

# Using Eigenposes for Lossless Periodic Human Motion Imitation

Rawichote Chalodhorn and Rajesh P. N. Rao

**Abstract**—Programming a humanoid robot to perform an action that takes the robot’s complex dynamics into account is a challenging problem. Traditional approaches typically require highly accurate prior knowledge of the robot’s dynamics and environment in order to devise complex control algorithms for generating a stable dynamic motion. Training using human motion capture is an intuitive and flexible approach to programming a robot but directly applying motion capture data to a robot usually results in dynamically unstable motion. Optimization using high-dimensional motion capture data in the humanoid full-body joint-space is also typically intractable. In previous work, we proposed an approach that uses dimensionality reduction to achieve tractable imitation-based learning in humanoids without the need for a physics-based dynamics model. This work was based on a 3-D “eigenpose” representation. However, for some motion patterns, using only three dimensions for eigenposes is insufficient. In this paper, we propose a new method for motion optimization based on high-dimensional eigenpose data. A one-dimensional computationally efficient motion-phase optimization method is implemented along with a newly developed cylindrical coordinate transformation technique for hyperdimensional subspaces. This results in a fast learning algorithm and very accurate motion imitation. We demonstrate the new algorithm on a Fujitsu HOAP-2 humanoid robot model in a dynamic simulator and show that a dynamically stable sidestep motion can be successfully learned by imitating a human demonstrator.

## I. INTRODUCTION

Imitation is an important learning mechanism in many biological systems including humans [1]. Learning through imitation is a powerful and versatile method for acquiring new behaviors. In humans, a wide range of behaviors, from styles of social interaction to tool use, are passed from one generation to another through imitative learning. The potential for rapid behavior acquisition through demonstration has made imitation learning an increasingly attractive alternative to manually programming robots. Demiris and Hayes [2] introduced the concept of imitative learning by demonstrating a wheeled mobile robot that learned to solve a maze problem by imitating another homologous robot. Billard [3] showed that imitation is a mechanism that allows the robot imitator to share a similar set of proprio- and exteroceptions with teacher. Ijspeert et al. [4] designed a nonlinear dynamical system to imitate the trajectories of joints and end-effectors of a human teacher. The robot learned and performed tennis swing motions by imitation. The resulting motions were robust against dynamic perturbation. Our work aims to solve a different challenging dynamics problem, namely, learning

Rawichote Chalodhorn and Rajesh P. N. Rao are with Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195-2350, USA {choppy, rao}@cs.washington.edu

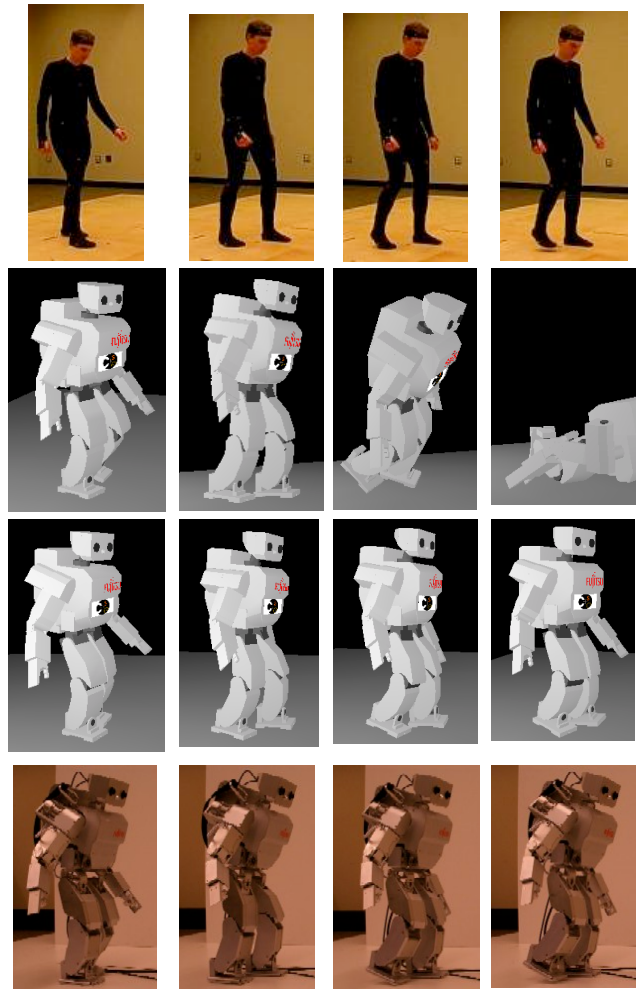


Fig. 1. Learning to walk through imitation. The pictures in the first row show a human subject demonstrating a walking gait in a motion capture system. The second row shows simulation results for this motion before learning. The third row shows simulation results after learning. The last row shows results obtained on the real robot.

bipedal locomotion in a humanoid robot based on imitating a human.

It is straightforward to recover kinematic information from human motion using, for example, a motion capture (mocap) system, but imitating this motion with stable robot dynamics is a much harder problem. A human demonstrator and a humanoid robot learner may share similar kinematic structure but their dynamics are usually very different. If the robot imitates the exact kinematic movement of human, the resulting motion is typically dynamically unstable as shown in the second row of Fig. 1.

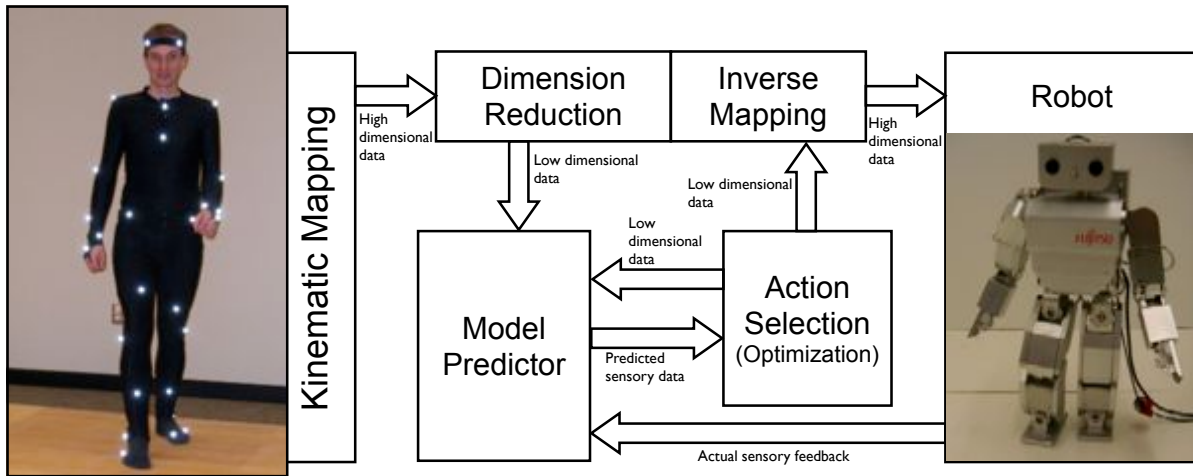


Fig. 2. A framework for learning human behavior by imitation through sensory-motor mapping in reduced dimensional spaces.

Traditional model-based approaches based on zero-moment point (ZMP) [5], [6], [7] or the inverted pendulum model [8], [9] require a highly accurate model of robot dynamics and the environment in order to achieve a stable walking gait. The mimesis theory of [10] is based on action acquisition and action symbol generation. This method also relies on a model-based approach, the so-called dynamic filter, to stabilize motion. Learning-based approaches such as reinforcement learning [11] are more flexible and can adapt to environmental change but such methods are typically not directly applicable to humanoid robots due to the curse of dimensionality problem engendered by the high dimensionality of the full-body joint space of the robot. Nonlinear dimensionality reduction algorithms had previously been applied to representation of human posture [12], [13]. Tatani and Nakamura [14] explored using a low-dimensional subspace to kinematically reproduce human motion on a humanoid robot via non-linear principal components analysis (NLPCA) [15]. However, these methods have some parameters that have to be well-tuned. Properties of the resulting low-dimensional subspaces used in these algorithms have not been well studied. Principal components analysis (PCA) is a linear dimensionality reduction technique whose properties have been well studied. We utilize PCA for the motion learning in our framework.

We previously introduced an approach for imitation learning that uses a compact low-dimensional representation of whole-body humanoid posture for optimizing the robot's dynamics [16], [17], [18]. The method successfully learned to imitate a human walking gait from mocap data. The approach is depicted in Fig. 2. Human mocap data are first mapped to the robot body. Motion data in robot joint space are transformed into a small number of orthogonal axes using linear principal component analysis (PCA). The motion data represented using PCA are called *eigenpose* data. For periodic motion, the eigenpose data are transformed again into a cylindrical coordinate frame. The eigenpose data in the cylindrical coordinate system are then modeled

as a single parameter closed-curve function called *action subspace embedding* (reviewed in section III-B). The single parameter  $\varphi$  of the action subspace embedding is defined to be the *motion-phase angle*. The motion-phase angle is the parameter that has direct effect on the timing of movement during motion.

At the beginning of the learning process, sensory feedback is recorded from motion produced via the eigenpose commands obtained from mocap data. Generally, this motion is not stable enough to begin the learning process. In order to be able to get a complete cycle of stable motion to begin the learning, the unstable motion is scaled down by *action subspace scaling* (reviewed in section III-C). The motion is scaled down until a complete cycle of stable motion is found. After sensory feedback from the first motion trial is obtained, a model predictor is trained to predict sensory feedback for the next time step based on the history of eigenpose commands and past sensory feedback. Using the model predictor, the eigenpose commands for a complete motion cycle are planned subject to the objective function of the motion. An optimal action plan can only be obtained when sensory prediction is accurate. The model predictor can deliver more accurate predictions only when it has sufficient history information of actions and feedback. Thus, an action plan for a complete motion cycle is achieved after a sufficient number of learning trials. Our previous work [18] demonstrated that the proposed approach allows a humanoid robot to learn to walk based solely on human motion capture without the need for a detailed physical model of the robot (Fig. 1).

In our previous work [16], [17], [18], [19], the low-dimensional eigenpose data used for motion learning were three dimensional. This 3-D data is convenient for visualizing and developing a motion optimization algorithm through analytical geometry. Periodic motion patterns such as walking gaits were successfully learned by the algorithm using 3-D eigenposes. However, for some motion patterns, using only three dimensions for eigenposes cannot preserve

the significant characteristics of postures, which is crucial for learning. In this paper, we extend our motion learning algorithm to be applicable to any number of dimensions of the eigenpose data. We develop a cylindrical coordinate transformation technique for hyperdimensional space for use with the one-dimensional *action subspace embedding* motion representation. We demonstrate the new algorithm by showing imitation learning of human side-step motion using 20-dimensional eigenpose data, which achieve 100 % accuracy of original posture data. The result is demonstrated on a HOAP-2 humanoid robot model in the commercial dynamic simulator [20].

## II. HUMAN MOTION CAPTURE

### A. Kinematic Mapping

We first need to solve the correspondence problem which, in our case, is kinematic mapping of whole-body postures between a human demonstrator and a Fujitsu HOAP-2 humanoid robot. The human subject and the robot share similar humanoid appearances, but their kinematic structures (skeletal models) are dissimilar. Initially, a set of markers is attached to the human subject and the 3-D positions of these markers are recorded for each pose during motion. We use a Vicon optical system running at 120Hz and a set of 41 reflective markers. These recorded marker positions provide a set of Cartesian points in the 3D capture volume for each pose. To obtain the robot's poses, the marker positions are then assigned as positional constraints on the robot's skeleton to derive the joint angles using standard numerical IK routines of Vicon system. For example, the shoulders were replaced with three distinct 1-dimensional rotating joints rather than one 3-dimensional human ball joint. The IK routine then directly generates the desired joint angles on the robot skeleton for each pose. A limitation of this technique is that we can only demonstrate actions that the target robot can perform. For example, we avoid using toes in our demonstrated walking gait.

### B. Human sidestep motion

A motion capture session of a human demonstrator performing a sidestep motion is shown in Fig. 3. In the figure a human demonstrator was sidestepping to the right. The motion sequence can be divided into four major phases, which we term *swing-off*, *landing*, *take-off* and *swing-in* (see Fig. 3 and caption). After the kinematic mapping process in section II-A was applied to the mocap data, 20 dimensions of joint angle data were obtained<sup>1</sup>. Subsequently, the joint angle data were transformed into orthogonal principal axes using PCA.

The reconstruction accuracy as a function of the number of principal components is plotted in Fig. 4. When the first three principal axes are used, only 71.92% reconstruction accuracy for the original joint angle data is achieved. More than 98% accuracy can be recovered when more than 10 dimensional

<sup>1</sup>Note that the Fujitsu HOAP-2 robot has 25 joints but out of these, two neck joints, two hand joints, and one torso joint are not used in the motion patterns studied in this paper.

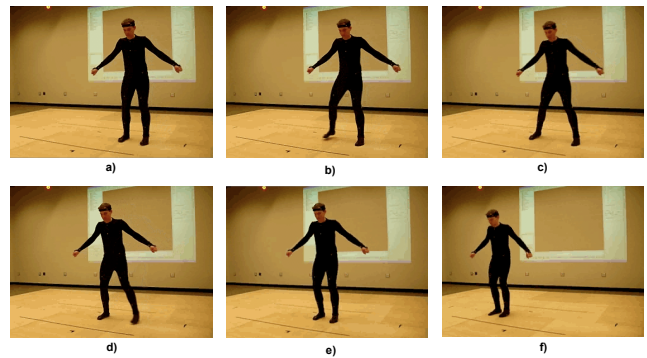


Fig. 3. Motion capture session of sidestep motion. Six samples of a right-hand side sidestep motion sequence are shown in subfigures a) through f). The motion sequence can be divided into four major steps starting from a standing posture in subfigure a). First, the right leg swings off as in subfigure b). Second, the right leg lands to the ground as in subfigure c). Third, the left leg takes off the ground as in subfigure d). Fourth, the left leg swings in toward the right leg as in subfigure e). Finally, we have the standing posture in subfigure f). Note that the subfigure f) is a standing posture after one more sidestep motion cycle has been performed. A sidestepping cycle takes about 1 second. The sidestep motion shall be defined in four phases labeled as *swing-off*, *landing*, *take-off* and *swing-in*.

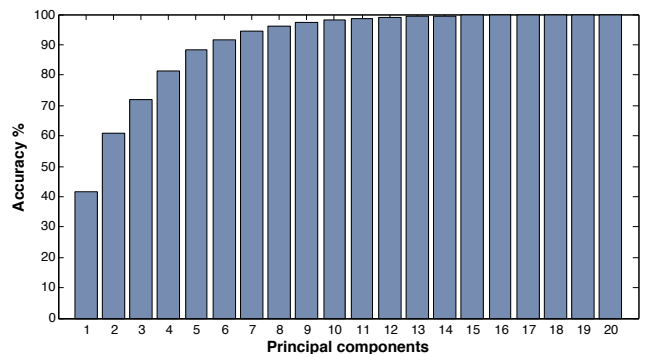


Fig. 4. Reconstruction accuracy as a function of number of principal components of the sidestep motion in figure 3.

eigenposes are used. 100% accuracy can be obtained only when all 20-dimensions are used. In this paper, we illustrate the approach using 20-dimensional eigenposes.

Three-dimensional eigenpose data for the first three principal components of the sidestep motion are plotted in Fig. 5 (black diamonds). The pattern of blue dots in Fig. 5 is the sidestep motion pattern when a scaling factor 0.5 was applied (described in section III-C). This scaling factor was found to be stable enough to begin the learning process. The phase-motion optimization was performed on this 0.5-scaled pattern.

## III. REVIEW OF 3-D EIGENPOSES

### A. Eigenposes as Low-Dimensional Representation of Postures

Particular classes of motion such as walking, sidestepping, or reaching for an object are intrinsically low-dimensional. We apply linear PCA to parameterize the low-dimensional motion subspace  $\mathbb{X}$ . Vectors in the high-dimensional space

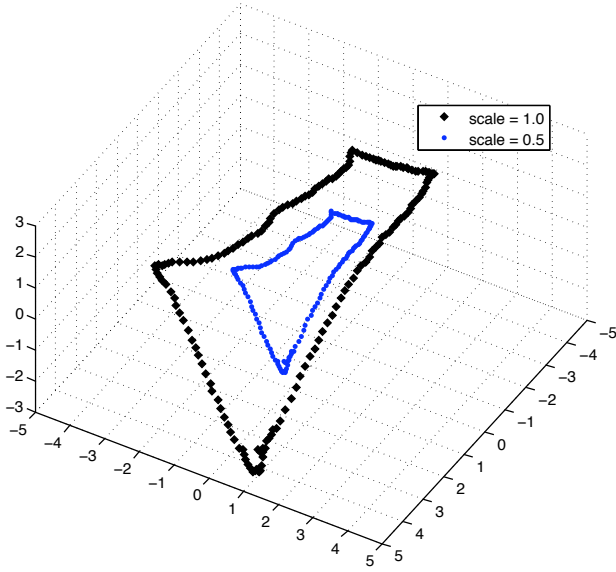


Fig. 5. First three dimensions of sidestep eigenpose data at scale 1.0 and 0.5.

are mapped to the low-dimensional space by multiplication with the transformation matrix  $\mathbf{C}$ . The rows of  $\mathbf{C}$  consist of the eigenvectors, computed via eigenvalue decomposition of the motion covariance matrix. This results in transformed vectors whose components are uncorrelated and ordered according to the magnitude of their variance.

For example, let  $\Theta$  be the  $20 \times 1$  vector of joint angles (the high-dimensional space) and  $\mathbf{x}$  be the  $3 \times 1$  vector in low dimensional 3D space. We can calculate  $\mathbf{x}$  in 3D space by using  $\mathbf{p} = \mathbf{C}\Theta$ , where  $\mathbf{p}$  is a  $20 \times 1$  vector of all principal component coefficients of  $\Theta$  and  $\mathbf{C}$  is the  $20 \times 20$  PCA transformation matrix. We then pick the first three elements of  $\mathbf{p}$  (corresponding to the first three principal components) to be  $\mathbf{x}$ . The inverse mapping  $\hat{\Theta}$  which is an estimation of  $\Theta$  from PCA can be computed by  $\hat{\Theta} = \mathbf{C}^{-1}\tilde{\mathbf{p}}$  when the first three components of a full-rank-vector  $\tilde{\mathbf{p}}$  are the elements of  $\mathbf{x}$  and the rest of the elements are zero.

### B. Action Subspace Embedding

High-level control of the humanoid robot reduces to selecting a desired angle for each joint servo motor. As discussed previously, optimization in the full space of all robot joint angles is typically intractable. We leverage the redundancy of the full posture space and use the reduced dimensional subspace  $\mathbb{X}$  to constrain target postures. Any desired posture (also referred to as an action) can be represented by a point  $\mathbf{a} \in \mathbb{X}$ .

A periodic movement such as walking or sidestep can be represented by a loop in  $\mathbb{X}$  as shown in Figure 6. In the general case, we consider a non-linear manifold representing the action space  $\mathbf{A} \subseteq \mathbb{X}$ . Non-linear parameterization of the action space allows further reduction in dimensionality. We embed a one dimensional representation of the original motion in the three dimensional posture space and use

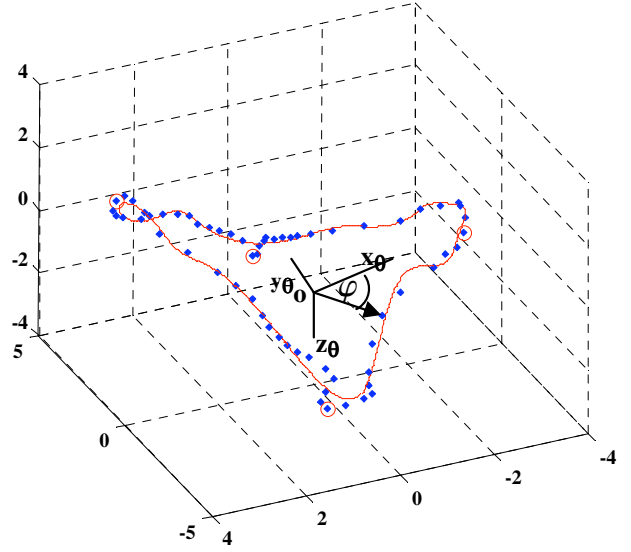


Fig. 6. Embedded action space of a humanoid walking gait. Training data points in the reduced posture space (shown in blue-dots) are converted to a cylindrical coordinate frame relative to the coordinate frame  $\mathbf{x}_\theta, \mathbf{y}_\theta, \mathbf{z}_\theta$ . The points are then represented as a function of the phase angle  $\varphi$ , which forms an embedded action space (shown in red-solid-curve). This action space represents a single gait cycle.

it for constructing a constrained search space for motion optimization. Using the feature representation of the set of initial training examples  $\mathbf{x}^i \subseteq \mathbb{X}$ , we first convert each point to its representation in a cylindrical coordinate frame. This is done by establishing a coordinate frame with three basis directions  $\mathbf{x}_\theta, \mathbf{y}_\theta, \mathbf{z}_\theta$  in the feature space. The zero point of the coordinate frame is the empirical mean of the data points in the reduced space. We recenter the data around this new zero point and denote the resulting data  $\hat{\mathbf{x}}^i$ .

We then compute the principal axis of rotation  $\mathbf{z}_\theta$  :

$$\mathbf{z}_\theta = \frac{\sum_i (\hat{\mathbf{x}}^i \times \hat{\mathbf{x}}^{i+1})}{\|\sum_i (\hat{\mathbf{x}}^i \times \hat{\mathbf{x}}^{i+1})\|} \quad (1)$$

Next,  $\mathbf{x}_\theta$  is chosen to align with the maximal variance of  $\mathbf{x}^i$  in a plane orthogonal to  $\mathbf{z}_\theta$ . Finally,  $\mathbf{y}_\theta$  is specified as orthogonal to  $\mathbf{x}_\theta$  and  $\mathbf{z}_\theta$ . The final embedded training data is obtained by cylindrical conversion to  $(\varphi, r, h)$  where  $r$  is the radial distance,  $h$  is the height above the  $\mathbf{x}_\theta - \mathbf{y}_\theta$ , and  $\varphi$  is the angle in  $\mathbf{x}_\theta - \mathbf{y}_\theta$  plane measured counter-clockwise from  $\mathbf{x}_\theta$ .

Given the loop topology of the latent training points, one can parameterize  $r$  and  $h$  as a function of  $\varphi$ . The embedded action space is represented by a learned approximation of the function:

$$[r, h] = g(\varphi) \quad (2)$$

where  $0 \leq \varphi \leq 2\pi$ . Approximation of this function is performed using a closed-curve cubic spline method. The angle  $\varphi$  can also be interpreted as the *motion-phase angle* because when  $\varphi$  sweeps from 0 to  $2\pi$ , it creates an action subspace  $\mathbf{A}$ , which in our case is a walking gait. The

parameter  $\varphi$  indicates how far the current posture is from the beginning of the motion cycle. The first order time derivative of  $\varphi$  also tells us the speed of movement.

### C. Action Subspace Scaling

A high-dimensional joint angles data set is usually pre-processed before dimensional reduction. In our case, we performed data normalization which makes the raw data have zero mean and unity standard deviation. Normalization ensures that the data for each joint (each dimension), which originally vary over different scales of values, are in the same range. When the normalized data is multiplied by a scalar value, the results are similar postures but with different magnitude. On the other hand, multiplying a vector of raw joint angles data by a scalar factor does not yield a similar posture.

From our study of four different motion patterns in the reduced subspace, we found that scaling the motion pattern up and down produces similar motion patterns except for the magnitude of motion. This means that posture scaling ability is preserved after PCA. Thus, multiplication of the action space  $\mathbf{A} \subseteq \mathbb{X}$  by a scalar value yields a similar action. If  $\mathbf{A}$  represents a walking gait, multiplying  $\mathbf{A}$  by a factor  $f > 1$  will result in a similar walking gait but with larger steps. Multiplying  $\mathbf{A}$  by a factor  $f < 1$  results in a walking gait with a smaller step size. Note that scaling of an action space is always performed with respect to the mean value of  $\mathbf{A}$  or the origin of the cylindrical coordinate frame  $(\mathbf{x}_\theta, \mathbf{y}_\theta, \mathbf{z}_\theta)$ . An example of action subspace scaling is shown in Figure 7.

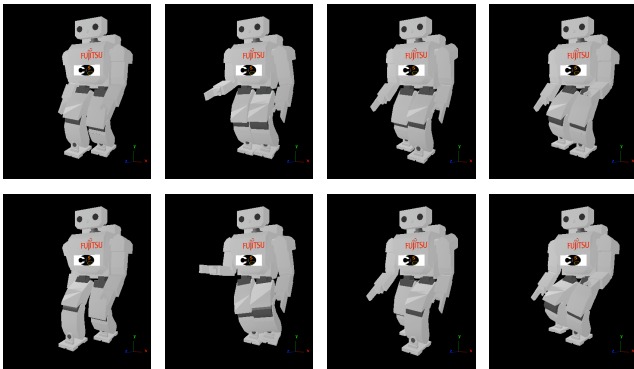


Fig. 7. Motion scaling of a walking gait. The first row of this figure shows four different postures of a walking gait. The second row shows a scaled version of this gait obtained when a multiplying factor  $f = 2.0$  is applied to the low-dimensional representation of the gait.

However, action subspace scaling only produces similarity of kinematic postures. The result of scaling may not be dynamically stable, especially when the scaling factor  $f > 1$ . To achieve stable motion, the new motion has to be gradually learned as originally proposed in [18].

## IV. CYLINDRICAL COORDINATE TRANSFORMATION OF HYPERDIMENSIONAL SUBSPACES

The motion-phase optimization is performed in a cylindrical coordinate system. Transformation of data from a 3-D

Cartesian coordinate system to a 3-D cylindrical coordinate is straightforward. However, that is not the case for transformation of data that have more than three dimensions. In this section, we introduce a new concept of cylindrical coordinate transformation for a hyperdimensional data.

For  $n$ -dimensional function  $f$  when  $n = 3$ , transformation from a Cartesian space  $\mathbb{X}$  to a cylindrical coordinate system  $\Phi$  is the mapping:

$$f(x, y, z) \rightarrow f(\varphi, r, h) \quad (3)$$

where

$$\varphi = \arctan\left(\frac{y}{x}\right), \quad (4)$$

$$r = \sqrt{x^2 + y^2}, \quad (5)$$

and

$$h = z. \quad (6)$$

When  $n > 3$ , a function of  $n$ -dimensions may be written as

$$f(d_1, d_2, d_3, \dots, d_n) \quad (7)$$

where  $d_i$  when  $i = 1, \dots, n$  and  $n > 3$ , represents a variable in an orthogonal axis in  $\mathbb{R}^n$ .

We can also express function (7) as:

$$f(x, y, z_1, \dots, z_{n-2}) \quad (8)$$

where  $z_i$  when  $i = 1, \dots, n - 2$  and  $n > 3$ .

Since the cylindrical coordinate system is a 3-dimensional coordinate system, transformation of a hyperdimensional function  $f$  where  $n > 3$  to cylindrical coordinate system  $\Phi$  is undefined. However, the hyperdimensional function  $f$  can be represented by a set of multiple cylindrical coordinate frames. Suppose that  $f$  is a 5-dimensional function. Then,  $f$  can be expressed in the form of (8) by:

$$f(x, y, z_1, z_2, z_3). \quad (9)$$

Mapping of  $f$  in (9) can be regarded as:

$$\begin{aligned} f(x, y, z_1) & \Rightarrow f(\varphi, r, h_1) \\ f(x, y, z_2) & \Rightarrow f(\varphi, r, h_2) \\ f(x, y, z_3) & \Rightarrow f(\varphi, r, h_3) \end{aligned} \quad (10)$$

where transformation of  $\varphi$  and  $r$  follow (4) and (5), and  $h_1, h_2$  and  $h_3$  are defined as in (6). Thus, mapping of the  $n$  orthogonal dimensions of  $f$  to multiple cylindrical coordinate systems can be defined as:

$$f(x, y, z_1, \dots, z_{n-2}) \rightarrow f(\varphi, r, h_1, \dots, h_{n-2}). \quad (11)$$

For the 20-dimensional sidestep eigenpose data, 18 cylindrical coordinate frames are needed to describe the data-set. The first three cylindrical coordinate frames for the sidestep motion are shown in Fig. 8. Let  $f_{\text{sidestep}}(\varphi, r, h_1, h_2, \dots, h_{18})$  be a function in the cylindrical coordinate systems that describes the sidestep motion. Subfigure a), b), and c) depicts data points of  $f_{\text{sidestep}}(\varphi, r, h_1)$ ,  $f_{\text{sidestep}}(\varphi, r, h_2)$  and  $f_{\text{sidestep}}(\varphi, r, h_3)$  respectively. Notice that subfigure a) in Fig. 8 and the

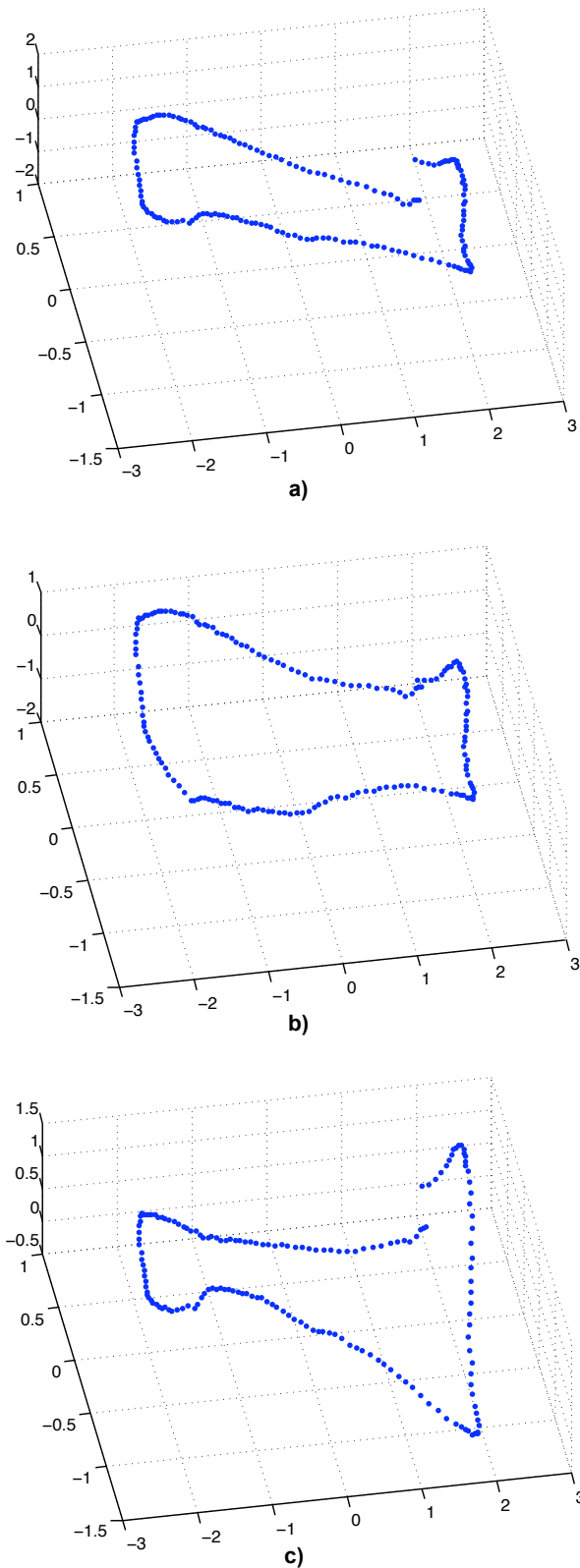


Fig. 8. First three cylindrical coordinates of sidestep hyperdimensional eigenpose data. Subfigure a), b) and c) show coordinate frames  $(\varphi, r, \mathbf{h}_1)$ ,  $(\varphi, r, \mathbf{h}_2)$  and  $(\varphi, r, \mathbf{h}_3)$  respectively.

blue dots in Fig. 5 both represent  $f_{\text{sidestep}}(\varphi, r, h_1)$  but in different perspectives. There is an opened gap in subfigure a) and in subfigure c) because the data pattern was manually segmented from human motion capture data. Even if the data are not a perfect closed-curve pattern, its embedded action subspace is modeled as a closed-curve function.

## V. MOTION-PHASE OPTIMIZATION OF HYPERDIMENSIONAL EIGENPOSES

To avoid the curse of dimensionality problem, we use the one-dimensional optimization of motion-phase angle of the 3-D eigenposes [16] but extended for the hyperdimensional case. For the 3-D case, the action subspace embedding is a single parameter function of motion-phase angle  $\varphi$  that is mapped to values of the radius  $r$  and the height  $h$  of a periodic motion pattern in a cylindrical coordinate system  $\Phi$ . For a hyperdimensional case of  $n$ -dimensional eigenpose data, the action subspace embedding is a single parameter function of motion-phase angle  $\varphi$  that maps to values of  $r, h_1, h_2, \dots, h_{n-2}$  of a periodic motion pattern. For example, in the case of the sidestep motion pattern, the action subspace embedding is one-to-nineteen mapping:

$$[r, h_1, h_2, \dots, h_{18}] = g(\varphi). \quad (12)$$

We use a nonlinear autoregressive network with exogenous inputs (NARX) [21] [22] for prediction of the gyroscope signal instead of the time-delay RBF network used in our previous work ([16], [17], [18]). The NARX network is a recurrent dynamic network with feedback connections enclosing several layers of the network. The NARX model is based on the linear ARX model, which is commonly used in time-series modeling. The NARX model for predicting gyroscope signal from motion-phase angle input is:

$$\omega_{t+1} = F(\omega_t, \varphi_t, \omega_{t-1}, \varphi_{t-1}), \quad (13)$$

where  $\omega$  is the three dimensional vector of gyroscope signals along  $x, y, z$  axes. Block diagram of the NARX predictor is shown in Fig. 9.

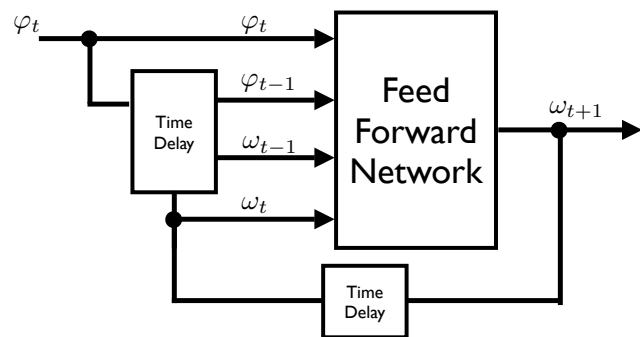


Fig. 9. NARX predictor for motion-phase optimization.

We define an objective function that indicates torso stability using the following function of gyroscope signals:

$$\Gamma(\omega) = \lambda_x \omega_x^2 + \lambda_y \omega_y^2 + \lambda_z \omega_z^2, \quad (14)$$

where  $\omega_x, \omega_y, \omega_z$  refer to gyroscope signals along the  $x, y, z$  axes respectively. The constants  $\lambda_x, \lambda_y, \lambda_z$  allow one to weight rotation in each axis differently. Assuming that the starting posture is statically stable, one may simply minimize overall rotation of the robot body during motion to maintain balance by minimizing the sum of squares of gyroscope signals. Thus, the objective function (14) provides a measure of stability of the robot's posture during motion.

The embedded action in (12) and the NARX predictor model in (13) can be directly applied to the motion-phase optimization:

$$\varphi_t^* = \arg \min_{\varphi_t} \Gamma(F(\omega_t, \omega_{t-1}, \varphi_t, \varphi_{t-1})). \quad (15)$$

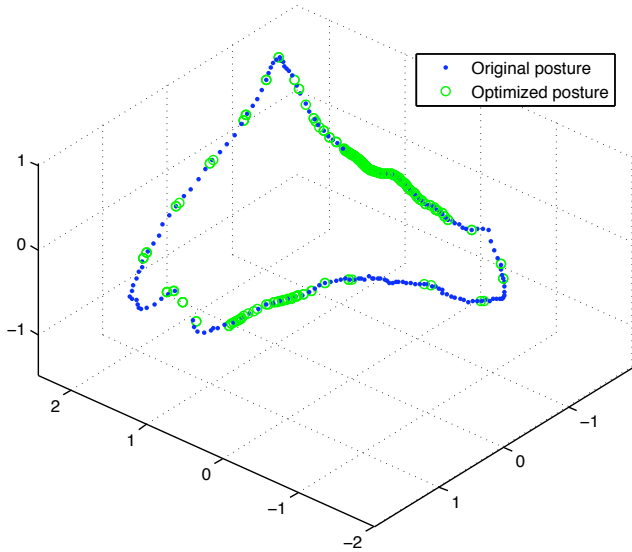


Fig. 10. Motion-phase angle optimization result of sidestep hyperdimensional eigenpose data.

## VI. SIMULATION RESULTS

Fig. 10 shows the optimization result in a 3-D coordinate frame defined by the first three principal axes after five learning episodes. The optimized eigenposes are points on the original motion pattern but are distributed differently from the original pattern because the motion-phase optimization is one-dimensional optimization of the parameter  $\varphi$  of (12). The optimized eigenposes are thus strictly constrained to be within the original set of postures. Only timing of postures during the motion has been altered. Notice in the figure that the opened gap of the original pattern is now closed by the optimized postures. There are two reasons for this phenomena. First, the action subspace embedding or the constraint pattern is modeled as a closed-curve. Second, based on sensory feedback received during learning episodes, the optimization algorithm found that it can achieve a lower gyroscope signal oscillation by assigning some postures

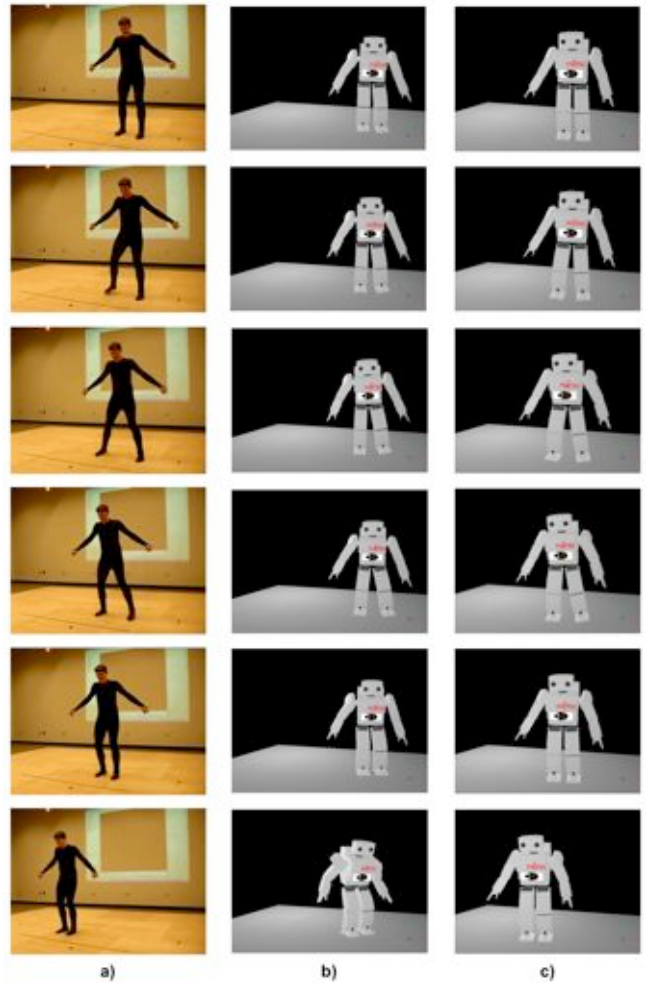


Fig. 11. Simulation result of sidestep hyperdimensional eigenpose data optimization. Subfigures in column a) show original sidestep motion sequence of the human demonstrator. Subfigures in column b) show the sidestep motion sequence on HOAP-2 robot in a dynamics simulator without optimization at the motion scale 0.5. Note the undesired rotation and lack of displacement after executing the motion. Subfigures in column c) show the sidestep motion after five learning episodes at motion scale 0.5. Robust sidestepping motion is achieved.

in the opened gap, resulting in smoother motion. Another attempt of the algorithm to obtain smoother movement can be noticed in the lower-left corner of the pattern in Fig. 10: the algorithm decided to plan the trajectory across an irregular corner of the original pattern.

Simulation results of the sidestep motion are shown in Fig. 11. Column a) in Fig. 11 shows the original sidestep motion sequence of a human demonstrator. Column b) shows the HOAP-2 robot performing the sidestep motion sequence at motion scale 0.5 without optimization in a dynamics simulator. Column c) shows the sidestep motion after five learning episodes at motion scale 0.5. In Fig. 11, the first row and the last row of subfigures in every column are the standing postures at the beginning and the end of the motion sequence, respectively. The second row is the swing-off phase. The third row is the landing phase. The fourth row is the take-off phase, and the fifth row is the swing-

in phase. In column b), the right foot and the left foot of the robot bounced at the landing phase and the take-off phase, causing the robot to be unable to lift its left foot up in the subsequent take-off phase. As a result, the robot dragged its left foot along the ground during the swing-in period, causing the body of the robot to turn as can be observed from the last two images of column b). In column c), which shows the optimized result, the robot could perform the sidestep motion without this undesired turn of the body. While all of the key postures in the figures of column c) look very similar to the postures in column a), the timing of the movements are significantly different. The landing phase and the swing-in phase of the optimized motion in column c) are relatively slower than the original human motion. These can also be observed in Fig. 10 where there are two parts of the motion pattern containing a high density of optimized postures. These correspond to a slow landing phase and swing-in phase. The slow landing phase and swing-in phase also prevent the robot from dragging its left foot on the ground, eliminating the undesired turn of the body and allowing the robot to learn the sidestep motion successfully.

## VII. CONCLUSION

This paper proposes a new method for learning by imitation that generalizes eigenpose-based motion optimization to a large number of dimensions. A newly developed cylindrical coordinate transformation technique for hyperdimensional subspaces is used in conjunction with a one-dimensional computationally efficient motion-phase optimization method. The resulting framework allows fast learning and accurate motion imitation. Results involving a Fujitsu HOAP-2 humanoid robot model in a dynamic simulator show that a dynamically stable sidestep motion can be rapidly learned from a human demonstrator. Future work will focus on scaling the approach to a wide class of motions, including non-periodic motion, and integrating real-time feedback with the learned motions on the actual humanoid robot.

## VIII. ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research (ONR), National Science Foundation (NSF) grant 0622252, and a Packard Fellowship to RPNR.

## REFERENCES

- [1] R. P. N. Rao, A. P. Shon, and A. N. Meltzoff, "A Bayesian model of imitation in infants and robots," in *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, C. L. Nehaniv and K. Dautenhahn, Eds. Cambridge University Press, UK, 2007.
- [2] J. Demiris and G. Hayes, "A robot controller using learning by imitation," in *Proceedings of the 2nd International Symposium on Intelligent Robotic Systems*, Grenoble, France, 1994.
- [3] A. Billard, *Imitation: a means to enhance learning of a synthetic protolanguage in autonomous robots*. Cambridge, MA, USA: MIT Press, 2002, pp. 281–310.
- [4] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Trajectory formation for imitation with nonlinear dynamical systems," in *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 752–757.

- [5] M. Vukobratović and S. Yu., "On the stability of anthropomorphic systems," in *Mathematical Biosciences*, vol. 15, October 1972, pp. 1–37.
- [6] M. Sobotka, D. Wollherr, and M. Buss, "A Jacobian method for online modification of precalculated gait trajectories," in *Proceedings of the 6th International Conference on Climbing and Walking Robots*, Catania, Italy, 2003, pp. 435–442.
- [7] D. Wollherr, M. Buss, M. Hardt, and O. von Stryk, "Research and development towards an autonomous biped walking robot," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics AIM2003*, Kobe, Japan, 2003, pp. 968–973.
- [8] S. Kajita, O. Matsumoto, and M. Saigo, "Real-time 3D walking pattern generation for a biped robot with telescopic legs," in *Proceedings of the 2001 IEEE International Conference on Robotics and Automation, ICRA 2001, May 21-26, 2001, Seoul, Korea*, 2001, pp. 2299–2306.
- [9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA 2003, September 14-19, 2003, Taipei, Taiwan*, 2003, pp. 1620–1626.
- [10] T. Inamura, I. Toshima, and Y. Nakamura, "Acquisition and embodiment of motion elements in closed mimesis loop," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, May 11-15, 2002, Washington, DC, USA*, 2002, pp. 1539–1544.
- [11] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
- [12] R. Chalodhorn, K. F. MacDorman, and M. Asada, "An algorithm that recognizes and reproduces distinct types of humanoid motion based on periodically-constrained nonlinear pca," in *RobuCup*, ser. Lecture Notes in Computer Science, D. Nardi, M. Riedmiller, C. Sammut, and J. Santos-Victor, Eds., vol. 3276. Springer, 2004, pp. 370–380.
- [13] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic, "Style-based inverse kinematics," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 522–531, 2004.
- [14] K. Tatani and Y. Nakamura, "Dimensionality reduction and reproduction with hierarchical nlpc neural networks-extracting common space of multiple humanoid motion patterns," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA 2003, September 14-19, 2003, Taipei, Taiwan*, 2003, pp. 1927–1932.
- [15] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *Journal of the American Institute of Chemical Engineers*, vol. 37, no. 2, pp. 233–243, 1991.
- [16] R. Chalodhorn, D. B. Grimes, G. Maganis, and R. P. N. Rao, "Learning dynamic humanoid motion using predictive control in low dimensional subspaces," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on, Humanoids 2005, December 5-7, 2005, Tsukuba, Japan*, 2005, pp. 214–219.
- [17] R. Chalodhorn, D. B. Grimes, G. Y. Maganis, R. P. N. Rao, and M. Asada, "Learning humanoid motion dynamics through sensory-motor mapping," in *Proceedings of the IEEE International Conference of Robotics & Automation (ICRA'06)*. San Francisco, CA: IEEE Press, 2006, pp. 3693 – 3698.
- [18] R. Chalodhorn, D. B. Grimes, K. Grochow, and R. P. N. Rao, "Learning to walk through imitation," in *IJCAI*, M. M. Veloso, Ed., 2007, pp. 2084–2090.
- [19] D. B. Grimes, R. Chalodhorn, and R. P. N. Rao, "Dynamic imitation in a humanoid robot through nonparametric probabilistic inference," in *Robotics: Science and Systems*, G. S. Sukhatme, S. Schaal, W. Burgard, and D. Fox, Eds. The MIT Press, 2006.
- [20] O. Michel, "Webots: Symbiosis between virtual and real mobile robots," in *VW '98: Proceedings of the First International Conference on Virtual Worlds*. London, UK: Springer-Verlag, 1998, pp. 254–263.
- [21] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for non-linear systems Part I: deterministic non-linear systems," *International Journal of Control*, vol. 41, no. 2, pp. 303–328, 1985.
- [22] —, "Input-output parametric models for non-linear systems Part II: stochastic non-linear systems," *International Journal of Control*, vol. 41, no. 2, pp. 329–344, 1985.