Real-time Posture Imitation of Biped Humanoid Robot based on Particle Filter with Simple Joint Control for Standing Stabilization

Yo Kondo, Shohei Yamamoto, and Yasutake Takahashi
Graduate School of Engineering, University of Fukui
3-9-1, Bunkyo, Fukui, Fukui, 910-8507, Japan
Email: {ykondo,shyamamoto,yasutake}@ir.his.u-fukui.ac.jp

Abstract—The purpose of this study is a development of real-time imitation learning system for a humanoid robot. It is needed to estimate joint angles of the robot from the observation of human demonstration. However, it is difficult to measure the joint angles directly. Conventional motion capture systems which measure the joint angles of a human subject precisely are often expensive or hard to use in daily life. Recently, depth sensor has become popular to provide fully-body 3D motion capture because it has advantages of low cost and no markers or trackers. It provides the position of joints of the human subject, however, the joint angles for the imitating robot have to be calculated in some way. Inverse kinematics is often used for the joint angle calculation, however, it needs relatively high computational cost for the optimization calculation and sometimes it has a difficulty to have a unique solution because of redundancy. We propose to use a particle filter for joint angle imitation so it provides a reasonable solution with a less computational cost to realize a real-time imitation of a humanoid robot through observation of human demonstration. However, the particle filter does not provide the standing stability of the humanoid robot. Therefore, we propose a novel and simple method of control of leg joints for the standing stabilization. While the humanoid robot imitates the human posture, the ankle and the hip joint angles of the robot are controlled based on the knee and hip joints provided by the particle filter. We evaluate the proposed method with experiments using a humanoid robot, Aldebaran Robotics NAO, and show its validity.

I. INTRODUCTION

In recent years, a lot of robots have increased exposure in human daily life. It is desirable for the robots to adapt to the human environment by themselves because it is difficult to design all motion which is desired in daily life beforehand. Reinforcement learning is one of the most representative methods that robot acquires appropriate behavior by itself[1][2]. However, learning by trial and error takes relatively long learning time and high risk of breakdown for a real humanoid robot. Therefore, real-time imitation learning through observation of human demonstration has been attracted to shorten the learning time and reduce unnecessary trials[3].

Anton et al.[4] developed a real-time Human-Robot Interactive Coaching System (HRICS). The robot sends its camera vision to portable computing devices. The human coach with a full-body motion capture suit demonstrates the teaching motion while he/she checks the robot camera vision. They teach skills for playing soccer to the robot by the system.

Koenemann et al.[5] have proposed whole body imitation technique that introduces the normalized offset to estimate robot stability replacement for center of mass (CoM) position. Their humanoid robot imitates the human motion measured by a wearable motion capture system while it stabilizes itself considering the support leg mode. Although the humanoid robot is able to walk, grasp a cup and stand on one leg, the wearable motion capture system is expensive and hard to use in daily life.

Stanton et al.[6] propose a learning method for teleoperating a humanoid robot using a full-body motion capture suit. They introduce a feedforward neural network to generate appropriate joint angles of the humanoid robot to imitate the human posture. The neural network is optimized to approximate the pre-training function using particle swarm optimization. Although the demonstrator is able to teleoperate the robot in real-time, their method needs pre-training of the neural network before the posture imitation.

We propose a real-time joint angle estimation system using particle filter for humanoid robot posture imitation[7]. We adopt a depth sensor, Microsoft Kinect, instead of conventional expensive motion capture systems. The depth sensor has become popular to provide fully-body 3D motion capture because it has advantages of low cost and no markers or trackers. Particle filter method is adopted for estimation of joint angles for the humanoid robot to imitate the human demonstrated posture. It provides a reasonable solution with a less computational cost to realize a real-time imitation of a humanoid robot through observation of human demonstration. However, the particle filter does not provide the standing stability of the humanoid robot so that the robot falls down if it follows the joint angles estimated by the particle filter. Therefore, this paper proposes a novel and simple method of control of leg joints for the standing stabilization. While the humanoid robot imitates the human posture, for example, a bow and a swaying a hip right and left, the ankle and the hip joints of the robot are controlled based on the knee and hip joint angles estimated by the particle filter. The simple strategy of controlling the leg joint angles provides a reasonable standing stabilization that keeps the center of mass within the supporting polygon in real time. We evaluate the proposed method with experiments using a humanoid robot,
Aldebaran Robotics NAO, and show its validity.

II. EXPERIMENTAL SETUP

Our system is tested on a humanoid robot, NAO from Aldebaran Robotics, which imitates demonstrator’s motion and a Kinect as a motion capture system. Popular motion capture systems in conventional researches are based on optical sensors, mechanical sensors, or magnetic sensors. Unfortunately, those systems request a human demonstrator to equip the expensive sensors and/or makers before he/she demonstrate and it is inconvenient to use them in daily life. Microsoft Kinect (Kinect for Xbox 360) is inexpensive and a popular simple motion capture system. Kinect has a depth camera and an SDK to track a skeleton of a captured human subject so that the demonstrator does not need to put on any markers or sensors.

![Joint positions detected by Kinect](http://doc.aldebaran.com/2-1/family/nao_dcm/actuator_sensor_names.html)

Figure 1 shows the link postures which we define and calculate from the detected joint positions. Finally, we estimate 22 joint angles from the link postures by particle filter shown in Fig.2. We ignore LHipYawPitch, RHipYawPitch, and HeadYaw when to calculate similarity because it is difficult to recognize them by the Kinect.

III. REAL-TIME JOINT ANGLE ESTIMATION USING PARTICLE FILTER

The Kinect provides joint positions of the human demonstrator. The imitating robot has to estimate its joint angles from the human joint positions accordingly. Typically, the inverse kinematics algorithm is used for this purpose in conventional researches[8][9]. However, the inverse-kinematics often has multiple solutions because of the redundancy and it is time-consuming to calculate a unique one that takes a reasonable sequential motion into account.

Particle filter provides a reasonable solution of joint angle sequence for the imitation in real-time. This section explains how the particle filter is applied for the application of humanoid robot real-time imitation. Particle filter initializes a set of particles, each of which represents a set of joint angles of the robot. First, the particles are updated based on a motion model of the robot. Second, each particle is evaluated based on a measurement model using observed data and assigned a weight that represents how much the particle fits the observation. Then, the particles are resampled according to the weights. The filter continues the procedures repeatedly.

A. Coordinate Systems

The joint position observed by the depth camera is represented by camera coordinate system $\Sigma_c$ with the origin at the camera position. The joint position of the robot is calculated in the robot coordinate system $\Sigma_r$. Then we need to transform the camera coordinate $\Sigma_c$ to the robot coordinate system $\Sigma_r$ shown in Fig.3. The origin of $\Sigma_r$ is defined at HIP position of the robot. The $X_r$-axis is the direction from HIP_L to HIP_R,
the $Z_r$-axis is the direction from HIP to HEAD, and the $Y_r$-
axis is the direction of the cross product of the $X_r$ and the $Y_r$.
Rotation matrix $^cR_r$ is defined by Eq.(1).

\[
^cR_r = (^cx_r,^cy_r,^cz_r) \tag{1}
\]

where $^cx_r$, $^cy_r$, and $^cz_r$ are unit vectors of $X_r$, $Y_r$, and $Z_r$
axes in camera coordinate, respectively. Arbitrary joint position in
camera coordinate $^c\mathbf{p}_j$ is transformed to the joint position in
robot coordinate $^r\mathbf{p}_j$ with Eq.(2).

\[
^r\mathbf{p}_j = T^cR_r(^c\mathbf{p}_j - ^c\mathbf{p}_{HIP}) \tag{2}
\]

where $^c\mathbf{p}_{HIP}$ is the HIP-joint position in camera coordinate
and $T$ is a transposition of a matrix.

B. Definition of Similarity based on Link Postures

The robot observes the human demonstration with a depth
sensor and it evaluates the particles based on measurement
model with the observed data. A particle has a high score if
it fits the observation well. We propose to use link postures
between the robot and the human subject in order to evaluate
the particle, the set of joint angles of the robot, based on the
observed joint positions of the demonstrating human subject.
Human joints angle is not applicable to the humanoid robot
directly because the human structure is different from the one
of the robot. Therefore, we use the link posture to compare
postures of the demonstrating human subject and imitating
humanoid robot. A link is defined as a rigid body with adjacent
two joints as shown in Figure 1. For example, a forearm link
is defined with two joints, a wrist and an elbow. The links
posture should be compared in a same coordinate system.

Link postures are defined as a unit vector between adjacent
joints as shown in Fig.4. The link posture $j$ between the joint

\[
\varphi^j = \frac{\mathbf{p}_l - \mathbf{p}_m}{||\mathbf{p}_l - \mathbf{p}_m||} \tag{3}
\]

where $\mathbf{p}_l$ and $\mathbf{p}_m$ are position vector of $l$ and $m$.

The similarity of the robot and human posture is defined
based on the link postures. The similarity is calculated by
Eq.(4).

\[
s = \frac{1}{N} \sum_{i=1}^{N} h(\varphi^i) \cdot r(\varphi^i) \tag{4}
\]

where $s$ is similarity between human and robot, $N$ is number
of link postures, $r(\varphi^i)$ and $h(\varphi^i)$ are link postures of the robot
and the human in robot coordinate $\Sigma_r$, respectively.

C. Particle Filter

The particle filter estimates and tracks the set of joint angles
for the imitating humanoid robot based on the observation of
human demonstration. Algorithm 1 shows the algorithm of
particle filter for it.

**Algorithm 1 Particle filter**

1: Initialize particles $\Theta_t = (\theta_t^1, \theta_t^2, \ldots, \theta_t^M)$
2: for $m = 1$ to $M$ do
3:   Update particles with the motion model:
4:     $\theta_t^{[m]} = \theta_t^{[m-1]} + N(0, \Sigma) \Delta t$
5:     Calculate the belief of each particle with the measurement model:
6:     $w^{[m]} = h(\mathbf{F}_{\varphi^1}, \ldots, \mathbf{F}_{\varphi^N} | \theta_t)$
7: end for
8: for $m = 1$ to $M$ do
9:   draw $m$ from $\Theta_t$ with probability $\propto w^{[m]}$
10: add $\theta_t^{[m]}$ to $\Theta_{t+1}$
11: end for
12: return $\Theta_{t+1}$

The first line initializes particles each of which represent a
set of joint angles of the robot, $\theta_t^{[m]}$ indicate the $m$th particle
with a set of joint angles of robot. The particles are updated
based on a motion model of the robot at the third line. The
paper assumes that the robot moves its joint angles at random.
Subsequently, it calculates likelihood based on a measurement
model with Eq.(5).

\[
h(\mathbf{F}_{\varphi^1}, \ldots, \mathbf{F}_{\varphi^N} | \theta_t) = \exp \left( -\frac{(1-s)^2}{\sigma} \right) \tag{5}
\]

where $s$ indicates similarity calculated from Eq.(4). Then,
resampling process is executed to remove low weight particles
and increase particles with high weights.

IV. LEG JOINTS CONTROL FOR STANDING

Although the particle filter can estimate appropriate joint
angles for the imitating robot from the observation of human
demonstration in real-time, it does not take standing stability
into account. Actually, the robot often falls down if it uses
estimated angle directly. In general, a human being controls
his/her leg joints unconsciously to keep standing stability
during motion, for example, squatting up and down. For the
standing stability, we propose a simple strategy for controlling
the hip and ankle joints to provide a reasonable standing
stabilization so that it keeps the center of mass within the
supporting polygon in real-time. We assume the surface is
horizontal and flat. The assumption is reasonable because arti-
ficial buildings are designed to have horizontal and flat floor.
Even if the floor is not horizontal or flat, an external process
for measuring the floor detects the incline and unevenness of
(a) Joints angle of pitch in lower-body
(b) Joints angle of roll in lower-body

Fig. 5. Definition of joint angles controlled for standing stability

V. EXPERIMENT FOR STABILITY EVALUATION

Figure 6 shows the experiment set-up of the real-time imitation. One laptop computer is connected to the Kinect and it estimates an appropriate set of joint angles of the humanoid robot and sends it to the robot over a network cable. Its CPU is 2.5GHz dual-core Intel Core i5 and its memory is 8GB 1,600MHz DDR3 SDRAM. The number of particles is set to 400 in the experiments. The sampling is 30 Hz.

We conduct experiments in three motions including bowing, knee bending and hip swaying motions to evaluate the validity of the proposed method. The robot was demonstrated on a flat surface. A demonstrator shows a motion in front of Kinect and the robot imitates the motion in real-time. Figures 7, 8 and 9 show examples of sequence of the demonstrated motions, bowing, knee bending and stretching, and hip swaying, respectively. Figures 10, 11 and 12 are the typical motions of bowing, knee bending, and hip swaying imitated by the NAO motion, respectively. The time lag is caused by the convergence of the particle filter, communication speed and so on. The time lag range is within 0.5 seconds on average. The time lag is not so serious big as the demonstrator does not feel difficulty.

In order to evaluate the stability of the robot, the center of mass (hereafter, CoM) is measured during the imitation while the demonstrator takes 5 times at each motion. Figure 13 shows the sequence of the CoM position during the imitation. The figure also shows the boundary of CoM position between the standing and falling down. The boundary gives the insight to evaluate the stability of the robot’s posture. The boundary is measured by hand as follows. We change NAO’s CoM position by changing NAO’s ankle joint in steps of 5[deg], and then when NAO falls down, we regard the center-of-mass position as the boundary of keeping the posture. As the result, the boundary of forward CoM position is 0.0504 [m], backward is -0.0357 [m], left-side is 0.0528 [m] and right-side is -0.0536 [m]. Here, the origin is the center of between left and right feet. If the robot keeps its CoM position in the range of the bounds during the imitation, the robot avoids falling down. In the bowing motion (Fig.13(a)) and the knee bending motion (Fig.13(b)), the CoM position moves in the...
back and forth (x-axis) direction. The CoM position moves in the side (y-axis) direction in the swaying motion (Fig.13(c)). The blue ranges in the graphs are the region of keeping the posture without falling.

In the bowing motion, NAO’s CoM position is within the blue area, and the CoM position moves smoothly. In the knee bending motion, NAO’s CoM position does not move significantly, but the movement of CoM is not smooth. It seems that the Kinect sensor has larger errors in back and forth direction compared to the side direction and does not detect the demonstrator’s joint position with good precision. Moreover, the depth sensor in Kinect is not able to measure the data behind the obstacle. The hip and torso joints are hidden and they cannot be detected when the demonstrator is knee bending. As the result, the particle filter does not provide good estimation, unfortunately. In the swaying motion, y-coordinate of CoM position is within the blue area, and the movement is smooth. However, there is a rapid change at about 8 seconds in one of the 5 trials. The NAO stands on one leg in this time because NAO’s ankle joint reach to the limit, and it causes the rapid change of CoM position. The whole of the results shows that our proposed method makes NAO standing stable enough to imitate the demonstrator in real-time although a small time lag and unstable estimation are found.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel method which compensates standing stability of an imitating biped humanoid robot during imitating the motion of demonstrator. A particle filter provides reasonable joint angles of the humanoid robot to imitate the human demonstrator posture while it absorbs gaps between the robot and the demonstrator. The proposed simple leg joint control, which is similar to a reflex motion of animals, ensures the standing stability with small computational resources. The proposed method is evaluated using a
Fig. 13. CoM position of the NAO during imitation: In case (a) the bowing motion and (b) the knee bending motion, the movements of CoM in back and forth (x-axis) direction are shown. In case of (c) the swaying motion, the movements of CoM in side (y-axis) direction are shown.

humanoid robot, NAO, and shows its validity in imitating the demonstrator’s posture without falling down in real-time.

One of the future work is considering an inclining and/or uneven surface. The proposed method can be extended to the inclining and/or uneven surface. Second, we should consider a combined model of joint angle estimation with stability condition in a one comprehensive state space model. We would like to introduce the state space model to improve this method in estimation step of the particle filter. Additionally, handling the single support leg should be taken into consideration. The single support leg control is different from the one of the double support legs, therefore, it is necessary to handle the two modes continuously. Imitation with the single support leg is necessary for more complex motion such as walking and kicking.

REFERENCES


