Study on Performance Motion Generation of Humanoid Robot

Kinam Lee¹ and Young-Jae Ryoo²

¹ Robot Artificial Intelligence Convergence Center, Mokpo National University, Mokpo, Korea

² Department of Electric and Control Engineering, Mokpo National University, Mokpo, Korea

Ijfis

Abstract

A study was conducted on the performance motion generation of a humanoid robot to generate robot motions and have it execute dance performances. The proposed method can directly capture motions of the humanoid robot. The captured humanoid joint angles are used as data for generating the motion of the robot through a performance generation program. This method does not require any additional equipment, has no space constraints, and can generate all the postures that the humanoid can assume. To verify the proposed method, we installed the performance generation program in a humanoid robot, which then executed a dance performance.

Keywords: Humanoid robot, Performance generation, Motion

1. Introduction

Humanoid robots attract enormous attention because of their human-like behavior. Therefore, attempts have been made to create robots that perform human-like actions, such as dancing on stage, around the world [1–3]. During the on-stage performance of a humanoid robot, it is essential to let the audience comprehend what it is conveying through each of its motions. Thus, for a performing humanoid robot, a technique for motion generation is needed [4].

A previous method for generating humanoid motions involved capturing the motion of a performing human actor and then applying the inverse kinematics of those motions to the robot [5–9]. In this motion capturing method, the main joints of the person performing the motions are marked and made recognizable, and the camera tracks the motion of these joints of the body.

Kinect, a line of motion sensing input devices produced by Microsoft, facilitates a more advanced way of capturing human motions and converting the captured images into a skeletal format. As the skeletal capturing of the human postures has kinematic differences from the actual humanoid robot, it is necessary to convert the skeletal motions to the inverse kinematics of the robot. However, this method has certain disadvantages. For example, the actor is required to dance with the camera and the sensor attached to his or her body to use Kinect. Thus, it is necessary to use additional equipment. Space constraints caused by the equipment installed is another drawback of this method. Furthermore, it is only possible to generate a posture that can be assumed by a human.

Therefore, here we propose a method to capture motions directly from a humanoid robot. As

Received: Feb. 21, 2019

Revised : Oct. 21, 2019

Accepted: Nov. 21, 2019

Correspondence to: Young-Jae Ryoo
 (yjryoo@mokpo.ac.kr)

©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

the user manipulates the posture of the robot, the angles of the joint motors in the robot posture are captured using the angle sensors built in the joint motors of the robot. The captured humanoid joint angles are used as data to generate the motion of the robot. The proposed method does not require any additional equipment, has no space constraints, and can generate all the postures that the humanoid could possibly assume. In particular, the proposed method can generate as many motions as the degrees of freedom of the robot.

2. Performance Motion Generator

Figure 1 shows an overview of the proposed humanoid performance editing method. The user manipulates the robot to assume a desired posture and captures the angle values of the joint motors of the humanoid posture with the help of built-in sensors in the joint motors of the humanoid to generate the robot motion. The captured posture is generated as a key posture and a motion is generated by arranging or rearranging several such postures. This proposed system does not require any additional equipment and has no space constraints. Furthermore, it can generate all the postures that the robot can assume.

2.1 Posture, Action, and Motion

As shown in Figure 2, there are postures, actions, and a motion involved in the motion taking place.

Figure 1. Overview of the proposed humanoid performance editing method.

Figure 2. Definition of posture, action, and motion.

A “posture” is the way in which a human or humanoid robot holds their shoulder, neck, and back. It could also be a particular position in which they stand, sit, or take a step at a particular moment. When a humanoid robot assumes a posture, its joint angles are determined. The joint angles are the key variables that define the posture. Thus, a posture is the position of a humanoid at a point in time and a collection of the joint motor angle values.

An “action” is a goal-oriented sequence of motions; for example, raising a hand up or placing a foot forward. Actions may have a clear beginning and end; however, they may also overlap owing to articulation, such as when playing a series of tones on a piano. This uncertainty regarding the segmenting of actions makes them subjective entities. As such, it is not possible to measure an action directly as there is no objective measure to decide when an action begins or ends, or how it is organized in relation to the other actions. However, based on the knowledge of human cognition, it is possible to create systems that can estimate various action features based on the measurement of motions.

A “motion” is a displacement of an object in space over time. This object could be a hand, foot, or head of a human or humanoid. The motion is an objective entity and can be recorded with a motion capture system. A motion capture system could be a simple slider (1-dimensional), a mouse (2-dimensional), a camera-based tracking system (3-dimensional), or an inertial system (6-dimensional: 3D position and 3D orientation). As motion is a continuous phenomenon, it is more meaningful to consider one or more motion sequences in the current context.

2.2 Humanoid Performance Editor

Figure 3 shows the process user generate a performance by a humanoid. The user captures the key postures of the humanoid using the proposed system, generates a set of actions by editing the parameters of the captured key postures, and then connects multiple actions to generate one motion. A performance is generated and executed by arranging and rearranging multiple generated motions.

Figure 4 shows the procedure for capturing the humanoid posture and generating a motion. Each key posture of the humanoid is captured by the system. The system acquires the angle data of each joint of the humanoid at the moment of the key posture, and the multiple captured postures are stored as one motion.

At this time, the angle data of the joint are represented by

Figure 3. Process to generate a dance performance.

Figure 4. Procedure to capture the posture of a humanoid, generate an action from the postures and a motion from the actions.

an encoder built into the joint motor and the angle data in the range of 0° to 360° are converted to a range of numbers from 0 to 4,095.

Figure 5 shows the timing chart of a motion play. Each action has a “play time” and a “hold time” for each key posture. The “play time” is used to connect the postures continuously to operate the motion. The “hold time” prevents sudden movements as the humanoid’s joint moves to the next posture. The “play time” is used to synchronize music and motion.

The system is configured to manage the overall “play time.” This allows the operation to be performed in sync with the music beat.

Multiple actions can be linked together to form a single

Figure 5. Timing chart of motion.

motion and multiple motions can be linked together to generate a single performance.

3. Experiments

3.1 Humanoid Robot, CHARLES2

The following are the specifications of the humanoid “CHARLES2.” The humanoid robot has a height of 110 cm, and its total weight, including the battery, is 8.16 kg. In addition to the 18 degrees of freedom (DoF), which are the minimum required to perform a human motion, 5 DoF are added to move its arm, head, and waist. Thus, CHARLES2 has a total of 23 DoFs.

To drive the joint actuators, we used an RS485 serial communication channel, motor driver circuits, and servo motors with the embedded encoders. The main controller is equipped with a small personal computer with Linux operating system to interface with I/O devices, such as a camera, microphone, and speakers. The sub controller is based on the ARM board technology with embedded acceleration sensors and gyro sensors, and is connected to the actuator through the RS485 serial communication channel. The main and sub controllers transmit the data from the actuators, sensors, and other devices through packet communication (Table 1).

3.2 Performance Motion Generation Program

We applied the proposed motion editing and playback tool to the developed humanoid robot and generated actions and motions to test it. The experiments involved generating and executing a performance through the arrangement and rearrangement of motions.

The humanoid motion refers to the motion data in the controller. These data can be seen and edited on the “motion editor”

Table 1. Specifications of CHARLES2

Item	Value
Height (cm)	110
Height of CoM (cm)	59.5
Feet size (cm)	11.4 × 23
Weight (kg)	8.16
DoF	
Head	2
Arm	8
Waist	1
Leg	12
Main controller	PC with Atom CPU
Sub controller	Arm 32-bit Cortex
Camera	2 M pixel HD
Power	
For motors	4,400 mAh, 14.8 V × 2 ea
For controller	2,200 mAh, 11.1 V × 2 ea

program, as shown in Figure 6(c). The “motion player” program can play the generated motions.

The functions of the motion editor are as follows:

The motion player program in the humanoid, CHARLES2, can control a total of 23 joint motors. Therefore, to generate a motion with the motion editor, we should set the numbers for the joint motors between 0 and 23.

The joint motors may be controlled by both the motion editor and the player program. Generally, the control priority is first on the motion editor and then the player program. In other words, once a motion is executed, the joint motor will be controlled by only the motion editor, and the player program will have no control over the joint motors.

The second function is to capture the postures. A posture is a humanoid program entity. Generally, it is a collection of the joint motor angle values required for the posture. Thus, the posture refers to the angle values of the humanoid’s joints. When the posture is modified, the humanoid moves accordingly.

In addition to the above, a “torque function” enables the humanoid to make a posture manually by turning the robot’s joints ON or OFF. If the torque is ON, the joint angle value is displayed. Otherwise, the value is displayed as OFF.

3.3 Posture Capture

To preserve the simple dance motions of a humanoid, the system maintains postures of dance motions. A motion editor is used

(a)

(b)

(c)

Figure 6. Screen capture of the motion editor for the humanoid robot: (a) posture editor, (b) action editor, and (c) motion editor.

Figure 7. Captured postures.

to produce the postures for the humanoid. CHARLES2 has 23 joint motors and the motion editor is used to preserve and control each joint motor angle. The joint angle represents a posture of the humanoid's dance motion. Users can easily change the joint angle with scrolling bars on the motion editor user interface screen. Additionally, they can check what type of posture is generated with those joint angles by transmitting the motor angles to the humanoid. By merely watching the computer graphical image of the humanoid on the screen, we can simulate the humanoid's postures and the dance motion.

The humanoid dance motion postures are described as 23 joint angles. These angles are saved as motion codes, which are maintained in the system in the dance motion posture database.

Figure 7 shows the process of capturing a posture and connecting it to create one action, namely action #1. First, on applying torque to all the joints of the humanoid robot, only the right arm joints of the part to be moved release the torque, the user holds the desired posture, and turns on the torque. When the torque is applied to all the joints of the humanoid again, the angle value of the current joint motor is simultaneously captured using the encoder. Here, the encoder is an angle sensor incorporated in the joint motor. At this time, if the captured total joint angles are applied to the kinematics of the humanoid, they are expressed as one posture. As shown in Figure 8, a total of four postures are captured during the lifting of the right arm of the robot, and they are linked and edited into one action.

3.4 Performance Motion Player

A robot motion is a set of the position and speed data of the joints while a humanoid robot is moving. For a humanoid to perform a dance, a motion is required. Thus, a suitable motion code should be determined for the humanoid. The motion code is identified by the motion editor program.

Figure 8. Joint angle values of the captured postures.

Figure 9. Change in joint angles according to the time of a motion composed of four actions.

While a motion code is a type of data, a replay program is a kind of software program. It is similar to a music database and a music player. If there is no music player, we cannot listen to the music because the music data cannot be played. The result is the same when there is a music player but no music data.

If we wish to make a humanoid move, a replay program is required. If the replay program embedded on the humanoid uses motions, the motion code should be generated beforehand. If no motions are used in the replay program, the motion code is not necessary.

Figure 9 shows the result of automatically generating a robot action between postures by setting the “play time” and “hold time” in each posture and inputting the joint acceleration parameter. A motion is generated by connecting several generated actions.

Figure 10 shows a motion of the humanoid robot of raising his right arm using the automatically generated motion.

The dance motion is generated using the joint angles of pos-

Figure 10. Humanoid playing a motion generated by the motion player.

tures loaded from the motion data. Twenty-three joint angles that describe a posture are downloaded from the motion editor in the computer to the controller in the humanoid.

The robot then assumes another posture and a new set of 23 joint angles and specific times are sent to the controller. The play time is the time slot between two successive postures. When the joint angles and the play time are sent to the controller on the robot, it calculates and interpolates each angle of the joints to take on the next posture in that specific time from the current posture.

By repeating this process, the humanoid's dance motion is generated and played. Furthermore, by changing the combination of postures and selecting different postures every time, the system can generate various types of dance performances.

4. Conclusion

In this study, a method to capture motion directly from a humanoid robot is proposed. When the user manipulates the posture of the robot, the angles of the joint motors involved in the robot posture are captured using the angle sensors built in the joint motors of the robot. The captured humanoid joint angles are used as data for generating the motion of the robot. The generated motion is tested on the humanoid robot, CHARLES2, to demonstrate a dance performance. The proposed method does not require any additional equipment, has no space constraints, and can generate all the postures that the humanoid can possibly assume. In particular, the proposed method can generate as many motions as the degrees of freedom of the robot.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgements

This Research was supported by the Research Funds of Mokpo National University in 2017.

References

- [1] Y. J. Ryoo, "Walking engine using ZMP criterion and feedback control for child-sized humanoid robot," *International Journal of Humanoid Robotics*, vol. 13, no. 4, article no. 1650021, 2016. <https://doi.org/10.1142/S0219843616500213>
- [2] K. Lee, Y. J. Ryoo, K. S. Byun, and J. Choi, "Omni-directional walking pattern generator for child-sized humanoid robot, CHARLES2," *International Journal of Humanoid Robotics*, vol. 14, no. 2, article no. 1750004, 2017. <https://doi.org/10.1142/S0219843616500213>
- [3] B. H. Kim, "Analysis on load torque effect for assistive robotic arms," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 4, pp. 276-283, 2018. <https://doi.org/10.5391/IJFIS.2018.18.4.276>
- [4] R. Qin, C. Zhou, H. Zhu, M. Shi, F. Chao, and N. Li, "A music-driven dance system of humanoid robots," *International Journal of Humanoid Robotics*, vol. 15, no. 5, Article no. 1850023, 2018. <https://doi.org/10.1142/S0219843618500238>
- [5] O. E. Ramos, N. Mansard, O. Stasse, C. Benazeth, S. Hak, and L. Saab, "Dancing humanoid robots: systematic use of OSID to compute dynamically consistent movements following a motion capture pattern," *IEEE Robotics & Automation Magazine*, vol. 22, no. 4, pp. 16-26, 2015. <https://doi.org/10.1109/MRA.2015.2415048>
- [6] P. Sousa, J. L. Oliveira, L. P. Reis, and F. Gouyon, "Humanized robot dancing: humanoid motion retargeting based in a metrical representation of human dance styles," in *Progress in Artificial Intelligence*. Heidelberg: Springer, 2011, pp. 392-406. https://doi.org/10.1007/978-3-642-24769-9_29
- [7] A. Wakabayashi, S. Motomura, and S. Kato, "Associative motion generation for humanoid robot reflecting human body movement," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 12, no. 2, pp. 121-130, 2012. <https://doi.org/10.5391/IJFIS.2012.12.2.121>

- [8] I. Infantino, A. Augello, A. Manfre, G. Pilato, and F. Vella, "Robodanza: live performances of a creative dancing humanoid," in *Proceedings of the 7th International Conference on Computational Creativity*, Paris, France, 2016, pp. 388-395.
- [9] M. Zebeay, V. Lippi, and T. Mergener, "Human-like humanoid robot posture control," in *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Colmar, France, 2015, pp. 304-309.
- [10] K. R. Hamid, A. Talukder, and A. E. Islam, "Implementation of fuzzy aided Kalman filter for tracking a moving object in two-dimensional space," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 2, pp. 85-96, 2018. <https://doi.org/10.5391/IJFIS.2018.18.2.85>
- [11] B. J. Park, H. J. Lee, K. K. Oh, and C. J. Moon, "Jerk-limited time-optimal reference trajectory generation for robot actuators," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 4, pp. 264-271, 2017. <https://doi.org/10.5391/IJFIS.2017.17.4.264>

Kinam Lee received his B.S., M.S., and Ph.D. degrees from the Department of Control Engineering and Robotics, Mokpo National University in 2011, 2013, and 2019, respectively. He is currently a researcher at Robot Artificial Intelligence

Convergence Center, and working on projects related to humanoid robotics.

Young-Jae Ryoo received his Ph.D, MS and BS degree in the Department of Electrical Engineering, Chonnam National University, South Korea in 1998, 1993, and 1991 respectively. He was a visiting researcher in North Carolina A&T State University, US in 1999. He was a visiting professor in the Department of Mechanical Engineering, Virginia Tech, US from 2010 to 2012. He is currently a professor in the Department of Control Engineering and Robotics, Mokpo National University, South Korea from 2000. He also served as a director with the intelligent space laboratory, and the robot and artificial intelligence convergence center in Mokpo National University, where he is responsible for the research projects in the area of intelligence, robotics and vehicles. He is currently a vice-president of Korean Institute of Intelligent Systems, an editor for the Journal of Korean Institute of Electrical Engineering from 2010, an editor for the Journal of Fuzzy Logic and Intelligent Systems from 2009, and a committee member of the International Symposium on Advanced Intelligent Systems from 2005. He served as a general chair of International Symposium on Advanced Intelligent System in 2015 and 2016. He is currently an associate editor of International Journal of Humanoid Robotics. He won the outstanding paper awards, the best presentation awards, and the recognition awards in International Symposia on Advanced Intelligent Systems. He is the author of over 200 technical publications. His research interests include intelligent space, humanoid robotics, legged robotics, autonomous vehicles, unmanned vehicles, wheeled robotics and biomimetic robotics.

E-mail: yjryoo@mokpo.ac.kr