

Instability Detection and Fall Avoidance for a Humanoid using Attitude Sensors and Reflexes

Reimund Renner and Sven Behnke
University of Freiburg
Computer Science Institute
D-79110 Freiburg, Germany
{renner, behnke}@informatik.uni-freiburg.de

Abstract—Humanoid robots are inherently unstable because their center of mass is high, compared to the support polygon’s size. Bipedal walking currently works well only under controlled conditions with limited external disturbances. In less controlled dynamic environments, such as RoboCup soccer fields, external disturbances might be large. While some disturbances might be too large to prevent a fall, some disturbances can be rejected by specific rescue behaviors.

This paper proposes a method to detect instabilities that occur during omnidirectional walking. We model the readings of attitude sensors using sinusoids. The model takes the gait target vector into account. We estimate model parameters from a gait test sequence and detect deviations of the actual sensor readings from the model later on. These deviations are aggregated to an instability indicator, which triggers one of two reflexes, based on indicator strength. For small instabilities the robot is slowing down, but continues walking. For stronger instabilities the robot stops and is brought into a stable posture with a low center of mass. Walking continues as soon as the instability disappears.

We evaluated our approach in simulation by disturbing the robot with a variety of impulses. The results indicate that our method is effective. For smaller disturbances, the probability of a fall could be reduced to zero. Most of the medium-sized disturbances could also be rejected.

I. INTRODUCTION

To act in the real world, a robot must be able to cope with the dynamics of the environment. Humanoid robots are much more flexible in comparison to wheeled or multi-legged robots, but are frequently not very stable during locomotion. Bipedal walking currently works well only under controlled conditions with limited external disturbances. This is due to a relatively high center of mass (CoM), compared to the size of the support polygon. Especially in the single-support phase of bipedal walking small external disturbances are sometimes sufficient to make a robot fall.

Although reliable standing up routines exist [1], falls should be avoided, as falling robots might damage themselves or parts of their environment. Hence, it is important to be able to detect instabilities and to prevent possible falls by specific rescue behaviors. If a fall cannot be avoided, the robot might try to minimize the damage by going into a protective pose [2].

Our robot makes use of omnidirectional walking [3] for flexible locomotion and has acceleration and gyroscope sensors for lateral and sagittal planes, which are fused to estimates of the tilt in roll and pitch direction. We use this attitude estimate and its derivative to detect disturbances during walking.

The sensor readings captured during undisturbed omnidirectional walking are modeled for some predefined gait speeds (support speeds). We model the means and standard deviations of the two tilts and their derivatives (turning rate). Each model consists of the low-frequency coefficients of the Fourier decomposition, which is synchronized to the gait cycle. Models for intermediate speeds are obtained by interpolation of the models of support speeds (support models).

We compute a stability indicator by comparing the actual sensor readings with the model for the actual gait speed. Depending on indicator strength, one of two reflexes is activated to stabilize the robot. For smaller instabilities, the robot is slowed down by decreasing the gait target speed. Thereby the robot still advances in the desired direction and we can further use the attitude model to detect instabilities. For stronger instabilities, a stop walk reflex is activated, which brings the robot into a standing posture with a lowered CoM in order to regain stability. A dynamical instability indicator is used to detect, whether the stop walk reflex is possible to stabilize the robot. If not, other reactions like a lunge step or damage protection are needed, which we do not present here. The dynamic instability is based on the distance of the capture point (CP) ?? to the border of the support polygon.

Baltes et al. [4] proposed a method to stabilize the walking gait of a humanoid robot using gyroscope sensors, which are similar to the derivative of our attitude sensors. However, they use it to control the gait itself by adjusting the starting points of the ankle patterns. The method is restricted to a fixed gait speed. In contrast, our approach is valid for any omnidirectional gait speed at limited ranges. We also take the tilt into account, which yields a better instability detection.

Another approach to prevent falls has been proposed by Höhn et al. [5]. They use pattern recognition to detect and classify instabilities. They also suggest that the ZMP has little significance as an instability indicator. The classification uses feature vectors, consisting of the linear and rotational velocity and the tilt of the torso and the foot, the CoP and the gait phase. A stabilizing reflex is initiated depending on the type of detected instability. The trained classification on the simulator has been successfully transferred to the real robot. The resulting classification time of instabilities was between 60 and 100ms, which is similar to the results of our approach.

We evaluated the proposed approach in a physics-based

simulation by disturbing the robot with a variety of impulses. The results indicate that our method is effective. For smaller disturbances, the probability of a fall could be reduced to zero. Most of the medium-sized disturbances could also be rejected. The evaluation on the real robot, by walking against a wall at different angles and velocities, showed similar results like in the simulation. An often used dynamic stability criterion, which was proposed by Vukobratović, is based on the zero moment point (ZMP) [6], [7]. Some of the humanoid robots using the ZMP for gait stabilization are Sony's QRIO [8], [9] and Honda's Asimo [10]. Another set of experiments showed that the proposed attitude-based instability indicator outperforms a ZMP-based indicator.

The structure of this paper is as follows. After reviewing the related work in the next section, we introduce our humanoid robot in Section II. Section III describes the model used for instability detection and details its parameter estimation procedure. In Section IV, we present the reflexive behaviors, which are used to stabilize the robot when instability is detected. Section V describes the experimental results obtained from systematic evaluation.

II. KIDSIZE HUMANOID ROBOT

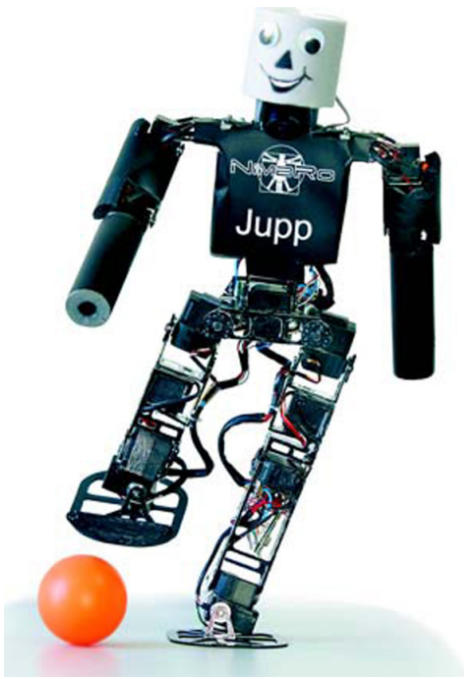


Fig. 1. KidSize robot Jupp of RoboCup Humanoid League team NimbRo.

Fig. 1 shows our humanoid robot Jupp, ready to kick the ball. It and its twin Sepp have been constructed in our lab to participate in the RoboCupSoccer Humanoid League [11]. In the 2005 competition, which took place in Osaka, Japan, they performed very well. As KidSize team NimbRo they came in second in the overall Best Humanoid ranking, next only to the titleholder Team Osaka.

Jupp is driven by 19 servos: 6 in each leg, 3 in each arm, and one in the trunk. Its mechanical design focused on human-like proportions and light weight. The robot has a size of 60cm and a total weight of only 2.3kg.

Jupp is fully autonomous. It is powered by Lithium-polymer batteries and equipped with a Pocket-PC and a wide-angle CF-camera. It has also two accelerometers and two gyros, which are used to estimate the tilt of the robot. The Pocket PC runs computer vision, behavior control, and communication. Using a hierarchical framework for reactive behavior control, we implemented omnidirectional walking [3], kicking, and basic soccer skills.

The 2 vs. 2 soccer games at RoboCup 2005 led to considerable physical contact between the robots, as multiple robots were going for the ball. Consequently, several falls occurred during the final match, which was played between NimbRo and Team Osaka. While both teams had implemented reliable standing-up routines for their robots [1], it would have been desirable to detect and to reject disturbances. A video of the game can be downloaded from <http://www.NimbRo.net>.

We used the attitude sensor of Jupp to detect a fall and to classify the robot posture (prone or supine). This sensor is also suitable to detect instabilities that occur during omnidirectional walking, which are mostly caused by external disturbances.

The attitude sensor is located in the trunk of the robot. It consists of a dual-axis accelerometer (ADXL203, $\pm 1.5g$) and two gyroscopes (ADXRS 150/300, $\pm 150/300$ deg/s). The four analog sensor signals are digitized with A/D converters of the HCS12 and are preprocessed by the microcontroller.

On the Pocket PC, the readings of accelerometers and gyros are fused to estimate the robot's tilt in roll and pitch direction. For each axis, the gyro bias is calibrated, assuming that over intervals of 2.4s the integrated bias-corrected gyro rates equal the difference between the tilts estimated from the accelerometers. Here we assume that, in the long run, the accelerometers measure the decomposition of the gravity vector. Combining the low-frequency components of the tilt estimated from accelerometers with the integrated bias-corrected gyro rates yields an estimate of the robot's attitude that is insensitive to short linear accelerations.

III. ATTITUDE MODEL

Because of the periodic nature of walking, a gait-cycle synchronous Fourier transform seems to be appropriate to model the attitude estimates. We use only the low-frequency coefficients of the transformed signals to keep the number of parameters small. The omnidirectional walking is parameterized by the gait target vector (v_x, v_y, v_θ) , where $v_x, v_y, v_\theta \in [-1, 1]$ are the lateral, sagittal and rotational speed components, respectively. The speeds are normalized, such that $v_y=1$ represents the maximal speed in forward direction.

The model of one constant gait speed consists of the first Fourier coefficients $C = (c_0, c_1, \dots, c_M)$ of the means μ_i and standard deviations σ_i at the gait phase $\Theta \in [-\pi, \pi)$. $i \in S = \{\varphi_l, \dot{\varphi}_l, \varphi_s, \dot{\varphi}_s\}$ enumerates the attitude sensors (φ) and their derivatives ($\dot{\varphi}$) in the lateral (l) and sagittal (s) planes.

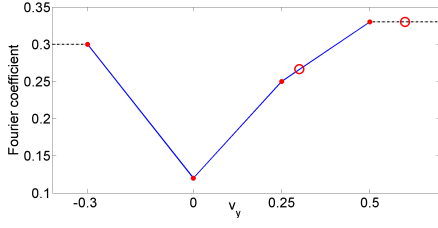


Fig. 2. Dependence of a single model parameter, i.e. one Fourier coefficient, on the sagittal speed v_y . The points denote support speeds. Circles are placed at arbitrary speeds. Parameters for intermediate speeds are interpolated linearly (solid line). For speeds outside the support range, nearest neighbors are used (dashed line).

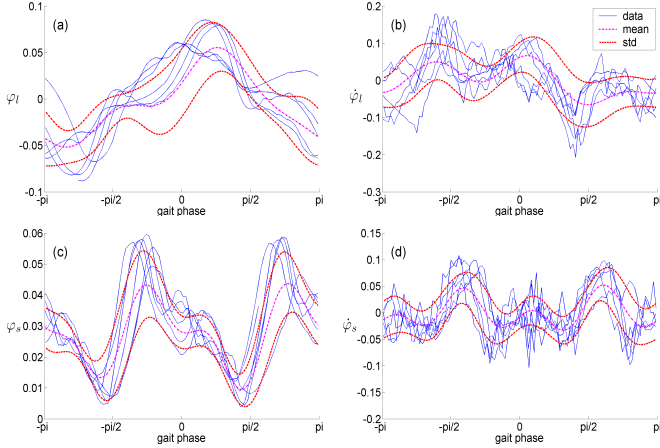


Fig. 3. Attitude sensor data of six gait cycles vs. gait phase Θ at speed $(0, 0.5, 0)$ with means and standard deviations obtained by inverse Fourier transform of the model: (a) lateral tilt, (b) derivative of lateral tilt, (c) sagittal tilt, (d) derivative of sagittal tilt.

M specifies the highest frequency used. The gait engine which generates omnidirectional walking is driven by a periodic gait phase Θ . $\Theta = \pm\pi$ denotes the left foot single support phase, $\Theta = 0$ the right foot single support phase, and $\Theta = \pm\frac{\pi}{2}$ the double support phases.

In order to obtain an attitude model for arbitrary gait speeds, we model attitude at specific support speeds and interpolate between the support models in a tri-linear fashion. For speeds outside the available support models we use the nearest neighbor. Fig. 2 depicts the speed space for one dimension, with its support speeds and intermediate values.

Fig. 3 shows attitude sensor data with its mean and standard deviation after inverse Fourier transform of the models, while walking at maximum forward support speed $(0, 0.5, 0)$.

A. Estimating Model Parameters

In order to estimate the parameters for the attitude models of the support speeds, the robot needs to walk at these speeds for a sufficiently long time. To explore the predefined support speeds, a walking test sequence is executed. The sequence contains mainly three parts:

- Walking with only one non-zero speed component (pure

lateral, sagittal and rotational speeds) up to the predefined maximal speeds,

- Walking with two speed components active at a time up to their absolute value maxima,
- Walking with all three speed components active.

As we cannot assume the sensor data to be linearly dependant on gait speed in general, many support speeds would be required to represent models for intermediate speeds. On the other hand, we wish to keep the length of the test sequence short. The total number of support models is $N_x * N_y * N_\theta$, where N_x, N_y, N_θ are the number of support speeds for the three speed components. In our simulation experiments we examined the detection for various parameter settings. Thus, already one or two support speeds for each speed component and direction (i.e. moving left, right, forward, backward and rotating left, right) gave an sufficient approximation. We chose $(\{-0.25, 0, 0.25\}, \{-0.3, 0, 0.25, 0.5\}, \{-1, -0.5, 0, 0.5, 1\})$ as the support speeds, so that the total number of support models was $3 * 4 * 5 = 60$. The number of Fourier coefficients for the models were selected as $M_{\varphi_l} = (4, 6)$, $M_{\dot{\varphi}_l} = (4, 7)$, $M_{\varphi_s} = (5, 6)$, $M_{\dot{\varphi}_s} = (5, 7)$.

Since the gait speed is changing continuously during the gait test sequence and thus never corresponds to any support speed, the sensor data is distributed to the adjacent support models by weighting it with the same weights, which are used for interpolation. The weights are proportional to the distance between the gait periods average speed v and the support speeds v_j and v_k , which are next to v with $v_j \leq v < v_k$ or $v_k \leq v < v_j$. The mean μ and standard deviation σ for support speed v_j is given by

$$\mu(v_j) = \frac{1}{\sum_{i=1}^N \rho_i} \sum_{i=1}^N \rho_i s_i \quad (1)$$

$$\sigma(v_j) = \sqrt{\frac{N}{N-1} \frac{1}{\sum_{i=1}^N \rho_i} \sum_{i=1}^N \rho_i (s_i - \tilde{\mu}(v))^2} \quad (2)$$

$$\rho_i = 1 - \frac{v - v_j}{v_k - v_j}, \quad (3)$$

where N is the number of gait periods belonging to the support speed v_j , $\tilde{\mu}(v)$ is the inverse Fourier transformed mean of $\mu(v)$, s_i is the sensor data of i^{th} gait period and ρ_i are the weights. As the standard deviation can be computed only at the end of the learning phase, it is useful to rewrite the sum in incremental form:

$$\sigma(v_j) = \sqrt{\frac{N}{N-1} \left(\frac{\sum_{i=1}^N \rho_i s_i^2}{\sum_{i=1}^N \rho_i} - \tilde{\mu}(v)(2\mu(v) - \tilde{\mu}(v)) \right)}. \quad (4)$$

In this way, we need to maintain only the sums $\sum \rho_i, \sum \rho_i s_i, \sum \rho_i s_i^2$ for each support speed.

Although the gait engine is symmetrical for the two legs, we observed that frequently the robot motion was not symmetrical. For example, the upper body might swing from left to right faster than from right to left. After some steps, the robot

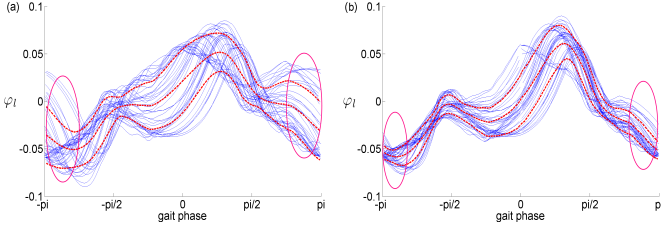


Fig. 4. Lateral attitude sensor at gait speed $(0, 0.5, 0)$ for 40 walking steps: original (left), with symmetry elimination (right). The dotted lines display the mean and standard deviation of the sensor values.

switches symmetry. If this effect is not accounted for, it leads to averaging between the two symmetrical cases, yielding a blurred model with high standard deviations. We eliminated this problem by converting the sensor data to one of the symmetric cases (symmetry elimination):

$$s_{l,s_1}(\Theta) \approx -s_{l,s_2}(\Theta + \pi) \quad (5)$$

$$s_{s,s_1}(\Theta) \approx s_{s,s_2}(\Theta + \pi), \quad (6)$$

where s_l and s_s denote the sensor values at gait phase Θ in lateral and sagittal plane, respectively, and s_1, s_2 are both symmetric cases. Figure 4 depicts the difference of sensor data with and without the lateral symmetry elimination. Especially, in the marked regions the difference in the variance is obvious.

To detect which case is present, the sensor data is converted to the symmetric case. The probability of case s_1 for each sensor $x \in S$ is obtained from the mean square error (MSE) to the corresponding model mean μ :

$$P_{x,s_1} = 1 - \frac{mse(s_{s_1}, \mu)}{mse(s_{s_1}, \mu) + mse(s_{s_2}, \mu)}. \quad (7)$$

The probability for the other case is computed accordingly. From the probabilities of all sensors, the overall probability is computed as weighted sum:

$$P_{l/s,s_2} = \frac{r_{\varphi_{l/s}} P_{\varphi_{l/s},s_2} + r_{\dot{\varphi}_{l/s}} P_{\dot{\varphi}_{l/s},s_2}}{r_{\varphi_{l/s}} + r_{\dot{\varphi}_{l/s}}} \quad (8)$$

$$P_{s_2} = \frac{r_l P_{l,s_2} + r_s P_{s,s_2}}{r_l + r_s}, \quad (9)$$

where l and s denote lateral and sagittal, respectively, and the weight $r = |0.5 - P|$ represents the certainty. Eqn. 8 first computes the lateral and sagittal sensor probability of s_2 . Eqn. 9 then computes the total probability P_{s_2} . Since the symmetric cases do not change frequently, the probability is recursively filtered with a time constant $\gamma \in [0, 1]$:

$$P_{s_2}^k = \gamma P_{s_2} + (1 - \gamma) P_{s_2}^{k-1}, \quad (10)$$

where $P_{s_2}^k$ is the probability of s_2 at gait period k . Best results were achieved with $\gamma = 0.25$ due to simulation results. Because of the averaging, wrongly detected symmetry cases, which may arise during the swap of the cases $s_1 \leftrightarrow s_2$, do not affect model learning. During instability detection, misclassifications could potentially trigger unnecessary stabilizing reflexes. As we show in the experimental results, however, the false alarm rate is very low.

B. Instability Detection

After the estimation of the model parameters is complete, we can use the learned attitude models to perform instability detection at every time step. Because the gait speed does not change fast, we can take for each whole gait cycle the model of the gait speed observed at the beginning of the gait cycle. Lateral symmetry elimination is done as before during parameter estimation (Eqns. 5-10). Since the symmetric case cannot be detected online, i.e. at every time step, but only after every finished gait cycle, the symmetric case of the previous gait cycle is used for the actual cycle.

The deviation δ_x^t of each sensor $x \in S$ from its model is computed at time step t as the difference to the mean, normalized by the standard deviation:

$$\delta_x^t = \frac{x^t - \mu^{\Theta(t)}(v_0)}{\sigma^{\Theta(t)}(v_0)}, \quad (11)$$

where $\Theta(t)$ is the gait phase at time step t and v_0 is the gait speed at the beginning of each gait period. The absolute value of the attitude deviation $\delta_{\varphi_{l/s}}^t$ and attitude derivative deviation $\delta_{\dot{\varphi}_{l/s}}^t$ results in the deviation $\delta_{l/s}^t$ in lateral or sagittal plane.

The greater $|\delta_{l/s}^t|$, the higher the instability in lateral and sagittal plane, respectively, which is expressed by the sigmoid function

$$\tau_{l/s}^t = 1 - \varsigma_{a,b}(|\delta_{l/s}^t|) = 1 - \frac{1}{1 + \exp\left(\frac{|\delta_{l/s}^t| - a}{b}\right)}, \quad (12)$$

where a is the threshold value and b is the slope.

The overall instantaneous instability τ^t is the maximum of lateral and sagittal instability τ_l^t, τ_s^t . Due to sensor noise and approximation errors it is quite noisy. We apply a recursive filter with two time constants, such that increasing instability is faster than decreasing it:

$$\tilde{\tau}^t = \max\left(\frac{\zeta \tau^t + (1 - \zeta) \tilde{\tau}^{t-1}}{\gamma \tilde{\tau}^{t-1}}\right), \quad (13)$$

with $\gamma \geq (1 - \zeta) \in (0, 1)$. $\tilde{\tau}$ is now a instability indicator, which can be used to trigger stabilizing reflexes. $\tilde{\tau} = 0$ represents no instability and $\tilde{\tau} = 1$ corresponds to maximal instability. The detection parameters were selected on simulation results: $a = 3$, $b = 1.4$, $\zeta = 0.15$, $\gamma = 0.91$.

IV. STABILIZING REFLEXES

In order to prevent falls, specific stabilizing reflexes are activated when instability is detected. Stumbling experiments with walking humans [12] showed that several reflexes were activated successively after a perturbation. While the first reflexes are fairly general, the responses of the next two reflexes depend on the gait phase. In a similar way, the instability indicator can be used to initiate general stabilizing reflexes at low instability, and more complex reflexes at higher instability. The importance of regulating reflexes during human locomotion is also emphasized in [13].

Currently, we use two simple reflexes to respond to detected instability:

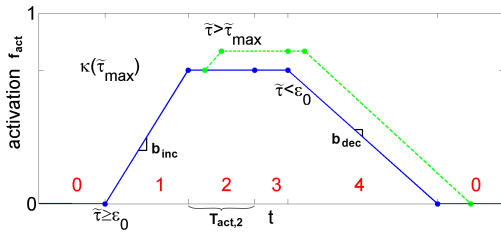


Fig. 5. Activation of slow down reflex (solid line) with phases 1-4 and adjustment after exceeding maximum instability $\tilde{\tau}_{max}$ (dashed line).

- **Slowing Down:** This reflex is activated by mild instabilities. The gait target speed is scaled down by at most a factor of two, depending on the instability $\varepsilon_0 \leq \tilde{\tau} < \varepsilon_1$. The activation of the slowing down reflex is structured into four phases (see Fig. 5):

- 1) Increasing activation with constant slope b_{inc} until
$$\kappa(\tilde{\tau}_{max}) = \begin{cases} \frac{\tilde{\tau}_{max} - \varepsilon_0}{\varepsilon_1 - \varepsilon_0} & \varepsilon_0 \leq \tilde{\tau}_{max} < \varepsilon_1 \\ 1 & \tilde{\tau}_{max} \geq \varepsilon_1 \end{cases},$$
 where $\tilde{\tau}_{max}$ is the maximum instability during activation
- 2) Maximum activation for a constant time period $T_{act,2}$
- 3) Maximum activation as long as instability is above ε_0
- 4) Decreasing slowly with a constant slope b_{dec} .

If at some point during activation the instability exceeds the maximal instability seen so far, the activation maximum is adjusted accordingly and the activation phase is set back to one (see dashed line in Figure 5). In the experiments the following parameters has been selected manually: $\varepsilon_0 = 0.3$, $\varepsilon_1 = 0.8$, $T_{act,2} = 0.6s$, $b_{inc} = 8.33 \frac{1}{s}$, $b_{dec} = 0.833 \frac{1}{s}$.

- **Stop Walking:** This reflex is activated for more serious instabilities. The robot stops walking and moves into a stable stand with a lowered CoM. Stop walking is activated when instability exceeds a threshold ε_2 . By setting $\varepsilon_2 \geq \varepsilon_1$, the stop walk is not triggered in the operating range of the slow down reflex. To detect whether the robot stands still, the derivative of the attitude sensor readings are processed in a similar way as for instability detection. We use a sigmoidal function and a recursive filter

$$I_{still}^t = \gamma_{s,a,b}(|(\dot{\varphi}_l, \dot{\varphi}_s)|) + (1 - \gamma)I_{still}^{t-1}. \quad (14)$$

I_{still}^t -values close to one indicate that the robot does not move any more. The five phases of activating the stop walking reflex are illustrated in Fig. 6:

- 1) Fast activation within a constant time period $T_{act,1}$ with smooth transition to phase 2
- 2) Full activation for a constant time period $T_{act,2}$
- 3) Full activation until standstill indicator $I_{still} > 0.9$
- 4) Slow deactivation within a constant time period $T_{act,4}$ and activation of an upright standing position, where the robot stands up

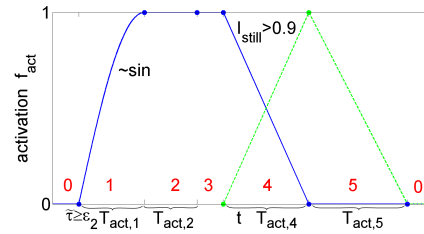


Fig. 6. Activation of stop walk reflex with phases 1-5.

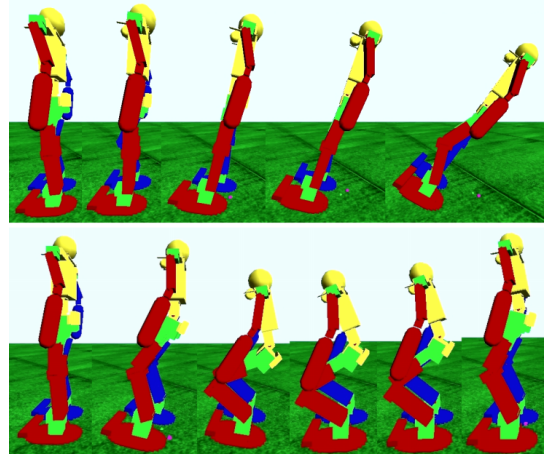


Fig. 7. Robot disturbed by an impulse force on its chest while walking backward at speed $(0, -0.3, 0)$. The upper picture shows the robot falling when no stabilizing reflex is activated. In the lower picture the stop walk reflex has been activated. The robot stops walking, crouches down and thereby prevents the fall.

- 5) Constant time period $T_{act,5}$ with zero activation and blocked reactivation to allow for reinitialization of stability detection. Thereby the upright standing is deactivated and the robot slowly increases the walking speed.

Here the parameters were chosen as $T_{act,1} = T_{act,2} = 0.6s$, $T_{act,4} = 1.44s$, $T_{act,5} = 2.64s$, $T_{act,0} = 2.4s$. The instability threshold for the activation has been set to the upper threshold of the slow down reflex, i.e. $\varepsilon_2 = \varepsilon_1 = 0.8$.

Additionally, we use the capture point (CP) [?], proposed by Pratt et al., to measure the dynamic instability of the robot as the distance of the CP to the border of the support polygon. The CP defines the point on the ground, where the robot comes to rest if it moves its CoP to this point. The location of the point is estimated by the linear inverted pendulum model (LIPM) and the robots CoM velocity. For the real robot the velocity can also be estimated by the gyros and the accelerometers after removing the gravity component through the estimated attitude sensor. The support polygon can be estimated by using the attitude sensor and applying forward kinematics with the requested joint angles. With this, it is possible to compute the positions of the feet and the CoM and their distances to the ground.

If the CP lies within the support polygon, then the stop walk reflex is able to stabilize the robot. Then, the depth of the

crouching during the stop walk reflex can be set proportional to the dynamic stability. Otherwise, if the CP lies outside the support polygon, other kind of reactions, like a lunge step, are needed. Or if the CP is too far, no fall avoidance is possible, and we need to make damage protecting reactions. However, this is not part of this work.

Another stopping method was proposed by Morisawa et al. [14]. When being in the single support phase, the robot first makes a final step to obtain double support. Then movements are computed to slow down the robot, based on approximated mathematical models using the CoM and the ZMP. The CoM is also lowered to increase stability. In our case, the double support is achieved by the continuously increasing activation of the stopping reflex. This moves the robot smoothly into a stable standing posture. Fig. 7 shows the robot after an impulse disturbance while walking. The robot falls when no stabilizing reflexes are activated, but stabilizes when they are.

V. EXPERIMENTAL RESULTS

To evaluate our fall detection and fall avoidance approach, we disturbed the robot during walking in the ODE-based simulator with a variety of impulse forces. The impulses are produced by a constant force, which acts for a single time step onto the robot.

We first compared the attitude sensor based instability with an instability indicator that is based on the ZMP criterion. Fall statistics have been made to show that the instability detection is effective for activating stabilizing responses.

As already mentioned in the first section, the ZMP can be used as a dynamic stability criterion so that a comparison is meaningful. To do this, an instability indicator is computed in the same way as for the attitude readings (Eqns. 12-13). We use the distance of the ZMP to the closest edge of the support polygon as input and tested parameters for the sigmoid and the recursive filter. We compare the forewarn time with the false alarm rate. The forewarn time is the time between the first rise of the instability above a threshold of 0.5 and the time when the attitude angle of the robot deviates more than 25° from the upright posture. The false alarm rate is the percentage of instability detections (>0.5) during undisturbed, stable gait. Both are averaged over varying gait speeds (forward, backward, left, turn left at different speeds), forces (6, 9, 12), force directions (trunk front, rear, left, right; left upper arm front, rear), and five disturbance time points (relative to gait phase). As the forewarn time and the false alarm rate are dependant on each other, the instability is computed for varying sigmoid and recursive filter parameters. Since the robot hardly gets onto the whole planar sole of a foot during walking in the simulator, the support polygon is defined as the part of the sole of a foot with a distance smaller than 1cm to the ground.

Fig. 8 shows the false alarm rate depending on the forewarn time for the attitude sensors and the ZMP. It can be observed that the ZMP-based instability indicator is outperformed by the

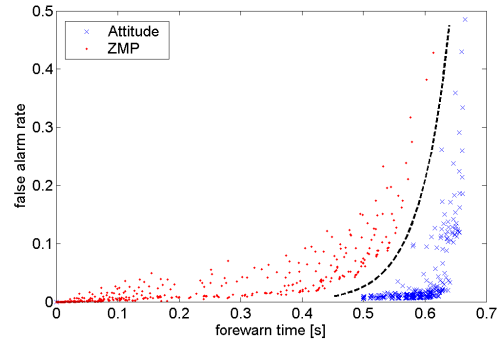


Fig. 8. Detection comparison of the attitude sensor instability indicator and the ZMP criterion: forewarn time and false alarm rate depending on detection parameters. The dashed line shows a clear separation of both instability indicators.

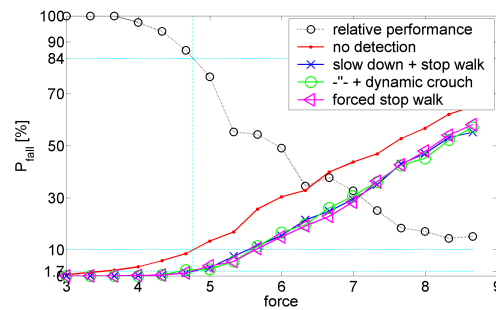


Fig. 9. Fall probability when disturbing the robot during walking by an impulse force without any reaction, with slow down and stop walk reactions, with dynamic crouch and with forced stop walk activation of 96ms before the disturbance. The dotted line shows the relative performance. The 10% working point is marked by the dashed lines.

attitude-based instability indicator. This proves the effectiveness of using attitude information for an instability indicator.

To evaluate the fall avoidance, topple down statistics have been made in the simulator. The robot has been disturbed again by an impulse force at varying gait target speeds, forces (3 to 8.66 in 0.33 steps), force directions, and support phases (double, single left, single right). Altogether, four kinds of experiments were done: without reaction, with slow down and stop walk reflexes, additionally dynamic crouching durch stop walk and forced stop walk 96ms before the disturbance. We made 20 repetitions.

Compared to stopping the robot in an upright posture, shortening the robot's legs and bending its trunk forward resulted in a more stable posture. Because the disturbance rejection capabilities of a feed-forward stopping reflex are limited, we chose the working point of 10% fall probability for comparison with continued omnidirectional walking. Fig. 9 depicts the fall probabilities and the relative performance vs. the impulse magnitude of the disturbance.

We achieved a fall probability of 1.7% in the working point. This corresponds to 83% prevented falls. For very strong disturbances, the relative performance decreases until static stopping has no effect anymore. As can be seen in

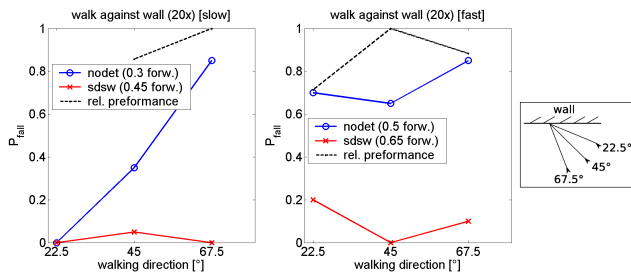


Fig. 10. Evaluation with real humanoid robot Jupp: Walking forward against a wall at various angles. Left: results at low speed, mid: results at high speed, right: experiment configuration.

the figure, there is basically no difference between the three experiments using reflexes. This shows, that the dynamic instability detection based on the capture point is practical. Furthermore, it shows, that the a significant disturbance is detected within the first 100ms.

The evaluation of instability detection and fall avoidance on a real robot is not easy, because it is hard to reproduce the perturbations. We decided to let the robot walk against a wall at different speeds and various angles, because it is easy to reproduce the experiments for statistical evaluation. The experimental configuration is shown at the right picture of fig. 10. The two left pictures show the results of 20 repetitions for a lower and higher walking speed, respectively. As can be seen, at lower speed, the fall probability without any reaction grows with increasing walking angle. This is due to the fact, that most of the falls emerge from the pushing of the swinging arms against the wall. In contrast, when instability detection and the both stabilizing reflexes are used, the fall probability is zero for two cases and 5% for the third case. At the higher speed the fall probability is not higher than 20%, while without reaction the probability is greater than or equal to 65%. Thus, the results with the real robot verify the effectiveness of our attitude based instability detection, to be fast enough to be able to prevent falls by suitable reactions.

During the experiments, we also observed that the robot was able to walk at higher speeds than during parameter estimation. The speed amplitudes for parameter estimation were set as high as possible, so that walking would be smooth. With the stabilizing reflexes, the robot could walk at higher speeds, while still producing smooth movements. This is a consequence of the slow down reflex, which decreases the walking speed depending on the instability.

VI. CONCLUSION

We presented an approach for instability detection during omnidirectional walking of humanoid robots, using attitude sensors and their derivative. The detection is based on the deviations of the actual sensor readings from a model. The model takes the walking speed into account. It is not restricted to any fixed walking speed or direction. We estimate the model parameters from a test sequence of undisturbed omnidirectional walking. Depending on the magnitude of the

detected instability one of two simple stabilizing reflexes is triggered. For minor instabilities we slow down walking. Larger instabilities trigger a stop walking reflex, which moves the robot into a stable posture with low CoM.

The comparison with a ZMP-based instability criterion showed much lower false alarm rates at equal forearm times when using our approach.

A systematic evaluation showed that the stabilizing reflexes reduce fall probability to zero for smaller disturbances. At the working point of 10% falls for continued walking, the stabilizing reflexes prevent more than four of five falls. In addition, the adaptive speed control caused by the slow down reflex resulted in smoother walking at speeds above the maximal speed used for parameter estimation.

The experiments with the real robot showed similar results as in the simulations. For a slower walking speed nearly all falls could be rejected. At the higher speed in the worst case still 70% of the falls could be rejected.

In the future, we plan to add more complex stabilizing reflexes, e.g. lunge steps, to be able to reject very large disturbances. In order to minimize damage, we also plan to move the robot into a protective pose when a fall turns out to be unavoidable.

ACKNOWLEDGMENT

Support for this work was provided by grant BE 2256/2-1 of Deutsche Forschungsgemeinschaft (DFG).

REFERENCES

- [1] J. Stückler, J. Schwenk, and S. Behnke, "Getting Back on Two Feet: Reliable Standing-up Routines for a Humanoid Robot," *The 9th International Conference on Intelligent Autonomous Systems (IAS-9)*, Mar. 2006.
- [2] K. Fujiwara, F. Kanehiro, S. Kajita, K. Kaneko, K. Yokoi, and H. Hirukawa, "UKEMI: Falling Motion Control to Minimize Damage to Biped Humanoid Robot," in *Proceedings of the 2002 IEEE/RSS International Conference on Intelligent Robots and Systems*, Oct. 2002, pp. 2521–2526.
- [3] S. Behnke, "Online Trajectory Generation for Omnidirectional Biped Walking," *IEEE International Conference on Robotics and Automation (ICRA)*, May 2006.
- [4] J. Baltes, S. McGrath, and J. Anderson, "The Use of Gyroscope Feedback in the Control of the Walking Gaits for a Small Humanoid Robot," in *RoboCup 2004: Robot Soccer World Cup VIII*, ser. LNCS, no. 3276, 2005, pp. 628–635.
- [5] O. Höhn, J. Gacnik, and W. Gerth, "Detection and Classification of Posture Instabilities of Bipedal Robots," in *Proceedings of the 8th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines - CLAWAR*, 2005.
- [6] 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.
- [7] A. Goswami, "Postural stability of biped robots and the foot rotation indicator (FRI) point," *International Journal of Robotics Research*, vol. 18, no. 6, pp. 523–533, June 1999.
- [8] Y. Kuroki, T. Ishida, M. Fujita, and T. T. Doi, "A Small Biped Entertainment Robot," *Journal of Robotics and Mechatronics*, vol. 14, no. 1, pp. 6–12, 2002.
- [9] F. Tanaka, K. Noda, T. Sawada, and M. Fujita, "Associated emotion and its expression in an entertainment robot qrio," in *Entertainment Computing - ICEC*, Sept. 2004, pp. 499–504.
- [10] K. Yokoi, F. Kanehiro, K. Kaneko, K. Fujiwara, S. Kajita, and H. Hirukawa, "A Honda Humanoid Robot Controlled by AIST Software," in *IEEE/RAS International Conference on Humanoid Robots*, Tokyo, Japan, 2001, pp. 259–264.

- [11] S. Behnke, "Playing Soccer with Humanoid Robots," *KI – Zeitschrift Künstliche Intelligenz*, 2006.
- [12] A. M. Schillings, B. M. H. V. Wezel, T. Mulder, and J. Duysens, "Muscular Responses and Movement Strategies during Stumbling over Obstacles," *Journal of Neurophysiology*, no. 83, pp. 2093–2102, 2000.
- [13] P. Zehr and B. Stein, "What functions do reflexes serve during human locomotion?" *Prog Neurobiol.*, vol. 58, no. 2, pp. 185–205, June 1999.
- [14] M. Morisawa, S. Kajita, K. Harada, K. Fujiwara, K. K. Fumio Kanehiro, and H. Hirukawa, "Emergency Stop Algorithm for Walking Humanoid Robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.