Coaching: An Approach to Efficiently and Intuitively Create Humanoid Robot Behaviors

Marcia Riley*, Aleš Ude†‡, Christopher Atkeson†§, and Gordon Cheng†¶
*College of Computing, Georgia Institute of Technology
Atlanta, Georgia 30332–0250, USA, Email: mriley@cc.gatech.edu
†ATR Computational Neuroscience Laboratories, Dept. of Humanoid Robotics and Computational Neuroscience
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan
‡Jožef Stefan Institute, Dept. of Automatics, Biocybernetics, and Robotics
Jamova 39, 1000 Ljubljana, Slovenia, Email: ales.ude@ijs.si
§Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA, Email:cga@cmu.edu
¶Japan Science and Technology Agency, ICORP Computational Brain Project
4-1-8 Honcho, Kawaguchi, Saitama, Japan, Email: gordon@atr.jp

***Abstract*— The advances in humanoid robots in recent years have given researchers new opportunities to study and create algorithms for generating humanoid behaviors. Not surprisingly, most approaches for creating or modifying behaviors for complex humanoids require specialized knowledge and a large amount of work. Our aim is to provide an alternative, intuitive way to program humanoid behavior. To do this, we examine human-to-human skill transfer, specifically coaching, and adapt it to the humanoid setting. We enable a real-time scenario where a person, acting as a coach, interactively directs humanoid behavior to a desired outcome. This tightly coupled interaction between a person and a humanoid allows efficient, directed learning of new behaviors, where behavior characteristics can be modified on-demand. Communication is realized through demonstration and a coaching vocabulary, and changes are effected by transformation functions acting in the behavior domain.**

I. Introduction and Related Work

In film and literature we often see people interacting with robots just as they do with other people: for example, they use natural communication such as speech and gesture to direct robots. In this fictional world, even people who are not robot experts can control complex machines including humanoids with ease. However, in today's reality creating behaviors for humanoid robots remains a task for specialists, where communication of behavior details is often time-consuming and takes place largely through the mechanisms of programming.

One way we can begin to address this disconnect between imagined possibilities and current reality is by focusing on paradigms which afford more intuitive methods for creating robot behaviors. Human-to-human skill transfer is an especially interesting model for building robot behaviors, as, besides efficiency, it offers a familiar context to people dealing with humanoids: rather than learning special skills, people can bring their own knowledge from interacting with each other directly into the humanoid domain. In this work we develop an approach to generating robot behavior modeled on a particular type of skill transfer: coaching, where a robot acquires new skills with guidance from a human coach. For this work, we explore the specific behavioral domain of movement. Where possible, we emulate the efficiency of human skill transfer, and because of the familiar, high-level control afforded by coaching we enable non-specialists to participate in creating robot behaviors.

Other robotics researchers inspired by coaching include Nakatani and co-authors [1], who use coaching to aid in balance and walking controller design for a biped robot. Their experiments nicely demonstrate the efficiency gains of introducing intuitive human instruction into the controller design loop, and although their solution is directed toward specialists, the authors encourage creation of adaptable interfaces to allow non-specialists a role in such control. Our approach is more general purpose, targeting trajectory-based movement acquisition and subsequent refinement, and provides mechanisms for novel behavior acquisition and an interface with affordances suitable to specialists and non-specialists alike.

In [2] robot coaching is used in a teaching scheme for a mobile robot where the emphasis is on learning representations for high level tasks rather than on motor skill acquisition. The coaching component, like our system, uses both demonstration and verbal input to direct a robot, although demonstration in [2] is limited to recognizing known primitives, and new behaviors are limited to combinations of these primitives.

In interactive evolutionary computation (IEC), human evaluation is used in optimizations as fitness functions [3], and although especially suited to topics like music retrieval where subjective evaluation is critical, IEC has proven useful in a number of fields including robotics [4]. It differs from coaching, however, in that evaluations usually take the form of selecting preferences from a range of current possibilities, while in coaching specific feedback about how to improve a performance is given.

Motion editors have also been used to create new robot [5] and virtual human behaviors [6], [7], [8]. In [5] Kuroki and colleagues present a motion editor specifically designed for a small biped robot using the graphical tools common in motion editors such as inverse and forward kinematics modes, pose control, pose interpolation, and blending functions. Our

1-4244-0200-X/06/$20.00 © 2006 IEEE

approach differs from this and from most motion editors from the graphics community in the way the user interacts with the robot: our human-robot communication takes the form of a coach's demonstrations and high-level qualitative instructions, while motion editors offer powerful but less intuitive motion editing paradigms requiring more training to master. In addition, our system keeps live robot performance in the loop, allowing for timely evaluation by the coach.

In the next sections we discuss the role of a coach in motor skill acquisition, followed by our adaptation and implementation of useful coaching formalisms comprising our humanoid coaching system, including domain-specific vocabulary, transformation functions, modes of demonstration, and mechanisms for focusing student attention in both time and body space. We then discuss our experiments coaching a robot in catching, and in throwing a ball into a basket. All exchanges occur in a real-time interactive setup that preserves the iterative nature of coaching and the tight coupling among effort, evaluation and guidance.

II. The Role of a Coach in Human Skill Transfer

In building our humanoid coaching system, we first studied human coaching, with particular emphasis on the role of the coach in teaching motor skills. In general, a coach is an expert whose job is to improve the performance of a student. This means providing instructions which are incorporated into the student's learning sessions to produce a successful outcome. Coaching, being a well-established field, offers us a number of formalisms for teaching new skills. These include acquiring new motor knowledge; focusing attention on relevant task features to improve learning of critical task aspects; assigning priorities among goals; giving specific feedback to improve the performance; giving a strategy for correction; and helping to iteratively define the characteristics of a successful outcome. These coaching methods imply a tightly coupled interaction between coach and student where close observation of student performance is followed by feedback or further instructions from the coach.

The role and usefulness of an expert to guide a student has been well-studied in humans. Performance and learning varies with the form of the supplied information, its amount and its timing. Frequent ways instructors give information are by showing videotapes of a person performing the task, directly demonstrating the task, physically guiding a person through a task, and providing verbal instructions. With the right guidance at the right time the student can adjust behavior both during and after a learning session until the desired motion or state is attained.

Students use live or video demonstration to observe strategies, spatial or temporal information, and as a reference of correctness for their own attempts at the behavior [9]. Some researchers have shown that mistakes may be more instrumental in facilitating learning than perfect performances, which by themselves are not giving the type of information the learner needs.

Several studies, however, found that showing videotapes alone, which is similar to direct demonstration, often did not improve motor learning [9]. It was postulated that too much information is available, particularly for complex tasks, and the viewer does not know which details are important to the outcome. In one study, showing a videotape by itself was even shown to hinder learning. On the other hand, as early as 1952, verbal instructions were shown to have a lasting effect on learning and performance, although verbal instructions are more useful when used in conjunction with other input, particularly demonstration [9].

Verbal instructions can communicate information including focus, specific stance, or strategies for error correction. Some verbal information takes the form of specific kinematic feedback, such as "bend your knees". Besides patterns of coordination, kinematic feedback can also be position, velocity, and acceleration information. Expert instructors play a valuable role in being able to observe, identify and correct kinematic errors by giving verbal descriptions to the student. The usefulness of kinematic information is supported by studies giving evidence of kinematic trajectory plans in the parietal cortex [10]; the presence of inverse dynamics models in the cerebellum [11]; and motor equivalence where different limbs are shown to produce kinematically similar patterns, despite having such different dynamical properties [12], [13].

In the next sections we discuss implementation of coaching components pulled from these ideas and tied together by an interface used in directing humanoid behaviors.

III. The Humanoid Robot Coaching System

A. Overview

In our humanoid coaching system the coach, much like a dance instructor or sports coach, wishes to change a given behavior to suit a particular end. In order to achieve this, the coach and humanoid must be able to communicate. The interface shown here facilitates and coordinates this communication. Embedded in it are access points for the different capabilities of the system which incorporate:

- vocabulary;
- a set of transformation functions;
- the ability to demonstrate a desired behavior, either through performance or by physically guiding the robot;
- the ability to focus on specific parts of a behavior for refinement (body and time segmentation);
- the ability to clarify instructions or resolve ambiguities through a student-coach dialogue.

Each capability is derived from an aspect of human coaching. The vocabulary, for instance, reflects verbal instructions coaches commonly use to give instructions. These commands center around kinematic descriptions of motion, such as *higher* and *bend*, used often when teaching motor skills. Movements are changed by transformation functions (TFs) articulated by this high-level vocabulary which manipulate appropriate behavioral parameters to achieve a specific outcome (see Section III-B for details). New movement acquisition is based

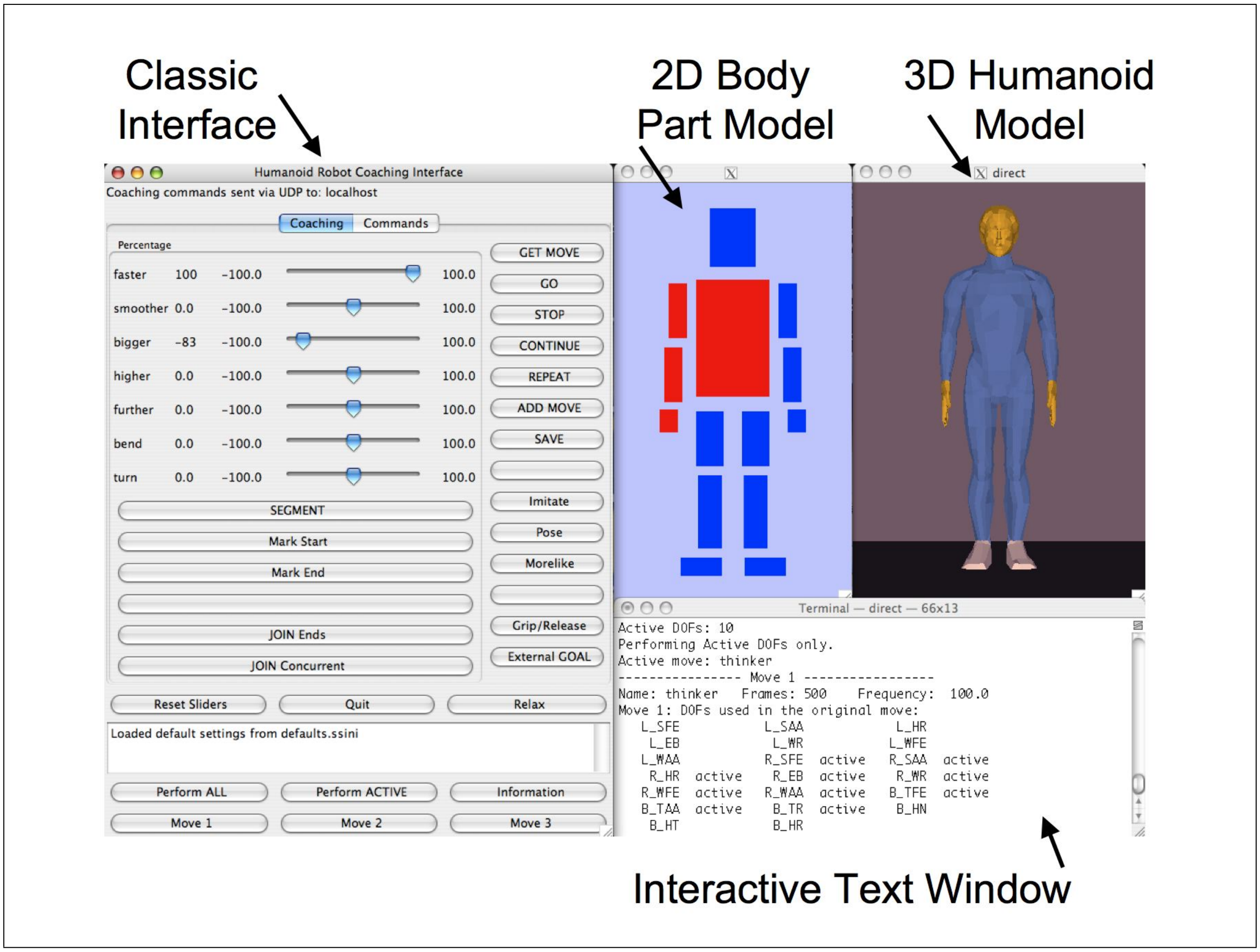

Fig. 1. The four modules of the humanoid robot coaching interface.

on two widely-used methods: demonstration and guiding. Focusing on behavioral features relevant to success as defined by the coach is achieved by selecting specific parts of a movement, such as arm or leg motion (segmenting in body space) for coaching. Attention can also be focused on certain sections of a movement (segmenting in time) by breaking it into sub-movements. Composition of partial movements into a complex movement is easily accomplished by joining segmented sub-movements. Lastly, during human coaching, students are free to ask for clarification when misunderstandings arise. We emulate this by giving the robot the ability to initiate a dialogue with the coach to ask for further instructions when faced with ambiguous or unclear situations.

The interface itself is shown in Fig. 1 and is comprised of modules representing the different functionalities. They are:

- A classic interface comprised mainly of buttons and sliders labeled with various coaching commands making up the explicit coaching vocabulary.
- A simple 2D representation of a robot body allowing the coach to easily focus changes on any part(s) of the body.
- A 3D graphics window which allows visualization of movements on a 3D humanoid to allow quick, intuitive segmentation, and real-time 3D visualization of color markers used in vision-based demonstrations.
- An interactive text-based window to facilitate student-initiated dialogue between coach and student, and to provide current state information to the coach on demand.

Information transfer is initiated by using the vocabulary on the classic interface. We use this type of interface for many higher-level (”verbal”) instructions in order to avoid the pitfalls of speech processing, such as the need for speaker-specific training, although the system has also been successfully tested with speech recognition software.

B. Transformation Functions

At the heart of the system lie transformation functions, which form the essential mechanism for bringing about changes in robot behavior. A TF is typically comprised of a label, which is the coaching command that invokes it, and a set of criteria that serves to define the high level command in terms of low level behavioral criteria. Label and criteria are wrapped together in a function that ultimately effects changes to the appropriate behavioral parameters in accordance with the TF's definition.

C. The Role of World and Self Knowledge

To set the criteria for TFs, the system needs access to certain types of knowledge relevant to the behavior domain. For the movement domain the robot needs an understanding of the relationship between its body and the world. In people, body and world knowledge for movement is gained from childhood on, beginning when children explore the space around them with seemingly random gestures. In our system we seek a minimal knowledge representation that affords the robot the same type of understanding.

We designate world and body (self) reference frames with a known correspondence, each comprised of a 3D Cartesian system where the axes correspond to left, up and front. At any time the robot is able to map its own local orientation to the world reference frame. A TF is defined as relative to either the world or body frame. For example, the notion of "front" and "back" embedded in the *further* TF is always relative to the robot body frame, so the current robot body orientation is used no matter where the robot is in the world, while *higher* is always relative to the world frame. Taken together, the TFs begin to define a type of domain-specific dictionary of behavioral knowledge.

Body knowledge in the humanoid coaching system is also represented in the form of kinematic chains whose connectivity is known to the robot. In our system, the interdependencies of the human skeleton are represented as 6 hierarchically dependent kinematic chains. By exploring the relationship of the robot body joints to the appropriate Cartesian reference frame, the robot can determine which joints may be useful in effecting change for a specified direction. For example, the robot may find that a higher arm movement could be accomplished by extending the arm front and up (shoulder flexion/extension) or to the side and up (abduction/adduction), or some combination of the two. Additionally, knowing its body connectivity, a robot may suggest using the torso to effect changes in an arm posture. In determining which changes to make, the robot engages in a dialog with the coach (see the appendix) resulting in the final set of relevant DOFs used to effect the change. During this exchange, the robot can demonstrate the effect of the candidate DOFs to provide immediate feedback to the coach.

DOF exploration starts with the body parts selected by clicking in the 2D window, which graphically represents body part vocabulary (right arm, head, etc.) in a simplified robot shape. Body part(s) are highlighted (in red) when active, and each part corresponds to a set of candidate DOFs that are considered in effecting subsequent changes. This selection process works in conjunction with the *Perform ACTIVE* and *Perform ALL* options on the classic interface which direct the robot to perform changes using only the selected DOFs, or with all DOFs involved in the movement. With this mechanism, the coach has the option of seeing the effect of partial changes on the entire movement while refining specific pieces.

To determine appropriate DOFs, the robot makes use of forward kinematics where each joint change is related to a change in the 3D positions of virtual points attached to the relevant body part. Our robot is comprised of revolute joints modeled with twists [14] as in our previous work [15], [16], [17]. Each candidate joint is moved by respectively increasing and decreasing its value, and the change in 3D point position attached to the body part moved by the joint is then compared to criteria for the TF, where the position of a point after rotation is given by

$$\boldsymbol{P_{t+1}} = \boldsymbol{g(R,d)} \cdot \boldsymbol{exp(\hat{\omega}\theta)} \cdot \boldsymbol{P_t} \tag{1}$$

where $\boldsymbol{P_t}$ and $\boldsymbol{P_{t+1}}$ are the initial and final 3D positions respectively of a point attached to the body part given in the body coordinate system, $\boldsymbol{g(R,d)}$ is the homogeneous matrix representing the body orientation and position in the world coordinate frame, and $\boldsymbol{exp(\hat{\omega}\theta)}$ is the exponential that maps a rotation of angle θ radians about $\boldsymbol{\omega}$, the unit vector in the direction of the joint axis, to the corresponding rotation matrix. (Note that for the special case of pure rotation, the exponential coordinates of rotation, θ and ω, suffice in the place of the twist coordinates, and the exponential mapping can be efficiently calculated by Rodrigues' formula.)

When both rotational directions match the TF criteria, the solution prefers to continue in the current direction of motion, but the final decision is left to the coach.

For world-based criteria like *higher*, it is important to test DOFs with respect to the robot's world position and orientation since changes therein can affect the solution set of DOFs. (Consider making a higher arm movement lying down versus standing, for example.)

D. Initial Behavior Acquisition

Another important use of domain knowledge is found in imitation, where the coach demonstrates movements that can be understood and reproduced by the robot during interactive coaching sessions. It is not surprising that imitation plays a key role in coaching motor skills, as it is a successful and fundamental strategy used for human learning [18], and has inspired much work in the robotics and virtual human communities [19], [20], [21].

To solve direct imitation, the robot already has crucial information: its position and orientation with respect to the world reference frame, and an understanding of its own body configuration.

Our approach, described in [15], [16], [17] relates the coach's kinematics to the robot's kinematics automatically, and acquires the motions in the robot joint space by matching the position of markers in Cartesian world space attached to the coach's body to the motion of corresponding virtual markers attached to the robot body and measured in body space.

In the past we have used a commercial optical motion capture system with active markers and trailing wires to track points on the body, but for coaching we usually use our own less intrusive (wireless) color tracking system, which tracks color blobs attached to clothing (Fig. 2). During coaching, imitation occurs in real time or immediately following a demonstration, and the solution is constrained to the robot joint limits.

In the coaching system, the *Imitate* command is used with the 3D window to allow real-time display of 3D vision markers attached to the coach, and to visualize solution markers as the transition from Cartesian space to joint angles is calculated. This is important in ensuring good tracking information is maintained, a reasonable solution ensues, and problems such as occlusion can be quickly identified and monitored.

Another common method of seeding behaviors is physically guiding a robot through a motion. This is invoked with the *Pose* command and is accomplished by lowering gains on the robot and directly capturing joint angles while the coach physically guides the robot through a motion.

The last demonstration-based command, *Morelike*, is intended to make a movement similar to the movement being shown. This is achieved by performing a weighted average on joint angles for each DOF used in the demonstration and in the current movement to drive them toward the demonstration.

E. Descriptions of Transformation Functions

Due to space constraints we present only brief descriptions of the remaining transformation functions, omitting most of the mathematical details. TFs were implemented using tools from various areas including digital signal processing, spline analysis, approximation theory, and computer vision.

We chose Cartesian and joint angle space to express movement information because they reflect common spaces for describing movements in human coaching, and lend themselves easily to change within this paradigm. Movements, M, are represented either by a sequence of points P_t in time, splines or radial basis functions, and transformation functions act on these representations.

At the top left of the classic interface, we find motion descriptors and associated sliders, which control the magnitude of the desired changes bounded by the robot's capabilities. *faster* changes the frequency of the movement under consideration, where robot velocity capabilities limit desired frequencies if necessary. *smoother* requires less sharp changes in position with respect to time. This is achieved using a moving average filter which smooths a curve in joint space representing the active motion segment (See Fig. 3). The slider value influences the filter window size. *bigger* corresponds to an increase in amplitude of the movement range measured in joint space and is achieved using a global scaling algorithm [22]. *higher* causes an increase along the vertical axis of the world Cartesian system, and is accomplished by moving the maximum (or minimum) of the current trajectory toward the robot's maximum joint position with a blending function. *further* directs the motion either further left or right, or front or back with respect to the robot body. *bend* bends a part of the body (e.g.,elbow, knee or waist) by increasing the appropriate joint angle over the movement segment under consideration. *turn* orients the body (here, the torso and head) right or left relative to body space, or toward an object in its surroundings.

Next we consider the time segmentation commands *SEGMENT*, *JOIN Ends*, and *JOIN Concurrent* that allow the coach to split a movement into sub-movements or join two movements together. The coach can visualize a movement in the 3D humanoid window to quickly select the beginning and end of a segment using the *SEGMENT*, *Mark Start* and *Mark End* buttons. Once a movement segment is identified, instructions from the coach will affect only this segment until segmentation is turned off.

In the case of *JOIN Ends*, the end of one movement is joined to the beginning of the second movement. When the two joined movements have different frequencies, relative frequencies are preserved by re-sampling the slower segment represented by splines at the higher frequency. *JOIN Concurrent* aligns the start of two segments and merges them into one. This action is intended to join movements with different DOFs (legs plus arms, for example), allowing the coach to create complex movements from simpler ones. The buttons *Move 1*, *Move 2* and *Move 3* allow the coach to switch between movements and select movements to be joined.

When movement segments are joined care is taken to smoothly blend the end and start of adjacent segments to avoid sharp discontinuities in the motion. In all cases the robot's joint limits (position and velocity) act as constraints during modifications, and joint velocities and accelerations are computed by finite differencing after position changes.

Also on the interface are the object interaction commands *Grip/Release* and *External Goal*. The first allows the coach to tell the robot when to grip or release objects in its hand, while the second tells the robot that the current behavior is associated with an external object found in its environs.

The remaining commands are meta-commands which control the flow of the overall coaching session (*GET MOVE*, *GO*, *STOP*, etc.); or housekeeping commands such as *Relax*, which resets the robot posture to reasonable values.

IV. Experiments and Results

Our previous work showed the feasibility of using real-time full-body imitation for movement acquisition [15], [16], [17]. Here we discuss our work on coaching the robot to throw and catch a ball where our student is a 30 DOF humanoid robot [23] shown in Fig. 5. The gross movement for throwing was acquired from direct demonstration using computer vision (see Fig. 2). The original trajectory acquired from the vision data, shown in Fig. 3, was too noisy for the robot to properly execute. So our coaching sequence was as follows:

- acquire a set of throwing movements using real-time demonstration;
- select one of the movements and use *SEGMENT* to extract the relevant part of the trajectory for the desired throw;
- smooth the movement several times, each time acting on the previous results with *smoother* (Fig. 3).

With an acceptable throwing movement, we could now focus on coaching the robot to throw the ball toward the basket. To do this we

- increase the velocity and acceleration with *faster*

Fig. 2. The initial throwing behavior was captured and processed in real time using color markers attached to the body and computer vision techniques.

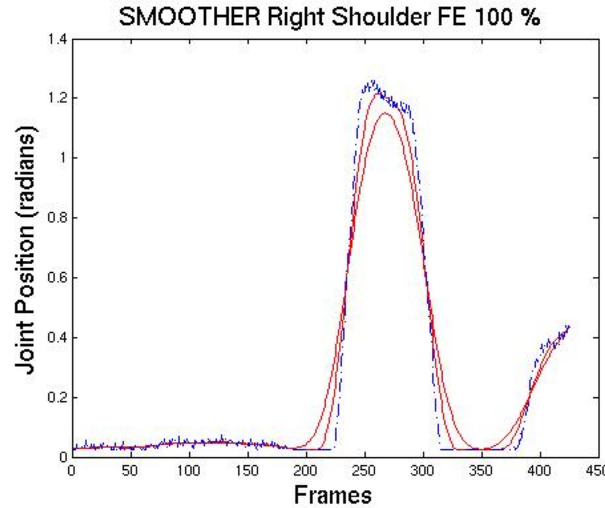

Fig. 3. Original (dashed, noisy line) and modified trajectories for the right shoulder flexion/extension DOF showing modification by two iterations of the *smoother* transformation function implemented with a moving average filter.

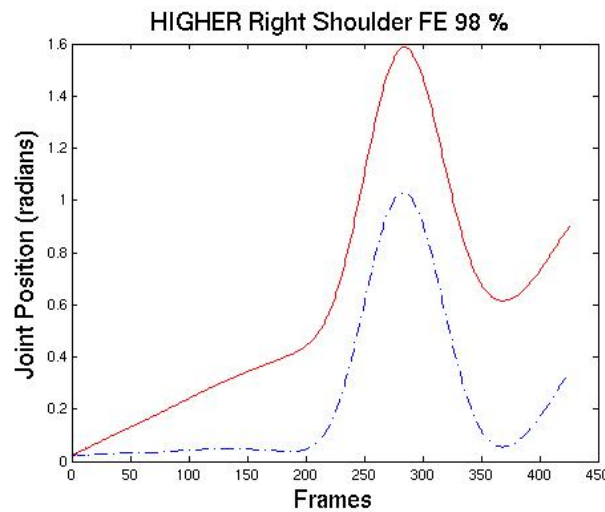

Fig. 4. Original (dashed line) and modified (solid) trajectories showing modification by the *higher* transformation function after using *smoother*.

- change the course of the trajectory with *higher* (Fig. 4) to extend the length of the throw,
- use *release* to specify the exact timing for the release.

During the coaching session, the robot demonstrated how *higher* can be accomplished using a variety of DOFs, and let the coach select the appropriate DOFs (shoulder and elbow flexion/extension) to make the new movement. After each refinement, we (the coaches) watched the robot to evaluate its performance, and then gave successive instructions based on what we saw. Throwing at this point was much improved, but still not satisfactory. This led us to constrain the body space for the movement from DOFs originally used in the movement to the DOFs most relevant for successful robot throwing until throwing was successful.

We then moved the basket, and again coached the robot until it could throw successfully to the new location. In the second coaching sequence, *further* was instrumental in directing the movement toward the robot's right, particularly for the robot torso, as the new target was further to the right. It is important to point out that the acquisition of this behavior was accomplished without any programming and without the input of accurate parameters like velocities and accelerations. The initial trajectories were acquired by observation and then modified using qualitative higher-level instructions. Fig. 5 shows a sequence of postures from a coached throwing movement.

In our catching experiments, we used coaching to improve the performance for an existing catching behavior [24]. In this case we used the transformation function *higher* to change the height where the robot catches the ball. This parameter had an effect on the time it took to catch the ball, with lower catches affording more time to plan and execute an intercept motion. *GO* was used to specify when to begin prediction of the ball's flight. For different types of ball trajectories, different parameters led to successful catching. Our system supports permanently associating the relevant behavior parameters to the movement primitives and thus expanding the knowledge base of the robot.

V. Conclusions and Future Directions

The presented system explores a new way to intuitively create behaviors for complex humanoid robots. Currently, much time is spent by specialists in creating each new behavior. Our intent is to introduce other methods with the potential to improve the time and ease of creating behaviors. Efficiency is often facilitated by intuitive solutions, as they are easy to understand and require less training to use. As we examined strategies people use to acquire new skills, we were inspired by coaching's proven merits in accelerating human skill acquisition. In addition, and perhaps because of

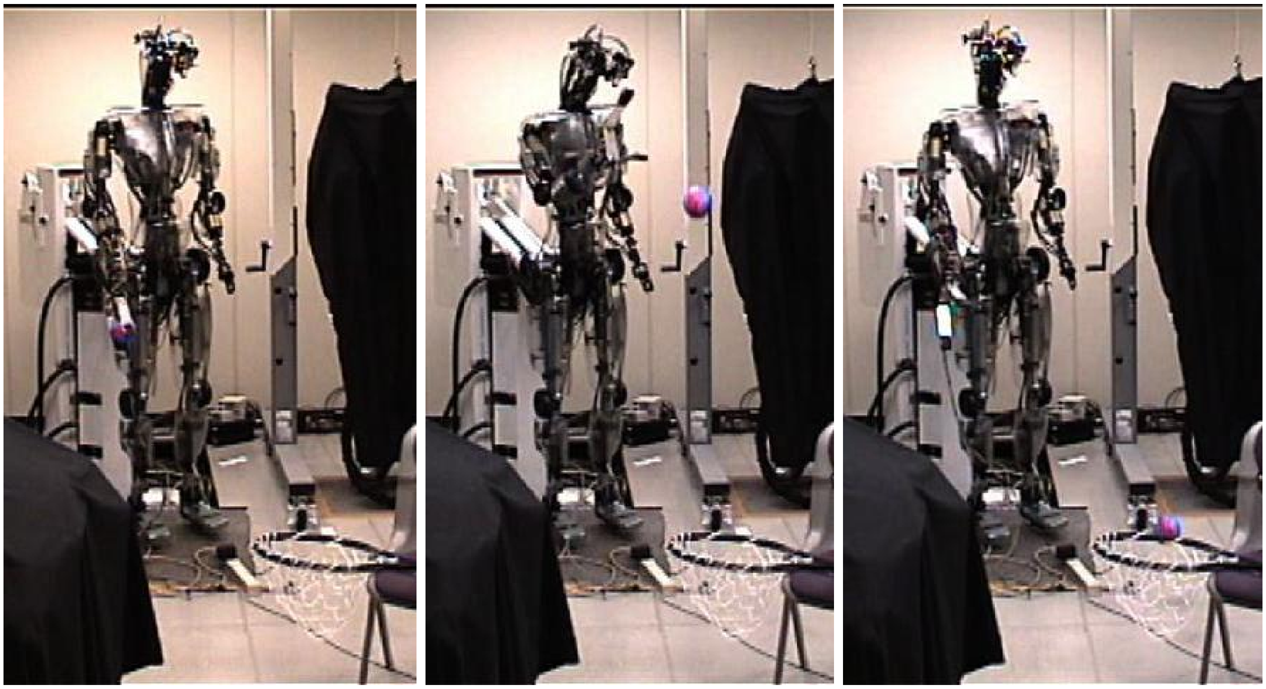

Fig. 5. Postures from a sequence of coached throwing movements.

its success in accelerating learning, coaching is a paradigm familiar on some level to most people. It is a special case of a more general teacher-student relationship that we meet from our infancy forward.

Because of this, our coaching system offers a familiar setting to most people for interacting with and directing the behavior of a complex humanoid robot where human-robot communication takes the form of coach's demonstrations and high-level qualitative instructions. This familiarity allowed us to create a "walk up and use" type of system, where, unlike many motion editing systems, little previous training is needed, and, unlike most current robot control schemes, non-specialists can participate in implementing complex robot behaviors such as throwing a ball in a basket. In doing so we do not obviate the need for specialists to create low-level algorithms for robot control. Instead, we look at the potential role of introducing the advantages of interactive high-level instruction and interactive goal specification used often by people in improving the overall efficiency of creating new robot behaviors. Our approach brings a collaborative nature of problem solving to the domain, where the intent is for widespread availability, ease of use, and the ensuing behavioral flexibility and customization these methods make possible.

Consistent with these goals, we wish to develop new methods for adding transformation functions to the system. The functions described here represent examples of domain-specific transactions related to the language of motion, but are not meant to be an exhaustive list. At present, more transformations can be added as needed by traditional programming methods. However, it would be more suitable and interesting to develop a mechanism for learning new transformations and attaching them to a particular label without the need for such programming. We will work on this in the future.

VI. Appendix

The following exchange shows an excerpt from an interaction between the robot and coach during a *higher* command. The position of a virtual point on the upper arm at its current position and after a positive and negative rotation from the current position is shown. An increase in the second (y) dimension corresponds to an increase along the vertical world axis, the criteria for *higher*. The main points of the robot's communication to the coach are shown in bold. The coach's responses are shown in italics. The robot first checks all active DOFs (those corresponding to body parts selected in the 2D window, here the left upper arm), and then checks any connected parts (here the torso) whether they are active or not to suggest additional possibilities to the coach.

HIGHER requested.
....checking right shoulder

Potential candidates to help with UP for this part:

DOF	Status:
...shoulder flexion/extension	Active (rsfe)
...shoulder abduction/adduction	Active (rsaa)
...shoulder rotation	Active (rshr)

I could also check:

....torso rotation	Not Active (btr)

....torso abduction/adduction Not Active (btaa)
....torso flexion/extension Not Active (btfe)

Cartesian frame changes:
x y z
testing dof shoulder flexion/extension (rsfe)
-10.296700 4.763384 2.019819 (starting position)
-10.296700 11.492188 5.085141 (positive rotation)
-10.296700 3.356163 -1.992737 (negative rotation)

testing dof shoulder adduction/abduction (rsaa)
-11.976195 4.334291 -2.170500
-10.514429 3.448190 -2.170500
-14.950410 12.791493 -2.170500

testing dof shoulder rotation (rhr)
-10.179647 3.349600 -1.507159
-9.144885 3.349600 -4.726161
-8.644364 3.349600 0.327433

Up: Checking displacement for: y
rsfe winner: y displacement: 6.7288
rsaa winner: -y displacement: 8.4572
rhr NO winner: displacement: 0.0000

Can change by using shoulder flex/ext.
Use it?(yes or no)?
Coach: *yes*
Can change by using shoulder abd/add.
Use it?(yes or no)?
Coach: *yes*

Finished with right shoulder. Testing torso next...

Acknowledgment

This material is based upon work supported in part by the DARPA Learning Locomotion Program and the National Science Foundation under NSF Grants CNS-0224419, DGE-0333420, and ECS-0325383. A. Ude was supported by the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) funded by the European Commission.

References

[1] M. Nakatani, K. Suzuki, and S. Hashimoto, "Subjective-evaluation oriented teaching scheme for a biped humanoid robot," in *Proc. IEEE-RAS Conference on Humanoid Robotics*, September/October 2003.

[2] M. N. Nicolescu and M. J. Mataric, "Natural methods for robot task learning: Instructive demonstrations, generalization and practice," in *Proc. Second Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, July 2003.

[3] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no.9, pp. 1275–1296, 2001.

[4] S. Kamohara, H. Takagi, and T. Takeda, "Control rule acquisition for an arm wrestling robot," in *IEEE International Conference on System, Man, Cybernetics*, October 1997, pp. 4227–4231 Vol.5.

[5] Y. Kuroki, B. Blank, T. Mikami, P. Mayeux, A. Miyamoto, R. Playter, K. Nagasaka, M. Raibert, M. Nagano, and J. Yamaguchi, "Motion creating system for a small biped entertainment robot," in *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, October 2003, pp. 1394–1399.

[6] J. Lee and S. Shin, "A hierarchical approach to interactive motion editing for human-like figures," in *Proceedings of ACM SIGGRAPH 99*, 1999, pp. 39–48.

[7] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard, "Interactive control of avatars animated with human motion data," in *Proceedings of ACM SIGGRAPH 2002*, 2002.

[8] M. Gleicher, "Comparative analysis of constraint-based motion editing methods," *Graphical Models*, vol. 63, no. 2, pp. 107–134, 2001.

[9] R. Schmidt and T. Lee, *Motor Control and Learning: A Behavioral Emphasis*. Human Kinetics, 3rd edition, 1999.

[10] J. Kalaska, *What parameters of reaching are encoded by discharges of cortical cells?* John Wiley & Sons, 1991.

[11] N. Schweighofer, J. Spoelstra, M. Arbib, and M. Kawato, "Role of the cerebellum in reaching movements in humans: Ii. a neural model of the intermediate cerebellum," *European Journal of Neuroscience*, vol. 10, pp. 95–105, 1998.

[12] J. S. Kelso, Ed., *Human Motor Behavior: An Introduction*. Lawrence Erlbaum Associates, Inc., 1982.

[13] N. Bernstein, *The control and regulation of movements*. London: Pergamon Press, 1967.

[14] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, New York: CRC Press, 1994.

[15] M. Riley, A. Ude, and C. Atkeson, "Methods for motion generation and interaction with a humanoid robot: Case studies of dancing and catching," in *AAAI and CMU Workshop on Interactive Robotics and Entertainment 2000*, April 2000, pp. 35–42.

[16] M. Riley, A. Ude, K. Wade, and C. Atkeson, "Enabling real-time full-body imitation: A natural way of transferring human movement to humanoids," in *Proc. IEEE Int. Conf. Robotics and Automation*, Taipei, Taiwan, September 2003, pp. 2368–2374.

[17] A. Ude, C. G. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robotics and Autonomous Systems*, vol. 47, pp. 93–108, 2004.

[18] J.Piaget, *Play, Dreams and Imitation in Childhood*. New York: W. W. Norton, 1945, translated 1962.

[19] C. Breazeal and B. Scassellati, "Robots that imitate humans," *TRENDS in Cognitive Sciences*, vol. 6, no. 11, pp. 481–487, 2002.

[20] M. Mataric, "Getting humanoids to move and imitate," *IEEE Journal of Intelligent Systems*, vol. 15, no. 4, pp. 18–24, July/August 2000.

[21] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3, pp. 233–242, 1999.

[22] N. Pollard, J. K. Hodgins, M. Riley, and C. G. Atkeson, "Adapting human motion for the control of a humanoid robot," in *IEEE-RAS Conference on Robotics and Automation*, May 2002, pp. 1390–1397.

[23] C. Atkeson, J. Hale, F. Pollick, M. Riley, S. Kotosaka, S. Schaal, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar, and M. Kawato, "Using humanoid robots to study human behavior," *IEEE Journal of Intelligent Systems*, vol. 15, no. 4, pp. 46–56, July/August 2000.

[24] M. Riley and C. Atkeson, "Robot catching: Towards engaging human-humanoid interaction," *Autonomous Robots*, vol. 12, pp. 119–128, 2002.