Learning to Walk through Imitation

Rawichote Chalodhorn, David B. Grimes, Keith Grochow, and Rajesh P. N. Rao
Neural Systems Laboratory
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350 U.S.A.
E-mail: {choppy,grimes,keithg,rao}@cs.washington.edu

Abstract

Programming a humanoid robot to walk is a challenging problem in robotics. Traditional approaches rely heavily on prior knowledge of the robot's physical parameters to devise sophisticated control algorithms for generating a stable gait. In this paper, we provide, to our knowledge, the first demonstration that a humanoid robot can learn to walk directly by imitating a human gait obtained from motion capture (mocap) data. Training using human motion capture is an intuitive and flexible approach to programming a robot but direct usage of mocap data usually results in dynamically unstable motion. Furthermore, optimization using mocap data in the humanoid full-body joint-space is typically intractable. We propose a new model-free approach to tractable imitation-based learning in humanoids. We represent kinematic information from human motion capture in a low dimensional subspace and map motor commands in this low-dimensional space to sensory feedback to learn a predictive dynamic model. This model is used within an optimization framework to estimate optimal motor commands that satisfy the initial kinematic constraints as best as possible while at the same time generating dynamically stable motion. We demonstrate the viability of our approach by providing examples of dynamically stable walking learned from mocap data using both a simulator and a real humanoid robot.

1 Introduction

Imitation is an important learning mechanism in many biological systems including humans [Rao and Meltzoff, 2003]. It is easy to recover kinematic information from human motion using, for example, motion capture, but imitating the motion with stable robot dynamics is a challenging research problem. Traditional model-based approaches based on zero-moment point (ZMP) [Vukobratovic and Borovac, 2004], [Kajita and Tani, 1996] or the inverted pendulum model [Yamaguchi et al., 1996] require a highly accurate model of robot dynamics and the environment in order to achieve a stable walking gait. Learning approaches such as reinforcement learning [Sutton and Barto, 1998] 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.

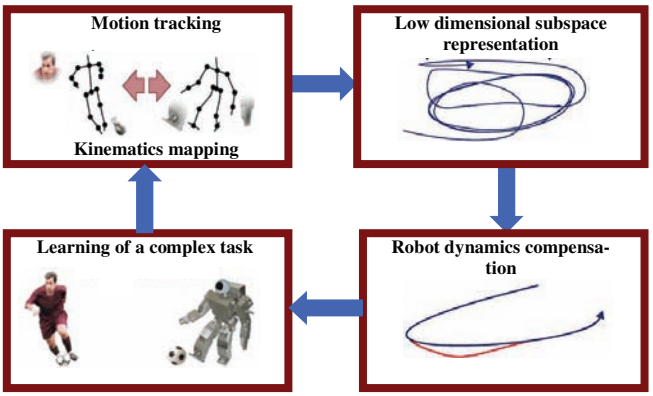

Figure 1. A framework for learning human behavior by imitation through sensory-motor mapping in reduced dimensional spaces.

In this paper, we propose a model-free approach to achieving stable gait acquisition in humanoid robots via imitation. The framework for our method is shown in Figure 1. First, a motion capture system transforms Cartesian position of markers attached to the human body to joint angles based on kinematic relationships between the human and robot bodies. Then, we employ dimensionality reduction to represent posture information in a compact low-dimensional subspace. Optimization of whole-body robot dynamics to match human motion is performed in the low dimensional space. In particular, sensory feedback data are recorded from the robot during motion and a causal relationship between actions in the low dimensional posture space and the expected sensory feedback is learned. This learned sensory-motor mapping allows humanoid motion dynamics to be optimized. An inverse mapping from the reduced space back to the original joint space is then used to generate optimized

motion on the robot. We present results demonstrating 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.

2 Human Motion Capture and Kinematic Mapping

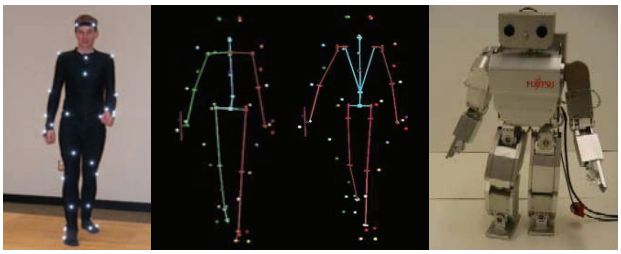

Figure 2. Kinematic mapping used in our approach (from left to right: human body, human skeleton, robot skeleton, and robot body, respectively).

In this paper, we manually map the joint angle data from a motion capture system to a kinematic model for a Fujitsu HOAP-2 humanoid robot. To generate the desired motion sequence for the robot, we capture example motions from a human subject and map these to the joint settings of the robot. 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 final subject poses, the marker positions are then assigned as positional constraints on a character skeleton to derive the joint angles using standard inverse kinematics (IK) routines.

As depicted in Figure 2, in order to generate robot joint angles, we simply replace the human subject's skeleton with a robot skeleton of the same dimensions. For example, the shoulders were replaced with three distinct 1-dimensional rotating joints rather than one 3-dimensional ball joint. The IK routine then directly generates the desired joint angles on the robot for each pose. There are limitations to such a technique (e.g, there may be motions where the robot's joints cannot approximate the human pose in a reasonable way), but since we are interested only in classes of human motion that the robot can handle, this method proved to be a very efficient way to generate large sets of human motion data for robotic imitation.

3 Sensory-Motor Representations

3.1 Low-Dimensional Representation of Postures

Particular classes of motion such as walking, kicking, or reaching for an object are intrinsically low-dimensional. We apply the well known method of principal components analysis (PCA) to parameterize the low-dimensional motion subspace $\mathbb{X}$. Although nonlinear dimensionality reduction methods could be used (e.g., [MacDorman et al., 2004], [Grochow et al., 2004]), we found the standard linear PCA method to be sufficient for the classes of motion studied in this paper.

The result of linear PCA can be thought of as two linear operators $\mathbf{C}$ and $\mathbf{C}^{-1}$ which map from high to low and low to high dimensional spaces respectively. The low dimensional representation of the joint angle space of the HOAP-2 robot executing a walking gait (in the absence of gravity) is shown in Figure 3.

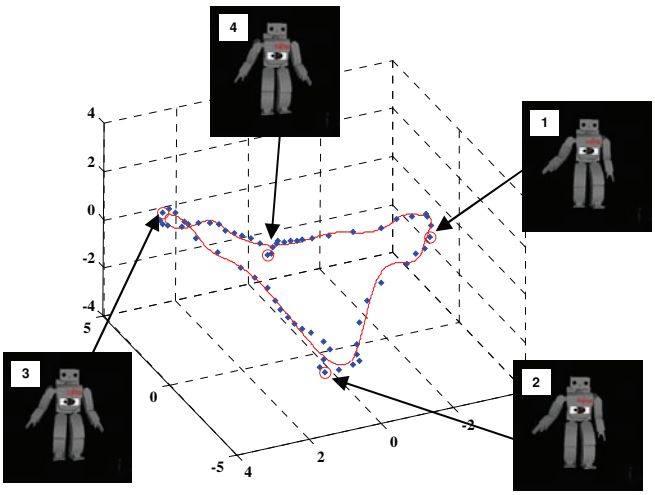

Figure 3. Posture subspace and example poses. A 3-dimensional reduced space representation of the postures of the HOAP-2 robot during a walking motion. We applied linear PCA to 25 dimensions of joint angle data of the robot that mapped from a human kinematic configuration as described in Section 2. Blue diamonds along the function approximated trajectory represent different robot postures during a single walking cycle. Red circles mark various example poses as shown in the numbered images.

We use a straightforward, standard linear PCA method to map between the low and high-dimensional posture spaces. Vectors in the high-dimensional space 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 singular value decomposition (SVD), of the motion covariance matrix. SVD produces transformed vectors whose components are uncorrelated and ordered according to the magnitude of their variance.

For example, let $\mathbf{q} = 21\times1$ vector of joint angles (the high-dimensional space) and $\mathbf{p} = 3\times1$ vector in 3D space. We can calculate $\mathbf{p}$ in 3D space by using $\mathbf{r} = \mathbf{Cq}$, where $\mathbf{r}$ is a 21×1 vector of all principal component coefficients of $\mathbf{q}$ and $\mathbf{C}$ is the 21×21 transformation matrix. We then pick the first three elements of $\mathbf{r}$ (corresponding to the first three principal components) to be $\mathbf{p}$. The inverse mapping can be computed likewise using the pseudo inverse of $\mathbf{C}$.

3.2 Action Subspace Embedding

High-level control of the humanoid robot reduces to selecting a desired angle for each joint servo motor. As discussed previously, operations in the full space of all robot joint angles tend to be 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 is represented by a loop in $\mathbb{X}$ as shown in Figure 4. 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 it for constructing a constrained search space for optimization as discussed in Section 5. Using the feature representation of the set of initial training examples $\mathbf{x}^i = \mathbf{C}\,\mathbf{z}^i$, 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 μ of the data points in the reduced space. We recenter the data around this new zero point and denote the resulting data $\hat{\mathbf{x}}^i$.

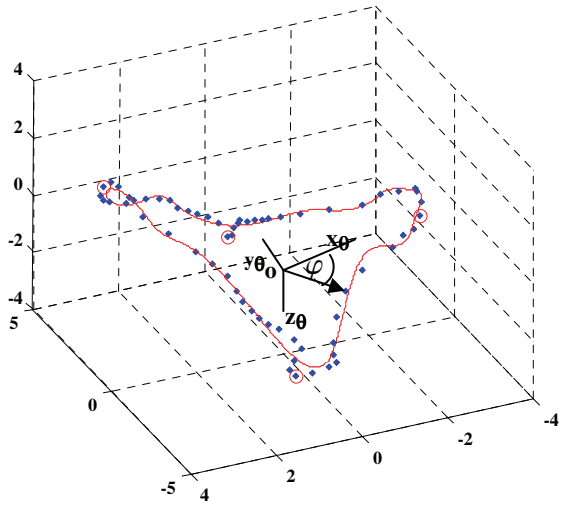

Figure 4. Embedded action space of a humanoid walking gait. Training data points in the reduced posture space (shown in blue-dots) are converted to a cylindrical coordinate frame relative to the coordinate frame $\mathbf{x}_\theta, \mathbf{y}_\theta, \mathbf{z}_\theta$. The points are then represented as a function of the angle φ, which forms an embedded action space (shown in red-solid-curve). This action space represents a single gait cycle.

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

$$\mathbf{z}_\theta = \frac{\sum_{\mathbf{i}} (\hat{\mathbf{x}}^{\,i} \times \hat{\mathbf{x}}^{\,i+1})}{\left\| \sum_{\mathbf{i}} (\hat{\mathbf{x}}^{\,i} \times \hat{\mathbf{x}}^{\,i+1}) \right\|}. \qquad (1)$$

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 (φ, r, h) where r is the radial distance, h the height above the $\mathbf{x}_\theta$–$\mathbf{y}_\theta$ plane, and φ the angle in the $\mathbf{x}_\theta$–$\mathbf{y}_\theta$ plane. The angle φ can also be interpreted as the phase angle of the motion.

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

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

where $0 \le \varphi \le 2\pi$. Approximation of this function is performed by using a radial basis function (RBF) network.

4 Learning to Predict Sensory Consequences of Actions

A central component of our proposed framework is learning to predict future sensory inputs based on actions and using this learned predictive model for action selection. The goal is to predict the future sensory state of the robot, denoted by s_{t+1}. In general, the state space $\mathbf{S} = \mathbb{Z} \times \mathrm{P}$ is the Cartesian product of the high-dimensional joint space $\mathbb{Z}$ and the space of other percepts P. Other percepts could include, for example, a torso gyroscope, an accelerometer, and foot pressure sensors as well as information from camera images. The goal then is to learn a function $F : \mathbf{S} \times \mathbf{A} \mapsto \mathbf{S}$ that maps the current state and action to the next state. For this paper, we assume that F is deterministic.

Often the perceptual state s_t is not sufficient for predicting future states. In such cases, one may learn a higher order mapping based on a history of perceptual states and actions, as given by an n-th order Markovian function:

$$s_t = F(s_{t-n}, \ldots, s_{t-1}, a_{t-n}, \ldots, a_{t-1}) \qquad (3)$$

We use a radial basis function (RBF) approximator to learn F from sensory-motor experience. In particular, the RBF network approximates F by learning a function $F' : \alpha \mapsto \beta$:

$$\beta = \sum_{k}^{K} \mathbf{w}_k \exp\left(-(\alpha - \mu_k)^T \sum_k^{-1} (\alpha - \mu_k)\right), \qquad (4)$$

where K represents the number of kernels, μ_k and Σ_k^{-1} are the mean and inverse covariance of the k-th kernel respec-

tively. The output weight vector $\mathbf{w}_k$ scales the output of each kernel appropriately, and the input and output are $\alpha = [s_t, s_{t-1}, ..., s_{t-n-1}, a_t, a_{t-1}, ..., a_{t-n-1}]$ and $\beta = s_{t+1}$ respectively. For convenience, one can instead view the RBF as a time delay network [Lang et al., 1990] for which the input simplifies to $\alpha = [s_t, a_t]$. The previous state and action inputs are implicitly remembered by the network using recurrent feedback connections.

In this paper, we use a second-order (n = 2) RBF network with the state vector equal to the three-dimensional gyroscope signal $(\mathbf{s}_t \equiv \boldsymbol{\omega}_t)$. As discussed in the previous section, an action represents the phase angle, radius, and height of the data in latent posture space $(\mathbf{a}_t \equiv \chi \in \mathbb{X})$.

5 Motion Optimization using the Learned Predictive Model

The algorithm we present in this section utilizes optimization and sensory prediction to select optimal actions and control the humanoid robot in a closed-loop feedback scenario. Figure 5 illustrates the optimization process.

One may express the desired sensory states that the robot should attain through an objective function $\Gamma(\mathbf{s})$. Our algorithm then selects actions $\mathbf{a}_t^*, ..., \mathbf{a}_T^*$ such that the predicted future states $\mathbf{s}_t, ..., \mathbf{s}_T$ will be optimal with respect to $\Gamma(\mathbf{s})$:

$$\mathbf{a}_t^* = \arg\min_{\mathbf{a}_t} \Gamma\left(F(\mathbf{s}_t, ..., \mathbf{s}_{t-n}, \mathbf{a}_t, ..., \mathbf{a}_{t-n})\right). \quad (5)$$

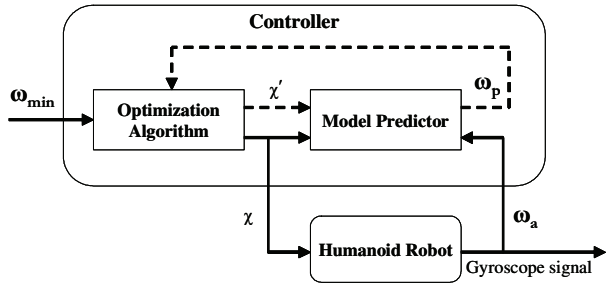

Figure 5. Model predictive controller for optimizing posture stability. The optimization algorithm and the sensory-motor model predictor produce the action $(\mathbf{a}_t \equiv \chi \in \mathbb{X})$ which is used for posture control of the humanoid robot. The resulting gyroscope signal is fed back to the predictor for retraining. The optimization algorithm utilizes a predicted gyroscope signal ω_p in order to optimize actions for posture stability.

The objective function used in this paper is a measure of torso stability as defined by the following function of gyroscope signals:

$$\Gamma(\omega) = \lambda_x \omega_x^2 + \lambda_y \omega_y^2 + \lambda_z \omega_z^2, \quad (6)$$

where $\omega_x, \omega_y, \omega_z$ refer to gyroscope signals in the **x, y, z** axes respectively. The constants $\lambda_x, \lambda_y, \lambda_z$ allow one to weight rotation in each axis differently. The objective function (6) provides a measure of stability of the posture during motion. For our second-order predictive function *F*, the optimization problem becomes one of searching for optimal stable actions given by:

$$\chi_t^* = \arg\min_{\chi_t \in \mathcal{S}} \Gamma\left(F(\omega_t, \omega_{t-1}, \chi_t, \chi_{t-1})\right) \quad (7)$$

$$\mathcal{S} = \begin{bmatrix} \varphi_s \\ r_s \\ h_s \end{bmatrix} \quad (8)$$

To allow for efficient optimization, we restrict the search space to a local region in the action subspace as given by:

$$\varphi_{t-1} < \varphi_s \le \varphi_{t-1} + \varepsilon_\varphi \quad (9)$$

$$r_a - \varepsilon_r \le r_s \le r_a + \varepsilon_r \quad (10)$$

$$h_a - \varepsilon_h \le h_s \le h_a + \varepsilon_h \quad (11)$$

$$0 < \varepsilon_\varphi < 2\pi \quad (12)$$

$$[r_a, h_a] = g(\varphi_s) \quad (13)$$

The phase motion command search range φ_s begins after the position of the phase motion command at the previous time step φ_{t-1}. The radius search range r_s begins from a point in the action subspace embedding $\mathbf{A}$ that is defined by (12) in both positive and negative directions from r_a along $\mathbf{r}$ for the distance $\varepsilon_r > 0$. The search range h_s is defined in the same manner as r_s according to h_a and ε_h. In the experiments, the parameters $\varepsilon_\varphi, \varepsilon_r$ and ε_h were chosen to ensure efficiency while at the same time allowing a reasonable range for searching for stable postures. An example of the search space for a walking motion is shown in Figure 6.

Selected actions will only truly be optimal if the sensory-motor predictor is accurate. We therefore periodically re-train the prediction model based on the new posture commands generated by the optimization algorithm and the sensory feedback obtained from executing these commands. After three iterations of sensory-motor prediction learning, an improved dynamically balanced walking gait is obtained. The trajectory of the optimized walking gait in the low dimensional subspace is shown in Figure 6.

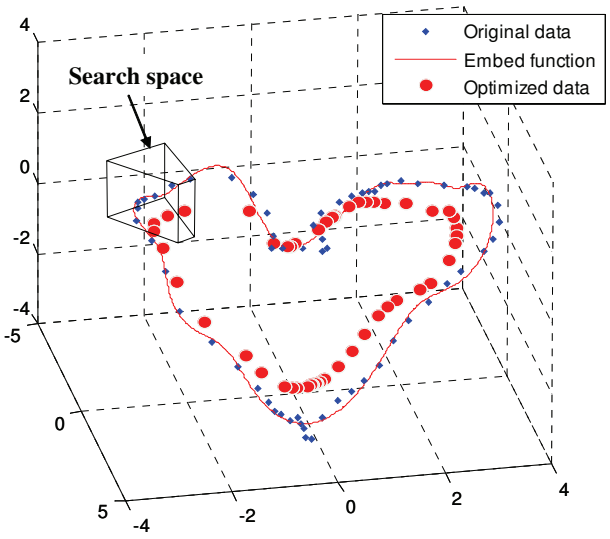

Figure 6. Optimization result for a walking motion pattern in a low-dimensional subspace based on an action subspace embedding.

We summarize below the entire optimization and action selection process:

1) Use PCA to represent in a reduced 3D space the initial walking gait data from human motion capture.
2) Employ the non-linear embedding algorithm for parameterization of the gait.
3) Start the learning process by projecting actions back to the original joint space and executing the corresponding sequence of servo motor commands in the Webots HOAP-2 robot simulator [Webots, 2004].
4) Use the sensory and motor inputs from the previous step to update the sensory-motor predictor as described in Section 4 where the state vector is given by the gyroscope signal of each axis and the action variables are φ, r and h in the low-dimensional subspace.
5) Use the learned model to estimate actions according to the model predictive controller framework described above (Figure 5).
6) Execute computed actions and record sensory (gyroscope) feedback.
7) Repeat steps 4 through 6 until a satisfactory gait is obtained.

6 Experimental Results

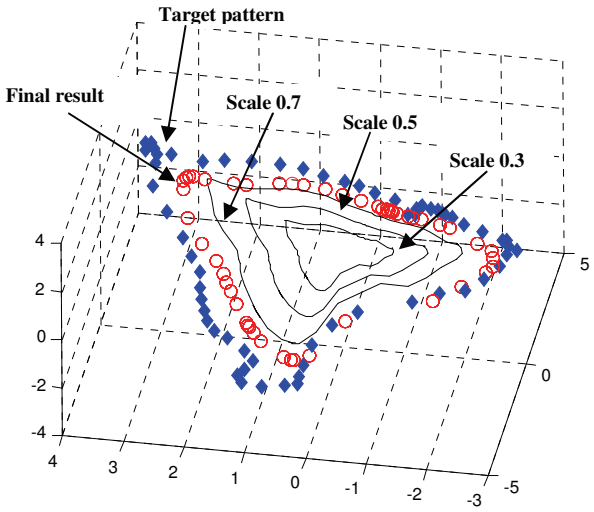

Figure 7. Motion pattern scaling. The target motion pattern is scaled down until it can produce a stable motion to start the motion optimization process.

This section explains how the optimization methodology in the previous section is used in conjunction with the mocap data. From our study of the motion pattern in the reduced subspace, we found that we can scale up and down the motion pattern and get similar humanoid motion patterns except for changes in the magnitude of motion. When we scale down the pattern in the reduced subspace, it produces a smaller movement of the humanoid robot, resulting in smaller changes in dynamics during motion. Our strategy is to scale down the pattern until we find a dynamically stable motion and start learning at that point. We apply the motion optimization method in Section 5 to the scaled-down pattern until its dynamic performance reaches an optimal point; then we scale up the trajectory of the optimization result toward the target motion pattern. In our experiments, we found that a scaling down of 0.3 of the original motion pattern is typically stable enough to start the learning process. Our final optimization result obtained using this procedure is shown as a trajectory of red circles in Figure 7. It corresponds to about 80% of the full scale motion.

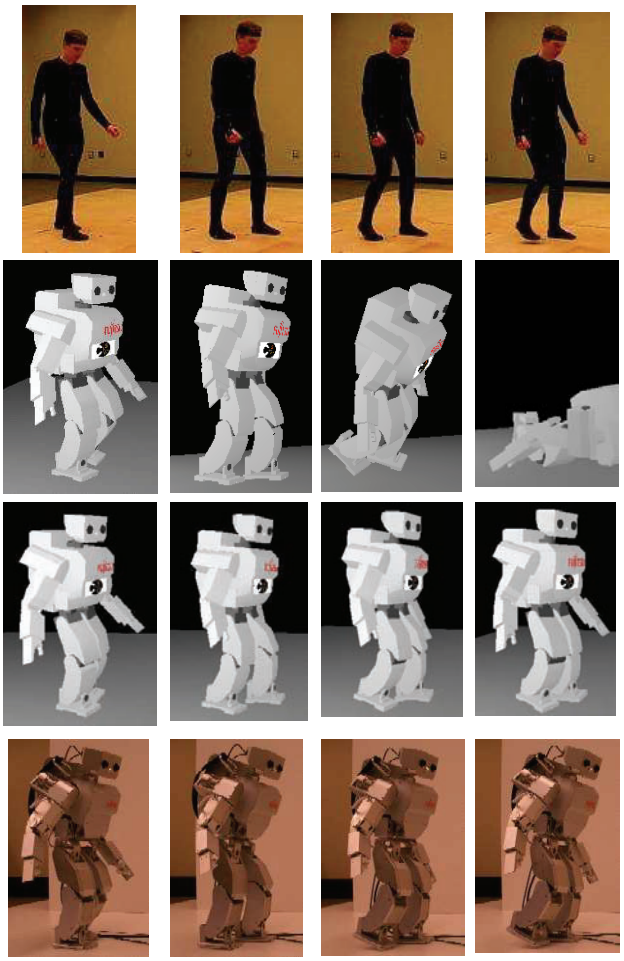

Figure 8. Learning to walk through imitation. The pictures in the first row show a human subject demonstrating a walking gait in a motion capture system. The second row shows simulation results for this motion before optimization. The third row shows simulation results after optimization. The last row shows results obtained on the real robot.

Our simulation and experimental results are shown in Figure 8. We performed the learning process in the simulator [Webots, 2004] and tested the resulting motion on the real robot. The walking gait on the real robot is not as stable as the results in the simulator because of differences in frictional forces between the simulator and the floor. We expect an improvement in performance when learning is performed directly on the real robot. We note that the learned motion is indeed dynamic and not quasi-static motion because there are only two postures in our walking gait that can be considered statically stable, namely, the two postures in the walking cycle when the two feet of the robot contact the ground. The remaining postures in the walking gait are not statically stable as the gait has a fairly fast walking speed.

7 Conclusion

Our results demonstrate that a humanoid robot can learn to walk by combining a learned sensory-motor model with imitation of a human gait. Our approach does away with the need for detailed, hard-to-obtain, and often fragile physics-based models that previous methods have relied on for stable gait generation in humanoids. Our approach builds on several previous approaches to humanoid motion generation and imitation. Okada, Tatani and Nakamura [Okada et al., 2002] first applied non-linear principal components analysis (NLPCA) [Kirby and Miranda, 1996] to human and humanoid robot motion data. The idea of using imitation to train robots has been explored by a number of researchers [Hayes and Demiris, 1994], [Billard, 2001]. In [Ijspeert et al., 2001], a nonlinear dynamical system was carefully designed to produce imitative behaviors. The mimesis theory of [Inamura et al., 2003] is based on action acquisition and action symbol generation but does not address dynamics compensation for real-time biped locomotion. The motion segmentation framework in [Jenkins and Mataric, 2003] uses dimensionality reduction and segmentation of motion data in the reduced dimensional space but without dynamics compensation.

The framework described in this paper has several practical applications. One scenario we are currently investigating is a general navigation task involving our humanoid robot. A modular architecture could be used to switch between a set of learned modules, such as walking-straight, turning-left, turning-right and stepping-backward. Using visual information as feedback, we could control robot direction by switching behaviors (actions) of the robot. We have also found that translation in the x-y plane in the reduced 3D latent space can be used for changing direction. Because our method does not depend on a model of the world, it can be applied to the problems of learning to walk up and down stairs, and walking on constant slopes. To learn actions other than walking straight, we can modify our optimization function in Eq. (6) or introduce additional sensory variables such as velocity or foot-contact pressure. We are currently investigating these lines of research.

Clearly, the present framework cannot be applied directly to the problem of navigation on uneven terrain. To effectively navigate on uneven terrain, we may need a higher degree of compliance control in the leg and foot actuators. However, a hybrid active-passive actuator will likely produce even more complex dynamics than typical actuators used in current humanoid robots. Since our approach does not require a physics-based dynamics model for learning, it lends itself naturally to tackling this problem. We hope to investigate this important research direction in the near future.

References

[Billard, 2001] A. Billard. Imitation: a means to enhance learning of a synthetic proto-language in an autonomous robot. In *Dautenhahn, K. and Nehaniv, C. (eds), Imitation in Animals and Artifacts,* Academic Press, pages 281-311, 2001.

[Grochow et al., 2004] Keith Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. *ACM Transaction on Graphics*, 23(3):522–531, 2004.

[Hayes and Demiris, 1994] Hayes, G. and Demiris, J. A robot controller using learning by imitation. In *International Symposium on Intelligent Robotic Systems,* pages 198 204, Grenoble

[Ijspeert et al., 2001] Auke Jan Ijspeert, Jun Nakanishi and Stefan Schaal. Trajectory formation for imitation with nonlinear dynamical systems. In *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems,* pages 752–757, 2001.

[Inamura et al., 2003] Tetsunari Inamura, Iwaki Toshima and Yoshihiko, Nakamura. Acquiring motion elements for bi-directional computation of motion recognition and generation. In Siciliano, B., Dario, P., Eds*., Experimental Robotics VIII*. Springer, pages 372–381, 2003.

[Jenkins and Mataric, 2003] Odest C. Jenkins and Maja J. Mataric. Automated derivation of behavior vocabularies for autonomous humanoid motion. In *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS),* pages 225–232, 2003.

[Kajita and Tani, 1996] S. Kajita and K. Tani. Adaptive gait control of a biped robot based on real time sensing of the ground profile. In *Proceeding of 1996 IEEE Conference on Robotics and Automation*, pages 570–577, 1996.

[Kirby and Miranda, 1996] Michael J. Kirby and Rick Miranda. Circular nodes in neural networks. *Neural Computation.* 8(2):390–402, 1996.

[Lang et al, 1990] K. J. Lang, A. H. Waibel, and G. E. Hinton. A time-delay neural network architecture for isolated word recognition. *Neural Networks*, 3(1):23–43, 1990.

[MacDorman et al., 2004] Karl F. MacDorman, Rawichote Chalodhorn, and Minoru Asada. Periodic nonlinear principal component neural networks for humanoid motion segmentation, generalization, and generation. In *Proceedings of the Seventeenth International Conference on Pattern Recognition,* pages 537–540, 2004.

[Okada et al., 2002] Masafumi Okada, Koji Tatani and Yoshihiko Nakamura. Polynomial design of the nonlinear dynamics for the brain-like information processing of the whole body motion. In *Proceeding of 2002 IEEE International Conference on Robotics and Automation,* pages 1410–1415, 2002.

[Rao and Meltzoff, 2003] Rajesh P. N. Rao and Andrew N. Meltzoff. Imitation Learning in Infants and Robots: Towards Probabilistic Computational Models. In *Proceeding of the Second International Symposium on Imitation in Animals and Artifacts,* pages 4-14, 2003.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *An Introduction to Reinforcement Learning*. MIT Press, 1998.

[Vukobratovic and Borovac, 2004] Miomir Vukobratovic and Branislav Borovac. Zero moment point-thirty-five years of its life. *International Journal of Humanoid Robotics,* 1(1):157–173, 2004.

[Webots, 2004] Webots, "http://www.cyberbotics.com," commercial Mobile Robot Simulation Software. [Online]. Available: http://www.cyberbotics.com

[Yamaguchi et al., 1996] J. Yamaguchi, N. Kinoshita, A. Takanishi, and I. Kato. Development of a dynamic biped walking system for humanoid: development of a biped walking robot adapting to the humans' living floor. In *Proceeding of 1996 IEEE Conference on Robotics and Automation*, pages 232–239, 1996.