Using Slow Feature Analysis to Extract Behavioural Manifolds Related to Humanoid Robot Postures

Sebastian Höfer **Manfred Hild** **Matthias Kubisch**
Neurorobotics Research Laboratory
Humboldt-Universität zu Berlin
10099 Berlin, Germany
{hoefer|hild|kubisch}@informatik.hu-berlin.de

Abstract

This paper demonstrates how Slow Feature Analysis (SFA), an unsupervised learning algorithm stemming from the domain of theoretical biology, can be used to extract behavioural manifolds related to a humanoid robot's body postures. On one hand, we show that SFA detects abstract semantic features, encoding high-level behaviours, which can be used for representation making and the classification of the robot's posture; on the other hand we propose a method for analysing the obtained SFA components in terms of the manifold that contains the robot's sensory states belonging to the detected postures. This allows further characterisation of the SFA results as well as a possible means for directed exploration of the sensory state space.

1. Introduction

The integration of a rich repertoire of sensory qualities is indispensable for providing an embodied autonomous agent with proprioceptive and exteroceptive perception. However, proper execution and planning of complex cognitive and behavioural tasks usually rely on high-dimensional and error-prone sensory data. Therefore, the agent needs to extract relevant information from the available data and acquire an efficient, low-dimensional representation of its morphological properties as well as its environment.

Computational models for learning and computing such representations are a current field of study, and in the past decades several promising methods have been proposed. Many of these methods can be subsumed under the term *dimensionality reduction*: The aim is to find a low-dimensional representation which captures important characteristics of the original high-dimensional state space. Some of the most prominent methods are *Principal Component Analysis (PCA)* (Pearson, 1901), *Self-organising maps (SOM)* (Kohonen, 1982) and *Locally Linear Embedding (LLE)* (Roweis and Saul, 2000). All of these methods have been widely studied and extended in several ways[1].

In this paper, we investigate an unsupervised, biologically inspired learning paradigm and its application to representation making and recognition of a humanoid robot's postures. Slow Feature Analysis (SFA), the learning algorithm presented in this paper, originates from the domain of theoretical biology and was developed in order to find a method for learning and extracting invariances from visual data, applying the principle of temporal slowness. The key assumption of this principle is that high-level and abstract features hidden in the input signal vary slowly over time. However, the slowness principle must not be confounded with low-pass filtering, for the latter operates locally, while SFA takes the whole input space into account, thus integrating global information.

SFA is able to operate on high-dimensional data, even from different sensory modalities, and adapts to small as well as to large training data sets. It is based on the generalised eigenvalue problem, for which fast and reliable algorithms exist, facilitating its efficient implementation and its application in real-time or at least in a batch processing manner. Finally, suitable methods for the analysis of the algorithm's solution can be given, for the solution is available in a compact and closed form.

In the first instance, it could be shown that temporal slowness is a fundamental learning principle, and that the application of SFA to visual data yields structures that resemble cells found in the primary visual cortex and the hippocampus (Berkes and Wiskott, 2002, Franzius et al., 2007). Beside its biological foundation, the algorithm's general capability to detect and extract hidden states and driving forces from non-stationary time-series (Wiskott, 2003a) as well as its use for pattern recognition (Berkes, 2006) have been

[1] See e.g. (Fodor, 2002) for a survey of dimensionality reduction methods.

investigated.

In recent papers, we successfully applied SFA to different sensory qualities for the detection of simple robot postures (Spranger et al., 2009) on one hand, and increased the reactivity of a biped gait pattern by using SFA as a filter structure on the other hand (Höfer and Hild, 2010). In this paper we extend the application of SFA to a more complex task related to representation making and dimensionality reduction. More precisely, we employ SFA to extract signals, which distinguish several presented postures and serve as a compact representation of the sensory state space. Additionally, we point out that SFA yields a transformation function in a compact form, facilitating the analysis of the obtained signal. In consequence, we propose a method to evaluate how changes in the sensory input data affect the obtained posture detector signals by relating SFA components to *quadrics*, which are used in robotics to model invariants in motion and behaviour (Selig, 2005). Based on the relation between SFA and quadrics we propose how SFA components can be used for guided exploration of the state space.

The outline of this paper is as follows: We begin with a brief introduction to SFA, illustrating the algorithm and its mathematical foundations. Subsequently, we shift the focus to quadratic forms and their relation to SFA. We derive a simple and effective method based on quadrics, that allows to analytically relate input and output of an SFA component. Next, we present an application of SFA to representation making and recognition of a humanoid robot's body postures. The obtained results are examined with the quadric-based analysis technique. We conclude this paper with a summary of the obtained results and give insights into future work.

2. Methods

2.1 Slow Feature Analysis

Slow Feature Analysis (SFA) is an unsupervised learning algorithm which aims to extract slowly varying features from a multi-dimensional input signal (Wiskott, 1998, Wiskott and Sejnowski, 2002). It solves a particular optimisation problem related to temporal slowness which can be stated as follows: Given a potentially multidimensional input signal $\mathbf{x}(t) = [x_1(t), .., x_N(t)]^T$, N being the dimensionality of the input, the algorithm searches for input-output functions $g_j(\mathbf{x})$, $j \in J$ that determine the output of the system $y_j(t) := g_j(\mathbf{x}(t))$. The objective function can be stated as

$$\Delta(y_j) \quad := \quad \langle \dot{y}_j^2 \rangle_t \qquad \text{is minimal} \qquad (1)$$

where $\langle \cdot \rangle_t$ denotes the average over time and $\dot{y}$ is the derivative[2] of y. For convenience, we usually omit the time index indicated in parentheses.

Since the equation states the intended learning problem of temporal slowness, $\Delta(y_j)$ is minimal if y_j varies slowly over time. Three additional constraints are formulated in order to prevent trivial solutions:

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

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

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

Without equation 3 every constant signal would easily fulfill the objective 1, so the output signal is forced to carry information. Equation 4 requires the set of output functions to be decorrelated; otherwise the signals would simply reproduce each other. It also induces an order on the output signals, i.e., the first signal y_1 will be the slowest one, y_2 will be less slower, etc.

The above stated optimisation problem is in general hard to solve. Therefore, SFA simplifies the problem by constraining the input-output functions g_j to be linear combinations of a finite set of basis functions. So, the input-output function $\mathbf{g} = [g_1(\mathbf{x}), ..., g_J(\mathbf{x})]^T$ is defined as the weighted sum of K basis functions $\mathbf{h} = [h_1, .., h_k]^T$:

$$y_j = g_j(\mathbf{x}) := \sum_{k=1}^{K} w_{jk} h_k(\mathbf{x}). \qquad (5)$$

In the linear case (called SFA(1) or *linear SFA*) no specific basis functions are used and the input-output functions compute as the weighted sum of the input data. However, in order to deal with nonlinearities in the input data, the basis functions are chosen to be a polynomial, usually quadratic, expansion of the input. This leaves only the weight vectors $\mathbf{w}_j$ to be learnt. A polynomial expansion up to degree two, prepended to a linear SFA is referred to as SFA(2) or *quadratic SFA*. Note that this technique is similar to the kernel trick, for the expanded signal serves as a basis for a finite dimensional subset of the vector space of polynomials.

Letting $\tilde{\mathbf{x}}$ be the original input data or in case of SFA(2) the expanded data, respectively, the parameters are learnt by applying SFA to the mean centered signal $\mathbf{x} = \tilde{\mathbf{x}} - \langle \tilde{\mathbf{x}} \rangle_t$. Obviously $\mathbf{x}$ automatically fulfills constraint 2, so $\mathbf{x}$ is inserted into the objective function 1 and into equation 4:

$$\Delta(y_j) \quad = \quad \langle \dot{y}_j^2 \rangle_t = \mathbf{w}_j^T \langle \dot{\mathbf{x}} \dot{\mathbf{x}}^T \rangle_t \mathbf{w}_j =: \mathbf{w}_j^T \mathbf{A} \mathbf{w}_j \qquad (6)$$

and

$$\langle y_i y_j \rangle_t \quad = \quad \mathbf{w}_i^T \langle \mathbf{x} \mathbf{x}^T \rangle_t \mathbf{w}_j^T =: \mathbf{w}_i^T \mathbf{B} \mathbf{w}_j. \qquad (7)$$

[2] The derivative is approximated by a finite difference $\dot{x}(t) := x(t) - x(t-1)$ for we are dealing with discrete signals.

For constraint 3 can be integrated into equation 1, we get the new objective function

$$\Delta(y_j) = \frac{\langle \dot{y}_j^2 \rangle_t}{\langle y_j^2 \rangle_t} = \frac{\mathbf{w}_j^T \mathbf{A} \mathbf{w}_j}{\mathbf{w}_j^T \mathbf{B} \mathbf{w}_j}. \quad (8)$$

The solution to this problem is given by the generalised eigenvalue approach as known from linear algebra,

$$\mathbf{AW} = \mathbf{BW}\mathbf{\Lambda}, \quad (9)$$

letting $\mathbf{W} = [w_1, \ldots, w_n]$ be the matrix of the generalised eigenvectors and $\mathbf{\Lambda}$ the diagonal matrix of the corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$. It was shown in (Berkes, 2006) that the orthonormal set of eigenvectors sorted in descending order accordingly to their corresponding eigenvalues yields the weight vectors $\mathbf{w}_j$.

What makes SFA an unsupervised learning algorithm is the fact, that the learnt weight vector set will generalise well to an unseen input signal, as long as the training signal shares most of the characteristics of the target input signal. Applying a trained SFA(2) to new data simply consists in the multiplication of the nonlinearly expanded, mean centered input signal by the SFA weight matrix $\mathbf{W}$, therefore being computationally less demanding than the previously described exact solution of the optimisation problem. However, SFA(2) does heavily suffer from the curse of dimensionality. If an extremely high-dimensional input signal is provided, the polynomial expansion results in a hardly manageable set of basis functions. In order to deal with this problem, SFA can be applied successively in subsequent or parallel networks of SFA units, with each unit passing only a limited amount of slowest components to the next one. Moreover, the successive application of several SFA units allows to extract features of higher polynomial degree. In this paper, we confine ourselves to the subsequent (non-hierarchical) application of several SFA units with a restricted amount of passed components and call this method *iterated SFA*, indicating the iteration from which a slowest component results in brackets, e.g., $y_1[2]$ denotes the slowest component obtained after two SFA iterations.

2.2 Quadratic Form and Quadrics

As shown in (Berkes and Wiskott, 2006) every input-output function $y_j(t) = g_j(\mathbf{x}(t))$ learnt by an SFA(2) can be formulated in a general inhomogenous quadratic form as given by the following equation:

$$y_j = c + \mathbf{f}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x}, \quad (10)$$

with $c \in \mathbb{R}$, $\mathbf{f} \in \mathbb{R}^N$ and $\mathbf{H} \in \mathbb{R}^{N \times N}$ being derived from the previously mentioned weight vector $\mathbf{w}_j$.

Restraining $y_j(t)$ to a fixed real value μ_j and bringing this value to the right hand side of the equation results in a homogenous quadratic form:

$$0 = (c - \mu_j) + \mathbf{f}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x}. \quad (11)$$

Considering the null space of this quadratic form, i.e., all the coordinates satisfying equation 11, results in an N-dimensional hypersurface, called *quadric*. In the case of $N = 2$, for instance, the possible surfaces correspond to conic sections, whose shape depends on the coefficients $\mathbf{f}$, $\mathbf{H}$ and $(c - \mu_j)$. A quadric derived from an SFA(2) component may also be called an *invariance manifold*, a term which has already been used in the original SFA publication (Wiskott, 1998).

In order to characterise a quadric hypersurface and calculate its points analytically, the quadric definition of equation 11 is normalised through elimination of the mixed terms by means of a PCA. For online exploration of the quadric surface, e.g., a gradient descent based method can be applied: The idea is to start with a point lying on the quadric surface, then execute a small random or directed movement and finally retract following the gradient towards the quadric surface.

Since it is not possible to visualise the complete quadric surface for dimensionality $N > 3$, in general, not all inputs of an SFA component can be evaluated at the same time. However, a three-dimensional subset of the input vector can be selected, fixating the remaining input variables to reasonable constant values. By subsequent application of this method to different subsets of the input vector, the unfolding subquadrics can be examined and the relationship between the inputs and the SFA component becomes evident.

As mentioned in the previous section, several SFA units can be applied in a row in order to extract features based on polynomials of higher degree. It is worth to mention, that the proposed analysis is also applicable to components resulting from an iterated SFA. For instance, a component from the second iteration of an SFA can be expressed by a polynomial of degree four. Though, the resulting geometrical surface does no longer correspond to a quadric, but to some structure of higher degree. However, the components can still be analysed, but in order to calculate the null space for a component resulting from a iteration higher than two (consisting of polynomials of degree higher than four) only numerical approaches like the proposed gradient-based method are applicable.

We will point out in the result section, that the analysis of quadrics proves useful in the case of posture detector signals extracted by SFA, for it enables us to characterise the sensory state space of the robot that is classified by SFA as a specific posture.

3. Experiments

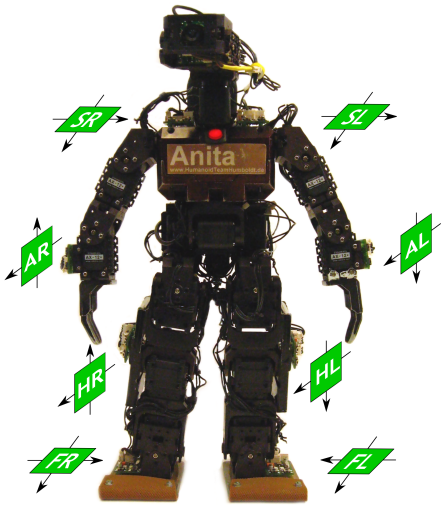


Figure 1: Picture of a robot from the A-series with schematic representation of the positions and directions of the acceleration sensor boards.

We confined ourselves to iterated quadratic SFA. The number of iterations was varied from one to five, the number of slowest components passed to a subsequent unit was set to 48. We observed that increasing the values of these parameters, particularly the number of iterations, results in overfitted and degenerate slowest components which is in agreement with the theoretical analysis of optimal slowest components (Wiskott, 2003b). Additionally, the values of the slowest components were cutoff at $[-2, 2]$ after each iteration in order to prevent high peaks which may arise due to the unit variance constraint (equation 3). For our experiments the SFA implementation available from the open source *Modular Toolkit for Data Processing (MDP)* (Zito et al., 2009) was used.

3.1 Embodiment

For our experiment we used robots of the *A-series* platform (Figure 1), which was developed at our laboratory for researching basic motion capabilities of humanoids . The robot platform features several proprioceptive sensors which are distributed across the body as well as a camera in the head. It exhibits 21 degrees of freedom, 19 in the body, including elbow, hand, hip, knee and foot joints, as well as a pan-and-tilt unit for the camera. Eight microprocessor boards are located on the hips, arms and shoulders, featuring a two-axes acceleration sensor each. Each board controls up to two actuators, while communicating via a shared system bus, that integrates incoming and outgoing data from the sensors, the motors and a PDA, which is attached to the back of the robot to process visual information provided by the camera.

3.2 Training Data

In our experiments a 16-dimensional input signal consisting of the acceleration sensor values was used. All sensor values were normalised to $[-1, 1]$. The training data consisted of a collection of sequences with an overall length of 120 seconds. In the recorded sequences, the robot executed different behaviours, namely laying down to the back and the front, standing up, doing the splits and squatting. The static postures were held for a longer period of time compared to the fairly swift transitions from one posture to another. The splits and squatting postures are more difficult to detect, for they affect less sensors than standing or lying. Selected acceleration sensor signals recorded during the aforementioned sequence can be seen in Figure 2.

4. Results

In Figure 3 the five slowest components from the second SFA iteration are depicted. In order to decide which two components form the best representation of the sensory state space, two numerical measures were used: The *silhouette* measure (Rousseeuw, 1987) was used to evaluate the discernibility of the postures, while procrustes analysis (Li et al., 1995) indicates the (linear) dissimilarity of the original data with the results. The values for the best SFA component pair $(y_2[2], y_5[2])$ are listed in Table 1. Examination of all the components from the second SFA iteration shows, why the pairing of these two components yields the best result: $y_1[2]$ is a merely binary component, extracting the postures standing and lying, rather irrespective of the side on which the robot is lying. $y_2[2]$ takes this difference into account and additionally exhibits a negative peak when the robot is doing the splits at $t = 10150$. While $y_3[2]$ resembles the slowest component $y_1[2]$ with additional strong peaks during posture changes, $y_4[2]$ is most sensitive to posture changes. Finally, $y_5[2]$ is the only component exhibiting a remarkable peak at $t = 8180$, which corresponds to the squatting pose, therefore being necessary for the salient distinction of all the available postures. Figure 4 shows a two-dimensional visualisation of the reduced state space constituted of the aforementioned SFA component pair. Each dot represents the output of the SFA components given a sensory input vector from the training data. The sensory states that correspond to static postures were highlighted manually and pictures of the respective poses were added for convenience. The yellow dots represent intermediary states, i.e., transitions from one posture to another. It can be clearly seen that the different postures are well distinguishable and separated in the SFA state space. Another interesting observation is that the trajectories of the transi-

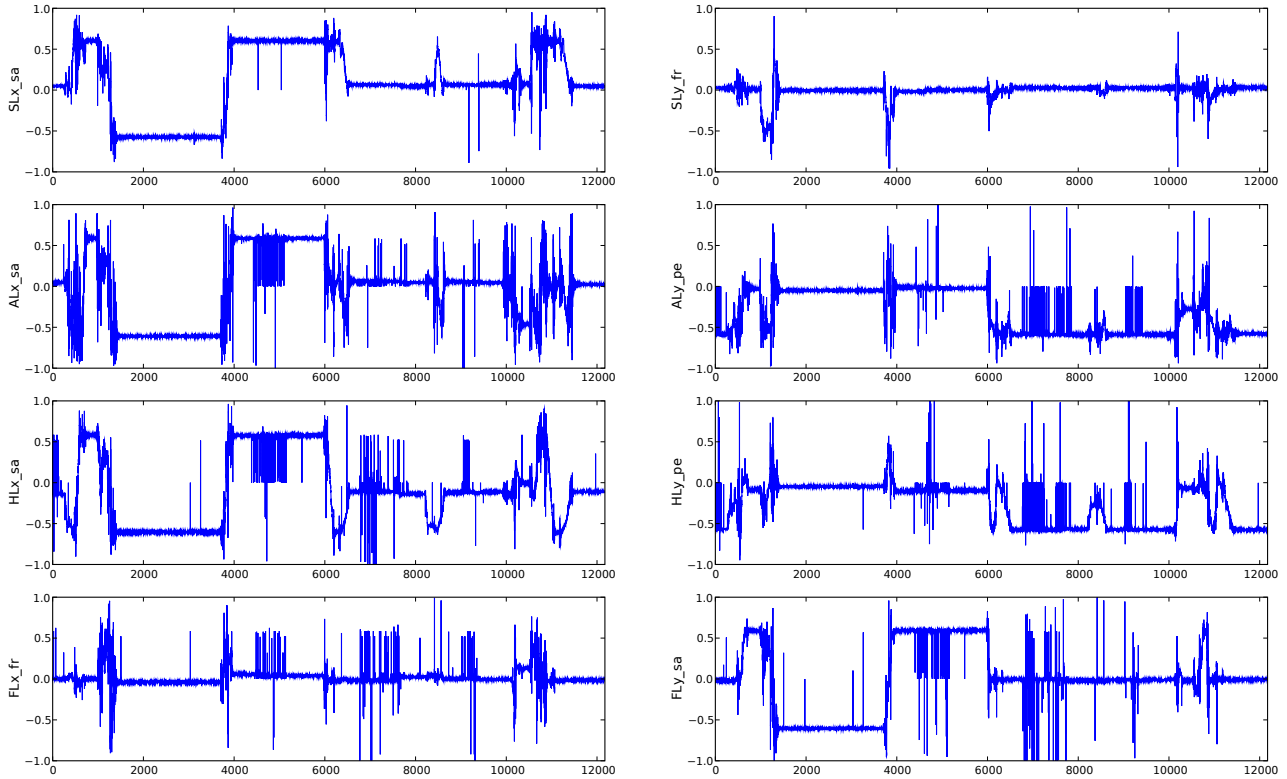


Figure 2: Selected acceleration sensor signals consisting of all the sensors located on the robot's left body part. The signals exhibit high noise which may be ascribed to the high sensitivity of the sensors as well as transmission errors.

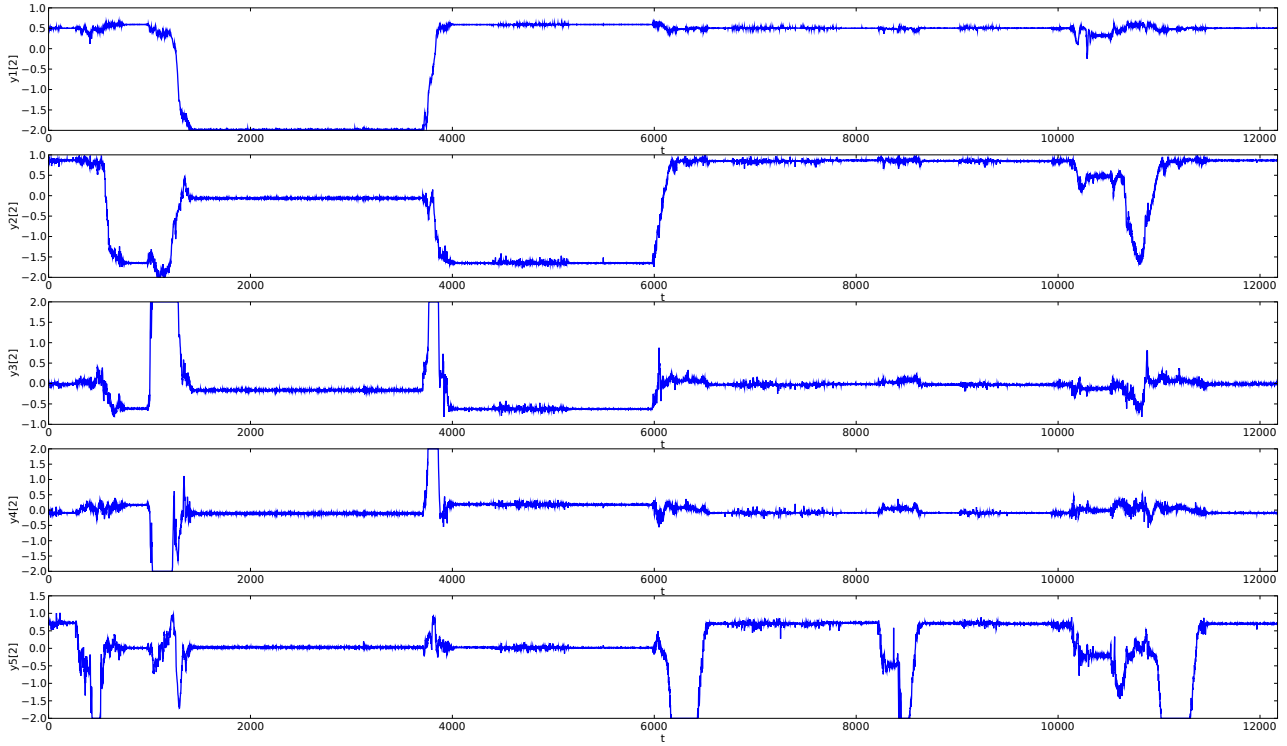


Figure 3: The five slowest components obtained after two SFA iterations.

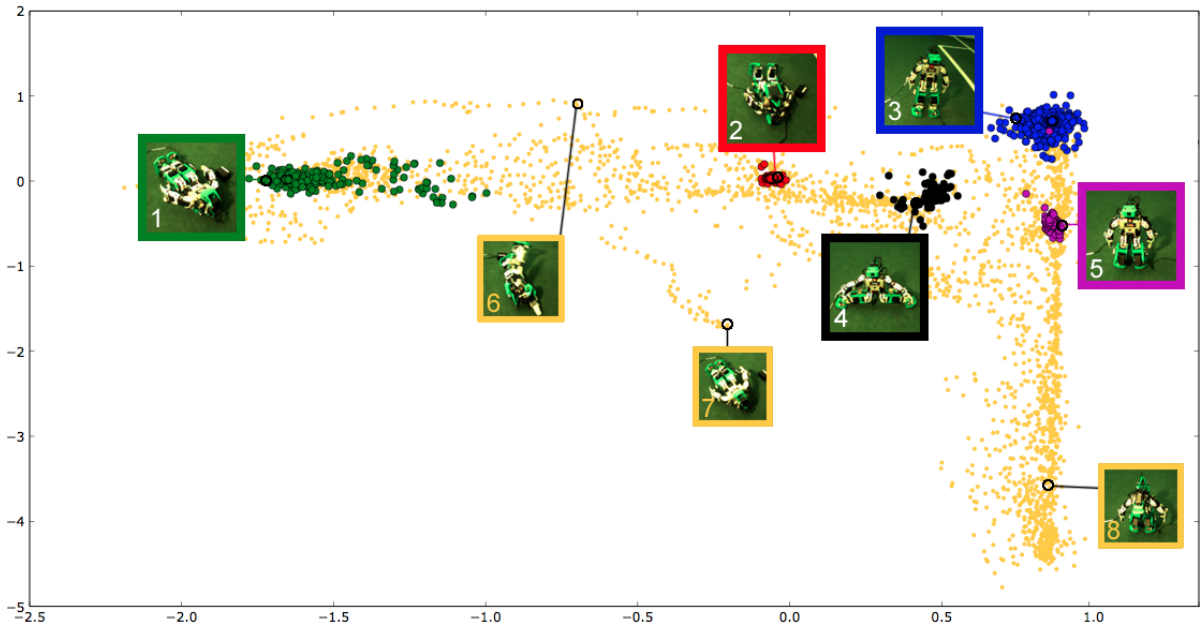


Figure 4: A two-dimensional visualisation of the postures executed by the robot and the most salient slowest components. The x-axis corresponds to $y_2[2]$, the y-axis to $y_5[2]$. Intermediary states are indicated by yellow dots.

tions are reasonable with respect to the intermediary states, depicted by the yellow dots: For instance, the sensory states illustrated by picture number 8 in the bottom right corner of Figure 4 are reached while the robot stands up from the ground. This is owed to an advantageous side effect of the temporal slowness objective: The signals do not only capture high-level semantic features, but are also smooth and deprived from noise.

4.1 Comparison with PCA and LLE

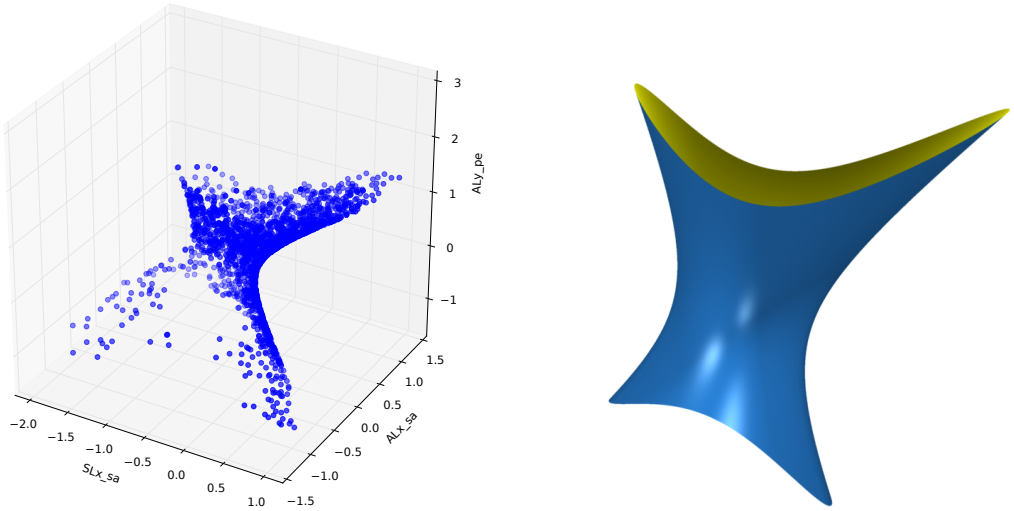
In order to assess the obtained SFA results, PCA and LLE (with k-neighbourhood size set to $k = 91$) were applied to the training sequence. The resulting dimensionality reductions were compared using the aforementioned measures. As may be expected, PCA and LLE preserve higher similarity with the input data, as being reflected in the lower procrustes value. The high dissimilarity of the SFA components with the input data is mostly owed to the fact that a quadratic and thus nonlinear variant of SFA was used. Though, this dissimilarity is not necessarily disadvantageous, quite the contrary: Due to the slowness objective, the most salient robot postures are pulled apart in the resulting state space, facilitating a better discernibility of the different postures. Besides, the postures are also much more locally concentrated in the SFA result. This explains why SFA exhibits the highest silhouette value. As observed before, the slowness principle makes the trajectories between the different poses look smooth, which is not the case for PCA and LLE.

	SFA	PCA	LLE
Silhouette	0.71	0.63	0.56
Procrustes	0.71	0.06	0.36

Table 1: Silhouette and procrustes measure results for SFA, PCA and LLE.

4.2 Analysis with quadrics

As proposed earlier, quadric analysis can be used to learn more about the characteristics of the extracted SFA components. In particular, it gives new insights to the response of the components when being applied to unseen sensory data. Moreover, it shows that the obtained SFA components constitute behavioural manifolds, which represent the robot's sensory states that belong to a posture. In this paper we present the results for the second slowest component from the first iteration $y_2[1]$. This component shows high resemblance to its counterpart in the second iteration, an observation which is underpinned by their correlation coefficient $\rho(y_2[1], y_2[2]) = -0.96$. In comparison to $y_2[2]$, the component $y_2[1]$ from the first iteration is not as smooth and has an opposite sign. We choose the standing posture for our analysis. Thus, we set $\mu_2 = 0.915$ which is the mean value of $y_2[2]$ while the robot executes this posture. In order to visualise the quadric three-dimensionally we cannot leave all sensor inputs variable, but we select three sensors located on the left part of the upper body: the sensors in sagittal direction on the



(a) Gradient descent

(b) Analytical calculation

Figure 5: The resulting quadric for $y_2[1]$ ($\mu_2 = 0.915$), obtained by the proposed gradient descent method and an analytical calculation, respectively. The quadric represents the simulation of movements of the left arm in a standing posture. Geometrically, the result set corresponds to a hyperboloid of one sheet.

shoulder (`SLx_sa`) and the arm (`ALx_sa`) as well as the arm sensor which is aligned perpendicular to the transversal plane of the robot (`ALy_pe`), i.e., the axis pointing from the robot's head down to its feet. All other sensors are fixated to their mean value while the standing posture is executed. The sensors have been chosen for $y_2[1]$ exhibits similarity with `ALx_pe` ($\rho(\texttt{ALy\_pe}, y_2[1]) = -0.83$), as well as for the fact that $y_2[1]$ distinguishes whether the robot is lying on its front or back side. The following analysis will show that also the sagittal sensors have a remarkable influence on the component.

In Figure 5 the resulting quadric, a hyperboloid of one sheet is depicted. Figure 5a shows the quadric surface which was calculated using the previously proposed gradient based method, constraining the input values to $[-1.5, 1.5]$. In fact, this value range is fairly ample, since the sensors do not exceed ± 0.6 for static postures, due to their calibration. Figure 5b depicts the analytically derived quadric surface.

First, we take a look at the relationship between the two sagittal sensors. These sensors are strongly coupled, since they have approximately the same value if the morphology of the upper body is not changed, i.e., if the arm is in its basic position, perpendicular to the transversal plane. As can be seen when considering only the sagittal axis and therefore fixating `ALx_pe` to its mean value for the standing posture, the two sensors are geometrically coupled with each other by an ellipse. Interestingly, the position of the arm does indeed have an impact on $y_2[2]$: If the robot moves its arm up to the front, using its pitch motor, it has to lean its upper body to the back in order to stay on the quadric surface, keeping the value of $y_2[2]$ at μ_2, respectively[3]. Though, moving the arm backwards does not have an effect on $y_2[2]$. Nevertheless, the observations imply that SFA makes use of the sagittal arm sensor in order to encode the standing posture; as indicated in our recent paper (Spranger et al., 2009), it is most probable that the dependence on the sagittal arm sensor can be reduced if the training data consists of more dynamical data including arm movements.

When looking at the pairs consisting of one of the sagittal sensors and the perpendicular sensor, the geometrical shapes correspond to hyperbolas. It can be seen that the coupling between these sensors is not as strong as the previously examined sensor pair: The perpendicular arm sensor is insensitive with respect to the sagittal shoulder sensor, thus allowing free movement of the arm. Moreover, the quadric surface points formed by the two sensors both located on the arm are practically not relevant, for most of the indicated sensor values are not reachable in terms of the robot's actual morphology.

5. Conclusion and Outlook

We have shown how Slow Feature Analysis, an unsupervised learning algorithm based on the principle of temporal slowness, can be applied to sensory data from a humanoid robot, extracting components which encode the robot's body postures. The presented components are suitable not only for classification of the robot's postures, but also provide an appropriate dimensionality reduction, exhibiting rea-

[3] However, the possible side effects on the remaining sensors when moving the upper body would have to be taken into account, too.

sonable transitions and trajectories through intermediate states. Moreover, we have presented a straightforward analysis technique based on quadrics, that allows for further characterisation of the relationship between the input and the extracted features. Eventually, we have demonstrated, how this technique can be used for the reconstruction of a manifold that SFA categorises as a specific posture.

Notwithstanding, further investigation concerning the applicability of SFA to larger training data sets, as well as its generalisability is necessary. Additionally, the proposed gradient based approach should be used to explore the quadric and the corresponding subset of the sensory state space in a directed manner. The result of this exploration could then serve as new training data for the SFA units, i.e., for further adaptation of SFA components.

Future work will mainly concentrate on the applicability of SFA to the newly available successor of the A-series platform, the *Myon* robot. This platform is equipped with a significantly higher amount and additional modalities of sensors, e.g., current and force sensors. Moreover, the concept of life-long learning was introduced into the new platform, enabling the robot to gather experience in form of sensory data which may serve as training data for advanced learning algorithms like SFA. We believe that the increased amount of diversified sensory data proves to be useful for the extraction of even more salient and robust high level abstract features by the SFA.

Acknowledgements

The authors would like to thank André Stephan for the generation of the employed sensory data. This work has been supported by the European research project ALEAR (FP7, ICT-214856).

References

Berkes, P. (2006). *Temporal slowness as an unsupervised learning principle*. PhD thesis, Humboldt-Universität zu Berlin.

Berkes, P. and Wiskott, L. (2002). Applying Slow Feature Analysis to Image Sequences Yields a Rich Repertoire of Complex Cell Properties. In Dorronsoro, J. R., (Ed.), *Proc. Intl. Conf. on Artificial Neural Networks - ICANN'02*, Lecture Notes in Computer Science, pages 81–86. Springer.

Berkes, P. and Wiskott, L. (2006). On the analysis and interpretation of inhomogeneous quadratic forms as receptive fields. *Neural Computation*, 18(8):1868–1895.

Fodor, I. (2002). A Survey of Dimension Reduction Techniques. Technical report.

Franzius, M., Sprekeler, H., and Wiskott, L. (2007). Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLoS Computational Biology*, 3(8):e166.

Höfer, S. and Hild, M. (2010). Using Slow Feature Analysis to Improve the Reactivity of a Humanoid Robot's Sensorimotor Gait Pattern (accepted). In *International Conference on Neural Computation (ICNC 2010)*, Valencia, Spain.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

Li, S., de Vel, O., and Coomans, D. (1995). Comparative performance analysis of non-linear dimensionality reduction methods. Technical report, James Cook University, North.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572.

Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65.

Roweis, S. and Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Selig, J. (2005). *Geometric Fundamentals of Robotics*. Springer, New York.

Spranger, M., Höfer, S., and Hild, M. (2009). Biologically inspired posture recognition and posture change detection for humanoid robots. In *Proc. IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 562–567, Guilin, China.

Wiskott, L. (1998). Learning Invariance Manifolds. In *Proc. of the 5th Joint Symp. on Neural Computation, May 16, San Diego, CA*, volume 8, pages 196–203, San Diego, CA. Univ. of California.

Wiskott, L. (2003a). Estimating Driving Forces of Nonstationary Time Series with Slow Feature Analysis.

Wiskott, L. (2003b). Slow Feature Analysis: A Theoretical Analysis of Optimal Free Responses. *Neural Computation*, 15(9):2147–2177.

Wiskott, L. and Sejnowski, T. (2002). Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural Computation*, 14(4):715–770.

Zito, T., Wilbert, N., Wiskott, L., and Berkes, P. (2009). Modular toolkit for Data Processing (MDP): a Python data processing frame work.