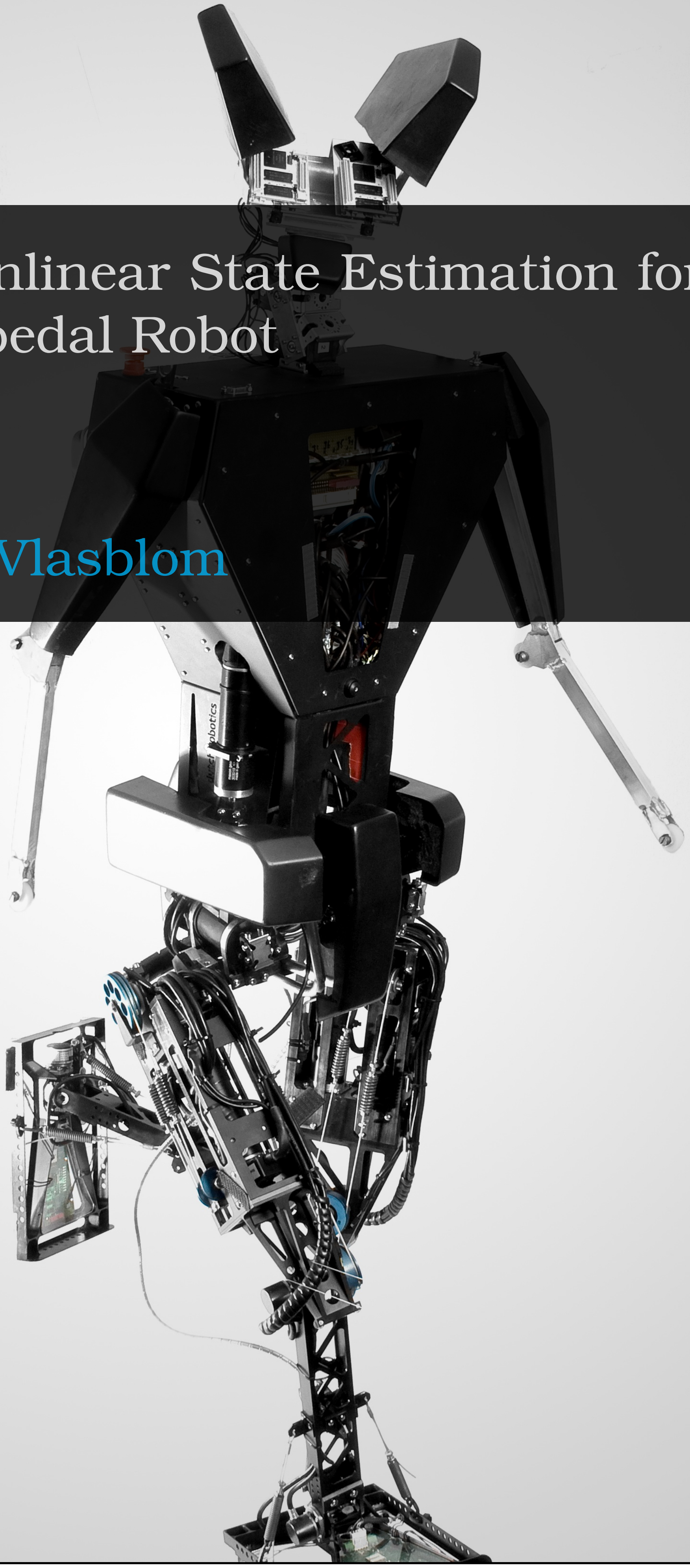
Nonlinear State Estimation for a Bipedal Robot

E. Vlasblom

Master of Science Thesis



Delft University of Technology

Delft Center for Systems and Control

Nonlinear State Estimation for a Bipedal Robot

Master of Science Thesis

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

E. Vlasblom

June 14, 2014

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

Delft University of Technology
Department of
Mechanical Engineering (me)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Nonlinear State Estimation for a Bipedal Robot

by

E. Vlasblom

in partial fulfillment of the requirements for the degree of

Master of Science Mechanical Engineering

Dated: June 14, 2014

Supervisor(s): ______________________

Dr.ir. H. Vallery

Reader(s): ______________________

Prof.dr. F.C.T. van der Helm

____

Dr.ir. M. Wisse

____

Dr. G.A. Delgado Lopes

Nonlinear State Estimation for a Bipedal Robot

Erik Vlasblom

***Abstract*—This report discusses the design, implementation and evaluation of a state estimator for bipedal robots, making use of the robot's full-body dynamics. The Unscented Kalman Filter is chosen to fuse the predictions with the measurements. To evaluate the new filter design, a second estimator is implemented as a benchmark, using a conventional Linear Inverted Pendulum Model for the prediction and a Kalman Filter for data fusion. Simulations of three different motions show how the CoM estimates of both methods deal with different kinds of uncertainties and an experiment was done as a preliminary application on the TUlip bipedal robot. Estimation with a pendulum model gave incorrect estimates in the presence of sensory bias. Using the full-body dynamics, however, the effect of sensory bias was reduced significantly. Moreover, the proposed method was shown to be robust against parametric errors. However, the performance varied between different motions, making it hard to tune. Improving the filter to inherently work for various movements as well as making the filter numerically efficient enough for online implementation still requires further research.**

***Index Terms*—Bipedal robot, state estimation, unscented kalman filter.**

I. Introduction

IN a world that has been designed for humans, the ideal robot would be one that is morphologically similar to a human. These type of robots would be able to navigate through our world and handle objects much like we do. Because of this, they could perform a large variety of tasks, ranging from helping the elderly and entertainment to carrying out repairs in hazardous environments such as nuclear power plants. Therefore, humanoid robots are an important topic of research. However, controlling the balance of two-legged robots still proves to be a challenge, especially during more complex motions such as walking.

Different walking strategies have been used during several decades of research on legged robotic platforms. The most popular strategy is locomotion based on the zero-moment point (ZMP). This is is the point where the influence of all forces acting on the mechanism can be replaced by one single force [1]. The key is to keep this point within the support polygon and away from the edges, which is usually done by generating and following a desired centre of mass (CoM) trajectory. Many robots successfully apply this technique for various types of movements, including walking. However, it is not very energy efficient. Humanoid robots using limit cycle walking mimic human gait by relying on the passive dynamics of the system. Hence, these robots have similar energy efficiency as humans [2], but they can not do diverse movements because they are limited to walking patterns that result in a limit cycle. Moreover, withstanding large unexpected pushes is not an inherent part of either of these techniques. This means they are not as robust as humans. To produce human-like gait in terms of energy efficiency, versatility and robustness, capturability-based control was developed [3] [4], based on instantaneous capture point control. The instantaneous capture point (ICP) was defined to be the point where one instantaneously has to step to come to a full halt. To maintain balance during motion, ankle and hip balance strategies can be used when the ICP lies within the support polygon and a step can be taken otherwise. By continuously forcing the ICP out of the support polygon and taking a step to prevent from falling, a stable gait pattern can be achieved. Independent research within the field of human movement science shows that the ICP, called the extrapolated center of mass (XcoM) by the researchers, indeed is a good measure of balance for humans while standing and walking [5] [6].

Crucial to walking is accurate knowledge about the state of the system. Still, state estimation is often named as a limiting factor in the control of humanoids [4] [7] [8] [9]. For balancing, important derived states like the support polygon and the CoM have to be obtained using the kinematic model of the robot. Though, uncertainties in the kinematic chain such as play, link flexibility and parameter uncertainty as well as sensory noise make it difficult to accurately obtain these states [10]. Because these uncertainties are hindering estimation, the control is limited and therefore, state estimation has to be improved. Seeing that control of humanoids becomes more advanced and human-like, it is essential that the same is done for state estimation.

Based on physiological studies we know that there is strong evidence for the presence of accurate internal models within the central nervous system (CNS) of humans. Forward dynamic models within the CNS are expected to help detect environmental changes, increase the accuracy of state estimates and make predictions for mental planning [11] [12]. Moreover, it was shown that humans under certain circumstances make use of forward models in a Bayesian framework to estimate their state and that of their environment [13] [14]. For this reason, using a complete description of the robot dynamics in combination with Bayesian estimation might be the best way of obtaining state estimates. And not only could this combination improve state estimation, but it could also function for other parts of sensorimotor learning like parameter estimation and detection of environmental changes.

Various studies have been done on state estimation in bipedal robots using internal models and Bayesian estimation. Most often, these studies focus on estimating the CoM. A common approach is using a Kalman Filter and a linearised pendulum as a model to predict the CoM motion [15] [16] [10]. Some researchers have modified this method to account

The author is with the department of Mechanical Engineering of the Delft University of Technology, the Netherlands (e-mail: erik.vlasblom@gmail.com).

for modelling errors by explicitly estimating them. In one study, an Extended Kalman Filter and a flexible inverted pendulum model were used to estimate the unmeasurable flexible motion of the biped [17]. Another research augmented the state of the linearised pendulum with a CoM offset and a disturbance force that could then be estimated [18]. These studies both showed improvements as compared to using a linearised pendulum only, but it was also noted that this is a lumped-mass generalisation of the real errors. The true error could be the sum of a variety of smaller errors in each body. Furthermore, it was suggested to improve the model by adding neglected effects like angular momentum. And indeed, by using a linearised pendulum, the complex dynamics of the robot are simplified. For humanoids using the ZMP strategy where a CoM trajectory is prescribed or for robots taking small, pendulum-like steps, assuming a linearised pendulum can work. For more complex movements and larger human-like steps the prediction might be insufficient. Some researchers have therefore used more sophisticated models like a constrained rigid multi-body model or a planar five-link model [19] [20]. The planar five-link model was compared against the pendulum model, and although it is less robust against parameter errors, it showed an improved estimation of the CoM. In addition, the estimation of the feet positions was improved . A planar five-link model was also successfully used with a Sliding Mode Observer to estimate the trunk orientation of a bipedal robot [21].

Because there is strong evidence that humans have accurate internal models and because more complete models have shown to improve state estimation, the use of a three-dimensional full-body model is suggested in this report. Using this model, the actual states of the robot are estimated. They can be used to improve not only the CoM, but also other derived states like the feet positions. And lastly, instead of using a lumped-mass generalisation, specific sources of uncertainty could be more precisely targeted by tuning the noise levels for each state of the robot. The prediction of the full-body model will be fused with the measurements by means of a nonlinear state estimator.

This report is organised as follows. In section II, various techniques for estimating nonlinear systems will be discussed. In section III the bipedal robot TUlip will be described after which in section IV the internal models will be explained. Section V shows the filter implementation. Section VI shows the simulation, application and results. Section VII gives a discussion of the results and section VIII concludes this report.

II. Nonlinear State Estimation

To be able to use more advanced models, nonlinear state estimation techniques have to be used. Roughly, they can be divided into deterministic and probabilistic types. The common Luenberger Observer as well as the Sliding Mode Observer fall in the first category. Although the Sliding Mode Observer has been used in robotics and was shown to be robust, only probabilistic filters will be considered here because of the similarities they have shown with human behaviour. These filters are based on Bayesian inference: given a prior belief of the state and some measurements, what is the actual state of the system. The belief is obtained via a prediction based on the previous estimate. As both the belief and the measurements are uncertain, it becomes a matter of predicting and finding a probability density function (pdf) of the state rather than a single value. The final estimate is then obtained by applying some optimisation criterion to the posterior pdf such as minimising the mean square error between the estimate and the true state. When the prediction model is linear and the noise is Gaussian, the Kalman Filter gives an exact optimal solution. This is because a Gaussian pdf is preserved under a linear transformation. For nonlinear systems, it is necessary to use computationally tractable sub-optimal solutions [22] [23]. Therefore, choosing the right technique comes down to making a trade-off between how accurate the posterior pdf has be be described and how much time is available for computation. The more accurate the predicted pdf can be approximated, the closer the outcome is to the optimal estimate, but the more calculation effort it requires.

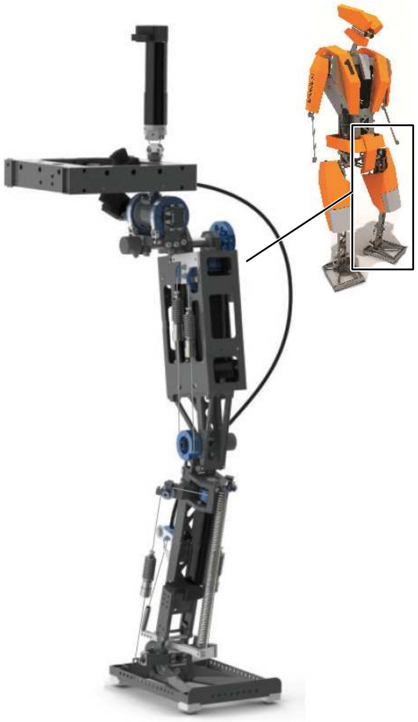


Fig. 1. A CAD model of the TUlip humanoid robot, showing the construction of the six degree of freedom leg.

Starting at the accurate but computationally expensive side of the spectrum, there is a wide range of particle filters making use of Sequential Monte Carlo Sampling to approximate the posterior pdf. The more samples, the more accurate the estimation will be [24] [25]. The Ensemble Particle Filter has a similar approach. Using an ensemble of particles and a slightly different update procedure from Particle Filters, this type can also capture higher order moments of the pdf [26]. The amount of particles required for these methods can range from 50 up to a several thousand, but is hard to predict up front. Because the computation effort scales with the amount of particles, computational complexity can become enormous [27] [22]. Less computationally expensive are the Sigma Point Kalman Filters. Examples are the Quadrature Kalman Filter, the Central Difference Kalman Filter [28] [29] and the more popular Unscented Kalman Filter [30] [31] [32]. They make use of deterministically chosen points with associated weights to approximate the posterior as a Gaussian pdf. Sigma Point Kalman filtes can not accurately describe a multi-modal pdf or higher moments, but the amount of samples is fixed and scales with the state size N. The Unscented Kalman Filter, for example, usually requires $2N + 1$ samples. This makes them computationally favourable over Particle Filters. On the other end of the spectrum, the less accurate Extended Kalman Filter can be found. It is an ad hoc solution, making use of local linearisation of the nonlinear function to be able to use the standard Kalman Filter equations [33]. Though it is often used, it can give very bad estimates in case the assumption of linearity is violated [27] [31]. It also requires the calculation of Jacobians that do not exist on the switching surface of a hybrid system. The previously mentioned sample based techniques can more easily deal with discontinuities [31] [22].

For a humanoid robot, estimation has to be done real-time with limited computation power. Moreover, the technique has to be able to deal with the hybrid nature of a biped. Sigma Point Filters are shown to have better accuracy when compared to the Extended Kalman Filter. Also, constraints and discontinuities can be more easily incorporated. They are preferred over Monte Carlo techniques because they require far less computations and are thus more suited for real-time systems. Considering the above, a Sigma Point Kalman filter seems like a good solution for state estimation in a humanoid robot. Specifically, the Unscented Kalman Filter is chosen because it can be tuned in various ways to either increase accuracy or decrease computational complexity.

III. TUlip Hardware and Control

The TUlip is a 1.1 m tall, 18 kg humanoid robot. Both legs have 6 actuated degrees of freedom: 3 in the hip, 1 in the knee and 2 in the ankle [34] [7]. Figure 1 shows a CAD model of the TUlip and figure 3 shows a schematic of the robot kinematics. Each joint is actuated with a Maxon DC motor and a planetary gearhead. Flexion and extension of the hip, knee and ankle as well as the inversion and eversion of the ankle are achieved with series elastic actuation (SEA) [35]. Here, the motors are connected to the joint via an elastic element. By measuring the elongation of this element, the joint torque is calculated which in a feedback loop is used for torque control. Each motor axle has a HEDS 5540 rotary encoder attached. The joints with series elastic actuation also have a Scancon 2RMHF encoder on the joint axle. The torso is equipped with an Xsens MTI inertial measurement unit that outputs estimates of the orientation, rotational velocity and linear acceleration in three dimensions. Lastly, each foot has Tekscan Felxiforce sensors to detect contact with the floor. The current software runs in a 1 kHz loop in an xPC Target real-time software environment. On-board calculations are done on a 1 GHz Diamond System Poseidon SBC with 256 MB RAM.

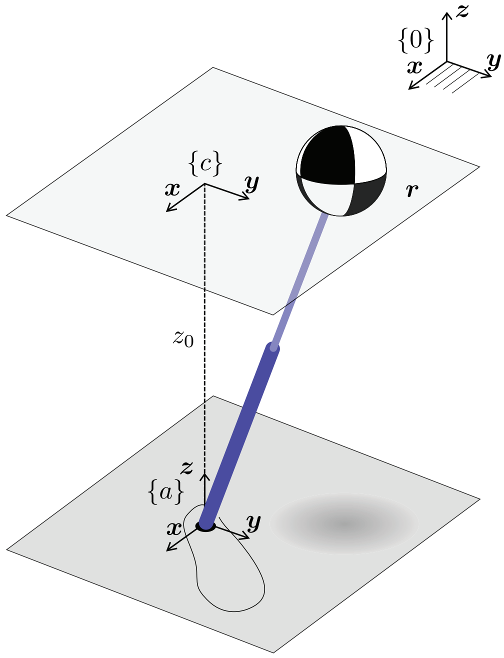


Fig. 2. The Linear Inverted Pendulum Model, which is often used to analyse the Centre of Mass motion.

IV. System Models

The internal robot models used for state estimation are usually simple models that neglect nonlinear behaviour, angular momentum and the motion in the third dimension. To be able to estimate more versatile movements of next-generation humanoids and to better cope with uncertainties present in the kinematic chain, a three-dimensional multi-body model will be introduced. In the first subsection, the Linear Inverted Pendulum Model is described. This is the linearised pendulum that is often used for CoM estimation. In order to evaluate the estimator proposed in this report, an estimator based on this linearised pendulum will also be implemented as a benchmark. In the sections thereafter, the kinematics and dynamics of the multi-body robot model will be given.

A. Linear Inverted Pendulum Model

The Linear Inverted Pendulum Model (LIPM) is an often used model to approximate the CoM motion of a humanoid robot, both for control and estimation. A schematic is shown in figure 2. The pendulum is constrained to move along a horizontal plane at height z_0. This is achieved by means of a telescopic, massless leg whose internal forces keep the CoM on the horizontal plane. The base of the pendulum is located at the Centre of Pressure (CoP) of the robot. For a point foot, this is the ankle location projected on the ground. Vector $\boldsymbol{r}$ is the position of the CoM in frame a. The location of a is expressed as $\boldsymbol{r}_{CoP}$ in some global frame 0. The equations of motion are then given by

$$\ddot{\boldsymbol{r}} = \omega_0^2 \mathbf{P} \boldsymbol{r}, \qquad \text{with } \mathbf{P} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{1}$$

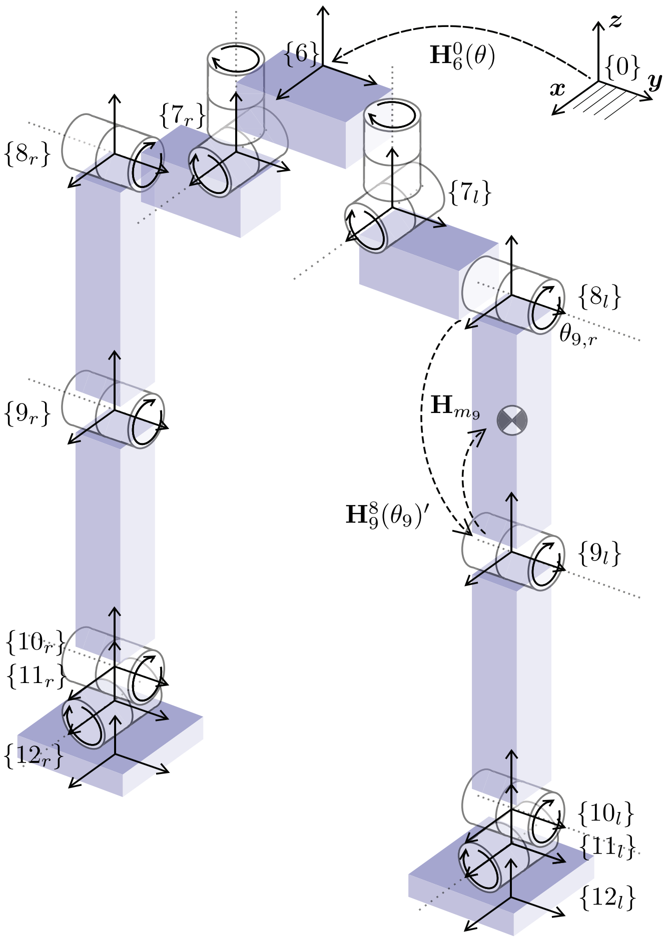


Fig. 3. The complete kinematic model of the TUlip humanoid robot.

where $\omega_0 = \sqrt{\frac{g}{z_0}}$ and $\mathbf{P}$ is a matrix that projects the equations onto the xy-plane. The result of the constraint is a set of uncoupled equations describing the motion of the CoM in frame c. Moreover, the equations are linear. This makes the LIPM useful for basic analysis. Also, the linear behaviour allows for using it as an internal model for state estimation, without needing nonlinear filters. A simple Kalman Filter suffices.

B. Kinematics

Figure 3 shows the complete kinematics of the TUlip with all degrees of freedom as described in section III. For the internal model used in the state estimator, the feet are removed. This greatly reduces model complexity by eliminating 2 degrees of freedom per leg and avoiding complex ground contact models. The frames 10_r and 10_l are the new end effector frames. The effect that the ankles have is incorporated by transforming the measured ankle torques to a wrench acting on the lower legs. The trunk is modelled as a floating body by defining a virtual, 6 degree of freedom kinematic chain that connects the trunk with the world frame. The model used for estimation thus has two 4 degree of freedom legs attached to the floating trunk giving a total of 14 degrees of freedom. It will be referred to as the Floating Footless Humanoid (FFH) model from here on. In this model, the 2 kinematic chains that can be recognized run from 0 to 10_r and from 0 to 10_l. For each chain, the body frames are related as

$$\mathbf{H}_i^{i-1}(\theta_i)' = \mathbf{H}_{s_i}\mathbf{H}(\theta_i)\mathbf{H}_{l_i} \quad \text{for } i = 1 \ldots n \tag{2}$$

where n is the number of the end effector frame. Homogeneous transformation $\mathbf{H}(\theta_i)$ represents a pure rotation or translation, depending on the type of joint. The transformation $\mathbf{H}_{s_i}$ contains a constant offset that splits the kinematic chain. At a bifurcation, the value of the offset depends on the chosen direction. It is an identity matrix otherwise. Transformations $\mathbf{H}_{l_i}$ contain constant offsets between the joints. The individual bodies are then related via their CoM positions according to

$$\mathbf{H}_i^{i-1}(\theta_i) = \mathbf{H}_{m_{i-1}}{}^{-1}\mathbf{H}_i^{i-1}(\theta_i)'\mathbf{H}_{m_i} \tag{3}$$

where $\mathbf{H}_{m_i}$ contains a constant center of mass position viewed from the body frame, as depicted in figure 3. The position kinematics are completed by recursively defining each CoM location relative to the global frame for each side as

$$\mathbf{H}_i^0(\theta_1,\ldots,\theta_i) = \mathbf{H}_{i-1}^0(\theta_1,\ldots,\theta_{i-1})\mathbf{H}_i^{i-1}(\theta_i) \tag{4}$$

The velocity kinematics are obtained using a geometric approach based on screw theory. For details, see Appendix A. Here, the basic steps in the derivation will be given using a slightly simpler notation of twists for readability. Also, the arguments of $\mathbf{H}_i^j$ shall be omitted from here on. First, the twist for each joint can be found as shown below.

$$\mathbf{T}_i^{i-1} = \frac{\partial}{\partial\theta_i}\left(\mathbf{H}_i^{i-1}\right)\mathbf{H}_i^{i-1^{-1}} \tag{5}$$

which is of the form

$$\begin{pmatrix} \tilde{\boldsymbol{\omega}} & \boldsymbol{v} \\ \mathbf{0} & 1 \end{pmatrix}$$

where $\tilde{\boldsymbol{\omega}}$ is a screw-symmetric matrix representing a rotational velocity and where the vector $\boldsymbol{v}$ represents a translational velocity. The twist $\mathbf{T}_i^{i-1}$ is expressed in twist coordinates as

$$\boldsymbol{t}_i^{i-1} = \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{pmatrix} \tag{6}$$

Using the twists of each joint, the Jacobian for the instantaneous spatial velocity $\boldsymbol{v}_i^s = \mathbf{J}_i^s\dot{\boldsymbol{\theta}}$ of the ith body relative to the global frame 0 can be constructed. The spatial velocity is the velocity as seen from the origin of the spatial frame 0. The vector $\boldsymbol{\theta}$ in this case represents all the degrees of freedom of one of the kinematic chains. The construction of the Jacobian is done by using all previously constructed twists and transformations matrices according to

$$\mathbf{J}_i^s = (\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n) \tag{7}$$

with

$$\boldsymbol{t}_j = \begin{cases} j \le i & \mathrm{Ad}_{\mathbf{H}_{j-1}^0}\boldsymbol{t}_j^{j-1} \\ j > i & 0 \end{cases}$$

Thus, column $\boldsymbol{t}_j$ is the contribution of the jth joint to the spatial velocity of body i. Because it is the spatial velocity as seen from frame 0, the contribution from all joints $j > i$ up to the end effector n for body i is zero. In the equation above, $\mathrm{Ad}_{(\cdot)}$ is the Adjoint transformation which transforms twists from one frame to another. The Adjoint transformation is defined as

$$\mathrm{Ad}_{\mathbf{H}} = \begin{pmatrix} \mathbf{R} & \tilde{\mathbf{d}}\mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \tag{8}$$

with $\mathbf{R}$ and $\boldsymbol{d}$ the rotation matrix and translation vector of transformation matrix $\mathbf{H}$. Note that $\tilde{\mathbf{d}}$ is the screw-symmetric form of $\boldsymbol{d}$. The body Jacobian can now be calculated using the inverse of the Adjoint transformation matrix.

$$\mathbf{J}_i^b = \mathrm{Ad}_{\mathbf{H}_n^0}^{-1} \mathbf{J}_i^s \tag{9}$$

This gives velocities of body i relative to the global frame 0 and seen from the body frame i as $\boldsymbol{v}_n^b = \mathbf{J}_i^b \dot{\boldsymbol{\theta}}$. These velocities can now be used to calculate the kinetic energy of the Lagrange equation. For completeness, all the body velocities will be collected by defining a total Jacobian for each chain according to

$$\mathbf{J}^b = \begin{pmatrix} \mathbf{J}_1^b \\ \vdots \\ \mathbf{J}_n^b \end{pmatrix} \tag{10}$$

This results in a Jacobian for the right and left leg. Both of which start with a Jacobian for the 6 degree of freedom virtual chain and end with the Jacobian of the 4 degree of freedom leg. The final Jacobian for all bodies is thus

$$\mathbf{J}_{tot}^b = \begin{pmatrix} \mathbf{J}_v \\ \mathbf{J}_r \\ \mathbf{J}_l \end{pmatrix} \tag{11}$$

with

$$\mathbf{J}_v = \begin{pmatrix} \mathbf{J}_1^b \\ \vdots \\ \mathbf{J}_6^b \end{pmatrix}, \ \mathbf{J}_r = \begin{pmatrix} \mathbf{J}_{7_r}^b \\ \vdots \\ \mathbf{J}_{10_r}^b \end{pmatrix}, \ \mathbf{J}_l = \begin{pmatrix} \mathbf{J}_{7_l}^b \\ \vdots \\ \mathbf{J}_{10_l}^b \end{pmatrix}$$

which gives the velocities $\boldsymbol{v}_{tot}^b = \mathbf{J}_{tot}^b \dot{\boldsymbol{\theta}}_{tot}$ of all bodies combined. From here on, $\boldsymbol{\theta} = \boldsymbol{\theta}_{tot}$ shall be used to represent all the 14 degrees of freedom.

C. Dynamics

The equations of motion for the FFH can now for example be obtained by using the Lagrange equation as shown in Appendix A. The final equations of motion are

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{G}(\boldsymbol{\theta}) = \mathbf{J}^b(\boldsymbol{\theta})^T \boldsymbol{f}_{ext} + \boldsymbol{\tau}_j \tag{12}$$

where the generalized coordinates are given as $\boldsymbol{\theta} = (x\ y\ z\ \psi_z\ \psi_y\ \psi_x\ \theta_{7_r}\ \dots\ \theta_{10_l})^T$. The states x, y, z are the prismatic joints and ψ_z, ψ_y, ψ_x the zyx Euler angles of the virtual chain. This rotation order is chosen such that it matches with the IMU output. The vector $\boldsymbol{\tau}_j$ represents the joint control torques and the vector $\boldsymbol{f}_{ext}$ contains the ankle torques on the end of both lower legs. They are mapped to the joint space making use of the body Jacobian of both lower legs. $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{G}$ are the mass matrix, the Coriolis forces and gravitational effects respectively.

Since a humanoid is transitioning between double and single support phases during walking, its dynamics are hybrid in nature. This is implemented by switching constraints on the feet using Lagrange multipliers. Let $\boldsymbol{p}_f = (p_{fx}\ p_{fy}\ h)^T$ represent the location of a foot with h the height of this foot above the ground. Then, the following constraint functions as a ground contact model for that foot.

$$h \geq 0, \qquad \lambda \geq 0 \tag{13}$$

Which means that when $h < 0$, the constraint $h = 0$ is activated. This ensures that the foot is always on or above the ground. It introduces the Lagrange multiplier λ that functions as a ground reaction force. As these forces can not pull the robot back to the ground, the extra condition on λ is required. Next, a no slipping condition is implemented for the case the foot is in contact with the ground. This is implemented as

$$\begin{pmatrix} p_{fx} \\ p_{fy} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{14}$$

The above constraints are conditional on whether or not the constraint on h is active. When these conditions and constraints are implemented for both feet, the hybrid dynamics can be represented using the same dynamics given in equation (12). In a similar way as the constraints of (13), joint motion limits are implemented for both knees. Finally, let $\boldsymbol{b}_s(\boldsymbol{\theta})$ represent the current active constraints of the hybrid state s. The governing equations in s are then given by

$$\begin{pmatrix} \mathbf{M} & \mathbf{B}_s^T \\ \mathbf{B}_s & \mathbf{0}_s \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{\theta}} \\ \boldsymbol{\lambda}_s \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_{tot} \\ \frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{B}_s)\dot{\theta} \end{pmatrix} \tag{15}$$

where $\mathbf{B}_s = \frac{\partial \boldsymbol{b}_s}{\partial \boldsymbol{\theta}}$. The total forces are $\boldsymbol{f}_{tot} = \mathbf{J}^b(\boldsymbol{\theta})^T \boldsymbol{f}_{ext} + \boldsymbol{\tau}_j - \mathbf{C}\dot{\boldsymbol{\theta}} - \mathbf{G}$ and $\mathbf{0}_s$ is a square zero matrix where the size depends on the number of active constraints.

To decrease computational burden, the bodies attached to the hip rotation and abduction are assumed massless. The upper and lower legs are assumed to be point masses. Furthermore, no impact equations were implemented. And, lastly, to further reduce the computational complexity, Coriolis forces were assumed to be small due to the low velocities and not taken into account.

V. State Estimation

In this section, the two state estimators are explained. In the first subsection the state estimation making use of the LIPM is given. It is based on the currently implemented CoM estimator for the TUlip [10]. Secondly, the implementation of the full-body state estimator will be explained. The results of both estimators shall be compared for evaluation.

TABLE I
PROCESS, MEASUREMENT AND SIMULATED SENSOR NOISE COVARIANCE FOR THE LINEAR INVERTED PENDULUM KALMAN FILTER.

	$r_x[m]$	$r_y[m]$	$\dot{r}_x[m/s]$	$\dot{r}_y[m/s]$
Process	$0.86e^{-6}$	$0.86e^{-6}$	$0.86e^{-6}$	$0.8583e^{-6}$
Measurement	$0.95e^{-4}$	$0.95e^{-4}$	$0.95e^{-4}$	$0.95e^{-4}$

A. Linear Inverted Pendulum Kalman Filter

The Linear Inverted Pendulum Kalman Filter (LIPM KF) uses the LIPM, given in section in section IV-A, as a forward model. As these equations are linear, a Kalman Filter can be used for estimation. To do so, the LIPM equation from (1) is put in state space form and discretised. The resulting equations are of the form

$$\begin{pmatrix} r_x \\ r_y \\ \dot{r}_x \\ \dot{r}_y \end{pmatrix}_{k+1} = \begin{pmatrix} 1 & 0 & h & 0 \\ 0 & 1 & 0 & h \\ \omega_0^2 h & 0 & 1 & 0 \\ 0 & \omega_0^2 h & 0 & 1 \end{pmatrix} \begin{pmatrix} r_x \\ r_y \\ \dot{r}_x \\ \dot{r}_y \end{pmatrix}_k \tag{16}$$

with the state $\boldsymbol{r} = (r_x \; r_y \; \dot{r}_x \; \dot{r}_y)$ expressed in frame a of figure 2. At each time step, the pendulum is fitted onto the current configuration of the robot by updating the location of a and the height z_0. These LIPM properties are updated by inserting the encoder measurements at the current time instant into the kinematic transformations. Frame a is chosen to be at the ankle position of the stance leg. For the double support phase, the CoP should lie somewhere in between both feet. As the pressure sensors on the TUlip can not measure the CoP but only determine if there is ground contact, frame a is placed in between both ankle positions for the double support phase. The previous estimate of r is then expressed in this frame to make the prediction for the next time instant.

The measurement equation that is used is

$$\boldsymbol{z}_{k+1} = (r_x \; r_y \; \dot{r}_x \; \dot{r}_y)^T_{k+1} \tag{17}$$

where the CoM positions r_x and r_y are calculated using the local CoM positions and the encoder measurements. The derivative is used as the velocity measurement.

Lastly, the noise levels were obtained from the currently implemented observer. They were tuned to achieve as good as possible tracking of the CoM and they are given in table I. For the simulation study, the simulated CoM measurement was calculated by using simulated joint angle sensors and forward kinematics. The noise covariance of the simulated sensors can be found in table II.

B. Full-Body Unscented Kalman Filter

The Full-Body Unscented Kalman Filter (FB UKF) uses the FFH model from section IV-C as an internal model to predict the next state. The Unscented Kalman Filter (UKF) is used to fuse the prediction with the measurements. Equation (15) is first put into a constrained state space form.

TABLE II
PROCESS, MEASUREMENT AND SIMULATED SENSOR NOISE COVARIANCE FOR THE FULL-BODY UNSCENTED KALMAN FILTER.

	x, y, z	ψ_x, ψ_y, ψ_z	θ_7, θ_8	θ_9, θ_{10}
	$[m]$	$[rad]$	$[rad]$	$[rad]$
Process	1^{-4}	1^{-4}	$1e^{-5}$	$1e^{-5}$
Measurement	1^{-5}	1^{-5}	$1e^{-5}$	$1e^{-6}$
Simulated sensor	$4e^{-4}$	$1.2e^{-3}$	$1e^{-6}$	$1e^{-6}$

	$\dot{x}, \dot{y}, \dot{z}$	$\dot{\psi}_x, \dot{\psi}_y, \dot{\psi}_z$	$\dot{\theta}_7, \dot{\theta}_8$	$\dot{\theta}_9, \dot{\theta}_{10}$
	$[m/s]$	$[rad/s]$	$[rad/s]$	$[rad/s]$
Process	1^{-4}	1^{-5}	5^{-5}	1^{-4}
Measurement	4^{-2}	2^{-4}	1^{-4}	1^{-4}
Simulated sensor	$4e^{-2}$	$5e^{-2}$	$1e^{-2}$	$1e^{-2}$

$$\begin{pmatrix} \dot{\boldsymbol{\theta}} \\ \ddot{\boldsymbol{\theta}} \\ \boldsymbol{\lambda}_s \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} & \mathbf{B}_s^T \\ \mathbf{0} & \mathbf{B}_s & \mathbf{0}_s \end{pmatrix}^{-1} \begin{pmatrix} \dot{\boldsymbol{\theta}} \\ \boldsymbol{f}_{tot} \\ \frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{B}_s)\,\dot{\boldsymbol{\theta}} \end{pmatrix} \tag{18}$$

where $\mathbf{I}$ is an identity matrix. This gives the state derivative as a function of the state $\boldsymbol{q}^T = (\boldsymbol{\theta}^T \; \dot{\boldsymbol{\theta}}^T)$. Discretisation is done via a Runge-Kutta 4^{th} order numerical integration scheme to obtain the prediction for time $k+1$ based on the estimate on time k. Equation (18) is solved using a damped pseudo-inverse method, which is numerically more stable than using a regular inverse. The constraints are stabilized by projecting the predicted value onto the constraint surface. Instead of using the predicted or estimated values, the sensor measurements are used to determine which constraints are active. This turned out to be a more robust solution. The foot sensors are used to check the conditions in (13) and activate the constraint together with the equality constraints of (14) if required. The joint angle measurement of each knee used check whether or not the joint limit has been reached.

The measurement equation for the FB UKF is

$$\begin{aligned} \boldsymbol{z}_{k+1} = (&x \; y \; z \; \psi_z \; \psi_y \; \psi_x \; \theta_{7_r} \; \dots \; \theta_{10_l} \\ &\dot{x} \; \dot{y} \; \dot{z} \; \dot{\psi}_z \; \dot{\psi}_y \; \dot{\psi}_x \; \dot{\theta}_{7_r} \; \dots \; \dot{\theta}_{10_l})^T_{k+1} \end{aligned} \tag{19}$$

The trunk position measurements x,y and z are calculated using the encoders and forward kinematics. The frame in which they are expressed depends on the current support setting as detected by the foot sensors. It is either the frame of the left or the right foot. The orientation measurements ψ_z, psi_y and ψ_x come from the IMU mounted on the trunk. For the SEA joints, the joint encoder measurements are as joint angle measurements θ_{9_r}, θ_{10_r}, θ_{9_l} and θ_{10_l}. For the other joint angle measurements θ_{7_r}, θ_{8_r}, θ_{7_l} and θ_{8_l}, the motor encoder measurements have to be used, multiplied with the gear ratio. The velocities of the trunk $\dot{x}$, $\dot{y}$ and $\dot{z}$ are acquired from the IMU. For the Euler angle rate of change $\dot{\psi}_z$, $\dot{\psi}_y$ and $\dot{\psi}_x$, the Euler angle derivatives are used as a measurement. Similarly, the derivatives of the encoder measurements are used as joint angular velocity measurements $\dot{\theta}_{7_r}$ to $\dot{\theta}_{10_l}$.

The UKF can be tuned in various ways. One of which is the choosing the set of sigma points used to approximate the posterior probability density function [31]. This approximation is obtained by propagating all sigma points through the nonlinear system. The symmetric set was chosen for the FB UKF because the equal weights for each sigma point make estimation more robust, especially near a constraint surface. The algorithm of the UKF that was used can be found in appendix B.

Finally, the noise covariance values have to be set. The simulated sensor noise covariance was obtained from sensor data sheets and by inspecting sensor output data. The process and measurement noise were tuned to overcome specific error sources. For example, unmeasured backlash is present in two of the hip joints of the TUlip making the angle measurements more uncertain. Also, more trust was placed in the model for several velocities to filter the noise. The values are given in table II. The estimates obtained by the FB UKF can finally be used to calculate the CoM using forward kinematics.

VI. Evaluation, Application and Results

This section consists of a simulation study an application to the TUlip humanoid robot. In the first part, ground truth data will be obtained via a dynamic simulation. The simulated estimator uses the same dynamics for the prediction but with either a parameter error or sensor bias. As the LIPM KF only estimates the CoM, the CoM estimates will be compared for both estimators. The actual state estimate obtained by the FB UKF will be evaluated individually. In the preliminary application, the estimates will be compared with the measurements. The set-up and results are given in the following subsections.

A. Simulation Study

Because both measurements and predictions are suffering from inaccuracies, it is important to know how an estimator performs under these circumstances. The overall robustness and the dependency on different uncertainties can be well investigated using simulation. This will be done for both estimators for several movements to see if the results are general or only hold for specific types of motion.

To do so, simulations with the FFH model will be used as the ground truth. A total of 3 different movements will be simulated. The first motion is ased on a balancing strategy using the ankles. The second motion is more dynamic and uses both the ankles and upper body momentum to balance. A wrench controller on the lower legs was used to emulate balance control of the ankles. The third movement is based on a walking motion. As there is currently no walking controller on the TUlip, it is a walking motion in mid air such that it matches with the experiment in the next section. Each simulation lasts 2 seconds. They are shown in figure 4. The direction of movement is the x-direction for all configurations. So the upper body movement in configuration A and B as well as the extension of the legs in configuration C are in the x-direction. The direction for which play has the biggest influence shall be the y-direction because of the play in the hip abduction and adduction joints.

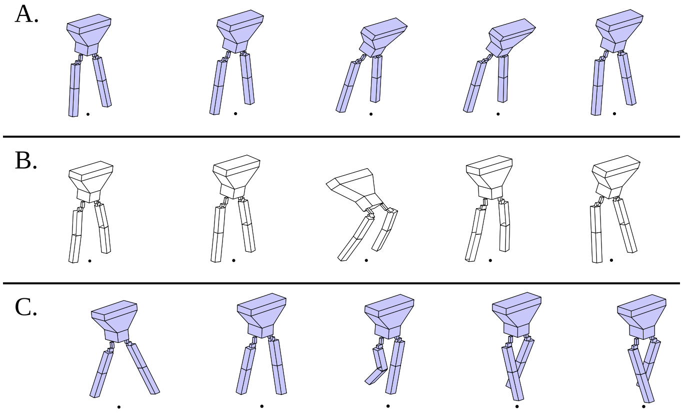


Fig. 4. Configurations used in simulation. From left to right, the robot is shown at $0s$, $0.5s$, $1s$, $1.5s$ and $2s$. The first two motions shown are, from top to bottom, based on the ankle and hip balancing strategy while standing. The third motion is a walking pattern while in the air. Here, the feet are not in contact with the ground. The black dot indicates the projected centre of mass on the ground.

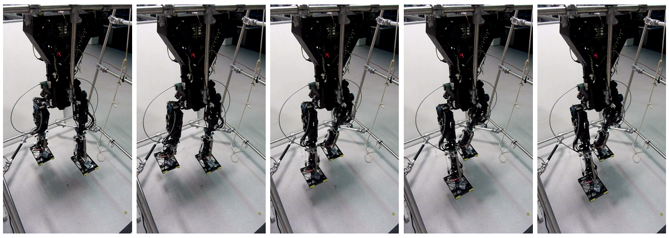


Fig. 5. The movement used for the experimental validation, implemented on the TUlip humanoid robot. It is a walking motion with the robot hanging in its rack and it based on configuration C of the simulation study as seen in figure 4. The video stills show a single step.

Next, several uncertainties were introduced. Sensory noise was added to he true state and state derivative. The simulated sensor noise is given in table II. Then, backlash was added. This was done for the joints where it was identified to be a problem, namely the hip rotation and hip abduction and adduction [10]. The backlash was systematically varied in 10 samples between 0° and 3° for each type of joint. It was modelled as a bias on the measurement. To make the effect a bit more realistic, the direction of the bias changed with the direction of the torque during simulation. The final measurement values including noise and possible bias were used as measurements of each estimator. Note that for the LIPM KF, these measurements are inserted in the kinematic model to obtain the CoM measurements. After that, the model parameters were systematically varied over 10 samples. Masses and inertias as well as CoM locations were varied between -10% and 10%. Dimensional properties were varied between -5% and 5%. These changes were then applied to the kinematic and dynamic models. The effect of link flexibility was not specifically investigated. There are 23 parameters and 2 different joints suffering from backlash giving a total of 25 different sources of uncertainty. Each source is varied in 10

TABLE III
THE AVERAGE RMSE OF THE COM ESTIMATIONS FOR ALL PARAMETER VARIATIONS FOR BOTH ESTIMATORS FOR EACH CONFIGURATION. THE ITALIC VALUES FOR THE LIPM KF ARE NOT ESTIMATED BUT CALCULATED USING FORWARD KINEMATICS.

		A	B	C
1. LIPM KF	x	0.0030	0.0017	0.0046
	y	0.0002	0.0002	0.0018
	z	*0.0015*	*0.0015*	*0.0015*
	$\dot{x}$	0.1883	0.1053	0.1811
	$\dot{y}$	0.0066	0.0013	0.0646
	$\dot{z}$	*0.0002*	*0.0008*	*0.0002*
2. FB UKF	x	0.0010	0.0068	0.0009
	y	0.0004	0.0006	0.0022
	z	0.0016	0.0027	0.0016
	$\dot{x}$	0.0522	0.1185	0.0675
	$\dot{y}$	0.0086	0.0157	0.0742
	$\dot{z}$	0.0088	0.0138	0.0159

steps for 3 motions. This results in a total of 750 simulations per estimator.

In the evaluation, first of all, the performance of the CoM estimation in the presence of parameter errors shall be compared between the configurations for each estimator. After that, the influence of different types of parameter errors and backlash on the CoM estimation of each estimator shall be compared. Lastly, the state estimations and foot location estimation of the FB UKF will be evaluated. For these comparisons, the root-mean-square error (RMSE) will be used as a metric. It is calculated for each simulation for both estimators. The results will be compared using box plots. Here, the average shall be used as an indicator of accuracy and the spread in the data as an indicator for robustness. To make comparisons between different data sets an ANOVA with a with a significance level of 0.01 was used to distinguish sets with different means. Levene's test for homogeneity of variance with a significance level of 0.01 will be used to find differences in variance.

B. Simulation Results

Table III summarises the results. The RMSE of the CoM estimate for each trial has been averaged over the configurations. The effect of backlash was excluded from the average as it will be inspected separately. In the rest of this subsection, more detailed results of both estimators will be shown.

1) Linear Inverted Pendulum Kalman Filter: The RMSE for all parameter variations is shown per configuration in the box plots of figure 6. The black dots indicate outliers. The data sets that are coupled with a horizontal bar are significantly different. From this, it can be observed that the performance differs per configuration. The estimator performs the least good for configuration C. The estimation of the x-position was better for configuration B. The vertical position can not be estimated with the LIPM KF so it was obtained using forward kinematics and no significant differences were found. In fact, no significant differences were found between the configurations for any of the forward kinematics calculations. Figure 8 shows the RMSE of configuration A, grouped per type of uncertainty. It can be seen that the spread is the largest when varying mass and backlash. Using Levene's test, the variance was indeed found to be significantly larger for these two types. Errors in the CoM positions were found to have the least influence on the variance. The same was found for the variances of the other configurations B and C, despite of the averages being different. When compared to forward kinematics, no significant reduction in variance was found for the backlash. The average RMSE was the same or even greater for the x-position.

2) Full-Body Unscented Kalman Filter: Figure 7 shows the effect of parameter variations for different configurations of the FB UKF. For configurations A and C, the estimation in x-direction was significantly better than the LIPM KF. In configuration B, though, the FB UKF performed worse in the x-direction. For the y and z-estimation, no differences were found. Figure 9 shows the influence of the different types of uncertainties. The variance caused by errors in mass parameters is significantly larger than the rest. The effect of backlash is reduced significantly in terms of variance as compared with the LIPM KF. This can be seen in figure 10, where a single simulation is shown in which 3° of backlash was added. In figure 11, the estimated velocities are shown. For configuration B, backlash was also reduced in the y-direction. For configuration C, the effect of backlash was not reduced. Moreover, the influence of the parameter errors was reduced as well in the first configuration. As opposed to the LIPM KF, the FB UKF estimates more than just the CoM. Figure 12, for example, shows the joint angles in the hip for the same simulation as figure 10. It can be seen that the bias of the hip abduction and adduction has indeed been reduced. Figure 13 shows the trunk velocities. In particular, the vertical velocity of the trunk is less noisy. However, the x-direction still contains noise and the y-direction has some small offsets. For completeness, figure 18 shows the distance between both feet, which can be used to construct the support polygon.

C. Application to the TUlip Humanoid Robot

Although a simulation gives insight in influencing factors, it does not say much about the actual performance of an estimator. Sensors may show drift and there might be more sources of uncertainty than modelled. Moreover, it was not investigated how a combination of wrong parameters influences the performance. Though, this is likely to be the case for the real robot. To find out more about these issues, a preliminary application is done.

Because there is currently no stable balance or walking controller implemented on the TUlip, a walking motion in mid air was implemented. Apart from different gains for PD control, the controller implementation was the same as for configuration C in the simulation study. The implemented motion is shown in figure 5.

Recorded data from the TUlip was imported and provided to the FB UKF. Estimation was done offline, not embedded in the closed-loop controller of the TUlip. The estimation was performed on a 2 GHz computer with 6 GB memory. The torques as measured by the TUlip were provided to the FFH

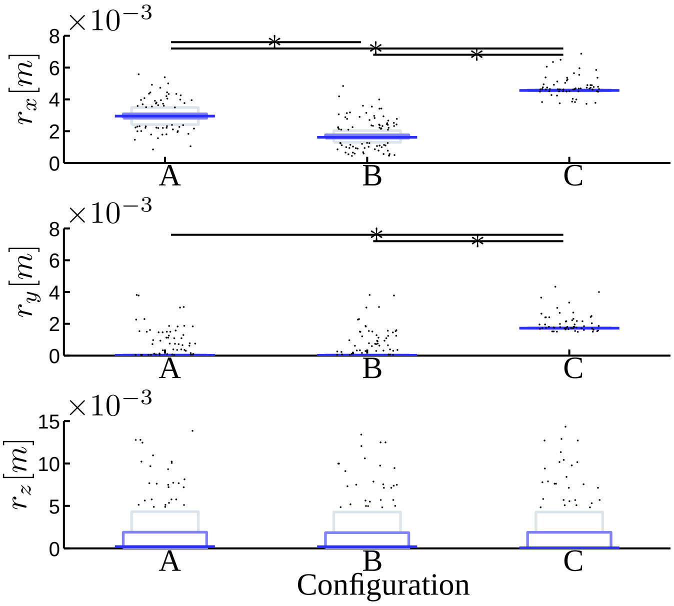


Fig. 6. Box plots showing the RMSE of the CoM position estimations using the LIPM KF for all parameter variations for each configuration. The connected data sets marked with an asterisk are significantly different.

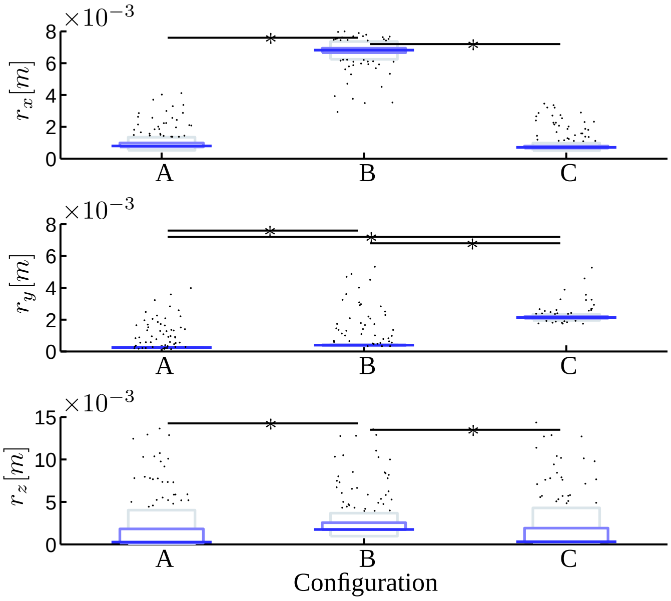


Fig. 7. Box plots showing the RMSE of the CoM position estimations using the FB UKF for all parameter variations for each configuration. The connected data sets marked with an asterisk are significantly different.

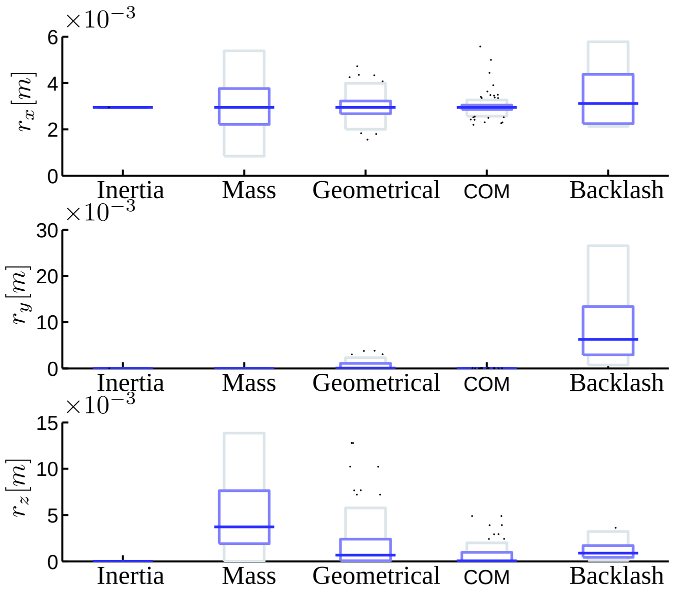


Fig. 8. Box plots showing the RMSE of the CoM estimation using the LIPM KF for each type of uncertainty for configuration A.

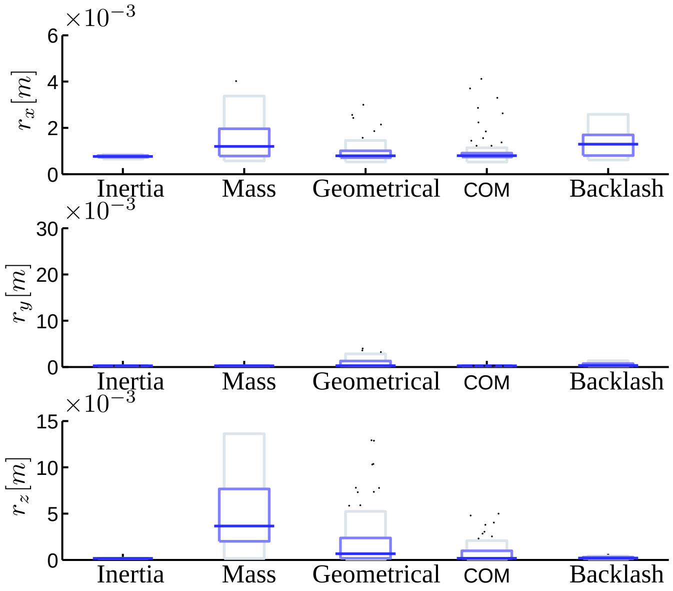


Fig. 9. Box plots showing the RMSE of the CoM estimation using the FB UKF for each type of uncertainty for configuration A.

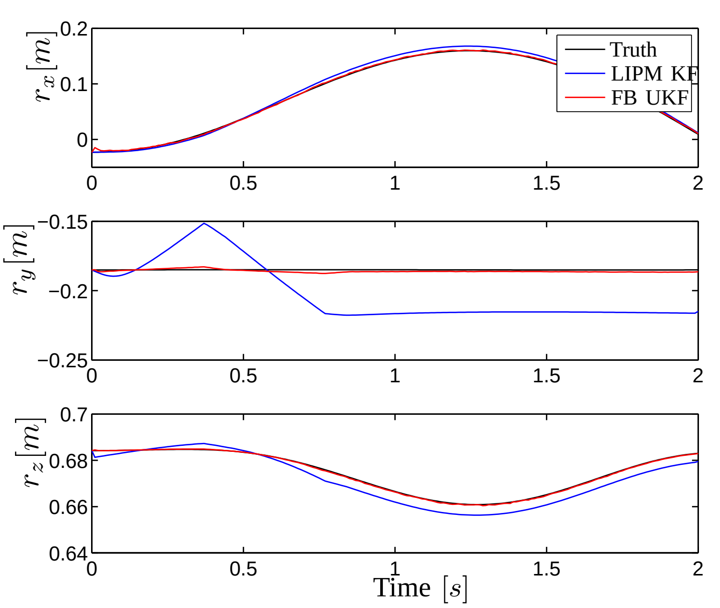


Fig. 10. CoM estimation of configuration A with the LIPM KF and the FB UKF. In this simulation, backlash of 3° was added to the hip abduction and adduction. It was modelled as a torque dependent measurement bias.

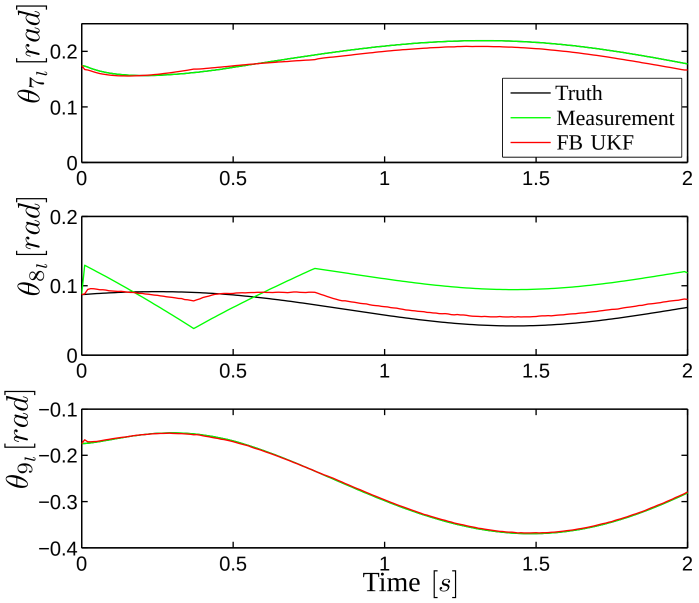


Fig. 12. Joint angle estimation of the left hip joints during configuration A with the FB UKF from the same simulation as figure 10. The measurement clearly θ_{8_l} has a torque dependent bias of 3° representing the unmeasured play.

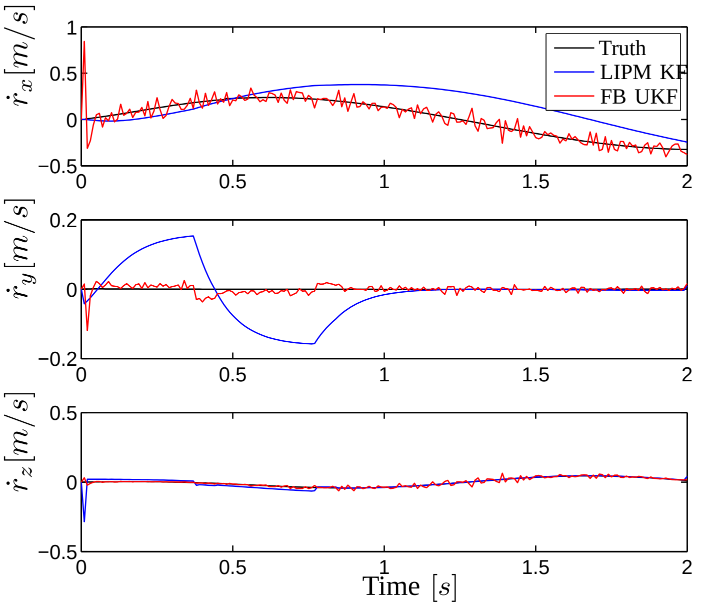


Fig. 11. CoM velocity estimation of configuration A with the LIPM KF and the FB UKF, obtained from the same simulation as figure 10

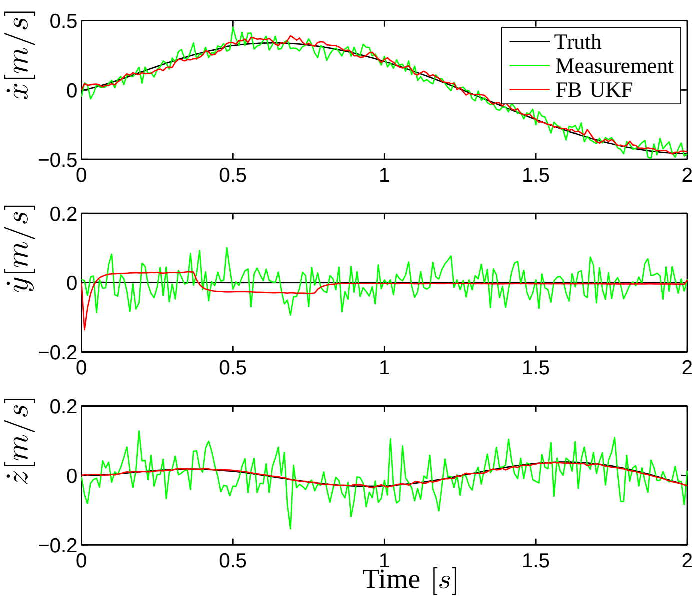


Fig. 13. Joint angle estimation of the left hip joints during configuration A with the FB UKF from the same simulation as figure 10.

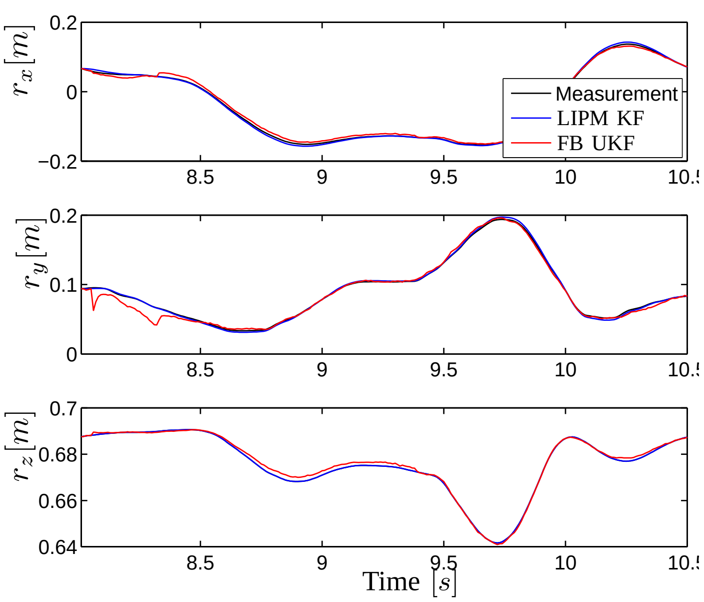


Fig. 14. CoM estimation as done in the experiment. No ground truth data was available so the estimates are compared with measured data.

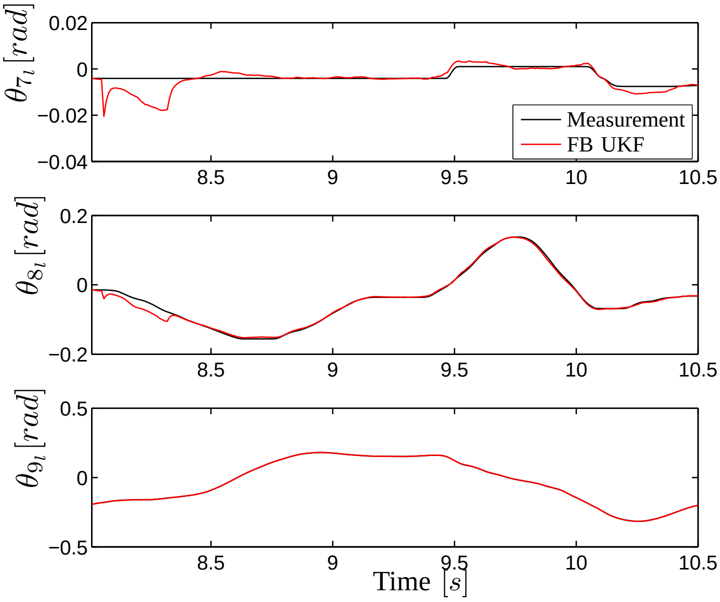


Fig. 16. Joint angle estimation, using the same experiment was used for estimating the CoM in figure 14.

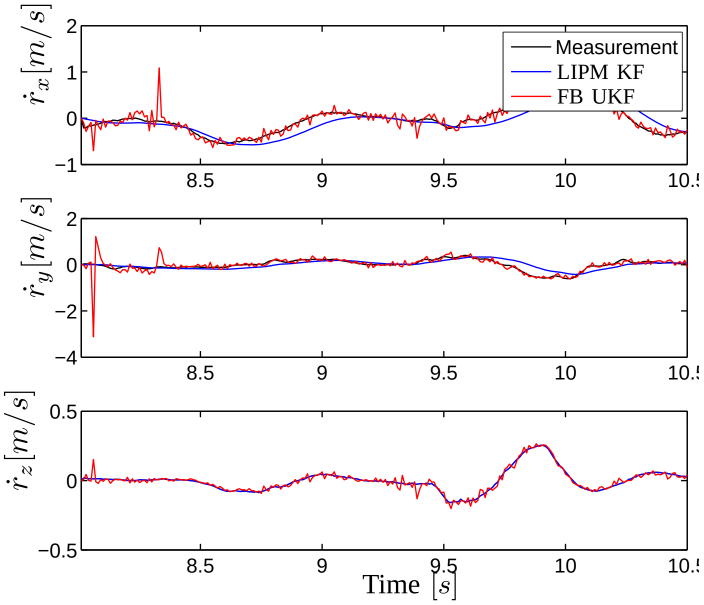


Fig. 15. CoM velocity estimation, using the same experiment was used for estimating the CoM in figure 14.

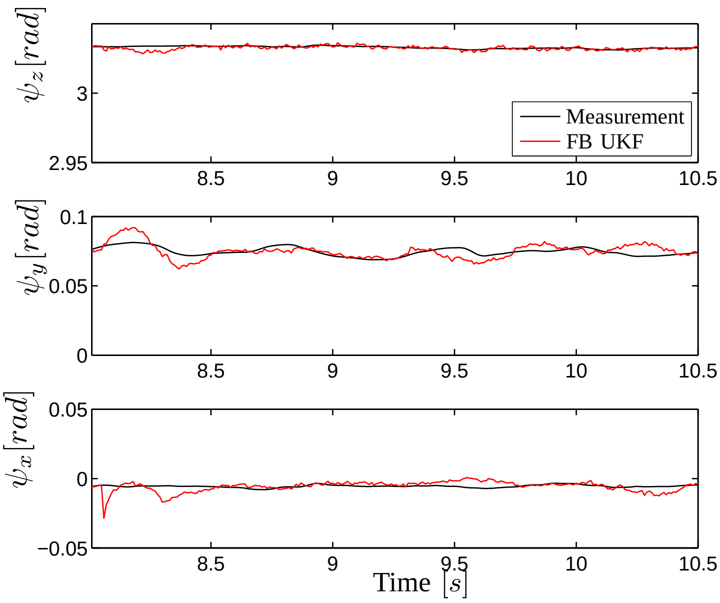


Fig. 17. Euler angle estimation, using the same experiment was used for estimating the CoM in figure 14.

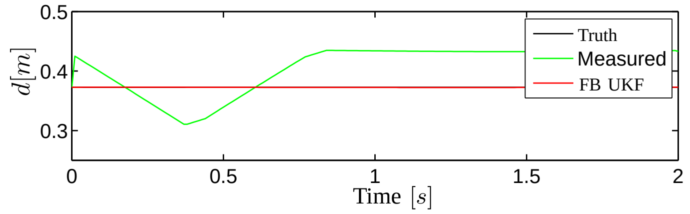


Fig. 18. The distance between the feet as estimated by the UKF for configuration A in the same simulation as figure 10.

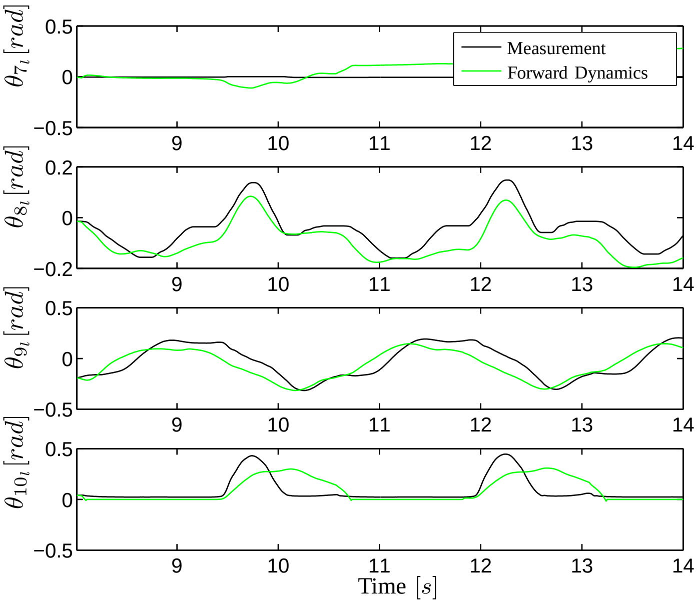


Fig. 19. A forward dynamics simulation using the measured joint torques from the experiment. The measured angles are shown together with the simulated angles using an open-loop simulation for several steps.

model and an open-loop simulation is performed. Damping was added to the joints in the FHH model and the damping coefficients β were used as a tuning parameter. The values can be seen in table IV. For estimation, the same noise covariance matrices were used as in the simulation. The measured angles and estimated angles were compared and both used to calculate the CoM. Finally, the estimated CoM and the measuerd CoM were also compared.

D. Experiment Results

Figures 14 and 15 show the CoM position and velocity estimation. Figures 16 show the estimated hip angles as compared with the measured angles. The estimated Euler angles are shown in figure 17. The simulation that was done with the FFH model using the torques measured by the TUlip is shown in figure 19.

VII. Discussion

In this report, a nonlinear estimator for a humanoid robot was proposed using the full-body dynamics of the robot. It was compared with an estimator using a simpler LIPM. Several observations and issues will be discussed in the following subsection.

TABLE IV
Damping added to the joints in the FFH model, used for tuning.

	θ_7	θ_8	θ_9	θ_{10}
$\beta[Nms/rad]$	2	2	0.75	0.6

A. Compensating for Sensor Bias

The play in the joints was not modelled with an extra degree of freedom, but as a varying bias on the measurement, dependent on the direction of the torque. A varying bias itself could also represent a loose or misaligned encoder. Using forward kinematics, such a bias directly influences the derived states like the CoM or feet locations. Making use of the LIPM KF as a prediction, this effect remained, because the filter directly uses forward kinematics to fit the pendulum model onto the real robot. The FB UKF, however, allows for fine-tuning the covariance on individual joints and for two of the three configurations, the influence of this bias was significantly lower. In configuration B, though, the estimates were not improved. The same can be observed in the experiment. The hip abduction and adduction is known to suffer from unmeasurable play. If this is compensated for, the estimated angles should have an offset from the measured ones. Though, in figure 16 it seems the estimator is tracking the measured angle θ_{8_l}, rather than compensating for sensory bias. For the hybrid state of configuration B with both feet in the air, the no support state, there is no additional information that tells the estimator how the legs are moving with respect to the trunk, apart from the encoders. However, these encoders suffer from measurement bias. In configuration A and B, with both feet on the ground in the double support state, the feet are not moving. This information is provided by the foot sensors to the model, translated into a constraint on both feet. With one foot on the ground in the single support state, it is therefore expected that the sensor bias can be partly reduced in most likely the stance leg only. Additional sensors like acceleration measurements or even IMUs in both feet could help to overcome this problem.

B. Robustness against Parametric Errors

With the simulation study, the sensitivity to model parametric errors can be checked. This gives an indication of the robustness. Using forward kinematics, the influence of varying model parameters is the same among the different configurations. For the LIPM KF as well as for the FB UKF, this effect is depicted in figures 6 and 7. It was found that the variance in the data was also the same for almost all data sets when comparing the two figures. Moreover, it was also similar to the variance of a pure kinematic calculation. When looking at the figures 8 and 9, it can be seen that the FB UKF compensates for parametric errors to some extent. The variance caused by the mass, for example, is lower in the x-estimation. Apart from that, the UKF itself can be altered to do parameter estimation or even dual estimation. [32]. The latter estimates parameters and states simultaneously. It can be useful in case parameters change during motion, for example, when the robot picks up and carries a load. This gives the FB UKF, when it is well tuned, an edge as compared to the LIPM KF. Lastly, in practice, when obtaining the robot parameters, it is suggested to focus on geometric parameters and body masses, instead of local CoM positions or inertia. This is true for both of the estimators as well as for a forward kinematic calculation.

C. Simple versus Complex Models

In this research, two estimators were compared. One reduced the complex robot dynamics to a two-dimensional pendulum and the other included almost all degrees of freedom of the robot. The effect of increasing model complexity can best be seen from figures 6 and 7. In configuration A, the FB UKF clearly outperforms the LIPM KF. The reason becomes apparent when inspecting the estimation of $\dot{r}_x$ in figure 11. As can be seen, the estimation of the velocity is incorrect. It was explained in section VI-A that backlash had limited influence on the x-direction in the simulations. Therefore, this can not be the cause of the incorrect prediction in figure 6. Instead, the LIPM KF simply predicts an incorrect velocity causing a bad estimate for the position as well. The reason behind this, is a bad fit from the pendulum onto the robot. This is even more the case for configuration C where the robot is in the air. Although no ground truth was available, the experiment suggests the same, as can be seen from 15. The FB UKF, though, can predict a no support state much better. Configuration B best reveals the crux of this section. While the LIPM KF might predict in the wrong direction in configuration A, it might just as well predict in the right direction, without actually having knowledge about the complex and fast motion that is happing. This is the case for configuration B. It shows that simple models can work in specific cases, whereas the FB UKF was unable to follow the fast movements in configuration B. Tuning the covariances tailor-made for configuration B, estimation performance can be significantly improved, but the tuning deteriorates the performance of the other motions. This demonstrates a disadvantage of using complex models, namely, that they are harder to tune for multiple motions.

D. Versatility

One of the benefits of the FB UKF is that it estimates all current states and it can thus provide more than just the CoM. By removing errors in the kinematic chain, the locations of the feet were also estimated and improved as compared to using forward kinematics. This is shown in figure 18. Figure 13 shows that the FB UKF can also filter noisy measurements of individual states. However, the figure also shows it can not filter all noise present in the measurements. One has to place trust in some noisy measurements to be able to remove sensory bias present in the kinematic chain. This directly results in a noisy CoM velocity estimate, seen in both the simulation results of figure 11 and the experimental results from figure 15 and 17. It can be easily solved by including a non-predictive filter like a Butterworth filter, but it does reveal an important aspect of the FB UKF. Though it can be used to solve multiple issues in the kinematic chain, it does not solve all because it relies on the availability of redundant information. Tuning the filter should focus on tackling the specific issues present in the robot. Hence, it is suggested to identify these issues before application of the estimator.

E. Numerical Implementation

In sections IV and V, several choices were made to decrease the compuational burden. However, the current filter was not implemented on the robot itself. Despite doing the estimations at $100Hz$, estimation generally took twice as long as the duration of the estimated motion. Currently, the prediction with the equations of motion is done in C-code. However, implementation of the equations of motion in C-code was done by directly exporting symbolic equations to C. It is therefore suggested to look at efficient implementations like the Articulated-Body Algorithm [36]. Also, the UKF itself was not implemented in C but in Matlab m-code, but the UKF calculations themselves are not as computationally complex as the equations of motion. Two other options that could speed up the calculations are reducing the amount of sigma points and implementing a numerically more efficient method like the Square-Root Unscented Kalman Filter. This square-root variant of the UKF increases computational speed and guarantees a positive semi-definite state covariance matrix [37]. Positive semi-definiteness can also be an issue in the numerical implementation of the normal UKF, where the matrix square root is calculated as shown in algorithm 1 of appendix B. An incorrect prediction that is weighed using a relatively large weigh factor can, for example, cause numerical instability. This is the reason for choosing the general symmetric sigma point set. Lastly, it should be noted that adding constraints can cause badly scaled matrices, causing problems when implementing equation (18). With a damping factor of $\lambda = 0.01$ for the damped pseudo-inverse, this problem is currently circumvented.

F. Implementation on the TUlip

While using measured data to test the tracking of the FB UKF, the measured torques were first checked in an open-loop simulation of which the final results are given in figure 19. It shows several steps of the walking pattern. The simulation was used to tune the damping in the model joints. This revealed several issues. First of all, a gain error was found between the torques measured by the series elastic actuation and the torques calculated from the pwm signals sent to the motor. It was solved manually by matching the signals, though it requires further investigation on motor drive level. When the gain error in the measurements was not manually compensated for, the estimator performance degraded. Furthermore, a stiffness was found to be present in the hip rotation. The joint seemed to prefer to go to its zero position. This is caused by the electric cables that run from the trunk over the leg, pulling the leg back to its zero position in some situations when the hip rotates around its z-axis. Moreover, from figure 19 it becomes clear the hip rotation drifts away. This was caused by stick-slip in the joint. Because an error in the angle is detected in the low-level control, the motor sends out a signal which is too small to overcome friction. This does not result in a motion of the robot, but does result in the hip rotation drifting away during the open-loop simulation. The effect was also observed for the hip abduction. The FB UKF was found to be a useful tool in discovering software and hardware related issues. Finally, although it was not investigated specifically, the experiment indicates there is some robustness against model input errors. However, no quantitative information can support this.

G. *Future Work*

Tuning the filter to work for various movements was found to be a difficult task. Making sure the FB UKF estimates all sorts of motion without having to retune the filter in between different configurations still requires further research. One major area of future work is to implement the proposed observer in a real-time system for a humanoid robot. Then, potential control performance improvements in a closed loop need to be investigated as well. Experiments should be done to acquire ground truth data about the states of the robot. Motion capture or force pressure sensors could be useful to obtain this information. Making the design robust, numerically stable and efficient are also issues future research could focus on. Finally, although different constraints can be added to the system, switching constraints and the effect of ground impact or other unexpected disturbance forces have not been investigated yet and require further research.

VIII. Conclusion

In this report, the Full-Body Unscented Kalman Filter (FB UKF) was presented. This estimator makes use of the Floating Feetless Humanoid (FFH) model for prediction and the Unscented Kalman Filter to estimate the state of a humanoid robot. The FFH model is, as the name suggests, a humanoid model with a floating base, without feet, modelled in 3D. Constraints were added to model ground contact and joint limits. The main benefit of this design is that it allows for estimating all states of the robot, except for the feet angles. When comparing the CoM estimates with a Kalman Filter that uses Linearised Pendulum Model for prediction (LIPM KF), the FB UKF reveals another strength. It can decrease the effect of play in the joints, modelled as a varying measurement bias, thereby improving the estimation of all derived states like the CoM. Both estimators were found to be robust against parametric errors. The LIPM KF, though, is easier to implement and tune for various types of motion.

Appendix A
Robot Kinematics

The following subsections give a compact overview of methods and definitions used in deriving the equations of motion. Readers are referred to [38] [39] [40] for more detailed explanations.

A. *Geometrical Background*

A rigid body motion is a mapping $g : \mathbb{R}^3 \to \mathbb{R}^3$ where distance and orientation are preserved. First, consider the case of pure rotation. The proper description of the motion that satisfies these conditions is via the Special Orthogonal group $SO(3)$, which is the space of rotation matrices. It is defined as

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} : \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1 \right\} \tag{20}$$

This is a group under the operation of matrix multiplication. Moreover, it is a smooth manifold that has three degrees of freedom representing the orientation. A group which is also a smooth manifold is called a Lie group, to which a Lie algebra $\mathfrak{so}(3)$ can be associated. On its own, a Lie algebra is nothing more than a vector space with an operator satisfying some axioms. Associated to a Lie group, it is the tangent space at the identity of the Lie group. So intuitively, the Lie algebra describes some velocity, which makes it useful for the analysis of rigid bodies. The Lie algebra is

$$\mathfrak{so}(3) = \left\{ \tilde{\boldsymbol{\omega}} \in \mathbb{R}^{3\times 3} : \tilde{\boldsymbol{\omega}} = -\tilde{\boldsymbol{\omega}}^T \right\} \tag{21}$$

which means it is skew-symmetric. Now, let us extend these notions to the general case that includes translation using the Special Euclidean group $SE(3)$. This group is defined as

$$SE(3) = \left\{ \begin{pmatrix} \mathbf{R} & \boldsymbol{d} \\ \mathbf{0} & 1 \end{pmatrix} : \mathbf{R} \in SO(3), \boldsymbol{d} \in \mathbb{R}^3 \right\} \tag{22}$$

which in its turn is also a Lie group with the associated Lie algebra $\mathfrak{se}(3)$. Here, $\boldsymbol{d}$ defines the translation. The Lie algebra has the form of

$$\mathfrak{se}(3) = \left\{ \begin{pmatrix} \tilde{\boldsymbol{\omega}} & \boldsymbol{v} \\ 0 & 1 \end{pmatrix} : \tilde{\boldsymbol{\omega}} \in \mathfrak{so}(3), \boldsymbol{v} \in \mathbb{R}^3 \right\} \tag{23}$$

An action of $\mathbf{H} \in SE(3)$ on a point in $\boldsymbol{p} \in \mathbb{R}^3$ then defines the rigid transformation $g : \boldsymbol{p} \mapsto \mathbf{R}\boldsymbol{p} + \boldsymbol{d}$.

B. *Twists of a Rigid Body*

In practice, to express the mapping g, let $\mathbf{H}^i_j \in SE(3)$. Then, the configuration of a point $\boldsymbol{p}$ on a rigid body with frame j relative to a spatial frame i is given by

$$\bar{\boldsymbol{p}}^i = \mathbf{H}^i_j \bar{\boldsymbol{p}}^j \tag{24}$$

where $\bar{\boldsymbol{p}} = (\boldsymbol{p}, 1)^T$. For simplicity, the bar will be omitted from here on. Now, the Lie algebra can be used to express the velocities. First, consider the pure rotational case $\boldsymbol{p}^i = \mathbf{R}^i_j \boldsymbol{p}^j$. Differentiating with respect to time gives $\dot{\boldsymbol{p}}^i = \dot{\mathbf{R}}^i_j \boldsymbol{p}^j$. It can be rewritten to $\dot{\boldsymbol{p}}^i = \dot{\mathbf{R}}^i_j {\mathbf{R}^i_j}^{-1} \boldsymbol{p}^i$. Here, the expression $\tilde{\boldsymbol{\omega}}^s = \dot{\mathbf{R}}\mathbf{R}^{-1}$ is an element of the Lie algebra. Because it is skew-symmetric, it can be written as a vector $\boldsymbol{\omega}^s$ containing the rotational velocities in the spatial frame i. In a similar way, $\tilde{\boldsymbol{\omega}}^b = \mathbf{R}^{-1}\dot{\mathbf{R}}$ can be found, which gives the rotational velocities in the body frame j as a vector $\boldsymbol{\omega}^b$. Now again, this can be extended to the general case. It can then be shown that the velocities of a rigid body are given by

$$\begin{aligned}\mathbf{V}_j^{is} &= \dot{\mathbf{H}}_j^i {\mathbf{H}_j^i}^{-1} &(25)\\ \mathbf{V}_j^{ib} &= {\mathbf{H}_j^i}^{-1}\dot{\mathbf{H}}_j^i &(26)\end{aligned}$$

Here, $\mathbf{V}_j^{is} \in \mathfrak{se}(3)$ and $\mathbf{V}_j^{ib} \in \mathfrak{se}(3)$ are the spatial and body twist respectively, also called the spatial and body twist. When writing out these expressions, it can indeed be seen that they have the form given in (23). A twist can also be expressed in twist coordinates. Define the vee operator $\vee$ as $(\cdot)^\vee : \mathfrak{se}(3) \to \mathbb{R}^6$. Then, a twist $\mathbf{V}$ of the form given in (23) can be expressed in twist coordinates as

$$\mathbf{V}^\vee = \begin{pmatrix} \tilde{\boldsymbol{\omega}} & \boldsymbol{v} \\ 0 & 1 \end{pmatrix}^\vee = \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{pmatrix} = \boldsymbol{v} \tag{27}$$

where $\boldsymbol{v}$ has the form of a vector. For completeness, the wedge operator $\wedge$ shall be used for the inverse operation $(\cdot)^\wedge : \mathbb{R}^6 \to \mathfrak{se}(3)$. When inspecting the terms of the twists, they can be interpreted as follows. The spatial twist $\mathbf{V}_j^{is}$ is the velocity of body j relative to some frame i as seen from frame i. The body twist $\mathbf{V}_j^{ib}$ is the velocity of body j relative to some frame i as seen from frame j.

C. Twists of a Serial Link

1) The Adjoint Transform: To obtain the twists that describe the velocities of a serial link, let us first find a way to relate two twists to each other. From equations (25) and (26) it can be easily seen that

$$\mathbf{V}_j^{is} = \mathbf{H}_j^i \mathbf{V}_j^{ib} {\mathbf{H}_j^i}^{-1} \tag{28}$$

which maps the twist from body to spatial coordinates. In twist coordinates, we write this mapping as

$$\boldsymbol{v}_j^{is} = \mathrm{Ad}_{\mathbf{H}_j^i} \boldsymbol{v}_j^{ib} \tag{29}$$

where

$$\mathrm{Ad}_{\mathbf{H}} = \begin{pmatrix} \mathbf{R} & \tilde{\mathbf{d}}\mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \tag{30}$$

is called the Adjoint transformation that maps a twist from one frame to another frame as $\mathrm{Ad}_{\mathbf{H}} : \mathbb{R}^6 \to \mathbb{R}^6$. This will be useful later on.

2) Unit Twists: The next step is to look at the twists of joints with a single degree of freedom θ. For such a joint, define a unit twist as

$$\mathbf{T}_j^{is} = \frac{\partial}{\partial\theta}\left(\mathbf{H}_j^i\right) {\mathbf{H}_j^i}^{-1} \tag{31}$$

which when multiplied with a magnitude $\dot{\theta}$ forms the spatial twist between link i and j as $\mathbf{V}_j^{is} = \mathbf{T}_j^{is}\dot{\theta}$. Or, in twist coordinates as $\boldsymbol{v}_j^{is} = \boldsymbol{t}_j^{is}\dot{\theta}$. When rearranging (26) to $\dot{\mathbf{H}}_j^i = \mathbf{V}_j^{is}\mathbf{H}_j^i$ the solution of the differential equation can be found which gives a relation between elements of $SE(3)$ and $\mathfrak{se}(3)$.

$$\mathbf{H}_j^i(\theta) = e^{\mathbf{T}_j^{is}\theta}\mathbf{H}_j^i(0) \tag{32}$$

Thus, the exponent of a twist gives the motion of a body relative to its initial position $\mathbf{H}_j^i(0)$ after some trajectory of θ.

3) The Geometric Jacobian: Finally, we can obtain the Jacobians that give the velocity kinematics. First, the velocities of end effector n will be derived. The spatial velocity is

$$\begin{aligned}\mathbf{V}_n^{0s} &= \dot{\mathbf{H}}_n^0 {\mathbf{H}_n^0}^{-1} \\ &= \frac{\partial \mathbf{H}_n^0}{\partial\theta_1}\mathbf{H}_0^n\dot{\theta}_1 + \ldots + \frac{\partial \mathbf{H}_n^0}{\partial\theta_1}\mathbf{H}_0^n\dot{\theta}_n\end{aligned}$$

which can also be written in twist coordinates.

$$\begin{aligned}\boldsymbol{v}_n^{0s} &= \mathbf{J}_n^s\dot{\boldsymbol{\theta}} &(33)\\ \boldsymbol{v}_n^{0s} &= \left(\frac{\partial \mathbf{H}_n^0}{\partial\theta_1}\mathbf{H}_0^n{}^\vee \ldots \frac{\partial \mathbf{H}_n^0}{\partial\theta_n}\mathbf{H}_0^n{}^\vee\right)\begin{pmatrix}\dot{\theta}_1\\ \vdots \\ \dot{\theta}_n\end{pmatrix} &(34)\end{aligned}$$

the vee operator $\vee$ transforms a twist into twist coordinates. The individual terms are then given as

$$\begin{aligned}\frac{\partial \mathbf{H}_n^0}{\partial\theta_i}\mathbf{H}_0^n &= \left(\mathbf{H}_{i-1}^0\frac{\partial \mathbf{H}_i^{i-1}}{\partial\theta_i}\mathbf{H}_n^i\right)\mathbf{H}_0^n \\ &= \mathbf{H}_{i-1}^0\left(\frac{\partial \mathbf{H}_i^{i-1}}{\partial\theta_i}\mathbf{H}_{i-1}^i\right)\mathbf{H}_0^{i-1} \\ &= \mathbf{H}_{i-1}^0\mathbf{T}_i^{i-1s}\mathbf{H}_0^{i-1} \\ \frac{\partial \mathbf{H}_n^0}{\partial\theta_i}\mathbf{H}_0^n{}^\vee &= \mathrm{Ad}_{\mathbf{H}_{i-1}^0}\boldsymbol{t}_i^{i-1s} &(35)\end{aligned}$$

This completes the spatial Jacobian $\mathbf{J}_n^s$ of body n. It can be seen that the jth column of $\mathbf{J}_n^s$ is the unit twist of joint θ_j, transformed to the global frame 0. Likewise, Jacobian in body coordinates can be derived and it can be shown that

$$\mathbf{J}_n^s = \mathrm{Ad}_{\mathbf{H}_n^0}\mathbf{J}_n^b \tag{36}$$

This finally gives the body velocities of the end-effector n.

$$\boldsymbol{v}_n^{0b} = \mathbf{J}_n^b\dot{\boldsymbol{\theta}} \tag{37}$$

D. Wrenches

In a similar way, the forces can be added to the system. Define a wrench to be a pair of a force and a moment acting on body i as

$$\mathbf{w}_i = \begin{pmatrix}\boldsymbol{f}\\ \boldsymbol{\tau}\end{pmatrix} \tag{50}$$

such that the power can be calculated from a twist and a wrench as $P = \boldsymbol{w}\cdot\boldsymbol{v} = \boldsymbol{\tau}\cdot\boldsymbol{\omega} + \boldsymbol{f}\cdot\boldsymbol{v}$. A wrench can also be transformed using the Adjoint. Under a coordinate change, the power should stay equal. Using this fact it can be seen that

$$\boldsymbol{w}_i = \mathrm{Ad}_{\mathbf{H}_i^j}^T\boldsymbol{w}_j. \tag{51}$$

And, similar to twists, the body Jacobian can be used to map external forces on a rigid body to the joint space as

$$\boldsymbol{\tau}_{ext} = \left(\mathbf{J}_n^b\right)^T\boldsymbol{w}_n. \tag{52}$$

With $\boldsymbol{\tau} \in \mathbb{R}^n$.

Algorithm 1 The Additive Unscented Kalman Filter [32]

Initialise:

$$\hat{\boldsymbol{x}}_0 \qquad \mathbf{P}_0 \qquad \mathbf{Q} \qquad \mathbf{R}$$

For $k = 1, 2, \ldots, \infty$:

1) *Sigma points:*

$$\boldsymbol{\mathcal{X}}_{k-1|k-1} = (\hat{\boldsymbol{x}}_{k-1|k-1} + \gamma\sqrt{\mathbf{P}_{k-1|k-1}}\hat{\boldsymbol{x}}_{k-1|k-1} - \gamma\sqrt{\mathbf{P}_{k-1|k-1}}) \tag{38}$$

2) *Prediction step:*

$$\boldsymbol{\mathcal{X}}^{*}_{k|k-1} = \boldsymbol{f}\left(\boldsymbol{\mathcal{X}}_{k-1|k-1}, \boldsymbol{u}_{k-1}\right) \tag{39}$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=1}^{2L} w_i^{(m)} \boldsymbol{\mathcal{X}}^{*}_{k|k-1,i} \tag{40}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=1}^{2L} w_i^{(c)} \left(\boldsymbol{\mathcal{X}}^{*}_{k|k-1,i} - \hat{\boldsymbol{x}}_{k|k-1}\right)\left(\boldsymbol{\mathcal{X}}^{*}_{k|k-1,i} - \hat{\boldsymbol{x}}_{k|k-1}\right)^T + \mathbf{Q} \tag{41}$$

3) *Correction step:*

$$\boldsymbol{\mathcal{X}}_{k-1|k-1} = (\hat{\boldsymbol{x}}_{k|k-1} + \gamma\sqrt{\mathbf{P}_{k|k-1}}\hat{\boldsymbol{x}}_{k|k-1} - \gamma\sqrt{\mathbf{P}_{k|k-1}}) \tag{42}$$

$$\boldsymbol{\mathcal{Y}}_{k|k-1} = \boldsymbol{h}\left(\boldsymbol{\mathcal{X}}_{k|k-1}, \boldsymbol{u}_{k-1}\right) \tag{43}$$

$$\hat{\boldsymbol{y}}_{k|k-1} = \sum_{i=1}^{2L} w_i^{(m)} \boldsymbol{\mathcal{Y}}_{k|k-1,i} \tag{44}$$

$$\mathbf{P}^{(\boldsymbol{y},\boldsymbol{y})}_{k|k-1} = \sum_{i=1}^{2L} w_i^{(c)} (\boldsymbol{\mathcal{Y}}_{k|k-1,i} - \hat{\boldsymbol{y}}_{k|k-1})(\boldsymbol{\mathcal{Y}}_{k|k-1,i} - \hat{\boldsymbol{y}}_{k|k-1})^T + \mathbf{R} \tag{45}$$

$$\mathbf{P}^{(\boldsymbol{x},\boldsymbol{y})}_{k|k-1} = \sum_{i=1}^{2L} w_i^{(c)} (\boldsymbol{\mathcal{X}}_{k|k-1,i} - \hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{\mathcal{Y}}_{k|k-1,i} - \hat{\boldsymbol{y}}_{k|k-1})^T \tag{46}$$

$$\mathbf{K}_k = \mathbf{P}^{(\boldsymbol{x},\boldsymbol{y})}_{k|k-1}\left(\mathbf{P}^{(\boldsymbol{y},\boldsymbol{y})}_{k|k-1}\right)^{-1} \tag{47}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}) \tag{48}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}^{(\boldsymbol{y},\boldsymbol{y})}_{k|k-1}K_k^T \tag{49}$$

Parameters: $\gamma = \sqrt{L}$ is a scaling parameter for the generation of the sigma points, where L is the dimension of the state. The weights $w_i^{(\cdot)}$ for the mean and covariance are equally distributed as $w_i^{(\cdot)} = 1/2L$.

E. Equations of Motion

With the twists and wrenches of previous sections, all information needed to set up the equations of motion is available. The twists can be made for every body inside a chain giving the total body Jacobian $\mathbf{J}^b$ for the system. Note that this Jacobian is different from the analytic Jacobian and does not require the local inertia matrices to be transformed. This can be used in for example the Lagrange Equations to get the equations of motion for the serial link.

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{\theta}}} - \frac{\partial \mathcal{T}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathcal{V}}{\partial \boldsymbol{\theta}} = \boldsymbol{\tau} \tag{53}$$

With $\boldsymbol{\theta} \in \mathbb{R}^n$ the generalized coordinates, $\mathcal{T}$ the kinetic energy and $\mathcal{V}$ the potential energy. They are

$$\begin{aligned}\mathcal{T} &= \sum_{i=1}^{n} \frac{1}{2}\left(\boldsymbol{v}_i^{0b}\right)^T \mathbf{M}_{l,i}\boldsymbol{v}_i^{0b} \\ &= \sum_{i=1}^{n} \frac{1}{2}\dot{\boldsymbol{\theta}}^T\left(\mathbf{J}_i^b\right)^T \mathbf{M}_l\mathbf{J}_i^b\dot{\boldsymbol{\theta}} \\ &= \frac{1}{2}\dot{\boldsymbol{\theta}}^T\left(\mathbf{J}^b\right)^T \mathbf{M}_l\mathbf{J}^b\dot{\boldsymbol{\theta}} \qquad (54) \\ \mathcal{V} &= \sum_{i=1}^{n} m_i g h_i(\theta_i) \qquad (55)\end{aligned}$$

where $\mathbf{M}_{l,i} \in \mathbb{R}^{6\times 6}$ denotes the local mass matrix of body i. Writing out the terms we can find the equations of motion as

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{G}(\boldsymbol{\theta}) = \boldsymbol{\tau}. \tag{56}$$

with inertia matrix $\mathbf{M}$, Coriolis matrix $\mathbf{C}$, gravity $\mathbf{G}$ and $\boldsymbol{\tau} = \boldsymbol{\tau}_j + \boldsymbol{\tau}_{ext}$ representing the joint control torques and projected end-effector forces.

APPENDIX B
UNSCENTED KALMAN FILTER

For the Full-Body Unscented Kalman Filter (FB UKF), the Unscented Kalman Filter (UKF) with additive noise is used for estimation [32]. Its working principle is shown in algorithm 1. For scaling, the general symmetric set is used [31]. First, the initial values for the state estimate and covariance $\hat{\boldsymbol{x}}_0$ and $\mathbf{P}_0$ are set, as well as the measurement and process noise $\mathbf{Q}$ and $\mathbf{R}$. Then, at each time step, the sigma points $\boldsymbol{\mathcal{X}}_{k-1|k-1}$ are distributed around the previous estimate based on the previous covariance and some scaling parameter γ. The next step is the prediction which is made by propagating the sigma points through the nonlinear function. The output is the set of transformed sigma points $\boldsymbol{\mathcal{X}}^*_{k|k-1}$. The new weighted mean and covariance are then calculated from the transformed sigma points. Next, new sigma points $\boldsymbol{\mathcal{X}}_{k-1|k-1}$ are created around the predicted values using the predicted covariance and γ. They are propagated through the nonlinear measurement equation and the weighted mean of the outcome then gives the predicted measurement. The final step is to calculate the Kalman gain and performing the update steps as in a regular Kalman filter to obtain the new estimate $\hat{\boldsymbol{x}}_{k|k}$ and covariance $\mathbf{P}_{k|k}$.

ACKNOWLEDGMENT

The author would like to thank H. Vallery for the helpful and inspiring discussions related to this research and for the contributions made to the TUlip software design. The author is grateful for the help and support G.A. van der Hoorn, ranging from software development to sensor implementations and hardware issues of the TUlip. The author is also grateful for the help of G. Liqui Lung and J. van Frankenhuyzen for their hardware related support. Lastly, the author would like to thank J. van der Weijde for the helpful discussions and for his cooperation in developing control for the TUlip humanoid robot.

REFERENCES

[1] M. Vukobratović and B. Borovac, “Zero-moment pointthirty five years of its life,” *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[2] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, “Efficient bipedal robots based on passive-dynamic walkers,” *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[3] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt, “Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models,” *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[4] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, “Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid,” *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1117–1133, 2012.

[5] A. Hof, M. Gazendam, and W. Sinke, “The condition for dynamic stability,” *Journal of biomechanics*, vol. 38, no. 1, pp. 1–8, 2005.

[6] A. L. Hof, “The extrapolated center of mass concept suggests a simple control of balance in walking,” *Human movement science*, vol. 27, no. 1, pp. 112–125, 2008.

[7] T. De Boer, “Foot placement in robotic bipedal locomotion,” *Delft University of Technology, Netherlands*, 2012.

[8] B. J. Stephens and C. G. Atkeson, “Dynamic balance force control for compliant humanoid robots,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1248–1255.

[9] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal, “Experiments with a hierarchical inverse dynamics controller on a torque-controlled humanoid,” Technical report, Tech. Rep., 1996.

[10] T. Vissers, “Sensor fusion on a humanoid robot,” Master’s thesis, Delft University of Technology, 2012.

[11] R. Miall and D. M. Wolpert, “Forward models for physiological motor control,” *Neural networks*, vol. 9, no. 8, pp. 1265–1279, 1996.

[12] D. M. Wolpert, R. C. Miall, and M. Kawato, “Internal models in the cerebellum,” *Trends in cognitive sciences*, vol. 2, no. 9, pp. 338–347, 1998.

[13] D. M. Wolpert, Z. Ghahramani, M. I. Jordan *et al.*, “An internal model for sensorimotor integration,” *SCIENCE-NEW YORK THEN WASHINGTON-*, pp. 1880–1880, 1995.

[14] K. P. Körding and D. M. Wolpert, “Bayesian integration in sensorimotor learning,” *Nature*, vol. 427, no. 6971, pp. 244–247, 2004.

[15] S. Kwon and Y. Oh, “Estimation of the center of mass of humanoid robot,” in *Control, Automation and Systems, 2007. ICCAS’07. International Conference on*. IEEE, 2007, pp. 2705–2709.

[16] C. Graf and T. Röfer, “A center of mass observing 3d-lipm gait for the robocup standard platform league humanoid,” in *RoboCup 2011: Robot Soccer World Cup XV*. Springer, 2012, pp. 102–113.

[17] S. Kwon and Y. Oh, “Real-time estimation algorithm for the center of mass of a bipedal robot with flexible inverted pendulum model,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 5463–5468.

[18] B. J. Stephens, “State estimation for force-controlled humanoid balance using simple models in the presence of modeling error,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3994–3999.

[19] D. Hauschildt, S. Kerner, S. Tasse, and O. Urbann, “Multi body kalman filtering with articulation constraints for humanoid robot pose and motion estimation,” in *RoboCup 2011: Robot Soccer World Cup XV*. Springer, 2012, pp. 415–426.

[20] C. G. Atkeson *et al.*, “State estimation of a walking humanoid robot,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 3693–3699.

[21] V. Lebastard, Y. Aoustin, and F. Plestan, “Estimation of absolute orientation for a bipedal robot: experimental results,” *Robotics, IEEE Transactions on*, vol. 27, no. 1, pp. 170–174, 2011.

[22] S. C. Patwardhan, S. Narasimhan, P. Jagadeesan, B. Gopaluni, and S. L Shah, “Nonlinear bayesian state estimation: A review of recent developments,” *Control Engineering Practice*, vol. 20, no. 10, pp. 933–953, 2012.

[23] Z. Chen, “Bayesian filtering: From kalman filters to particle filters, and beyond,” *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[24] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, 2002.

[25] N. J. Gordon, D. J. Salmond, and A. F. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” in *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, no. 2. IET, 1993, pp. 107–113.

[26] G. Burgers, P. Jan van Leeuwen, and G. Evensen, “Analysis scheme in the ensemble kalman filter,” *Monthly weather review*, vol. 126, no. 6, pp. 1719–1724, 1998.

[27] F. Daum, “Nonlinear filters: beyond the kalman filter,” *Aerospace and Electronic Systems Magazine, IEEE*, vol. 20, no. 8, pp. 57–69, 2005.

[28] I. Arasaratnam, S. Haykin, and R. J. Elliott, “Discrete-time nonlinear filtering algorithms using gauss–hermite quadrature,” *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.

[29] K. Ito and K. Xiong, “Gaussian filters for nonlinear filtering problems,” *Automatic Control, IEEE Transactions on*, vol. 45, no. 5, pp. 910–927, 2000.

[30] S. J. Julier and J. K. Uhlmann, “A general method for approximating nonlinear transformations of probability distributions,” Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, Tech. Rep., 1996.

[31] ——, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[32] R. Van Der Merwe, “Sigma-point kalman filters for probabilistic inference in dynamic state-space models,” Ph.D. dissertation, University of Stellenbosch, 2004.

[33] G. Welch and G. Bishop, “An introduction to the kalman filter,” 1995.

[34] D. Hobbelen, T. de Boer, and M. Wisse, “System overview of bipedal robots flame and tulip: Tailor-made for limit cycle walking,” in *Intel-*

ligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. IEEE, 2008, pp. 2486–2491.

[35] G. A. Pratt and M. M. Williamson, “Series elastic actuators,” in *Intelligent Robots and Systems 95.’Human Robot Interaction and Cooperative Robots’, Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1. IEEE, 1995, pp. 399–406.

[36] R. Featherstone, *Rigid body dynamics algorithms*. Springer New York, 2008, vol. 49.

[37] R. Van Der Merwe and E. A. Wan, “The square-root unscented kalman filter for state and parameter-estimation,” in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP’01). 2001 IEEE International Conference on*, vol. 6. IEEE, 2001, pp. 3461–3464.

[38] R. M. Murray and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[39] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons New York, 2006.

[40] S. Stramigioli and H. Bruyninckx, “Geometry and screw theory for robotics,” *Tutorial during ICRA*, vol. 2001, 2001.