SICE Annual Conference 2013
September 14-17, 2013, Nagoya University, Nagoya, Japan

Motion Generation For the Stuffed-toy Robot

Yutaka Takase[1], Hironori Mitake[1], Yohei Yamashita[1] and Shoichi Hasegawa[1]

[1]Precision and Intelligence Laboratory, Tokyo Insutitute of University , Tokyo, Japan
(Tel: +81-45-924-5049; E-mail: {takase, mitake, yama, hase}@haselab.net)

Abstract: A number of entertainment robots to entertain or heal people have applied hard mechanism in order to equip sensors for taking surrounding conditions, or making various motions. However, in point of familiarity, softness and good hand feeling are important factors for the robots. A stuffed-toy robot we proposed has five motion parts made of fabrics (cloth, cotton and thread) and can various body motions. In this paper, we propose a reaction generation method for the stuffed-toy robot. In this method, we use external sensors to get positions and actions of users around the robot. And we derive the amount of attention of robot to each object based on these data. Thereby, the robot gazes and reaches to a target decided the amount of attention. It is expected that the robot is able to react to varied user inputs.

Keywords: Stuffe-toy robot, Motion Generation, Selective Attention, Kinect Sensor

1. INTRODUCTION

Recently, a lot of entertainment robots to entertain or heal people by various motions, gestures, dance and so on, are put to practical use. To prompt user to do more close interaction such as touch or hug with the robots, some of them are modeled on stuffed animals or real animals.

However, most of these robots apply hard mechanism at moving parts to achieve the various motions. Hence, these hard mechanism robots provide user with the sense of incongruity due to their tactile feeling differing from their looks. It may be suspected that it's a factor of avoiding interaction with the robots by users. Additionally, we have to give attention to prevent breaks and accidents when children play with the robots. A study by Harlow [1] using the neonatal and infant macaque monkey suggests that softness and good hand feeling are important factors to improve familiality for robot. Thus, we should mount these factors for entertainment robots.

We proposed a stuffed-toy robot [2, 3] who has softness and tactile impression as same as stuffed animal at moving parts. And the robot can do various body motions. In addition, the study suggested that the robot makes a favorable impression on users than hard mechanism robot.

In this paper, we propose a motion generation method for the stuffed-toy robot shown in Figure 1 to interact with users. This method based on study of Mitake *et al* [4]. By using selective attention model, the robot can select a point with highest priority from among multiple target points. And the robot gazes and reaches the point.

Thereby, the stuffed-toy robot can generate reaction behaviors for various user input.

2. PREVIOUS WORKS

A number of entertainment robots are coming into practical use. There are many methods to generate motion for the robots.

PINKYO[5] is a ring-like device that provides movement to any stuffed animals. The device gives simple and limited motions such as raise and lower the arms to any stuffed animals.

Fig. 1 The Stuffed-toy Robot

Some entertainment robots for example AIBO[6] and PARO[7] change their motion by their own state or the environmental state. These motions are prepared in advance. However, they are only able to react by simple gesture motions.

In other hand, Robot aimed to physical tasks such as nursing care[8] must instantly change the motion in a precise depending on current posture, force applied the robot and so on. These robots have a lot of sensors inside their body parts to improve the operation accuracy and the safety. The reason of this, they have hardmechanism and bad hand feeling.

In our stuffed-toy robot, to keep softness and good hand feeling, we must avoid installing a lot of sensors in it. The motion generation method we proposed in this paper is to generate reaction behavior for various user inputs by using a external sensor.

3. STUFFED-TOY ROBOT SOFT TO THE BONE

In this section, we elaborate about the stuffed-toy robot we proposed.

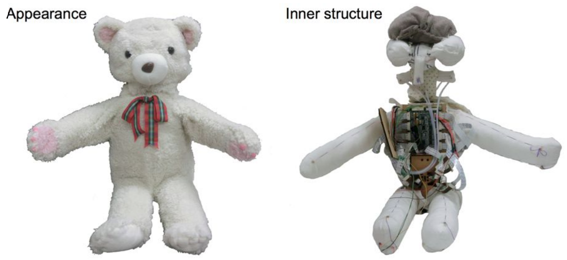

Fig. 2 Stuffed-toy Robot

Fig. 3 Principle of the Soft Mechanism

3.1 Overview

Figure 2 shows appearance and inner structure of the robot. The robot has five moving parts, 2-DOF arms, 2-DOF legs, and a 2-DOF neck. All of them consist of fabrics such as cloth, thread and cotton. Hence, the motion parts have softness and good hand feeling as same as stuffed animals. The robot is modeled on bear stuffed animal because it seems that bear stuffed animal is loved by everyone, regardless of nationality and generation, like teddy bear.

Normally, the robot is seated. In the basic posture, the robot measures 380 mm in height, 240 mm in width, 300 mm in depth and 1.4 kg in weight. And, it is battery powered. Operating voltage is 8.3 V. The power consumption is about 27 Watt. In addition, it is controlled via wireless LAN from laptop computer.

3.2 Soft Mechanism

The soft mechanism consists of two parts, moving parts made of fabrics and actuators to drive the moving parts. Figure 3 shows the principle of the soft mechanism. Base mechanism of the moving parts uses cotton crammed cylindrical cloth with threads stitched on the surface. By pulling these threads by motors, the cylindrical cloth is bent. Thereby, hard motors can the separated from soft moving parts. The motors and the drive circuit board to drive them are put into the body parts to keep the softness of the moving parts of the robot.

Moreover, using multiple threads and pulling some of them allow the bend to any directions.

3.2.1 Design of the Arms and Legs

Figure 4 shows the designs of arms and legs to achieve softness and to perform the various motions. Each cylindrical cloth uses three threads, and up to two threads are pulled at once for bend.

3.2.2 Design of the Head

Figure 5 shows a design of the head. We use two threads for looking up and down, and four threads for looking left and right. By pulling some of the six threads, the robot can do action such as nod, looking up, taking a look-around and so on.

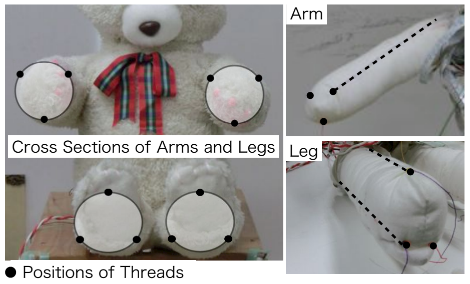

Fig. 4 Positions of the Threads of the Arms and Legs

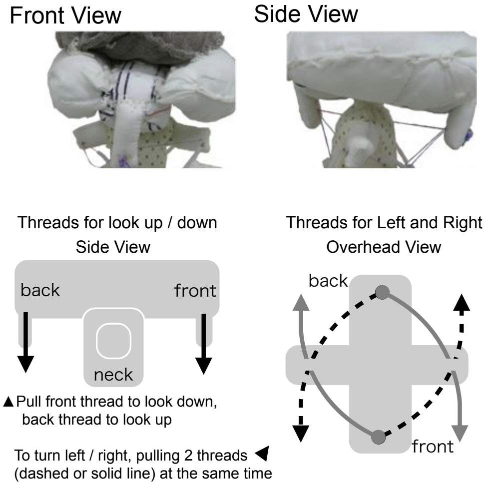

Fig. 5 Structure and Positions of Threads of Neck

3.3 Motion Ranges

Figure 6 shows motion ranges of the left arm, the left leg and the head from the anterior view (the range of the right arm and leg are symmetrical to the left one). It is capable of moving its arm toe and leg toe up to about ± 100 mm. And, the head is capable of moving up to 70 degrees to left and right, up to 20 degrees to up and down. As mentioned above, the stuffed-toy robot has a softness and good hand feeling as same as stuffed animals at moving parts and large motion ranges to perform various body motions.

3.4 Posture Control

We use coreless DC motors with rotary encoder for getting current thread lengths (equal to the current posture of the robot) and pulley for pulling thread. In order to change the posture of the robot, the thread lengths are controlled by adjusting the duty cycle for PWM signals to the motors.

4. MOTION GENERATION METHOD

In this section, we elaborate a motion generation method for the stuffed-toy robot. It is based on selective attention model [4]. By using this method, the robot moves its head and hands to a target that is decided by the position and velocity from among the objects around it. It can be expected that the robot respond with varied reactions according to user input.

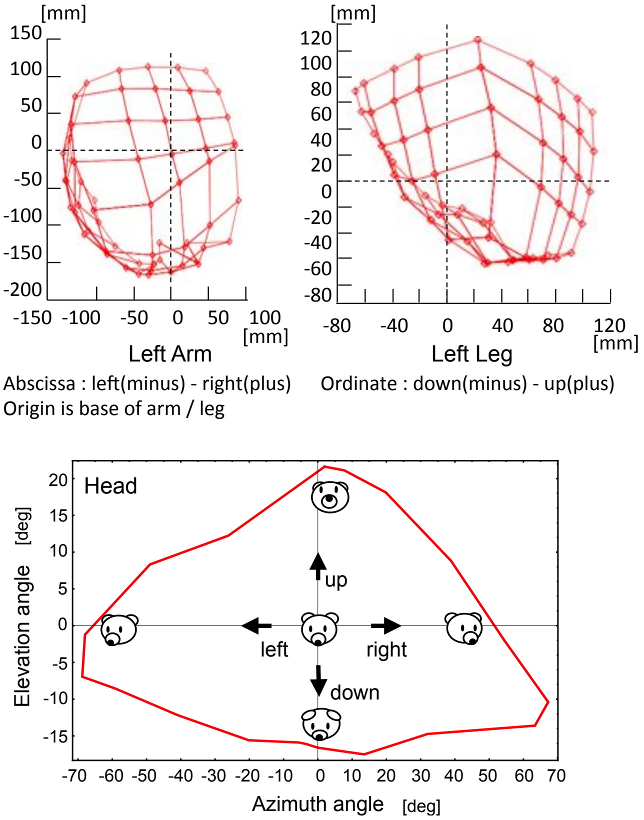

Fig. 6 Motion Ranges : Left Arm is Upper Left , Left Leg is Upper Right and Head is below

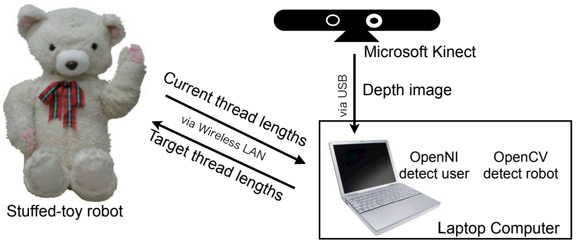

Fig. 7 System for Motion Generate

4.1 Configureation of the System

Figure 7 shows a configuration of system for motion generation. We use Microsoft's Kinect, OpenNI and OpenCV to recognize around the robot and a laptop computer to process information, generate motion and communicate with the robot. Due to installing Kinect sensor outside of the robot, we can keep softness and good hand feeling of it.

4.2 Recognize the Posture of the Robot

As mentioned above, we use the length of threads to recognize posture of the robot. Therefore, in the case where we want to control its posture, we need to know the relationship between thread lengths and the posture. So we measure it in advance for each arms, legs and head as follows. We call this measured data **Lengths Posture Map**.

Fig. 8 Position of Kinect Sensor

4.2.1 Measuring for the Arms and Legs

Three threads are used at each arm and leg. To make maps, we pull each thread bit-by-bit and record thread lengths and position of hand/leg toe from the root of the arm/leg by using Optotrack (NDI). As a result, every part has 57 points.

4.2.2 Measuring for the Head

The head mechanism is separated elevation angle control and azimuth angle control. So, we hold only five points of thread lengths that is origin point, max of right, left, up and down. And we assume that the relation of angles and thread lengths is linear. Thereby, we are capable of moving the head to any directions with some simple calculation.

As presented above, we measure these points in the local coordinate system with their origin at the root of arm/leg or the angel of facing the front. All maps are hold in the computer.

4.3 Motion Generation

The motion generation method consist of 4 steps : 1) Recognize users and the robot by Kinect. 2) Derive amount of attention. 3) Determine a trajectory of motion. 4) Send target thread lengths to the robot.

4.3.1 Recognize Users and the Robot

The system takes positions of users around the robot by using Microsoft Kinect and OpenNI. As illustrated in Fig.8, the Kinect sensor is installed behind the robot. Using the Kinect depth camera, we can take positions of users from the Kinect (world coordinate system). Regarding users, the system takes positions as four parts (head, hands and body) in each user. OpenNI can recognize up to 15 people. Hence, the system holds a maximum of 60 parts' positions.

Next, the system derives verocities of the each object from the equation (1).

$$V_o(t) = \frac{P_o(t) - P_o(t+\Delta t)}{\Delta t} \tag{1}$$

In the equation, we define the velocity of an object at time t is $V_o(t)$. $P_o(t)$ and $P_o(t+\Delta t)$ are positions of the object at time t and $t+\Delta t$. In the system, Δt is 30 msec. It depends on the frame rate of the Kinect. As a result, the system holds positions and velocities of all objects it recognized.

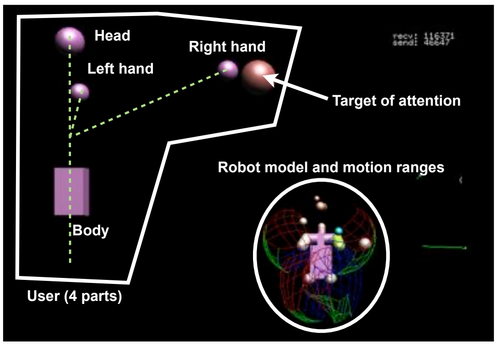

Fig. 9 Recognized world by Kinect Sensor

In addition, the system must take positions of the robot in the world coordinate system. To do this, we use not only depth image but also template matching with OpenCV. By using back shot image of the robot for template image, we detect position of it in the world coordinate system. Essentially we have to take the direction of the robot, however, the robot has no sensors to get the direction of it. Therefore, as yet, we assume the robot fixes the direction in the world coordinate system.

And, the Length Posture Maps are converted from the local coordinate system to the world coordinate system with the position of the robot. Figure 9 shown a recognized world constructed by the system. This world includes the robot and users.

4.3.2 Calacurate Amount of Attention

Next, we derive the amount of attention each part of users. For example Fig. 9, the system recognizes a user, hence four positions and velocities are held. We define the position and velocity of one part is $P_o(t)$,$V_o(t)$ and the position of the robot is $P_r(t)$. We use the position of the head of the robot at time t as $P_r(t)$. Then, the amount of attention $A_t(t)$ is calculated for every object as described Equation (2).

$$A_t(t) = \frac{|V_o(t)|}{|P_o(t) - P_r(t)|} \tag{2}$$

This equation means that the robot is interested more close object or active object. An object that has a maximum attention is selected as target object of the robot.

4.3.3 Determination of the Trajectory

Finally, we determine motion trajectories with a target object position $P_t(t)$ for the arms and the head. Figure 10 shows a schematic of the determination of one arm motion. In the picture, $P_{ra}(t)$ is the position of the root of the arm. The grid and dots express Length Posture Map and each posture. LP_n is the current posture of the arm. The trajectory of the arm is decided as follows : 1) Connect $P_t(t)$ and $P_{ra}(t)$ with a straight line $\boldsymbol{l}$. 2) Get intersection point of a surface projected Lenght Posture Map and the line $\boldsymbol{l}$. 3) Get a neighbor point LP_m of the Length Posture Map from the intersection point. Conclusively, the trajectory is calculated linear interpolation of the thread length from LP_n to LP_m. And this calculation also applies to the other arm.

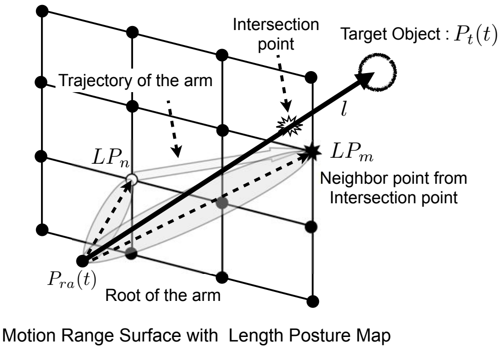

Fig. 10 Determination of Arm Trajectory

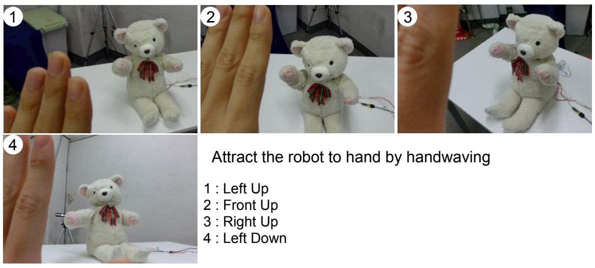

Fig. 11 Generated Motion

Regarding head motion, we get elevation angle and azimuth angle of target object. And, the trajectory is derived by linear interpolation from current angles to target angles.

4.3.4 Sent Target Thread Lengths

The system divides the three trajectories (head and arms) in each 200 points. The points are sent to the control circuit of the robot in small quantity. We call a unit of the sending data "Packet". One packet contains 20 points of each trajectory.

At first, the system sends 16 packets to the robot. When the robot reaches final point of one packet, the system send next one packet to the robot. The reason for this is to avoid influence of network delay. The control circuit received data controls all thread lengths by adjusting the duty cycle for PWM signals.

For example, by using this method, multiple user can scramble for attention of the robot by handwaving all together.

5. RESULT AND DEMONSTRATION

Figure 11 shows some examples of motions which the system generated. I attracted the robot to hand by handwaving from several directions. These results showed that arms moved comparatively well, but the head didn't work as expected. One reason for this is because the motion range of the head especially elevation angle is narrow. Further studies about the structure of the head are needed in order to expand the motion range. Additionally, it seems that we must develop an efficient motion generation method for the head.

And, as shown in Figure 12, we demonstrated the stuffed-toy robot at SIGGRAPH 2012 (Los Ange-

Fig. 12 Demonstration

les, USA) and Entertainment Computing 2012 (Hyogo, JAPAN).

Some visitors attracted attention of the robot by hand-waving gesture with multiple users. Children play with the robot by taking it in their arms. In that case, the robot could not run normally. Because, detection by Kinect sensor failed. Even in this case, they enjoyed playing with the robot by touching the soft and running moving parts.

The robot attracted favorable comment from visitors about its softness and dynamic motion. And, several people are bewildered by uncertainness or delay of reaction behaviors.

6. CONCLUSION AND FUTURE

In the present study, we proposed motion generation method for stuffed-toy robot.

With an external sensor, Microsoft Kinect, the proposed method generates gazing and reaching motions without preventing softness and good hand feeling of the robot.

And, we will work through tasks as described below.

First, because of we can not understand the robot direction, it seems that accuracy of the generated motion decreases if the robot is moved by user. It would solve that we install direction sensor to the robot.

Second, by reason of problem of head structure, motion of gazing does not work we expected. To solve this, we must upgrade both hardware and software of the robots.

Moreover, we must evaluate generated motions about both quantitative (such as delay and accuracy of motions) and qualitative (such as impression of motions).

And, to achieve interacting in more complex the robot with people, we would use soft touch sensors to detect that the robot is touched by user. If it does, by using selective attention model, the robot would be able to generate more complex motions.

In the future, interactions with entertainment robots would become more complex, various and close. In such cases, the stuffed-toy robot would provide people with further entertaiment and comfort.

REFERENCES

[1] Harry F. Harlow. The Nature of Love. *American Psychologist*, Vol. 13, pp. 673–685, 1958.

[2] Yohei Yamashita, Tatsuya Ishikawa, Hironori Mitake, Yutaka Takase, Fumihiro Kato, Ikumi Susa, Shoichi Hasegawa, and Makoto Sato. Stuffed toys alive! In *ACM SIGGRAPH 2012 Emerging Technologies on - SIGGRAPH '12*, pp. 1–1, New York, New York, USA, August 2012. ACM Press.

[3] Takase Yutaka, Hironori Mitake, yohei Yamashita, Tatsuya Ishikawa, Mina Shiina, and Shoichi Hasegawa. Cuddly robot soft to the bone (wroking paper / in japanese). 2013.

[4] Hironori Mitake, Shoichi Hasegawa, and Makoto Sato. Reactive Virtual Creatures for Dexterous Physical Interactions. In JanM. Allbeck and Petros Faloutsos, editors, *Motion in Games SE - 8*, Vol. 7060 of *Lecture Notes in Computer Science*, pp. 86–97. Springer Berlin Heidelberg, 2011.

[5] Yuta Sugiura, Calista Lee, Masayasu Ogata, Anusha Withana, Yasutoshi Makino, Daisuke Sakamoto, Masahiko Inami, and Takeo Igarashi. PINOKY: a ring-like device that gives movement to any plush toy. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, p. 725, New York, New York, USA, May 2012. ACM Press.

[6] M. Fujita. On activating human communications with pet-type robot AIBO. *Proceedings of the IEEE*, Vol. 92, No. 11, pp. 1804–1813, November 2004.

[7] T. Shibata, T. Mitsui, K. Wada, A. Touda, T. Kumasaka, K. Tagami, and K. Tanie. Mental commit robot and its application to therapy of children. In *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No.01TH8556)*, Vol. 2, pp. 1053–1058. IEEE.

[8] Tadashi Odashima, Masaki Onishi, Kenji Tahara, Toshiharu Mukai, Shinya Hirano, Zhi Wei Luo, and Shigeyuki Hosoe. Development and Evaluation of a Human-interactive Robot Platform "RI-MAN". *Journal of the Robotics Society of Japan*, Vol. 25, No. 4, pp. 554–565, 2007.