Biologically Inspired Posture Recognition and Posture Change Detection for Humanoid Robots

Michael Spranger, Sebastian Höfer and Manfred Hild

Abstract— **This paper presents a biologically inspired approach to posture recognition and posture change detection for a biped robot. Slow Feature Analysis, an algorithm developed by theoretical biologists for extracting slowly changing signals from signals varying on a fast time scale, is applied to the problem of recognizing the posture of biped humanoid robots over time and successively on the recognition of the change of posture. Both the recognition of basic static postures, like lying and standing, of peer robots via visual sensory information and the recognition of the same postures via internal proprioceptive sensors are considered. Given promising results in this domain we extend the application of the method onto the dynamic domain of detecting the change of posture, specifically we show the utility of the algorithm for detecting when a robot falls.**

I. INTRODUCTION

In recent years there has been a growing interest in the robotics community to build robots that can interact with humans in "natural" environments in a human-like fashion. As of today robots in robot-human interaction are usually seen as inferior with regards to basic human motion capabilities, requiring in many cases that the human adopts herself to the need of the robot or, even more, makes humans constrain the environment to be suitable for the particular robot platform. However, to truly integrate robots in non-augmented human environments one has to equip them with human-like capabilities, which evidently not only includes high level cognitive functions, such as language, categorization or social behavior, but also basal motion and perceptual competence.

One such basic capacity is the recognition of basic postures, be it intrinsically, when the robot has to decipher his own posture, or extrinsic, when interpreting the postural state of robots or agents in the environment. In order for humanoid robots to aid and support humans in their daily life they need to be aware of their posture or the posture of others. Humans are without doubt excellent in recognizing postures, which is due to the ubiquity of basic postures such as sit, stand and lie in the interaction with the environment. Postures are a wide field, we just want to point out one particularly interesting linguistic phenomenon highlighting their significance. In some languages postures underlie the entire conceptualization of space and are even extended metaphorically into time and abstract domains (see [4] for a case study involving robots).

M. Spranger is with Sony CSL Paris, 6 rue Amyot, 75005 Paris, France spranger@csl.sony.fr

S. Höfer and M. Hild are with the Neurobotics Research Laboratory, Computer Science Department, Humboldt-Universität zu Berlin, Berlin, Germany [hoefer|hild]@informatik.hu-berlin.de

Another human capacity, even more entrenched with the sensorimotor control loops driving humans incredible biped walking competence, is that of posture change detection. Humans are astonishingly good at detecting even slightest perturbations in walking gait patterns and can correct their movements accordingly when tripping or when the ground changes. When all correction reflexes fail we quickly adopt a safety posture and we do so largely without conscious awareness, i.e. reflexes like extending arms to dampen the collision are executed after short reaction times.

This paper reports on recent progress in using a biologically inspired algorithms on these two tasks of basic human motion control. The algorithm called Slow Feature Analysis is first applied to the problem of statically recognizing postures. We analyze the performance of the algorithm given visual and proprioceptive stimuli. Given promising results from the application of the algorithm to the static recognition problem, we supplement the investigation and report on first results extending the usage to detecting the change of posture, that is the change from upright position to vertical position, which is particularly interesting for stabilizing walking gait patterns.

Slow Feature Analysis is an algorithm developed in a group of theoretical biologists headed by Laurenz Wiskott and is optimal for extracting slow varying parameters from time series data. The algorithm belongs to a family of algorithms that try to extract signals that change slowly over time, that still carry a maximum of information from fast changing source signals. While SFA has been applied to the domain of vision and, for instance place cells in the hippocampus, its potential for mobile robotics has been left largely unexplored. We demonstrate the applicability of the algorithm to autonomous robot systems research.

We start by introducing the particular algorithm used in this paper, followed by introduction to the specific robot platform used to test the algorithm. The subsequent sections introduce the approach taken for the particular domains of posture recognition and posture change detection. We conclude by discussing the outlook for using the algorithm in real-time settings and will compare the approach proposed here briefly to existing literature.

A. Slow Feature Analysis

Slow feature analysis belongs to a class of unsupervised learning algorithms that try to solve a particular optimization problem related to temporal slowness (see [8] for the initial publication and [10] for a detailed analysis). The problem is formulated as follows: [1] Given a potentially multidimensional input signal $\mathbf{x}(t) = [x_1, .., x_N]^T$ the algorithm identifies scalar functions $g_j(x)$, $j \in J$ that determine the output of the system $g_j(\mathbf{x}(t)) = y_j(t)$, where y_j are the output functions over time. The process of finding the g_j real valued functions is governed by a set of constraints that capture the essence of the problem that SFA tries to solve. The algorithm tries to find a set of functions g_j such that for the output functions y_j the following constraint holds

$$\Delta\langle y_j \rangle \quad := \quad \langle \dot{y_j}^2 \rangle_t \qquad \text{is minimal} \tag{1}$$

where $\langle \cdot \rangle_t$ signifies the average over time and $\dot{y}$ is the derivative of y. This constraints specifies the actual learning problem. Namely, we are searching for the slowest varying output signal. $\Delta\langle y_j \rangle$ is minimal if y_j is not changing rapidly, but varies leisurely over time. Since every constant function would easily fulfill this restriction, three additional requirements are formulated.

$$\langle y_j \rangle_t \quad = \quad 0 \qquad \text{(zero mean)} \tag{2}$$

$$\langle y_j^2 \rangle_t \quad = \quad 1 \qquad \text{(unit variance)} \tag{3}$$

$$\forall i < j \langle y_i y_j \rangle_t \quad = \quad 0 \qquad \text{(decorrelation, ordering)} \tag{4}$$

Equations 2 to 4 constrain the possible set of solution functions to a meaningful subset of all functions satisfying 1. Especially Equations 3 and 4 are important. Equation 3 forces the output signal to carry information. Equation 4 induces an ordering on the output signals: the slowest signal should be first. But more importantly, it requires output signals to be decorrelated. Different aspects of the input are coded for by different output signals [1].

Slow Feature Analysis provides a solution to learning the real valued functions g_j by proposing a sequence of processing stages[1]. When learning the functions g_j, sometimes also called transfer functions, the algorithm operates on a given time series data set and estimates a set of parameters such that later in application the slowest varying signal is determined. Let the input signal be $\mathbf{x} = [x_1, .., x_N]^T$, where N is the dimensionality of the input. And let $\mathbf{g} = [g_1(\mathbf{x}), ..., g_J(\mathbf{x})]^T$ be the transfer functions, where each $g_j(\mathbf{x}) := w_{jk} h_k(\mathbf{x})$ is defined as the weighted sum of some functions h_k.

Since the optimization problem is in general hard to solve, the set of possible solutions is further constrained by asserting that the g_j transfer functions are linear combinations of a finite set of basis functions. In order to solve the optimization problem linearly, the input signal is first expanded given an appropriate set of basis functions $\mathbf{h} = [h_1, .., h_k]^T$ with $k \in K$. Typically monomials are chosen as basis functions, because they serve as a basis for the vector space of polynomials or at least some finite dimensional subset of that vector space. Since we are only interested in computations that conclude in a finite amount of time, monomials up to a certain order are used to nonlinearly expand the input signal.

[1]We will focus in the following on a specific linear variant of SFA, because it is the most useful when dealing with real world data. Additional variants exist and are mentioned in [8], [10], [9].

The expansion basis is given by the system designer, leaving the weights to be learned. Parameters are learned on a training input signal called $\tilde{x}$ (same dimensions as $\mathbf{x}$), which should share some of the properties of the target input signal, since the characteristics of the signal will lead to a specific set of parameters that will only generalize well if learned on sufficiently rich training data. First, the training input signal is normalized to obtain a zero mean, unit variance signal. The resulting signal is expanded given the set of nonlinear basis functions $\mathbf{h}$, which yields $\tilde{\mathbf{z}}(t) := \mathbf{h}(\tilde{\mathbf{x}}(t))$. The expanded signal is sphered (sometimes also called whitened), an affine transformation, which yields $z(t) := S\tilde{z}(t)$, where S is the sphering matrix, such that $\langle z \rangle = 0$ and $\langle zz^T \rangle = I$ (z has zero mean and $\mathrm{cov}(\mathbf{z}) = I$). S is usually obtained using Principal Component Analysis and, therefore, dependent on the training set. Finally, PCA is also applied to the time derivative of the sphered signal. However, ordering the computed eigenvalues λ_j and their corresponding eigenvectors, not like usually done when computing the PCA biggest first, but the other way around, gives us the parameters w_{jk} for the transfer functions $g_j(\mathbf{x}) = w_j^T \mathbf{h}(\mathbf{x})$, with $\lambda_j w_j = \langle \dot{z}\dot{z}^T \rangle w_j$, where λ_j is one of the eigenvalues of the covariance matrix of $\dot{z}$ and $\lambda_1 \leq ... \leq \lambda_J$. It can be shown that the w_{jk} fulfill the constraints posed in the original optimization problem. Notice how PCA is employed to find the smallest eigenvalues. While PCA orders the eigenvalues of the covariance matrix decreasingly, we here search for the slowest varying signal, that is the signal with the smallest time derivative, hence the smallest eigenvalue.

Since the input signal (which might already be from a high dimensional input space) is further extended in dimensionality by the basis functions, SFA does suffer heavily from the curse of dimensionality. There are two principal ways to deal with the explosion in dimensionality. First, Slow Feature Analysis can be applied successively in networks of SFA nodes, where every node is described by the nonlinear expansion basis (equal for all nodes), plus the set of linear transfer functions g and their weights w. In such a scheme, the SFA algorithm is applied first to an input signal, but than the output of the SFA component is fed to other SFA units performing the same algorithm with possibly different estimated parameters based on the already filtered signal and so on and so forth. Another way of dealing with high dimensional input data is by reducing dimensionality of the input space, for instance by applying PCA before applying SFA. Notice that, applying PCA before applying SFA does not necessarily harm the outcome of SFA, as PCA does not operate on local phenomena (like the derivative) but tries to account for highest variability, a time independent measure of a signal (i.e. the order of data points does not matter).

B. Embodiment

In the next sections we show how we applied SFA to sensori data streams stemming from a humanoid robot platform, called A-series. The platform was specifically developed for researching basic motion capacities, most importantly biped walking, which as of today is still arguably an unsolved task.

The robot is based on a commercially available robot kit, called Bioloid. The kit was augmented by adding processing power, a camera and additional sensors. A PDA computer attached to the back processes visual information inflowing from the camera. Eight microprocessor boards, each featuring a two-axes acceleration sensor, are distributed across the body for actuator control (boards are located on the hips, arms and shoulders). Each board controls up to two actuators, while communicating via a shared system bus, that connects the boards with the PDA and allows for blackboard style sharing of information. The robot features 21 degrees of freedom, 19 in the body, including elbow, hand, hip, knee and foot joints, as well as motors driving the pan tilt unit for the camera.

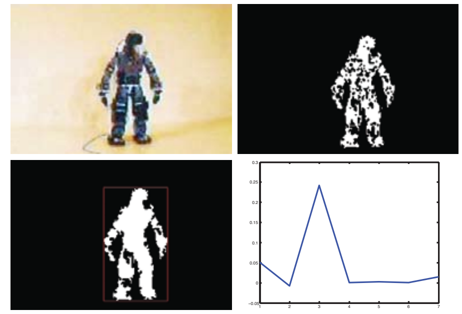

Fig. 1. Extraction of visual features. Top left: an original image captured by an onboard camera of a robot. Top right: foreground/background subtracted image. Bottom left: the connected component processing unit has identified a single connected area, depicted by the bounding box. Bottom right: seven normalized and centralized image moments, visual features computed for the connected region, shown as a parallel plot (see [3] for a detailed description of moments in image processing).

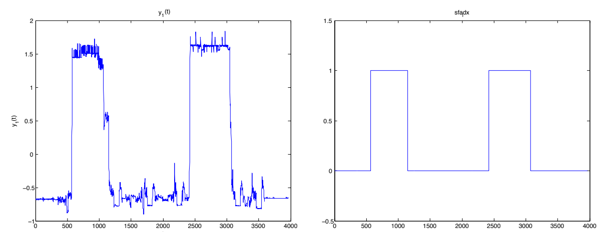

Fig. 3. Ground truth signal and result signal from quadratic Slow Feature Analysis on the complete input data stream (dimension reduced via a prior application of PCA). Left: slowest signal extracted by SFA, when applied to the complete 86 dimensional input signal. Right: ground truth signal, the value zero means the robot is standing or in an upright position walking and gesticulating with its arms, one means the robot is lying either on its back or on its front. Without need for further evaluation the point is conveyed, that indeed SFA extracts a signal that very precisely codes for the posture of the robot. When applying simple thresholding to the slowest SFA result signal, for example everything below zero, is one class of activity and everything above zero another, we clearly see the correspondence between the SFA generated signal and ground truth.

II. POSTURE RECOGNITION

For the experiments reported in this section we equipped the robots with a basic vision system segmenting the environment and with basal motion capabilities, such as walking, standing, getting up after falling and so forth. Moreover, robots are given motor control programs for performing various dance-like arm movements. While motions are performed, robots collect sensori data streams in real-time. These streams consist of proprioceptive data (acceleration sensor values, controlled and sensed motor positions as well as torque values), as well as exteroceptive data from the vision system (see Figure 4 for an overview and a graph of the time series data used in this section),

We hypothesize that the ensemble of proprioceptive data and visual data can be used to identify basic postures or pivotal poses of robots, most importantly standing and lying, or more generally speaking upright and horizontal position, and that Slow Feature Analysis is useful in extracting semantically relevant signals, i.e. the slowest varying signal codes for the basic posture of the robot. To show the correctness of the hypothesis and study the effect of the algorithm, we apply SFA to parts and the whole of recorded sensori data streams. In the next section we first present our approach to implementing SFA for the particular task of posture recognition and subsequently show the results.

A. Algorithm and Data

The complete set of data recorded by the robot is an 86 dimensional signal, including all proprioceptive measurements and exteroceptive data. Exteroceptive data stems from a vision system segmenting and tracking robots in the environment based on visual data from the camera of the robot. The system extracts a set of scale and translation invariant global shape description features for all objects that have not been in the environment before (see Figure 1 for an overview of processing steps, as well as [4] and [7] for a more detailed description, which has been omitted here for space constraints).

We apply quadratic SFA, which means that the nonlinear basis functions are all combinations of variables of length two, plus all input dimensions squared, plus all original signals. Because this leads to a huge increase in dimensionality, we apply PCA before applying nonlinear expansion and only consider the transformed dimensions that together account for 80% of variability in the input signal.

The algorithm operates on a sequence of data measured while the robot is performing different motions, e.g. walking and arm movements. The sequence has a total length of approximately eight minutes, given an average camera frame rate of eight frames per second, we recorded approx. 4000 frames. The robot trips and falls at two points in the sequence, lies and gets up again after some time. While performing these actions, the robot is watched by another robot that executes the vision system just described. The two data streams, the one from the proprioceptive sensors on the performing robot and the visual feature stream extracted by the observing robot are time aligned and recorded together. Since the proprioceptive sensors are updated on a much smaller time scale then the camera captures images, inflowing proprioceptive data is subsampled and time aligned

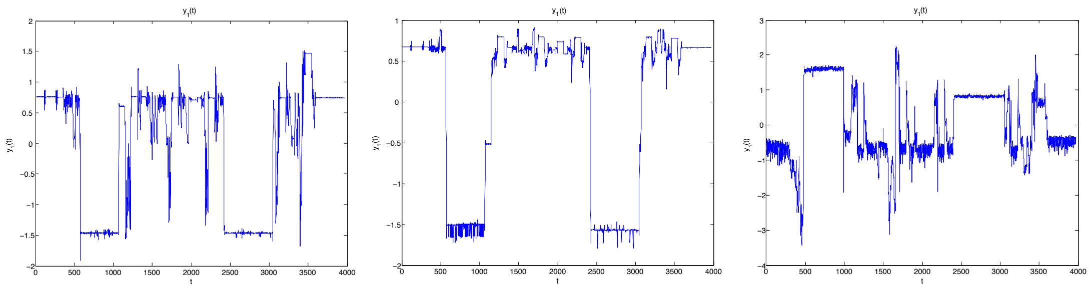

Fig. 2. SFA resulting slowest signals on subset dimensions of the complete data stream. Left: slowest signal extracted by SFA when applied to acceleration sensors only. Middle: slowest signal extracted by SFA when applied to all proprioceptive sensors Right: signal extracted by SFA when applied to visual features over time. All data is generated by applying SFA on subset of the dimensions available in the original data set (see Figure 4). The graphs show that just looking at certain input dimensions, such as only acceleration sensor data or only visual features, SFA can already extract a meaningful signal (see Figure 3 for ground truth comparison) but jitter and noise are still high. Where as using combinations of data from different sources (middle image showing the result when SFA is applied to all proprioceptive sensors) can yield substantially better performance. The middle image clearly allows for separation of two activity classes (when taking everything above zero as one class and everything below zero as a second class).

with the camera images. All data dimensions are normalized to the range of -1 and 1. A ground truth signal to measure the performance of the algorithm was generated by the experimenter to allow for direct comparison with the output of SFA (see Figure 3).

For the experiment reported in this section we compute the parameters of the transfer functions $\mathbf{g}$ on the complete set of recorded data and reapply the learned weights, as well as the preprocessing PCA transformation matrix onto the training data. To study the influence of different dimensions we also applied a single quadratic SFA node to subsets of all input dimensions. Two subsets were of special interest: first, all data stemming from the 8 acceleration sensors (in total 16 dimensions because of two axes per sensor) and the seven visual feature dimensions. Both were studied in isolation and SFA applied to them as well as to the complete signal.

B. Experimental Results

a) Results for acceleration data only: We first apply the algorithm on acceleration data only, with the idea being that essentially using just motor values does not disambiguate between lying and standing. However, adding information from global position indicators such as accelerations sensors should be sufficient to decipher the posture. Notice that we refrain here from preselecting data channels. All acceleration sensors including the ones situated on the arms and legs are used, which given gesticulating or walking movements show a posture uncorrelated rapid change of values. As Figure 2, shows given even hasty varying changes a meaningful signal can be extracted (see Figure 3 for comparison).

b) Results for visual data only: Next we were interested in exploring the effect of SFA when applied to visual data only. Again a meaningful signal is extracted as the slowest varying signal roughly compatible with the outcome of acceleration only. However jitter is still high mostly due to noise in the input visual features, which are fed unfiltered to the SFA node, resulting in a harsh influence of some wrongly segmented frames due to i.e. segmentation errors which make the features behave discontinuously in time (for a short period, in general the visual features behave quite nicely and postures that are close in motor space are usually also close in visual feature space).

c) Results for complete input dimension data set: Combining all data dimensions is the last experiment we conducted. The result of this application is that combining additional information from different sources obviously works best and can extract a rather clean semantic signal (see Figure 3 for the result and the ground truth signal).

III. POSTURE CHANGE DETECTION

Given promising results when applying Slow Feature Analysis to static posture recognition, we hypothesized that the algorithm might proof helpful when trying to detect the dynamic change of posture. In adaptive and tight sensorimotor control strategies of biped walking, detecting the tripping and falling of the robot is evidently of utmost importance. However, due to the fast changing nature of acceleration sensor values when the robot is moving, detecting such changes is not an easy task. Moreover, since the detection has to be instantaneous classical signal processing strategies like low pass filtering are not applicable. Until the low pass filter has integrated enough information the robot might already be lying on the ground.

To investigate the power of SFA to detect sudden posture changes, we applied SFA to several sequences of acceleration data generated by a neural implementation of a sensorimotor loop controlled walking pattern executed on A-series robots. The gait pattern starts with an oscillation in the coronal plane, initiated by letting the robot move its feet such that it subsequently displaces its weight from one foot to the other. Then, as soon as a sensory threshold is reached, the robot starts moving its feet to the front, beginning to walk. Although the walking pattern is quite stable, robots tend to fall to the ground when walking on surfaces with a high grip, such as carpets or natural surfaces.

In applying SFA we are specifically looking for two types of information: first, we want to extract information about normal changes of the robot's state generated by the coronal and sagittal oscillations of the body due to control asserted by the neural controller. Second, we want to detect abnormal

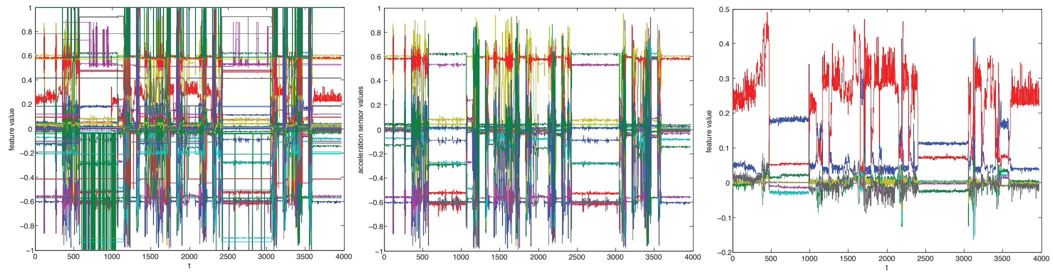

Fig. 4. Original data for posture recognition. Left: original data over 4000 time steps with 86 data dimensions (one signal per dimension) including acceleration sensor data, visual features, driven and measured motor positions as well as torque. Middle: values from acceleration sensors only over the same time series. Right: visual features only over the same time series. All time series (inevitably, since from the same pattern of activity) clearly exhibit periods of relative calmness. These are the periods where the robot has tripped, fallen over and rested on the ground. Essentially, the changes between these periods is what we are interested in detecting using Slow Feature Analysis.

state changes shortly before the robot trips and falls to the ground. Components that reflect such exceptional changes will be used to trigger reflex like correction movements that help prevent the robot from falling.

A. Data

The set of 16 acceleration sensors located on different parts of the robot was used as input data to a quadratic SFA algorithm. Contrary to the previous experiments data was recorded using a frame rate of 50Hz. Data was not aligned and time synced with the much lower frequency camera frame rate. Analyzed sequences had a length of 15-30 seconds and all had the same event structure: the robot would initially stand still for a short period of time, followed by the start of the coronal oscillation, which eventually results in forward motion. After some time of walking, the robot, in all sequences, trips and falls to its back or front.

The same sequences were first used for training and testing. Later, in order to test the generalization potential of the algorithm and, in particular, of learned weights, SFA learning step was applied to a combination of different sequences, which contained sections that included the robot falling to its back and front. The learned SFA weights were then applied to previously unencountered sequences of data of the same robot as well as of other robots of the same type.

B. Algorithm

Best results were achieved when using quadratic SFA repeatedly, passing only a restricted number k of components from one SFA to the next one. k is chosen small to avoid computation time. Experiments showed that for our particular data repeating SFA 5 times yields the best results with respect to the shape (less jitter) of the resulting components (see Figure 5 for comparison).

The choice of the number of components passed to the next round slightly differed when applying learned SFA weights to the training data and the test data. While learning and testing on the same data clear components are obtained when repeating SFA five times and letting $k = 24$. Decreasing the number of iterations or of the passed components leads to noisier extracted components, while an increase does not yield any significant improvement for the particular data.

In order to compare the resulting components the η value (proposed in [10]) was used: $\eta(y) := \frac{T}{2\pi}\sqrt{\Delta(y)}$. We used the correlation coefficient to measure similarity between calculated slowest components and input data.

C. Experimental Results

Figure 5 shows results for data from a short sequence that consists of the robot starting to walk about $\frac{1}{3}$ into the sequence, and a tripping and falling of the robot towards the end. The change in posture when the robot falls are picked up by the slowest varying components for both application patterns of SFA ($y_{1(5)}$ being the slowest component computed by applying SFA five times, and $y_{1(1)}$ resulting from the application of a single SFA node). However, only the single quadratic SFA node application also picks up the transition from the oscillatory phase (the robot is still standing) to the actual walking phase.

Slower components extracted by SFA reflect oscillations in the coronal and sagittal plane. Figure 6 depicts the component with the highest correlation to the coronal acceleration sensor on the shoulder of the robot in comparison with the sensory input. Figure 7 plots the component highly correlated with sagittally oriented acceleration sensor on the robot's shoulder.

The first component potentially useful for detecting the robot's falling, is the eighth slowest signal $y_{8(5)}$ resulting from a sequential five node SFA network (see Figure 8). This signal exhibits a strong peak at $t = 1380$ shortly *before* the slowest components y_1 registers the fall. Another peak shows that the signal picks up the robot's initial step to initiate the walking at $t = 300$. Analyzing the weights w_j calculated by SFA reveals that on average the sensors located on the right foot assert the strongest influence on the extracted signal. More precisely, the ten signals that influence $y_{8(5)}$ most, consist of linear combination of nonlinear expanded monomials with at least one factor being the right foot acceleration sensor, suggesting that at least in the observed sequences the robot falls down because its right foot trajectory collides unfavorably with the ground.

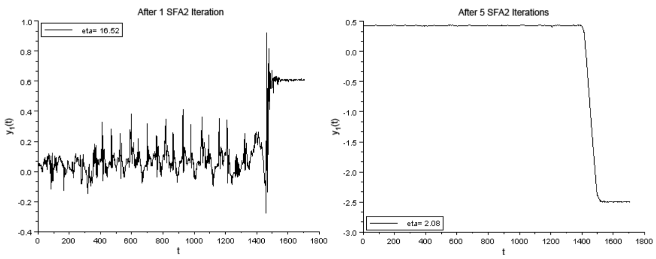

Fig. 5. Slowest varying signals extracted by SFA for different iterations ($k = 24$) on a short walking sequence. Left: signal extracted by a single quadratic SFA node. Right: slowest signal obtained when using a sequential SFA network of five nodes. Notice that the right signal does not change when the robot starts walking at $t = 300$. However, both signals mirror the robot's fall to the ground around $t = 1400$.

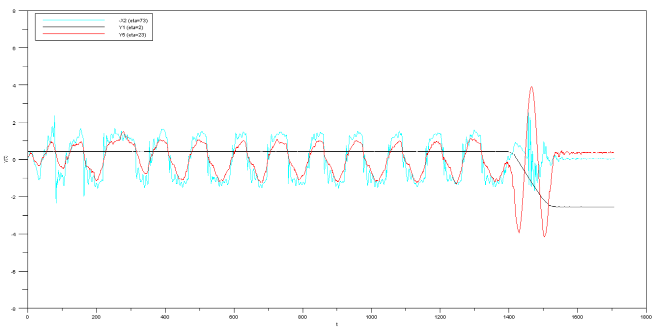

Fig. 6. Slowest varying signal that correlates highest with coronal shoulder acceleration sensor values for single quadratic SFA node y_1 as well as for the five node sequential SFA condition $y_{5(5)}$, as well as sensor input x_2. $y_{5(5)}$ is clearly a smoothed version of the original coronal sensor data.

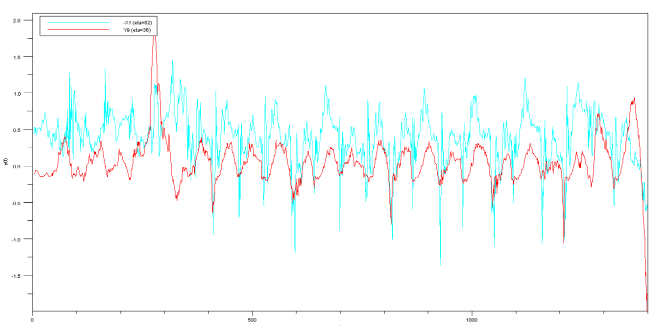

Fig. 7. Magnified section of a highly sagittally correlated slowest component and sagittal shoulder acceleration sensor.

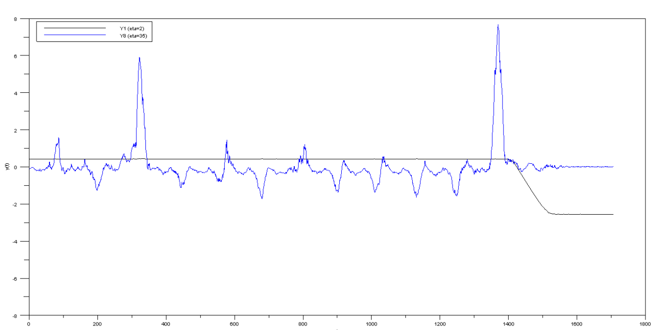

Fig. 8. Component indicating collisions of the right foot with the ground. The right foot sticks to the ground, which is the reason for the successive tripping of the robot.

IV. CONCLUSION AND FUTURE WORK

We have demonstrated how Slow Feature Analysis, a biologically inspired algorithm based on the slowness principle can be used to extract semantically relevant signals for detecting the posture and the change of posture of humanoid robots. Results were compared to ground truth provided by the experimenter and proved to be promising. In fact the results described in this paper are already employed in research modeling the grounded semantics of posture verbs with robots [4]. The results for posture change detection are even more exciting as they seem to open the possibility of stabilizing walk gait patterns in an unsupervised fashion. The unsupervised extraction of such meaningful signals to our knowledge is prior art for this particular field of application.

While previous work has been conducted on supervised classification of motion patterns (especially for human subjects augmented with acceleration sensors), nobody has tried unsupervised classification with robots. Which is largely due to the recency of humanoid robot development, but also due to the dominant paradigm in humanoid motion control, which is based on forward models coupled with no or sparse sensori feedback [2].

The investigations described in this paper are by no means completed. Future work will be concerned with the application of the extracted signals as feedback into the neural walking gait controllers and as a training signal for posture categorization algorithms.

V. ACKNOWLEDGMENTS

This research has been carried out with partial support from the ALEAR project, funded by the EU Cognitive Systems program. The authors would like to thank all members of the Humanoid Team Humboldt who have worked on the hard and software empowering the robot. We are most grateful to Benjamin Werner who provided the neural walk controller.

REFERENCES

[1] P. Berkes and L. Wiskott, Slow feature analysis yields a rich repertoire of complex cell properties, *Journal Of Vision*, vol 5, 2005, pp 579-602

[2] Y. Hurmuzlu, F. Génot and B. Brogliato, Modeling, stability and control of biped robots?A general framework, *Automatica*, vol 40(10), Elsevier, 2004, pp 1647–1664

[3] R. Mukundan and KR. Ramakrishnan, *Moment Functions in Image Analysis: Theory and Applications*, World Scientific, Singapore, 1998

[4] M. Spranger and M. Loetzsch, SIT, STAND, LIE - a computational model of lexicon acquisition, semantics and image schematic extension. In N.A. Taatgen and H. van Rijn (Eds.), *Proceedings of the 31th Annual Conference of the Cognitive Science Society*, (accepted), Cognitive Science Society, Amsterdam, NL, 2009.

[5] M. Spranger, C. Thiele and M. Hild, Integrating high level cognitive systems with sensorimotor control, *Advanced Engineering Informatics*, Elsevier Science, submitted, 2009

[6] L. Steels and M. Spranger, How experience of the body shapes language about space, *in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 09)*, Pasadena, CA, 2009.

[7] L Steels and M. Spranger, The robot in the mirror, *Connection Science*, vol. 20(4), Taylor & Francis, 2008

[8] L. Wiskott Learning invariance manifolds, in L. Niklasson, M. Bodén, and T. Ziemke, (Eds.) *Proceedings of the International Conference on Artificial Neural Networks*, pp. 550-560, Springer, London, 1998

[9] L. Wiskott, Slow feature analysis: A theoretical analysis of optimal free responses, *Neural Computation*, vol. 15, 2003, pp 2147-2177.

[10] L. Wiskott and T. Sejnowski, Slow feature analysis: Unsupervised learning of invariances, *Neural Computation*, vol. 14, 2002, pp 715-770.