

Realistic Human Motion Preservation-Imitation Development on Robot with Kinect

Abdul Muis¹, Wisnu Indrajit²

Electrical Engineering Department, University of Indonesia, Depok, Indonesia
Kampus UI Depok, Indonesia
e-mail: muis@ui.ac.id¹, wisnu.indrajit@ui.ac.id²

Abstrak

Umumnya, urutan gerakan robot diperoleh dengan perhitungan yang kompleks secara off-line ataupun pendekatan langsung. Dalam pendekatan langsung, robot digerakkan sampai pada posisi dan orientasi tertentu dengan langsung menggerakkan robot secara manual dengan tangan atau melalui panel kendali. Posisi selanjutnya di simpan di memori setelah mencapai posisi yang di inginkan. Namun, teknik ini sangat membutuhkan waktu lama. Adapun teknik yang mudah dan cepat adalah dengan meniru gerakan objek melalui sensor untuk di implementasikan langsung ke robot. Beberapa riset mengatasi hal ini dengan menggunakan position sensitive detector (PSD) ataupun dengan kamera stereo. Namun untuk melakukan hal ini diperlukan set-up awal yang cukup rumit dan tidak cocok untuk merekam gerakan alami. Karenanya, paper ini mengembangkan pelestarian gerak dengan menangkap gerakan tubuh manusia secara alami melalui Kinect. Selanjutnya gerakan yang terbaca direproduksi pada robot humanoid secara bersamaan. Selain itu, gerakan tersebut juga di rekam dalam database untuk menggerakkan robot sesuai gerakan manusia maupun mengajarkan manusia seperti gerakan tari di kemudian hari. Dalam paper ini juga dikembangkan gerakan robot yang lembut serta mendekati kemampuan mata membaca perubahan gerakan. Metoda yang diusulkan telah di validasi dalam eksperimen dengan menangkap dan mereproduksi gerakan manusia pada robot dengan kecepatan 20Hz. Dimana waktu komputasi untuk setiap proses sekitar 340us.

Kata kunci: dynamixel, imitasigerak, kinect, ROS

Abstract

At most, motion generation on robot is usually done through complex computation in off-line mode and straightforward method. In straightforward method, the operator drives robot to certain pose either with moving manipulator tool-tip with hand or remotely. Once the desired pose achieved, the current pose is saved to memory. However, these methods are time consuming. An easy and quick approach is by imitating an object motion to robot with sensing devices. There have been numerous efforts for motion imitation either by using position sensitive detector (PSD) or by using stereo camera. However, a calibrated pre-condition should be done initially, which is not possible for natural movement. Here, this paper proposed motion preservation by capturing human motion naturally through Kinect and then reproduced human motion on humanoid robot simultaneously. In addition, the motions are also preserved in database for later used on robot motion generation and teaching as well. Furthermore, the robot motions are developed to run smoothly and close to human eye ability. The proposed method has been validated in experimental results by capturing and reproducing human motion on robot in rate of 20Hz with 340us computation cost for each process.

Keywords: dynamixel, kinect, motion imitation, ROS

1. Introduction

The need for human motion preservation especially in Indonesia is unavoidable. There have been numerous claims on National traditional dances by other country while teenagers are starting to loose appetite in learning national cultures such as traditional dance. For that purpose, there have been national robot competitions on national dancing performance since three years ago. Here, the dancing should be carried out by humanoid robots since this kind of robot is more similar to human being. Also, interesting challenge in humanoid robot is how to design trajectory control for each joint. Obviously, the robot should have generated motion in memory.

There have been many ways to generate robot motion. For industrial need, robot motion is generated through complex computation to achieve high precision. This is done in off-line mode through engineering software. Other method uses straightforward approach. Here, the operator drive robot to certain pose either by moving manipulator tool-tip with hand or remotely. Once the desired pose achieved, the current pose is saved to memory. However, these methods are time consuming especially for long-duration motion. Each pose may takes in minutes. Thus, these methods are not effective for more than hundred poses. In spite of that, an easy and quick approach is by imitating an object motion to robot with sensing device, or often called motion capture [1-3]. There have been numerous efforts for motion imitation either by using position sensitive detector (PSD) or by using stereo camera. Unfortunately, initial preparation is required such as mounting detector on object along with cabling or calibrating camera position for each measurement. This is not possible for natural movement.

Programming by demonstration, or also known as imitation learning or motion capture is an answer to a problem how to control humanoid robot joints. This method offers flexibility and easy modification of the robot control. Humans as the actor would make a move, and then the robot as an agent will perform the movements performed by humans. The system is very useful to train the robot to make movements and in the future can be used to create a natural human-like motion of humanoid robot motion that can do housekeeping, dangerous work, and even entertain humans [4-8].

Some researches on robotic imitation previously have been done among others. Hsin-yu et.al developed human motion imitation to humanoid robot through 13 mark points with webcam [2]. Nakaoka joint research group also performed traditional dances imitation to robot [9]. Eight cameras were used to obtain three dimensional informations of 30 mark points on actor. Similar research was done through markless tracking by Marc Rosanes to imitate the human arm movement on the Bioloid robot with TOF capture device Camera: Swiss Ranger 3000 [10]. Generally, all of those were using multi-camera as motion capture devices, which is too costly either in hardware and computation process due to complex algorithm. Moreover, preparation for environment, mark points and multi camera calibration should be done initially before capturing human motion.

In this paper, realistic motion preservation by capturing human motion through Kinect is proposed. Kinect that usually used along with Xbox games has a stereo camera system. Jing Tong et.al have done scanning 3D full human bodies with Kinect [3]. However, capturing full human bodies parts are not necessary for human motion which is time consuming. With certain library, some of human ankles are captured and extracted along with depth value. Later, the captured human pose is translated to quaternion form of each human joint. This is done naturally without marking human body. This paper used Robot Operating System (ROS) as work platform. In addition, the extracted human motion is preserved in database for later used on robot motion generation and teaching as well. In order to show the imitation ability, an upper part body with 4DOF on each hand are developed. Each DOF is driven with Dynamixel AX-12 servo. The rest problem was to realize realistic motion on these servos through parameter adjustment on servo and modification on ROS library. Here, the operation process is extended to run in rate 20Hz, which is close for human eye ability up to 25Hz.

The rest of this paper is organized as follows; section 2 describes the proposed method and systems used to realize the realistic motion preservation and imitation. Later, tracking process, mapping process and database design are explained in section 3. Afterwards, section 4 described the experimental results along with analysis. Finally, concluded in section 5.

2. The Proposed Method

In order to realize the realistic motion preservation and imitation, we proposed markless tracking human motion with Microsoft Kinect and inverse kinematic method to mapping human arm motion to humanoid robot. All process of tracking and mapping worked in Robot Operating System (ROS). Here, we describe the details of robot, Kinect and ROS.

2.1. Humanoid Robot

The used robot in this paper is humanoid robot consisting of two arms with each arm has four degrees of freedom with dynamixel AX-12 actuators in each joint. The advantages of

this actuator is to have a feedback position, speed, voltage, temperature, and load. Here is a picture robot model and its arm movements that used in this paper.

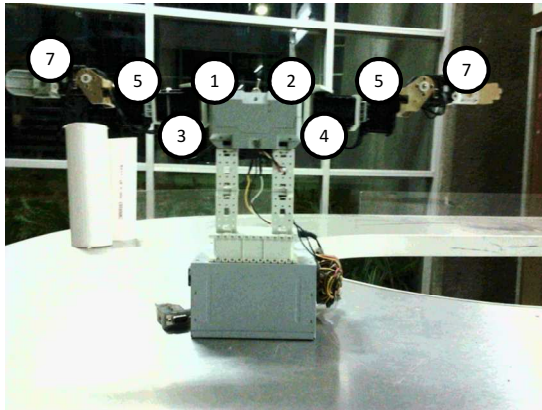


Figure 1. Robot Model and Dynamixel Position

Table 1. Dynamixel Positions and Movements

	Dynamixel AX-12 ID	Movement
Right Arm	1	Right Shoulder Fleksion - Extension
	3	Right Shoulder Abduction - Adduction
	5	Right Hand Internal Rotation
	7	Right Elbow Fleksion - Ekstension
Left Arm	2	Left Shoulder Fleksion - Extension
	4	Left Shoulder Abduction - Adduction
	6	Left Hand Internal Rotation
	8	Left Elbow Fleksion - Ekstension

Robot movement is regulated directly by the PC via USB2 Dynamixel. Identification of the configuration of the robot models will be described in the next section. As a source of power supply in this robot, we used a computer power supply.

2.2. Microsoft Kinect

Microsoft Kinect is an accessory for Microsoft's Xbox 360 platform. Kinect is motion sensing unit device that was developed to interpret human gestures. It has sensors connected to a motorized pivot based. This device has a RGB camera, depth sensor, and multi-array microphone. With these sensors, Kinect can provide the ability to capture full-body 3D motion, facial recognition and voice recognition. Kinect has a good performance, small dimensions, and open source libraries that are able to make Kinect used on multiple OS platforms (Windows, Linux, and Mac), and also kinect is cheaper than other capture device. In this paper, human motion will be captured using kinect with markless method. With the skeleton tracking software from the Open NI, three-dimensional vectors of the joints in humans are achieved.

2.3. ROS (Robot Operating System)

ROS (Robot Operating System) is a framework for the development of robot software. ROS was originally developed by the Artificial Intelligence Laboratory Standford, but in 2008 the further development carried out by Willow Garage with more than 20 institutions collaborating in the development of ROS. Currently ROS has supported the development of various robots.

ROS provides a standard OS (Operating System) such as hardware abstraction, low level device control, implementation of common functions of an OS, message-passing between processes and package management. This architecture is based on a graph in which a process is located on a node that can send and receive multiplex sensors, controls, knowing the state, actuators, and other posts. Library of ROS-based UNIX systems (preferably Ubuntu Linux support, while other variations such as Fedora and Mac OS X is still a "experimental").

This paper developed a rob_mocap, ros-package that can perform real time saving features of imitation and repetition with the motion that has been imitated.

3. Research Method

There are three steps for doing imitation learning in this paper. First, we did human tracking to get important skeleton vectors. Second is mapping human motion to robot, joint position and velocity control. Finally, designed database motion, and then extract database to repeated movements that have been made.

3.1. Tracking Process

To track the human skeleton, `openni_tracker` node from stack `openni_kinect` in ROS is employed. This node will provide three-dimensional vector of 15 points on the human body. To perform tracking, someone needs to stand up and do "surrender" pose in front kinect (raise both hands). Here is pose calibration picture and 15 vector `openni_tracker` published from this node.

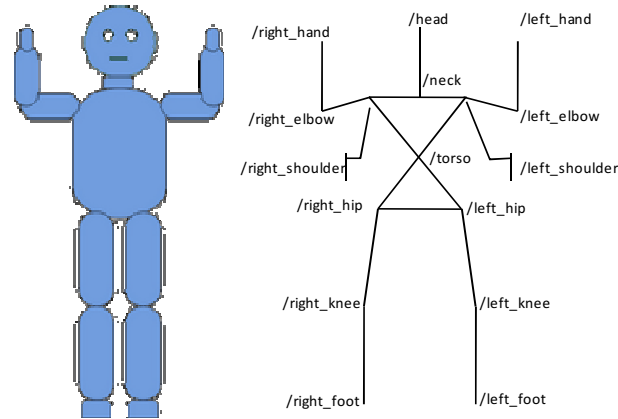


Figure 2. Calibration Pose and Vectors Published by `openni_tracker`

Important vectors used in the mapping process are `/right_shoulder`, `/left_shoulder`, `/right_elbow`, `/left_elbow`, `/right_hand`, and `/left_hand`. Combinations of these vectors are used to form the v_{sh} (shoulder-hand), v_{se} (shoulder-elbow), and v_{eh} (elbow-hand).

3.2. Mapping Process

In mapping process, firstly, kinematics model of the robot must be identified which used to find the inverse kinematic solution of the model later. The used model robot consists of two arms with each arm consists of four degrees of freedom. Configuration of the model robot in normal position (each joint at 0°) is shown in the figure below.

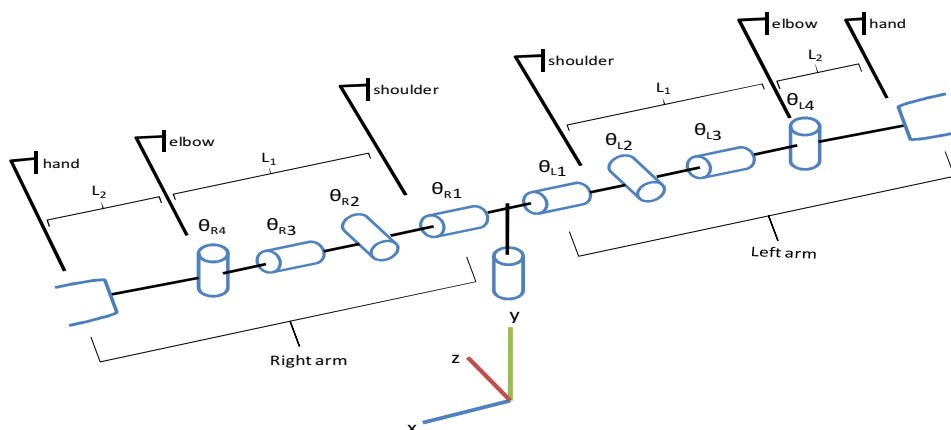


Figure 3. Robot Arm Configuration Diagram

Robot arm model compared with humans movements only have shoulder flexion-extension movement, shoulder abduction-adduction movement, internal rotation of the hand, and flexion-extension movement at the elbow.

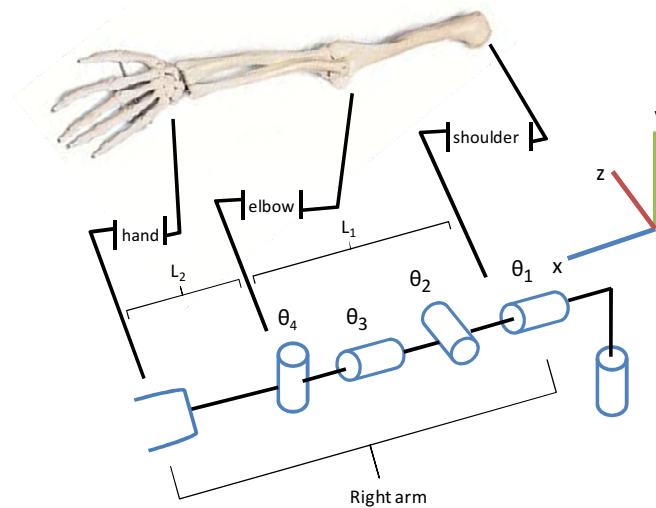


Figure 4. Comparison Robot and Human Right Arm

Figure 4 shows the kinematic chain of the robot's right arm. θ_1 , θ_2 , and θ_3 angles are the rotation angle of the shoulder. And then, θ_4 angle is the rotation angle of the elbow. Forward kinematic matrix G from the shoulder to the hand is obtained by multiplying the transformation matrix from T_1 to T_4 . With the matrix T is 4x4 dimension homogenous matrix transformations from a frame that represents the frame rotation by a 3x3 matrix R_g and frame translation is shown by the 3x1 dimensional matrix, t_g .

$$T_{4 \times 4} = \begin{bmatrix} R_g & t_g \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1 \text{Error! Bookmark not defined.})$$

From the model at Figure 4, each frame transformation matrix obtained as follows

$$T_1 = \text{Rot}_x(\theta_1) \quad (2)$$

$$T_2 = \text{Rot}_z(\theta_2) \quad (3)$$

$$T_3 = \text{Rot}_x(\theta_3) \cdot \text{transl}(L_1, 0, 0) \quad (4)$$

$$T_4 = \text{Rot}_y(\theta_4) \cdot \text{transl}(L_2, 0, 0) \quad (5)$$

So the transformation matrix from the shoulder to the hand is:

$$G = T_1 T_2 T_3 T_4$$

$$G = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{bmatrix} \quad (6)$$

Cartesian coordinates of the hand from the shoulder frame indicated by the elements g_{14} , g_{24} , and g_{34} in the matrix G . The vector v_{sh} can be expressed in the equation.

$$v_{sh} = \begin{bmatrix} L_1 c_2 + L_2 c_2 c_4 + L_2 s_2 s_3 s_4 \\ L_2 s_4 (c_1 c_3 - s_1 c_2 s_3) + L_1 s_1 s_2 + L_2 s_1 s_2 c_4 \\ L_2 s_4 (s_1 c_3 + c_1 c_2 s_3) - L_1 c_1 s_2 - L_2 c_1 s_2 c_4 \end{bmatrix} \quad (7)$$

In the tracking process with kinectwe got some important vectors (v_{sh} , v_{se} , and v_{eh}). So that, by use the forward kinematic equations above, we got the inverse kinematic solution which we used to transform human movement to robot. The steps to find the inverse kinematic solution presented as follows.

1) Find θ_4 Angle Solution

θ_4 is an elbow internal angle, which is the angle formed between v_{se} (shoulder_elbow) and v_{eh} (elbow_hand). θ_4 angle can be found by using the dot product vector equation. In Euclidean geometry, the angle formed between the two vectors is the dot product of vectors that expressed in an orthonormal basis related to the length and the angle formed between the vectors. The relationship is represented in Figure 5.

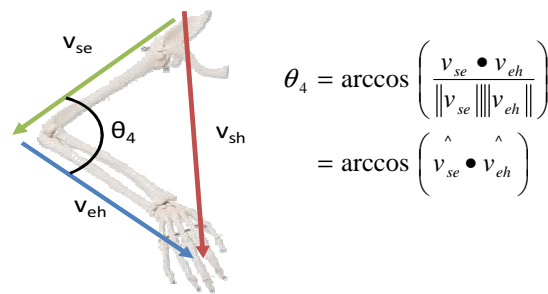


Figure 5. Elbow Internal Angle Solution

2) Find θ_1 and θ_2 Angle Solutions

To find solutions if angles θ_1 and θ_2 can be used forward kinematic equations from the shoulder to the elbow. The position of the elbow from the shoulder is the function of three first joints, ie.

$$T_1 T_2 T_3 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} v_{eh} \cdot x \\ v_{eh} \cdot y \\ v_{eh} \cdot z \\ 1 \end{bmatrix} \quad (8)$$

Expanding the matrix equation above will yields the following inverse kinematic solutions for θ_1 and θ_2 .

$$\theta_2 = -\arccos \left(\frac{v_{se} x}{L_1} \right) \quad (9)$$

$$\theta_1 = \arcsin \left(\frac{v_{se} y}{\sin(\theta_2)} \right) \quad (10)$$

3) Find θ_3 Angle Solution

θ_3 angle is the internal rotation angle of the hand. The solution of this angle can be found using the equation in xx after obtaining the solutions of θ_1 , θ_2 , and θ_3 .

$$\theta_3 = \arcsin \left(\frac{v_{sh} x - L_1 c_2 - L_2 c_2 c_4}{L_2 s_2 s_4} \right) \quad (11)$$

In order for the imitated movements in real time, it need to map each joint angular velocity. Speed control block diagram is shown in Figure 6.

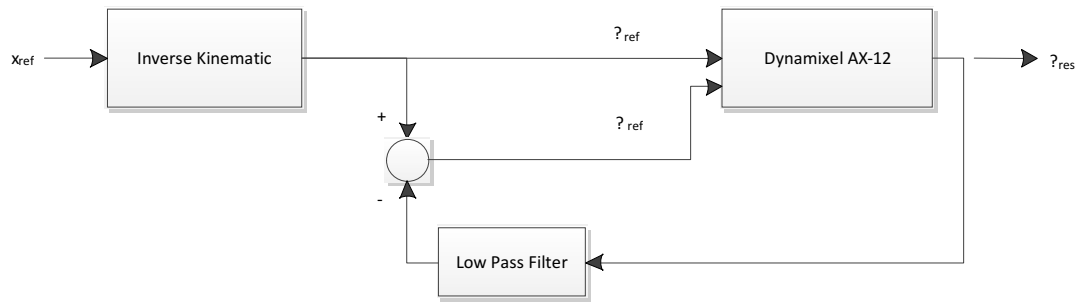


Figure 6. Position and Speed Control Block Diagram

3.3. Motion Database Design

Motion database is needed in order the robot can be able to repeat the results of imitation motions. In this paper, human motion is saved in a csv extension by writing in the following format.

0	1	2	3	4	5	6	7	8
time	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8
secs.nsecs	Goal Positon							

Figure 7. Motion Database Format

With that format, less memory is saved because there is no need to keep the speed memory of each joint. Velocity of each joint can be derived from the subtraction of the current and previous angle position divide by time difference.

4. Results and Discussion

4.1. Validation Invers Kinematic Solutions

There are 7 DOF in human arm (3 DOF in hand), to make robot motion looks like human natural motion at least there are 4 DOF in robot arm. In other research, some robots are only having 3 DOF in their arm. Error of validation invers and forward kinematic using 3 DOF in arm is so high. This section provides validation testing of inverse kinematic solution to test the equations that have been obtained previously. Testing is done by taking images of vectors in the following positions.

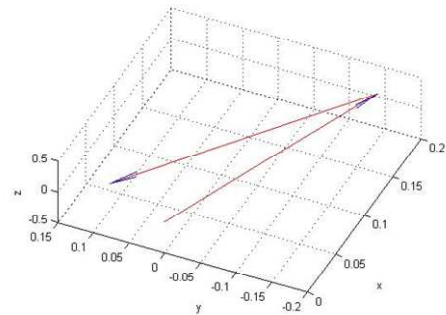


Figure 8. Testing Pose of Validation Inverse Kinematic Solution

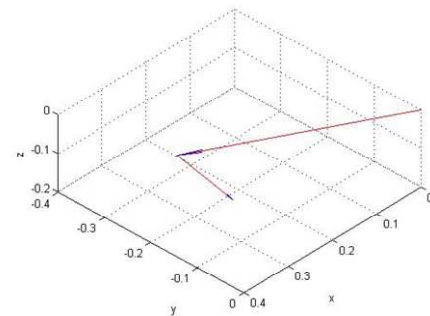
Once these vectors are obtained, the inverse kinematic solution is computed like the steps explained previously. And then, performed forward kinematic process and compared the results of the forward kinematic vectors which generated by the solution with reference vectors. The compared results show the errors of solution vectors and reference vector are 0%.

Table 2. Validation Inverse Kinematic Solution Result

	Position					
	1			2		
	x (m)	y (m)	z (m)	x (m)	y (m)	z (m)
Right Shoulder	0.0285	0.6977	2.8075	0.0187	0.6995	3.2847
Right Elbow	0.2088	0.5426	3.1497	0.2848	0.4293	3.1727
Right Hand	0.2534	0.6550	2.8475	0.5718	0.2988	3.0518
Vrse	0.1804	-0.1551	0.3422	0.2660	-0.2702	-0.1120
Vreh	0.0445	0.1124	-0.3022	0.2870	-0.1305	-0.1209
Vrsh	0.2249	-0.0427	0.0400	0.5531	-0.4007	-0.2329
Lrse	0.4167			0.3954		
Lreh	0.3255			0.3377		
Lrsh	0.2324			0.7216		
q1 (degree)	24.3821			-21.4327		
q3 (degree)	64.3640			42.3094		
q5 (degree)	82.8884			20.3791		
q7 (degree)	146.2656			20.3791		
Vrse (FK)	0.1804	-0.1551	0.3422	0.2660	-0.2702	-0.1120
%Error Vrse	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Vrsw (FK)	0.2249	-0.0427	0.0400	0.5531	-0.4007	-0.2329
%Error Vrsh	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000



a. Solution Vectors vs Reference Vectors in Position 1



b. Solution Vectors vs Reference Vectors in Position 2

Figure 9. Comparison of Solution Vectors (Red) and Reference Vectors (Blue)

4.2. Real Time Imitation

Process of real time imitation can be seen in Figure 10.



Figure 10. Real Time Imitation Result

The experiments were realized in 20Hz with single loop computation mapping process took about 340 μ s. Since the process is close to human eye ability, there is no significant different between human motion and generated motion on robot. Thus, the imitation looked real-time. The robot motion was done through inverse kinematic mapping process. In addition, some weakness conditions of multi-solution inverse processes were anticipated to make robot follow human motion. Mainly, the provided kinect capture library in ROS only support up to 5Hz.

4.3. Motion Record and Generation

The results of recording motion database and generated motion are shown in Figure 11. The recording motion data are shown in reference position (q_{ref}). While the generated motion data on robot are shown in result position (q_{res}). Although there is a slight delay in generated motion due to actuator response, those figures showed the robot was able to imitate the preserved motion. Mainly, the generated speed is limited to the actuator behavior. The utilized servo in this paper is driven through serial communication with 115kbps, where each command for each servo took almost 1ms. Here, having higher quality actuator would extend the motion generation performance. Nevertheless, generated motion on robot in this paper is sufficient since the generated motion still close to human eye ability. Also, the results are better than results as shown in references [1], [2], [9] since this paper used kinect with computation cost less than 0.5ms while others used multi cameras that took more computation cost on processing images.

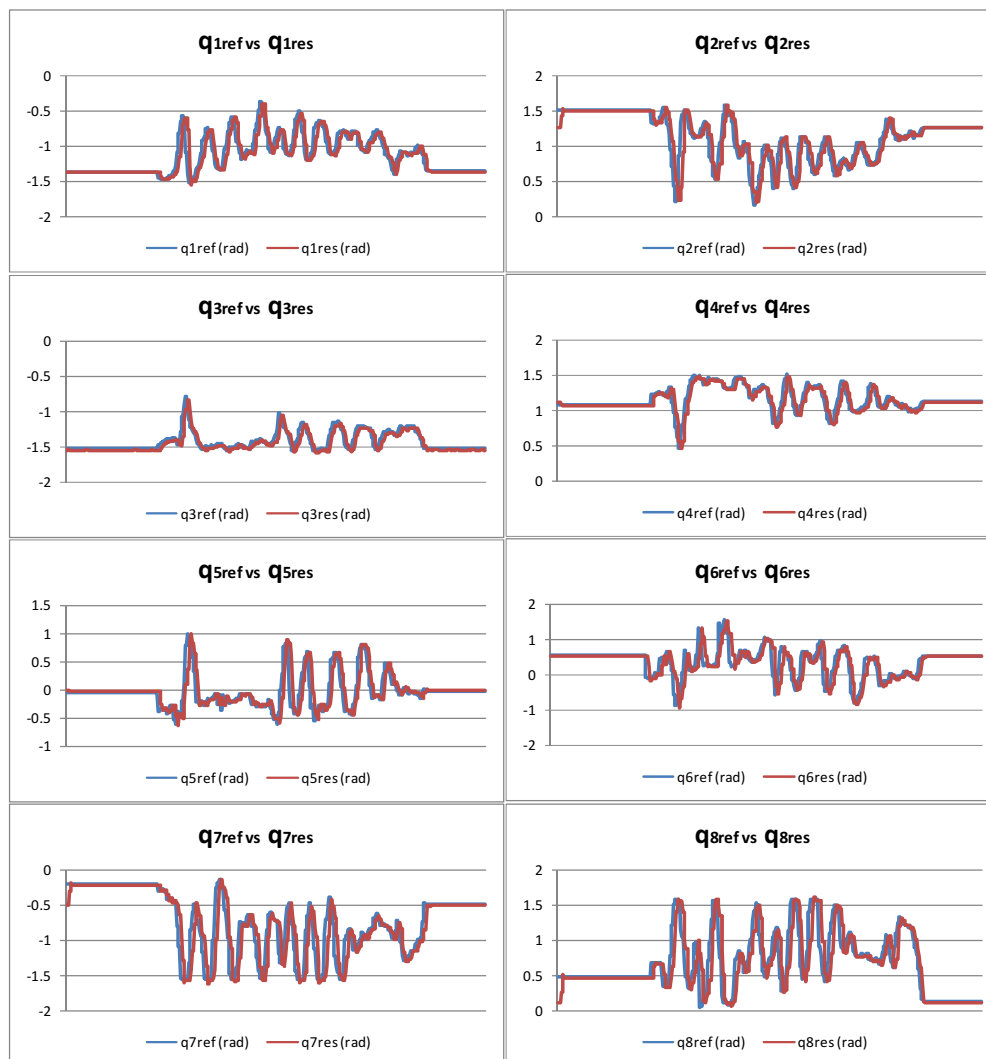


Figure 11. Extraction Database Result

5. Conclusion

A realistic motion preservation and imitation are realized in this paper. The preservation is done by capturing motion through Kinect and saved the extracted motion to database. At the same time, the upper part of humanoid robot imitated human motion. By tuning Robot Operating System, the proposed method has been validated in rate of 20Hz which is close to human eye ability. Here, the ROS rob_mocap package that has been developed, successfully solve the challenges of the imitation process as well humanoid robot capable of performing database extraction and storage and motion that has been done by humans. Also, extraction of inverse kinematic solutions, real time process of imitation, and the motion database extraction of the robot has shown good results.

Acknowledgment

This work is part of Autonomous Control Electronics (ACONICS) Research Group Project, Department of Electrical Engineering, University Indonesia.

References

- [1] Shon AP, Keith Grochow, Rajesh P.N Rao. *Robotic Imitation from Human Machine Capture using Gaussian Processes*. IEEE Int. Conf. on Humanoid Robots. 2005; 129-134.
- [2] Hsin-Yu Liu, Wen-June Wang, Rong-Jyue Wang, Cheng-Wei Tung, Pei-Jui Wang, I-Ping Chang. *Image Recognition and Force Measurement Application in the Humanoid Robot Imitation*. *IEEE Transaction on Instrumentation and Measurement*. 2012; 61(1): 149-161.
- [3] Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, Hao Yan. *Scanning 3D Full Human Bodies Using Kinects*. *IEEE Transaction on Visualization and Computer Graphics*. 2012; 18(4): 643-650.
- [4] Parul Gupta, Vineet Tirth, R.K. Srivastava. *Futuristic Humanoid Robots: An Overview*. IEEE Int. Conf. on Industrial and Information Systems. Srilanka. 2006; 247-254.
- [5] MZ Al-Faiz. *Analytical Solution for Anthropomorphic Limbs Model (IK of Human Arm)*. IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009). Kuala Lumpur. 2009; 684-689.
- [6] K Saenko, S Karayev, Y Jia. *Practical 3-D Object Detection using Category and Instance Level Appearance Models*. International Conference on Intelligent Robots and Systems. 2011.
- [7] R Zhang, Wang Y. *Research and Implementation from Point Cloud to 3D Model*. 2nd International Conference on Computer Modeling and Simulation. 2010.
- [8] Craig, John J. *Introduction to Robotics: Mechanics and Control*. USA: Addison Wesley Longman, 1989.
- [9] Nakaoka, Shinichiro. *Generating Whole Body Motions for a Biped Robot from Captured Human Dances*. Master Thesis, Tokyo: University of Tokyo. 2003.
- [10] Siscart, Marc Rosanes. *Algorithms and Graphic Interface Design to Control and Teach a Humanoid Robot Through Human Imitation*. Master Thesis, Catalunya: Universitat Politècnica de Catalunya. 2011.