imu,k} &= \dot{x}_{com,imu,k-1} + \ddot{x}_{com,imu,k} \cdot h. \end{aligned}$$

However, since the acceleration measurement from the IMU contains noise and possibly a small bias, integrating this signal would cause the velocity to drift. Therefore it is decided to use the CoM velocity estimate as shown in (6-7). This ensures that the velocity from the IMU is more accurate.

Similar to (2-5), now the sensor model can be constructed. With all the measurement signals known, this becomes:

$$\begin{aligned} \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k, \\ \begin{bmatrix} x_{com,mass} \\ \dot{x}_{com,mass} \\ \dot{x}_{com,imu} \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{com} \\ \dot{x}_{com} \end{bmatrix}. \end{aligned} \quad (6-8)$$

Finally, the process and measurement noise covariance matrices have to be set. However, at the time of the implementation, the true values of the CoM position and velocity were unknown. It was thus not possible to determine these covariances. It is therefore decided to use the same values as with the Kalman filter that is used in the old calculator to estimate the trunk velocity (Section 4-2-2):

$$\mathbf{Q} = \begin{bmatrix} 0.8583 & 0 \\ 0 & 0.8583 \end{bmatrix} \cdot 10^{-6} [\text{m}^2], \quad \mathbf{R} = \begin{bmatrix} 0.95 & 0 & 0 \\ 0 & 0.95 & 0 \\ 0 & 0 & 0.95 \end{bmatrix} \cdot 10^{-4} [\text{m}^2]. \quad (6-9)$$

6-4 Conclusion

First it is shown how the measurement signals are calculated. Subsequently, the design of the new estimator for the CoM position and velocity is presented. It is based on the Kalman filter and makes use of the LIPM as a forward model. The measurement signals that are combined are the position, velocity and acceleration of the CoM and the ankle torque.

Experimental Results

The new CoM estimator that was introduced in the previous chapter, is implemented on the robot TULip. It is based on the Kalman filter and Linear Inverted Pendulum Model, and is used to estimate the position and velocity of the CoM in x and y direction. In this chapter, the results from an experiment with a force plate illustrate the performance of the new estimator. The estimator is compared to the old CoM calculation. Subsequently, the Kalman filter is tuned using the measurement data from the force plate.

To test if the new Kalman filter is more accurate than the old calculator, an experiment was done where the robot is placed on a force plate. This is similar to the experiment in Section 3-2-2. While the robot was standing on the force plate, an oscillating reference trajectory of 0.1 Hz was used to slowly move the robot. From the data of the force plate the center of pressure (CoP) could be calculated. Since the robot is moving with low accelerations, it is assumed that the CoP is equal to the CoM position. Subsequently, the CoP is differentiated to be able to compare it to the CoM velocity. This was done separately in the x and y direction, and for both double and single support. The following sections show the results of the experiments for both support settings, followed by a section on the tuning of the Kalman filter.

7-1 Double Support

In the first experiment, the robot was standing in double support, while the ICP was oscillated in x direction. To keep the inertial forces at a minimum, a frequency of 0.1 Hz was applied. Figure 7-1 and 7-2 show the results for the CoM position and CoM velocity, respectively.

For the position, the difference between the old and new CoM estimator is marginal. In both cases a large constant error is present with respect to the CoP position.. For both estimators, the error is 23 mm on average, with a standard deviation of 1.1 mm.

In the velocity signal, a large improvement is observed for the new estimator. As also shown in Section 3-2-2, the average error in the old estimator is 2 mm/s with a standard deviation of 21 mm/s. For the new estimator, the average error is similar with 4 mm/s, but the standard deviation is now just 3 mm/s.

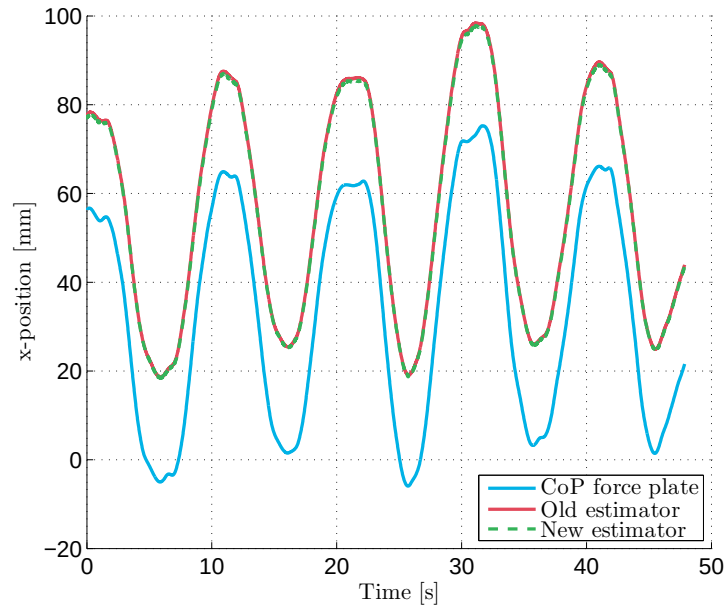


Figure 7-1: While the robot was in double support, a sinusoidal ICP trajectory was used to slowly oscillate the robot in x direction. In the graph the old and new CoM position estimations are compared to the CoP position measured by the fore plate. It shows that both estimators perform similar, with a large constant error with respect to the CoP position.

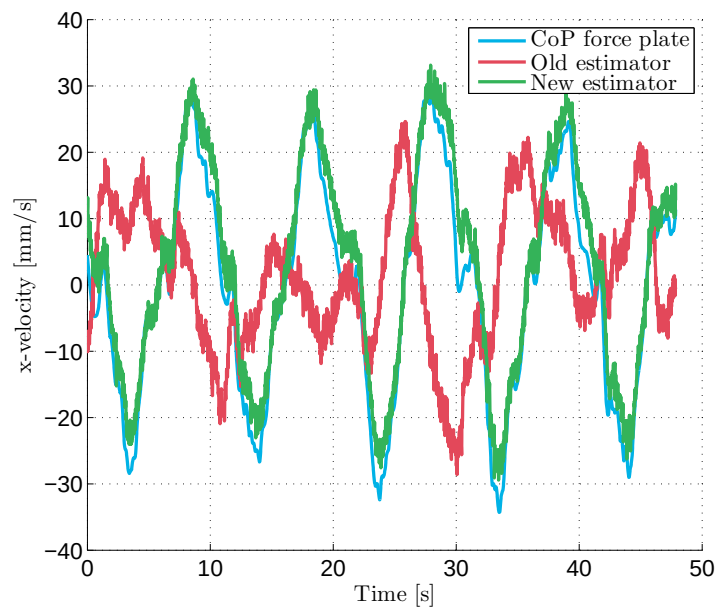


Figure 7-2: Comparison of the old and new CoM velocity estimations, and the differentiated CoP position, for the x direction in double support. It shows that there is a large error between the old estimator and the CoP velocity. However, the new estimator is much more accurate.

Analogous to Section 3-2-2, the effect of the erroneous CoM on the initial step location prediction is made. The results for the old and new estimator are summarized in Figure 7-3 and Table 7-1. It shows that due to the improved CoM velocity estimation, the standard deviation of the predicted ICP error is significantly smaller. The maximum allowable error in the ICP prediction was determined to be 48 mm (3-8), therefore the new CoM velocity estimation is sufficiently accurate for the prediction of the step location.

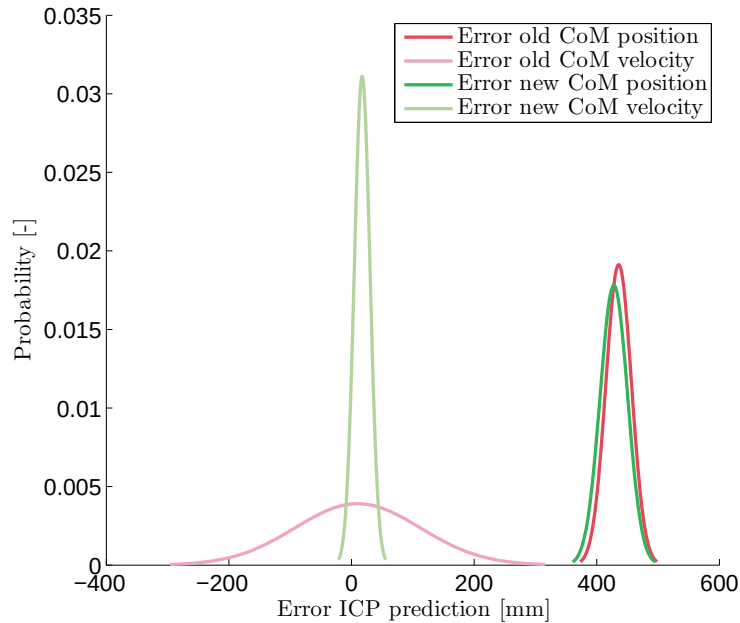


Figure 7-3: Probability distribution of the error on the predicted ICP in x direction, resulting from an inaccurate CoM position and velocity estimation. The new estimator shows only a minor improvement for the error due to the CoM position. However, the error due to the CoM velocity estimation has become much less. The corresponding averages and standard deviations are listed in Table 7-1.

Estimated signal	Initially predicted ICP error	
	Old estimator	New estimator
CoM position x	436 ± 21 [mm]	428 ± 22 [mm]
CoM velocity x	10 ± 102 [mm/s]	17 ± 13 [mm/s]

Table 7-1: Comparison of the old and new estimators, on the effect on the predicted step location. For the CoM position of little improvement is observed, but the CoM velocity has become much more accurate. Figure 7-3 shows the corresponding probability distributions.

The measurement of the CoM position and velocity in y direction was unsuccessful. Due to the sideways movement of the robot, the one of the feet lifted off the ground and caused the robot to start shaking. This resulted in an inaccurate CoP measurement by force plate. Additionally, due to the shaking, the location of the feet on the force plate changed. Since this was used as reference for both the force plate and the internal calculation of the CoM,

no reliable comparison of the estimators could be made.

7-2 Single Support

A similar experiment was conducted to test the performance of the estimator in single support. Here the right foot was used as stance leg. This way also the estimator in the y direction could be tested, since in single support the foot will not start shaking.

The CoM positions and velocities of the right foot in x direction are shown in the Figures 7-4 and 7-5, respectively. For clarity only 20 seconds is shown for the CoM velocity, the entire experiment lasted 81 seconds. The results are similar to the findings of the double support experiment. Thus a large, constant bias in both the old and new CoM position estimators. The new CoM velocity estimator is significantly more steady than the old one. The average errors and standard deviation for both the old and new estimators are listed in Table 7-2.

Estimated signal	CoM error	
	Old estimator	New estimator
CoM position x	8 ± 3 [mm]	8 ± 3 [mm]
CoM position y	-11 ± 1 [mm]	-11 ± 2 [mm]
CoM velocity x	3 ± 18 [mm/s]	6 ± 4 [mm/s]
CoM velocity y	1 ± 14 [mm/s]	-8 ± 8 [mm/s]

Table 7-2: The average errors and standard deviations of the CoM position and velocity estimators, for the experiment in single support. Again, it is shown that the improvement in the CoM position estimator is only minor. For the new CoM velocity estimation, the standard deviations are much smaller, which indicates a significant improvement.

This experiment was also conducted for the y direction on the right foot. In this case the difference between the old and new estimator is smaller, but still an improvement in accuracy is observed for the CoM velocity estimation.

As with the previous experiment, the effect of the CoM error on the step location prediction is investigated. The results for the x and y direction are listed in Table 7-3. Still, a improvement in performance is observed for the new CoM velocity estimator. However, both the average errors and standard deviations are larger, which makes it less probable that the error in the ICP prediction is below the threshold of 48 mm.

Thus, overall a smaller improvement is observed with the experiment in single support compared to the experiment in double support. However the trend is the same: minor improvements are found in the CoM position error, for the new CoM velocity estimator the standard deviation is smaller which results in an improved ICP prediction.

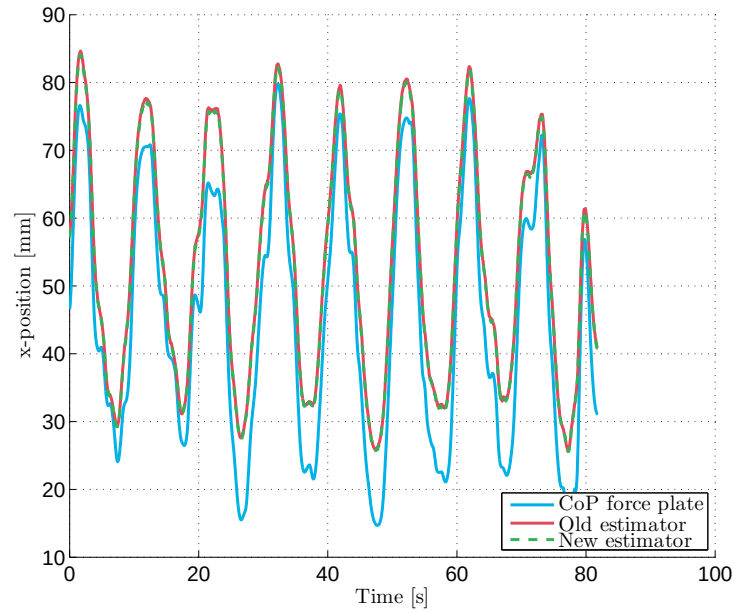


Figure 7-4: Comparison of the old and new CoM position estimator with the measurements from the force plate, for the x direction. The robot was standing solely on the right foot. As with the experiment in double support, a large constant offset is present.

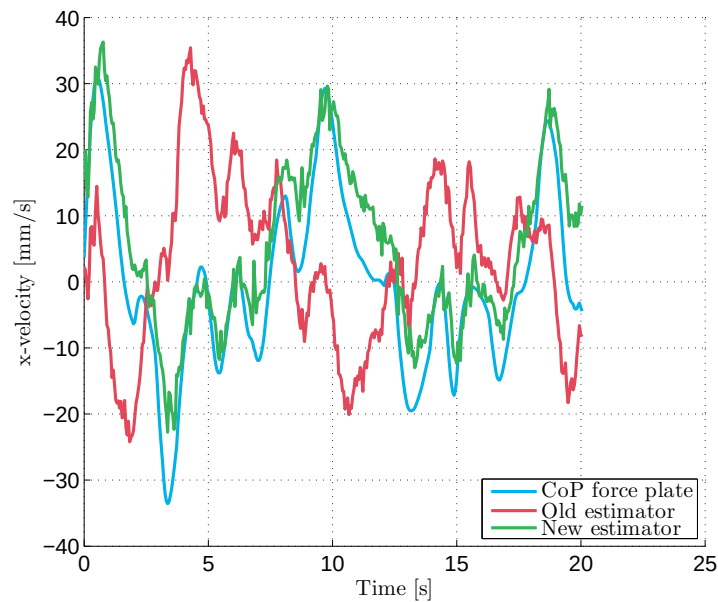


Figure 7-5: The CoM velocity in x direction of the CoM velocity estimators and the force plate measurement, while the robot was standing on its right foot. For clarity only the first 20 seconds of the experiment is shown. Again, the new estimator performs significantly better than the old one.

Estimated signal	Initially predicted ICP error	
	Old estimator	New estimator
CoM position x	152 ± 51 [mm]	146 ± 52 [mm]
CoM position y	-210 ± 24 [mm]	-202 ± 29 [mm]
CoM velocity x	14 ± 88 [mm/s]	27 ± 21 [mm/s]
CoM velocity y	6 ± 65 [mm/s]	-37 ± 36 [mm/s]

Table 7-3: The average errors and standard deviations in the initially predicted ICP due to an error in the CoM, for the experiment in single support on the right foot.

7-3 Discussion of the Results

The error in the CoM position that was already observed in Section 3-2-2, has not been reduced significantly by the new estimator. With the motion capture experiments Section 3-1-3, it was also observed that a large unknown bias is present with the perception of the feet. It is likely that both errors are caused by the same source. Although tuning of the Kalman filter may slightly reduce the error for the CoM position, it is expected that this error is largely due to an error in the kinematic model.

Moreover, the force plate experiment allows an accurate measurement of total the mass of the robot. It was found that it currently weighs 18.17 kg, while according the kinematic model it weighs only 16.59 kg. This difference of 1.58 kg, which is 9% of the total body weight, is thus omitted in the calculation of the CoM position.

For the CoM velocity estimator a large improvement is observed, which results in a more steady prediction of the step location. However, it was only possible to do experiments at low velocity. If the robot would perform more realistic tasks, such as stepping and walking, the CoM velocity is generally higher. In this case it is expected that the difference between the old and new estimator would be even larger. But unfortunately, it is not possible to verify this with the force plate experiment.

The oscillating reference signal is the same for both the single and double support experiments. However a clear difference is visible between the CoP velocity trajectory in double support (Figure 7-2) and in single support (Figure 7-5). In single support the CoP trajectory is much less steady, indicating that the robot has more difficulty controlling the upper body movements. These non-smooth movements cause higher inertial forces, which may cause that CoP measurement deviates from the actual CoM. Therefore the results of the single support experiment are less reliable.

7-4 Tuning the Kalman Filter

As important aspect of the Kalman filter is the proper selection of the $\mathbf{Q}$ and $\mathbf{R}$ matrices. These should be picked such that the variance of each signal matches the true signal as best as possible. At the time of the implementation of the Kalman filter, the true CoM position and velocity were unknown, therefore the covariance matrices could not be determined. However, using the measurements from the force plate an estimation can be made of the variances. The

following calculations are based on the experiment in double support. Here the robot moved more smooth than with the single support experiment, therefore the CoP measurement gave a better estimate of the CoM.

The variance indicates the spread of the measurement error, and is calculated with:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \quad (7-1)$$

Here $\mathbf{x} = [x_1, x_2, \dots, x_i]$ represents the error between the measurement signal and the value observed by the force plate. The variances of the position and velocity measurements are calculated using Matlab, and are listed in Table 7-4.

Finding the variance of the acceleration measurement is not straightforward. In the Kalman filter, each time step the measurement signal is integrated and added to the a posteriori velocity estimate (6-7). This results in the CoM velocity measurement from the IMU. The variance of this signal is therefore biased due to the addition of the a posteriori velocity estimate. Therefore it is decided to use the catalogued standard deviation of the acceleration measurement, and then manipulated so that it corresponds to the variance of the velocity estimation.

The standard deviation of the Xsens inertial measurement unit is $\sigma_{acc,imu} = 0.02 \text{ m/s}^2$ [14]. To calculate the variance on the velocity estimate, first the height difference of the CoM and IMU is incorporated, similar to (6-3). For the measurement signal, the acceleration is then numerically integrated by multiplying it with the time step (h). Similarly, this is done for the standard deviation. And finally, the standard deviation is squared to find the variance:

$$\sigma_{vel,com}^2 = \left(\frac{z_{com}}{z_{imu}} \sigma_{acc,imu} \cdot h \right)^2 \quad (7-2)$$

When the robot is standing, $z_{com} = 0.64 \text{ m}$ and $z_{imu} = 0.86 \text{ m}$. Furthermore, the control loop runs at 1 KHz, so $h = 0.001 \text{ s}$. When substituting, it is found that the variance is $2.1935 \cdot 10^{-10} \text{ m}^2$.

Measurement signal	Variance $\sigma^2 \text{ [m}^2\text{]}$
$x_{com,mass}$	$1.2135 \cdot 10^{-6}$
$\dot{x}_{com,mass}$	$7.7862 \cdot 10^{-6}$
$\dot{x}_{com,imu}$	$2.1935 \cdot 10^{-10}$

Table 7-4: Noise variances that are found with the experiment in the double support.

The measurement noise covariances matrix is thus:

$$\mathbf{R} = \begin{bmatrix} 1.2135 & 0 & 0 \\ 0 & 7.7862 & 0 \\ 0 & 0 & 2.1935 \cdot 10^{-4} \end{bmatrix} \cdot 10^{-6} \text{ [m}^2\text{]} \quad (7-3)$$

The process noise covariance is more difficult to determine, as it does not just depend on the variance of the torque signals, but also on the accuracy of the LIPM model. Additionally, the

variance of the torque signals need to be translated to the resulting variance of the states. Here the input matrix ($\mathbf{B}$) may introduce additional noise. For these reasons the process noise covariance matrix could not be determined, however, it is possible to calculate the variance in the torque measurement.

The torque that is observed by the force plate is found with:

$$\tau_a = x_{cop} \cdot F_z. \quad (7-4)$$

Here F_z is the vertical ground reaction force observed by the force plate. When comparing this to the torque that is applied by the robot, it is found that the variance of the torque signal is $0.23 \text{ [(Nm)}^2\text{]}$.

7-5 Conclusion

In this experiment TULip is placed on a force plate to compare the old and new CoM estimator, and find the error to the real CoM. It is found that the new CoM position estimator does not perform significantly better than the old estimator. Although tuning of the Kalman filter may partially improve the estimate, it is expected that this mostly is caused by errors in the kinematic model of TULip.

For the new CoM velocity estimation however, a large improvement was observed. It now follows the CoP velocity as measured by the force plate quite accurately, especially during the double support experiment. Here it is found that the old estimator has an average error of 2 mm/s with a standard deviation of 21 mm/s. For the new estimator this was improved to an average error of 4 mm/s with a standard deviation of just 3 mm/s, which is a factor 7 smaller. It can therefore be concluded that the CoM velocity estimation is now sufficiently accurate for a proper initial prediction of the step location.

Finally, the force plate measurement data was used to find the variances of the measurement signals, so that the measurement noise covariance matrix ($\mathbf{R}$) could be determined.

Discussion and Future Work

It was found that the center of mass (CoM) was incorrectly calculated for the humanoid robot TULip. For improving this, an estimator based on the Kalman filter and the Linear Inverted Pendulum Model was implemented. An accurate measurement of the CoM is important for the calculation of the Instantaneous Capture Point. Implementing a new CoM estimator therefore led to a significant improvement for the prediction of the step location.

In this chapter the results and methods that are used throughout this thesis are discussed. Additionally, possible future work is presented that can further improve the state estimation on TULip.

8-1 Using a Software Based Solution

During the measurements with the motion capture system, it was shown that a large error is present that is caused by backlash in the joints. The approach for thesis was to find a solution by applying a filter algorithm, while the inaccurate hardware persisted. Alternatively the robot could be improved by relocating or adding encoders to increase measurement accuracy, or redesign the hip joint so that the backlash would be reduced. This way the estimation of the CoM, along with various other issues, would be improved.

8-2 The Motion Capture Experiment

An experiment with a motion capture system was conducted to investigate the perception of the feet of the robot. As explained in Section 3-1-2, quite a lot of operations were required to be able to compare the data from the robot (VTD) to that of the motion capture (MC) cameras. The estimated accuracy of the system is below a millimeter, but some operations were sensitive to human errors. This specifically holds for the first step, where several additional points, such as the location of the ankle joints and the IMU, were defined in the MC data. Here the distance between the markers and physical points on the robot were measured

using a caliper and the 3D SolidWorks model of the robot. It is likely that this introduced a measurement error of several millimeters.

Ideally the measurement data of the motion capture system could also be used to find the error in the CoM position and velocity. However, the CoM can not be directly measured by the motion capture system, this has to be extracted from the individual marker locations that are placed on the limbs. As with the creation of the ankle joints and IMU location, here human measurement errors would be introduced. Moreover, the real mass and CoM of each limb are unknown, therefore the same the kinematic model that is currently implemented in TULip would have to be used. The CoM measurement by the motion capture system would then contain both measurement errors and modelling errors. It was therefore expected that too much accuracy is lost to draw a conclusion on the accuracy of the CoM calculation on the robot. It was thus decided to only focus on the error in the foot location, and use this to get an indication in the error in the state estimation of the robot.

8-3 The Force Plate Experiment

Using a force plate the accuracy of the current CoM calculation was measured (Section 3-2-2). It was also used to test the performance of the new estimator (Chapter 7). During the experiments, the foot location of the robot with respect to the force plate was measured with a slide ruler. As with the motion capture experiment, it is likely that this introduced several millimeters of error on the CoP location with respect to the ankle joints.

The measurements were done only at a low velocity to keep the difference between floor projection the CoM and the CoP at a minimum. For the experiments in single support already an unsteady trajectory of the CoP was observed. This behavior is also expected in double support when the oscillation frequency would be increased. This illustrates that it is difficult to accurately measure the CoM with higher velocities. However, it is expected that for an increasing velocity, the performance gain by the new CoM estimator would be larger.

It would have been insightful to do a CoM position and velocity measurement for the course of a step, to investigate the effect on the ICP throughout the entire step. However, the step trajectory and foot impact results disturbed upper body motions of the robot, which leads to a non-smooth CoM velocity and high inertial forces. Since the force plate only measures the CoP, this could therefore no longer be accurately compared to the CoM velocity or ICP of the robot.

8-4 Implementation of the Kalman Filter

In the current implementation of the Kalman filter (Chapter 6), the **A** and **B** matrices that describe the dynamics of the Linear Inverted Pendulum Model (LIPM) are not constant. Instead, these change depending on the height of the CoM, which means that the model description is not linear. However in practice the variation is small, since the robot is generally standing upright. Moreover, varying these parameters ensures that the LIPM model describes the dynamics as accurately as possible when the robot is moving.

The CoM acceleration that is measured using the IMU, is integrated before it is used in the Kalman filter (Section 6-3-2). It is believed that this is the best solution, but two alternative methods were considered, which are now briefly explained.

As a first alternative, the acceleration could be added as a state to the LIPM, using

$$\begin{bmatrix} x_{com} \\ \dot{x}_{com} \\ \ddot{x}_{com} \end{bmatrix}_k = \begin{bmatrix} 1 & h & 0 \\ \omega_0^2 h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{com} \\ \dot{x}_{com} \\ \ddot{x}_{com} \end{bmatrix}_{k-1}, \quad (8-1)$$

as the **A** matrix in the forward model of the Kalman filter. The measurements are then simply mapped to the states with a 3×3 identity matrix as the sensor model. This method was also tested in simulation. However, the acceleration state is then completely disconnected from the position and velocity states. When implementing this on the robot, the estimator would then filter the acceleration estimate, but it would not be used for improving the estimation of the CoM velocity.

Alternatively, it was considered to add the acceleration measurement as an input to the state equations, by using the **B** matrix:

$$\begin{bmatrix} x_{com} \\ \dot{x}_{com} \end{bmatrix}_k = \begin{bmatrix} 1 & h \\ \omega_0^2 h & 1 \end{bmatrix} \begin{bmatrix} x_{com} \\ \dot{x}_{com} \end{bmatrix}_{k-1} + \begin{bmatrix} 0 \\ h \end{bmatrix} \ddot{x}_{com,imu}. \quad (8-2)$$

However, this interferes with the ankle torques that currently serve as the input, which would therefore have to be omitted with this method. Therefore it was decided to integrate the CoM acceleration separately, and then add it to the Kalman filter as an additional velocity measurement.

8-4-1 Tuning the Kalman Filter

For the Kalman filter, the **Q** and **R** matrices have to be set to represent the real process and noise covariance. In simulation this is quite straightforward since **Q** and **R** are chosen up front, and then used to generate the corresponding noise signals. However, on the robot noise is already present and it is difficult to determine the covariances, especially for the process noise.

After the force plate experiment the variances of the measurement signals were determined, to calculate the measurement noise covariance matrix (Section 7-4). However, the process noise covariances could not be determined. The variance of the torque signal was found, but **Q** additionally depends on the accuracy of the LIPM, the total mass and the CoM height. It may be possible to approximate **Q** by manual tuning. Alternatively, offline optimization methods such as the Autocovariance Least-Squares technique [22, 23] can be applied to find the covariance matrices for the Kalman filter.

8-5 Extensions to the Kalman Filter

A basic discrete Kalman filter was implemented on the robot. It is expected that this will suffice for this situation, however various extensions are available to further improve the performance, which are summarized below.

The process noise covariance $\mathbf{Q}$ and measurement noise covariance $\mathbf{R}$ are currently constant. These can also be made adaptive [17], for instance depending on the support setting or task of the robot, to better resemble the variance of the measurement signals.

Currently, the dynamics of the robot are linearized with the LIPM. However, by using the Extended Kalman filter [17] a non-linear dynamic model can be used in the state estimator. This way a more complex model can be used to describe the dynamics of the robot, which may lead to a more accurate state estimation.

8-6 Future Work

The CoM estimator that was implemented on TULip significantly improved the accuracy. However, several future research topics are proposed to further improve the state estimation of the robot.

8-6-1 Investigate Modelling Errors

With the motion capture experiments and the force plate experiments, a large bias was observed in the perception of the feet and the measurement of the CoM. Since no mechanical cause could be found, it is expected that the kinematic model that is currently used on TULip contains one or more errors. The kinematic model is a fundamental part of the software, which has a large effect on the motion controllers. Therefore it is vital that it is bug-free. Tracking down the error in the model is difficult, but the motion capture data that is collected during the experiments may help isolate the exact source.

8-6-2 Improve the Mechanics

It is believed that the current problems with TULip, such as the instability when stepping, are mostly caused by the motion controllers. However, improving the hardware will make accurately sensing and controlling the robot easier.

Currently, the hip joint is most problematic. With the motion capture experiments it was shown that it contains backlash that causes a large error in the perception of the feet. Mounting an additional sensor to measure the angle of the leg directly on the joint would give a considerable improvement. However, backlash would then still be present. So ideally, the hip joint is redesigned and equipped with gearboxes that do not contain backlash, such as harmonic drives or ball screw spindles.

In Section 2-1-2 it was briefly mentioned that the robot is equipped with pressure sensors underneath the feet. These are now only used to detect ground contact. It was intended to also use these for determining the CoP of the robot, but this measurement was too inaccurate. However, a direct measurement of the CoP would be valuable information for determining the CoM position and velocity. It could then be used as an additional signal in the Kalman filter, to improve the prediction. Furthermore, the direct measurement of the CoP is not biased by modeling errors or mechanical deficiencies in the kinematic chain. Therefore, for further improving the CoM estimation, it is recommended to improve or redesign the foot pressure sensors.

Chapter 9

Conclusions

The goal of this thesis is *Improving the estimation of the center of mass position and velocity by applying sensor fusion on the humanoid robot TUlip*. Therefore, two experiments were done to identify the exact problem. Subsequently, the current Center of Mass (CoM) calculator was analysed and improvements were proposed. Next, a new CoM estimator based on the Kalman filter was designed and implemented on the robot TUlip. And finally, the performance of the new CoM estimator was experimentally tested. Below the main conclusions are presented.

Two separate experiments were done to identify the error in the perception of the feet and the error in the current CoM estimation. With first experiment, a motion capture system was used to show that an error of up to 80 mm is present in the location of the feet. For a large part, this is caused by backlash in the hip joint, which leads to an error at the feet of 26 mm and 32 for the x and y direction. For the second experiment, a force plate was used to measure the error in the CoM calculation. It was found that the CoM position contains an average error of 23 mm with a standard deviation of 1.1 mm, and the CoM velocity has an average error of 2 mm with a standard deviation of 21 mm/s. It was shown these errors result in an inaccurate initial prediction of the step location, which is off by 436 mm with a standard deviation of 21 mm due to the CoM position error, and 10 mm with a standard deviation of 102 mm due to the CoM velocity error.

Subsequently, a new CoM estimator was designed. This was done in several steps. First the current CoM calculator was analysed and flaws were found, specifically for the calculation of the CoM velocity. These calculations were then improved, since these serve as an input to the new estimator.

This estimator is based on the Kalman filter, which is an algorithm to estimate the system states by combining multiple noisy measurement signals. In the implementation, the Linear Inverted Pendulum Model with the ankle torques as input is used to make a forward prediction. It takes the calculated CoM position and velocity, and the CoM velocity measured by the inertial measurement unit, to make an improved estimate of the CoM position and velocity.

With an experiment on a force plate the performance of the new estimator was tested and compared to the old calculator. For the estimation of the CoM position no significant im-

provement was found, which is due to errors in the kinematic model of TULip. However, the CoM velocity estimation is now much more precise. In double support the average error is now 4 mm/s with a standard deviation of 3 mm/s. Compared to the old estimator, the standard deviation thus is a factor 7 smaller, which is sufficient for an accurate prediction of the step location.

Bibliography

- [1] J. Pratt, B. Krupp, and C. Morse, "Series elastic actuators for high fidelity force control," *Industrial Robot: An International Journal*, vol. 29, no. 3, pp. 234–241, 2002.
- [2] T. De Boer, *Foot placement in robotic bipedal locomotion*. PhD thesis, Delft University of Technology, 2012.
- [3] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 200–207, IEEE, 2006.
- [4] T. Koolen, "Capturability-based analysis and control of legged locomotion , part 1 : Theory and application to three simple gait models," *Office*, 2011.
- [5] P. M. Bays and D. M. Wolpert, "Computational principles of sensorimotor control that minimize uncertainty and variability.," *The Journal of physiology*, vol. 578, pp. 387–96, Jan. 2007.
- [6] J. Pratt, "Capturability-based analysis and control of legged locomotion , part 2 : Application to m2v2 , a lower body humanoid," *Control*, 2011.
- [7] B. Stephens, *Push Recovery Control for Force-Controlled Humanoid Robots*. PhD thesis, Carnegie Mellon University, 2011.
- [8] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [9] D. Hobbelen, T. de Boer, and M. Wisse, "System overview of bipedal robots flame and tulip: Tailor-made for limit cycle walking," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2486–2491, IEEE, Sept. 2008.
- [10] "Robocup." <http://www.robocup.org/>. Accessed: 08/08/2012.
- [11] D. Hobbelen and M. Wisse, "Limit cycle walking," *Humanoid Robots, Human-like Machines, book*, 2007.

- [12] J. J. Craig, *Introduction to robotics: mechanics and control*. Prentice Hall, 2004.
- [13] T. Ishida and A. Takanishi, “A robot actuator development with high backdrivability,” in *Robotics, Automation and Mechatronics, 2006 IEEE Conference on*, pp. 1–6, IEEE, 2006.
- [14] Xsens Technologies B.V., *MTi and MTx User Manual and Technical Documentation*, 2010.
- [15] Tekscan, “Flexiforce load/force sensors and systems.” <http://www.tekscan.com/flexiforce.html>. Accessed: 14/08/2012.
- [16] “Surveyor stereo vision system.” <http://www.surveyor.com/stereo/>. Accessed: 14/08/2012.
- [17] G. Welch and G. Bishop, “An introduction to the kalman filter,” *Design*, vol. 7, no. 1, pp. 1–16, 2001.
- [18] Qualisys, “Oqus camera series.” <http://www.qualisys.com/products/hardware/oqus/>. Accessed: 08/08/2012.
- [19] AMTI, “Force plate OR6-7-1000.” www.amti.biz/OR6-7.aspx. Accessed: 08/08/2012.
- [20] S. Kajita and K. Tani, “Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode,” in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pp. 1405–1411, IEEE Comput. Soc. Press, 1999.
- [21] 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 *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1, pp. 239–246, IEEE, 2001.
- [22] M. Rajamani, *Data-based techniques to improve state estimation in model predictive control*. ProQuest, 2007.
- [23] B. Åkesson, J. Jørgensen, N. Poulsen, and S. Jørgensen, “A tool for kalman filter tuning,” *Computer Aided Chemical Engineering*, vol. 24, pp. 859–864, 2007.