

Kinect-Based Humanoid Robotic Manipulator for Human Upper Limbs Movements Tracking

Mohammed Z. Al-Faiz¹, Ahmed F. Shanta²

¹College of Information Engineering, Al-Nahrain University, Baghdad, Iraq

²Department of Computer Engineering, Al-Nahrain University, Baghdad, Iraq

Email: mzalfaiz@ieee.org, AhmedSlauf@gmail.com

Received 19 December 2014; accepted 2 January 2015; published 15 January 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper presents a humanoid robotic arms controlled by tracking the human skeleton movement in real-time using Kinect upper limbbody tracking. Using Kinect tracking algorithm, the positions of upper limb arms of the body to the wrist in 3D space can be estimated by processing depth images from the Kinect. An extraction of 3D co-ordinates of the user's both arm in real-time then Arduino microcontroller is transferring the data between both of computer and the humanoid robotic arm. This method provides a way to send movement task to the humanoid robotic manipulator instead of sending the end position motion like gesture-based approaches and this method has been tested in detect, tracking and following the movement of human skeleton gesture. Designing complete prototype of a humanoid robotic arms with 4DOF three joints in shoulder and one elbow joint to the wrist that look like the Human arm Structure, Appearance and Action that represent human arm movement performed by the humanoid robotic arm. The error was and response time result generated is small (less than 4.6% and 105 ms).

Keywords

Humanoid Robotic Manipulator, Human Arm, Kinect

1. Introduction

In these days, the Robots can be helpful as possible and be able to assist humans in their everyday activities, it is proposed that they should have a human-like structure and behavior. In order to achieve such a tremendous target, Human Computer Interaction (HCI) and Human Robot Interaction (HRI) have to work closely together so

as to allow the users to be able to interact with the robot's behavior in the easiest way possible, both to program the robot's functionalities as well as to control the triggering of these functionalities. When the robot is in a dangerous environment, robot human controlling may be necessary [1]. The idea of robots being able to have the abilities of a human brain is not entirely accepted. This is due to the possibility that, if robots have proper intelligence, they might turn against the human race and take over. Scientists take this into account and tend to program robots in such a way that the main target is to assist humans and make everyday life easier. Some human-robot interfaces like joysticks, dials and robot replicas, have been commonly used, but these contacting mechanical devices require unnatural arm motions to complete an operation task [2] [3]. Another way to communicate complex motions to a robot, which is more natural, is to track the operator arm motion which is used to complete the required task using contacting electromagnetic tracking sensors, inertial sensors and gloves instrumented with angle sensors. However, these contacting devices may hinder natural human-limb motions [4]-[6].

This paper presents a method of humanoid robotic arms controlled by tracking human upper body limb movement using Kinect-based 3D coordination arms to the wrist. Kinect arms tracking is used to acquire 3D skeleton position, and then it sends the data to the humanoid robotic manipulator by an Arduino interface to enable the robotic manipulator to copy the operator's arms motion in real-time. This natural way to communicate with the robot allows the operator to focus on the task instead of thinking in terms of limited separate commands that the human-robot interface can understand like gesture-based approaches [1]. Using the Kinect tracking device, **Figure 1** avoids the problem that physical sensors, cables and other contacting interfaces may hinder natural motions and that there may be marker occlusion and identification when using marker-based approaches [1].

2. Human Arm Tracking and Positioning System

Human arm detection and tracking is carried out by continuously processing depth images of an operator who is performing the arms motion to extract 3D coordination and sent it to the humanoid robotic manipulation. The RGB images and depth images are captured by the Kinect which is fixed in the front of the operator [1]. The Kinect has infrared projection that collaborate with infrared camera for depth detection and one RGB vision cameras.

3. Inverse Kinematics Model

Consider the human arm shown in **Figure 2** are modeled as a total 4-DOF, which the shoulder joint is represented by three intersecting revolute joint and elbow joint is represented by one revolute joints. From the D-H parameters it can get the four transformation matrices which transfer the movement from joint to another and they are denoted by (T_{i-1}^i) where $(i = 1, \dots, 4)$ **Table 1** gives the D-H parameters for human arm with $d_1 = 25 \text{ cm}, d_2 = 25 \text{ cm}$.

The concept of inverse kinematics refers to the method of calculating the appropriate movements that a body needs to move from a point A to B . In this paper, the robotic arm is required to the observed posture of the user. As a human arms is made up of joints and bones, the joints have been treated as "start" and "end" points whereas the bones have been treated as vectors.

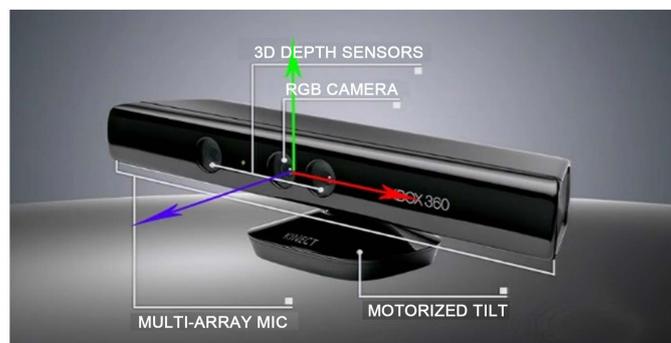


Figure 1. The Kinect device by Microsoft with the Kinect or camera reference frame.

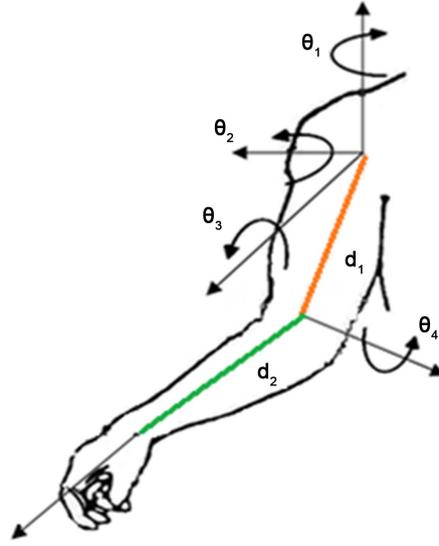


Figure 2. Human arm 4 DOF model.

Table 1. Numeric value for D-H parameters of human arm.

Frame	α_i (rad)	a_i (cm)	θ_i (rad)	d_i (cm)
1	$\frac{\pi}{2}$	0	θ_1	0
2	$-\frac{\pi}{2}$	0	θ_2	0
3	$\frac{\pi}{2}$	0	θ_3	d_1
4	$-\frac{\pi}{2}$	0	θ_4	0
5	$\frac{\pi}{2}$	0	θ_5	d_2

On the left side of **Figure 3** is illustrated where the Kinect is tracking a user's both arms. Points A , B and C represent the tracked shoulder, elbow and wrist joints respectively. As it can be seen on the right side of **Figure 3**, the joints A and B have been treated as "start" and "end" points respectively which have now formed a vector $\rho(AC)$. Using simple trigonometric methods, the angles $\theta_1, \theta_2, \theta_3$ and θ_4 are calculated. The equations to obtain the magnitudes of $\rho(AC)$, angles equations given below:

Step 1: Finding vector AB , BC and AC .

$$AB = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Step 2: Finding $\theta_1, \theta_2, \theta_3$ and θ_4 angles between the vectors as following:

$$\theta_1 = \sin^{-1} \left(\frac{AB_x}{\sqrt{AB_x^2 + AB_y^2 + AB_z^2}} \right) \quad (2)$$

$$\theta_2 = \sin^{-1} \left(\frac{AB_y}{\sqrt{AB_x^2 + AB_y^2 + AB_z^2}} \right) \quad (3)$$

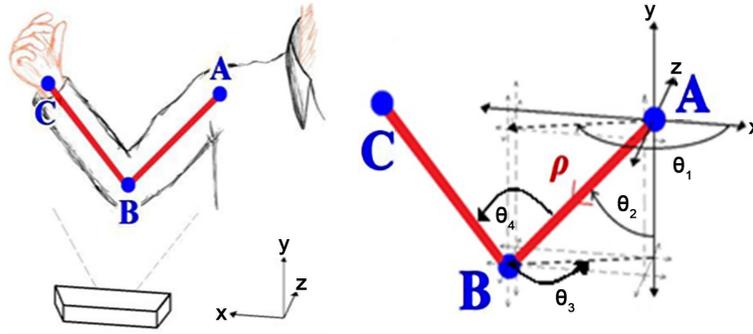


Figure 3. Geometric inverse kinematics.

$$\theta_3 = \sin^{-1} \left(\frac{BC_x}{\sqrt{BC_x^2 + BC_y^2 + BC_z^2}} \right) \quad (4)$$

$$\theta_4 = \cos^{-1} (AC^2 - AB^2 - BC^2) \quad (5)$$

Those functions has been implemented where the X, Y and Z coordinates of the joints A, B and C are passed as arguments and the vector AB , BC and AC values are returned. The function is then passed the vectors and calculate requires angles. This function is run on every frame (image) received, *i.e.* every 0.03 s as the system runs at 30 FPS. For every frame, the angles values are then sent to the Arduino using serial communication so as for the robot to move accordingly.

4. Designing Robot Manipulation System

The humanoid robotic arm was designed to have four degrees of freedom from shoulder up to and including the forearm for each arms. Each degree of freedom has been represented an angle of movement that can be performed by the human arm. A servo motor has been used to implement each angle of movement ($\theta_1 \rightarrow \theta_4$). As the shoulder joint feature 3D motion and elbow joint feature 1D motion, pan and tilt mechanisms [7] have been implemented so as to allow servo motors at each joint to perform the required motion. The arm's bones have been summed to two parts, the upper arm (shoulder to elbow) and the lower arm (elbow to wrist). As it can be seen in the complete humanoid robotic arm design **Figure 4**. The upper and lower arm parts have been represented and printing by using 3D printer. So as to represent the human body, a rectangular chest shape (center pillar) has been used to elevate the arm to a height of 460 mm from its base. The chest had been design so that Servo A can fitted exactly in the hole inside the chest indicating the first joint, the shoulder. All servo motors have been fitted to the plastic parts by using bolts and nuts to allow future modifications while maintaining consistency through the structure. Finally overall system was implement as shown in **Figure 5**.

5. Experiments

To evaluate the Kinect and humanoid robotic arms operation algorithm described in this paper, we use Processing language to develop a humanoid robotic interface system and this system is used to operate the a four-axis humanoid robot. This experimental system includes:

- 1) Use the human arm tracking and following system to get the arm skeleton 3D positions arm points, and then calculate the required Invers kinematic angles of the arms.
- 2) Humanoid robotic manipulation system based on the joint angles which are calculated through inverse kinematic and then they will be transmitted to the real robot to control it.

In the experiment, the operator stand in front of the Kinect to control the robotic arm. After the angles been calculated by invers kinematics algorithm the data are send by USB cable to the robot through an Arduino interface microcontroller and measure values can be taken from potentiometer sensor build inside the servo motors.

Arduino communication with pc without any delaying to ensure that data is correct functionality. Some jittering and excessive oscillations in the structure of humanoid robotic arm encourages oscillations due to the arm

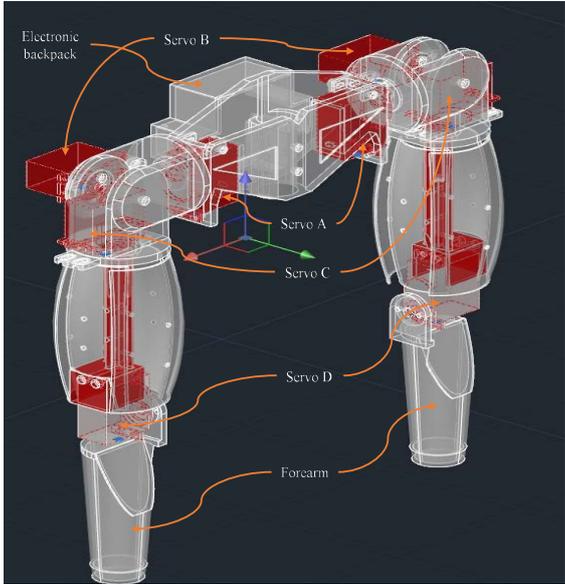


Figure 4. Humanoid roboticmanipulators.

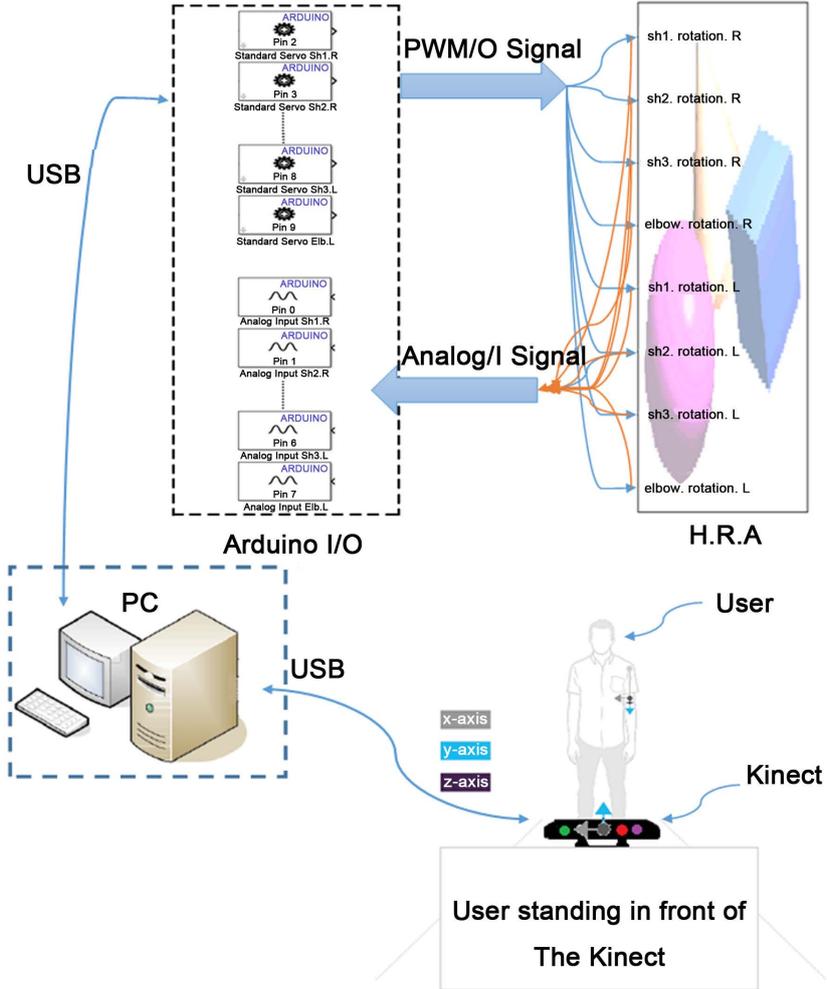


Figure 5. System architecture.

being suspended only from servo A without any additional support. Jittering is present due to the hypersensitivity of the skeletal tracking algorithms. Controlling and error tolerance servos feature an embedded PID controller proved to work efficiently and effectively minimizes rotational errors and overshoots. By the structure consisting of only four servos and embedded PID controllers on each individual servo, the control of the robotic arm is straightforward and robust. The use of a single medium used between the Windows PC and the servos (*i.e.* the Arduino), allows low latency times between the time of user input and the time of output performed by the robotic arm.

The quality of output performed by the humanoid robotic arm is justified by its similarity to the movement of that of the human arm. Fluidity and human like smoothness is required so as for the developed structure to be considered humanoid. **Figure 6** shows the entire system data flowchart and how information is passing through every component. Even though the actuators consist of pre-calibrated built in PID controllers, the output movement can be characterized to have a fast settling time but yet a critically damped approach to the given settling point.

The fluctuations visible within the structure refer to a poor construction link between servo A and servo B. This is due to the fact that the pan and tilt mechanism (containing servo B and the rest of the arm), connects to servo A by screwing on the servo's shaft. As the diameter of the servo shaft is small (5 mm), it fails to keep the arm in a sturdy posture but still allowing full control of the arm's movements. The rest of the arm construction comprises of solid connections between the servos themselves as well as tight fittings between the servos and the chassis.

6. Result

Next ten cases were used to test the developed system controller in different posture **Figure 7** from 1-10 positions, the result in **Table 2**, the table contains in the first column the posture number according to **Figure 7**, the second column contains the desired angles (calculated angles by IK Equation (1)-(5)) in degree, the third column contains measure, actual angles that the robotic arms move to in degree. The respond time was calculated as the time required to completed desire posture position. The error percentage was calculated as Equation (6). **Figure 8** shows the position of the robot's end-effector and the operator's arm during operation experiments for posture number 7 in **Figure 7** and **Table 2**, the dashed red line represents the end-effector's path. The solid blue line represents the path of the operator's arm, the values in the Seventh position in **Table 2** was taken for the worse result represent by a point in **Figure 8** for each angle.

$$e_g \% = \sqrt{\Delta\theta_1^2 + \Delta\theta_2^2 + \Delta\theta_3^2 + \Delta\theta_4^2} \quad (6)$$

Table 2. Result of arm movement.

Position	Desired Angles $[\theta_1, \theta_2, \theta_3, \theta_4]$	Measure Angles $[\theta_1, \theta_2, \theta_3, \theta_4]$	Error Percentage
1	[87,139,102,38]	[88,144,100,38]	5.47%
2	[85,171,173,10]	[86,171,173,11]	1.41%
3	[165,101,93,11]	[160,100,97,10]	6.55%
4	[93,86,80,14]	[90,85,80,14]	3.16%
5	[20,109,51,82]	[20,110,54,82]	3.16%
6	[79,151,70,84]	[76,145,69,84]	6.78%
7	[59,97,84,81]	[52,94,86,81]	7.87%
8	[13,102,94,17]	[16,101,94,17]	3.74%
9	[95,170,87,88]	[95,173,85,88]	3.60 %
10	[165,102,90,12]	[163,105,90,13]	3.74%

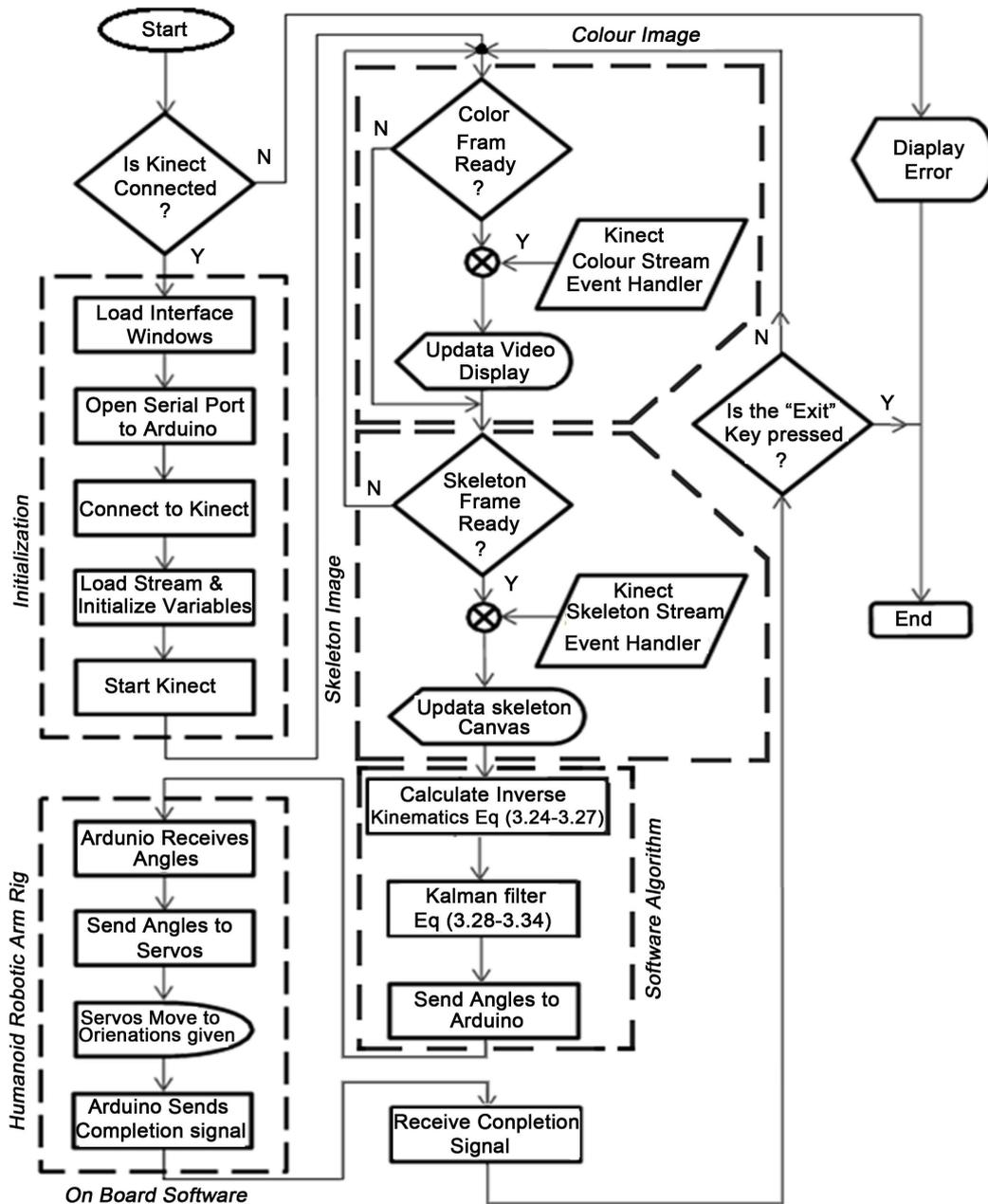


Figure 6. Data system flowchart.

7. Conclusions

The aim of designing such a structure was to replicate the movement of the human arm (from the shoulder down to the wrist joint). Such movement is capable and proven to be capable by the developed humanoid robotic arm through the extensive testing carried out. The possible scenarios were chosen to investigate the accuracy and validity of the skeletal tracking algorithms performed by the Microsoft Kinect sensor.

This work proves that 3D humanoid robot control does not necessarily require the user to utilize any specialized devices attached to their bodies so as to perform 3D tracking of their movements. In addition, this work allows comprehension of the function of the Microsoft Kinect sensor and analyses the extent to which this may be used for general Human Robot Interaction purposes.

The human skeletal tracking algorithms appear to be operating normally but during some of the testing

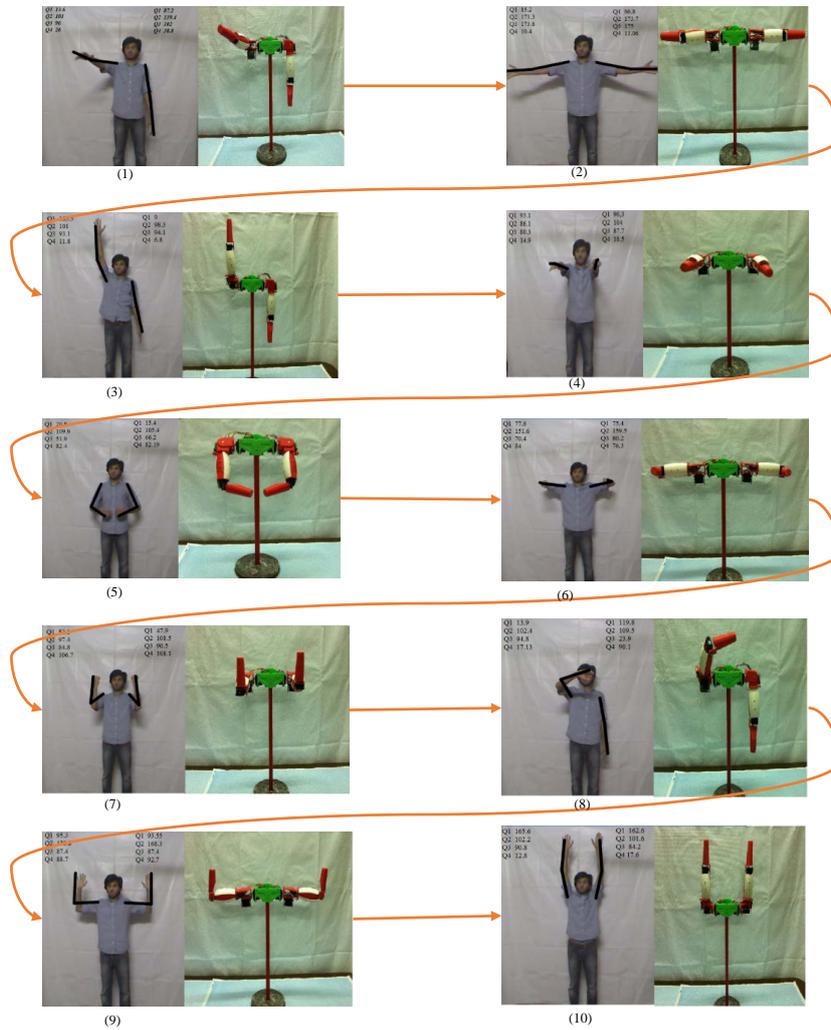


Figure 7. Cases study.

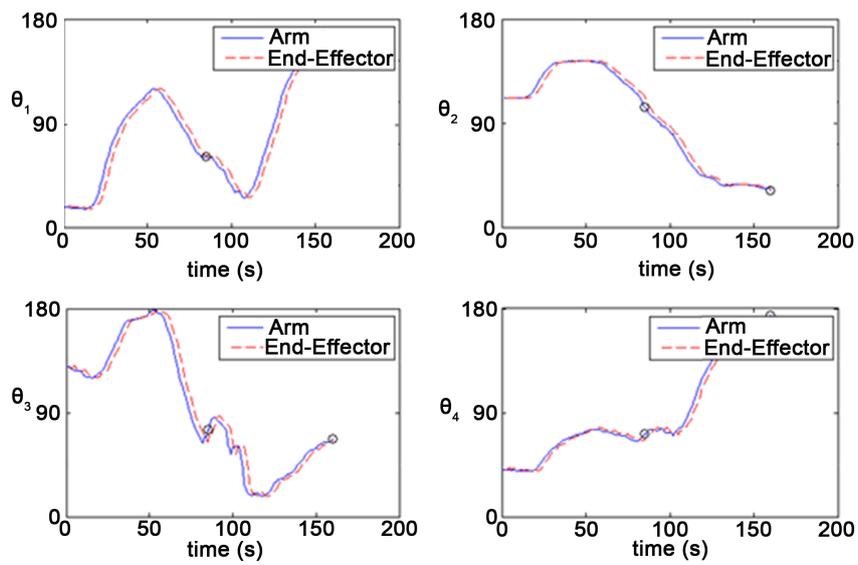


Figure 8. Analysis of the experiment for position 7.

undertaken, weaknesses have appeared. When the user moves their arm in front any other part of the body (e.g. other arm, main body, head or legs), the accuracy of the tracked joints drops significantly, causing the system to perform erratically indicating system instability in these cases. The algorithm seems to be affected when tracked joints get close together or overlap. Such an issue can be considered as one of the main weaknesses of the system. Even though such events are only experienced in extreme cases, they are still possible to occur and cause instabilities in the system.

To complete the complex tasks, multi-Kinect will be used to work together, more DOF, full size humanoid and wireless communication in future work.

References

- [1] Du, G.L., Zhang, P., Mai, J.H. and Li, Z.L. (2012) Markerless Kinect-Based Hand Tracking for Robot Teleoperation. *International Journal of Advanced Robotic Systems*, **9**, 36.
- [2] Yussof, H., Capi, G., Nasu, Y., Yamano, M. and Ohka, M. (2011) A CORBA-Based Control Architecture for Real-Time Teleoperation Tasks in a Developmental Humanoid Robot. *International Journal of Advanced Robotic Systems*, **8**, 29-48.
- [3] Mitsantisuk, C., Katsura, S. and Ohishi, K. (2010) Force Control of Human-Robot Interaction Using Twin Direct-Drive Motor System Based on Modal Space Design. *IEEE Transactions on Industrial Electronics*, **57**, 1338-1392. <http://dx.doi.org/10.1109/TIE.2009.2030218>
- [4] Hirche, S. and Buss, M. (2012) Human-Oriented Control for Haptic Teleoperation. *Proceedings of the IEEE*, **100**, 623-647. <http://dx.doi.org/10.1109/JPROC.2011.2175150>
- [5] Villaverde, A.F., Raimundez, C. and Barreiro, A. (2012) Passive Internet-Based Crane Teleoperation with Haptic Aids. *International Journal of Control Automation and Systems*, **10**, 78-87.
- [6] Wang, Z., Giannopoulos, E., Slater, M., Peer, A. and Buss, M. (2011) Handshake: Realistic Human-Robot Interaction in Haptic Enhanced Virtual Reality. *Presence-Teleoperators and Virtual Environments*, **20**, 371-392. http://dx.doi.org/10.1162/PRES_a.00061
- [7] Xu, D. and Acosta, A. (2005) An Analysis of the Inverse Kinematics for a 5-DOF Manipulator. *International Journal of Automation and Computing*, **2**, 114-124. <http://dx.doi.org/10.1007/s11633-005-0114-1>

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or [Online Submission Portal](#).

