

Periodic Nonlinear Principal Component Neural Networks for Humanoid Motion Segmentation, Generalization, and Generation

Karl F. MacDorman, Rawichote Chalodhorn, 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@ams.eng.osaka-u.ac.jp

Abstract

In an experiment with a soccer playing robot, periodic temporally-constrained nonlinear principal component neural networks (NLPCNNs) are shown to characterize humanoid motion effectively by exploiting fundamental sensorimotor relationships. Each network learns a periodic or transitional trajectory in a phase space of possible actions, and thus abstracts a kind of protosymbol. NLPCNNs can play a key role in a system that learns to imitate people, enabling a robot to recognize the behavior of others because it has grounded that behavior in terms of its own bodily movements.

1. Introduction

Human motion is fundamentally periodic. While walking is the prototypical example, virtually any human activity involves periodicity — including the unique series of bodily movements that begins with getting up in the morning and ends with retiring at night — because the skeletal structure deviates from and returns to a given configuration. Representational forms that best match the inherent sensorimotor constraints of biological systems, such as periodicity and spatiotemporal continuity, produce more appropriate abstractions with less computation than representational forms that do not.

Nonlinear principal component neural networks (NLPCNNs), augmented with periodic and temporal constraints, provide an effective means of characterizing the motions of humans and other animals. 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 also learn transitions between motion patterns. Once augmented with distributed regulators that set up appropriate flow vectors, learned trajectories correspond to basins of attraction in a phase space of possible actions.

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. Section 5 discusses how NLPCNNs fit into a system that imitates human behavior as an integral part of the “mimesis loop.”

2. A periodic nonlinear principal component neural network

The human body has 244 degrees of freedom [13] 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., [9]). 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

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

resulting in P principal components $[y_1 \cdots y_p] = f(\vec{x})$. Kramer [5] first solved this problem by training a multilayer perceptron similar to the one shown in Figure 1 using the backpropagation of error, although a second order method such as conjugant gradient analysis converges to a solution

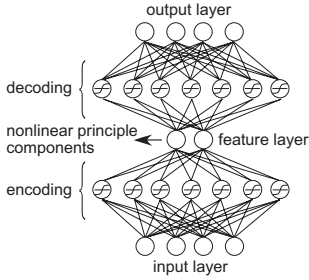


Figure 1. Target values presented at the output layer of a nonlinear principal component neural network are identical to input values. Nonlinear units comprise the encoding and decoding layers, while either linear or nonlinear units comprise the feature and output layers.

faster for many large data sets. Tatani and Nakamura [12] were the first to apply an NLPCNN to human and humanoid motions, though for dimensionality reduction only.

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-constructed¹ NLPCNN has been shown to minimize error several times more than PCA when reducing to two-to-five dimensions [12].

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 [8]. 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 [10][11] at the feature layer can learn self-intersecting curves and surfaces.

Kirby and Miranda [4] 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:

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

The delta values for backpropagation of the circular node-pair are calculated by the chain rule [4], resulting in the

¹The 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.

update rule

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

at the feature layer.

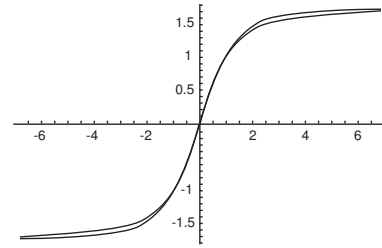


Figure 2. The popular hyperbolic tangent activation function $y \leftarrow 1.7159 \tanh(\frac{2}{3}y)$ can be approximated by a pair of circular nodes where the activation of the second node y_q is fixed at $\sqrt{1.9443}$ and the activation of the first node is calculated accordingly $y_p \leftarrow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

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 [6].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. 2). 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 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

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

³Although 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.

⁴ $k \approx 1.7$ for zero-mean data with variance equal to 1 based on principles discussed in [6].

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

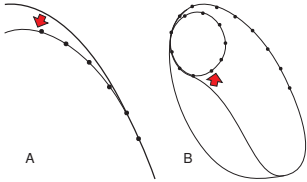


Figure 3. The thick line shows the output of an NLPCNN and the thin line shows the underlying distribution. The dots are data points. **A.** Before learning converges, allowing the network to learn data points despite a high prediction error accelerates learning. **B.** However, after convergence, it leads to segmentation errors.

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.

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

⁶For 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.

```

j ← 1, bucket ← 1, E ← 0
∀x̄i {
  train(networkj, x̄i)
  Ei = ||x̄i - g(f(x̄i))||2, E ← E + Ei
  if (bucket > Bmax ∨
      (learning?(networkj) ∧ E/bucket > Emax) ∨
      Ei > Ei+1)
    j ← j + 1, bucket ← 1, E ← 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. 3A). As learning converges, however, the slack leads to segmentation errors (see Fig. 3B). Therefore, we alter the method of segmentation once the network nears convergence (as determined by Bayesian methods [7] or crossvalidation) so that a network may acquire neighboring points if its prediction error for those points is lower than the network currently assigned to those points.

4. Humanoid experiments

We recorded motion data while a HOAP-1 humanoid robot played soccer in accordance with a hard-coded program. 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. 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

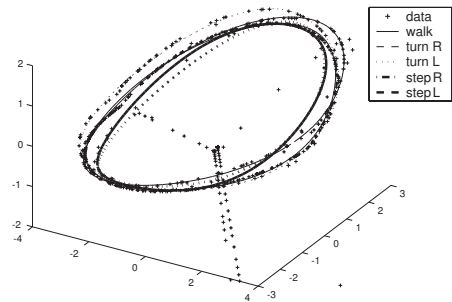


Figure 4. Recognized motion patterns embedded in the dimensions of the first three nonlinear principal components of an NLPCNN.

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 4 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. More detailed experiments and a technique for eliminating redundant segments and their corresponding networks is presented in a companion paper [1].

5. Discussion

An important goal is to develop humanoid robots that are capable of learning to imitate our behavior while they participate in everyday activities. This may be understood as a mimesis loop [3] (see Fig. 5): (1) attentional mechanisms support the robot to recognize the body parts of others, and (2) the robot maps these parts onto its own body. (3) Periodic NLPCNNs characterize the robots own kinematic or dynamic motion patterns. In addition, by mapping the bodies of others to its own, it can use NLPCNNs to recognize their activities. Since each network correspond to a particular type of motion (i.e., an attractor or a transition between attractors in a proprioceptive phase space), it can act as a protosymbol. Thus, the robot is able to recognize the behavior of others because it has grounded their behavior in terms of its own body. (4) 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 (e.g., [2]). Thus, periodic NLPCNNs may be able to play a central role in a system that recognizes, learns, and responds to patterns of behavior. Whether they can segment human data, which has higher variance, is yet to be determined.

References

- [1] R. Chalodhorn, K. MacDorman, and M. Asada. Automatic extraction of abstract actions from humanoid motion data. In *IROS-2004: IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, September 28–October 2 Submitted.
- [2] T. Fujii. A new approach to the LQ design from the viewpoint of the inverse regulator problem. *IEEE Transactions on Automatic Control*, 32(11):995–1004, 1987.

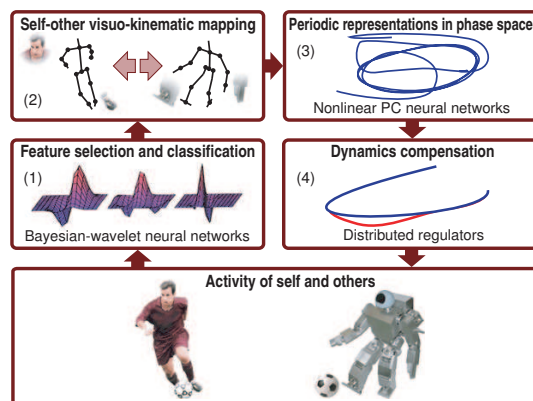


Figure 5. Periodic nonlinear principal component networks may characterize motion patterns in a much larger system for recognizing, learning, and responding behavior.

- [3] T. Inamura, I. Toshima, and Y. Nakamura. Acquiring motion elements for bidirectional computation of motion recognition and generation. In B. Siciliano and P. Dario, editors, *Experimental Robotics VIII*, pages 372–381. Springer, 2003.
- [4] M. J. Kirby and R. Miranda. Circular nodes in neural networks. *Neural Computation*, 8(2):390–402, 1996.
- [5] M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *Journal of the American Institute of Chemical Engineers*, 37(2):233–243, 1991.
- [6] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient BackProp. In G. B. Orr and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, pages 1–44. Springer, 1998.
- [7] D. J. MacKay. Probable networks and plausible predictions: A review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6:469–505, 1995.
- [8] E. C. Malthouse. Limitations of nonlinear PCA as performed with generic neural networks. *IEEE Transactions on Neural Networks*, 9(1):165–173, 1998.
- [9] M. Okada, K. Tatani, and Y. Nakamura. Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion. In *IEEE International Conference on Robotics and Automation*, pages 1410–1415, 2002.
- [10] S. Ridella, S. Rovetta, and R. Zunino. Adaptive internal representation in circular back-propagation networks. *Neural Computing and Applications*, 3(4):222–333, 1995.
- [11] S. Ridella, S. Rovetta, and R. Zunino. Circular backpropagation networks for classification. *IEEE Transaction on Neural Networks*, 8(1):84–97, 1997.
- [12] K. Tatani and Y. Nakamura. 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*, pages 1927–1932, Taipei, Taiwan, September 2003.
- [13] V. M. Zatsiorsky. *Kinematics of Human Motion*. Human Kinetics, Urbana Champaign, 2002.