An Algorithm That Recognizes and Reproduces Distinct Types of Humanoid Motion Based on Periodically-Constrained Nonlinear PCA

Rawichote Chalodhorn, Karl MacDorman, and Minoru Asada

Department of Adaptive Machine Systems *and* Frontier Research Center,
Graduate School of Engineering, Osaka University,
2-1 Yamada-oka, Suita, Osaka 565-0871 Japan
kfm@ed.ams.eng.osaka-u.ac.jp
http://ed.ams.eng.osaka-u.ac.jp

Abstract. This paper proposes a new algorithm for the automatic segmentation of motion data from a humanoid soccer playing robot that allows feedforward neural networks to generalize and reproduce various kinematic patterns, including walking, turning, and sidestepping. Data from a 20 degree-of-freedom Fujitsu HOAP-1 robot is reduced to its intrinsic dimensionality, as determined by the ISOMAP procedure, by means of nonlinear principal component analysis (NLPCA). The proposed algorithm then automatically segments motion patterns by incrementally generating periodic temporally-constrained nonlinear PCA neural networks and assigning data points to these networks in a *conquer*-and-divide fashion, that is, each network's ability to learn the data influences the data's division among the networks. The learned networks abstract five out of six types of motion without any prior information about the number or type of motion patterns. The multiple decoding subnetworks that result can serve to generate abstract actions for playing soccer and other complex tasks.

1 Introduction

The development of robots that can learn to imitate human behavior as they participate in social activities is important both for understanding ourselves as a species and for transforming society through the introduction of new technologies. A mimesis loop [1] may be used to capture many aspects of this kind of imitative learning. This paper addresses one aspect of the mimesis loop: the abstraction of a robot's own kinematic motions from its proprioceptive experience.

Figure 1 roughly outlines how a mimesis loop might be realized in a soccer playing robot. Attentional mechanisms direct the robot's sensors toward the body parts of other players, and the robot maps successfully recognized body parts onto its own body schema. This paper introduces a method to abstract the robot's own kinematic patterns: our segmentation algorithm allocates proprioceptive data among periodic temporally-constrained nonlinear principal component neural networks (NLPCNNs) as they form appropriate generalizations.

D. Nardi et al. (Eds.): RoboCup 2004, LNAI 3276, pp. 370–380, 2005.

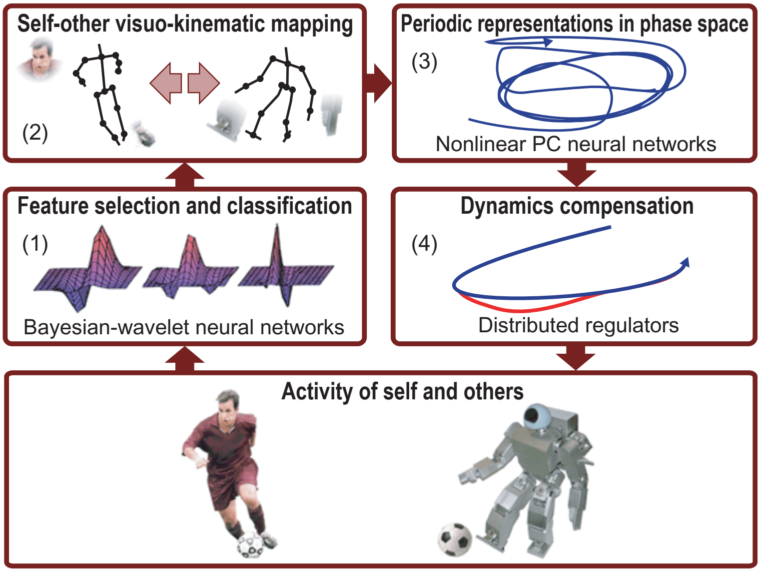

Fig. 1. Periodic nonlinear principal component networks may characterize motion patterns in a much larger system for recognizing, learning, and responding behavior

NLPCNNs, augmented with periodic and temporal constraints, provide an effective means of characterizing many typical human motions. These networks may be used to recognize, learn, and respond to behavior. A single network abstracts a particular type of periodic motion from joint angles and other proprioceptive data. A different network learns a different type of periodic motion until all the various kinds of motion have been learned. Networks can also learn transitions between motion patterns.

The robot can use NLPCNNs to recognize the activities of other players, if the mapping from their bodies to its own has already been derived by some other method. Since each network correspond to a particular type of motion in a proprioceptive phase space, it can act as a protosymbol. Thus, the robot would be able to recognize the behavior of others because it has grounded their behavior in terms of its own body.

Although periodic NLPCNNs may be used to generate motion patterns, the robot must continuously respond to unexpected perturbations. There are a number of approaches to this control problem that do not require an explicit model. For example, distributed regulators [2] could set up flow vectors around learned trajectories, thus, converting them into basins of attraction in a phase space of possible actions.

This paper is organized as follows. Section 2 extends an NLPCNN with periodic and temporal constraints. Section 3 presents a method of assigning observations to NLPCNNs to segment proprioceptive data. Section 4 reports experimental results using NLPCNNs to characterize the behavior of a Fujitsu HOAP-1 humanoid robot that has been developed to play RoboCup soccer.

2 A Periodic Nonlinear Principal Component Neural Network

The human body has 244 degrees of freedom [3] and a vast array of proprioceptors. Excluding the hands, a humanoid robot generally has at least 20 degrees of freedom — and far more dimensions are required to describe its dynamics precisely. However, many approaches to controlling the dynamics of a robot are only tractable when sensory data is encoded in fewer dimensions (e.g., [4]). Fortunately, from the standpoint of a particular activity, the effective dimensionality may be much lower.

Given a coding function $f : \mathbb{R}^N \mapsto \mathbb{R}^P$ and decoding function $g : \mathbb{R}^P \mapsto \mathbb{R}^N$ that belong to the sets of continuous nonlinear functions $\mathcal{C}$ and $\mathcal{D}$, respectively, where $P < N$, nonlinear principle component networks minimize the error function E

$$\|\boldsymbol{x} - g(f(\boldsymbol{x}))\|^2, \quad \boldsymbol{x} \in \mathbb{R}^N$$

resulting in P principal components $[y_1 \cdots y_p] = f(\boldsymbol{x})$. Kramer [5] first solved this problem by training a multilayer perceptron similar to the one shown in Figure 2 using the backpropagation of error, although a second order method such as conjugant gradient analysis converges to a solution faster for many large data sets. Tatani and Nakamura [6] were the first to apply an NLPCNN to human and humanoid motions, though for dimensionality reduction only.

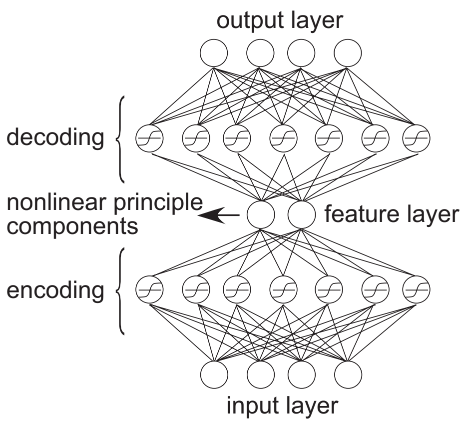

Fig. 2. Target values presented at the output layer of a nonlinear principal component neural network are identical to input values. Nonlinear units comprise the encoding and decoding layers, while either linear or nonlinear units comprise the feature and output layers

Nonlinear principal components analysis, unlike PCA (Karhunen-Loève expansion), which is a special case where $\mathcal{C}$ and $\mathcal{D}$ are linear, does not have a unique solution, and no known computational method is guaranteed to find any of the globally optimal solutions.

Nevertheless, for a 20-DoF humanoid robot, a hierarchically-constructed[1] NLPCNN has been shown to minimize error several times more than PCA when reducing to two-to-five dimensions [6].

2.1 The Periodicity Constraint

Because the coding function f of an NLPCNN is continuous, *(1)* projections to a curve or surface of lower dimensionality are suboptimal; *(2)* the curve or surface cannot intersect itself (e.g., be elliptical or annular); and *(3)* projections do not accurately represent discontinuities [7]. However, since the physical processes underlying motion data are continuous, discontinuities do not need to be modelled. Discontinuities caused by optimal projections can create instabilities for control algorithms (e.g., they allow points along the axis of symmetry of a parabola to be projected to either side of the parabola). Moreover, an NLPCNN with a circular node [8][9] at the feature layer can learn self-intersecting curves and surfaces.

Kirby and Miranda [10] constrained the activation values of a pair of nodes p and q in the feature layer of an NLPCNN to fall on the unit circle, thus acting as a single angular variable:

$$r = \sqrt{y_p^2 + y_q^2}, \;\; y_p \leftarrow y_p/r, \;\; y_q \leftarrow y_q/r$$

The delta values for backpropagation of the circular node-pair are calculated by the chain rule [10], resulting in the update rule

$$\delta_p \leftarrow (\delta_p y_q - \delta_q y_p) y_q / r^3, \;\; \delta_q \leftarrow (\delta_q y_p - \delta_p y_q) y_p / r^3$$

at the feature layer.

The hyperbolic tangent and other antisymmetric functions (i.e., $\varphi(x) = -\varphi(x)$) are generally preferred to the logistic function as the sigmoid in part because they are compatible with standard optimizations [11].[2] In addition, antisymmetric units can more easily be replaced with linear or circular units in the feature layer, since these units can produce negative activations. We propose using a slightly flatter antisymmetric function for the sigmoidal units with a similar response characteristic to $tanh$ (see Fig. 3). The advantage of this node is that it can be converted to a circular node-pair while still making use of its perviously learned weights.

2.2 The Temporal Constraint

Neither linear nor nonlinear principal components analysis represent the time, relative time, or order in which data are collected.[3] This information, when available, can be

[1] The joint encoder dimensionality of limbs is independently reduced, then the arms and the legs are paired and their dimensionality further reduced, and then finally the dimensionality of the entire body.

[2] These include mean cancellation, linear decorrelation using the K-L expansion, and covariance equalization.

[3] Although a temporal dimension could be added to an autoassociative network, one drawback for online learning is that this dimension would need to be continuously rescaled as more data is collected to keep it within the activation range of the nodes.

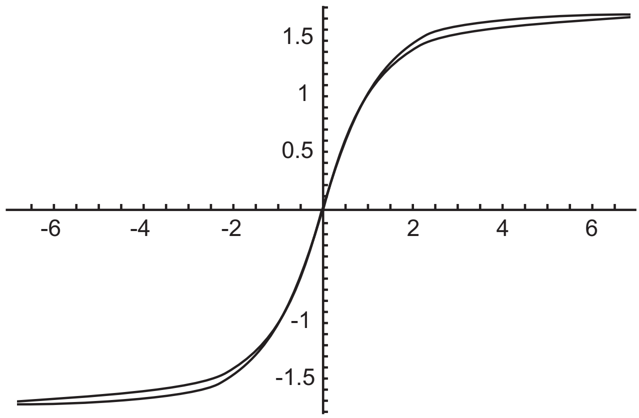

Fig. 3. The popular hyperbolic tangent activation function $y \leftarrow 1.7159\tanh(\frac{2}{3}y)$ can be approximated by a pair of circular nodes where the activation of the second node y_q is fixed at $\sqrt{1.9443}$ and the activation of the first node is calculated accordingly $y_p \leftarrow 1.7159 y_p/\sqrt{y_p^2 + 1.9443}$

used to reduce the number of layers and free parameters (i.e., weights) in the network and thereby its risk of converging slowly or settling into a solution that is only locally optimal. Since the activations y_p and y_q of the circular node-pair in the feature layer in effect represent a single free parameter, the angle θ, if θ is known, we can train the encoding and decoding subnetworks separately by presenting $k\cos(\theta)$ and $k\sin(\theta)$ as target output values for the encoding subnetwork and as input values for the decoding network.[4] Once a single period of data has been collected, temporal values can be converted to angular values $\theta = 2\pi\frac{t_k - t_0}{t_n - t_0}$ for data collected at any arbitrary time t_k during a period, starting at t_0 and ending at t_n. A network may similarly learn transitions between periodic movements when using a linear or sigmoidal activation node in the feature layer because these open-curve transitions do not restrict us to using nodes capable of forming a closed curve.[5] NLPCNNs with a circular feature node remain useful to identify the period of a motion pattern, especially when the pattern is irregular and, thus, begins and ends at points that are somewhat far from each other.

3 Automatic Segmentation

We conceived of the automatic segmentation problem as the problem of uniquely assigning data points to nonlinear principal component neural networks. It is possible to partition the points without reference to the predictions of the networks.[6] However, for

[4] $k \approx 1.7$ for zero-mean data with variance equal to 1 based on principles discussed in [11].

[5] $y_{target} = 2k(\frac{t_k - t_0}{t_n - t_0} - \frac{1}{2})$, with $k \approx 1.4$.

[6] For example, data points may be partitioned at the point at which a trajectory most closely doubles back on itself, if the distance between the two paths is within a certain threshold and the paths then diverge beyond another threshold.

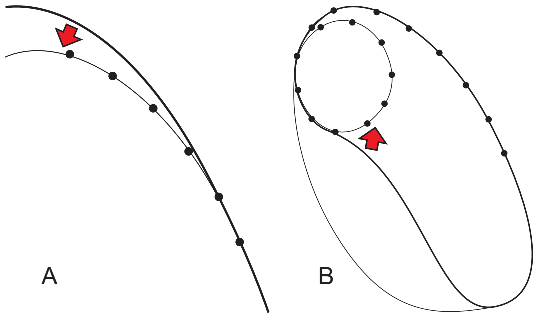

Fig. 4. The thick line shows the output of an NLPCNN and the thin line shows the underlying distribution. The dots are data points. **A.** Before learning converges, allowing the network to learn data points despite a high prediction error accelerates learning. **B.** However, after convergence, it leads to segmentation errors

our method each network's performance influences segmentation with more networks assigned to regions that are difficult to learn.

As the robot begins to move, the first network is assigned some minimal number of data points (e.g., joint-angle vectors), and its training begins with those points. This gets the network's learning started quickly and provides it with enough information to determine the orientation and curvature of the trajectory. If the average prediction error of the data points assigned to a network is below some threshold, the network is assigned additional data points until that threshold has been reached. At that point, data points will be assigned to another network, and a network will be created, if it does not already exist. To avoid instabilities, only a single data point may shift its assignment from one network to another after each training cycle.

$j \leftarrow 1,\ bucket \leftarrow 1,\ E \leftarrow 0$
$\forall \boldsymbol{x_i}$ {
  $\mathtt{train}(network_j, \boldsymbol{x_i})$
  $E_i = \|\boldsymbol{x_i} - g(f(\boldsymbol{x_i}))\|^2,\ E \leftarrow E + E_i$
  if ($bucket > B_{max}$ $\vee$
    ($\mathtt{learning?}(network_j)$ $\wedge$ $E/bucket > E_{max}$) $\vee$
    $E_i > E_{i+1}$)
    $j \leftarrow j+1,\ bucket \leftarrow 1,\ E \leftarrow 0$ }

Listing 1. Pseudocode for segmentation

Since a network is allowed to learn more data points as long as its average prediction error per point is low enough, it may learn most data points well but exhibit slack near peripheral or recently learned data points. At the start of learning, the network should be challenged to learn data points even when its prediction error is large (see Fig. 4A). As learning converges, however, the slack leads to segmentation errors (see Fig. 4B). Therefore, we alter the method of segmentation once the network nears convergence (as determined by Bayesian methods [12] or crossvalidation) so that a network may acquire neighboring points if its prediction error for those points is lower that the network currently assigned to those points.

4 Humanoid Experiments

This section shows the result of automatic segmentation and neural network learning. We assess the accuracy of the result based on a manual segmentation of the data points and an analysis of how they are allocated among the networks.

First, we recorded motion data while a HOAP-1 humanoid robot played soccer in accordance with a hard-coded program [13]. Each data point is constituted by a 20-dimensional vector of joint angles. A standard (noncircular) NLPCNN reduced the dimensionality of the data from 20 to 3, which was determined to be the intrinsic dimensionality of the data by the ISOMAP procedure [14] We then applied our algorithm to segment, generalize, and generate humanoid motion.

Our algorithm uniquely assigned the data points among a number of circularly-constrained NLPCNNs. Each of the networks learned a periodic motion pattern by conjugate gradients. Our algorithm successfully generalized five out of six primary motion patterns: walking forward, turning right or left, and side-stepping to the right or left. It failed to generalize as a single periodic trajectory the kicking motion, which has a highly irregular, self-intersecting shape. However, human subjects were also unable to determine the kicking trajectory from the data points.

Figure 5 shows that the automatic segmentation algorithm successfully employed circular NLPCNNs to separate and generalize five of the periodic motions. (The open-curve segmentation of transitions between periodic motions are omitted for clarity.) The periodic trajectories were generated by varying from 0 to 2π the angular parameter θ_i at the bottleneck layer of each of the circularly-constrained networks and mapping the result to the output layer for display. This demonstrates our method's capacity to generate periodic motions.

We calculated statistics based on running the automatic segmentation for 20 trails. The algorithm resulted in five decoding subnetworks for 45% of the trials, which is the most parsimonious solution. It resulted in six subnetworks for 50% of the trials, and seven for the remaining 5%. In results published elsewhere [15, 16], we developed a method resembling linear integration that consistently eliminated redundant networks for this data set. If the area between the predicted curves of two networks is sufficiently small, on network is removed and its data points are reassigned to the other network.[7]

Since the data was generated by the predefined behavior modules used by the Osaka University team in the 2003 RoboCup humanoid competition, each data point was already labeled and could be segmented into the five types of motion that had been successfully abstracted. To assess the accuracy of the automatic segmentation algorithm, we manually assigned the data points corresponding to each type of motion to five pe-

[7] The algorithms presented in [15] deviate from those presented here and in [17] in some minor details, the most significant being that learning and segmentation occur sequentially rather than simultaneously.

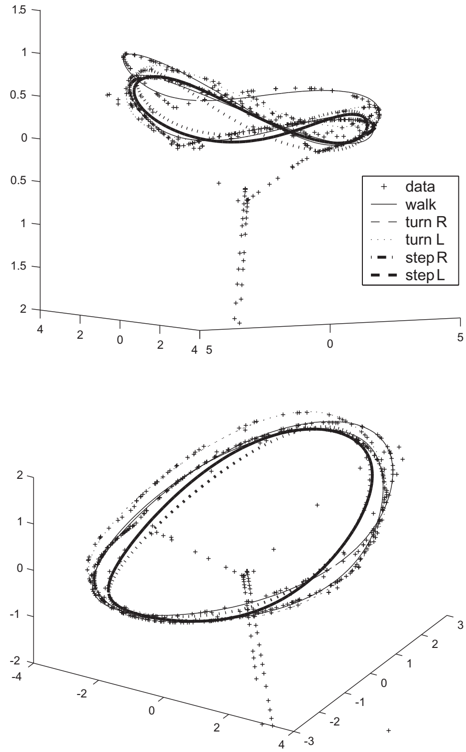

Fig. 5. Recognized motion patterns embedded in the dimensions of the first three nonlinear principal components of the raw proprioceptive data. The top and bottom plots differ only in the viewpoint used for visualization

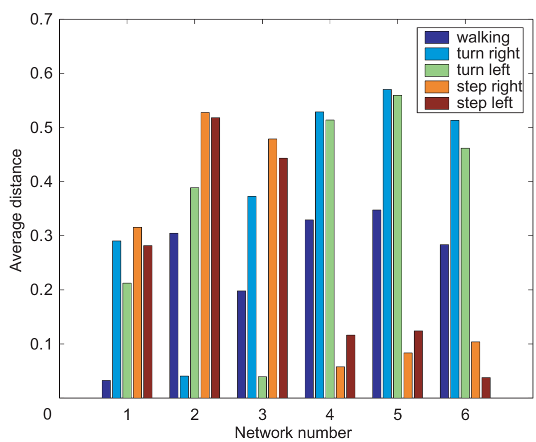

Fig. 6. The average distance between the prediction of a network trained on manually segmented data and each of the automatically generated networks

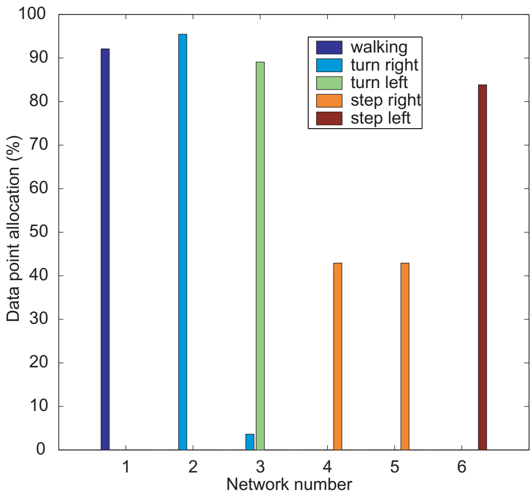

Fig. 7. The allocation of data points to each network

riodic temporally constrained NLPCNNs. Figure 6 shows the average distance between the prediction for each of these networks and each of the networks resulting from automatic segmentation.

The lowest bar indicates which pattern the networks, numbered 1 to 6 best match in terms of least average distance. Hence, the first network represents walking; the second represents turning right; the third turning left; the fourth and fifth sidestepping right; and the sixth sidestepping left. The fact that the fifth network is redundant, abstracting the same type of motion as the fourth, does not prevent the abstracted actions from supporting the mastery of soccer or some other task. Both networks can be used. The algorithm's capacity to reduce a vast amount of complex, raw data to just a few states may help reinforcement learning approaches to finesse the curse of dimensionality [18].

We counted the number of point belonging to each network. Figure 7 shows that the first network captured 92% of the walking data, the second 95% of the turning right data, the third 89% of the data for turning left and 3.6% for turning right, the fourth captured 43% of the data for sidestepping right, and the fifth 84% of the data for sidestepping left. The total number of point from each pattern allocated to the networks is not 100% because the segmentation algorithm successfully excluded most outliers.

5 Conclusion

Our proposed algorithm abstracted five out of six types of humanoid motion through a process that combines learning and data point assignment among multiple neural networks. The networks perform periodic, temporally-constrained nonlinear principal component analysis. The decoding subnetworks generate motion patterns that accurately correspond to the five motions without including outliers caused by nondeterministic perturbations in the data. During 45% of training episodes, the algorithm generated no redundant networks; a redundant network appeared in 50% of the training episodes, and two appeared in 5% of them. The fourth and fifth networks represent the same type of motion. Although they would symbolize a redundant state in the reinforcement learning paradigm, this does not prevent the learning of a complex task. In companion papers [15, 16], we propose a method that successfully removes redundant networks according to the proximity of their predictions. In future work, we will improve segmentation by competitively reassigning temporally-adjacent data points to the network that predicts the points with the least error.

References

1. Inamura, T., Toshima, I., Nakamura, Y.: Acquiring motion elements for bidirectional computation of motion recognition and generation. In Siciliano, B., Dario, P., eds.: Experimental Robotics VIII, Springer (2003) 372–381
2. Fujii, T.: A new approach to the LQ design from the viewpoint of the inverse regulator problem. IEEE Transactions on Automatic Control **32** (1987) 995–1004
3. Zatsiorsky, V.M.: Kinematics of Human Motion. Human Kinetics, Urbana Champaign (2002)
4. Okada, M., Tatani, K., Nakamura, Y.: Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion. In: IEEE International Conference on Robotics and Automation. (2002) 1410–1415
5. Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. Journal of the American Institute of Chemical Engineers **37** (1991) 233–243
6. Tatani, K., Nakamura, Y.: Dimensionality reduction and reproduction with hierarchical NLPCA neural networks extracting common space of multiple humanoid motion patterns. In: Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan (2003) 1927–1932
7. Malthouse, E.C.: Limitations of nonlinear PCA as performed with generic neural networks. IEEE Transactions on Neural Networks **9** (1998) 165–173
8. Ridella, S., Rovetta, S., Zunino, R.: Adaptive internal representation in circular backpropagation networks. Neural Computing and Applications **3** (1995) 222–333
9. Ridella, S., Rovetta, S., Zunino, R.: Circular backpropagation networks for classification. IEEE Transaction on Neural Networks **8** (1997) 84–97
10. Kirby, M.J., Miranda, R.: Circular nodes in neural networks. Neural Computation **8** (1996) 390–402
11. LeCun, Y., Bottou, L., Orr, G.B., Müller, K.R.: Efficient BackProp. In Orr, G.B., Müller, K.R., eds.: Neural Networks: Tricks of the Trade, Springer (1998) 1–44
12. MacKay, D.J.: Probable networks and plausible predictions: A review of practical Bayesian methods for supervised neural networks. Network: Computation in Neural Systems **6** (1995) 469–505
13. Chalodhorn, R., Aono, M., Ooga, J., Ogino, M., Asada, M.: Osaka University "Senchans 2003". In Browning, B., Polani, D., Bonarini, A., Yoshida, K., eds.: RoboCup-2003: Robot Soccer World Cup VII, Springer Verlag (2003)
14. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290** (2000) 2319–2323

15. Chalodhorn, R., MacDorman, K., Asada, M.: Automatic extraction of abstract actions from humanoid motion data. In: IROS-2004: IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan (Submitted)
16. MacDorman, K., Chalodhorn, R., Ishiguro, H., Asada, M.: Protosymbols that integrate recognition and response. In: EpiRob-2004: Fourth International Workshop on Epigenetic Robotics, Genoa, Italy (2004)
17. MacDorman, K., Chalodhorn, R., Asada, M.: Periodic nonlinear principal component neural networks for humanoid motion segmentation, generalization, and generation. In: ICPR-2004: International Conference on Pattern Recognition, Cambridge, UK (2004)
18. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, Massachusetts (1998)