A Closed-loop 3D-LIPM Gait for the RoboCup Standard Platform League Humanoid

Colin Graf #1, Thomas Röfer *2

#Department for Mathematics and Informatics, Universität Bremen
Postfach 330440, 28334 Bremen, Germany
1cgraf@informatik.uni-bremen.de

*Safe and Secure Cognitive Systems, German Research Center for Artificial Intelligence (DFKI)
Enrique-Schmidt-Str. 5, 28359 Bremen, Germany
2Thomas.Roefer@dfki.de

Abstract— **The generation of a robust and fast gait for the humanoid robot Nao is still one of the major research topics of the Standard Platform League. In this paper, we present a closed-loop gait, as it was used at the RoboCup competition 2010. We will explain how the inverted pendulum model is used to create trajectories for the center of mass of an omnidirectional gait and how it allows to eliminate the need of the so-called double-support phase by dynamically adjusting the point in time at which the support foot alternates. Furthermore, we will briefly describe our approach for integrating sensor feedback to this model and how to transform the target trajectory of the center of mass into positions for the foot placement.**

I. Introduction

Since 2008, the humanoid robot *Nao* [1] that is manufactured by the French company Aldebaran Robotics is the robot used in the RoboCup Standard Platform League (cf. Fig. 1). The Nao has 21 degrees of freedom (cf. Fig. 2 left). 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.

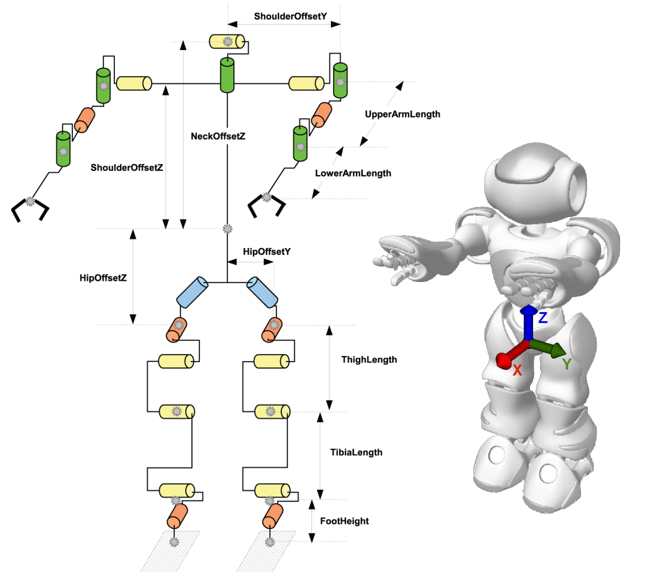

Fig. 2. The joints of the Nao [1] (left). The robot coordinate system used in this paper (right).

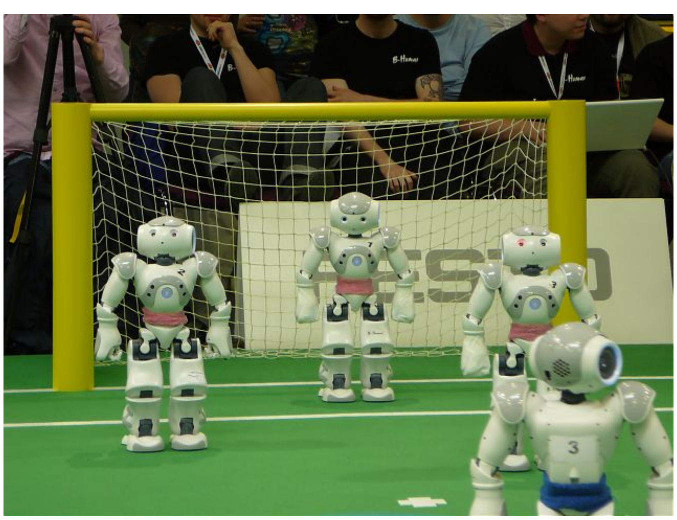

Fig. 1. Naos on a soccer field at the RoboCup German Open 2010.

Since the beginning of 2010, Aldebaran Robotics provides a gait for the Nao [1] 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. It is approximately 10 cm/s. For RoboCup 2008, Kulk and Welch 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 [2]. The gait reached 14 cm/s. However, since that walk was based on the previous walking module provided by Aldebaran Robotics, it shared the major drawback of not being omni-directional. Two groups worked on walks that keep the Zero Moment Point (ZMP) [3] above the support area using preview controllers. Both implement real omnidirectional gaits. Czarnetzki *et al.* [4] 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* [5] modeled the robot as an inverted pendulum in their ZMP-based method. They reached speeds of around 10 cm/s.

In [6] and [7], 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, which nearly doubles the speed achieved.

The main contribution of this paper is using the inverted pendulum with dynamic phase duration as a model for implementing omnidirectional walking and for balancing on the Nao humanoid robot. 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. It contains the core ideas of this paper. In Section III, the integration of sensor feedback is presented. Afterwards, Section IV describes, how the actual target joint angles are determined from the desired position of the center of mass. Section V briefly discusses the results achieved, followed by Section VI, which concludes the paper and gives an outlook on future work.

II. Using the Inverted Pendulum

Walking with a humanoid robot means to create a series of joint angles. The approach presented in this paper is a further development of the approach described in [6] and [7] with an improved method for controlling the center of mass motion and altered usage of sensor feedback. The trajectory of the center of mass is based on the 3-Dimensional Linear Inverted Pendulum Mode (3D-LIPM) [8]. Hence, the position and velocity of the center of mass relative to the origin of the inverted pendulum 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 ($\approx 9.81 \frac{\mathrm{m}}{\mathrm{s}^2}$), h is the height of the center of mass above the ground, x_0 is the position of the center of mass relative to the origin of the inverted pendulum at $t = 0$, and $\dot{x}_0$ is the velocity of the center of mass at $t = 0$ (cf. Fig. 3).

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

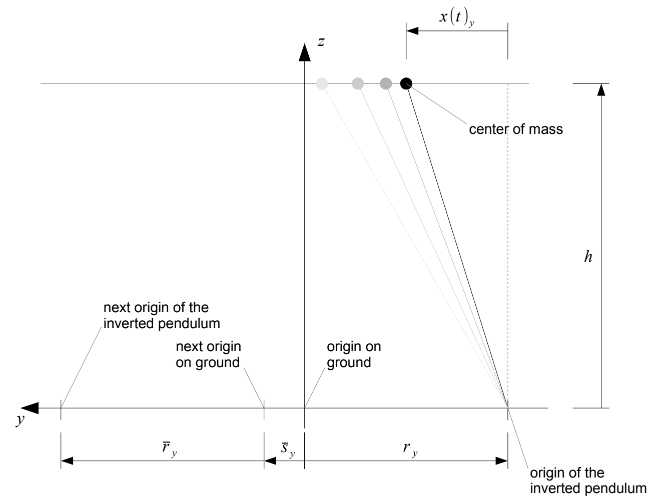

Fig. 3. Cross section of the coordinate system used for the altering inverted pendulums.

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 origins of the inverted pendulums should thereby be placed as close as possible under the center of the feet so that the positions of the origins are defined by the steps that should be performed. Since the steps that should be performed can be chosen without severe constraints, the movement of the center of mass has to be adjusted for every step to result into origins that match to the feet positions. Most walking approaches 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 passed 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 used aims on not using a double support phase at all, while keeping the origins of the inverted pendulums close to their optimal positions.

To proceed without a double support phase, a method is required to manipulate the movement of the center of mass. Therefore, the point in time for altering the support leg is used to control the velocity of the center of mass in y-direction (for the system of coordinates used cf. Fig. 2 right) as well as shifting the origin of the inverted pendulum along the x-axis towards the elongated shape of the feet. This way the velocity of the center of mass can be manipulated in x-direction as well as in y-direction which allows controlling these velocities to pass a specific distance (step size) while swinging from one leg to the other.

First of all, 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

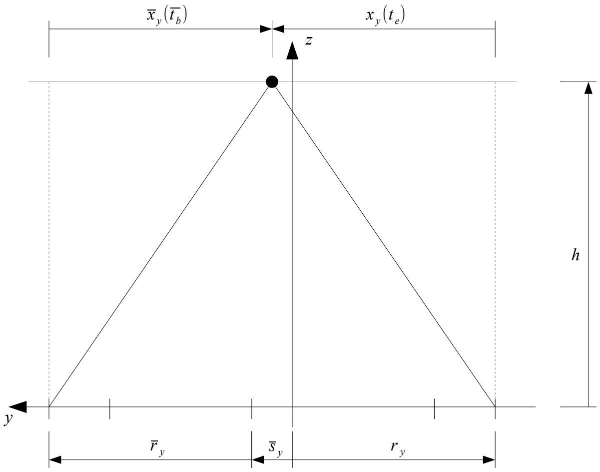

Fig. 4. Two facing inverted pendulums.

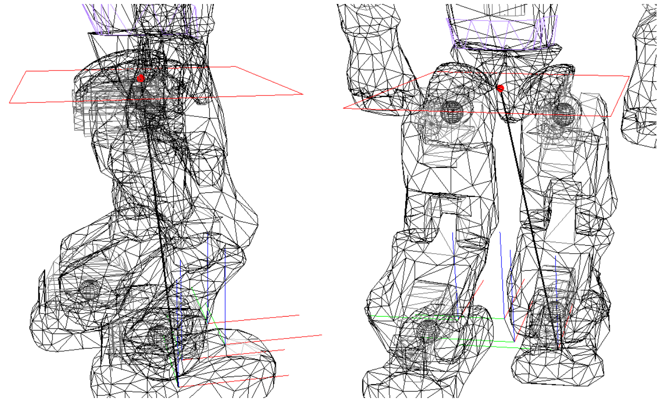

Fig. 5. The inverted pendulum attached on a simulated model of the Nao. The Nao is walking to front left. The red square marks the plain on which the center of mass moves, the red dot is the center of mass position and the black line on the ground is the step size. Also visible: the several origins as named in Fig. 3.

active support leg. This allows using

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

in the range between two points in time t_b and t_e as equation that provides 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 $t = t_e$ ($t_e > 0$).

Besides the origin of the inverted pendulum there are two more coordinate systems (origin on ground and next origin on ground) that are in parallel to the origin of the inverted pendulum with a constant distance r (cf. Fig. 4). These coordinate systems are placed in a way that the step size s is the distance from the origin on ground to the next origin on ground. If the robots walks in place, the step size is 0 and the origin on ground as well as the next origin on ground are directly between both feet.

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 (cf. Fig. 4) and if $\bar{x}(\bar{t})$ and $\dot{\bar{x}}(\bar{t})$ 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 and the beginning of the next single support phase $\bar{t_b}$ where:

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

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

Finding the ending of a single support phase t_e and the beginning of the next single support phase $\bar{t_b}$ is not that simple since it cannot be assumed that the functions $(x(t))_y$ and $(\bar{x}(\bar{t}))_y$ are symmetric and hence they cannot be solved for t_e and $\bar{t_b}$. To handle this problem an iterative method is used in which t_e is guessed at first. The equation (5) can be transformed into

$$\bar{t_b} = \frac{1}{k} \cdot \operatorname{asinh}\left(\frac{(x_0)_y \cdot \sinh(k \cdot t_e)}{(\bar{x_0})_y}\right) \tag{6}$$

to compute a value for $\bar{t_b}$ that matches to the guessed t_e. This allows calculating $(x(t_e))_y - (\bar{x}(\bar{t_b}))_y$ and to compare it with $\bar{r}_y + \bar{s}_y - r_y$. An improved guess t_e can then be determined with half of the difference between $(x(t_e))_y - (\bar{x}(\bar{t_b}))_y$ and $\bar{r}_y + \bar{s}_y - r_y$ and the velocity $(x(t_e))_y$. Only a few iterations are necessary (≈ 3) to get an estimate that is precise enough (± 0.1 ms) to work with. $(\bar{x_0})_y$ remains constant within a single support phase and $(x_0)_y$ changes decent (because of balancing) so that a value for t_e that was once determined can be reused to reduce the number of required iterations.

Algorithm 1 Computing t_e and $\bar{t_b}$

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

As function for the x-component of the center of mass position, a distorted pendulum motion of the following form is used:

$$x_x(t) = c_1 \cdot \cosh(k \cdot t) + c_2 \cdot \frac{1}{k} \cdot \sinh(k \cdot t) + c_3 \cdot t + c_4 \tag{7}$$

This function should meet two properties. First it should lead to a position and velocity that matches the position and velocity at the beginning of the single support phase (at t_b) and second it should be near to a straight pendulum function ($c_3 \approx 0$, $c_4 \approx 0$) if the current step size is similar to the step size of the previous single support phase. To achieve such properties it is defined that the inverted pendulum in x-direction should be directly over its origin at the point in time $t = 0$. Hence,

$(x(0))_x$ and $(\bar{x}(0))_x$ are equal 0 which results in $(x_0)_x = 0$ and $(\bar{x_0})_x = 0$. Given the points in time t_e and $\bar{t_b}$ already determined and the step size $\bar{s}_x$, the velocity for $t = 0$ can be calculated by solving the equations

$$(x(t_e))_x - (\bar{x}(\bar{t_b}))_x = \bar{r}_x + \bar{s}_x - r_x \tag{8}$$

$$(\dot{x}(t_e))_x = (\dot{\bar{x}}(\bar{t_b}))_x \tag{9}$$

for $(\dot{x}_0)_x$.

$$(\dot{x}_0)_x = \frac{k \cdot (\bar{s}_x + \bar{r}_x - r_x)}{\sinh(k \cdot t_e) - \cosh(k \cdot t_e) \cdot \tanh(k \cdot \bar{t_b})} \tag{10}$$

This allows calculating the optimal center of mass position and velocity at the point in time t_e as long as the step size remains constant.

Now, sufficient values are known for the function $x_x(t)$ and its derivative $\dot{x}_x(t)$ so that a linear system of equations can be used to determine c_1, c_2, c_3 and c_4.

III. Adding Sensor Feedback

Although it would be possible to walk slowly using this approach without sensor feedback, the usage of sensors is essential to reach faster walking speeds and to add robustness to the walk. On one hand, this allows reacting to unexpected disruptions such as unevenness of the ground or the forces exerted on the robot. On the other hand, it compensates the forces and hardware characteristics that were not considered within the model used. The sensor readings are used to observe the motion of the center of mass relative to the current pendulum origin. The errors between the desired and the observed center of mass positions are then used to adjust the controlled center of mass motion to reduce the error occurring.

The Nao has good joint angle sensors and a two-axes gyroscope sensor in its chest. As long as the Nao stands on a single leg in a dynamically stable way, the area of the supporting foot gets pressed complacently on the ground since the whole robot weighs on it. Hence, the kinematic chain can be used to calculate the rotation of the robot's thorax relative to the ground. The rotation is consolidated with the gyroscope sensor readings using an Unscented Kalman filter. Then, the rotation of the robot's thorax relative to the ground is used to calculate the position of the supporting foot relative to the center of mass, which is also calculated using the joint angle sensor readings. Now, the desired and observed center of mass positions can be compared. In each iteration of the gait generation process an arbitrary percentage of the error occurring is then added the current pendulum parameters (r, $\bar{s}$, and x_0) and is used to customize the current point in time t. Figure 6 shows the difference between using and not using sensor feedback.

IV. Generating Joint Angles

Since the desired position of the center of mass relative to the origin of the supporting leg's inverted pendulum is always known, the next step in gait generation is to determine the target foot positions relative to the center of mass. Without considering a rotation of the robot's thorax, which means that

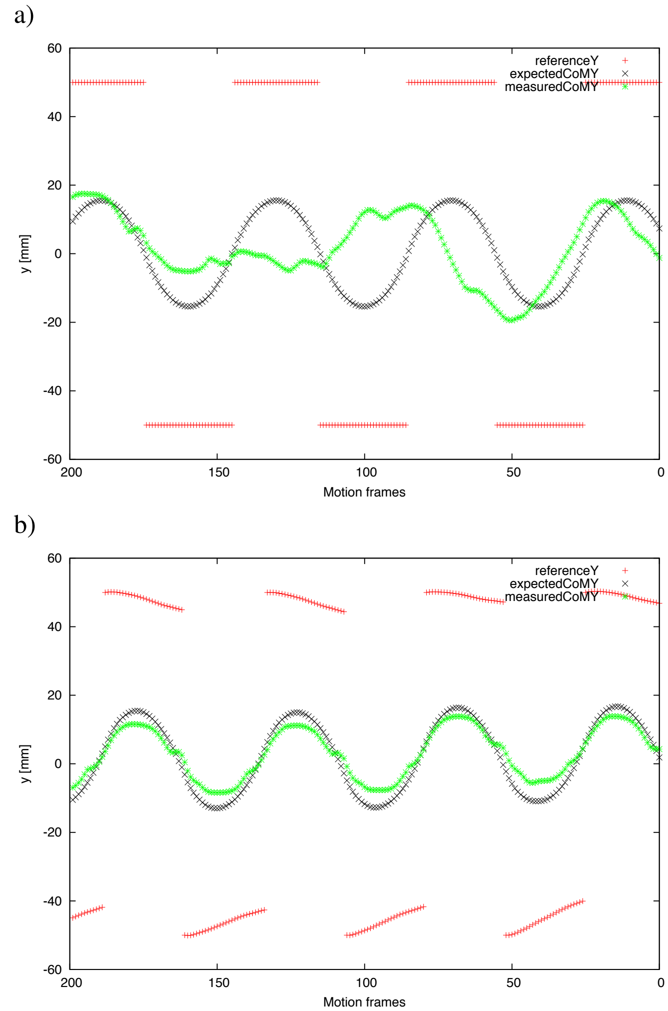

Fig. 6. The y-component of the pendulum origin (red) and the expected (black) and measured (green) center of mass positions. Both plots show a walk created with the same walking parameters where a) does not use sensor feedback.

it is always in parallel to the ground, the position of the supporting foot is given by the functions $x_x(t)$ and $x_y(t)$. However, the position of the other foot has to be determined.

Therefore, the known position and length of the current single support phase is used to get a percentage of the progressing phase.

$$\varphi(t) = \frac{t - t_b}{t_e - t_b} \tag{11}$$

At the beginning of the single support phase ($\varphi(t_b) = 0$), the target foot position of the nonsupporting foot should be equal to its position at the ending of the previous single support phase and at the ending of the single support phase ($\varphi(t_e) = 1$) it should be completely repositioned to the origin of the next inverted pendulum. In between these two stages the non supporting foot should be moved smoothly to its final position. For this purpose, the function

$$q(t) = \frac{1 - \cos(\pi \cdot \varphi(t))}{2} \tag{12}$$

is used.

The target position of the supporting foot $p_s(t)$ and the target position of the nonsupporting foot $p_a(t)$ are now specified as

$$p_s(t) = (-x_x(t), -x_y(t), -h) \tag{13}$$

$$p_a(t) = p_s(t) + (\bar{r} + \bar{s} - r) \cdot q(t) - (r + s - \breve{r}) \cdot (1 - q(t)) + l(t) \tag{14}$$

where $\breve{r}$ is r of the previous single support phase and where $l(t)$ is a function to lift the nonsupporting leg.

Finally, $p_s(t)$ and $p_a(t)$ have to be transformed into target positions relative to the robot's origin to be used as input for the inverse kinematics as described in [7]. This task is handled with another iterative method. A vector b is guessed at first by using the vector b of the previous iteration of the gait generation process. The change of the center of mass position in comparison to its previous position is then added to this vector and the inverse kinematics function is used to calculate temporary joint angles for calculating a temporary center of mass position. The difference between the temporary position and the desired position of the center mass can then be used to refine the vector b. The refining step is repeated only two times to give results with a precision of better than 0.1 mm. A scale factor for the difference between the temporary position and the desired position that was experimentally determined can improve the efficiency of this method.

V. Results

The gait described in this paper was used by the RoboCup team B-Human at the RoboCup 2010 in Singapore, and, in an earlier stage, also at the RoboCup German Open 2010 in Magdeburg. B-Human won both competitions. One of the main reasons for this success was superior walking speed. The YouTube channel at `http://www.youtube.com/user/TeamBhuman` contains recordings of most games, showing the gait in action.

The maximum forward speed achieved is 28 cm/s, which is nearly twice as fast as our gait used in RoboCup 2009. The maximum backward speed is 17 cm/s, the maximum sideways speed is 7 cm/s, and the maximum rotational speed is 90°/s.

The implementation of the approach described in this paper is part of B-Human's 2010 code release [9] and can be downloaded from `http://www.b-human.de/en/publications`.

VI. Conclusions and Future Work

In this paper, we present a robust closed-loop gait for the Nao, the humanoid robot used in the RoboCup Standard Platform League. The gait uses the center of mass as simplified representation of the robot and a sophisticated model for the center of mass movement that is based on two alternating inverted pendulums to create trajectories for each foot. 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. The stabilization methods, which are base on an estimated pose of the torso of the robot update the parameters and the state of the pendulum to react on external disturbances in a farsighted manner. As a result, the maximum speed achieved in comparison to the gait used in 2009 could be almost doubled, and the gait is less reliant on perfect joint calibration.

Work that was already started for RoboCup 2010 and is still ongoing is to extend the walking engine to a general motion engine that, e. g., also integrates dynamic kicks [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] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "The NAO humanoid: a combination of performance and affordability," *CoRR*, vol. abs/0807.3223, 2008.

[2] J. A. Kulk and J. S. Welsh, "A low power walk for the NAO robot," in *Proceedings of the 2008 Australasian Conference on Robotics & Automation (ACRA-2008)*, J. Kim and R. Mahony, Eds., 2008.

[3] M. Vukobratovic and B. Borovac, "Zero-moment point – thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

[4] S. Czarnetzki, S. Kerner, and O. Urbann, "Observer-based dynamic walking control for biped robots," *Robotics and Autonomous Systems*, vol. 57, no. 8, pp. 839–845, 2009.

[5] J. Strom, G. Slavov, and E. Chown, "Omnidirectional walking using ZMP and preview control for the nao humanoid robot," in *RoboCup 2009: Robot Soccer World Cup XIII*, ser. Lecture Notes in Artificial Intelligence, J. Baltes, M. G. Lagoudakis, T. Naruse, and S. Shiry, Eds. Springer, to appear in 2010.

[6] T. Röfer, T. Laue, J. Müller, O. Bösche, A. Burchardt, E. Damrose, K. Gillmann, C. Graf, T. J. de Haas, A. Härtl, A. Rieskamp, A. Schreck, I. Sieverdingbeck, and J.-H. Worch, "B-Human Team Report and Code Release 2009," 2009, only available online: http://www.b-human.de/file_download/26/bhuman09_coderelease.pdf.

[7] C. Graf, A. Härtl, T. Röfer, and T. Laue, "A Robust Closed-Loop Gait for the Standard Platform League Humanoid," in *Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the 2009 IEEE-RAS International Conference on Humanoid Robots*, C. Zhou, E. Pagello, E. Menegatti, S. Behnke, and T. Röfer, Eds., Paris, France, 2009, pp. 30 – 37.

[8] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, "A Realtime Pattern Generator for Biped Walking," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA 2002)*, Washington, D.C., USA, 2002, pp. 31–37.

[9] T. Röfer, T. Laue, J. Müller, A. Burchardt, E. Damrose, A. Fabisch, F. Feldpausch, K. Gillmann, C. Graf, T. J. de Haas, A. Härtl, D. Honsel, P. Kastner, T. Kastner, B. Markowsky, M. Mester, J. Peter, O. J. L. Riemann, M. Ring, W. Sauerland, A. Schreck, I. Sieverdingbeck, F. Wenk, and J.-H. Worch, "B-Human Team Report and Code Release 2010," 2010, only available online: http://www.b-human.de/file_download/33/bhuman10_coderelease.pdf.

[10] J. Müller, T. Laue, and T. Röfer, "Kicking a Ball – Modeling Complex Dynamic Motions for Humanoid Robots," in *RoboCup 2010: Robot Soccer World Cup XIV*, ser. Lecture Notes in Artificial Intelligence, E. Chown, A. Matsumoto, P. Plöger, and J. R. del Solar, Eds. Springer, to appear in 2011.