O. Höhn
Baker Hughes, INTEQ GmbH,
29221 Celle, Germany
oliver.hoehn@bakerhughes.com

W. Gerth
Institute of Automatic Control
Leibniz University of Hannover
30167 Hannover
hoehn@irt.uni-hannover.de

Probabilistic Balance Monitoring for Bipedal Robots

Abstract

In this paper, a probability-based balance monitoring concept for humanoid robots is proposed. Two algorithms are presented that allow us to distinguish between exceptional situations and normal operations. The first classification approach uses Gaussian-Mixture-Models (GMM) to describe the distribution of the robot's sensor data for typical situations such as stable walking or falling down. With the GMM it is possible to state the probability of the robot being in one of the known situations. The concept of the second algorithm is based on Hidden-Markov-Models (HMM). The objective is to detect and classify unstable situations by means of their typical sequences in the robot's sensor data. When appropriate reflex motions are linked to the critical situations, the robot can prevent most falls or is at least able to execute a controlled falling motion. The proposed algorithms are verified by simulations and experiments with our bipedal robot BARt-UH.

KEY WORDS—robot, balance monitoring, fall detection, reflex motions, HMM, viability kernel, classification

1. Introduction

When bipedal robots step out of the laboratories they will suffer from real world problems. The ground will be uneven and slippery and there will be external incidents like a shove. The common stability criteria such as center of gravity (CoG), zero moment point (ZMP) or center of pressure (CoP) (Goswami1999; Vukobratović and Borovac 2004; Popovic et al. 2005) are applicable for offline path planning or for a feedback control when there are minor instabilities. If the robot is about to fall, they are of little help in deciding on the appropriate reaction. This is due to their stability concept whereby the robot is stable as long as the stance foot rests on the ground surface. In any case, a ZMP at the border of the stability region is only a necessary but not a sufficient condition for a forthcoming fall. Even a foot tilting angle does not necessarily mean that the robot will fall down. Consider as an example a wobbling motion if a standing robot has been slightly pushed.

In this paper it is intended to provide a monitoring system that initiates appropriate reactions when the robot is falling down. Since the states in which a fall is unavoidable depend on the given control strategies, it is also essential to consider some control policies. For our research, the border region between states that are still stable and those where a fall is unpreventable is especially of interest. Hence, we used reflex steps because of their excellent ability to avert falls in severe situations. Additionally, special crouching motions were implemented to minimize damages when a fall is unavoidable. The ability of our robot BARt-UH (Albert 2002) to recover its balance with reflex motions was verified in former experiments by Höhn et al. (2004).

Other strategies that deal with optimal reactions when the robot is in an exceptional situation can be found in the literature. Morisawa et al. (2005) proposed an emergency stop algorithm that allows the robot to take a statically stable posture within one step. The signal to trigger the emergency stop motion was provided externally. Fujiwara et al. (2003, 2004, 2006) developed methods that enable a human size robot to fall over safely without becoming seriously damaged. For their fall experiments, the robot's initial state was a standing posture. In this special case it was possible to evaluate the CoG for the initiation of the falling motion. As described above, this is a very sensitive criteria. Hence, it is possible that a reflex falling motion is triggered even if the robot is not about to fall.

The International Journal of Robotics Research
Vol. 28, No. 2, February 2009, pp. 245–256
DOI: 10.1177/0278364908095170

The main objective of this paper is the design of a balance monitoring system that is supposed to work when the robot is walking or executing any other arbitrary tasks. Depending on the current situation, the system decides which of the available control strategies, reflex steps or reflex crouching motions to apply. We made the observation that even humans need some exercises in order to improve their ability to avoid falls or severe injuries. Unfortunately, a training phase with the real robot would take a long time and could entail serious damages. Our approach is to use a simulated robot model for the training process, since in simulated fall scenarios it is quite easy to evaluate which reaction is best in a given situation. The provided skills are transferred to the real biped afterwards.

The paper is organized as follows. Section 2 provides information on basic reflex motions. In Section 3 stability considerations in the state space are discussed. From this, follows the idea to trace back the balance monitoring to a common classification problem (Section 4). The classification with Gaussian-Mixture-Models (GMMs) is introduced in Section 5 and the Hidden-Markov-Model (HMM) approach is described in Section 6. Furthermore, the simulated training process that is used to gather the feature vectors for the parameterization of the probabilistic algorithms is presented in Section 7. Finally, results that have been achieved in simulations and by experiments with our bipedal robot BARt-UH are discussed in Section 8.

2. Basic Reflex Motions

Optimal reflex trajectories can be obtained quite easily by mimicking human behaviors. Bortolami et al. (2003) show that humans demonstrate three strategies to prevent falls. The ankle and hip strategy are used if there are minor disturbances. These strategies are similar to the common control approaches for bipedal robots where a ZMP-based control is applied or the reference trajectory is adapted depending on the robot's current state. The third strategy, the execution of reflex actions, is used if humans are close to a fall. The characteristic of a reflex is the abortion of the active task, e.g. the walking. Since only severe disturbances are regarded here, the control strategies for our examinations are precalculated reflex motions. Each reflex is determined by two third-order splines that describe the motion of the torso and the swing foot. The splines have a fixed end point and start at the current position of torso and foot, respectively. The execution time for a reflex motion is ≈500 ms.

When the robot is close to a fall, it can be advantageous to perform a controlled falling motion (strategies to upright a fallen robot afterwards can be seen e.g. in the multimedia extensions 3 and 4). However, the design of reflex falling motions raises the question of how to minimize the impact energy. As well as passive techniques such as padding the first impact points (Fujiwara et al. 2002), a general question is how to move the robot's center of mass. An appraisal can be carried out by regarding the simplified inverted pendulum model in Figure 1.

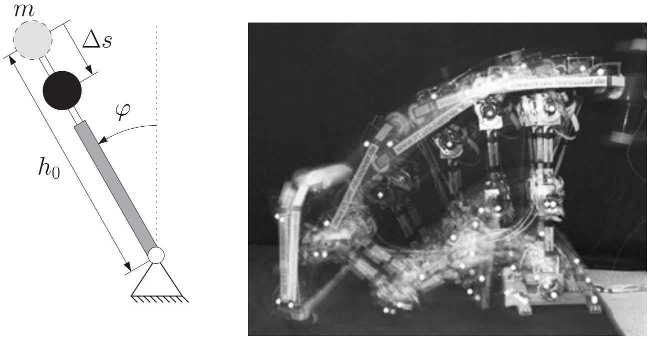

Fig. 1. Left: simplified pendulum model that is used to analyze falling down strategies; right: sequence of a typical crouching motion (see multimedia extensions 1 and 2).

Assuming that the crouching and stretching motions can be executed infinitely fast, from the conservation of momentum follows

$$\dot{\varphi}_1 = \dot{\varphi}_0 \frac{J_0}{J_1} = \dot{\varphi}_0 \frac{h_0^2}{(h_0 - \Delta s)^2}. \tag{1}$$

The pendulums rotational speed is termed with $\dot{\varphi}$, J is the mass moment of inertia, h_0 is the pendulum length and Δs denotes the displacement of mass m during the reflex motion (Figure 1).

The index 0 denotes the state before and the index 1 the state immediately after the reflex motion. This leads to a change of the robot's energy by

$$\begin{aligned} \Delta E &= \underbrace{\frac{m\,h_0^4\,\dot{\varphi}_0^2}{2}\left(\frac{1}{(h_0-\Delta s)^2} - \frac{1}{h_0^2}\right)}_{\Delta E_{\mathrm{kin}}} \\ &- \underbrace{\Delta s\, m\, \mathrm{g}\, \cos(\varphi_0)}_{\Delta E_{\mathrm{pot}}}. \end{aligned} \tag{2}$$

From Equation (2) it follows that a displacement of the CoM by Δs affects the kinetic and potential energy in opposite ways. The maximum reduction of the impact energy can be achieved by lowering the CoM at an early stage. This minimizes the potential energy at most and only has a little influence on the kinetic energy since $\dot{\varphi}_0$ is still small at the beginning of a fall. The alternative would be to reduce the kinetic energy by the maximum by lifting the CoM as late as possible. Assuming realistic fall intensities, no minimization can be achieved at tilting angles φ_0 near 45°. These considerations agree with the measurements presented in Figure 2. From a practical point of view the only falling down reflex can be to lower the CoM as far and as soon as possible.

However, this is contradictory to the reflex steps that are used to recover the robot's balance. Here, the goal is to lift the

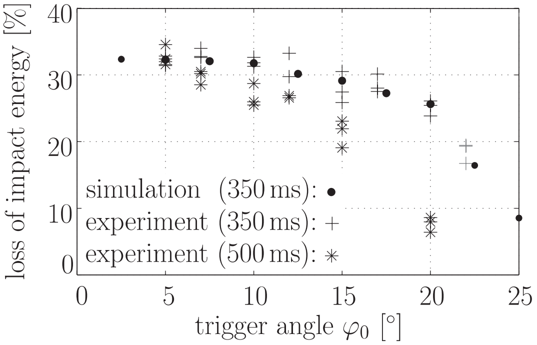

Fig. 2. Reduction of impact energy obtained by lowering the robot's torso by 11 cm. The squatting motions took 350 ms or 500 ms and were triggered at different foot rotational angles φ_0. For the simulation we used our multi-body model of BARt.

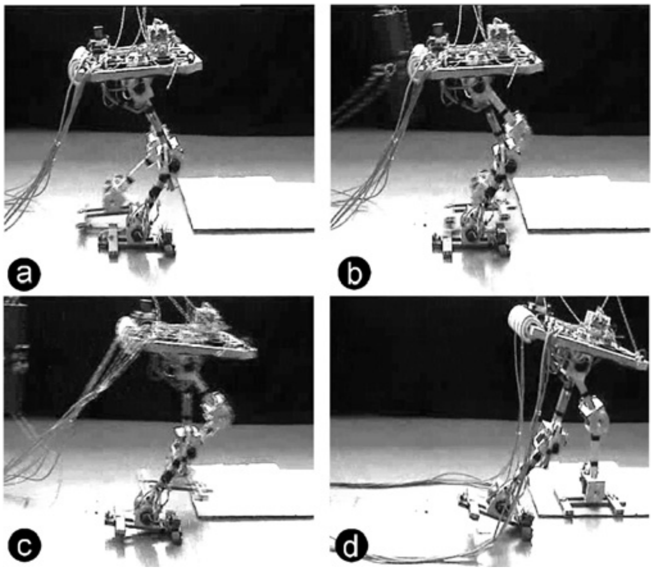

Fig. 3. The sequence shows an impact experiment with BARt-UH. After the swinging weight hit the robot, a reflex step is performed. The duration of the whole sequence was approximately 0.5 s.

torso as fast and as far as possible to prevent the swing foot from touching the ground prematurely (Figure 3).

Hence, it is an ultimate decision how to react since there is no way back once a reflex motion has been initiated. From this it follows that the monitoring system has to be reliable and fast. Humanoid robots have a certain dimension and mass, so it takes some time for a falling motion to proceed. Our experiments show that the reflex motions should be initiated not later than 150–200 ms after the robot has started to lose its balance.

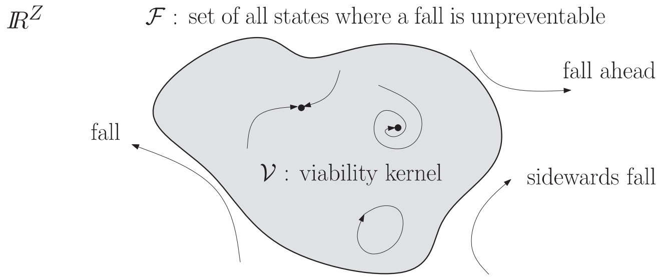
$\mathbb{R}^Z$
$\mathcal{F}$: set of all states where a fall is unpreventable
fall ahead
fall
$\mathcal{V}$: viability kernel
sidewards fall

Fig. 4. The viability kernel $\mathcal{V}$ gathers all states in the state space $\mathbb{R}^Z$ where at least one control input sequence exists that can prevent the biped from falling over. If the robot's state gets into $\mathcal{F}$, a fall is unavoidable.

3. Stability Considerations

To cope with situations where the robot's balance is no longer controllable, a new stability concept is needed. This section explains the underlying idea of the proposed algorithms with help of a stability notion that was presented by Wieber (2002). This approach is based on state space considerations and perfectly matches the requirements when dealing with severe balance disturbances.

The stability of a bipedal robot is dependent upon a combination of influences. As well as the foot–ground reaction forces, the robot's configuration (the position of the torso and of the swing foot with respect to the stance foot) plays an important role in determining if an impending fall can be averted. For this reason, it is not sufficient to look only at the reaction forces: it is rather a question of the robot's state. Wieber (2002) terms all states where a fall is unavoidable with $\mathcal{F}$. A state is called viable if at least one control input sequence exists that prevents the robot's state from getting into $\mathcal{F}$ (Figure 4).

The union of all viable states is called viability kernel $\mathcal{V}$. The robot is stable as long as its state stays inside $\mathcal{V}$ and never gets inside $\mathcal{F}$. Whether a fall can effectively be prevented depends on the robot's capabilities. If the appropriate input sequence is unknown, the robot will fall down even if its state is inside $\mathcal{V}$.

For each control law, a subset of $\mathcal{V}$ exists where a fall is avoidable. For a given law, the robot's state is therefore only stable inside the associated set. By providing a number of different control strategies, the effective stable subset is composed by the union of all subsets.

Our approach is to provide various reflex step motions that cover the viability kernel as far as possible with their sets $\mathcal{R}_k^{\text{step}}$. Additionally, there are multiple reflexes that minimize damages if the robot is falling. Of course, other reflex motions and control strategies can be included as well.

A schematic of the feature space is presented in Figure 5. Within the viability kernel $\mathcal{V}$, different subsets can be seen where a given control law can prevent the robot from falling.

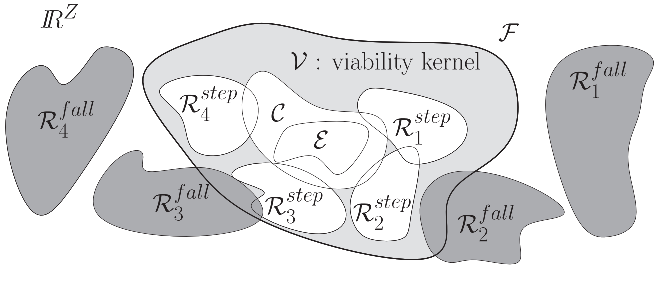

Fig. 5. The different sets denote regions in the state space where a given behavior is adequate in terms of fall prevention or damage minimization.

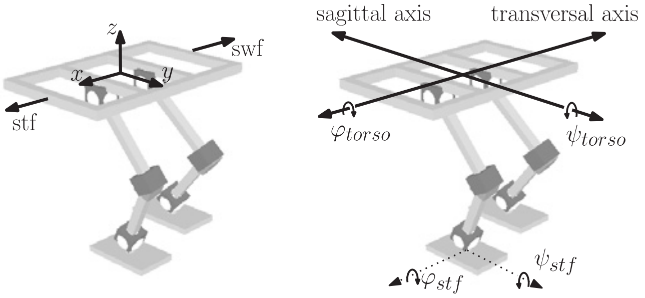

Fig. 6. Denomination of the coordinate systems.

The set $\mathcal{E}$ contains the inherent stable states, where no control intervention is needed. The states where a feedback control (e.g. based on the ZMP) is appropriate are centralized in $\mathcal{C}$. Additionally, different reflex motions are available. Sets where a reflex step can prevent a fall are indicated with $\mathcal{R}_k^{\text{step}}$. States where serious damages can be averted by a reflex motion are termed $\mathcal{R}_k^{\text{fall}}$.

4. Classification

As can be seen in Figure 5, multiple control laws are required for maximum stability of the robot. Deciding on the most appropriate control law is a typical classification problem where the classes correspond with the given control strategies. A control strategy (class) is composed of two parts as follows.

1. Control Law: The control law is either the algebraic formulation of a feedback control law or a precalculated reflex motion.

2. Activation Set: The activation set Ω_k of a reflex R_k is related to the set $\mathcal{R}_k$ but is not identical. Ω_k only contains the states where the corresponding control law is suitable. This considers the fact that multiple subsets $\mathcal{R}_k$ in feature space may overlap. The set Ω_k does not cover all states where a fall can be prevented with the control law, but covers the states where the stabilizing performance is best compared to the other strategies. If a fall can be avoided by a feedback control (e.g. an ankle strategy based on the ZMP) as well as by a reflex step, the feedback control is preferred and the reflex is not adequate.

One primary object for the proposed balance monitoring concept is the applicability to the real robot. Therefore, the stability considerations are mapped from the state space to the feature space. This space is spanned by the feature vector $\boldsymbol{x} \in \mathbb{R}^M$ that contains the normalized measuring data from the robot's sensors (Höhn et al. 2005). This vector is advantageous for the classification algorithms because it is easy to measure and it usually has a lower dimension than the state vector.

As features, we select all information appropriate to express the robot's state. However, the precondition which must be considered is that these features are measurable. Our robot BARt is equipped with the following sensors.

1. Force sensors: Four force sensors are situated below each of BARt's feet. They allow the measurement of CoP of the current stance foot (stf) in transversal (trans) and sagittal (sag) direction (Figure 6). Additionally, the CoP of the swing foot (swf) is evaluated. This becomes interesting if the robot trips over a step and the swing foot has ground contact prematurely. In conclusion, the following four features are used: $\text{CoP}_{\text{stf,trans}}$, $\text{CoP}_{\text{stf,sag}}$, $\text{CoP}_{\text{swf,trans}}$, $\text{CoP}_{\text{swf,sag}}$.

2. Inertial Measurement Unit (IMU): An Inertial Measurement Unit (Strasser et al. 2003) is mounted on top of BARt. It allows the measurement of the torso rotation angle φ_{torso} around the sagittal axis and its rotational velocity $\dot{\varphi}_{\text{torso}}$ as well as the rotation angle ψ_{torso} around the transversal axis and its rotational velocity $\dot{\psi}_{\text{torso}}$. At the beginning of a typical fall, no rotation around the vertical axis occurs. For this reason, features that describe the gyration around the vertical axis (z axis) were not evaluated.

3. Micro Inertial Measurement Unit (μIMU): The μIMU is a smaller version of the IMU. It is mounted on the feet and is used to measure the attitude and rotational velocities φ_{stf}, $\dot{\varphi}_{\text{stf}}$ ψ_{stf}, $\dot{\psi}_{\text{stf}}$ of the stance foot.

4. Step time: The step time T_S is no sensor information. It is a virtual, software-generated time that is used to incorporate the nominal configuration of the robot. T_S takes the values from –1 to +1 during one step. The robot is in the double support phase for $T_S = \pm 1$. In the following we distinguish $P = 6$ step phases, where $-1 \leq T_S < -0.66$ is phase one, $-0.66 \leq T_S < -0.33$ is phase two, and so on.

The feature vector $\boldsymbol{x}$ is composed by the listed features m_i. Since the features m_i usually have different metric units, they have to be normalized as follows: $\tilde{m}_i = m_i/\sigma_i$. The normalization prevents $\boldsymbol{x}$ from being dominated by single features with large absolute values. As a scaling factor, we use the standard deviation

$$\sigma_i^2 = \frac{1}{N-1} \sum_{m_i \in \boldsymbol{X}} (m_i - \bar{m}_i)^2 . \tag{3}$$

The expectation value $\bar{m}_i$ and the standard deviation σ_i are calculated from the set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ that contains all training vectors that were used for the parametrization of the classification algorithms (see Section 7). To improve the robustness of the parameter estimation it is advantageous to reduce the dimension of the feature space. Less parameters therefore have to be estimated. As can be imagined, there are some features such as $\dot{\varphi}_{\text{torso}}$ and $\dot{\varphi}_{\text{stf}}$ that are highly correlated. For this reason, we used a Karhunen-Loève transformation to decorrelate the feature vector and to reduce its dimension (Fukunaga 1972; Fink 2008). It emerged that the classes of the balance monitoring system (proposed in Section 7) can still be separated robustly with the HMM classifier if the dimension is reduced to $M' = 2$ (Section 8.3).

It must be noted that we do not distinguish between falls to the left or to the right side. Since the reactions depend on whether the robot is falling to the stance foot or to the swing foot side, we distinguish between stf and swf side. For this reason, the feature vector is dependent upon the current foot configuration.

5. GMM Classification Approach

Classifying the robot's state is defined as assigning the current feature vector $\boldsymbol{x}$ to an activation set $\Omega_k \in \{\Omega_1, \ldots, \Omega_K\}$. In this section, a probability-based approach is used to determine the most likely set Ω^* given a vector $\boldsymbol{x}$:

$$\Omega^* = \underset{\Omega_k}{\operatorname{argmax}} P(\Omega_k|\boldsymbol{x}). \tag{4}$$

Here, $P(\Omega_k|\boldsymbol{x})$ is the probability of Ω_k being the best activation set given $\boldsymbol{x}$. Applying the Bayes rule, we get:

$$P(\Omega_k|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\Omega_k)P(\Omega_k)}{P(\boldsymbol{x})}. \tag{5}$$

$P(\boldsymbol{x})$ defines the probability of the occurrence of vector $\boldsymbol{x}$. This probability is constant for all Ω_k and can be neglected for the maximizing task. The *a priori* probability $P(\Omega_k)$ is the probability of occurrence of a (fall) scenario. Since stability disturbances are mostly unpredictable, a convenient assumption is to set $P(\Omega_k) = \text{const} > 0 \, \forall \, k \in \{1, \ldots, K\}$. If there are specific control strategies that depend on the robot's environment, $P(\Omega_k)$ can alternatively be used to fade some strategies out by setting it to zero. The control strategy for the case of falling down stairs can then be disabled if the robot is walking on a plane.

The conditional *a posteriori* probability $P(\boldsymbol{x}|\Omega_k)$ states the likelihood that the feature vector $\boldsymbol{x}$ is emitted when the robot's state is inside the activation set Ω_k. In general, the probability density of an activation set is not normally distributed. In the following, $P(\boldsymbol{x}|\Omega_k)$ is approximated by a superposition of multiple Gaussians, i.e.

$$P(\boldsymbol{x}|\Omega_k) = \sum_{i=1}^{C_k} w_{ki} \, \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{ki}, \Sigma_{ki}) \tag{6}$$

with

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^M |\Sigma_k|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_k)}. \tag{7}$$

The Gaussian-Mixture-Models (GMM) are universal approximators and can be used to model arbitrary distributions. For our research, we typically used a superposition of $C_k = 1, \ldots, 4$ normal distributions. With it, each activation set Ω_k is completely described by the weights w_{ki}, the mean vectors $\boldsymbol{\mu}_{ki}$ and the covariance matrices Σ_{ki} of the GMM. To improve the robustness of the parameter estimation (Section 7), the number of unknown parameters has to be reduced. We therefore approximated Σ_{ki} by a diagonal matrix.

The GMM-based classification was discovered to be slightly susceptible to misclassifications because the algorithm only evaluates the current feature vector. A significant improvement of the robustness can be achieved by introducing a minimum classification duration $T_V \approx 50\,\text{ms}$. This means that an unstable situation has to be detected for at least 50 ms before a reflex motion is triggered.

6. HMM Classification Approach

Instead of classifying the robot's state by means of a single feature vector as with the GMM, the HMM classification evaluates the whole occurring sequence $\boldsymbol{O} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ of feature vectors.

As visualized in Figure 7, there is a limit cycle inside the inherent stable subset $\mathcal{E}$ when the robot is walking. For each undisturbed step, the robot's state continues along this path. In case of a disturbance, the state will exit the limit cycle and there will be a trajectory in the feature space that is heading towards the border of the viability kernel. The course of the trajectory is dependent upon the kind of disturbance and the degree of instability. The classification of these trajectories can therefore be used to switch the control strategy or to trigger appropriate reflex motions.

It is obvious, even for similar fall situations, that the course of the trajectories is not completely identical and the time needed for the trajectories may differ slightly. To describe

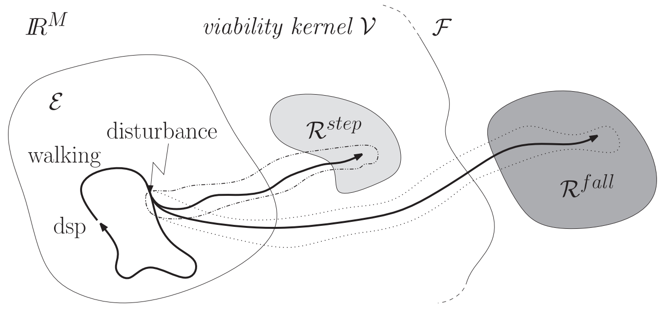

Fig. 7. Different situations such as walking, struggling, falling, etc. have characteristic trajectories in the feature space.

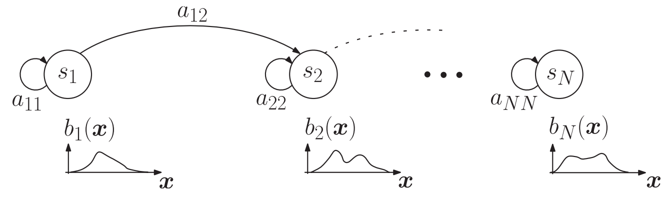

Fig. 8. Linear HMM topology that is used as a signal model to describe the occurring sequence of feature vectors when the robot is in a certain situation. For every state s_i, a continuous emission probability density function $b_i(\vec{x})$ exists.

the typical sequence of feature vectors that occurs in a certain situation, a Hidden-Markov-Model (HMM) is used as signal model. With the HMM it is possible to respect as well as chronological fluctuations of the trajectory as variations in the feature space.

An HMM is a double stochastic process. Basically, it is a discrete time state machine (Figure 8) where the transition from state s_i to state $s_j,\ i, j \in \{1, \ldots, N\}$ represents the first random process. Due to the Markov property, the changeover only depends on the predecessor state s_i and can be described by a transition coefficient a_{ij}. The second stochastic process is the emission of a feature vector $\boldsymbol{x}$. The emission probability distribution for state s_i is denoted $b_i(\boldsymbol{x})$. Multivariate Gaussian mixture models, as described in Equation (6), are used to model $b_i(\boldsymbol{x})$.

The state machine can have different topologies. Due to the distinct progression of a fall we used a linear model that only allows transitions a_{ii} from state s_i to itself or transitions $a_{ij},\ j = i + 1$ i.e. from one state to the next. In the following, HMM are denoted λ_k. For each HMM, the probability $P(\lambda_k|\boldsymbol{O})$ that an observation sequence $\boldsymbol{O}$ has been produced by the HMM can be calculated. An introduction to HMM can be found in Rabiner (1989); Rabiner and Juang (1993); Fink (2008).

In Section 5, the classification was based on the probability $P(\Omega_k|\boldsymbol{x})$ of Ω_k being the best activation set given feature vector $\boldsymbol{x}$. Here we evaluate the probability $P(\lambda_k|\boldsymbol{O})$ of HMM λ_k being the most likely HMM given an observation sequence $\boldsymbol{O}$. Analogous to Equation (5), this can be traced back to:

$$\lambda^* = \underset{\lambda_k}{\operatorname{argmax}} P(\boldsymbol{O}|\lambda_k)P(\lambda_k). \tag{8}$$

Again, the *a priori* probabilities $P(\lambda_k)$ have to be set to a constant value or can be used to disable some reflexes. For the calculation of $P(\boldsymbol{O}|\lambda_k)$ the Viterbi Algorithm is used (Rabiner 1989). This algorithm is based on Dynamic Programming (Bellman 1957) and calculates the most likely path through the HMM given an observation sequence $\boldsymbol{O}$.

7. Training

The parameter estimation for the probabilistic models requires sufficient feature vectors for each HMM and GMM. In this section, the definition of the control strategies (classes) and the generation of the training vectors is explained.

In the following, we set up a basic monitoring system with four reflex steps R_D^{step} for the directions $D \in \{[\text{ahead}],[\text{aback}],[\text{stf}],[\text{swf}]\}$. Additionally, four crouching motions R_D^{fall} are provided to minimize damages if the balance cannot be recovered with the corresponding reflex step. In order to cluster the feature space, several points in $I\!R^M$ have to be surveyed. This is done with impulses of varying intensities and directions that are used to create a representative set of different fall situations.

An impact experiment is termed with $\Psi \in {}_S^I\Psi_C^D$. The index $S \in \{1, 2, \ldots, P\}$ denotes the step phase where the impact is applied, $I \in \{0, \ldots, 35\}$ [Ns] is the impulse intensity and $D \in \{[\text{ahead}],[\text{aback}],[\text{stf}],[\text{swf}]\}$ denominates the impulse direction. Each impact experiment has to be repeated several times. The first training run is used to record the training vectors and to evaluate if the robot can recover without reflex motion. If it cannot recover, further training runs are used to determine the optimal control strategy. To indicate if the robot is forced to execute a reflex, we use the index $C \in \{[\text{no reflex}], [R_D^{\text{step}}]\}$. A function

$$f({}_S^I\Psi_C^D) = \begin{cases} 1 & \text{for robot has not fallen} \\ 0 & \text{for robot has fallen} \end{cases} \tag{9}$$

is used to evaluate the stabilizing success. The sequence of training vectors $\boldsymbol{O} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ that is recorded during the first training experiment ${}_S^I\Psi_{\text{no reflex}}^D$ is labeled as follows:

$$L(\boldsymbol{x}) = \begin{cases} [\mathcal{E}] & \text{for} \quad f({}_S^I\Psi_{\text{no reflex}}^D) = 1 \\ \mathcal{R}_D^{\text{step}} & \text{for} \quad f({}_S^I\Psi_{\text{no reflex}}^D) = 0\ \wedge \\ & \qquad\ f({}_S^I\Psi_{R_D^{\text{step}}}^D) = 1 \\ \mathcal{R}_D^{\text{fall}} & \text{for} \quad f({}_S^I\Psi_{\text{no reflex}}^D) = 0\ \wedge \\ & \qquad\ f({}_S^I\Psi_{R_D^{\text{step}}}^D) = 0. \end{cases} \tag{10}$$

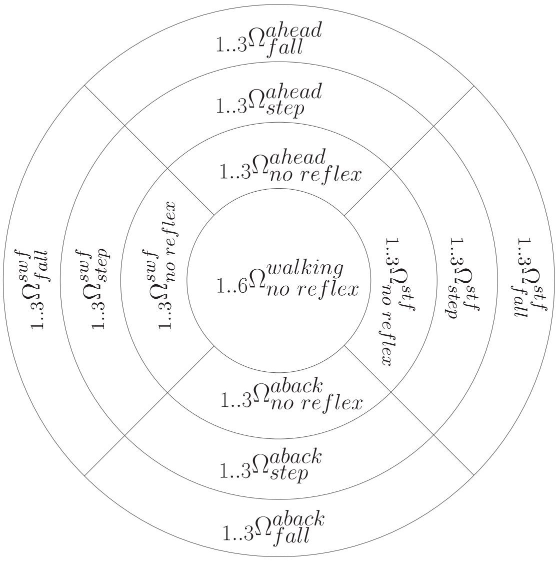
$1..3\Omega^{ahead}_{fall}$
$1..3\Omega^{ahead}_{step}$
$1..3\Omega^{ahead}_{no\ reflex}$
$1..3\Omega^{swf}_{fall}$
$1..3\Omega^{swf}_{step}$
$1..3\Omega^{swf}_{no\ reflex}$
$1..6\Omega^{walking}_{no\ reflex}$
$1..3\Omega^{stf}_{no\ reflex}$
$1..3\Omega^{stf}_{step}$
$1..3\Omega^{stf}_{fall}$
$1..3\Omega^{aback}_{no\ reflex}$
$1..3\Omega^{aback}_{step}$
$1..3\Omega^{aback}_{fall}$

Fig. 9. Activation sets Ω_k that were set up for the basic monitoring system. In further experiments, the number of sets was increased by more specialized control strategies, e.g. for tripping (Section 8). Optimal reflex motions were designed for three step phases (swf lift off, middle step phase, swf settle down).

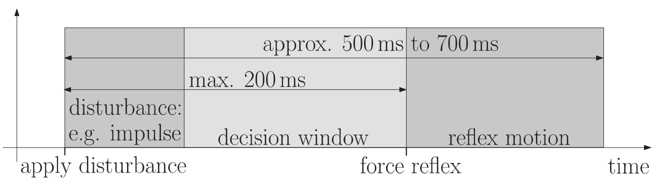

Fig. 10. Progress of a typical fall experiment. We aim to classify the robot's state during the decision window.

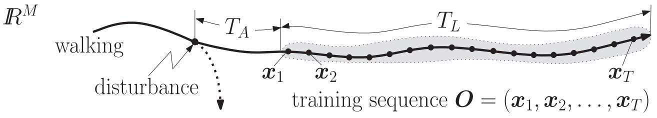
$I\!R^M$
walking
disturbance
T_A
T_L
$\boldsymbol{x}_1$
$\boldsymbol{x}_2$
$\boldsymbol{x}_T$
training sequence $\boldsymbol{O} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T)$

Fig. 11. Training sequence for an HMM.

$L(\boldsymbol{x})$ becomes more complicated if there are more control strategies that are appropriate for one scenario. In the case of multiple-fall reflexes, it could also be useful to consider the impact energy when the robot hits the ground. The classes for the basic monitoring system are visualized in Figure 9. To distinguish the exceptional states from normal operations it is necessary to set up activation sets for the undisturbed walk as for all other tasks the robot is supposed to perform. The basic system is intended to distinguish between six phases during one undisturbed step. This leads to the six activation sets

$$_{1,\ldots,6}\Omega^{\text{walking}}_{\text{no reflex}}.$$

With it, the basic monitoring system consists of 42 classes altogether.

To minimize the risk of damages, the training is not performed with our real robot. Instead, a multi-body dynamics simulation program (Höhn et al. 2005) is used. The simulation model is equivalent to our 6-DoF (degrees of freedom) bipedal robot BARt with the exception of two additional active joints in the hip and in the ankle of each leg. Due to the extended degrees of freedom, this model also allows us to examine sidewards motions. With some restrictions, the results can still be verified with the real biped (Section 8.3). In Figure 10, the progress of a typical fall simulation can be seen. Approximately 50–100 ms after the impact has abated, the robot is forced to execute the designated reflex step R_D^{step} for the examined fall situation.

Between disturbance and reflex execution the so-called decision window is situated. It is desired that the monitoring system switches to the appropriate control strategy within this time period. This ensures that there is enough time left for the execution of a reflex motion. Opposite to a common classification problem, the balance monitoring is highly time-dependent since a preventable fall is not any longer preventable if it is classified too late. For this reason, the right choice of training vectors is very important for the performance of the monitoring system. If the vectors are chosen too late, a fall would not be detected in time. On the other hand, if the vectors are gathered during the impulse period, the algorithm would be too specialized. This would result in only those situations where falls were caused by impulses being detected. Moreover, different fall situations cannot be separated robustly until approximately 100 ms after the robot has started to lose its balance.

The feature vectors that were recorded during the decision window are used to estimate the GMM parameters. This is done with the iterative expectation maximization algorithm (Durbin et al. 1998; Bishop 2006). The parameterization of the HMM is done with the well-known Baum–Welch Algorithm (Rabiner 1989). For numerical reasons, it is important that the training sequences $\boldsymbol{O} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ for the HMM are not too short. Additionally, there should be significant differences within each sequence that can be associated with the HMM states. For this reason, it can be advantageous to consider the features within the impulse period as well. As can be seen in Figure 11, there are two parameters that can be used to influence the training sequences. In our experiments, good results were achieved for a sequence offset of $T_A = 50$ ms and a sequence length of $T_L \approx 200 - 250$ ms.

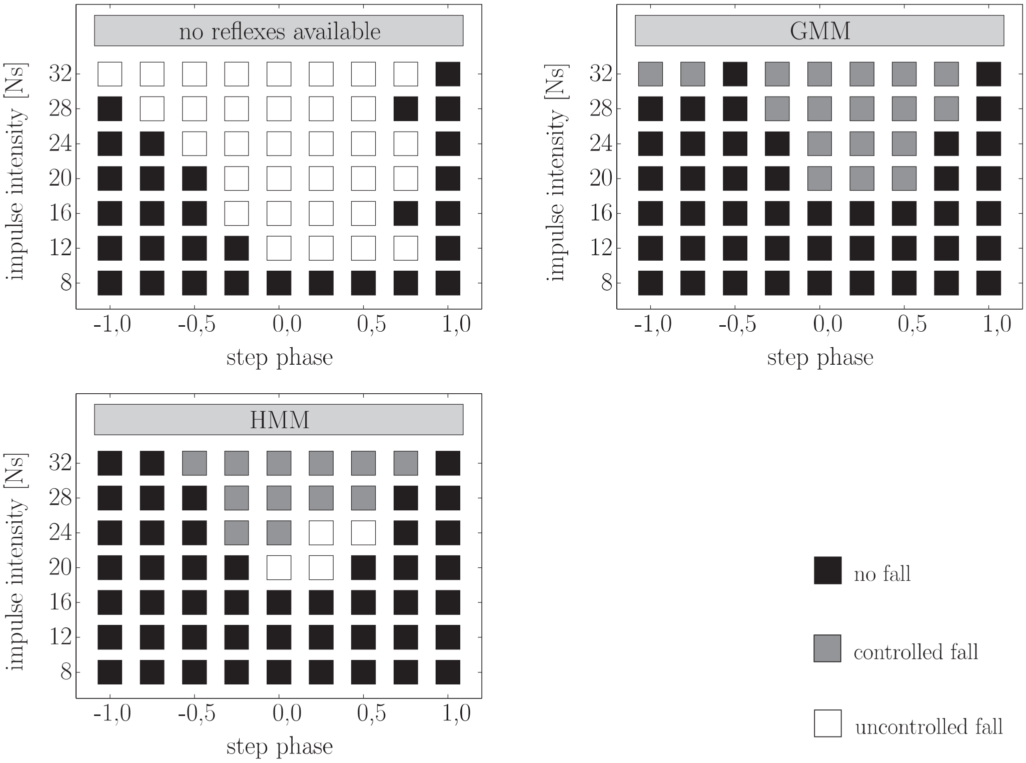

Fig. 12. The robot's stabilizing abilities during several punch experiments. At different step phases (–1: swing foot lift off, 0: middle step phase, 1: swing foot settle down), impulses of variable intensity were applied to the back of the walking robot. The reflexes were triggered by the proposed monitoring system.

8. Results

In this section, the robot's behavior with the proposed monitoring systems and the reflex motions is analyzed. Two simulated benchmarks and a verification using BARt are presented.

8.1. Impulse Benchmark

The results of the first benchmark can be seen in Figure 12. Unstable situations were provoked by impulses of different intensities that were applied at several step phases to the back of the torso (Figure 13). As could be expected, the robot is inherently stable (Section 3) near the double support phase. If there are stronger punches or the robot is in the middle step phase, falls cannot be prevented without an appropriate control strategy. With reflex steps and a GMM- or HMM-based monitoring system, significantly more falls could be averted. Moreover, the GMM monitoring system initiated adequate crouching motions in all remaining situations so that no uncontrolled falls occurred.

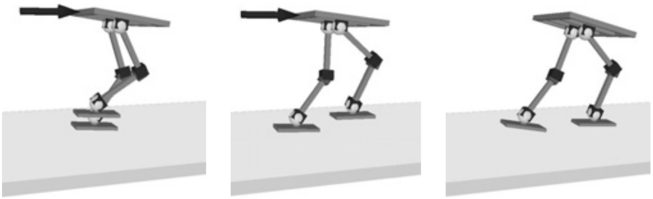

Fig. 13. Unstable situation caused by an impulse to the back of the robot's torso.

The HMM system was seen to be over-optimistic. This meant that in some situations, reflex steps were initiated even though a fall could no longer be prevented. The classification performance of the HMM can be affected e.g. by resizing and shifting the training window with the parameters T_A and T_L (Section 7). When applying the monitoring system to the real robot, it is possible to preset the *a priori* probabilities $P(\Omega_k)$. By making a fall more likely than a situation where a step is appropriate, it is possible to adjust the borderline between reflex crouching and reflex stepping motions. This is preferred since

Table 1. Benchmark results.

	GMM (%)	HMM (%)	HMM*(%)
No fall	36	29	57
Controlled fall	28	50	29
Uncontrolled	36	21	14
Unnecessary reflex	14	7	7

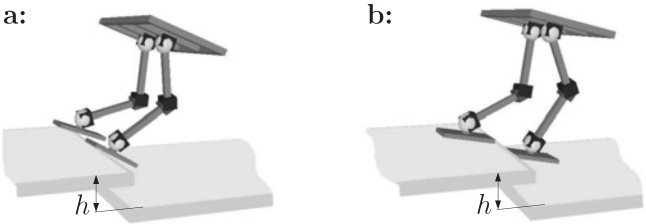

Fig. 14. Falling down a step: (a) the swing foot is placed on the edge of the step and (b) the robot steps onto the lower platform.

it is better to execute a controlled fall than to attempt a reflex in vain.

8.2. Falling Down a Step

Table 1 shows the results of another simulated benchmark experiment where a fall was provoked by a step placed in the robot's way (Figure 14). The GMM and HMM models were trained with simulated impulses as described in Section 7. Therefore, no additional activation sets for stumbling exist. The HMM* system conforms to the HMM monitoring system, but additionally contains control strategies for falling down stairs. The values in Table 1 are averaged over several simulation passes. The step height was varied from $h = 2\,\text{cm}$ to $h = 14\,\text{cm}$. Additionally, the position of the step was altered.

As can be seen, the uncontrolled falls could be reduced to 36% with the GMM approach, 21% with the HMM approach and 14% with the HMM* approach. An uncontrolled fall occurred when the robot was falling down and the appropriate crouching motion had not been initiated. In other words, despite a reflex step the fall could no longer be prevented or else no reaction was executed at all. Moreover, it can be seen that the fall avoidance rate with the HMM* system increased to 57%. This benchmark shows that fall situations not caused by impulses are also detected reliably with the GMM as with the HMM approach. The kind of disturbance that leads to a fall therefore plays a relatively minor roll.

8.3. Verification with the Real Biped

Figure 15 shows a fall experiment with our bipedal robot BARt. A swinging weight of 3 kg mass was used to apply an impulse to the back of the walking robot. The occurring sensor data was recorded and used for an offline comparison of the classification approaches afterwards. In the lower half of Figure 15, the results of an HMM classifier with three states per model and a GMM classifier with three mixtures per model can be seen. For the HMM classification, the feature vector was transformed with a Karhunen–Loève transformation whereby the dimension was reduced to $M' = 2$.

Near the double support phase (dsp), a slightly unstable situation (direction: ahead, intensity: ok, no reaction necessary) is detected, because the robot falls on its swing foot at the end of each step. In the period between dsp and impulse, the HMM-based algorithm robustly classifies the step phases 1 to 4. The activation sets $\Omega_{\text{no fall}}$, that indicate unstable situations where the robot is still inherently stable and no reflex is needed, are situated close together in the feature space. As can be seen, these sets cannot be separated from the GMM classifier very well.

However, this does not influence the performance in detecting fall situations. A few milliseconds after the swinging weight hits the robot a slightly unstable situation (direction: ahead) is classified. About 50 ms later, the fall is classified as unpreventable with the GMM approach. Since a reaction delay of $T_V \sim 50\,\text{ms}$ is needed for a robust classification with the GMM (Section 5) the crouching motion would have been initiated 100–150 ms after the impact. As expected, the classification with the HMM takes a little bit longer (≈150–200 ms) but is still sufficiently fast. From 2.7 s onwards, the classification results allow no more statement since the robot is already falling down.

9. Summary and Outlook

Two probabilistic approaches are proposed which allow the observation of the balance of humanoid robots. The first algorithm uses Gaussian-Mixture-Models (GMMs) to classify the robots's state. The second is based on Hidden-Markov-Models (HMMs). With both approaches, abnormalities from regular operations can be detected and evaluated by a probability ratio. A learning process is required for the determination of the distribution densities and HMM parameters. The training data is gathered with help of a simulation program. This facilitates the training of exceptional situations which should not be executed with the real robot. In simulated experiments, the algorithms separated different situations robustly (Figure 12). Even although the algorithms were parameterized with simulated scenarios, they can be applied to the real biped. As far as BARt's limited degrees of freedom permitted, their performance could be verified successfully.

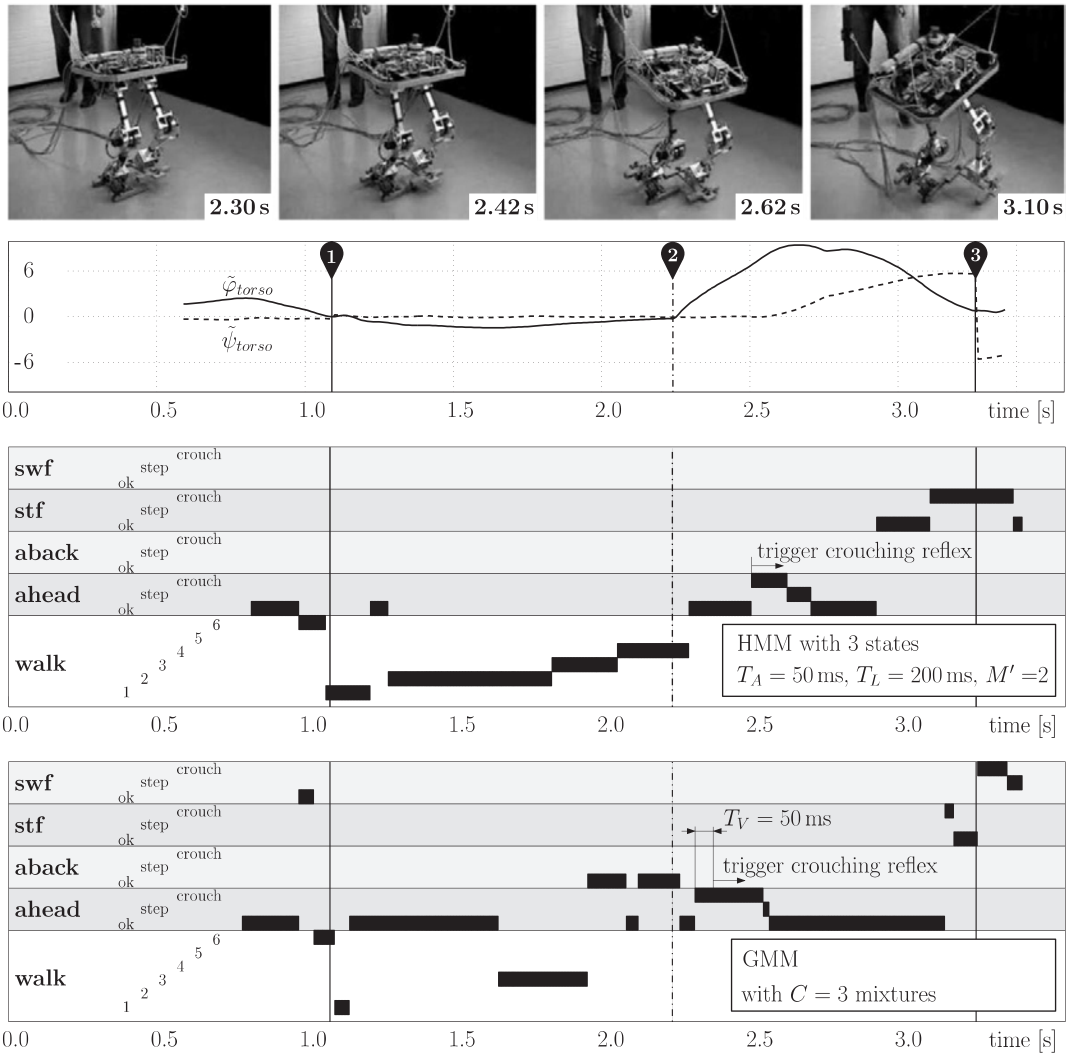

Fig. 15. Fall experiment with BARt. A swinging weight is used to punch the robot. The marks 1 and 3 represent the double support phase and mark 2 indicates the impact time. The chronological sequences of two features are shown. The features $\tilde{\varphi}_{\text{torso}}$ and $\tilde{\psi}_{\text{torso}}$ are normalized with their standard deviation ($\tilde{\varphi}_{\text{torso}} = \varphi_{\text{torso}}/\sigma_{\varphi}$). The results with the HMM and GMM classifiers are displayed in the lower half of the figure. For every sampling instance, the most likely class is marked. The classes correspond to the activation sets that are visualized in Figure 9.

During the experiments, the feature vectors were sampled every 10 ms. The time needed to calculate the proposed GMM and HMM algorithms took considerably less than a millisecond on a desktop PC, which was also calculating the dynamical model of the robot. Hence, a realtime calculation with the robot's microcontroller is possible. Since the calculation effort for the HMM algorithm is significantly higher than for the GMM approach, a combination of both methods could be of interest. A fast pre-classification could be performed with the GMM models. If the probability of a fall exceeds a given threshold, the HMM model could be used to classify the situation more precisely.

In addition to the possibility of analyzing risky situations, the combination of a simulation environment and the proposed monitoring system offers further prospects. When the robot is switched off at night the simulation tool could be used to optimize the monitoring system. HMMs or GMMs for new tasks that were taught to the robot by day could be trained and measurement data gathered from the real biped could be incorporated. Furthermore, it would be possible to optimize the reflex motions.

Tumbling is not only a problem in the domain of bipedal robots but also in many fields of medicine. Research into falls, especially in geriatric medicine, is a current issue. When elderly persons experience a fall it is very likely that they will not be able to walk again. Being bedridden or chairbound causes serious health impairments and is a grave loss of quality of life. Especially affected are geriatric persons with leg prostheses, since an aggravating factor is their limited movement.

Besides the known passive devices such as hip protectors (Robonovitch et al. 1995), a research topic is active impact-absorbing mechanisms similar to airbags. Active devices demand a balance surveillance system that provides a trigger signal. The proposed algorithms are also well suited for these demands. The same problem as with humanoid robots occurs, since fall experiments must not be executed with elderly persons. Once again, a feasibility is to gather the training data in a simulation environment and to use sensor data recorded during the person's everyday life. We are presently in contact with an international medical technology company with the prospect of cooperating in this field. A future application of the results could therefore be an active injury protection device or an intelligent prostheses for geriatric amputees.

Acknowledgements

This work emerged in the framework of the project GE 451 founded by the German Research Foundation (DFG). The authors would like to thank the DFG for the support.

Appendix: Index to Multimedia Extensions

The multimedia extensions for this article can be found online by following the hyperlinks from www.ijrr.org.

Extension	Media type	Description
1	Image	Strobe picture of the falling-down sequence shown in Figure 1
2	Image	Falling-down sequence
3	Video	Video of a statical stable uprighting motion
4	Video	Video of a dynamical get-up motion

References

Albert, A. (2002). *Intelligente Bahnplanung und Regelung für einen autonomen, zweibeinigen Roboter.* VDI Verlag, Düsseldorf.

Bellman, R. E. (1957). *Dynamic Programming.* Princeton University Press, Princeton, New Jersey.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer, Science + Business Media, LLC, New York.

Bortolami, S., DiZio, P., Rabin, E., and Lackner, J. (2003). Analysis of human postural responses to recoverable falls. *Experimental Brain Research,* 151: 387–404.

Durbin, R., Eddy, S., Krogh, A., and Mitchinson, G. (1998). *Biological sequence analysis – Probabilistic models of proteins and nucleic acids.* Cambridge University Press, Cambridge.

Fink, G. A. (2008). *Markov Models for Pattern Recognition.* Springer, Berlin.

Fujiwara, K., Kajita, S., Harada, K., Kaneko, K., Morisawa, M., Kanehiro, F., and Hirukawa, S. N. H. (2006). Towards an optimal falling motion for a humanoid robot. *Humanoids,*: 524–529.

Fujiwara, K., Kanehiro, F., Kajita, S., Kaneko, K., Yokoi, K., and Hirukawa, H. (2002). UKEMI: Falling motion control to minimize damage to biped humanoid robot. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland.*

Fujiwara, K., Kanehiro, F., Kajita, S., Yokoi, K., Saito, H., Harada, K., Kaneko, K., and Hirukawa, H. (2003). The first human-size humanoid that can fall over safely and stand-up again. In *Proceedings of the 2003 International Conference on Robots and Systems, Las Vegas, Nevada,* Volume 2, pp. 27–31.

Fujiwara, K., Kanehiro, F., Saito, H., Kajita, S., Harada, K., and Hirukawa, H. (2004). Falling motion control of a humanoid robot trained by virtual supplementary tests. In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation, New Orleans, LA,* pp. 1077–1082.

Fukunaga, K. (1972). *Introduction to Statistical Pattern Recognition.* Academic Press, Inc., New York.

Goswami, A. (1999). Foot rotation indicator (FRI) point: A new gait planning tool to evaluate postural stability of biped robots. In *Proceedings of the 1999 IEEE International Conference on Robotics & Automation, Detroit, Michigan,* pp. 47–52.

Höhn, O., Gačnik, J., and Gerth, W. (2005). Detection and classification of posture instabilities of bipedal robots. In *Proceedings of the 8th International Conference on Climbing and Walking Robots, London, UK,* pp. 409–416. Springer.

Höhn, O., Schollmeyer, M., and Gerth, W. (2004). Sturzvermeidung von zweibeinigen Robotern durch reflexartige Reaktionen. In *Eingebettete Systeme,* Informatik aktuell, pp. 60–69. Springer.

Morisawa, M., Kajita, S., Harada, K., Kaneko, K., Kanehiro, F., Fujiwara, K., and Hirukawa, H. (2005). Emergency stop algorithm for walking humanoid robots. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 2109–2115.

Popovic, M. B., Goswami, A., and Herr, H. (2005). Ground reference points in legged locomotion: Definitions, biological trajectories and control implications. *The International Journal of Robotics Research,* 24(12): 1013–1032.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition.* Prentice Hall, Inc, Englewood Cliffs, New Jersey.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE,* 77(2): 257–285.

Robonovitch, S. N., Hayes, W. C., and McMahon, T. A. (1995). Energy-shunting hip padding system attenuates femoral impact force in a simulated fall. *Journal of Biomechanical Engineering,* 117: 409–413.

Strasser, R., Seebode, M., and Gerth, W. (2003). A very small low cost Inertial Measurement Unit (IMU) for robotic applications. *Symposium Gyro Technology,*: 18.0–18.9.

Vukobratović, M. and Borovac, B. (2004). Zero-Moment Point – Thirty Five Years of its Life. *International Journal of Humanoid Robotics,* 1(1): 157–173.

Wieber, P. B. (2002). On the stability of walking systems. In *Proceedings of the International Workshop on Humanoid and Human Friendly Robotics*, pp. 53–59, Tsukuba, Japan.