A Center of Mass Observing 3D-LIPM Gait for the RoboCup Standard Platform League Humanoid

Colin Graf[1] and Thomas Röfer[2]

[1] Universität Bremen, Fachbereich 3 – Mathematik und Informatik,
Postfach 330 440, 28334 Bremen, Germany
cgraf@informatik.uni-bremen.de

[2] Deutsches Forschungszentrum für Künstliche Intelligenz,
Sichere Kognitive Systeme, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany
Thomas.Roefer@dfki.de

Abstract. In this paper, we present a walking approach for the Nao robot that improves the agility and stability of the robot when walking on a flat surface such as the soccer field used in the Standard Platform League. The gait uses the computationally inexpensive model of an inverted pendulum to generate a target trajectory for the center of mass of the robot. This trajectory is adapted using the observed real motion of the center of mass. This approach does not only allow compensating the inaccuracies in the model, but it also allows for reacting to external perturbations effectively. In addition, the method aims at facilitating a preferably fast walk while reducing the load on the joints.

1 Introduction

Since 2008, the humanoid robot *Nao* [4] that is manufactured by the French company Aldebaran Robotics is the robot used in the RoboCup Standard Platform League. The Nao has 21 degrees of freedom (see Fig. 4). It is equipped with a 500 MHz processor, two cameras, an inertial measuring unit, sonar sensors in its chest, and force-sensitive resistors under its feet. The camera takes 30 images per second while other sensor measurements are delivered at 100 Hz (50 Hz until 2009). The joints can be controlled at the same time resolution, i. e. walking means to generate 100 sets of 21 target joint angles per second.

Since the beginning of 2010, Aldebaran Robotics provides a gait for the Nao [4] that, although being a closed-loop walk, only takes the actual joint angles into account, not the measurements of the inertial measurement unit in Nao's chest. Thus the maximum speed reachable with the walk provided is still severely limited. As delivered by the manufacturer, it is approximately 10 cm/s. For RoboCup 2008, Kulk and Welsh designed an open-loop walk that keeps the stiffness of the joints as low as possible to both conserve energy and to increase the stability of the walk [9]. The gait reached 14 cm/s although it was based on the previous walking module provided by Aldebaran Robotics. Two groups worked on walks that keep the Zero Moment Point (ZMP) [13] above the support area using preview controllers. Both implement real omni-directional gaits. Czarnetzki *et al.* [1] reached speeds up to 20 cm/s with their approach. In their paper, this was only done in simulation. However, at RoboCup 2009 their robots reached similar speeds on the actual field, but they seemed to be hard to control and there was a certain lack in robustness, i. e., the robot fell down quite often. Strom *et. al* [12] modeled the robot as an inverted pendulum in their ZMP-based method. They reached speeds of around 10 cm/s.

In [6], [11], and [5], we already presented a robust closed-loop gait for the Nao. The active balancing used in the approach is based on the pose of the torso of the robot. In addition, we also presented an analytical solution to the inverse kinematics of the Nao, solving the problems introduced by the special hip joint of the Nao, i. e. dealing with the constraint that both legs share a degree of freedom in the hip. The gait presented in this paper is a continuation of this work.

The main contribution of this paper is presenting a computational inexpensive way for using the inverted pendulum model with dynamic phase duration for modeling a fast but omnidirectional and responsive walk and to use the same model to react to perturbations. Hence in addition to previous works, the focus is placed on using sensor feedback to observe the state of the robot and to adjust the inverted pendulum model to the observed state in order to improve the stability of the walk. The resulting walk is one of the fastest omnidirectional walks implemented on the Nao so far.

The structure of this paper is as follows: in the next section, modeling the walking robot as an inverted pendulum to control position and speed of its center of mass is discussed. In Section 3, the integration of sensor feedback is presented. Section 4 discusses the results achieved, followed by Section 5, which concludes the paper and gives an outlook on future work.

2 Using the Inverted Pendulum to Create Walking Motions

Generating a walking motion for humanoid robot basically means to create a sequence of joint angle sets, where each joint angle set will be executed successively. To be able to create a single set and a series of joint angles, a method is required to represent the state and the change of the state of the robot while it is walking. The approach presented in this paper reduces the model of the robot to its center of mass and uses the position and desired velocity of the center of mass to describe the state of the robot. The change of the state is described by determining a trajectory for the movement of the center of mass. From the position of the center of mass, the actual joint angles are determined by generating an additional trajectory for the position of the nonsupporting foot and by applying inverse kinematics to both legs (details are given in [5]).

To describe the movement of the center of mass, the 3-Dimensional Linear Inverted Pendulum Mode (3D-LIPM) [7] is used, which provides an approximation of a physically respectable model for the motion of the center of mass. In addition, changes in rotation, as they occur while rotating on the spot or while walking along a curve, are ignored. Hence, the position and velocity of the center of mass on a plane of height h in parallel to the ground relative to the origin of the inverted pendulum (see Fig. 1) are given by

$$x(t) = x_0 \cdot \cosh(k \cdot t) + \dot{x}_0 \cdot \frac{1}{k} \cdot \sinh(k \cdot t) \tag{1}$$

$$\dot{x}(t) = x_0 \cdot k \cdot \sinh(k \cdot t) + \dot{x}_0 \cdot \cosh(k \cdot t) \tag{2}$$

where $k = \sqrt{\frac{g}{h}}$, g is the gravitational acceleration ($\approx 9.81 \frac{\text{m}}{\text{s}^2}$), $x_0 \in \mathbb{R}^2$ is the position of the center of mass relative to the origin of the inverted pendulum at $t = 0$, and $\dot{x}_0 \in \mathbb{R}^2$ is the velocity of the center of mass at $t = 0$. A point under the currently supporting foot is used as origin of the inverted pendulum (see Fig. 6).

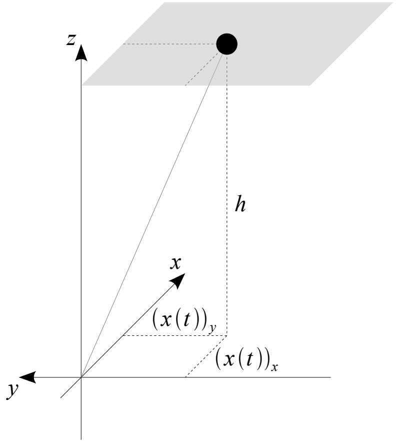

Fig. 1. An inverted pendulum in the three-dimensional space with fixed height h.

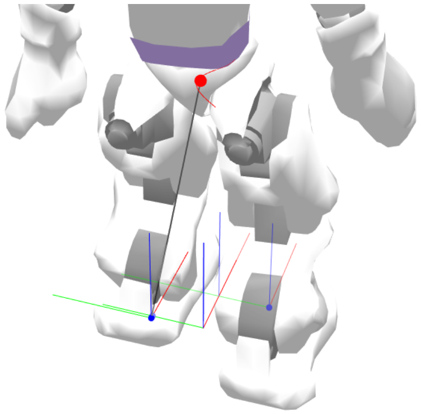

Fig. 2. An inverted pendulum attached to an obliquely forwards walking simulated model of the Nao.

In a single support phase, the inverted pendulum defines the motion of the center of mass according to its position and velocity relative to the origin of the inverted pendulum. Hence at the beginning of a single support phase, the position and velocity of the center of mass should be in a state that leads to the proper position and velocity for the next single support phase (of the other leg). The origin of the inverted pendulum should thereby be placed as close as possible to an optimal position under the foot (see Fig. 6). Since the step sizes to be performed can be chosen without severe constraints, the movement of the center of mass has to be adjusted for each step so the origins of the inverted pendulums used fit to the feet positions that are defined by the step sizes. Most walking approaches applied on the Nao [12, 3, 1] use a short double support phase for accelerating or decelerating the center of mass to achieve such an adjustment. To maximize the possible range that can be covered within a phase, the single support phase should make up as much as possible of the whole step phase to reduce the accelerations that are necessary for shifting the foot. Hence, the approach presented in this paper aims on eliminating the need of a double support phase, while keeping the origins of the inverted pendulums close to their optimal positions.

Even though no double support phase is used, a method to manipulate the movement of the center of mass is required. Therefore, the point in time for altering the support leg is used to control the velocity of the center of mass in the y-direction (see Fig. 3 for the system of coordinates used). To control the velocity in the x-direction, the origin of the inverted pendulum is shifted along the x-axis towards the elongated shape of the feet (see Fig. 6). This way the velocity of the center of mass can be manipulated enough to cover a specific distance (step size) while swinging from one leg to the other.

2.1 The System of Coordinates

Walking is a sequence of single support phases. In this paper, the symbols used to describe the current single support phase have no extra markings, while the symbols used to describe the following single support phase are dashed (e. g. x vs. $\bar{x}$). For each new single

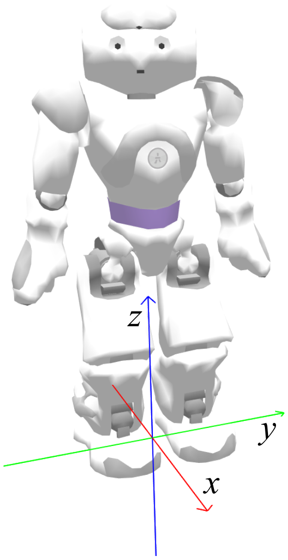

Fig. 3. The coordinate system Q used in this paper for describing the center of mass and the inverted pendulum origin positions.

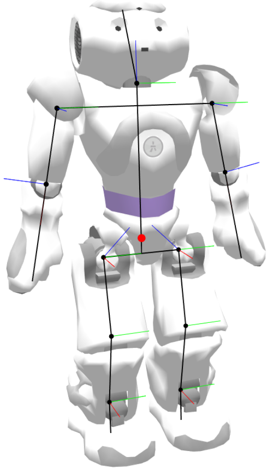

Fig. 4. Schematic view of the kinematic chain of the Nao. Each black dot marks a joint with up to 3 degrees of freedom. The bigger, red dot marks the center of mass position.

support phase, a new coordinate system Q (see Fig. 3) is used to describe the set points of the center of mass and the feet positions. The origin of the inverted pendulum has the distance $r \in \mathbb{R}^2$ on the x-y-plane from the origin of $\bar{Q}$. This distance remains (almost) constant within a single support phase. The origin of Q is located between both feet so that a step size $\bar{s}$ describes the offset from the origin of Q to the origin of $\bar{Q}$ (see Fig. 5 and Fig. 6) where $\bar{Q}$ is the coordinate system Q of the upcoming single support phase. If the robot walks in place, the step size is 0 and Q is the same as $\bar{Q}$.

2.2 Computing Step Durations

To apply the functions (1) and (2) for generating walking motions, a definition of the point in time $t = 0$ is required to determine when to alter the support leg. $t = 0$ is defined as the inflection point of the pendulum motion where the y-component of the velocity is 0 ($(\dot{x}_0)_y = 0$). The position of the center of mass at this point $(x_0)_y$ is an arbitrary parameter and has a value of greater or lower than 0 depending on the active support leg. Since $(\dot{x}_0)_y = 0$, the function

$$x_y(t) = (x_0)_y \cdot \cosh(k \cdot t) \tag{3}$$

in the range $t \geq t_b$ and $t \leq t_e$ can be used to compute the y-component of the center of mass position relative to the origin of the inverted pendulum. A single support phase starts at $t = t_b$ ($t_b < 0$) and ends at $t = t_e$ ($t_e > 0$).

If the nonsupporting foot should be placed with a distance of $\bar{r}_y + \bar{s}_y - r_y$ to the supporting foot at the end of the single support phase (see Fig. 7) and if $\bar{x}(\bar{t})$ and $\dot{\bar{x}}(\bar{t})$

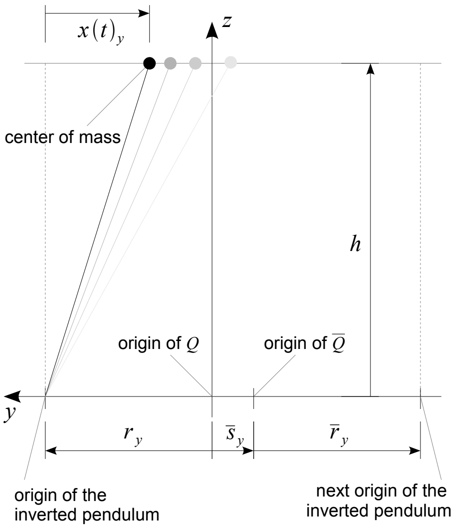

Fig. 5. y-z-cross section of the coordinate system used for the altering inverted pendulums.

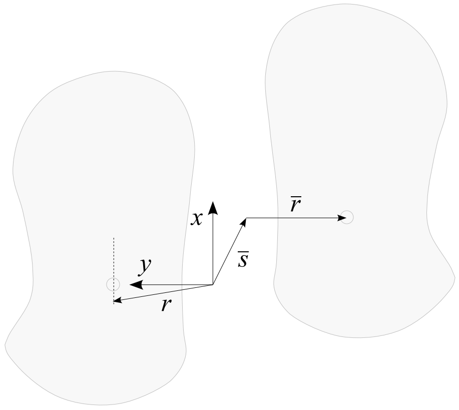

Fig. 6. x-y-cross section showing the step size $\bar{s}$ and the inverted pendulum origins r and $\bar{r}$. The small gray circle marks the foot position that is also referred as optimal inverted pendulum origin. The dotted line marks allowed inverted pendulum origins.

are position and velocity of the center of mass relative to the next pendulum origin, the point in time to alter the support leg can be determined by finding the ending of a single support phase t_e where:

$$(x(t_e))_y - (\bar{x}(\bar{t}_b))_y = \bar{r}_y + \bar{s}_y - r_y \tag{4}$$

$$(\dot{x}(t_e))_y = (\dot{\bar{x}}(\bar{t}_b))_y \tag{5}$$

The ending of a single support phase t_e and a matching beginning of the next single support phase $\bar{t}_b$ cannot be found by simply solving equation (4) and (5) for t_e and $\bar{t}_b$. This is not possible since it cannot be assumed that the functions $(x(t))_y$ and $(\bar{x}(\bar{t}))_y$ are symmetric. To handle this problem an iterative method (see Algorithm 1) is used, in which t_e is initially guessed. The equation (5) can be transformed into

$$\bar{t}_b = \frac{1}{\bar{k}} \cdot \operatorname{arcsinh}\left(\frac{(x_0)_y \cdot k \cdot \sinh(k \cdot t_e)}{(\bar{x}_0)_y \cdot \bar{k}}\right) \tag{6}$$

to compute a value for $\bar{t}_b$ that matches to the guessed t_e. The guessed t_e can then be refined using the velocity of the center of mass at t_e and the length $(x(t_e))_y - (\bar{x}(\bar{t}_b))_y$.

2.3 Walking Forwards and Backwards

Up to now, the length of a single support phase and the y-component of the center of mass position at every point in time can be determined. To cover a step size $\bar{s}_x$, the origin of the inverted pendulum of the next single support phase should be placed in a distance of $\bar{r}_x - \bar{s}_x - r_x$ from the origin of the current inverted pendulum. In addition, the velocities of the center of mass relative to each pendulum origin should be equal at the point in

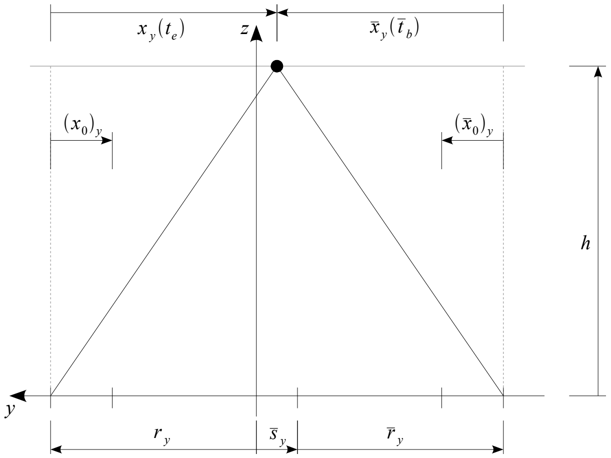

Fig. 7. Two facing inverted pendulum pendulums used to cover step size $\bar{s}_y$.

Algorithm 1 Computing t_e and $\bar{t}_b$

```
$t_e \leftarrow$ initial guess
repeat
    $\bar{t}_b \leftarrow \frac{1}{\bar{k}} \cdot \operatorname{arsinh}\left(\frac{(x_0)_y \cdot k \cdot \sinh(k \cdot t_e)}{(\bar{x}_0)_y \cdot \bar{k}}\right)$
    $y \leftarrow (x_0)_y \cdot \cosh(k \cdot t_e) - (\bar{x}_0)_y \cdot \cosh(\bar{k} \cdot \bar{t}_b)$
    $\dot{x}_{t_e} \leftarrow k \cdot (x_0)_y \cdot \sinh(k \cdot t_e)$
    $\Delta t_e \leftarrow \frac{\bar{r}_y + \bar{s}_y - r_y - y}{2 \cdot \dot{x}_{t_e}}$
    $t_e \leftarrow t_e + \Delta t_e$
until $|\Delta t_e| \leq$ desired precision (e. g. 0.0001 s)
$\bar{t}_b \leftarrow \frac{1}{\bar{k}} \cdot \operatorname{arsinh}\left(\frac{(x_0)_y \cdot k \cdot \sinh(k \cdot t_e)}{(\bar{x}_0)_y \cdot \bar{k}}\right)$
```

time when the support leg alternates. So, analogically to the equations (4) and (5) for the y-direction, the following equations should apply in the x-directions:

$$(x(t_e))_x - (\bar{x}(\bar{t}_b))_x = \bar{r}_x + \bar{s}_x - r_x \tag{7}$$

$$(\dot{x}(t_e))_x = (\dot{\bar{x}}(\bar{t}_b))_x \tag{8}$$

Two properties are planned ahead for the next single support phase. On the one hand, the position of the origin of the inverted pendulum should be optimal, so that $\bar{r}_x = 0$. On the other hand, the center of mass should be exactly over the next pendulum origin $(\bar{x}(0))_x = 0$ at $t = 0$. The latter is substantiated on simple forward walking ($\bar{s}_x = 0$) where $-t_b = t_e$, so that $(\bar{x}(0))_x = 0$ guarantees an evenly distributed center of mass motion that allows using optimal inverted pendulum origins (r_x and $\bar{r}_x = 0$) when walking with a constant step size. In order to cover a distance of $\bar{s}_x$, the origin of the inverted pendulum r_x of the current single support phase is chosen in a way that equation (7) applies.

When the center of mass has the position x_{t_b} and velocity $\dot{x}_{t_b}$ relative to the origin of Q at the beginning of a single support phase, r_x can be computed by using the following linear system of equations:

$$\begin{array}{llllll} r + x_0 \cdot \cosh(k \cdot t) & + \dot{x}_0 \cdot \frac{\sinh(k \cdot t)}{k} & & & & = x_t \\ \quad x_0 \cdot k \cdot \sinh(k \cdot t) & + \dot{x}_0 \cdot \cosh(k \cdot t) & & & & = \dot{x}_t \\ \quad x_0 \cdot k \cdot \sinh(k \cdot t_e) & + \dot{x}_0 \cdot \cosh(k \cdot t_e) & - \bar{x}_0 \cdot \bar{k} \cdot \sinh(\bar{k} \cdot \bar{t}_b) & - \dot{\bar{x}}_0 \cdot \cosh(\bar{k} \cdot \bar{t}_b) & & = [0,0]^T \\ r + x_0 \cdot \cosh(k \cdot t_e) & + \dot{x}_0 \cdot \frac{\sinh(k \cdot t_e)}{k} & - \bar{x}_0 \cdot \cosh(\bar{k} \cdot \bar{t}_b) & - \dot{\bar{x}}_0 \cdot \frac{\sinh(\bar{k} \cdot \bar{t}_b)}{\bar{k}} & - \bar{s} & = \bar{r} \end{array} \tag{9}$$

It is not only possible to compute r_x using the linear system of equations (9), but also to compute $(x_0)_x$ and $(\dot{x}_0)_x$, so that a complete set of pendulum parameters (r, x_0 and $\dot{x}_0$) can be determined.

Depending on the desired step size $\bar{s}$, position x_{t_b}, and velocity $\dot{x}_{t_b}$ of the center of mass at the beginning of the single support phase, the absolute value of r_x can reach values that would shift the pendulum origin out of the convex hull of the foot or out of a range that can be considered to ensure a stable walk. Hence, a computed r_x can be limited and an alternative value for $\bar{s}_x$ can be computed using the linear system of equations (9) as well. This allows making sure that only step sizes are used that result in inverted pendulum origins close enough to the optimal positions. $|r_x|$ can be capped to only a few millimeters (e. g. 4 mm), to reduce the possible acceleration and deceleration of walking speeds and to improve the stability of the walk.

3 Balancing

The results of a walk generated solely using the center of mass trajectory as described in Section 2 are not convincing. It might be possible to find parameters that keep the robot upright and that allow slow locomotion, but it is obvious that the model alone is not suitable to keep the walk permanently stable. Furthermore, the robot is not capable to react on perturbations of any kind.

Without balancing the trajectory of the center of mass is static and can only be executed as it was computed before. In the case that an external perturbation affects the robot or when the model used is simply not precise enough to represent the dynamics of the robot, the motion of the center of mass does not follow the trajectory as intended and the robot may fall in any direction. There are several approaches to handle this problem. As soon as a deviation in the motion of the center of mass is detected, a counteraction can be executed to bring the center of mass back to the desired position. Another approach is to compute a slightly modified trajectory for the center of mass to continue the deviated motion according to the inverted pendulum model and to adjust future steps to the deviated motion. The walk presented in this paper uses the second approach. Using the first approach, it is hardly possible to compensate a perturbation completely, but it is quite useful to prevent the error from increasing with further movement of the center of mass. The advantage of the second approach is that it acts in a more farsighted manner. If an error is in the system, it is absorbed to continue the intended step as stable as possible. But at first, both approaches require observing the actual position of the center of mass to detect an error.

3.1 Observing the Center of Mass

For determining an observed position of the center of mass, a four-dimensional (simple) Kalman filter [8] is used for the x and y-components of the position and the velocity of the center of mass. It is actually implemented using two independent two-dimensional Kalman filters. Estimating the z-component is left out, since the 3D-LIPM used can not handle any dynamic pendulum heights. Instead of computing the sensor readings for a predicted center of mass position, the error Δx_i between the expected center of mass position x_{ei} relative to Q and the measured center of mass position x_{mi} at time t_i are computed and used as innovation. The expected center of mass position is computed using $x(t)$ with the inverted pendulum parameters of the current single support phase. The measured center of mass position x_{mi} is basically computed by using an estimated orientation of the robot torso and the kinematic chain to the supporting foot that is constructed using the joint angle sensor readings of the Nao.

$$x_{ei} = r + x(t_i) \tag{10}$$

$$\Delta x_i = x_{mi} - x_{ei} \tag{11}$$

The Kalman filter uses x_{ei} and the function $\dot{x}(t_i)$ with the inverted pendulum parameters of the current single support phase as the predicted state ${\mu_i}'$. The covariance ${\Sigma_i}'$ of the predicted state is computed as in an ordinary four-dimensional Kalman filter that estimates a position and a velocity of an object in two-dimensional space with a process noise $\Sigma_{\varepsilon i}$.

$${\mu_i}' = \begin{bmatrix} (x_{ei})_x \\ (x_{ei})_y \\ (\dot{x}(t_i))_x \\ (\dot{x}(t_i))_y \end{bmatrix}, {\Sigma_i}' = A \cdot \Sigma_{i-1} \cdot A^T + \Sigma_{\varepsilon i} \quad \text{with } A = \begin{bmatrix} 1 & 0 & \Delta t_i & 0 \\ 0 & 1 & 0 & \Delta t_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \Delta t_i = t_i - t_{i-1} \tag{12}$$

The Kalman gain K_i for the innovation Δx_i can be computed assuming covariance Σ_{mi} for the measurement x_{mi}.

$$K_i = {\Sigma_i}' \cdot C^T \cdot (C \cdot {\Sigma_i}' \cdot C^T + \Sigma_{mi})^{-1} \quad \text{with } C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{13}$$

And finally, the filtered position x_{f_i}, the filtered velocity $\dot{x}_{f_i}$, and the updated covariance Σ_i can then be computed.

$$\mu_i = \begin{bmatrix} \left(x_{f_i}\right)_x \\ \left(x_{f_i}\right)_y \\ \left(\dot{x}_{f_i}\right)_x \\ \left(\dot{x}_{f_i}\right)_y \end{bmatrix} = {\mu_i}' + K_i \cdot \Delta x_i, \Sigma_i = {\Sigma_i}' - K_i \cdot C \cdot {\Sigma_i}' \tag{14}$$

The parameters of the Kalman filters, i. e. the assumed process noise $\Sigma_{\varepsilon i}$ and the deviation of the computed error Σ_{mi}, can control how much sensor feedback is used to correct the pendulum parameters (see Section 3.2). Using a small process noise and a large deviation of the computed error results in only little corrections according to the measured position of the center of mass.

3.2 Correcting the Inverted Pendulum Parameters

The filtered position x_{f_i} and the filtered velocity $\dot{x}_{f_i}$ of the Kalman filter that estimates the true position and velocity of the center of mass are used to re-determine the parameters of the inverted pendulum of the current single support phase. Since the prediction of the natural motion of the center of mass is based on the pendulum parameters, the main purpose of the correction is to improve the prediction in the next iteration. So the pendulum parameters r, x_0, $\dot{x}_0$, and t_i are re-determined to find a pendulum function that fits to the estimated position and the estimated velocity of the center of mass:

$$r' + x'({t_i}') = x_{f\,i} \tag{15}$$

$$\dot{x}'({t_i}') = \dot{x}_{f\,i} \tag{16}$$

The y-component ${(x_0)_y}'$ of the center of mass at the point in time $t = 0$ as well as the corrected current time ${t_i}'$ can be computed using the estimated position and the estimated velocity. ${(x_0)_y}'$ is calculated by using equation (3) and its derivation:

$$(x_f)_y - {r_y}' = {(x_0)_y}' \cdot \cosh(k' \cdot {t_i}') \tag{17}$$

$$(\dot{x}_f)_y = {(x_0)_y}' \cdot k' \cdot \sinh(k' \cdot {t_i}') \tag{18}$$

Equation (17) can be solved for $k' \cdot {t_i}'$ and inserted into (18) to solve the resulting equation for ${(x_0)_y}'$:

$${(x_0)_y}' = \sqrt{((x_{f\,i})_y - {r_y}')^2 - \frac{{(\dot{x}_{f\,i})_y}^2}{k'^2}} \tag{19}$$

${t_i}'$ can then be computed by solving equation (18) for ${t_i}'$:

$${t_i}' = \frac{1}{k'} \cdot \operatorname{arcsinh}\left(\frac{(\dot{x}_f)_y}{k' \cdot (x_0)_y}\right) \tag{20}$$

Given the corrected y-component of the pendulum position at $t' = 0$ (which is possibly shifted to $t = 0$) a corrected point in time ${t_e}'$ for the ending of the current single support phase can be computed using the iterative method as described in Section 2.2 with the current step size $\bar{s}$. The x-components of the estimated position $x_{f\,i}$ and the estimated velocity $\dot{x}_{f\,i}$ can be used for computing the corrected pendulum origin ${r_x}'$ and the other pendulum parameters ${(x_0)_x}'$ and ${(\dot{x}_0)_x}'$ by using the linear system of equations (9).

3.3 Controlling the Predicted Positions of the Center of Mass

A general problem with balancing a walk is that the sensor readings are not in sync with the controlled joint angles. On the Nao, 40 ms or 4 cycles in the walk generation process elapse until a reaction from a joint angle request becomes recognizable. Hence, a pendulum motion in sync with the measurements is considered at first. At this frame, the estimated position and the estimated velocity of the center of mass are used to compute the corrected pendulum parameters. To control a position of the center of mass x_{p_i}, the position of the center of mass 40 ms in the future is predicted according the corrected pendulum parameters. If the 40 ms exceed the ending of the current single support phase, the pendulum parameters of the next single support phase are used instead:

$$x_{p_i} = \begin{cases} r' + x'({t_i}' + 40\text{ms}) & \text{if } {t_i}' + 40\text{ms} < {t_e}' \\ \bar{s} + \bar{r} + \bar{x}(\bar{t}_b' + {t_i}' + 40\text{ms} - {t_e}') & \text{otherwise} \end{cases} \tag{21}$$

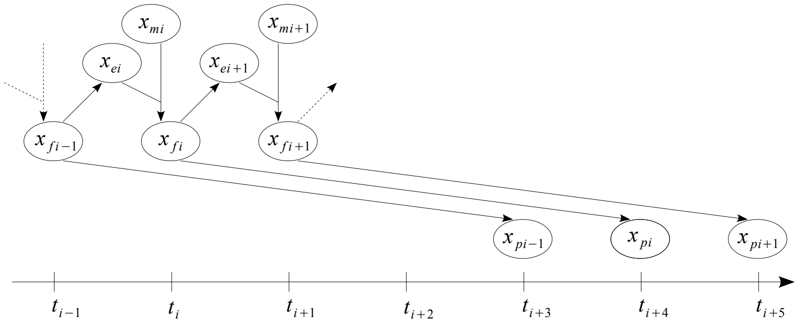

Fig. 8. The progressing filtering process of the position of the center of mass. The position of the center of mass x_{f_i} is used to look 10 ms ahead to compute a predicted position $x_{e\,i+1}$ that is then used together with the measured position $x_{m\,i+1}$ to compute the filtered position $x_{f\,i+1}$. The controlled position of the center of mass $x_{p_{i+1}}$ is then computed based on the filtered position $x_{f\,i+1}$.

4 Results

To find parameters for the gait, the robot was walking in a stationary position and some initial parameters were slightly changed to minimize the average error between expected and measured center of mass position and to result in a preferably constant and smooth motion from side to side (see Fig. 9). Some parameters define the stance and the motion with values such as the height of the center of mass above the ground, the offset between both feet (normally 10 cm because of the robot's leg design), the step height, and the magnitude of a step-bound rotation around the x-axis to simplify the center of mass shifting to a side. Furthermore, there are parameters that define the motion of the inverted pendulum such as the pendulum width $(x_0)_y$, the position of the optimal pendulum origin within a foot r_y, and the assumed height of the pendulum h, which can be chosen independently from the actual position of the center of mass to adjust the inverted pendulum model to the actual dynamics of the robot. The parameters can be chosen by hand or with an automatic method such as the Particle Swarm Optimization [2] that uses the averaged error between the expected and the measured position of the center of mass to rate the quality of a set of parameters.

The walking parameters used in the work presented aim at compromising between a maximum walking speed and a minimum load on joints. Hence, the center of mass is located quite high at 262 mm over ground, so that thigh and lower leg stand with an obtuse angle to each other to reduce the load on the knee joint. This can be a huge advantage when the Nao should stand or walk for a long period of time, since the knee joint can overheat quickly at a sharper angle due to the higher load (e. g. in less than 20 minutes, i. e. the duration of a game). Another advantage of the high center of mass is that—compared to a lowered stance—smaller changes in the joint angles are necessary to reach target foot positions that are far away. But a drawback of the high center of mass is that the maximum reachable target foot distance is more limited by the maximum length of a leg.

Using the best performing parameters found so far, the robot reaches 31 cm/s forwards with a step size of 7 cm, 12 cm/s sideways with a step size of 8 cm, 22 cm/s backwards,

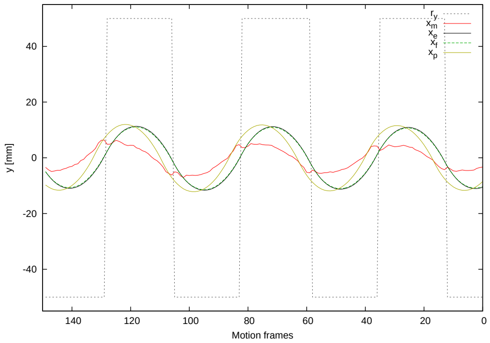

Fig. 9. A plot showing the y-component of the measured (red) and expected (black) center of mass position while the robot was walking on the spot. The motion of the robot is steady although the average error appears to be immense (4.26 mm).

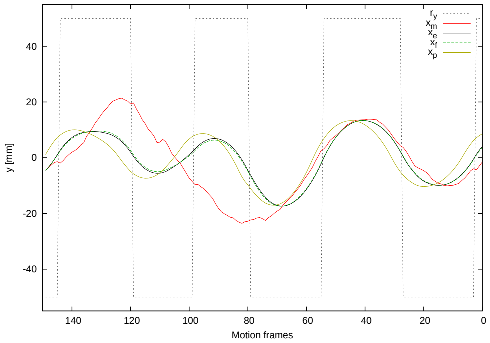

Fig. 10. A plot showing the y-component of the measured (red) and expected (black) center of mass position when the robot was pushed from the side. The expected center of mass position chases the measured position and the walk stabilizes quickly.

and 92°/s when rotating on the spot. At full speed, the average error of the position of the center of mass in the y-direction is 7.76 mm. The error is smaller with reduced walking speed (e. g. 5.88 mm at 10 cm/s forwards). With reduced walking speed, the gait is also substantially more robust against perturbations. The correction of the inverted pendulum parameters causes the motion of the center of mass to adjust quickly to the expected trajectory (see Fig. 10).

5 Conclusions and Future Work

In this paper, we present a robust closed-loop gait for the Nao robot. The gait uses the center of mass as simplified representation of the walking robot and a model for the movement of this center of mass that is based on two alternating inverted pendulums. The model allows eliminating the need of a double-support phase by dynamically adjusting the point in time at which the support leg alternates. Thus, the load on the joints for bridging over larger distances can be reduced.

In addition, we have suggested a method for estimating the actual position and the actual velocity of the center of mass. We have explained how the estimate can be used for correcting the inverted pendulum model. The correction does not only allow using the inverted pendulum model on real hardware without perfect joint calibration, but it also adds robustness against perturbations such as forces exerted on the robot. The maximum speed achieved could be increased in comparison to the results of previous works.

Work that was already started for RoboCup 2010 and is still ongoing is to add a mechanism to learn the difference between the inverted pendulum model and the real dynamics of the robot. Furthermore, we plan to extend the walking engine to a general motion engine that, e. g., will also integrate dynamic kicks as presented by [10]. This will significantly improve the robot's ability to dribble the ball and speed up the transitions between walking and kicking.

Acknowledgment

The authors would like to thank all B-Human team members for providing the software base for this work.

References

1. Czarnetzki, S., Kerner, S., Urbann, O.: Observer-based dynamic walking control for biped robots. Robotics and Autonomous Systems 57(8), 839–845 (2009)
2. Eberhart, R.C., Kennedy, J.: A new optimizer using particles swarm theory. In: Sixth International Symposium on Micro Machine and Human Science. pp. 39–43 (1995)
3. Gouaillier, D., Collette, C., Kilner, C.: Omni-directional closed-loop walk for NAO. In: Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on. pp. 448–454 (2010)
4. Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., Maisonnier, B.: The NAO humanoid: a combination of performance and affordability. CoRR abs/0807.3223 (2008)
5. Graf, C., Röfer, T.: A closed-loop 3D-LIPM gait for the RoboCup Standard Platform League humanoid. In: Zhou, C., Pagello, E., Behnke, S., Menegatti, E., Röfer, T., Stone, P. (eds.) Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the 2010 IEEE-RAS International Conference on Humanoid Robots (2010)
6. Graf, C., Härtl, A., Röfer, T., Laue, T.: A Robust Closed-Loop Gait for the Standard Platform League Humanoid. In: Zhou, C., Pagello, E., Menegatti, E., Behnke, S., Röfer, T. (eds.) Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the 2009 IEEE-RAS International Conference on Humanoid Robots. pp. 30–37. Paris, France (2009)
7. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K., Hirukawa, H.: A realtime pattern generator for biped walking. In: Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA 2002). pp. 31–37. Washington, D.C., USA (2002)
8. Kalman, R.E.: A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering 82(Series D), 35–45 (1960)
9. Kulk, J.A., Welsh, J.S.: A low power walk for the NAO robot. In: Kim, J., Mahony, R. (eds.) Proceedings of the 2008 Australasian Conference on Robotics & Automation (ACRA-2008) (2008)
10. Müller, J., Laue, T., Röfer, T.: Kicking a ball – modeling complex dynamic motions for humanoid robots. In: Chown, E., Matsumoto, A., Plöger, P., del Solar, J.R. (eds.) RoboCup 2010: Robot Soccer World Cup XIV. pp. 109–120. No. 6556 in Lecture Notes in Artificial Intelligence, Springer (2011)
11. Röfer, T., Laue, T., Müller, J., Burchardt, A., Damrose, E., Fabisch, A., Feldpausch, F., Gillmann, K., Graf, C., de Haas, T.J., Härtl, A., Honsel, D., Kastner, P., Kastner, T., Markowsky, B., Mester, M., Peter, J., Riemann, O.J.L., Ring, M., Sauerland, W., Schreck, A., Sieverdingbeck, I., Wenk, F., Worch, J.H.: B-Human team report and code release 2010 (2010), only available online: `http://www.b-human.de/file_download/33/bhuman10_coderelease.pdf`
12. Strom, J., Slavov, G., Chown, E.: Omnidirectional walking using ZMP and preview control for the NAO humanoid robot. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009: Robot Soccer World Cup XIII. pp. 378–389. No. 5949 in Lecture Notes in Artificial Intelligence, Springer (2010)
13. Vukobratovic, M., Borovac, B.: Zero-moment point – thirty five years of its life. International Journal of Humanoid Robotics 1(1), 157–173 (2004)