Using Humanoid Robots to Study Human Behavior

Christopher G. Atkeson, Joshua G. Hale, Frank Pollick, and Marcia Riley, ATR Human Information Processing Research Laboratories

Shinya Kotosaka, Stefan Schaal, Tomohiro Shibata, Gaurav Tevatia, Ales Ude, and Sethu Vijayakumar, Erato Kawato Dynamic Brain Project

Mitsuo Kawato, ATR Human Information Processing Research Laboratories and Erato Kawato Dynamic Brain Project

Our understanding of human behavior advances as our human robotics work progresses—and vice versa. This team's work focuses on trajectory formation and planning, learning from demonstration, oculomotor control, and interactive behaviors. They are programming robotic behavior based on how we humans "program" behavior in—or train—each other.

We are working on easier ways to program behavior in humanoid robots, and potentially in other machines and computer systems, based on how we "program" behavior in our fellow human beings. We have already demonstrated several simple behaviors, including juggling a single ball by paddling it on a racket, learning a folk dance by observing a human perform it,[1] drumming synchronized to sounds the robot hears (karaoke drumming),[2] juggling three balls, (see Figure 1) performing a T'ai Chi exercise in contact with a human,[3] and various oculomotor behaviors.[4]

Our current robot is DB (which stands for Dynamic Brain), a hydraulic anthropomorphic robot with legs, arms (with palms but no fingers), a jointed torso, and a head (see Figure 2a and www.erato.atr.co.jp/DB). DB was designed by the Sarcos company and the Kawato Dynamic Brain Project (www.erato.atr.co.jp) and was built by Sarcos (www.sarcos.com). The robot is approximately 1.85 meters tall, weighs 80 kg, and contains 25 linear hydraulic actuators and five rotary hydraulic actuators. It has 30 degrees of freedom: three in the neck, two in each eye, seven in each arm, three in each leg, and three in the trunk (see Figure 2b). Every DOF has a position sensor and a load sensor except the eye DOFs, which have no load sensing. The robot is currently mounted at the pelvis, so that we do not have to worry about balance and can focus our studies on upper-body movement. We plan to explore full-body motion in the future, probably with a new robot design.

Inverse kinematics and trajectory formation

One problem that robots with eyes face is visually guided manipulation—for example, choosing appropriate joint angles that let it reach out and touch a visual target. We use learning algorithms (described later in the article) to learn the relationship between where the robot senses its limb is using joint sensors and where the robot sees its limb (referred to in robotics as a model of the *forward kine-*

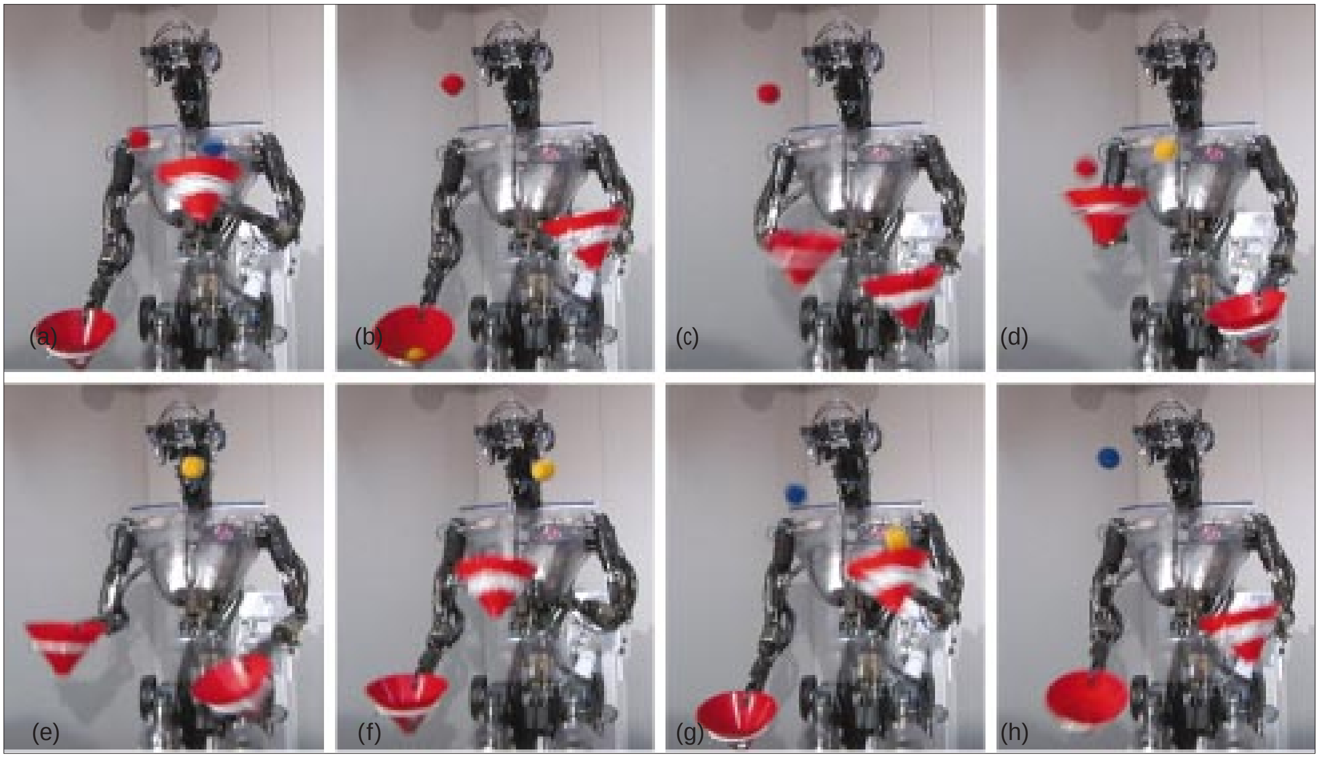

Figure 1. DB can juggle three balls, using kitchen funnels for hands.

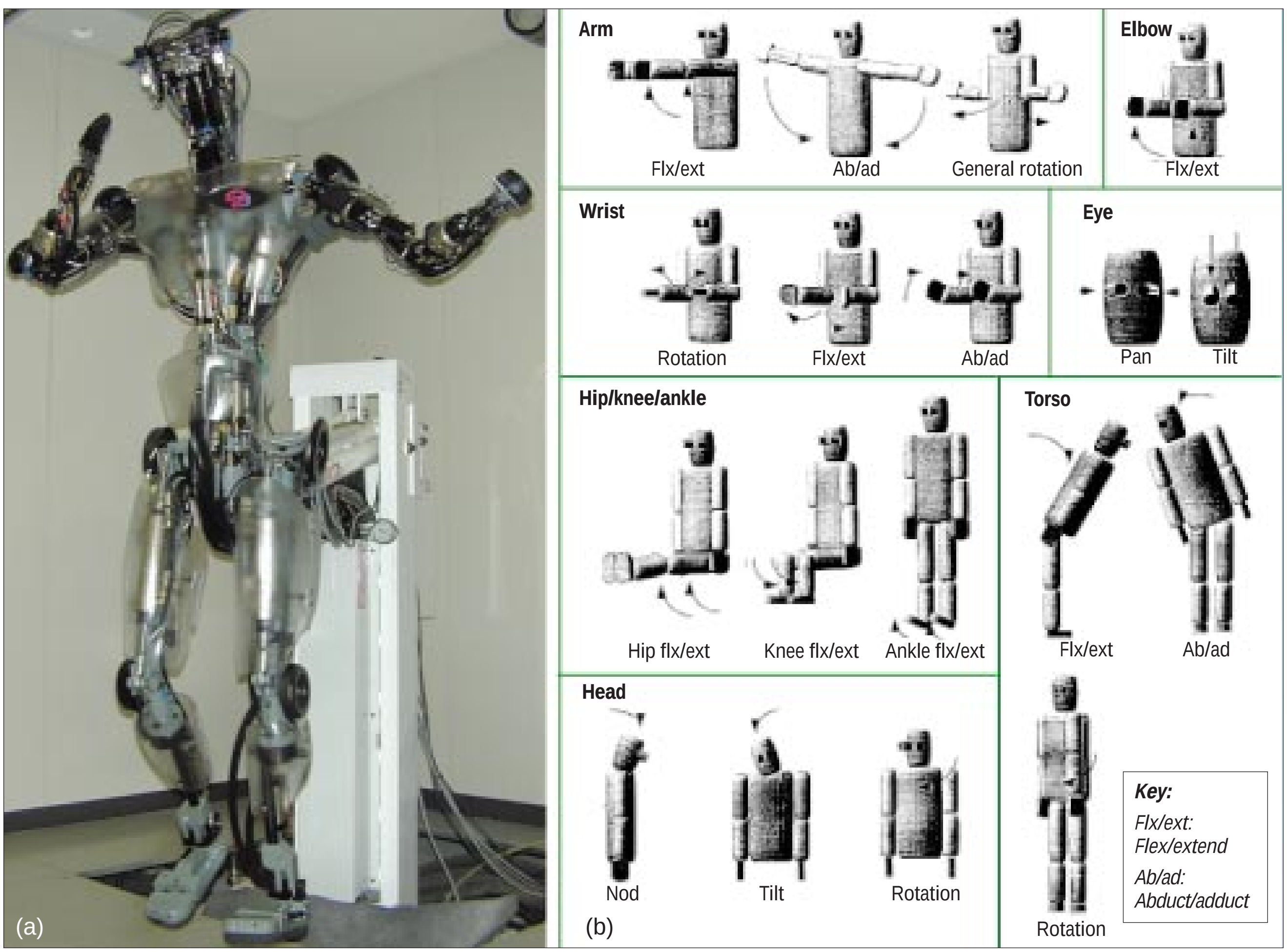

Figure 2. The humanoid robot DB: (a) the full robot mounted at the pelvis; (b) the robot joints.

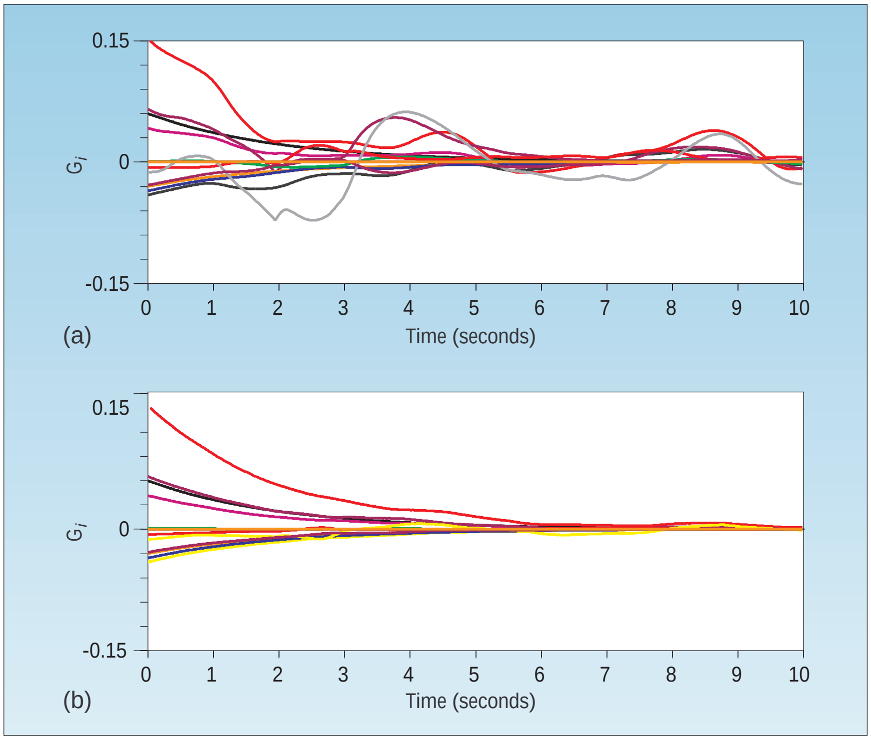

Figure 3. Convergence of inverse kinematics for the humanoid robot using (a) the pseudo-inverse method and (b) the modified extended Jacobian method. The latter is less oscillatory.

matics). To touch a visual target, the robot must choose a set of joint angles that will cause its finger to be at the target (known in robotics as the *inverse kinematics* problem).

Complex robots are interesting because the inverse kinematics problem has no unique solution: there are many ways for a robot to touch a target. What makes humanoid robots especially interesting is that they have a large number of "extra" joints, organized in a humanlike fashion with several kinematic chains, with dynamic constraints such as balance in addition to geometric constraints. When the finger touches a target, the elbow might be up or down, or the back or waist might be bent to change the shoulder's position. This redundancy is advantageous because it enables a robot to avoid obstacles and joint limits and attain more desirable postures. From a control and learning point of view, however, redundancy also makes it quite complicated to find good movement plans. How do we humans decide what to do with our extra joints, and how should humanoid robots control all their joints to make a coordinated movement?

To solve this problem, we first used a redundant inverse kinematics algorithm known as the extended Jacobian method, which reliably gave us reasonable answers. We also developed a more computationally efficient version of the extended Jacobian that searches for appropriate joint angles to reach toward a target and simultaneously optimizes a criterion (such as minimizing gravitational load). We improved the algorithm by making it search locally (using gradient descent) for a better set of joint angles in the nearby space of all joint angle vectors that successfully caused the robot to touch the target. This local search let us remove a lot of calculations, for which we compensated by using other learning algorithms.[5]

The speed we gained let us apply the algorithm to our 30-DOF humanoid robot in real time. We compared the algorithm's performance with a different state-of-the-art algorithm that uses the pseudo-inverse with optimization. In both cases, DB started in a nonoptimal posture and tried to follow a target with its right hand. The target moved with pseudorandom motion generated by summing sinusoids of various frequencies. Deviations from a nominal posture were penalized in the optimization criterion. Our algorithm had much better convergence than the other, as Figure 3 shows.

The work just described implements a classical way to make choices: imposing optimization criteria on movement planning—for instance, by requiring that the system accomplish a task in minimum time or with minimal energy expenditure. However, finding good cost functions that generate appropriate behavior is difficult. Our research on trajectory planning explores an alternative method of constraining complex movement planning—by building movements from two kinds of movement primitives. The first kind is known in neuroscience as *motor tapes*, in that the robot stores an explicit representation of a movement trajectory in memory. When the robot needs information on how to pitch a baseball, it finds the appropriate tape or template in memory and executes it. More sophisticated versions of this approach blend and edit a set of tapes to produce a movement.

Another kind of movement primitive is based on *dynamical systems*. We are exploring simple dynamical systems that can generate either discrete or rhythmic movements.[2] In this case, the robot initially needs only speed and amplitude parameters to start a movement. Learning is required to fine-tune certain additional parameters to improve the movement. This approach lets the robot learn movements by adjusting a relatively small set of parameters. We are also exploring how to

- use these different types of primitives to generate full-body movement,
- learn their parameters using reinforcement learning, and
- sequence and superimpose such movement primitives to accomplish more complex movement tasks.

For example, we have implemented adaptive dynamic systems that enable a robot to drum in time with a human drummer, as Figure 4 shows.[2] This ability to synchronize to external stimuli is an important component of interactive humanoid behavior.

Inspiration from biology also motivates a related trajectory-planning project on which we are working. A common feature in the brain is to employ topographic maps as basic representations of sensory signals. Such maps can be built with various neural-network approaches—for instance, Self-Organizing Maps or Topology Representing Networks.[6] From a statistical point of view, topographic maps can be thought of as neural networks that perform probability density estimation with additional knowledge about neighborhood relations. Density estimation is a powerful tool for performing mappings between different coordinate systems, for performing sensory integration, and for serving as a basic representation for other learning systems. Topographic maps can also perform spatial computations that generate trajectory plans. For instance, using diffusion-based path-planning algorithms, we were able to learn obstacle avoidance algorithms. This work is also interesting from a biological point of view, because the usefulness of topographic maps in motor control is far from understood.

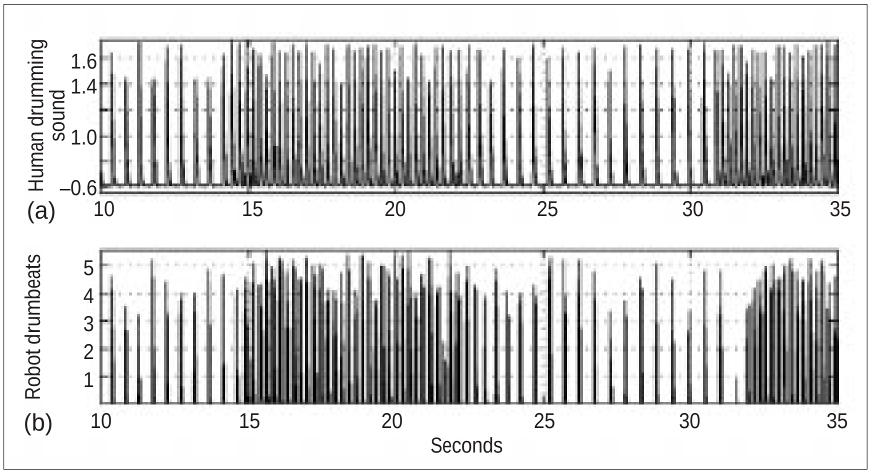

Figure 4. The robot drums in synchrony with external sounds: (a) the average magnitude of the sound the robot hears; (b) the robot drumbeats measured by a vibration sensor on the drum.

Learning

We are interested in how people and machines can learn from sensory information to acquire perceptual and motor skills. So, we are exploring neural networks, statistical learning, and machine learning algorithms. We are investigating three areas: supervised and unsupervised learning, learning from demonstration, and reinforcement learning.

Supervised and unsupervised learning. Function approximation can be used to learn nonlinear coordinate transformations and internal models of the environment.[7] Working with humanoid robots has forced us to develop algorithms that

- learn incrementally as training data is generated,
- learn in real time as the robot behaves, and
- scale to complex, high-dimensional learning problems.

Idealized engineering models often do not accurately model the mechanisms used to build humanoid robots. For example, rigid-body-dynamics models perform poorly for lightweight systems dominated by actuator dynamics, as is the case with our current robot DB. Therefore, we are developing learning algorithms and appropriate representations to acquire useful models automatically. Our ultimate goal is to compare the behavior of these learning algorithms with human (for example, cerebellar) learning.[8]

One algorithm that can deal with the high dimensionality of humanoid robot learning is *locally weighted projection regression*.[9] LWPR models data with many local models, each one assuming only a few directions. This is similar to what sigmoidal feedforward neural networks with one hidden layer do. Each hidden-layer neuron applies a simple 1D function to a weighted sum of its inputs; this is equivalent to placing that function in a certain direction in the input space. We collected data on the distribution of both human and robot arm movement and developed computationally efficient methods to correlate the inputs and the output. Using principal components analysis (PCA), we discovered that we need only a 4D to 6D model in any small region to fit the data well, even though the movements span a very high-dimensional space. The tough part of this problem is to efficiently determine the important directions in each part of the space. LWPR can do this using the input–output correlations. LWPR

- learns rapidly using second-order learning methods supporting incremental training,
- uses statistically sound stochastic cross-validation to learn,
- adjusts its local weighting kernels (how much and what shape area the local model covers) based only on local information to avoid interference with other models,
- has a computational complexity that is linear in the number of inputs, and
- can detect redundant or irrelevant inputs.

We have tested LWPR on modeling the dynamics of our anthropomorphic robot arm, which has 21 inputs. To make the test more challenging, we added 29 features that were irrelevant random noise. LWPR handled the 50-dimensional learning problem well: on average, each local model was 4D, and there were 325 local models. LWPR used local models of lower dimensions than our previous PCA-based algorithm. To our knowledge, this is the first incremental neural-network learning method that combines all these properties and is well suited for the high-dimensional real-time learning problems posed by humanoid robots.

Learning from demonstration. One way we program our fellow human beings is to show them how to do a task. It is amazing that such complex sensory input is useful for learning. How does the learner know what is important or irrelevant in the demonstration? How does the learner infer the performer's goals? How does the learner generalize to different situations?

A graduate student typically needs at least a year to program one of our humanoid robots to do a task. Humanlike learning from demonstration (LFD) should greatly reduce the cost of programming these complex systems. We also expect that humanoid robots will be asked to perform tasks that people do, tasks that a person can easily demonstrate.

LFD might also provide one of the most important footholds to understand the information processes of sensorimotor control and learning in the brain. People and many animals do not just learn a task from scratch by trial and error. Rather, they extract knowledge about how to approach a problem by watching others perform a similar task

Figure 5. Perceiving human motion. We can see how well our perception algorithms track. (a) A person walks by and his motion is recorded. (b) The perception system lays a graphical model on top of the human motion where it believes the person's body parts are.

and by using what they already know. From the viewpoint of computational neuroscience, LFD is a highly complex problem that requires mapping a perceived action that is given in an external-coordinate (world) frame of reference into a totally different internal frame of reference to activate motor neurons and subsequently muscles. Recent research in behavioral neuroscience has shown that specialized "mirror neurons" in the frontal cortex of primates seem to be the interface between perceived movement and generated movement; that is, these neurons fire very selectively when a particular movement is shown to the primate, and when the primate itself executes the movement. Brain-imaging studies with people are consistent with these results.

Research on LFD also offers tremendous potential for medical and clinical applications. If we can start teaching machines by showing, our interaction with machines will become much more natural. If a machine can understand human movement, it can also be used in rehabilitation as a personal trainer that watches a patient and provides specific new exercises to improve a motor skill. Finally, insights into biological motor control that are developed in LFD can help us build adaptive prosthetic devices.

We hypothesize that a perceived movement is mapped onto a finite set of movement primitives that compete for perceived action. We can formulate such a process in the framework of competitive learning: each movement primitive predicts the outcome of a perceived movement and tries to adjust its parameters to achieve an even better prediction, until a winner is determined. In preliminary studies with humanoid robots, we have demonstrated the feasibility of this approach. Nevertheless, many open problems remain for research. We are also trying to develop theories on how the cerebellum could be involved in learning movement primitives.

To explore these issues, we implemented LFD for a number of tasks, ranging from folk dancing to various forms of juggling. We identified three key challenges:

- to be able to perceive and understand what happens during a demonstration;
- to find an appropriate way to translate the behavior into something the robot can actually do—it is humanoid, not human, so it has many fewer joints and ways to move, and it is weaker and slower than a human; and
- to enable the robot to fill in missing information using learning from practice—many things are hard or impossible to perceive in a demonstration, such as muscle activations or responses to errors that do not occur in the demonstration.

Solving these challenges is greatly facilitated by enabling the robot to perceive the teacher's goal.

Perceiving human movement. To understand a task demonstration, the robot must be able to see what is going on. We have focused on the perception of human movement, exploiting our knowledge of how people move to inform our perception algorithms. For example, one theory is that we move in such a way as to minimize how fast muscle forces change.[7] We can use this theory about movement generation to select the most likely interpretation of ambiguous sensory input.[10]

Our first thought was to borrow motion-capture techniques from the movie and video game industry. We experimented with optical systems that track markers, systems where the teacher wears measurement devices, and vision-based systems with no special markers. However, we found that controlling a physical device rather than drawing a picture required substantially modifying these techniques.

The organizing principle for our perception algorithms is that they should be able to recreate or predict measured images based on the recovered information. In addition, we can make the movement recovery more reliable by adding what are known as *regularization* terms to be minimized. These terms help resolve ambiguities in the sensor data. For example, one regularization term penalizes high rates of estimated muscle force change. We also process a large time range of inputs simultaneously rather than sequentially, so we can apply regularization operators across time and easily handle occlusion and noise. Thus, perception becomes an optimization process, trying to find the underlying movement or *motor program* that predicts the measured data and deviates the least from what we know about human movement.

To deal with systems as complex as the human body and the humanoid robot, we had to use a representation with adaptive resolution. We chose B-spline wavelets. Wavelets are removed when their coefficients are small and added when the prediction error is large. We have also developed large-scale optimization techniques that handle the sparse representations we typically find in observed data. We designed these optimization techniques to be reliable and robust, using second-order optimization with trust regions and ideas from robust statistics. Figure 5 shows an example of our perception algorithms applied to frames from a high-speed video camera.

Translating movement and inferring goals. As one test case for LFD, we captured the movements of a skilled performer doing the Okinawan folk dance Kacha-shi.[1] Using the perception techniques just described, we found that the teacher's motions exceeded the robot's possible joint movements. We had to find a way to modify the demonstration to preserve the "dance" but make it possible for the robot to do. We considered several options:

- Scale and translate the joint trajectories to make them fit within robot joint limits, without taking into account the Cartesian location of the limbs.
- Adjust the visual features the robot is trying to match until they are all within reach. This can be done by translating or scaling the images or 3D target locations. How to do this in a principled way is not clear, and the effects on joint motion are not taken into account.
- Build the joint limits into a special version of the perception algorithms, so that the robot can only "see" feasible postures in interpreting or reasoning about the demonstration. This approach trades off joint errors and Cartesian target errors straightforwardly.
- Parameterize the performance in some way (knot-point locations for splines, for example), and adjust the parameters so that joint limits are not violated. Human observers score how well the original performance's style or essence is preserved and select the optimal set of parameters. This is very time consuming, unless we can develop an automatic criterion function for scoring the motion.

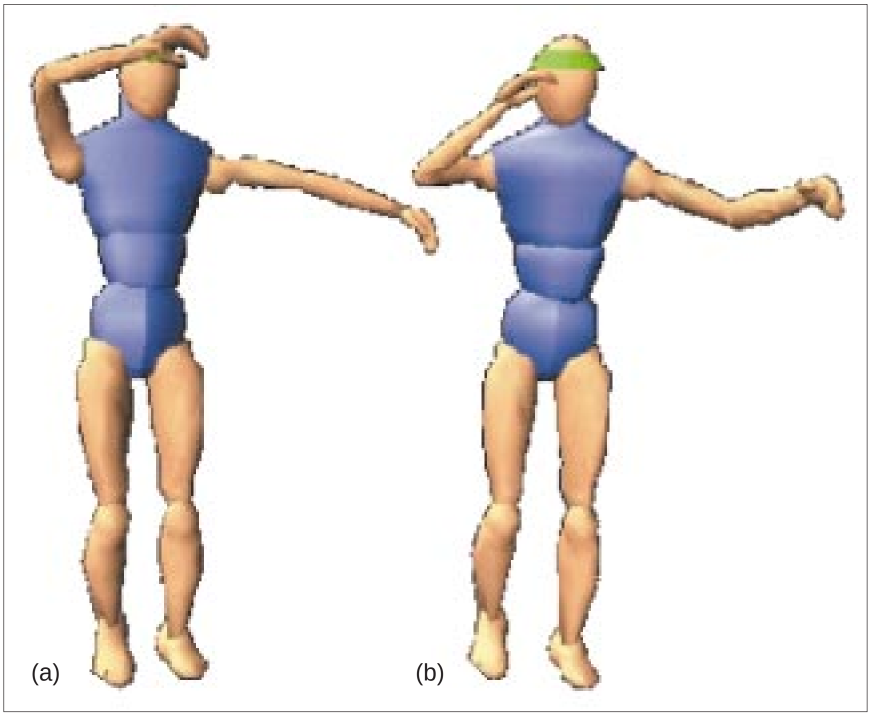

Figure 6. A frame from a graphics visualization of the reconstructed motion. Compared to the human visualization (a), the visualized robot's shoulder and elbow degrees of freedom (b) are constrained.

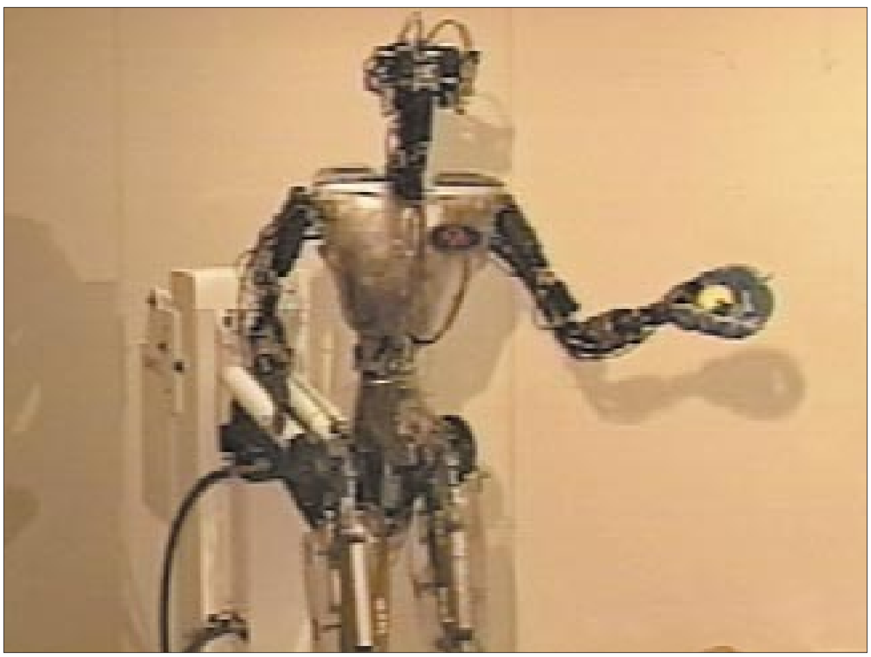

Figure 7. A frame of motion showing the end of a catching sequence.

We implemented the first option, as shown in Figure 6. Clearly, we should also consider the alternative approaches. We learned from this work that we need to develop algorithms that identify what is important to preserve in learning from a demonstration, and what is irrelevant or less important. For example, we have begun to implement catching based on LFD (see Figure 7), where the learned movement must be adapted to new requirements

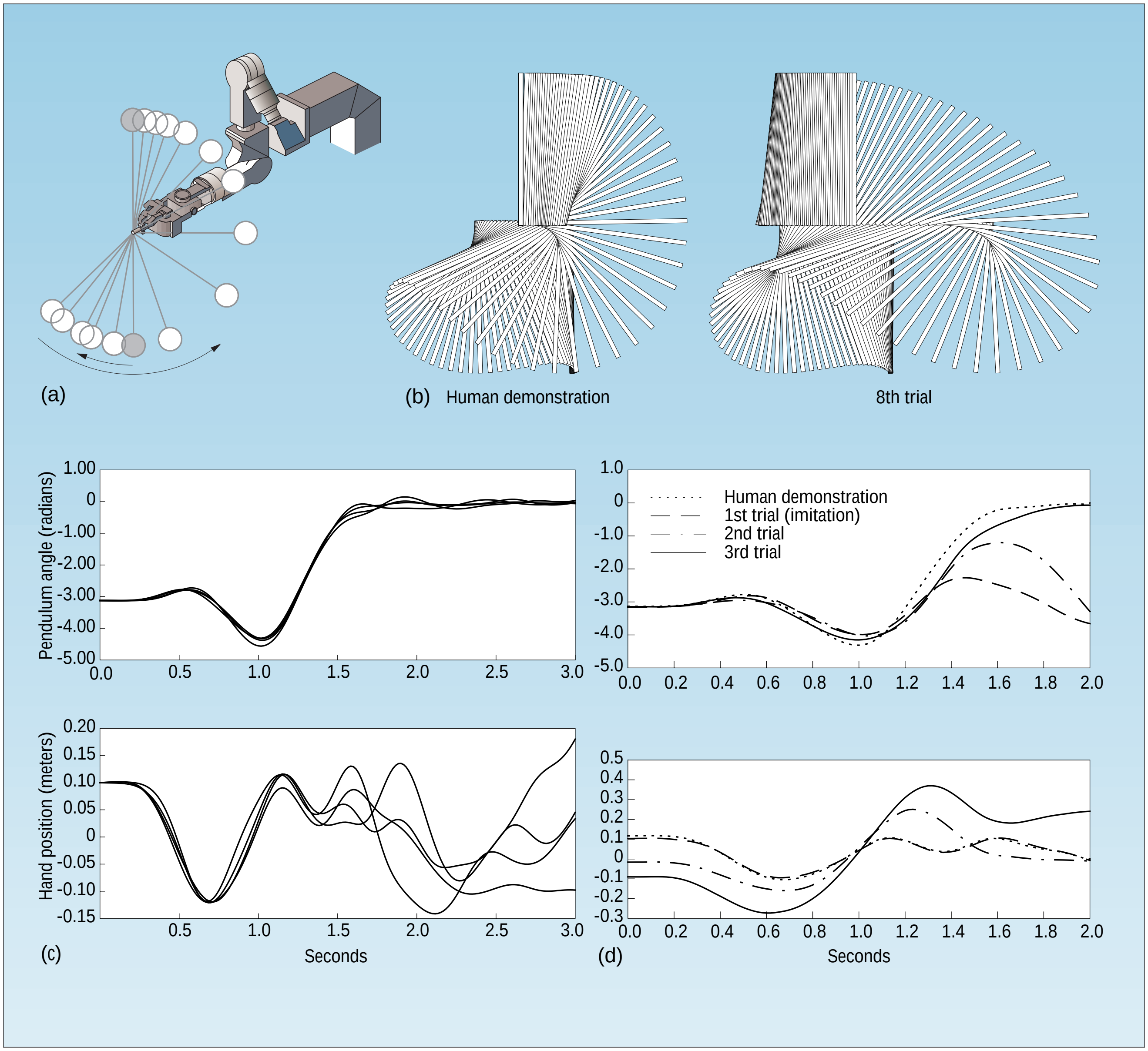

Figure 8. (a) The anthropomorphic robot arm gripping a pendulum. In this configuration, the pendulum axis is aligned with the fingers and with the forearm. (b) The pendulum configurations during a human swing upward and a successful robot swing upward after learning. (c) The pendulum angles and hand positions for several demonstration swing-ups by a person. The pendulum starts at $\theta = -\pi$, and a successful swing up moves the pendulum to $\theta = 0$. (d) The hand and pendulum motion during robot LFD using a nonparametric model.

such as the ball's trajectory.[1] In catching, the hand must intercept the ball at the right place in space and at the right time; the joint angle trajectories are secondary.

We have begun to implement learning how to juggle three balls from demonstration, where actuator dynamics and constraints are crucial. Because the hydraulic actuators limit the joint velocities to values below that observed in human juggling, the robot must significantly modify the observed movements to juggle successfully. We have manually implemented several feasible juggling patterns; Figure 1 shows one such pattern.

In summary, something more abstract than motion trajectories must be transferred in LFD. The robot must perceive the teacher's goals to perform the necessary abstraction. We are exploring alternative ways to do this.

Learning from practice using reinforcement learning. After the robot observes the teacher's demonstration, it still must practice the task, both to improve its performance and to estimate quantities not easily observable in the demonstration. In our LFD approach, the robot learns a reward function from the demonstration that then lets it learn from practice without further demonstrations.[12] The learned function rewards robot actions that look like the observed demonstration. This simple reward function does not capture the true goals of actions, but it works well for many tasks.

The robot also learns models of the task from the demonstration and from its repeated attempts to perform the task. Knowledge of the reward function and the task models lets the robot compute an appropriate control mechanism. Using these methods, our robot arm learned how to balance a pole on a finger tip in a single trial. The arm also learned the harder task of swinging a pendulum from a downward-pointing position to point up (see Figure 8).

We learned these lessons from these implementations:

- Simply mimicking demonstrated motions is often not adequate.
- Given the differences between the human teacher and the robot learner and the small number of demonstrations, learning the teacher's policy (what the teacher does in every possible situation) is often impossible.
- However, a task planner can use a learned model and a reward function to compute an appropriate policy.
- This model-based planning process supports rapid learning.
- Both parametric and nonparametric models can be learned and used.
- Incorporating a task-level direct learning component that is non-model-based, in addition to the model-based planner, is useful in compensating for structural modeling errors and slow model learning.

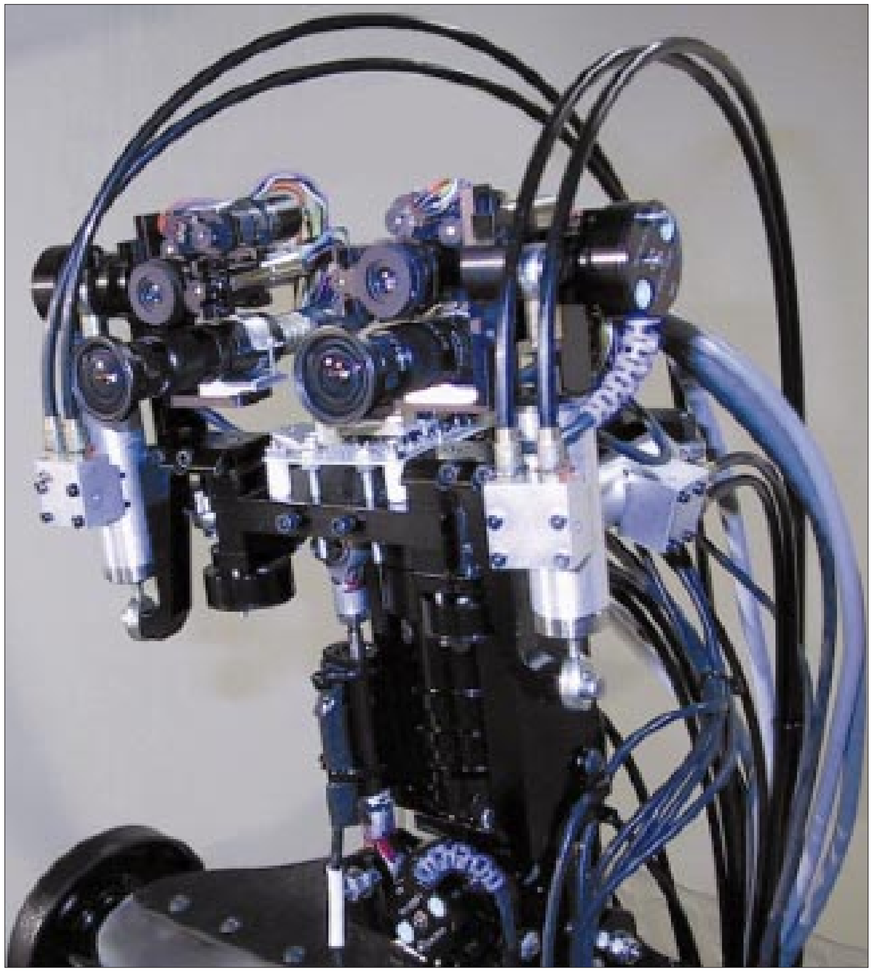

Figure 9. A close-up of the robot head, showing the wide-angle and narrow-angle cameras that serve as eyes.

Oculomotor control

The humanoid robot's complexity forces us to develop autonomous self-calibration algorithms. Initially we are focusing on controlling eye movements, where perception and motor control strongly interact. For example, the robot must compensate for head rotation by counter-rotating the eyes, so that gaze is stabilized. This behavior is known as the *vestibulo-ocular reflex*. Miscalibration of VOR behavior strongly degrades vision, especially for the robot's narrow-field-of-view cameras that provide its "foveal" vision.

We are exploring a learning algorithm known as *feedback error learning*, where we use an error signal (in this case an image slip on the retina during head motion) to train a control circuit. This approach is modeled on the adaptive control strategies used by the primate cerebellum. We used eligibility traces, a concept from biology and reinforcement learning, to compensate for unknown delays in the sensory feedback pathway.

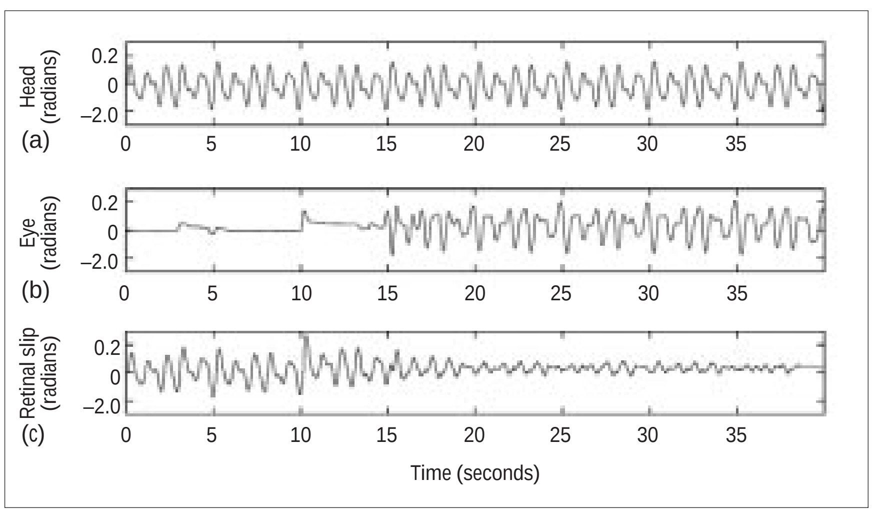

Figure 10. The robot's (a) head position, (b) eye position, and (c) retinal image slip during vestibulo-ocular reflex learning.

In experiments, our humanoid oculomotor system (see Figure 9) converged to excel-

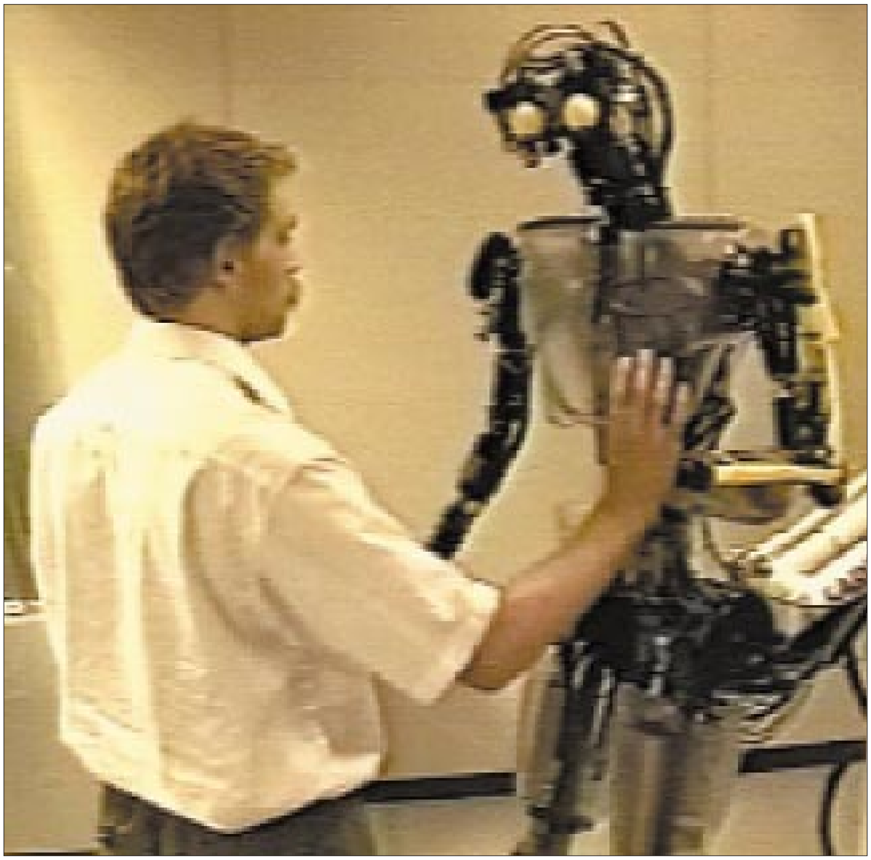

Figure 11. Sticky-Hands interaction with a humanoid robot.

(a) (b)

Figure 12. The Sensuit motion-capture system.

lent VOR performance after approximately 30 to 40 seconds (see Figure 10), even in the presence of control system nonlinearities. Our future work will address adding smooth pursuit and saccadic behavior and enabling all these learning systems to run simultaneously without interfering with each other.

Interactive behaviors

We have explored two kinds of interactive behavior with DB: catching and a T'ai Chi exercise known as Sticky Hands or Push Hands.[3] The work on catching forced us to develop trajectory generation procedures that can respond flexibly to demands from the environment, such as where the ball is going. The work on Sticky Hands explored robot force control in contact with a person (see Figure 11). This task involves the person and the robot moving together through varied and novel patterns while keeping the contact force low. Sometimes the human "leads" or determines the motion, sometimes the robot leads, and sometimes it is not clear who leads.

A key research issue in generating interactive behaviors is generalizing learned motions. It also became clear that when people interact with a humanoid robot, they expect rich and varied behavior from all parts of the body. For example, it is disconcerting if the robot does not exhibit humanlike eye and head movements or fails to appear to be attending to the task. Interacting with the robot rapidly becomes boring if the robot always responds in the same way in any given situation. How can the robot recognize a particular style of interaction and respond appropriately? If humanoid robots are going to interact with people in nontrivial ways, we will need to address these issues as well as control and learning issues.

Understanding human behavior

We are using a variety of motion capture systems to understand the psychophysics of human movement. We are also exploring how our theories implemented in DB compare to human behavior to find out which movement primitives biological systems employ and how the brain represents such primitives. One goniometer-based measurement system is the Sarcos Sensuit (see Figure 12), which simultaneously measures 35 DOFs of the human body. It can be used for real-time capture of full-body motion, as an advanced human–computer interface, or to control sophisticated robotic equipment. The complete Sensuit, worn like an exoskeleton, does not restrict motion for most movements, while an array of lightweight Hall-effect sensors records the relative positions of all limbs at sampling rates up to 100 Hz. A platform-independent OpenGL graphical display can be used to simultaneously show the captured motion in real time and to generate and play back animated sequences of stored data files.

Our primary interest is to analyze human data from the Sensuit and other motion capture and vision systems with respect to certain task-related movements. One key question we seek to answer in this context is how the human motor cortex efficiently analyzes, learns, and recalls an apparently infinite number of complex movement patterns while being limited to a finite number of neurons and synapses. Are there underlying regularities, invariances, or constraints on human behavior? We have already discussed how we can reduce the dimensionality of the movement data in any local neighborhood to under 10 dimensions, and how we have observed that people tend to move so as to minimize the rate of change of muscle forces. These preliminary studies will help us develop new concepts for controlling humanoid robotic systems with many degrees of freedom.

PROGRAMMING HUMANLIKE BEhaviors in a humanoid robot is an important step toward understanding how the human brain generates behavior. Three levels are essential for a complete understanding of brain functions: the computational-hardware level, information representation and algorithms, and computational theory. We are studying high-level brain functions using multiple methods such as neurophysiological analysis of the basal ganglia and cerebellum, psychophysical and behavioral analysis of visuo-motor learning, measurement of brain activity using scanning techniques such as MRI, mathematical analysis, computer simulation of neural networks, and robotics experiments using humanoid robots. For instance, one of our approaches is trying to have a robot learn a neural-network model for motor learning that includes data from psychophysical and behavioral experiments as well as from brain MRIs. The robot reproduces a learned model in a real task, and we can verify the model's ability to generate appropriate behavior by checking its robustness and performance. This is only one example of the attention being given to the study of brain functions using humanoid robots. This body of work should be an important step toward changing the future of brain science. ■

Acknowledgments

This work is a joint project between the ATR Human Information Processing Research Laboratories and the Kawato Dynamic Brain Project, an Exploratory Research for Advanced Technology (Erato) project funded by the Japan Science and Technology Agency. It was also funded by National Science Foundation Awards 9710312 and 9711770 as well as a University of Southern California Zumberge grant.

References

1. M. Riley, A. Ude, and C.G. Atkeson, "Methods for Motion Generation and Interaction with a Humanoid Robot: Case Studies of Dancing and Catching," *Proc. 2000 Workshop on Interactive Robotics and Entertainment*, Robotics Inst., Carnegie Mellon Univ., Pittsburgh, 2000, pp. 35–42.
2. S. Schaal, S. Kotosaka, and D. Sternad, "Nonlinear Dynamical Systems as Movement Primitives," *Proc. Humanoids 2000, IEEE-RAS Int'l Conf. Humanoid Robots*, to be published by IEEE Press, Piscataway, N.J., 2000.
3. J.G. Hale and F.E. Pollick, " 'Sticky Hands' Interaction with an Anthropomorphic Robot," *Proc. 2000 Workshop on Interactive Robotics and Entertainment*, Robotics Inst., Carnegie Mellon Univ., Pittsburgh, 2000, pp. 117–124.
4. T. Shibata and S. Schaal, "Fast Learning of Biomimetic Oculomotor Control with Nonparametric Regression Networks," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE Press, Piscataway, N.J., 2000, pp. 3848–3854.
5. G. Tevatia and S. Schaal, "Inverse Kinematics for Humanoid Robots," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE Press, Piscataway, N.J., 2000, pp. 294–299.
6. T. Martinetz and K. Schulten, "Topology Representing Network," *Neural Networks*, Vol. 7, No. 3, 1994, pp. 507–522.
7. S. Schaal, C.G. Atkeson, and S. Vijayakumar, "Real-Time Robot Learning with Locally Weighted Statistical Learning," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE Press, Piscataway, N.J., 2000, pp. 288–293.
8. M. Kawato, "Internal Models for Motor Control and Trajectory Planning," *Current Opinion in Neurobiology*, Vol. 9, 1999, pp. 718–727.
9. S. Vijayakumar and S. Schaal, "Fast and Efficient Incremental Learning for High-Dimensional Movement Systems," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE Press, Piscataway, N.J., 2000, pp. 1894–1899.
10. A. Ude, C.G. Atkeson, and M. Riley, "Planning of Joint Trajectories for Humanoid Robots Using B-spline Wavelets," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE Press, Piscataway, N.J., 2000, pp. 2223–2228.
11. C.G. Atkeson and S. Schaal, "How Can a Robot Learn from Watching a Human?" *Proc. 14th Int'l Conf. Machine Learning*, Morgan Kaufmann, San Francisco, 1997, pp. 12–20.

Christopher G. Atkeson is an associate professor in the Georgia Institute of Technology College of Computing. His research focuses on numerical approaches to machine learning and uses robotics to explore the behavior of learning algorithms. He also is applying machine learning to future computer environments. Contact him at the College of Computing, Georgia Inst. of Technology, 801 Atlantic Dr., Atlanta, GA 30332-0280; cga@cc.gatech.edu; www.cc.gatech.edu/fac/Chris.Atkeson.

Joshua G. Hale is a PhD student in computer science at the University of Glasgow and a member of the 3D-Matic Faraday Partnership. His research interests include human-figure animation, dynamic simulation, and biomechanics. He received his BA in computation from the University of Oxford and his MSc in computer science from the University of Edinburgh. Contact him at Boyd Orr, Room 418, Univ. of Glasgow, Glasgow G12 8QQ, Scotland; halej@dcs.gla.ac.uk; www.dcs.gla.ac.uk/~halej.

Mitsuo Kawato is a department head in ATR Human Information Processing Research Laboratories and the director of the Kawato Dynamic Brain Project. He received his BS in physics from Tokyo University and his ME and PhD in biophysical engineering from Osaka University. Contact him at Department. 3, ATR HIP Labs, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan; kawato@hip.atr.co.jp; www.hip.atr.co.jp/departments/dept3.html.

Shinya Kotosaka is a researcher in the Kawato Dynamic Brain Project. He received his PhD in mechanical engineering from Saitama University, Japan. He is a member of the Society for Neuroscience, the Japan Society of Mechanical Engineers, and the Robotics Society of Japan. Contact him at 2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan; kotosaka@erato.atr.co.jp.

Frank Pollick is a lecturer in psychology at the University of Glasgow. From 1991 to 1997 he was an invited researcher at ATR Human Information Processing Research Labs in Kyoto, Japan. His research interests include the visual control of action and the production and perception of human movement. He received his PhD in cognitive sciences from the University of California, Irvine, his MSc in biomedical engineering from Case Western Reserve University, and a BS degree in physics and in biology from the Massachusetts Institute of Technology. Contact him at the Psychology Dept., Univ. of Glasgow, 58 Hillhead St., Glasgow G12 8QB, Scotland; frank@psy.gla.ac.uk; www.psy.gla.ac.uk/frank/home.html.

Marcia Riley is a PhD student at the Georgia Institute of Technology College of Computing. Her research interests include generating interesting behavior for humanoids, and she explores incorporating techniques and results from human motor control studies and machine learning to create behaviors. She earned her MS in computer science from the University of Geneva, Switzerland, and her BS in mathematics from The Johns Hopkins University. Contact her at the College of Computing, Georgia Tech, 801 Atlantic Dr., Atlanta, GA 30332-0280; mriley@cc.gatech.edu.

Stefan Schaal is an assistant professor of computer science and neuroscience at the University of Southern California, head of the Computational Learning Group of the Kawato Dynamic Brain Project, and an adjunct assistant professor of kinesiology at the Pennsylvania State University. His research interests include statistical and machine learning, neural networks, computational neuroscience, nonlinear dynamics, nonlinear control theory, and biomimetic robotics. Contact him at the Computational Learning and Motor Control Lab, USC, Hedco Neurosciences Bldg., HNB-103, 3614 Watt Way, Los Angeles, CA 90089-2520; sschaal@usc.edu.

Tomohiro Shibata is a researcher at the Japan Science and Technology Corporation and is working on the Erato project. He received his PhD in information engineering from the University of Tokyo. His research focuses on oculomotor control and visual processing for humans and humanoids. He is a member of the Society for Neuroscience; the Institute of Electronics, Information and Communication Engineers; the Robotics Society of Japan; and Japan Cognitive Science Society. Contact him at 2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan; tom@erato.atr.co.jp.

Gaurav Tevatia is a PhD student in computer science at the University of Southern California. He is a member of the Computational and Motor Control Lab at USC and the Computational Learning Group of the Kawato Dynamic Brain Project. His research interests include statistical and machine learning, neural networks, and nonlinear control theory. He completed his undergraduate studies at Punjab University, India. Contact him at the Computational Learning and Motor Control Lab, USC, Hedco Neurosciences Bldg., HNB-103, 3614 Watt Way, Los Angeles, CA 90089-2520; tevatia@usc.edu.

Ales Ude is an Science & Technology Agency fellow in the Kawato Dynamic Brain Project and a researcher at the Jozef Stefan Institute in Ljubljana, Slovenia. His research focuses on computer vision and robotics. He studied applied mathematics at the University of Ljubljana, Slovenia, and received his PhD in computer science from the University of Karlsruhe, Germany. Contact him at 2-2 Hikaridai, Seika-cho, Soraku-gu, Kyoto 619-0288 Japan; aude@isd.atr.co.jp.

Sethu Vijayakumar is a researcher at the Computational Learning and Motor Control Lab at the University of Southern California and holds a part-time affiliation with the Riken Brain Science Institute, Japan. His research interests include statistical and machine learning, neural networks, and computational neuroscience. He received his BE in computer science from the Regional Engineering College, Tiruchirapalli, India, and his master's and PhD from the Tokyo Institute of Technology. Contact him at the Computational Learning and Motor Control Lab, USC, Hedco Neurosciences Bldg., HNB-103, 3614 Watt Way, Los Angeles, CA 90089-2520; sethu@usc.edu.