Cheek to Chip: Dancing Robots and AI's Future

Jean-Julien Aucouturier, *University of Tokyo*

It probably started as a funny one-minute demo in the entertainment industry, but more and more AI researchers are trying to make robots dance to music. And as the ideas and technologies develop, it's clear that dancing robots can be serious indeed. In this installment of Trends & Controversies, you'll see how they address issues that are central to modern AI—and how they do so in original ways.

- *Situated knowledge*. The view of knowledge as the dynamical representation of actions, rather than facts, involves learning by conceiving ways to do, feel, or move—often by observing the actions of others. The ability to map such actions from one body to another is therefore a key component of intelligence. Katsushi Ikeuchi and his colleagues show how a robot can imitate a human dancer by adapting the movements it observes to its own motor architecture. This requires, they say, intermediate representations of motor primitives observable in the human actions and then instantiated according to the robot body.
- *Symbol grounding*. Because dance maps sounds to movements, it addresses the problem of building stable sensorimotor signatures over different modalities—that is, a simple amodal symbolic system. Tetsuya Ogata, Hideki Kozima, and Hiroshi Okuno show that such structures can be acquired from environmental observations. They've developed a system that learns features simultaneously in both the sound and gesture domains. In this way, mappings reflect correspondences in the real world—for example, things make a "bang" when they fall down.
- *Social intelligence*. Finally, dance is a tractable metaphor for studying social interactions and how the body mediates them. Marek Michalowski and Hideki Kozima report on rhythmic imitation as a prerequisite to interaction. Their studies show that children interact longer with a robot if it shows rhythmic mimetism. However, pure imitation isn't sufficient to sustain interaction, which also needs some degree of creativity. In the fourth essay, Yuta Ogai, Takashi Ikegami, and I show how to implement this through chaotic dynamics. Of course, a truly socially intelligent robot should recognize and compensate for variances in its human partners and plan accordingly. In the final essay, Kazuhiro Kosuge, Takahiro Takeda, and Yasuhisa Hirata take literal steps in this direction: their Dance Partner Robot dynamically responds to forces and moments exerted by the human partner's dance lead.

I hope you'll enjoy reading this installment of Trends & Controversies as much as I did preparing it. —*Jean-Julien Aucouturier*

Robots That Learn to Dance from Observation

Katsushi Ikeuchi, Takaaki Shiratori, and Shunsuke Kudoh, *University of Tokyo*
Hirohisa Hirukawa, Shin'ichiro Nakaoka, and Fumio Kanehiro, *National Institute of Advanced Industrial Science and Technology, Japan*

Recent generations of humanoid robots increasingly resemble humans in shape and articulatory capacities. This progress has motivated researchers to design dancing robots that can mimic the complexity and style of human choreographic dancing. Such complicated actions are usually programmed manually and ad hoc. However, this approach is both tedious and inflexible.

Researchers at the University of Tokyo have developed the *learning-from-observation* (LFO) training method to overcome this difficulty.[1,2] LFO enables a robot to acquire knowledge of what to do and how to do it from observing human demonstrations. Direct mapping from human joint angles to robot joint angles doesn't work well because of the dynamic and kinematic differences between the observed person and the robot (for example, weight, balance, and arm and leg lengths). LFO therefore relies on predesigned task models, which represent only the actions (and features thereof) that are essential to mimicry. It uses task models to recognize and parse the sequence of human actions—for example, "Now, pick up the box." Then it adapts these actions to the robot's morphology and dynamics so that it can mimic the movement. This indirect, two-step mapping is crucial for robust imitation and performance.

LFO has been successfully applied to various hand-eye operations.[1,2] Here we describe how to extend it to a dancing humanoid.

Modeling leg and upper-body tasks

Because the leg and upper body have different purposes as well as different motor constraints, we apply different strategies to design task models for leg[3] and upper-body[4] motions. The leg motions stably support robot bodies,

Published by the IEEE Computer Society

while the upper-body motions express dancing patterns. We concatenate and adjust these two motion types in the final stage.[5]

Defining the leg-task model

Using a top-down analytic approach,[3] we define the four leg-task models shown in figure 1: Stand, Squat, R-step, and L-step. These models consider the state of contact between the humanoid's feet and the floor. For Stand and Squat, both feet are in contact with the floor; the difference is in the humanoid's waist position. R-step and L-step represent single-foot contact with the floor, with one swinging foot in the air.

Each task model has its own skill parameters that determine the timings and characteristics of the motion needed to achieve the task. All the task models have timing parameters to denote the beginning and ending period. These values enable tasks to be arranged in a time sequence to support choreography and rhythmic performance. Each task model has its own prototypical trajectory of a foot to perform that task. We use positional skill parameters to characterize the trajectory, representing it in a relative coordinate system with respect to the stationary foot touching the ground. This lets us locally modify trajectories in a task sequence.

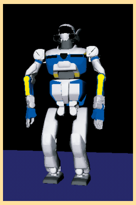
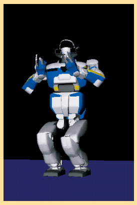
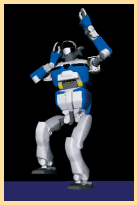
	Standing	Squat	Step
Task model: what to do			 
Skill parameter: how to do	Period	Foot width and depth	Foot width and highest point

Figure 1. Leg-task models for Stand, Squat, R-Step, and L-Step. The skill parameters characterize foot trajectories such as step width or highest point.

Recognizing leg tasks. The LFO program recognizes a leg task sequence from marker trajectories obtained using a motion-capture system.

First, the program recognizes Step tasks from a motion sequence by analyzing a swing foot's trajectory. If a foot marker's speed has continuously positive values in an interval, that interval is a candidate for a Step task. Additionally, to avoid erroneous small intervals, we constrain a minimum moving distance. Then the program classifies the Step tasks as either R-step or L-step, depending on which foot is contacting the ground. After removing the corresponding intervals to Step tasks, it further classifies the remaining intervals into Stand or Squat, depending on the waist positions.

Determining skill parameters. The LFO program determines timing and positional skill parameters for each task model from recognized intervals.

In each task, it obtains timing parameters from the beginning and end of the recognized interval. It calculates the positional parameter values, such as highest point and width, by using positions of related motion-capture markers at these timings. It modifies the prototypical swing-foot trajectory for each task according to the positional skill parameters. From this trajectory, the program calculates a series of joint angles for a swinging foot by solving inverse-kinematics equations.

Generating upper-body motion

Ideally, modeling upper-body motion will conserve the original motion's features as much as possible, because few physical constraints exist in this motion. On the other hand, the configuration and balance between a robot and a human (and even between two humans) differ, so a robot can't follow exact trajectories to mimic a particular performance. We therefore use the same task-model strategy as for leg motions. In this case, each model represents a key pose of the imitated dance.

The key poses are called "Kata" in the Kabuki and Kyogen dance traditions and "Tome" in the Nichibu tradition. They represent a fixed dance posture for impressing viewers with the dancer's body line.[4] It's important in Japanese dance performance to properly represent those key poses with appropriate timings. We therefore employ the strategy that a robot should represent the key poses as closely as possible without sacrificing accuracy in the intervallic trajectories that connect the poses.

Recognizing key poses. We extract key poses from motion sequences with a bottom-up generative method that's based on both brief pauses and musical information. Extracting key poses solely by detecting brief pauses tends to detect too many of them, so we define a key pose as the body configuration when a musical beat co-occurs with a brief motion pause. After analyzing motion and musical information separately, the LFO program combines both results and extracts the key poses.

Figure 2a represents the key poses from a dance textbook. Figure 2b shows key poses automatically extracted from the dance motion sequences of the performance of an Aizu-Bandaisa, a traditional Japanese folk dance. We can easily confirm that our method has a result very close to an expert's understanding.

Generating skill trajectories. The final phase for defining the upper-body motion generates skill trajectories that smoothly connect two consecutive key poses. Our method consists of two steps: hierarchical motion decomposition from key-pose information and motion reconstruction within the mechanical constraints.

The hierarchical motion decomposition employs a hierarchical series of B-spline curves with different knot spacing. Higher layers are based on finer knot spacing and can preserve the higher-frequency components of the original sequence. To construct the robot motions, the LFO program iteratively removes higher layers of B-spine curves so that the generated motion satisfies the robot's physical constraints such as joint angles and torque capacities.

Generating whole-body motion

Given leg motions generated on the basis of the leg-task models and skill parameters and upper-body motion generated on the

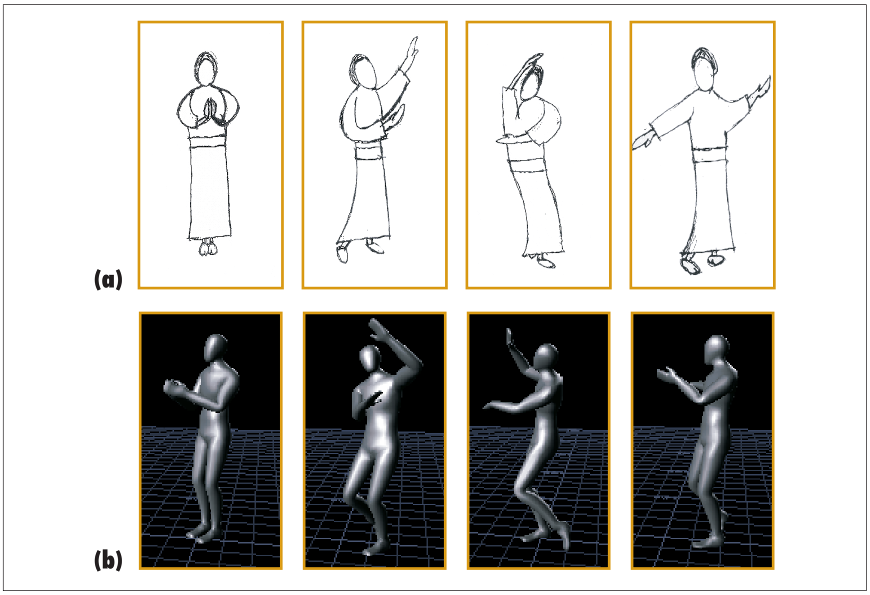


Figure 2. Key poses in an Aizu-Bandaisan dance. (a) The drawings show poses from a dance textbook. (b) The learning-from-observation program extracted these poses from the dance motion sequences.

Figure 3. Humanoid robot HRP-2 in performance. The robot mimics the human dancer's movements within the robot's physical constraints.

basis of key-pose models and skill trajectories, the simplest way to generate whole-body motion is to directly concatenate the leg and upper-body motions. However, because the current LFO program does not consider dynamic conditions such as balance, a humanoid robot often can't execute this motion.

To generate executable motion, we use a dynamic filter and conduct skill refinement. The dynamic filter compensates the zero-moment point and the yaw-axis moment. The skill refinement resolves other kinematic problems, such as self-collision.

We used the National Institute of Advanced Industrial Science and Technology's humanoid robot, HRP-2, in performance. Figure 3 shows the robot performing with a female dancer, whose movements generated the robot motion. The robot is matching the human performance. A video sequence is available at www.cvl.iis.u-tokyo.ac.jp/movie/HRP2_dance_bandaisan.MPG.

Acknowledgments

A team of researchers from the University of Tokyo, the Japan National Institute of Advanced Industrial Science and Technology (AIST), and Kawata Ltd. conducted the research described in this essay. Shin'ichiro Nakaoka designed the leg-motion modules as part of his PhD thesis in computer science at the University of Tokyo. Takaaki Shiratori designed the upper-body modules in his Electric, Information, and Communication departmental PhD thesis at the University of Tokyo. Shunsuke Kudoh developed a theory on the concatenation procedure in his PhD thesis in computer science at the University of Tokyo. Hirohisa Hirukawa and Kenji Kaneko led the AIST humanoid-robot group in designing the dynamic filters, as employed here. The research received partial financial support from Japan Science and Technology's Core Research for Evolutional Science and Technology program.

References

1. K. Ikeuchi and T. Suehiro, "Toward an Assembly Plan from Observation Part I: Task Recognition with Polyhedral Objects," *IEEE Trans. Robotics & Automation*, vol. 10, no. 3, 1994, pp. 365–385.
2. J. Takamatsu et al., "Recognizing Assembly Tasks through Human Demonstration," *Int'l J. Robotics Research*, vol. 26, no. 7, 2007, pp. 641–659.
3. S. Nakaoka et al., "Learning from Observation Paradigm: Leg Task Models for Enabling a Biped Humanoid Robot to Imitate Human Dance," *Int'l J. Robotics Research*, vol. 26, no. 8, 2007, pp. 829–844.
4. T. Shiratori, A. Nakazawa, and K. Ikeuchi, "Detecting Dance Motion Structure through Music Analysis," *Proc. 6th Int'l Conf. Automatic Face and Gesture Recognition* (FGR 04), IEEE, 2004, pp. 857–862.
5. S. Kudho, T. Komura, and K. Ikeuchi, "Modeling and Generating Whole Body Motion of Balance Maintenance," *Systems and Computers*, vol. 37, no. 13, 2006, pp. 11–19.

Motion from Sound: Intermodal Neural Network Mapping

Tetsuya Ogata and Hiroshi G. Okuno, *Kyoto University*
Hideki Kozima, *National Institute of Information and Communications Technology, Japan*

Robots that interact with humans must be able to react to multimodal sensory input. Most conventional studies have handled these multimodal sensory data streams independently, before synchronizing and integrating them into a recognized event. However, designing this integration for robots is generally quite difficult. Humans deal naturally with cross-modal information—for example, they can express auditory information, such as the sounds of a collision, by using physical gestures—for example, moving the hand quickly and stopping it sharply. We call this intermodality mapping.

Our study aims to design a technological method for intermodal mapping. We apply the method not only to generate robot motion from various sounds but also to generate sounds from motions.

Intermodality mapping

Our intermodality-mapping procedure con-

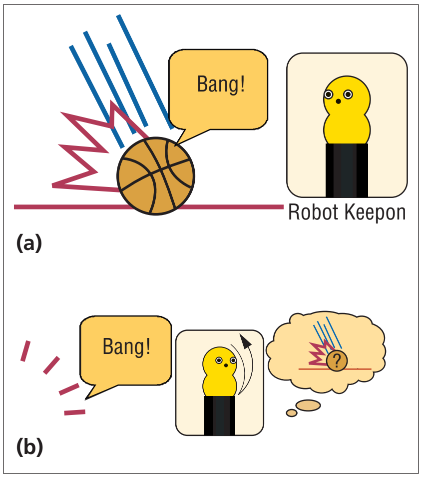


Figure 4. Intermodal mapping. The robot (a) looks at sounds and (b) associates the sound with a motion.

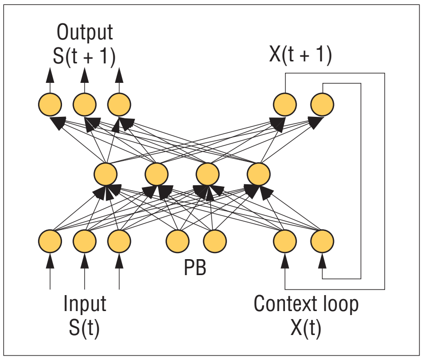


Figure 5. The recurrent-neural-network model with parametric bias (RNNPB) configuration. The model is trained using the back-propagation-through-time algorithm that organizes the values that encode input dynamics into special PB nodes.

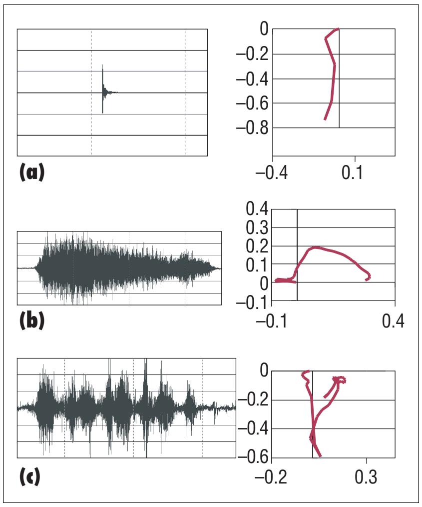


Figure 6. The sounds and generated motions for three novel events: (a) clapping, (b) a spray jet, and (c) shaking a plastic bag. The left-side graphs show sounds, which lasted 2 seconds. The vertical axis shows power, and the horizontal axis shows time. The red lines in the right-side graphs show motion trajectories of the robot head. The black vertical lines show the measures of scale normalized from –1 to 1. The point (0, 0) means the center.

sists of two phases (see figure 4).[1] First, in the learning phase, the robot observes some events together with associated sounds. The robot memorizes these sounds along with the motions of the sound source (figure 4a). Then, in the interacting phase, the robot receives limited sensory information from a single modality as input. The robot associates this with different modality information and expresses it. For example, in figure 4b, the robot uses its body to express its understanding of the falling motion of a given sound source.

The neural network model

Constructing a database that can systematically store all environmental sounds is almost impossible. To achieve intermodal mapping, the robot must be able to generalize various sounds from the limited examples it can actually observe. Figure 5 shows the recurrent-neural-network model with parametric bias (RNNPB) that we use for this purpose.[2] The RNNPB model is a predictor that uses the current state-vector as input for outputting the next state-vector. The model articulates complex time sequences into units that it encodes not as trajectories but as dynamics of limit cycling or fixed-point attractors.

Like the Jordan-type RNN,[3] the RNNPB model is trained using the back-propagation-through-time algorithm. The difference is that the RNNPB model self-organizes the values that encode the input dynamics into special parametric-bias nodes.

The model has three modes:

- In learning mode, the RNNPB model uses the back-propagated errors of each sensorimotor output to update the weight connections and PB values.
- In the generation mode, it inputs the PB value for a desired sequence to the PB node. The desired sequence is generated by closing a loop in which the current step's output becomes the input data for the next step.
- In recognition mode, the corresponding PB value is obtained for a given sequence by updating only the PB values.

Robot implementation

For our experiments, we used the Keepon robot[4] (also featured in the next essay). Keepon was developed mainly for communicative experiments with infants; it's approximately 12 cm tall with four degrees of freedom and incorporates two cameras in its eyes and one microphone on its nose.

In the learning mode, the robot observes several events where a human manipulates a solid blue box. The robot system normalizes the box's positions in camera images and the input sound's values in a mel-filter bank (that is, an array of filtered values corresponding to the results of filtering the sound spectrum through multiple filters). The robot system synchronizes for input to the RNNPB. Next, in the observation mode, the robot estimates the RNNPB parameters using the network's prediction errors. Finally, in the generation mode, the robot uses the estimated RNNPB parameters to reproduce the multimodal sensory flow. Keepon reproduces a trajectory by moving along two of its four degrees of freedom (pitch and yaw). Alternatively, it can output colored noise by multiplying white noise with the mel-filter-bank value obtained from the RNNPB output.

Keepon observed four kinds of manipulations of the blue box along with different types of sound: rotating on the wall, moving back and forth on the table, turning over on the table, and falling on the table. The robot observed each event three times. The numbers of neurons in the input, middle, context, and PB layers were set to 8, 35, 25, and 2, respectively. The event lengths were 10 to 40 steps (0.5 to 2 seconds).

We then investigated the PB values corresponding to three novel sounds: clapping, a spray jet, and a plastic bag being shaken randomly.

Study results and discussion

Figure 6 shows graphs of the novel sounds and the generated motions of the robot's head after hearing them. Figure 6a shows the clapping sound features, which clustered similarly to the features of the learned falling sound. So, the generated clapping motion is similar to that of the falling manipulation for the blue box. Similarly, figure

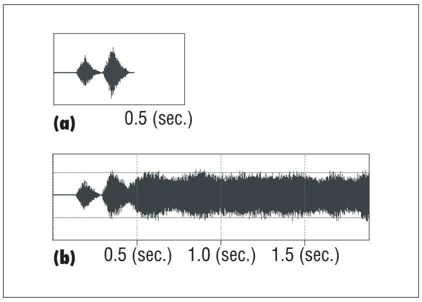


Figure 7. The generated sounds for (a) quick and (b) slow sliding motions.

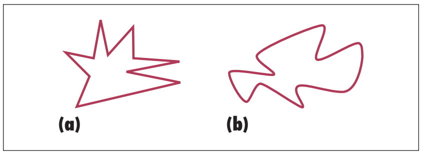


Figure 8. The test figures in the bouba-kiki effect: (a) sharply angled and (b) rounded. People tend to associate the sharply angled shape with the sharper-sounding word and vice versa.

6b shows the generated motion for the spray jet sound, which is similar to that of the rotating manipulation for the blue box.

As figure 6c shows, the motion generated from the sound of the plastic bag is more complex. It contains various sound features, and the motion reflects complex characteristics such as shaking up and down and rotations.

Keepon also generated novel sounds from unknown silent motions such as moving the box horizontally quickly (see figure 7a) and slowly (see figure 7b). The generated sound for the quick motion is similar to that of the falling manipulation. Similarly, the generated sound for the slow motion is similar to that of the rotating manipulation.

The *bouba-kiki effect* is one of the most important for understanding intermodality mapping.[5] The phenomenon gets its name from experiments conducted by Vilayanur Ramachandran and Edward Hubbard.[5] When they asked subjects which of the two shapes depicted in figure 8 corresponded to which of these two meaningless words, 95 percent of them answered that "kiki" corresponded to the sharply angled shape.

The bouba-kiki effect indicates synesthesia, an innate intermodal phenomenon that we can interpret as follows: When objects move along a sharply angled shape (see figure 8a), they make sounds with a rising edge, which are similar to the utterance of "kiki" in power modulations. When objects move along a rounded shape (see figure 8b), they make sounds with a gradual edge, which are similar to the utterance of "bouba" in power modulations. On the basis of this consideration, we can say that our Keepon robot could respond to Ramachandran and Hubbard's question in almost same way as human subjects.

Future work

Applying our method to a humanoid robot is an interesting challenge for future work. One crucial problem is selecting the joints to best express the sounds. We selected the joints available for Keepon in advance. Now we're focusing on *body babbling*, a process that enables infants to acquire mapping between their body-part configurations and sensory experience, such as visual movements and sounds. We'll try to introduce body babbling into our mapping model.

References

1. Y. Hattori et al., "Robot Gesture Generation from Environmental Sounds Using Intermodality Mapping," *Proc. Int'l Workshop Epigenetic Robotics* (EpiRob 05), 2005, pp. 139–140.
2. J. Tani and M. Ito, "Self-Organization of Behavioral Primitives as Multiple Attractor Dynamics: A Robot Experiment," *IEEE Trans. Systems, Man, and Cybernetics, Part A*, vol. 33, no. 4, pp. 481–488.
3. M. Jordan, "Attractor Dynamics and Parallelism in a Connectionist Sequential Machine," *Proc. 8th Ann. Conf. Cognitive Science Soc.*, Lawrence Erlbaum Associates, 1986, pp. 513–546.
4. H. Kozima et al., "A Toy-like Robot in the Playroom for Children with Developmental Disorders," *Proc. Int'l Conf. Development and Learning* (ICDL 04), 2004; http://mplab.ucsd.edu/~icdl/rev-proceedings/ICDL-Book/ICDL-fin2.pdf.
5. V.S. Ramachandran and E.M. Hubbard, "Hearing Colors, Tasting Shapes," *Scientific American*, vol. 288, no. 5, 2003, pp. 53–59.

Rhythm in Human-Robot Social Interaction

Marek P. Michalowski,
Carnegie Mellon University
Hideki Kozima, *National Institute of Information and Communications Technology, Japan*

The disposition to engage in music and dance appears throughout human cultures. The evolutionary processes and cognitive mechanisms behind such behavior are under debate. However, music and dance are largely social activities—whether for aesthetic pleasure, entertainment, communication, or ceremony—and are likely linked to our innate sociality.

Rhythm is a basic element of both music and dance. Repeating temporal patterns in sound or movement characterize almost everything our bodies do—from heartbeats to walking gaits to sleep cycles. Our social behaviors, too, are infused with rhythms: speech has a tempo with varying speeds, and gestural communication often uses repetition—for example, in head and arm motions. Not only is human social behavior rhythmic, but it also often reveals a synchrony between the interactors' sounds and movements.[1] Sharing a common rhythm lets us anticipate the future and thereby coordinate our activities as we perform joint tasks.

The synchrony in social behavior is often compared to a dance that regulates our interactions with each other. Although the rhythms of everyday interaction can be quite complex, dance offers a formal context of constraints and opportunities that are useful in developing rhythmically intelligent robotic technologies. Dance doesn't require verbal communication. It features mostly regular, structured bodily rhythms externally reinforced by sound. It involves visual, auditory, and tactile modalities, and it's familiar and enjoyable to people of all ages. Furthermore, dance is popularly used in pedagogical and therapeutic scenarios with a range of disorders from autism to schizophrenia, suggesting potential applications for rhythmic intelligence in assistive technologies.

Keepon dancing

The design of robots that can interact with humans through natural social behavior often focuses on verbal communication or emotional facial expressions. However, if we neglect to endow our robots with the ability to perceive subtle human rhythms and to move dynamically in corresponding temporal patterns, they will never get beyond the stilted, rigid interaction that people currently expect from machines.

We consider rhythmic tempo to be an abstract feature of multimodal sensory stimuli. We've been developing technologies for the perception, generation, and synchrony of rhythmic behaviors that enable robots to dance with children. Figure 9 shows Keepon, a small robot we developed to study social development.[2] Its minimal appearance is designed to promote comfort and understanding in children, particularly those with developmental disorders such as autism. Keepon's eyes are cameras and its nose is a microphone. Four motors in its base give the robot four degrees of freedom in its motion (see figure 9).

We've been using Keepon in our BeatBots project (http://beatbots.org), which aims to develop simple robots capable of interactive engagement through entrainment to social and environmental rhythms. One requirement for the robot is to perceive rhythms in a number of different sensory modalities. For example, it uses signal-processing techniques to detect a musical tempo, clapping, or drumbeats. Computer-vision techniques, accelerometers, and pressure sensors enable the perception of repetitive movements by people's heads, arms, or bodies.

The robot must also be able to move rhythmically and compellingly. Each of Keepon's four degrees of freedom has several parameters (such as velocity and range of movement) that determine how it moves. Random combinations of these parameters result in different dance styles that we can change to maintain interest or to match other changes in sensory input. These degrees of freedom move quickly, in a lifelike manner with oscillatory commands, resulting in rhythmic movement that can match the frequency of a wide range of musical beats and dance behaviors we would expect to see from children.

Finally, we can smoothly adjust the frequency of the robot's movement to match the particular rhythms it perceives in the environment. This lets us study how the robot's synchrony affects various qualities of playful interactions.

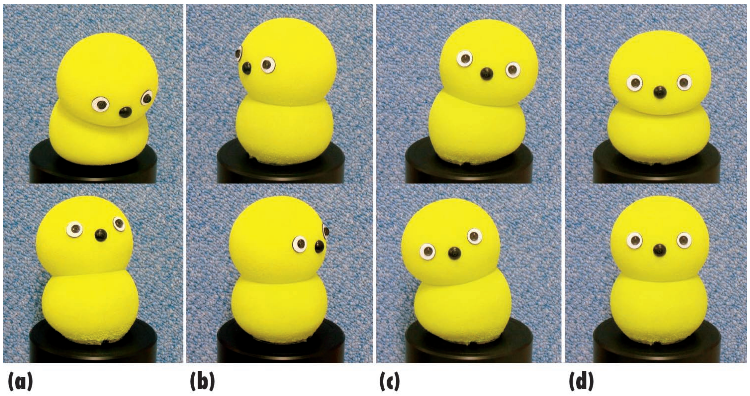


Figure 9. The Keepon robot. Its eyes are cameras and its nose is a microphone; four motors in the base let Keepon (a) nod its head up and down, (b) turn its body right and left, (c) rock its body side to side, and (d) compress and extend its body vertically in a bobbing motion.

Studies and results

In an initial study, we configured Keepon to synchronize its rhythmic movement to visually perceived movement using optical flow.[3] The robot interacted with over 200 children in an open-house event. Under these circumstances, because of imperfect perception (and variance in the children's movements), the robot's movement was sometimes synchronized to a child's movement (or, indirectly, to music playing in the environment) and sometimes not.

The dancing robot stimulated a wide range of play modes, but we were particularly interested in observing the effects of the robot's musical synchrony on children's interactive behavior. If children encountered the robot when it happened to be dancing in sync with the music, would they be more likely to dance themselves? Our analysis suggested that children could, in fact, perceive the robot's synchrony to environmental stimuli and that this heightened its attractiveness to them and elicited their physical playful interaction with the robot.

We're developing more controlled studies to examine differences between the robot taking an active or passive role in leading or following the dance rhythms. For example, we can synchronize the robot's dancing either to the movements of an accelerometer-instrumented toy (see figure 10) or to prepared children's songs. By controlling for other variables, we try to identify the condition in which the child's movements are most closely synchronized to the music. On one hand, the robot might reinforce the music's rhythm and lead the child to dance with it; on the other hand, a child's recognition of his or her influence over the robot's dance might encourage more explicit demonstration of "correct dancing" to the robot.

We've found that the robot's novelty and the variance between children's comfort and attitudes make controlling the scenario difficult in a laboratory setting.[4] We believe that the holistic nature of social interaction requires us to let each child's interaction unfold naturally, with appropriate scaffolding and facilitation from caregivers. We're planning field studies in a naturalistic playroom environment with the robot operating under different control configurations—for example, synchronizing to different rhythms or intentionally behaving asynchronously—to identify general differences between the interactions that take place under different conditions. We also intend to explore the application of this technology to the types of rhythmic play used in therapeutic practice.

Figure 10. Keepon dancing with a child. The soft paddle includes an accelerometer that transmits rhythmic-movement information to the robot to enable synchronized dancing.

A rhythmic-intelligence framework

Our long-term goal is to develop a general framework for rhythmic intelligence in interactive robots. We envision the framework as a set of increasingly sophisticated layers:

- First, the robot must have a mechanism for rhythmic attention. From the flood of sensory stimuli, it must determine which

rhythms are meaningful or important to respond to—natural or social, multimodal or particularly strong in a single modality.
- Second, the robot must have a method for matching the rhythms it exhibits to the rhythms it perceives. It might need to evaluate how successfully it's following a person's rhythm, or it might need to recognize the extent to which the person is following its lead and then behave so as to achieve rhythmic entrainment.
- Third, the robot should intelligently use its rhythmic awareness to recognize (or cause) changes in the "shape" of the interaction.

From dancing, to the turn-taking in verbal communication, to the coordinated manipulation of physical objects by human-robot teams, interactions have certain rhythmic trajectories that regulate the participants' behaviors. Robots will need to either learn or be given such models. A wide range of disciplines can inform this undertaking, including AI, neuroscience, cognitive science, signal processing, music, and dance. While rhythm is already central to many areas of robotic research, such as legged locomotion and snake robots, it's been lacking in human-robot interaction. Robots such as Keepon promise to not only entertain and assist us but also improve our understanding and appreciation of the rich backdrop of living rhythms that guide us in our lifelong dance with each other.

References

1. J.K. Burgoon, L.A. Stern, and L. Dillman, *Interpersonal Adaptation: Dyadic Interaction Patterns*, Cambridge Univ. Press, 1995.
2. H. Kozima, C. Nakagawa, and Y. Yasuda, "Children-Robot Interaction: A Pilot Study in Autism Therapy," *From Action to Cognition*, C. Von Hofsten and K. Rosander, eds., Elsevier, 2007, pp. 385–400.
3. M.P. Michalowski, S. Sabanovic, and H. Kozima, "A Dancing Robot for Rhythmic Social Interaction," *Proc. 2nd Ann. Conf. Human-Robot Interaction* (HRI 07), ACM Press, 2007, pp. 89–96.
4. M.P. Michalowski and H. Kozima, "Methodological Issues in Facilitating Rhythmic Play with Robots," *Proc. 16th IEEE Int'l Symp. Robot and Human Interactive Communication* (RO-MAN 2007), IEEE Press, 2007, pp. 95–100.

Using Chaos to Trade Synchronization and Autonomy in a Dancing Robot

Jean-Julien Aucouturier, Yuta Ogai, and Takashi Ikegami, *University of Tokyo*

Music makes people want to move—either in their imaginations or actually, as in dance. Various mappings from sound to movement are easily observed in musicians and dancers but difficult to predict, and designing such dynamic behavior into an artificial system is even more difficult.

Dancing robots so far have relied mostly on preprogrammed patterns triggered either randomly or through hard-coded interaction with a human partner. None of these programming paradigms allows for a realistic compromise between local synchronization and global autonomous behavior, so the dance behavior soon becomes monotonous and can't sustain long-term interest.

> While rhythm is already central to many areas of robotic research, such as legged locomotion and snake robots, it's been lacking in human-robot interaction.

Chaotic itinerancy

Rather than preprogramming dance patterns or their alternation with one another, we propose building basic dynamics into the robot through a special type of chaos and letting the behavior emerge in a seemingly autonomous manner. Specifically, *chaotic itinerancy*[1] (CI) is a relatively common feature in high-dimensional chaotic systems. In CI, an orbit wanders through a succession of low-dimensional ordered states (or attractors), but transits from one attractor to the next by entering high-dimensional chaotic motion.

Recently, researchers have proposed CI to model many exploratory strategies in living systems, including attachment and detachment mechanisms in conscious states.[2] The work we report here is, to our knowledge, the first to apply CI to an interactive entertainment system.

Implementing CI

We generate the robot motor commands in real time by converting the output from a neural network that processes a pulse sequence corresponding to the beats of music. (Details of our implementation approach are available elsewhere.[3])

The algorithm reads audio buffers from the robot sensors at regular time intervals and processes them by successively filtering out all frequencies above 600 Hz, extracting their amplitude envelope and feeding it to a filter bank of comb-shaped filters (or resonators), each tuned to a specific tempo (from 60 to 180 beats per minute). After each processed buffer, the algorithm selects the resonator whose output has the highest energy (this gives the current tempo) and then finds the position of the latest maximum in the resonator's output buffer (this gives the position of the latest beat). If this position differed from that of the previous beat (that is, if it's a new beat), the neural network receives a new input pulse.

In the network, the biologically inspired FitzHugh-Nagumo model controls the spiking behavior of individual neurons. Each neuron is a coupled system of a fast variable u responsible for the excitation of membrane potential and a slow variable w controlling the membrane's refractory state. Neurons are connected to one another with time-delayed connections of two types (fast and slow). When a neuron overshoots ($u > 0$), it transmits a pulse to the neurons to which it's connected, after the appropriate time delay.

Researchers have studied the dynamic properties (attractor, bifurcations) of the FitzHugh-Nagumo equations intensively. (For a review, see "Fitzhugh-Nagumo Revisited."[4]) For instance, it's known that the membrane spiking behavior is well controlled by the periodicity of the input spike train $I(t)$. For large ranges of input periods, the output spike train enters an entrained periodic state. However, chaotic and aperiodic responses also occur for certain input periods. Our study exploited such chaotic behavior.

Finally, the network's output layer simply integrates the output spike trains of a selected part of the neurons. This constitutes the motor commands that are sent to the robot.

Implementation results

In this study we used the Miuro, a relatively simple vehicle-like robot manufactured by ZMP (www.zmp.co.jp/e_home.html). The robot is equipped with an IPod MP3 player interface and a set of loudspeakers. It performs dance movements that are 2D trajectories controlled with left- and right-wheel velocities. This is well below the complexity and expressivity of humanoid robots but enough to test our chaotic control architecture.

Figure 11 shows successive steps of a simulation of the robot trajectory for a given music piece. The orbit shows typical CI behavior, with locally quasiperiodic trails in attractors of various shapes, such as drifting circular or pentagonal orbits, and abrupt transitions from one attractor to the next through higher-dimensional chaos. We observed that different songs generate different types of orbits and styles of motion.

During a single "dance," the robot exhibits phases where its motion is strongly coupled to the musical rhythm (quasiperiodic orbits tend to synchronize with the tempo) and others where it's more independent. Figure 12 shows the *information circulation*, an information-theoretic measure computed between the input and output series of interspike intervals in the neural network for a given dance performance. Information circulation characterizes how much knowledge of the input contributes to the prediction of future output, compared to knowledge of only the output's previous state.[5] The system (under certain conditions and parameter ranges) exhibits seemingly spontaneous oscillations of attachment and detachment to the stimuli.

The resulting behavior is completely deterministic because it's the solution of a nonlinear dynamical system. Yet, it adapts to the music being played: the style of motion in the system's quasiperiodic states, as well as the triggering of a transition from one state to the next, finely depend on the input sequence. Moreover, with its complex transitions from attachment to detachment, our algorithm establishes a dynamic compromise between synchronization and autonomy. Our future work will attempt to characterize the emergent conditions of this interesting behavior and study its subjective impact on human observers. For video footage of the real robot, see online supplemental material at www.jj-aucouturier.info/docs/miuro.

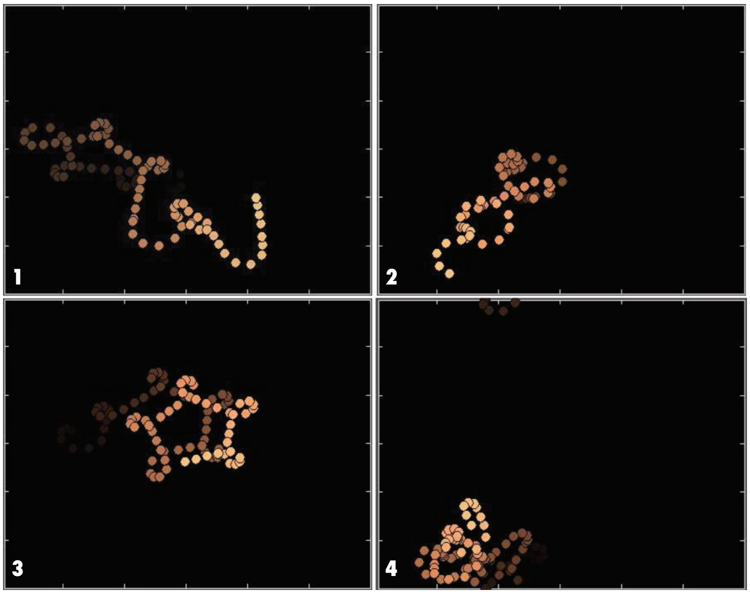


Figure 11. Simulation of the robot trajectory in the (*x*, *y*) plane at four different times in the same music piece. Each figure is an overlay of 100 successive robot time steps; successive figures correspond to different stages of the simulation (every 25 seconds). The trail itinerates through a variety of quasiperiodic patterns, each lasting a few tens of seconds, with disordered transitions from one pattern to the next—a typical chaotic-itinerancy behavior.

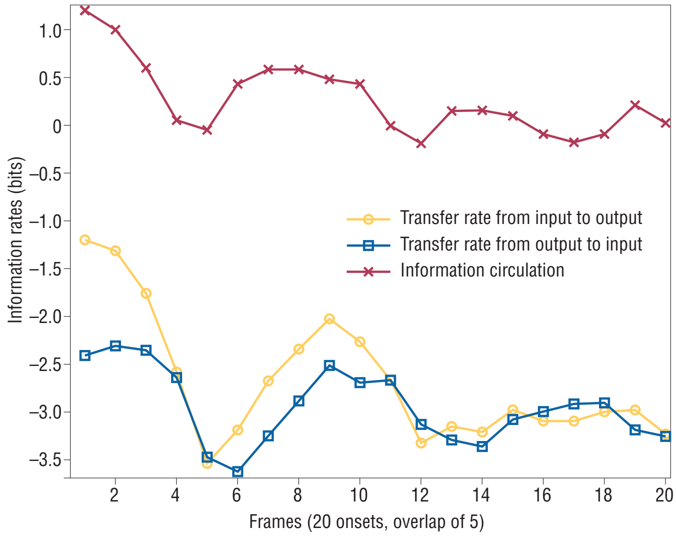


Figure 12. Information circulation and information transfer rates from the input to the output time series of interspike intervals, measured in the robot's neural network during a given dance performance. Oscillations of the information circulation reveal successive attachment and detachment phases of the robot motor behavior from its musical input.

References

1. K. Kaneko and I. Tsuda, "Chaotic Itinerancy," *Chaos*, vol. 13, no. 3, 2003, pp. 926–936.
2. T. Ikegami, "Simulating Active Perception and Mental Imagery with Embodied Chaotic Itinerancy," *J. Consciousness Studies*, vol. 14, no. 7, 2007, pp. 111–125.

Figure 13. The Dance Partner Robot. A force-torque sensor between the robot's upper and lower body measures the human leading force-moment. An omnidirectional mobile base uses special wheels to move along dance-step trajectories.

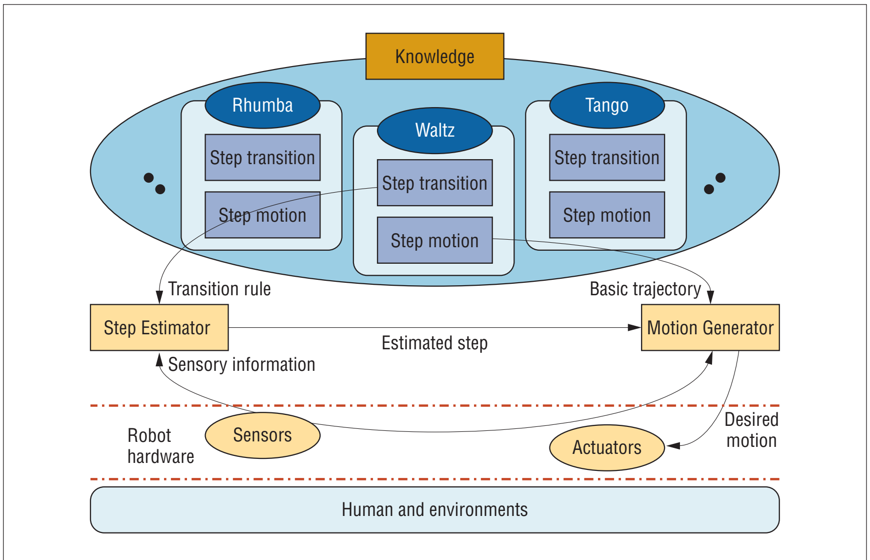


Figure 14. The dance partner robot's control architecture. A Knowledge module stores information on dancing, The Step Estimator module implements step transitions, and the Motion Generator module makes the robot move along dance-step trajectories.

3. J.-J. Aucouturier, Y. Ogai, and T. Ikegami, "Making a Robot Dance to Music Using Chaotic Itinerancy in a Network of FitzHugh-Nagumo Neurons," *Proc. 14th Int'l Conf. Neural Information Processing* (ICONIP 07), Springer, 2007; www.jj-aucouturier.info/papers/ICONIP-2007.pdf.
4. T. Kostova, R. Ravindran, and M. Schonbek, "Fitzhugh-Nagumo Revisited: Types of Bifurcations, Periodical Forcing and Stability Regions by a Lyapunov Functional," *Int'l J. Bifurcation and Chaos*, vol. 14, no. 4, 2004, pp. 913–925.
5. M. Palus et al., "Synchronization as Adjustment of Information Rates: Detection from Bivariate Time Series," *Physical Rev. E*, vol. 63, no. 4, 2001.

Dancing with a Robot: The Human Leads

Kazuhiro Kosuge, Takahiro Takeda, and Yasuhisa Hirata, *Tohoku University*

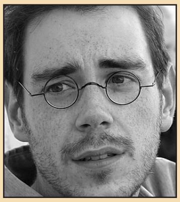
Robots are expected to execute various tasks in many human environments, but they can't easily perform tasks by themselves when the environment contains many kinds of uncertainty. Human-robot cooperation solves this problem by letting the robot execute tasks according to the commands of humans who recognize and manage the environmental complexities. In most human-robot coordination systems, the robots move passively according to the force-moment applied by a human.[1]

These systems are effective for simple tasks such as object handling. However, if robots could behave not only passively but also actively according to human intentions, environmental information, task knowledge, and so on, they could realize more effective coordinated tasks than ones done by the conventional passively controlled robots. To study such active human-robot coordination, we've focused on a robot that dances with a human.

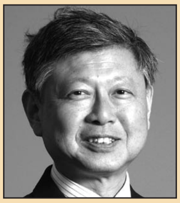
The Dance Partner Robot

Each type of ballroom dance (for example, the waltz, tango, and rhumba) consists of steps—a sequence of separated motion patterns. In ballroom dances, a male dancer leads his female partner and decides the next step according to transitional relationships between steps, information about the environment, stylistic mannerisms, and so on. The female partner estimates the next step her partner intends on the basis of his lead and executes the step in active coordination with him.

Figure 13 shows our Dance Partner Robot, which acts as a female partner and realizes ballroom dances in coordination with a male human dancer by estimating his intention. The robot has a force-torque sensor, installed between its upper body and lower body, to measure human leading force-moment. An omnidirectional mobile base uses special wheels to move along dance-step trajectories. Moreover, the robot also has two arms, each with four degrees of freedom; a neck joint with one



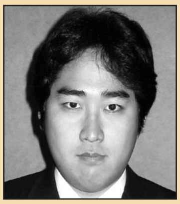
Jean-Julien Aucouturier is a postdoctoral fellow of the Japanese Society for the Promotion of Science in the University of Tokyo's Department of System Studies, Graduate School of Arts and Sciences. Contact him at aucouturier@gmail.com.



Katsushi Ikeuchi is a professor at the University of Tokyo's Graduate School of Interdisciplinary Studies. Contact him at ki@cvl.iis.u-tokyo.ac.jp.

Hirohisa Hirukawa is a deputy director of the Intelligent Systems Research Institute at the National Institute of Advanced Industrial Science and Technology in Tsukuba, Japan. Contact him at hiro.hirukawa@aist.go.jp.

Shin'ichiro Nakaoka is a researcher in the Humanoid Research Group at the National Institute of Advanced Industrial Science and Technology's Intelligent Systems Research Institute in Japan. Contact him at nakaoka@cvl.iis.u-tokyo.ac.jp.

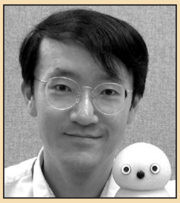


Takaaki Shiratori is a postdoctoral fellow at the University of Tokyo. Contact him at siratori@cvl.iis.u-tokyo.ac.jp or siratori@cs.cmu.edu.

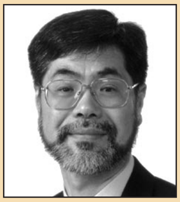
Shunsuke Kudoh is a project research associate at the University of Tokyo's Institute of Industrial Science. Contact him at kudoh@cvl.iis.u-tokyo.ac.jp.

Fumio Kanehiro is a member of the Intelligent Systems Research Institute at the National Institute of Advanced Industrial Science and Technology in Tsukuba, Japan. Contact him at f.kanehiro@aist.go.jp.

Tetsuya Ogata is an associate professor in the Department of Intelligence Science and Technology at Kyoto University's Graduate School of Informatics. Contact him at ogata@i.kyoto-u.ac.jp.



Hideki Kozima is a senior research scientist at the National Institute of Information and Communications Technology, in Kyoto. Contact him at xkozima@nict.go.jp.



Hiroshi G. Okuno is a professor in the Department of Intelligence Science and Technology at Kyoto University's Graduate School of Informatics. Contact him at okuno@i.kyoto-u.ac.jp.

Marek P. Michalowski is a PhD student at Carnegie Mellon University's Robotics Institute. Contact him at michalowski@cmu.edu.

Yuta Ogai is a PhD student in the University of Tokyo's Department of System Studies, Graduate School of Arts and Sciences. Contact him at yuta@sacral.c.u-tokyo.ac.jp.

Takashi Ikegami is an associate professor in the University of Tokyo's Department of System Studies, Graduate School of Arts and Sciences. Contact him at ikeg@sacral.c.u-tokyo.ac.jp.

Kazuhiro Kosuge is a professor in the Department of Bioengineering and Robotics, Tohoku University. Contact him at kosuge@irs.mech.tohoku.ac.jp.

Takahiro Takeda is a PhD student in the Department of Bioengineering and Robotics, Tohoku University. Contact him at taketaka@irs.mech.tohoku.ac.jp

Yasuhisa Hirata is an associate professor in the Department of Bioengineering and Robotics, Tohoku University. Contact him at hirata@irs.mech.tohoku.ac.jp.

DoF; and a waist actuated by a parallel-link mechanism with three DoFs for realizing upper-body motions such as rising and falling, swaying side to side, and leaning backward.

Control system architecture

Step transitions—that is, when and how the human partner changes steps in a dance sequence—are important for the robot to generate its own dancing motion. We therefore base our robot's control architecture on such transitions, using three modules (see figure 14).

First, the *Knowledge* module stores information on ballroom dances. It consists of two submodules: *Step Motion* stores the dance-step trajectories, and *Step Transition* stores the transitional rules.

Second, the *Step Estimator* estimates the next dance step intended by the human according to his lead at the transition. This module uses hidden Markov models (HMMs) to estimate the step.[2] For each step transition, each HMM stochastically models the time series of the leading force-moment. We use a left-to-right-type HMM with states that have continuous probability density functions. The HMM is designed to form a close relationship between states and time. The force-moment features at each time step are inputted to the probability-density functions, the parameters of which are trained beforehand on the basis of the Baum-Welch algorithm using force-moment training samples. So, the likelihood of the step estimation calculated on the basis of HMM will be high if the human applies a force-moment feature series similar to that of the training samples. The Step Estimator outputs the step that maximizes the likelihood at the transition.

Third, the *Motion Generator* generates cooperative dancing motion based on the estimated step and physical interaction with the human.[3] We consider cooperative dancing to require at least two motion elements. One motion follows the dance-step trajectory actively; the other coordinates the robot's motion adaptively with the human. Both elements mutually influence each other, enabling the robot to fit its own dance-step stride to the human partner's on the basis of physical interactions within the dance-step repetitions.

Exhibitions and possibilities

Using this control architecture, the Dance Partner Robot has waltzed with a human partner in robot exhibitions, including the Aichi Expo 2005 in Japan and the *Wired* NextFest 2006 in New York City. A video of the Aichi Expo dance show is available at www.irs.mech.tohoku.ac.jp/PBDR/movie/PBDR.mpg.

The Dance Partner Robot certainly has entertainment potential. However, our goal is also to apply the technologies involved in estimating human intention and generating motions based on this intention to improve various assistive apparatus. For example, we're developing an intelligent walker for elderly people that helps prevent falling accidents by controlling servo breaks according to estimations of user state.[4] Such technology can improve the quality of life for the elderly and other people needing similar support. ■

References

1. H. Kazerooni, "Human Robot Interaction via the Transfer of Power and Information Signals," *Proc. 1989 IEEE Int'l Conf. Robotics and Automation*, IEEE Press, 1989, pp. 1632–1647.
2. T. Takeda, Y. Hirata, and K. Kosuge, "Dance Step Estimation Method Based on HMM for Dance Partner Robot," *IEEE Trans. Industrial Electronics*, vol. 54, no. 2, 2007, pp. 699–706.
3. T. Takeda, Y. Hirata, and K. Kosuge, "Dance Partner Robot Cooperative Motion Generation with Adjustable Length of Dance Step Stride Based on Physical Interaction," *Proc. 2007 IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, IEEE Press, 2007, pp. 3258–3263.
4. Y. Hirata, A. Muraki, and K. Kosuge, "Motion Control of Intelligent Passive-Type Walker for Fall-Prevention Function Based on Estimation of User State," *Proc. 2006 IEEE Int'l Conf. Robotics and Automation*, IEEE Press, 2006, pp. 3498–3503.