Indian Journal of Science and Technology, Vol 8(25), DOI: 10.17485/ijst/2015/v8i25/86812, October 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Implementation of Humanoid Robot Dancing Performance Scenario Generation Program

Hyun-Gon Seo* **and Ho-Jin Ha**

School of Information Communication and Broadcasting Engineering, Halla University, Korea; hgseo@halla.ac.kr

Abstract

Recently, humanoid robot dancing performance is a new business model at IT convergence. In this paper, we have developed HrDPS (Humanoid robot Dancing Performance Scenario Generation) program. The HrDPS based on GUI generates a robot dancing scenario file using robot motion segment which is written in BASIC language. The motion Database in HrDPS provides reusable dancing motions and permits generating efficient robot dancing scenario.

Keywords: Humanoid Robot, Motion Database, Performance Scenario, Robo Basic

1. Introduction

Currently robot performance planning using humanoid robots are being highlighted as a new growth area of the robot industry. Thus a business model such as imitating human movement with robots and making robots dance to music (Culture content such as shows and performances) are being activated[1]. One of the most important aspects of performances using robots is performance content. For performance content, in the case of singer robot, it needs to be able to perform songs and dances according to music, and in the case of acting robot, it must be able to perform dialogue, actions, expressions, and background music according to scenario, and being able to be in tune with the fellow performer is also required[2].

For this the study tries to develop a humanoid robot performance scenario (Humanoid robot Dancing Performance Scenario Generation: HrDPS) that can help in making a piece of work in short time with low cost in robot performance planning. Previously a robot performance planner listened to the sound source then made the actions of the robots according to music to make the entire performance scenario. In this study it was made possible to make robot performance production scenario easy and fast by saving basic actions that can be done by robots into a database and enabling the planner to use the stored motions in the database.

The study is composed as follows. Chapter 2 explains the development and research trends in robot performance, chapter 3 explains the implementation of the humanoid robot performance scenario automatic creation system suggested in this study, and chapter 4 is the conclusion.

2. Related Works

2.1 Domestic and International Robot Performance Development and Research Trends

"Robot Tata and music robot: Save the earth!" is Korea's first experience type robot musical play independently developed by the only entertainment robot performance agency in Korea, KoIAN[3]. The biggest characteristic of this musical play is that it is not a one way performance where spectators watch, but where spectators actively communicate with the robot to participate in the play. "Legend of robot land" is the first robot animation musical in Korea where actual humanoid robots act, and it pioneered new culture content[4]. mRobo of Tosy company

* Author for correspondence

is a dancing robot with integrated portable speaker and when music starts, mRobo changes from standstill mode to dancing mode (full body dancing robot) to dance according to performance scenario[5].

Aldebaran Robotics announced a NAO that enables high interaction with robots with increased computing power. NAO has 1.6 GHz Intel Atom processor and 2 HD cameras and it has the characteristics of fast in object recognition and ability to process high definition video stream. It also has electric motors and actuators that can move 25 degrees and has the voice recognition program from iPhone 4S[6]. In[7] suggested a method for obtaining meaningful motion from motion capture equipment to convert it for humanoid robots, then changing the data in real time according to situation. Nakaoka is a motion capture system that suggested systematic process and imitation algorithm of capturing traditional Japanese dances with motion capture system so that humanoid robot (Model HRP-IS) could imitate it[8].

HRP-4C is a program for motion creation and it introduced a motion composition program through EveR-3 female actor robot[9]. Also in[10], for the natural motion creation of performance robots, using UML technique, eyes, neck, and body movement plan method was suggested.

2.2 Robot Movement using Servo Motor

Movement of the humanoid robot is achieved by controlling a number of servo motors that make up the robot. In this paper, the Humanoid robot body has 16 joints; each joint is adjusted to move in one of the servo motor. Servo motor of a humanoid robot is a group according to the location where the A, B, C, belongs to D. Configuration is shown in Figure 1. Group A, as shown in Figure 1 controls the movement of the right leg of a humanoid robotic, and group B is to control the movement of the left leg of a humanoid robotic. Typical basic language does not provide a motor driving command or related to the sensor signal. However, roboBASIC language provides such instruction. So we move the robot by specifying the value of each servo motor drive using roboBASIC language. Figure 1 of the instruction "MOVE G6A, 100, 155, 27, 140, 100" are left to the first motor of the robot 100, the second motor 155, the third motor 27, respectively fourth 140, and 100 in the last motor the means specified.

The roboBASIC code is made up of several mds (moving distance segmentation) that denotes of robot motion value at a time. It consists of motion value of robot's legs and arms. The following example shows the roboBASIC code of the one mds using both legs and arms of humanoid robot.

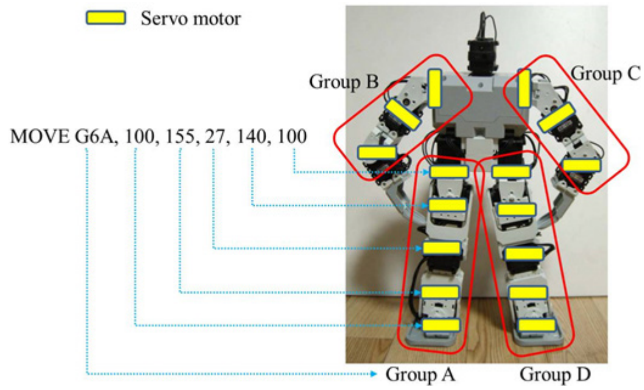

Figure 1. Humanoid robot (16 joints).

```
MOVE G6A, 100, 155, 27, 140, 100 ; left leg
MOVE G6D, 100, 155, 27, 140, 100 ; right leg
MOVE G6B, 130, 30, 85 ; left arm
MOVE G6C, 130, 30, 85 ; right arm
```

Here is the roboBASIC code showing the *mds* with only two legs of the robot.

```
MOVE G6A, 110, 105, 75, 170, 70; left leg
MOVE G6D, 110, 135, 20, 62, 40; right leg
```

Humanoid Robot movements can be made using several mds. In other words, the dance movements of a humanoid Robot is made meaningful combination of mds. The following example is part of robobasic source code that runs on the "Raise both arms of the Robot shaking from side to side" behavior of a humanoid Robot.

```
; Motion : raise both arms of the robot shaking from side to side
; #1 mds
SPEED 10
MOVE G6B, 185, 10, 100
MOVE G6C, 185, 10, 100
WAIT
; #2 mds
SPEED 10
MOVE G6B, 185, 160, 25
MOVE G6C, 185, 160, 25
WAIT
; #3 mds
SPEED 15
MOVE G6A, 100, 160, 110, 140, 100
MOVE G6D, 100, 160, 110, 140, 100
```

```
MOVE G6B, 140, 70, 20
MOVE G6C, 140, 70, 20
WAIT
RETURN
```

3. Implement of HrDPF

3.1 Motion Database

In robot performance there are many fundamentally used motions. By saving these motions into a database and enabling recalling of necessary motion when making performance scenario, it is very efficient in making robot performance planning program. Therefore this study designed a motion database so that frequently used motion and previously made motion can be stored in the database to enable saving cost and time when making robot performance planning program. The motion database was made possible to categorize by time of execution or certain motion.

Motion database uses 2 schemas, which are motion_schema that display motion and music_schema that display certain songs. Motion_schema specifies motions used in robot performances and data such as name of motion, time to execute motion (unit: seconds), characteristics of motion, and music used in making motion, other details, and speed of motion is specified. Also the motion type is recognized as "standard motion", "applied motion", and "specific music insert action." Music_schema specifies data about the sounds used in the performance and data such as name of sound file, artist, genre (ballad, R&B, rock, dance, etc.), total play time, and sound segment where sound is segmented to divide into dance motions, is specified. In this study the sound was divided into several sound segments and it is made possible for each sound segment actions to be chosen from the stored motions in the motion database to make a humanoid overall performance scenario.

Currently the number of created basic motions in this study is around 120 and motions that were made according to certain music is saved in the database, and it is planned to continuously make more diverse motion and motion that can add to the fun of robot performances.

3.2 mp3 Decoding

The mp3 decoding process consists of pivoting headers, side information and main data from the input bit stream. Decoding algorithm first finds 12 bit sync word from the bit stream, then finds all the data needed in the decoding process from the side data, and using the data found here the scale factor and Huffman samples are derived from the main database and after the other data is used in the process of reconfiguring audio sample[11–13]. The reconfiguration process uses each extracted information and data and through inverse quantization process audio data in the frequency domain which consist of 32 sub bands is reconfigured, and through IMDCT (Inverse Modified Discrete Cosine Transform) and sub band synthesis and filtering the final audio is outputted.

In this study, the input mp3 file is decoded so that the robot performance planner listens to the sound and chooses the most appropriate motion in the motion database to make scenario files. In figure 2 looking at PCM (Pulse Code Modulation) waves the motion categorization is decided and the motion appropriate for the decided area is chosen from the motion database, and in the first area the motion "greeting1" for 5 seconds was chosen with the start of music, then for 8 seconds it executes "shake arms and legs" motion, then executes

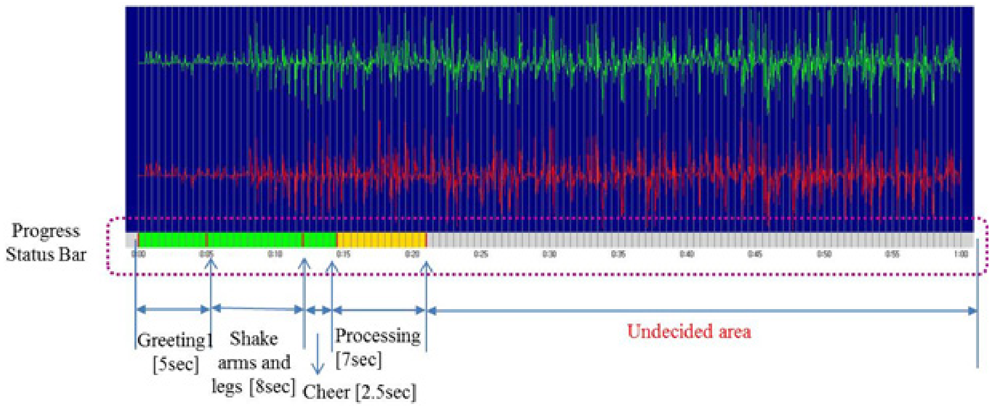

Figure 2. Motion binding process.

"cheer" motion for 2.5 seconds. Then the next 7 seconds of area is in process of choosing a motion and the areas after that is undecided area were specific motion is not specified.

Figure 3 is the representation of overall process of HrDPS. First the mp3 file is inputted and PCM pulse data is created, then PCM waveform graph is drawn. When PCM pulse waveform graph is made the sound is cut into segments and desired motion is chosen from motion database so that at certain times dance motions can be executed and source course binding to all scenario segments is conducted. If this process is executed in the event processing technique it is repeated until the binding for all sounds is done.

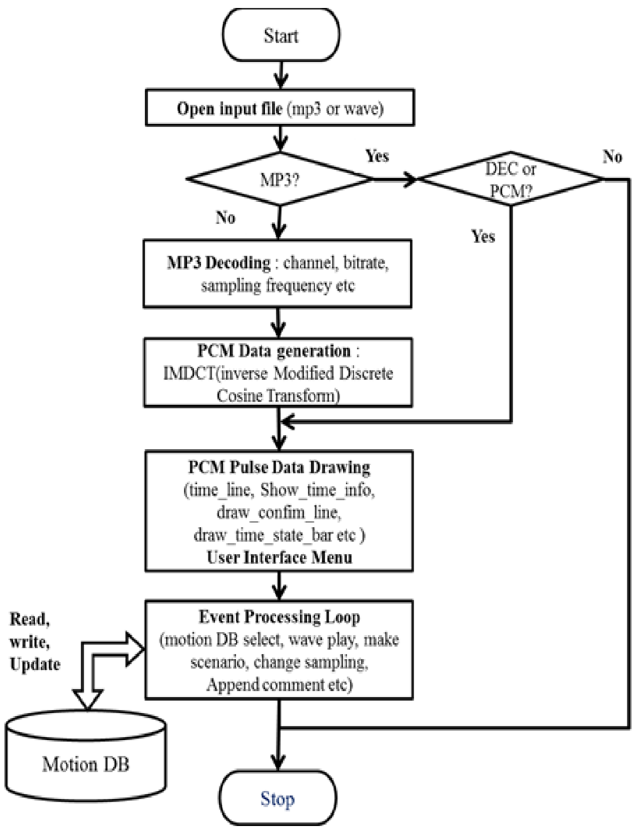

Figure 3. HrDPS flowchart.

3.3 Robot Dancing Scenario

In the proposed HrDPS program, the process of generating a RDS (Robot Dancing Scenario) consists of following steps: RDS made by BASIC language controls servo motors within the robot, so it moves to left, right, up and down. RDS planners listen to the mp3 file analyzing its attribute, and design dancing motions. But it's a very difficult work and requires high cost.

First, select the music (mp3 file) to be used in robot performance and then enter it into HrDPS program. Since the configuration of the scenario is affected by the selection of the music, the selection of the music to be used in the robot performance is important. If you choose a fast music, motion of the robot would be large and its moving speed would be fast as well. On the other hand, if you select slow tempo music, the robot would move slowly. Next step is to produce a PCM (Pulse Code Modulation) data which is obtained after decoding mp3 music file. Then draw the PCM pulse waveform graph using it. Decoding process of the mp3 file is composed of a process of extracting the headers, side information and main data from the input bit stream. Final step is a binding process. PCM pulse graph is divided into several fragments by RDS planner. Each fragment is mapped to MS (Motion Segment) in Motion Database. Thus each fragment corresponds to appropriate MS that contains BASIC source code for robot dance. After the last step, each motion segments are combined into a single RDS file, which is a final scenario.

The RDS is divided into two parts, a head portion and a body. The former one contains general scenario information such as file name, mp3 file name, date, total length. The latter one stores each MS records. Each MS has to execute a particular dancing mode, which is made by BASIC source code. It is stored in the Motion Database and selected according to the mp3 music by RDS planner. Figure 3 shows the head and body RDS file of this paper.

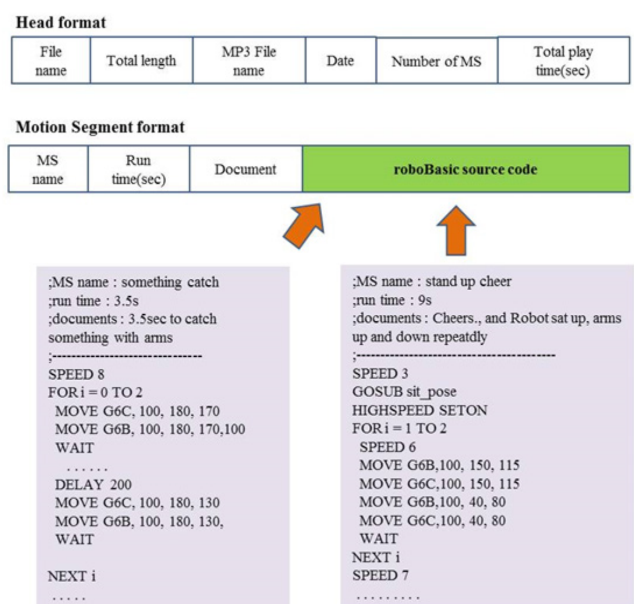

Figure 4. RDS format.

3.4 HrDPS Program

We implement the HrDPS program which is made by C++ program. Figure 5 shows the HrDPS execution windows. The HrDPS consists of seven parts. Figure 5 (a) shows Motion database window aligned along the MS runtime. If you want to make the robot dance for 7seconds with any music, you can select appropriate MS which corresponds to 7 seconds. MS source codes are displayed at Figure 5 (b) Motion Segment Source window. Scenario planner can modify code. Figure 5 (c) shows Scenario Script windows that represents all scenario codes. HrDPS program has Control buttons such as play_music, set_MS_range, make_scenario, stop etc. PCM pulse graph window which is drawn by mp3 decoder shows decoding values with two channels. Scenario planner decides MS range through observed PCM waveform graph. Figure 5 (f) is Progress Bar windows currently processing MS and figure 5 (g) is PCM graph position window which controls PCM wave graph to move forward and backward. Scenario planner analyzes PCM pulse wave shape and running time so that it can select motion segment range.

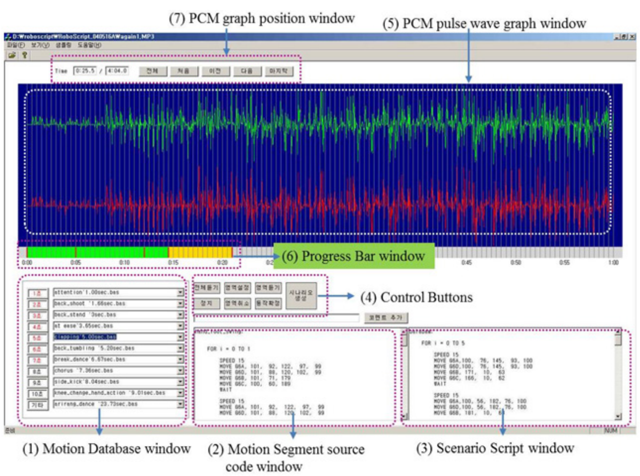

Figure 5. HrDPS execution. (a) Motion Database window. (b) Motion Segment source code window. (c) Scenario script window. (d) Control buttons. (e) PCM pulse wave graph window (f) Progress bar window. (g) PCM graph position window.

The scenario file, basic source program which created by HrDPS must be compiled to execution at Humanoid robot. The roboBASIC is a compiler that makes executable object program that should be downloaded on Humanoid Robot[14]. In the proposed HrDPS program, the process of generating a RDS (Robot Dancing Scenario) consists of following steps: RDS made by BASIC language controls servo motors within the robot, so it moves to left, right, up and down. RDS planners listen to the mp3 file analyzing its attribute, and design dancing motions. But it's a very difficult work and requires high cost. First, select the music (mp3 file) to be used in robot performance and then enter it into HrDPS program. Since the configuration of the scenario is affected by the selection of the music, the selection of the music to be used in the robot performance is important. If you choose a fast music, motion of the robot would be large and its moving speed would be fast as well. On the other hand, if you select slow tempo music, the robot would move slowly.

4. Conclusions

In this study a performance scenario planning program HrDPS used in robot performance system was developed. HrDPS receives mp3 input to create PCM, chooses motion stored in the motion database for specified sound segment so that the scenario according to music for the robot to perform can be created. The motion database stores motion such as standard motion previously used in robot performance, applied motion, and motion made specifically for certain music, and according to need the motion can be edited and updated. Motion database promotes reutilization of motion, management of developed motion and it provides efficiency and time and production cost saving. As a future research project there is saving various motion in the motion database using such things as motion capture and implementing a function where graphic based robot performance motion can be developed.

5. References

1. Hwan KJ, Gon SH. Motion database design and build a humanoid robot. IEEK Summer Conference; 2013. p. 2557–9.
2. Seok AH, Uk LD, Un CD, Hyun LD, Hong HM, Gil LH. Behavior composition program for performance robots. ICROS 2012; 2012. p. 27–8.
3. Koian. Available from: http://www.koian.com/
4. ESan. Available from: http://www.robotlegend.co.kr/
5. Tosy. Available from: http://tosy.com/index.php/mrobo.html
6. Aldebaran. Available from: http://www.aldebaran.com/en
7. Seungsu K, Hwan KC, Hyeon PJ, Bumjae Y. Regeneration of upper-body motion of humanoid robot using human motion capture data. The Korean Institute of Electrical Engineers. 2006; 1935–6.

8. Shinichiro N, Atsushi N, Kazuhito Y, Hirohisa H, Katsushi I. Generating whole body motions for a bipel humanoid robot from captured human dances. International Conference on Robotics and Automation. 2003; 3:3905–10.
9. Kanako M, et al. Trials of cybemetic human HRP-4C toward humanoid business. In Proceedings of the 2010 IEEE international Workshop on Advanced Robotics and its Social Impacts (ARSO 2010); Seoul. 2010 Oct 26-28. p. 165–9.
10. Kim WT, An YS, Lee HG, Lee TH. UML-based motion planning for natural behavior of performance robots. ICROS 2010; 2010. p. 524–5.
11. Byungwoo Y. Implementation of MP3 decoder with TMS320C541 DSP. The Korea Institute of Signal Processing and System. 2003; 4(3):7–14.
12. Davis P. A tutorial on MPEG/audio compress. IEEE Multimedia. 1995; 2(2):60–74.
13. Chales DM, Anandakumar K. Real-time MPEG-1 audio coding and decoding on a DSP chip. IEEE Transactions on Consumer Electronics. 1997; 43(1):40–7.
14. miniRobot. Available from: http://www.minirobot.co.kr/