

# Adapting Human Motion for the Control of a Humanoid Robot

Nancy S. Pollard,<sup>1</sup> Jessica K. Hodgins,<sup>2</sup> Marcia J. Riley,<sup>3</sup> Christopher G. Atkeson<sup>4</sup>

## Abstract

Using pre-recorded human motion and trajectory tracking, we can control the motion of a humanoid robot for free-space, upper body gestures. However, the number of degrees of freedom, range of joint motion, and achievable joint velocities of today’s humanoid robots are far more limited than those of the average human subject. In this paper, we explore a set of techniques for limiting human motion of upper body gestures to that achievable by a Sarcos humanoid robot located at ATR. We assess the quality of the results by comparing the motion of the human actor to that of the robot, both visually and quantitatively.

## 1 Introduction

Humanoid robots are already common in theme parks such as Disneyworld and Universal Studios where the investment for a new attraction is substantial. To make humanoid entertainment robots a viable alternative for smaller scale attractions such as location-based entertainment venues (Disney Quest and Dave and Buster’s, for example) and in museums or theme restaurants, we need easier ways of programming these robots. Entertainment robots must have a natural and entertaining style of motion and often require substantial motion databases to ensure a large variety of behaviors.

For a humanoid robot, such as the Sarcos robot at ATR (DB) [1] shown in figure 1, one obvious approach is to drive the motion of the robot with motion capture data recorded from a professional actor. Such data would contain the timing and many of the other subtle elements of the actor’s performance. However the current mechanical limitations of humanoid robots prevent the recorded motion from being directly applied, unless the human actors use only a fraction of their natural joint range and move with slower velocities than those commonly seen in human motion.

We addressed these limitations with straightforward techniques: the location of the markers in the motion capture data

<sup>1</sup>Pollard is at the Computer Science Department, Brown University. Email: nsp@cs.brown.edu

<sup>2</sup>Hodgins is at the School of Computer Science, Carnegie Mellon University. Email: jkh@cs.cmu.edu

<sup>3</sup>Riley is at the College of Computing, Georgia Institute of Technology and ATR Human Information Science Laboratories. Email: m-riley@cc.gatech.edu

<sup>4</sup>Atkeson is at the School of Computer Science, Carnegie Mellon University and ATR Human Information Science Laboratories. Email: cga@cmu.edu



**Figure 1:** The Sarcos humanoid robot at ATR (DB) tracking motion capture data of a human actor.

is first mapped to the degrees of freedom of the robot by inverse kinematics on individual limbs. Then joint angle and velocity limits are imposed on the motion via a local scaling technique. The robot tracks the trajectories of the transformed data using a position and velocity tracking system with one or two iterations of feedforward trajectory learning.

We tested these techniques with fourteen motion sequences from seven professional actors. Each subject performed to the same audio track of the children’s song, “I’m a little teapot.” We chose this selection because it was familiar enough that most actors would perform the motion in a similar but not identical way. It was our hope that an individual’s style would be preserved through the transformations necessary to allow the robot to perform the motion. These transformations result in significant changes to the motion even for the simple actions that accompany this children’s song making it important to choose scaling techniques that preserve the salient aspects of the motion and to assess the results with both qualitative and quantitative measurements of the motion.

## 2 Related Work

Two bodies of work are relevant to this research: robotics researchers have explored algorithms for adapting human motion to humanoid and legged robots and researchers in computer animation have explored algorithms for adapting human motion to animated characters. The problems are similar in that both require human motion to be adapted to a system with different kinematics than that of the actor who was captured. But the difficulties encountered in the two fields are not the same. The range of body types found in animated characters is much greater than that seen in humanoid robots making the adaptation problem more challenging. On the other hand, the motion of animated characters is less restricted than that of current humanoid robots, reducing or even eliminating the problem of handling joint and velocity limits that we faced in

adapting human motion to the Sarcos humanoid robot at ATR.

Riley et al. adapted motion data recorded from an Optotrak motion capture system to the same robot used in this work [2]. If the motion was outside of the robot's joint angle limits, their system translated and scaled the joint trajectory globally. This approach kept the magnitude of the motions as large as possible but at the cost of perhaps using a part of the joint space that the human had not used. In some situations, the global scaling might also reduce the magnitude of small but stylistically important oscillations. Because their sample motion, an Okinawan folk dance, had a musical beat, this scaling law was a good solution and the robot motion resembles that of the human dancer albeit with a slower overall velocity because of the reduced range of motion.

Motion data has been proposed by Dasgupta and Nakamura as a way to modify the walking pattern of legged robots to appear more human-like [3]. Because balance is of primary concern in walking, they used the motion data to specify a trajectory for the zero moment point in the control system rather than using it explicitly as desired joint angles.

Dynamically simulated humans have also been controlled with human motion data. Zordan used data recorded from a magnetic motion capture setup to drive the upper body motion of simulated human figures that gesture, drum and play patty cake [4]. Task constraints for contact are enforced by using inverse kinematics to modify the motion trajectories for the kinematics of a particular animated character.

Playter built a simulation system for a two-dimensional running character in which the controller is constructed with the aid of motion capture data [5]. Simple servo-mechanisms control each joint to approximately follow the desired motion trajectory. The input motion trajectories are scaled in time in an on-line fashion to match the flight and stance time for the character being simulated and to modify the desired joint trajectories in order to maintain balance and reduce errors in forward speed and flight time.

Because of the difficulty of animating complex characters, the animation community has devoted a significant effort to adapting human motion to animated characters. In his motion editing and retargeting work, Gleicher chose to perform a trajectory optimization that did not maintain physical realism in order to allow interactive response for the user [6, 7]. In performance animation or computer puppetry, in contrast, the motion is applied directly to the character without the opportunity for user intervention. Shin et al. implemented an importance-based approach in which the relative importance of joint angles for freespace movements and end effector position and orientation were evaluated based on the proximity of objects and on a priori notations in the performer's script [8]. Their system was used to animate a character in real time for broadcast.

### 3 Capturing Human Motion

We used a commercially available system from Vicon to capture the motion data of the actors [9]. The system has eight cameras, each capable of recording at 120Hz with images of 1000x1000 pixel resolution. We used a marker set with 35 14mm markers that allowed us to measure whole body motion. In particular, the system captured wrist angles and head orientation but not finger or facial motion.

Our actors were professionally trained in expressive movement and had extensive dance and drama education. We used seven actors who are identified in this paper as Actors #1-#7. They were instructed to perform to a pre-recorded audio track which enforced a similar timing for all actors.

The audio track was the children's song, "I'm a little teapot." We selected this piece because it was familiar to all of our actors and contained some sequences that were relatively well prescribed by childhood training ("Here is my handle, here is my spout.") and some sequences that allowed more freedom of expression ("When I get a steam up hear me shout."). We recorded each subject performing the motion twice. For the first trial, subjects were told to stand without moving their feet and to tell the story through gestures. Before the second trial, the subjects were shown a movie of the robot in motion, informally coached on the joint angle limits of the robot, and instructed to try to constrain their motion to be closer to the limits of the robot. We did not see a noticeable difference in their gestures before and after instruction.

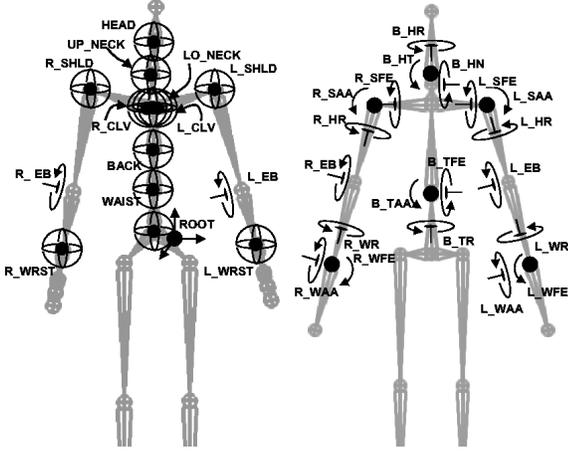
### 4 Techniques for Limiting Human Motion

The captured motion must be processed in several ways before it can be applied to the humanoid robot. First the motion is constrained to match the degrees of freedom of the robot. Then joint angle and velocity limits are applied. A trajectory tracking control system is then used to follow the motion. Subsequent iterations of trajectory learning improve the trajectory tracking performance. Each of these stages is described in detail in the sections below.

#### 4.1 Constraining the motion to the joints of the robot

The processed motion capture data used for our experiments was generated from raw data collected during the motion capture session using Vicon's Bodybuilder software. The raw marker data is mapped onto a skeleton having 39 degrees of freedom in the upper body (figure 2). Link lengths for the skeleton are specific to each human actor, and are established by the software based on a calibration pose taken during the motion capture shoot for that actor.

The motion capture data is mapped to the less flexible skeleton of the robot (figure 2) by setting joint angles to match the orientation of each segment of the robot with the corresponding link or set of links of the human skeleton. Head, upper arm,



**Figure 2:** Degrees of freedom for the upper body of the motion capture skeleton (left) and the robot (right). For the human skeleton, all joints are three degree of freedom ball joints except the ROOT (6DOF), the elbows L\_EB and R\_EB (1DOF each), and the clavicles L\_CLV and R\_CLV (2DOF each).

lower arm, and hand orientations could be matched directly, because both skeletons contain those links. The torso orientation of the robot was set so that the triangle containing the waist and shoulder positions had the same orientation for the two skeletons. Joint limits were disregarded during this part of the process.

**4.1.1 Gimbal Lock:** Joint angles show large variations when the robot is near gimbal lock. Regions near gimbal lock were encountered frequently in these motions because the robot shoulder has a singularity when the arms are at 90 degrees abduction, or swung out to the side of the body to a horizontal position. In this position, the rotational axes of the motors for flexion/extension (R\_SFE or L\_SFE) and for humeral rotation (R\_HR or L\_HR) align with one another and one degree-of-freedom is lost.

To address this problem, we locate regions in the data that are near gimbal lock and compute a restricted degree-of-freedom solution within those regions. The process for a single joint first creates two solution tracks. The desired rotational transform for the joint is extracted from the motion capture data at each frame of the motion sequence. Two solutions are possible for converting any rotational transform to desired joint angles. These two solutions are swapped and adjusted by  $2\pi$  as needed to create two tracks (of three joint angles each) that vary smoothly over time.

Second, desired joint angles are constructed from the two solution tracks. Time periods when the robot is near gimbal lock are marked. For time periods when the joint is not near gimbal lock, the solution closest to the middle of the joint angle range is selected. For time periods when the joint is near gimbal lock:

- An initial guess is generated by linearly interpolating between joint angle values at the start and end of the time period.
- Given this initial guess, a joint angle solution is computed assuming the robot is in a singular configuration. When the robot is in a singular configuration, only the sum or the difference of the two aligned joint angles is known. For ZYX fixed-axis angles, for example, if  $\gamma$ ,  $\beta$ , and  $\alpha$  are the Z, Y, and X axis rotations respectively, the solution is:

$$(\alpha + \gamma) = \tan^{-1} \left( \frac{M_{1,0}}{M_{1,1}} \right), \quad \sin(\beta) > 0 \quad (1)$$

$$(\alpha - \gamma) = \tan^{-1} \left( \frac{M_{2,1}}{M_{2,0}} \right), \quad \sin(\beta) \leq 0 \quad (2)$$

where  $M_{i,j}$  indicates row  $i$ , column  $j$  of rotation matrix  $M$ .

The two angles  $\alpha$  and  $\gamma$  are adjusted equally from their linearly interpolated initial guess to obtain the desired sum in the case where  $\sin(\beta) > 0$ , or difference in the case where  $\sin(\beta) \leq 0$ .

## 4.2 Joint Angle Limits

The range of motion of the joints of the robot is significantly less than that commonly seen in human motion. Therefore, scaling the motion appropriately to bring it within the joint angle limits of the robot is particularly important for preserving the style of the motion. We used a non-uniform, local scaling to modify each trajectory of the motion to lie within the joint limits of the robot while retaining most of the individual oscillations seen in the original motion.

Looking at each joint angle independently, each segment of motion that is greater than the joint angle limit  $\theta_U$  is identified (similarly for segments of motion that are less than the joint angle limit  $\theta_L$ ). The segment has endpoints in time of  $i_0$  and  $i_1$  and a maximum value of  $\theta_{SU}$ . The segment is then expanded in time by a user-controlled margin that is a fraction of the length of the segment. This expansion provides new endpoints  $i'_0 = i_0 - i_{mar}$  and  $i'_1 = i_1 + i_{mar}$  where  $i_{mar} = 0.2(i_1 - i_0)$  for the trials shown in this paper. All of the motion within this expanded segment is then scaled by a linear blend between two scale factors,  $S_0$  and  $S_1$  where

$$S_0 = \left| \frac{\theta_U - \theta_{i'_0}}{\theta_{SU} - \theta_{i'_0}} \right| \quad (3)$$

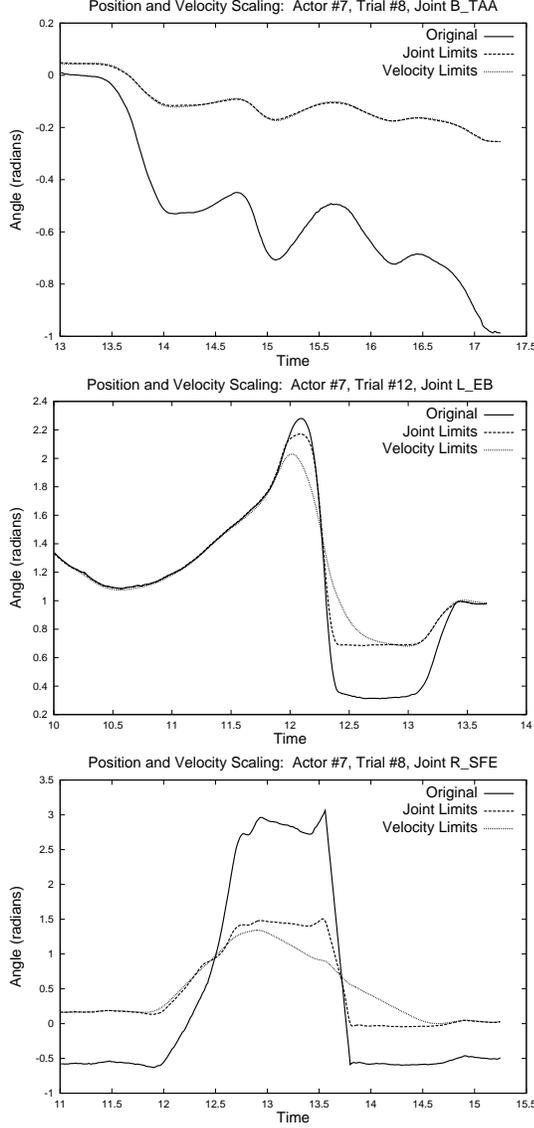
$$S_1 = \left| \frac{\theta_U - \theta_{i'_1}}{\theta_{SU} - \theta_{i'_1}} \right| \quad (4)$$

With  $w$  a weighting function for a linear blend from 0 to 1 over the expanded segment, we have

$$S_i = (1 - w)S_0 + wS_1 \quad (5)$$

$$\Delta\theta_i = (1 - w)\theta_{i'_0} + w\theta_{i'_1} \quad (6)$$

$$\theta'_i = S_i(\theta_i - \Delta\theta_i) + \Delta\theta_i \quad (7)$$



**Figure 3:** Results of position and velocity scaling. All plots show the original mapping to the robot’s degrees of freedom, the curve after local scaling to joint angle limits, and the curve after both joint angle and velocity scaling. (Top) Scaling of the waist abduction/adduction (B\_TAA) axis as the performer bends to the left. The original bobbing motion is still evident in the robot’s motion, although it has been moved to a different part of the workspace. (Middle) Scaling of the left elbow joint (L\_EB). Both local joint scaling and velocity scaling effects are seen as the performer throws her arms out nearly straight. (Bottom) Scaling of the right shoulder flexion / extension joint (R\_SFE). Here the performer flings her arms up into the air, holds them there for a short time and pulls them downward abruptly. The robot cannot raise its elbows much above the shoulder, and velocity limits for the shoulder are somewhat severe, so the scaling techniques convert this motion to a more subdued extension of the arms out to the sides.

Figure 3 shows joint angles that were scaled using this algorithm. The local oscillations that occur outside of the range of motion of the robot are preserved albeit at a reduced scale in the top graph. The bottom graph shows a motion segment that is difficult to scale because the amplitude was drastically reduced by the joint angle limits.

For some joint angle trajectories, local scaling may not give the best answer. For example, percussive movements such as striking a drum might best be scaled by truncation to preserve the velocity of the movement up until the joint angle limit is reached. If maintaining the speed of a movement as closely as possible is more important than the absolute joint angle, then a global scaling algorithm would allow the use of the full range of motion at the cost of moving the joint into a part of the range that the original motion did not use.

### 4.3 Velocity Limits

The joint velocities are computed numerically from the joint position data. Velocity limits for each joint were obtained empirically. Each joint angle curve is scaled by averaging the results of an ideal, simulated, velocity-limited tracking controller run both forward and backward in time. Given the original curve  $\theta$ , the controller is run forward to produce  $\theta_{F,i+1}$  for all time steps  $i$  as follows:

$$\dot{\theta}_i = \theta_i - \theta_{i-1} \quad (8)$$

$$\ddot{\theta}_{F,i+1} = 2\sqrt{k_s}(\dot{\theta}_{F,i} - \dot{\theta}_i) + k_s(\theta_{F,i} - \theta_i) \quad (9)$$

$$\dot{\theta}_{F,i+1} = \max(\dot{\theta}_L, \min(\dot{\theta}_U, \dot{\theta}_{F,i} + \ddot{\theta}_{F,i+1})) \quad (10)$$

$$\theta_{F,i+1} = \theta_{F,i} + \dot{\theta}_{F,i+1} \quad (11)$$

where  $\dot{\theta}_L$  and  $\dot{\theta}_U$  are the lower and upper velocity limits for this joint respectively, and all values for out of bounds indices  $i$  are set to zero.

Run backward, the controller computes  $\theta_B$  as follows:

$$\dot{\theta}_i = \theta_i - \theta_{i+1} \quad (12)$$

$$\ddot{\theta}_{B,i-1} = 2\sqrt{k_s}(\dot{\theta}_{B,i} - \dot{\theta}_i) + k_s(\theta_{B,i} - \theta_i) \quad (13)$$

$$\dot{\theta}_{B,i-1} = \max(\dot{\theta}_L, \min(\dot{\theta}_U, \dot{\theta}_{B,i} + \ddot{\theta}_{B,i-1})) \quad (14)$$

$$\theta_{B,i-1} = \theta_{B,i} + \dot{\theta}_{B,i-1} \quad (15)$$

The final, velocity-limited curve  $\theta_V$  is the average of the forward and backward passes:

$$\theta_{V,i} = 0.5(\theta_{F,i} + \theta_{B,i}) \quad (16)$$

The controller is set to be critically damped, and the time constant for response to errors is determined by a single stiffness parameter  $k_s$ . Averaging the forward and backward simulations produces a curve that has been modified symmetrically in time (figure 3).

### 4.4 Trajectory Tracking and Learning

The joint angle and velocity limited trajectory is used as input to a trajectory tracking control system that computes joint torques for the robot using a PD servo:

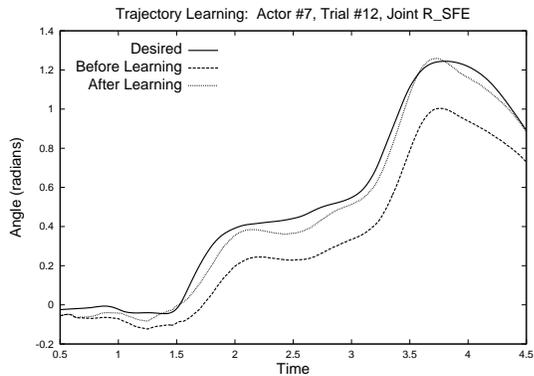
$$\tau = k_p(\theta_d - \theta) + k_v(\dot{\theta}_d - \dot{\theta}) \quad (17)$$



**Figure 5:** Two trials for each of seven actors. The images show the actor in the pose that accompanied the end of “Here’s my handle, here’s my spout” as well as the corresponding pose for the robot.

where  $k_p$  and  $k_v$  are position and velocity gains respectively,  $\theta_d$  is the desired trajectory calculated from the human motion data,  $\dot{\theta}_d$  is the velocity of that trajectory (computed numerically) and  $\theta$  and  $\dot{\theta}$  are the joint angle and joint velocity of the robot. The gains were hand-tuned to be stiff without producing vibration for the motion trajectories in our test set.

The joint angles produced by the trajectory tracking servo necessarily lag behind the desired joint angles. We reduced this lag and provided gravity compensation with a feedforward term. The feedforward term was computed by running a particular trajectory and recording the torque from that trial for use as the feedforward term on the next trial. This approach to trajectory learning was originally proposed by Kawato et al. [10, 11]. Figure 4 shows an example trajectory before and after trajectory learning. The results presented in this paper are after one or two iterations of feedforward learning.



**Figure 4:** Several seconds of an example trajectory before and after trajectory learning.

## 5 Assessment of the Motion Quality

Figure 6 shows the motion of one of the trials for actor #7 sampled every 0.33sec. In many of the frames the robot motion is a good match for the actors. However, the end of the fourth row of paired images shows the effect of the limited shoulder motion both in how high the robot can raise its arms (L\_SAA, R\_SAA) and in humeral rotation (L\_HR, R\_HR). The motion from this trial and others can be viewed at <http://www.cs.cmu.edu/~jkh/style/>.

	1a	1b	2a	2b	3a	3b	4a	4b	5a	5b	6a	6b	7a	7b
1a	<b>10</b>	<b>20</b>	65	74	67	65	61	58	53	53	57	61	59	57
1b	<b>16</b>	<b>9</b>	62	70	67	65	53	51	44	46	54	52	57	51
2a	55	53	<b>9</b>	<b>34</b>	65	58	34	34	68	62	46	53	52	55
2b	61	59	<b>32</b>	<b>7</b>	60	58	38	37	79	70	43	45	47	53
3a	59	62	63	54	<b>15</b>	<b>57</b>	60	56	71	69	64	67	67	64
3b	55	56	55	56	<b>57</b>	<b>15</b>	55	52	77	70	68	67	63	63
4a	53	48	36	40	68	61	<b>8</b>	<b>10</b>	56	52	44	43	50	41
4b	51	46	37	40	64	58	<b>12</b>	<b>8</b>	55	54	42	41	46	39
5a	48	43	69	82	74	87	55	52	<b>11</b>	<b>37</b>	65	64	69	51
5b	45	42	64	71	72	79	53	54	<b>31</b>	<b>14</b>	65	68	68	57
6a	55	56	55	52	71	75	53	50	67	71	<b>7</b>	<b>30</b>	43	36
6b	50	47	58	47	66	70	50	45	54	64	<b>22</b>	<b>12</b>	<b>46</b>	38
7a	56	57	56	59	76	72	58	52	72	72	43	52	<b>12</b>	<b>29</b>
7b	56	54	67	69	75	75	50	46	57	66	38	48	<b>32</b>	<b>9</b>

**Figure 7:** Difference between the recorded human motion mapped to the robot’s degrees of freedom but not joint angle or velocity limited and the actual motion executed by the robot. Each number is the average of the sum squared error over the eight joint angles (in tenths of  $\text{rad}^2$ ). The table shows the human motion for each of the fourteen trials compared to the robot motion for each trial. If the robot trajectory is a good match for the human motion, we would expect smaller numbers on the diagonal (bold). If the other trial from that actor (also bold) has a low number then the two trials were similar.

Figure 5 shows two trials for each of seven actors. The images show the actor and the robot in the pose that accompanied “Here is my handle, here is my spout.” The differences between each of the two performances for an actor are in general much smaller than the differences between the actors. Similarly, the robot motion for one actor tends to be more similar than the motion generated by the trajectory recorded from different actors.

Figure 7 shows the difference between the recorded human motion mapped to the robot’s degrees of freedom but not joint angle or velocity limited and the actual motion executed by the robot. Small numbers along the diagonal indicate that the robot motion was a good match for the corresponding human motion.

## 6 Discussion

This paper describes our approach to scaling human motion capture data to the capabilities of a humanoid robot. Joint and velocity limits were incorporated into the motion using local scaling algorithms, and the motion was processed to avoid artifacts due to the singularity within the workspace of the robot’s shoulder.

Probably the greatest limitation of this approach is that we are scaling the degrees of freedom of each joint independently. In motion segments where one degree of freedom of a joint exceeds its limits but another does not, this produces motion that

does not match that of the actor. For example, when the actor bends both sideways and forward, the resulting robot motion is primarily forward because of the joint limits at the waist joint. The motion might more closely match that of the actor if the motion of both degrees of freedom were scaled to keep their relative movement the same. The principle of scaling all degrees of freedom of a joint together could be extended to scaling the motion of a limb as one element. By using inverse kinematics, we could ensure that the location of the robot’s hand matched that of the actor’s as closely as possible. While this criterion is not important for all tasks it would allow contact between body parts and with objects in the environment.

Earlier work with this robot has included a dynamics model for use in feedforward control [12]. We considered that approach but chose to use trajectory learning instead because it was easy to apply and handled task dependent actuator dynamics and friction. Inverse dynamics requires data for the desired acceleration which has to be computed by differentiating the motion capture position data and filtering heavily.

Although the resulting robot motion is often recognizable as having many of the salient characteristics of the actor’s motion, there are also many details that are missing. We asked the actors not to move their feet, but some of the captured sequences contain significant hip, knee, and ankle motion. This motion was discarded and only the motion from the pelvis, torso, arms and head was processed. We restricted the motion in this way because the robot was not designed to support its own weight and tracking the lower body motion while holding the pelvis stationary and the feet off the ground would result in motion that did not at all resemble that of the actor standing on the ground. If whole body motion was required, for example for adapting motion for a humanoid robot capable of walking then motion stability would have to be addressed in addition to the joint and velocity limits as discussed here.

Our motion capture was limited as well. For example, direction of gaze is often important for communication, particularly in conversational turn taking. Our motion capture system records the position and orientation of the head but cannot record the gaze direction of the eyes relative to the head. The robot has actuators to control the gaze direction of cameras located as eyes in the head which were not used in these experiments. We could synthesize the missing information with heuristics about how people direct their gaze relative to their head or use an eye tracker to record eye gaze along with the motion data. Although the importance of eye gaze as a stylistic element is not known, it has been demonstrated to be important both in the performance of a conversational robot [13] and in an avatar used for a negotiation task [14].

We recorded several other motions in addition to “I’m a Little Teapot” but, as performed by our actors, each of these motions involved significant contact or near contact between the actor’s body segments. When these motions were scaled to the degrees of freedom and limb lengths of the robot, there was interpenetration of body segments and the trajectories could not

be run safely on the robot. Small interpenetrations could be resolved by detecting collisions and using the Jacobian of the robot to perturb the trajectory and avoid the collision. Larger interpenetrations might require a planning algorithm to determine how the trajectory should be altered. Because the original motion was produced by actors whose kinematic structure is similar to that of the robot, we hypothesize that the simple approach might be sufficient.

In the longer term, we would like to more rigorously answer the question of whether the style of an individual actor is retained despite the transformations performed on the data to fit it to the robot. We plan to show paired videos of motions (one robot, one human) to subjects and ask them whether the two videos are based on the same motion sequence. If the results of this experiment show that the subjects are able to make this distinction then we will have demonstrated that the style of an particular performance was retained. A second, more difficult, test would assess whether an individual's style is recognizable. In this experiment, subjects would view several instances of a particular actor performing and then decide if a given robot motion had been performed by that actor. If subjects can make this judgment successfully, then the actor's style has been retained in the motion.

## References

- [1] DB, "<http://www.erato.atr.co.jp/db/home.html>."
- [2] M. Riley, A. Ude, and C. G. Atkeson, "Methods for motion generation and interaction with a humanoid robot: Case studies of dancing and catching," *AAAI and CMU Workshop on Interactive Robotics and Entertainment 2000*, April 2000.
- [3] Anirvan Dasgupta and Yoshihiko Nakamura, "Making feasible walking motion of humanoid robots from human motion capture data," *Proceedings fo the 1999 IEEE International Conference on Robotics and Automation*, pp. 1044 – 1049, May 1999.
- [4] V. B. Zordan and J. K. Hodgins, "Tracking and modifying upper-body human motion data with dynamic simulation," in *Computer Animation and Simulation '99, Eurographics Animation Workshop*, 1999.
- [5] R. Playter, "Physics-based simulations of running using motion capture," *SIGGRAPH 2000 Course Notes, Course #33*, 2000.
- [6] Michael Gleicher, "Motion editing with spacetime constraints," *1997 Symposium on Interactive 3D Graphics*, pp. 139–148, April 1997.
- [7] Michael Gleicher, "Retargeting motion to new characters," *Proceedings of SIGGRAPH 98*), pp. 33–42, July 1998.
- [8] Hyun Joon Shin, Jehhee Lee, Michael Gleicher, , and Sung Yong Shin, "Computer puppetry: An importance-based approach," *ACM Transactions on Graphics*, April 2001.
- [9] Vicon Motion Systems, "<http://www.vicon.com/>," .
- [10] M. Kawato, K. Furukawa, and R. Suzuki, "A hierarchical neural-network model for control and learning of voluntary movement," *Biological Cybernetics*, vol. 57, pp. 169–185, 1987.
- [11] Mitsuo Kawato, "Feedback-error-learning neural network for supervised motor learning," in *Advanced Neural Computers*, R. Eckmiller, Ed., pp. 365–472. Elsevier Science Publishers, 1990.
- [12] S. Schaal, C. G. Atkeson, and S. Vijayakumar, "Scalable techniques from nonparameteric statistics for real-time robot learning," *Applied Intelligence*, vol. 16, no. 1, January 2002.
- [13] Hideaki Kikuchi, Masao Yokoyama, Keiichiro Hoashi, Yasuaki Hidaki, Tetsunori Kobayashi, and Katsuhiko Shirai, "Controlling gaze of humanoid in communication with human," *Proceedings of the 1998 IEEE/RSI International Conference on Intelligen Robots and Systems*, October 1998.
- [14] Maia Garau, Mel Slater, Simon Bee, and Martina Angela Sasse, "The impact of eye gaze on communication using humanoid avatars," *Proceedings of the SIGCHI conference on Human factors in Computing Systems*, pp. 309 – 316, 2001.

## Acknowledgments

The authors would like to thank Don Marinelli for recording the audio track for these experiments and Jehhee Lee, Justin Macey, and Rory Macey for their assistance in recording and processing the data. The second author was supported in part by NSF IIS Award #0196089 and this research was supported in part by the CRL Keihanna Human Info-Communication Research Center.

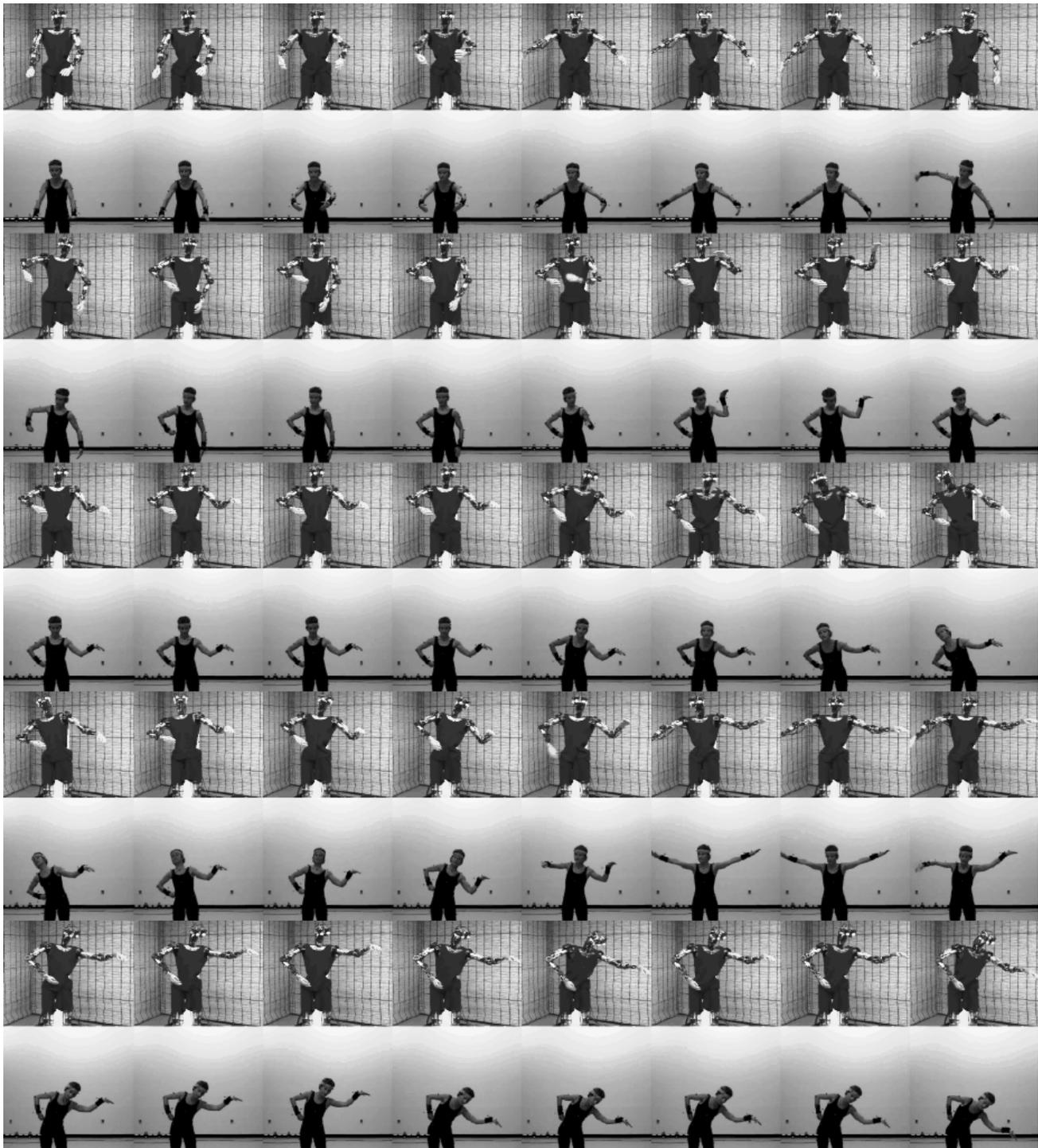


Figure 6: Actor #7, Trial #12 sampled every ten frames (0.33sec).