Proceedings of the 2003 IEEE
International Conference on Robotics & Automation
Taipei, Taiwan, September 14-19, 2003

Generating Whole Body Motions for a Biped Humanoid Robot from Captured Human Dances

Shinichiro Nakaoka Atsushi Nakazawa Kazuhito Yokoi†
Hirohisa Hirukawa † Katsushi Ikeuchi
Institute of Industrial Science, University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan
nakaoka@cvl.iis.u-tokyo.ac.jp
†Intelligent Systems Institute, National Institute of Advanced Industrial Science and Technology
Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan

Abstract— **The goal of this study is a system for a robot to imitate human dances. This paper describes the process to generate whole body motions which can be performed by an actual biped humanoid robot. Human dance motions are acquired through a motion capturing system. We then extract symbolic representation which is made up of primitive motions: essential postures in arm motions and step primitives in leg motions. A joint angle sequence of the robot is generated according to these primitive motions. Then joint angles are modified to satisfy mechanical constraints of the robot. For balance control, the waist trajectory is moved to acquire dynamics consistency based on desired ZMP. The generated motion is tested on OpenHRP dynamics simulator. In our test, the Japanese folk dance, 'Jongara-bushi' was successfully performed by HRP-1S.**

I. Introduction

Traditional dances are considered as intangible cultural assets. However, some of them are disappearing because of a lack of successors. We are attempting to preserve these dances through computer and robotic technology.

The simplest way to preserve the traditional dances is to record their motions through a motion capturing system. However, the recorded data is insufficient for preservation because we cannot watch the actual dances again. Furthermore, it is difficult to master the dances only from the recordings.

This has motivated us to develop a robot system to imitate human dances. Pollard et al.[8] have realized a dance by a humanoid based on captured human motion . But the motion is limited to the upper body and the waist is fixed on a stand. In our study, a robot performs dances including leg motions and is able to maintain its balance independently.

A. Imitation and Behavior Models

It is desirable that humanoid robots can master many kinds of tasks without a burden of complicated programming by a human. This requires the ability to imitate human behaviors just as humans learn from one another. Many researchers have defined an abstract model to recognize and reproduce human behavior. For example, Inamura et al.[2] proposed a general framework for imitating whole body motions.

We hypothesized that a specific model to dance motions is required and the model has two-level structure: *motion primitives* and *styles*. The motion primitives are the high level structure which constructs the motion overview like a musical score. They represent the intentions of the dancer in some sense. The styles express skill or characteristics of motion details. We proposed a method to extract the motion primitives by analyzing trajectories of limbs [5]. Representation based on our model enables various applications such as an adaptive performance to stage condition or creating new sequences of dance action. Also, the model in itself can contribute to dance preservation because extracted information on the model helps people learn the dances.

In this paper, we define *primitive steps* as motion primitives to generate feasible leg motions of the actual robot. The motion primitives of arms are used as *essential postures* to express the characteristics of the original dances.

B. Balance Control during Complex Motions

Recently the development of biped humanoid robots has been active and their walking ability has been advanced [1]. Kajita et al.[3] proposed the generation method of stable walking patterns.

In addition to simple walking, these robots must have the potential to perform more complex motions with the whole body. However, a complex motion such as dances is difficult to maintain balance.

To realize these motions, one reasonable approach is to create an initial motion and to transform it to keep balance [9][10]. This process should preserve the characteristics of the original motion particularly for dances. For this, the important factors of the characteristics in the dance motions must be clarified.

II. Overview of the Proposed System

A robot motion is generated from human dance performance through a series of steps. Figure1 depicts the

0-7803-7736-2/03/$17.00 ©2003 IEEE

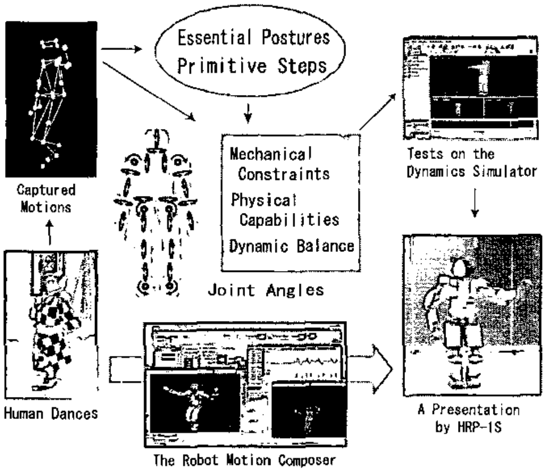


Fig. 1. Overview of the System

overview of our proposed system. We currently focus on Japanese folk dances, 'Jongara-bushi' (Fig.2).

Dance motions are acquired by an optical type motion capturing system made up of eight cameras. A dancer has 30 markers on his or her joints, and marker positions are recorded at the rate of 200 frames per second. Then marker positions are analyzed to extract primitive motions: the essential postures and the step primitives.

The robot is moved according to the joint angle trajectories. The trajectories are generated from marker positions and motion primitives. The generating process for arm and leg motions are different. Arm joint angles are mainly calculated by inverse kinematics of markers and leg joint angles are mainly generated as patterns from step primitives.

The generated angles are not under the constraints of the robot and they tend to have inconsistent balance for dynamic forces. Therefore the angle trajectories must be modified to solve such problems. Information of essential postures is used to express dance characteristics.

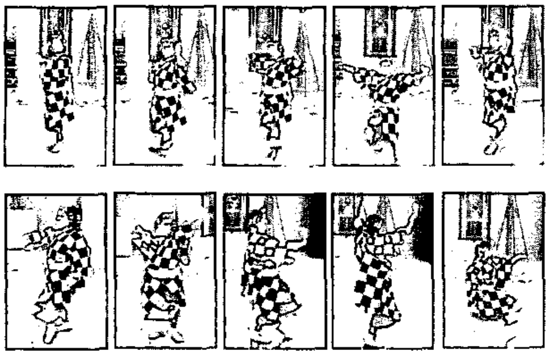


Fig. 2. Jongara-Bushi

The motion is tested on OpenHRP dynamics simulator [4] to verify its validity. Last of all, the motion is carried out on a real humanoid robot HRP-1S [11].

We have developed the Robot Motion Composer, which can automate the generating process. Users can interactively change the motion parameters and create new motion sequence which can be verified by animation. Information of a target robot can be given in the same format as the OpenHRP model file.

III. Generating Joint Angle Trajectories

A. Constraints of the Robot

Humanoid robots are supposed to have the body structure similar to that of human. For example, HRP-1S has 7-DOF arms, 6-DOF legs and a 1-DOF neck; this seems sufficient for human-like movement. However, the structure has considerable constraints for dance motions, because many dances include motions such as torso twists, swings of arms in wide arcs, etc. For some parts of motions, adequate DOF may not be available. Also the robot structure has singular points. For example, the shoulder joint is constructed of a three rotation sequence of pitch, yaw and roll. When the robot raises its arm horizontally, the DOF of the shoulder joint decreases and the robot cannot freely change the direction of the arm from that position.

One approach to avoid the problems of the mechanical structure is to increase the number of the joints. But it would be not easy to attach the new joint mechanism and actuator because of the lack of space within the body. Another approach is to develop a new mechanism. Okada et al. proposed the Cybernetic Shoulder which has no singular point on simple mechanism [7].

The capacity of actuators is another constraint. Dance motions often require quicker motions than walking or everyday tasks. Although HRP-1S has the most powerful actuators currently available, it is not sufficient for most dances. More powerful actuators or lighter weight body are necessary.

The hardware development to eliminate these constraints is important. However, we are now concentrating on realizing a seemingly good motion by using the current robots.

B. Arm Joint Angles

In arms, the initial value of the joint angle trajectories are calculated directly from position correlations with related joint markers. Angles and angular velocities of the initial value must be limited within possible ranges of the robot. The initial motion may imply posture which is in the neighborhood of singular points. At the posture

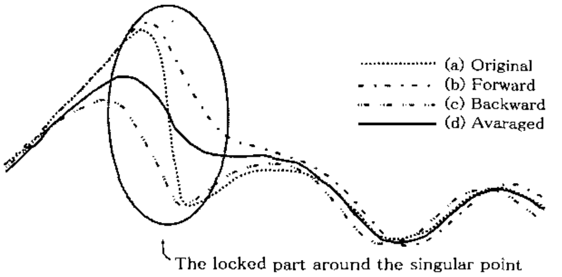


Fig. 3. The limiter of the joint angular velocity and a trajectory around singular point

near singular point, the valid moving patterns are limited and a movement may be locked. This problem can be considered as a problem of angular velocity because a non-continuous curve of velocity around the locked frame can be considered to be a high gradient curve on the discrete system.

Pollard et al. [8] proposed a method to limit angular velocities. In their method, the new angle trajectory is created by following the original trajectory under the velocity limits. This process is formulated as follows.

$$\dot{\theta}_i = \theta_i - \theta_{i-1}, \tag{1}$$

$$\ddot{\theta}'_{i+1} = 2\sqrt{K_s}(\dot{\theta}_i - \dot{\theta}'_i) + K_s(\theta_i - \theta'_i), \tag{2}$$

$$\dot{\theta}'_{i+1} = max(\dot{\theta}_L, min(\dot{\theta}_U, \dot{\theta}'_i + \ddot{\theta}'_{i+1})) \tag{3}$$

$$\theta'_{i+1} = \theta'_i + \dot{\theta}'_{i+1} \tag{4}$$

where θ_i is the original joint angle, θ'_i is new joint angle, i is a joint number, $\dot{\theta}_L$ and $\dot{\theta}_U$ are the lower and upper velocity limits.

This process generates a similar trajectory to the original one within the limit. Consequently, it must delay from the original one all through the motion (Fig.3-b). Then another trajectory is created by the inverse process of the above equations from the end to the start point, a result becomes the trajectory of the future instance compared to the original one (Fig.3-c). Finally, both trajectories are averaged to get a trajectory whose shape is overlapped with the original one. (Fig.3-d). This trajectory preserves the characteristics of the original one well. The locked parts around singular points in the initial motion is also changed into a smooth curve.

C. Clarifying the Essential Postures

We analyzed the trajectory of the hands and the arm motion is segmented at the frames in which the speed of hands is approximately zero [5]. At these frames, the dancer makes particular postures. These postures are the important, essential ones, because the set of these postures is the unique representation of a particular dance. Hence, the dancer pauses to show it clearly.

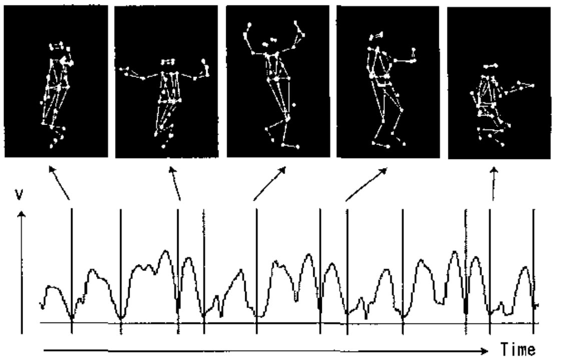


Fig. 4. Motion segments and stopping postures. The graph shows a speed of a hand movement. Vertical lines show segment boundaries.

However, in the modified trajectories, the essential postures becomes ambiguous, because the adapting process separately changes the stopping time of each joint. As a result, the motion becomes obscured and it does not resemble to the original one. For a better representation, it is important to clearly stop the motions at the essential postures.

Figure 4 shows the joint angle trajectories of the original and the modified motion. To clarify the essential postures, the extreme points (where velocity is zero) of all the joint should be arranged at the segment boundary within the possible range (Fig.5). This process is as follows.

1) In each segment boundary, find the nearest extreme point.
2) For the point far from the boundary,
    - Slide the point horizontally to the boundary, if new gradient is under the velocity limit.
    - Otherwise, slide the point vertically to the level which is under the limit.
    - connect the trajectory to the new extreme point by stretching the original trajectory.

D. Leg Pattern Generation

The generating process of leg motions is different from that of the arm motions. First, symbols of the motion primitives are extracted from the original motion. Then all leg motions are created from patterns associated with the primitives.

One of the reasons for using primitives is the limitation of leg movement in robots. In the actual robots, the constraints of the leg motions in dances is more stricter than that of the arm motion. Humans often use toe support, which is incapable of our robot. The robot sole must be completely flat against the ground when it contacts. Support legs cannot expand sufficiently without toe support. Because of the articulatio coxa constraint, the robot cannot cross its legs adequately. These constraints

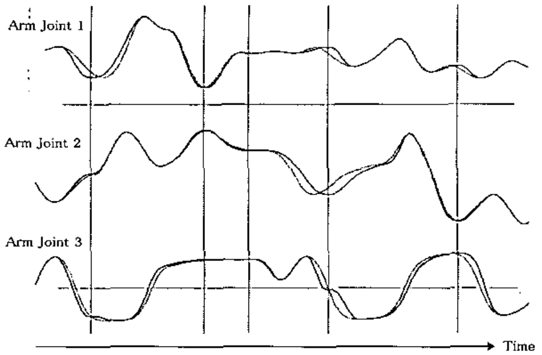
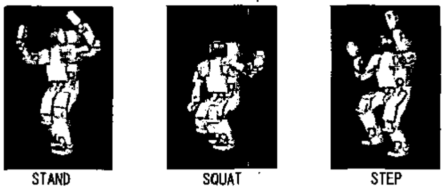


Fig. 5. Arrangement of the extreme points near by the segment boundaries (vertical lines). Dotted lines are the original trajectories and solid lines are arranged ones.



	Parameters of the primitive
STAND	A previous state is maintained
SQUAT	•The time and the waist height at the deepest posture •The period of the action
STEP	•Which is the swing foot ? •The time at the highest foot posture •The highest position and attitude of the swing foot •The position and the attitude of the swing foot when it lands on the ground. (The positions and the attitudes is relative to the waist coordinate.)

Fig. 6. Motion primitives of the legs

make it unsuitable to generate the leg motions directly from the captured ones.

Therefore, in order to satisfy the limitation of leg motions, a feasible motion is created from the motion primitives instead of the raw motion information. This approach generates proper motions according to the capacity of a robot. In addition, this makes it possible to perform the dance interactively, or adaptively in various stage conditions. Essential of this approach is to achieve a performance with intentions.

In Jongara-bushi, the movement of the legs is segmented into actions and the actions are classified into three primitive symbols; STAND, SQUAT and STEP. Figure 6 shows these primitives. Primitive symbols are extracted by analyzing sole and waist positions. The extraction process is as follows.

From the height of toe and heel markers, we can recognize whether the foot is supporting the body on the ground (*support state*) or floating for the next step (*swing state*) . The motion is segmented at which the foot state is changed and each segment is classified. In each segment, if the body is supported by one leg, primitive is STEP. If the position of the waist is low in the segment, primitive is SQUAT. Otherwise, it is STAND. Primitive parameters are determined from the position and the attitude of the foot at the segment boundary, the highest position of a swing leg, the lowest waist position, etc.

A leg motion is initially created as a trajectory of foot position and a transition of foot attitude. The position and attitude are based on the waist coordinate. In the STEP primitive, the trajectory is created by interpolating through the initial, peak and the final state. The sole attitude is constrained to lie flat against the ground when the foot is on the ground. In the SQUAT primitive, the trajectory is created by interpolating the height of the waist position. Then joint angles of each frame are calculated with inverse kinematics of the foot trajectory.

IV. Balance Control

The robot has the capacity to take postures along the motion generated through the above process. However, the robot is unable to keep its balance when it performs the motion standing by itself on the ground. Because the generated motion does not always satisfy the dynamic consistency in interaction with the ground. This section describes the method of motion modification for the balance control.

A. Dynamic Balance and Zero Moment Point

Leg motion is generated under the assume that all the area of a foot sole contacts with the floor when it is supporting the body. In other words, a sole does not rotate during that time. In terms of dynamics, this assume is satisfied when the point at which the moment to the robot body is zero exists in the area of the sole surface. In this time, the sole does not rotate. The point is called 'zero moment point (ZMP)' and the area is called *supporting area.* If a robot is supported by both feet, supporting area corresponds to the convex area which consists of both soles.

Given the physical model of a robot, a trajectory of ZMP can be calculated from motion data for the robot, under the assume that the supporting area is infinite. If ZMP moves out of an actual supporting area, the motion is impossible to perform because the actual motion must imply rotation of the supporting sole at that time, so that the sole moves away from the ground and the robot falls down.

Therefore the motion must be modified to keep ZMP in the supporting area. It is necessary to create a modified ZMP trajectory which is always in the supporting area and acquire knowledge of how to modify the original motion to realize the desired ZMP.

ZMP is calculated from the motion of all the robot body segments. Following equation shows the calculation on x-axis. (The calculation is separately performed on each axis.)

$$x_{zmp} = \frac{\sum m_i z_i \ddot{x}_i - \sum\{m_i(\ddot{z}_i + g)x_i + (0,1,0)^T \mathbf{I}_i \dot{\omega}_i\}}{-\sum m_i(\ddot{z}_i + g)}$$

where x_{zmp} is the position of ZMP on x axis, x_i, z_i are the position of each segment, m_i is the mass, I_i is the inertia tensor ω_i is the vector of angular velocity and g is gravitational constant.

Under this equation, we need a modification of segment motions from the modification of ZMP. But this problem is difficult just as it is. Nishiwaki et al. [6] proposed a method to solve this problem. On discrete system, supposing all the segments are restricted to be translated horizontally in the same distance, the following equation is acquired.

$$x^e_{zmp}(t_i) = \frac{-hx^e(t_{i+1}) + (2h + g\Delta t^2)x^e(t_i) - hx^e(t_{i-1})}{g\Delta t^2}$$

where x^e_{zmp} is a difference between the original ZMP and desired ZMP, x^e is a difference between the position of original segment and the modified position, t_i is the time at frame i, h is the height of the center of mass, Δt is time per one frame. x^e is calculated from x^e_{zmp} as tridiagonal simultaneous liner equations.

In practice, the proposed translation can be approximated to an upper body translation. This method is easily converge in iteration. Although a modification is limited to horizontal translation of upper body, this method is effectively control the balance.

B. Creating a ZMP Trajectory

It is fundamental that the desired ZMP is inside the supporting area. If a supporting area remains on one state, ZMP should remain a stable point in the area such as a center of the area or just below the ankle joint. In practice, supporting state changes with steps. A stability of the motion depends on a ZMP trajectory with state transitions. For a stable transition, the ZMP trajectory should be as smooth as possible. In this study, we applied the following criteria.

- In STEP period, ZMP must locate at center of a supporting sole.
- In STAND period, ZMP moves from a previous position to the next supporting position by third order polynomial equation. Initial velocity and accelerations and final ones are kept zero.
- If period of STAND is long, transition is separated into three steps: (1) from a previous position to center of supporting area, (2) stay there and (3) move to the next supporting position.
- If period of STAND state is short, ZMP movement speeds up and robot motion becomes unstable. Adequate transition time is required for stable motion. In this case, ZMP movements is expanded so that it starts in the previous state and extends into the next state. Acceleration and deceleration of ZMP is done in those states.

From above method, we generate a desired ZMP trajectory and modify the trajectory of the upper body position. Figure 7 shows a sequence of support state and ZMP trajectory.

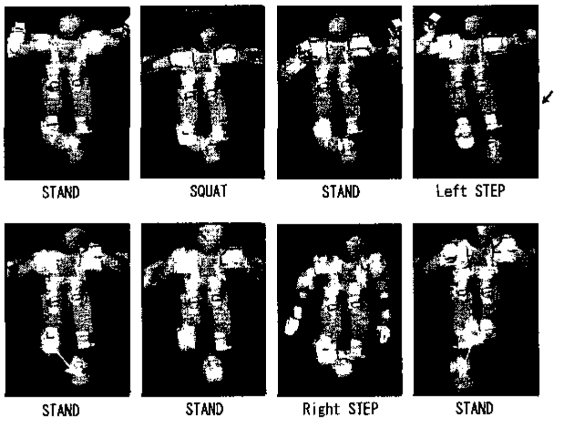


Fig. 7. A motion with support state transition. A markers on the feet shows ZMP.

C. Dynamic Stability of Arm Motions

Pollard method in Section 3 can also control angular acceleration by constant Ks in equation (2). The higher Ks is, the bigger acceleration average becomes. The response to the original motion is better with higher Ks. On the contrary, when Ks is smaller, response is worse but dynamics stability is better. When arms move in wide arc with high acceleration, the robot will be unstable. If the leg motion is in unstable state at that time, balance control is difficult. In this case, Ks around such arm actions should be small to restrict ZMP movement. On the contrary, when a leg action is in stable state, Ks can be increased for better imitation.

V. Experiments

We tested the generated motions on the OpenHRP platform. On this platform, both a real and a virtual robot can be controlled by the same software. Since we currently focus on generating the feasible motion patterns in advance, we used the default controller which requires a sequence of joint angles and desired ZMP, and just plays that motion.

On simulation, the virtual robot can perform the motion keeping balance along the entire sequence. However, a support foot slides and the motion becomes unstable when the robot widely rotates the waist to turn. Balance modification does not consider horizontal (yaw) element of moment, which is not concerned with ZMP. Balance modification should be improved to solve this behavior.

Then we tested the motion on the real robot HRP-1S. As the first step, we prepared the motion in which leg action is restricted within squats. In an initial experiment, arms could not follow the given motion and became unstable. We had to reduce the speed of the motion as 2.5 times slow throughout the whole sequence. Then the robot could perform the whole motion standing on its legs in stable (Fig. 8).

VI. Conclusion

This paper described the system for robots to perform dance motions acquired from human dancers. Our system could generate the robot motions which satisfy the mechanical constraints and dynamics consistency. We realized the dance performance by the humanoid robot standing on its legs. It has been confirmed that the step primitives are valid for generating feasible leg motions from complex motions such as dances. However, we should improve balance modification to consider the yaw rotation so that more stable motion is acquired. Then we are going to test the motion including step actions on the actual robot.

VII. References

[1] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. *Proceedings of International Conference on Robotics and Automation*, May 1998.

[2] Tetsunari Inamura, Iwaki Toshima, and Yoshihiko Nakamura. Acquisition and embodiment of motion elements in closed mimesis loop. *Proceedings of International Conference on Robotics and Automation*, 2002.

[3] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3d linear inverted pendulum mode: A simple modeling biped walking pattern generation. *Proceedings of International Conference on Intelligent Robots and Systems*, 2001.

[4] Fumio KANEHIRO, Kiyoshi FUJIWARA, Shuuji KAJITA, Kazuhito YOKOI, Kenji KANEKO, Hirohisa HIRUKAWA, Yoshihiko NAKAMURA, and Katsu YAMANE. Open architecture humanoid robotics platform. *Proceedings of International Conference on Robotics and Automation*, 2002.

[5] Atsushi Nakazawa, Shinichiro Nakaoka, and Katsushi Ikeuchi. Imitating human dance motions through motion structure analysis. *Proceedings of International Conference on Intelligent Robots and Systems*, 2002.

[6] Koichi NISHIWAKI, Satoshi KAGAMI, Yasuo Kuniyoshi, Masayuki INABA, and Hirochika INOUE. Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp. *Proceedings of International Conference on Intelligent Robots and Systems*, 2002.

[7] Masafumi OKADA, Yoshihiko NAKAMURA, and Shin ichiro HOSHINO.

[8] Nancy S. Pollard, Jessica K.Hodgins, Marcia J. Riley, and Christopher G. Atkeson. Adapting human motion for the control of a humanoid robot. *Proceedings of International Conference on Robotics and Automation*, 2002.

[9] Seyoon Tak, Oh young Song, and Hyeong-Seok Ko. Motion balance filtering. *Proceedings of EUROGRAPHICS*, 19(3), 2000.

[10] Katsu Yamane and Yoshihiko NAKAMURA. Dynamics filter - concept and implementation of on-line motion generator for human figures. *Proceedings of International Conference on Robotics and Automation*, 2000.

[11] Kazuhito Yokoi, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Shuji Kajita, and Hirohisa Hirukawa. A honda humanoid robot controlled by aist software. *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2001.

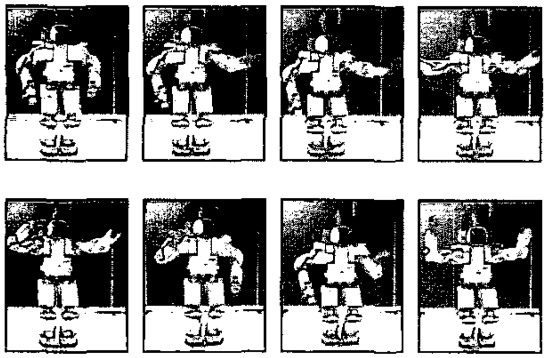


Fig. 8. A Performance of Jongara-bushi by HRP-1S