

# Design and building motion capture system using transducer Microsoft kinect to control robot humanoid

Gun Gun Maulana<sup>1\*</sup>, Yuliadi Erdani<sup>1</sup>, Aris Budiarto<sup>1</sup>, and Wahyudi Purnomo<sup>1</sup>

<sup>1</sup>Politeknik Manufaktur Bandung, Teknik Otomasi Manufaktur dan Mekatronika, Jl. Kanayakan 21, Bandung, Indonesia

**Abstract.** Currently the development of robots has reached a high level of complexity. However, it is also accompanied by increasing complexity problems in its control. Some control with the image processing method also requires competent operators, at least memorize each command used. Therefore, to reduce the complexity in its control, in this research robot control, especially humanoid robot is made using motion capture method. Motion capture is a control technique using a camera transducer in the form of Microsoft Kinect, to obtain the coordinate skeleton of the joint user. Furthermore the data is processed in Visual Studio software to obtain the angular information that will be formed by the robot. So that the robot can perform the same movement with the user. The information is transmitted wirelessly to the microcontroller contained on the robot in real-time. The results of research showed that the system can translate the user's movement into information movement of humanoid robot. With an average skeleton vector detection error of 1.69 cm, an average response time of 1.3 seconds and an absolute error average position of the end effector on the x axis of 3.2 cm and on the y-axis of 1.28 cm.

## 1 Introduction

The use of robots in the present has blended into all aspects of everyday life. Because with the robot, various human activities can be helped so that more effective and efficient, therefore they are usually called service robot. Service robot created more likely humanoid robot because the humanoid robot offers high flexibility.

The problem of the humanoid robot is the control of the robotic joints. Manual programming can be done to move the robot joint from one position to another, then record the state of that position. But such programming techniques are not flexible and difficult to do. So comes the more recent method of control, namely by using a camera that serves as a transducer producing gesture data then used to drive the humanoid robot.

The transducer chosen for this research spell is Microsoft Kinect. Kinect was chosen because the device is a unit of several other devices such as stereovision camera, infrared camera, RGB camera and microphone especially because Kinect has good compatibility with Microsoft Visual Studio software which is used as software interface in this research

Some research on humanoid robot control using image processing has been done, but most transducers used are web-cam to generate RGB image data or by using 2 cameras (stereo vision). One of them is research conducted by Ikai Takuya at Nagoya University in 2012. Using a camera that reads commands through the movement of a finger, which is then translated by the controller into a robot movement [1], Then R. Deepan at SASTRA University in 2015 uses a camera for tracking

human fingers and recognizing gestures based on color and contour changes, and by counting the distance from the midpoint of the palm to the tip of each finger [2], Another study conducted at Al-Nahrain University is already using Microsoft Kinect transducer [3]. Imitation process movements can already be done in real-time, but the imitated part is still limited to the arm, so as not to maximize the available features [4].

From the background are formulated several problems, including:

- Human / human readout process using Microsoft Kinect transducer into gesture skeleton data.
- The process of translation of skeleton gesture data into a humanoid robot movement.
- The process of sending commands wirelessly from PC to microcontroller contained in humanoid robot.
- Performance in the operation of humanoid robot.

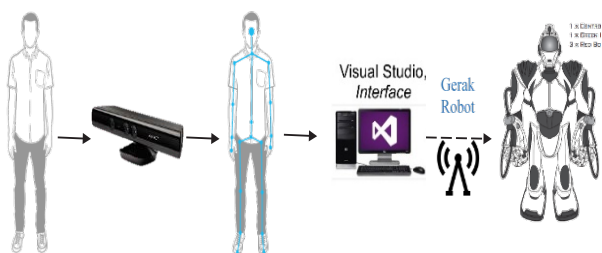
The purpose of this research project is to control the humanoid robot using image processing method with motion capture, as the solution of the control of humanoid robot that is less flexible. Human movement can be detected and imitated in real-time by humanoid robot, with the limitation of movement according to the specification of the robot

## 2 Research methods

### 2.1 System overview

Here is the general architecture of the system that has been created. Described in fig. 1 below.

\* Corresponding author: [gungun@polman-bandung.ac.id](mailto:gungun@polman-bandung.ac.id)



**Fig. 1.** System Architecture.

Users serve as a set point that produces movement, the movement of the user will be read as a skeleton data which then in-tracking by Microsoft Kinect device. The coordinates data  $x$ ,  $y$ ,  $z$  that have been tracked by Microsoft Kinect will be the input for the program in Visual Studio software, which is then calculated into corners formed from the joint combination.

The user corner that has been obtained from the calculation, then converted into a command movement of the humanoid robot. The command data of the humanoid robot movement is transmitted wirelessly to an additional module located behind the humanoid robot. Wireless communication device used is Bluetooth HC-06 with serial communication protocol.

In testing phase of suitability of humanoid robot position with user position, used object detection method to read coordinate end effector from humanoid robot. The object used is a small orange ball placed on the end of the arm. Furthermore the ball is read using the camera and searched the middle point. That point is the end effector of humanoid robot. Before the system translates the user's motion into humanoid robot motion, initial initialization of sound command "Follow Me" is required. Initialization is useful as a security that indicates the user has been in a position ready to run the system. For emergency conditions was used sound command in the form of "Stop".

## 2.2 Plot of making stages

### 2.2.1 System design

Place the figure as close as possible after the point where it is first referenced in the text. If there is a large number of figures and tables it might be necessary to place some before their text citation. If a figure or table is too large to fit into one column, it can be centred across both columns at the top or the bottom of the page.

This stage is done to choose the right method to answer the problems that have been defined previously. Starting from the determination of hardware or actuator, transducer used, until the selection of software used for research.

For the Control section, the Controller must have fast computing capabilities and media storage large enough to ensure maximum data processing runs. Because the average image processing device requires a device with a minimum 2.6 GHz processor frequency. This research chooses the use of PC (Personal Computer) as part of the control based on the consideration of criteria, endurance,

flexibility and process speed. PC is better when compared to microcontroller, mini PC or PLC.

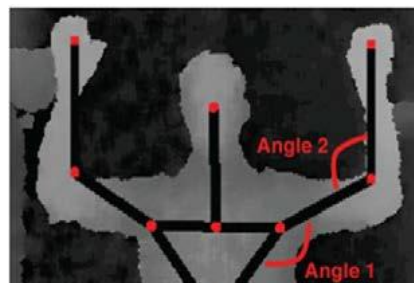
For the transducer device used in this study selected Kinect device from Microsoft. Microsoft Kinect was chosen because of its easy availability, and the support of software and libraries better than other devices.

In the software section, there are also some selection options. Selection of software becomes very important to ensure the system is made to meet the demands and expectations. Some of the software available in the market including Robot Operation System, Labtech, LabVIEW, Matlab, and Visual Studio. The software used in this study is Visual Studio from Microsoft, because it is able to manipulate, analyze and display data and has the ability to arrange multimedia with the help of additional devices. Especially because Microsoft Visual Studio has good compatibility with Microsoft Kinect transducer because it comes from the same manufacturer.

### 2.2.2 Creation of skeleton tracking program

That acts as a transducer, to generate gesture data from users. Human movement has a high degree of complexity and has a wide range of motion possibilities. However, the robot used in this study has a simple structure with a degree of freedom (degree of freedom) is less than humans. It is therefore necessary to cluster any human movements that can be implemented into a humanoid robot movement. The thing to note is the number of joints found in robots and humans. The human arm has a mechanism consisting of a bullet joint and a hinge joint with rotation axes to inculcate the abduction-adduction, flexion-extension, and internal rotation of the upper arm [5]. While the robot used in this study only has two joints in each arm. Each robotic arm is designed only to perform flexion and extension movements on the shoulder, as well as flexion and extension at the elbow.

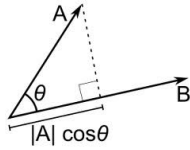
In the arm, there are several movements to be imitated, including the flexion and extension of the rotating arm moves around the parallel axis through both shoulders. Both sections of the joints become the tracking point by the kinect transducer. As illustrated in the fig. 2 below.



**Fig. 2.** Examples of parts of the joint are read by kinect

The Microsoft Kinect Transducer can read the  $x$ ,  $y$ , and  $z$  coordinates at each point found in the joint. To find angles formed on each joint can be done by calculation using the dot vector equation. In Euclidean geometry, the angle formed between two vectors is the

dot multiplication of a vector expressed in an orthonormal basis corresponding to the length and angle formed between the vectors [5]. The relationship is repressed in fig. 3 below.



**Fig. 3.** Representation dot vector ineuclidean.

If A and B are a vector and  $\theta$  is the angle formed by the two vectors, the dot product of vectors A and B is expressed as:

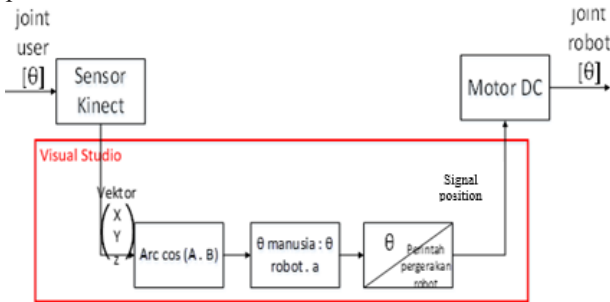
So the angle value  $\theta$  can be searched using the following equation:

$$\theta = \cos^{-1} \left( \frac{A \cdot B}{\|A\| \|B\|} \right) \quad (1)$$

The form  $\frac{A}{\|A\|}$  is the unit vector form of, so the angle  $\theta$  can be expressed as:

$$\theta = \cos^{-1}(\bar{A} \cdot \bar{B}) \quad (2)$$

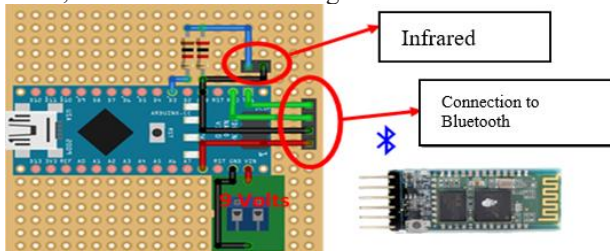
The following explanation of the system in the form of a comprehensive block diagram illustrated in the picture below.



**Fig. 4.** Block diagram of system.

**2.2.3 Control robot using Arduino**

Another upgrade done to the humanoid robot is the addition of the arduino module so that the robot can be controlled via PC. Initially the robot can only be controlled using a joystick provided specifically for the robot. After being given additional modules, communication between humanoid robot and PC / computer is doing by wirelessly using bluetooth module. Additional modules are installed on the back of the robot, with the circuit in the fig. 5 below.



**Fig. 5.** Additional module for communication.

**2.2.4 Integration of Kinect readings into Arduino**

After the robot works and can be controlled using microcontroller, then input data is converted into data from Microsoft Kinect reading with skeletal tracking program that has been made before. However, due to the limitations of the motor movement resolution on the humanoid robot, then made a comparison to restrict human motion into the motion of humanoid robots. Described in the table 1 below.

**Table 1.** Comparison between angle of user and motor rotation.

State of position	Angle of user	Motor rotation
9	<30°	0.00°
8	30° - 37,14°	22.50°
7	37,15° - 44,28°	45.00°
6	44,29° - 51,42°	67.50°
5	51,43° - 58,57°	90.00°
4	58,58° - 65,71°	112.50°
3	65,72° - 72,85°	135.00°
2	72,86° - 80°	157.50°
1	>80°	180.00°

Because the smallest resolution of motor movement on the humanoid robot of 22.5 degrees with a maximum rotation angle of 180 degrees, therefore made the position segmentation for the movement made proportional. The following equation of transformation matrix shows the relationship between angle to end-effector coordinate.

$$T_4^0 = A1 * A2 * A3 * A4$$

$$T_4^0 = Rot_{x,\alpha} \cdot Rot_{y,\theta_1} \cdot Rot_{x,\beta} \cdot Rot_{x,\theta_2}$$

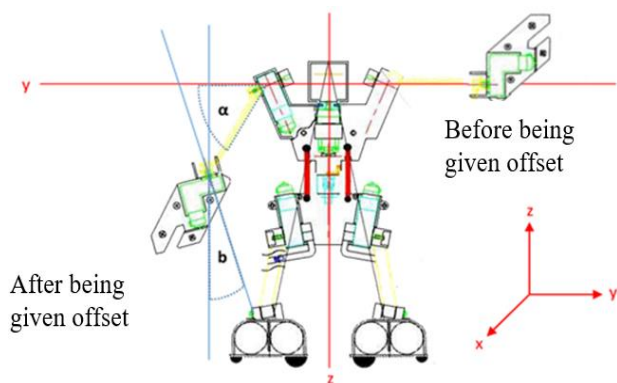
$$A1 = \begin{bmatrix} 1 & 0 & 0 & a1 \\ 0 & \cos\alpha & -\sin\alpha & b1 \\ 0 & \sin\alpha & \cos\alpha & c1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A2 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & a2 \\ 0 & 1 & 0 & b2 \\ -\sin\theta_1 & 0 & \cos\theta_1 & c2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A3 = \begin{bmatrix} 1 & 0 & 0 & a3 \\ 0 & \cos\beta & -\sin\beta & b3 \\ 0 & \sin\beta & \cos\beta & c3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A4 = \begin{bmatrix} 1 & 0 & 0 & a4 \\ 0 & \cos\theta_2 & -\sin\theta_2 & b4 \\ 0 & \sin\theta_2 & \cos\theta_2 & c4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{initial coordinates} = \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \text{final coordinates} = T_4^0 * \text{initial coordinates}$$

Since the position of the motor laying on the joints of the elbows and shoulders is not the same and not as axis, the first step is to rotate the shoulder and elbow link to be on a single axis of alpha ( $\alpha$ ) and beta ( $\beta$ ) cartesian, shown in A1 and A3 matrices. So the position of the arm after being given offset to be like in fig. 6 below.

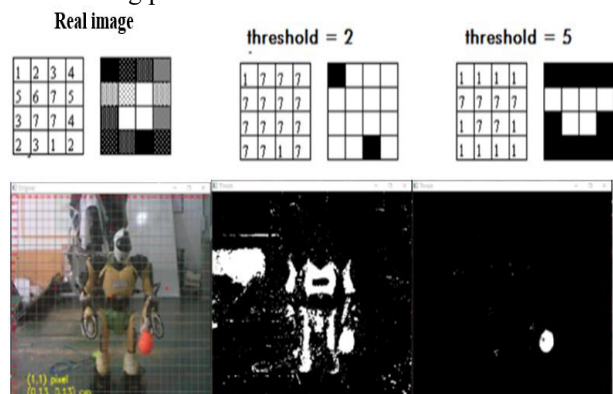


**Fig. 6.** Offsetting.

### 2.2.5 Full system testing procedure

The main variable in this system is the position of end effector to the user position, which is tested by object detection method using camera. The first step of object detection is image segmentation, which is the process of dividing the image into several different values. The division of the image includes the color of the object and the background. The technique that will be used to segment the research is thresholding, or in Indonesian language can be interpreted as a threshold. The thresholding process defines a pixel value that is above the threshold to a certain value, whereas the pixel value below the threshold becomes a value far below the threshold or zero [6].

The image below shows an example of a thresholding process.



**Fig. 7.** Result of thresholding process.

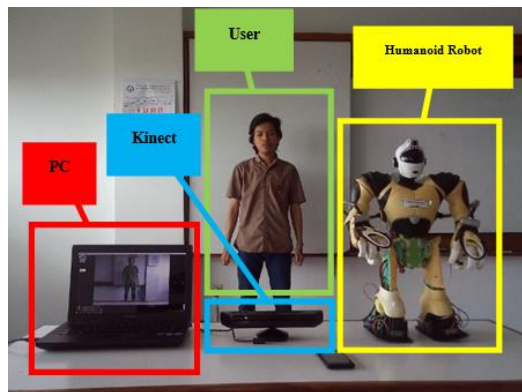
Next, it is determined how to calculate the position of the object in the image representing the position of the object in the real world. To define the position of an object, the object center is determined by using the CoG (center of gravity) calculation provided by the program library. The image below shows an example of a program for determining the midpoint of an object.

Furthermore, objects that have been read by the system will have coordinate values on the image. The coordinate data on the image is converted to real world coordinate data (real), with a ratio of 1 pixel: 0.128 cm. The Gambar below shows the camera calibration position against the humanoid robot calibration pose.

## 3 Implementation result

### 3.1 Design result

The design that has been done before in Chapter III, has been realized with the following results.



**Fig. 8.** Result of design implementation.

### 3.2 Sensor testing

This test is performed to determine the accuracy and limitation of kinect sensor readings. The process is to take the data depth by reading every change 10 cm, then the data compared with the reading ruler with a precision of 0.5 mm. The results of kinect sensor testing are listed in table 2 below.

**Table 2.** Accuracy of sensor testing result.

Reference distance (cm)	Actual distance (cm)	Error (%)
70	72,4	3.31
80	81,6	1.96
90	91,9	2.07
00	102,1	2.06
200	202,3	1.14
300	302,	0.66
400	402,4	0.6
<b>Average</b>		<b>1.69</b>

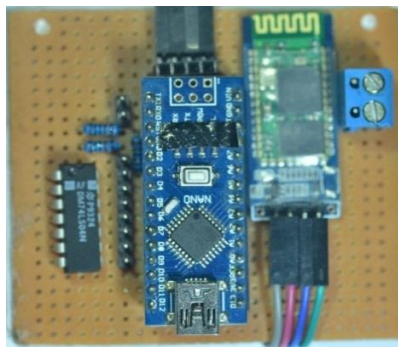
From the test results, it can be concluded that the sensor depth on the kinect is a precise and accurate sensor. The average error of measurement is 1.69 cm. In the measurements made, it can generally be seen that the errors that occur even greater if the distance of the depth sensor test even further. In this experiment also obtained that the deadzonekinect is <50 cm and> 4 m. In the deadzone, the depth sensor can not read distances and can not perform human tracking.

### 3.3 Additional module

To be able to communicate with PCs, humanoid robots are given an additional module as described previously.



The following fig. 9 shows the actual results of additional modules.



**Fig. 9.** Additional modul.

The additional module is equipped with bluetooth HC-06 as a wireless communication device with PC.

However, due to the addition of bluetooth HC-06, the delay in data reception becomes increased. It is also affected by the distance between the sender / PC and the receiver. Testing is done using the mailing list command on arduino that is made to resemble the stopwatch function.

### 3.4 Real-time imitation testing

After the gesture data can be read by kinect, then the data is converted into information movement of humanoid robot.

The results show that the program created successfully translates the movement / state of user position into a command state robot position in real-time.

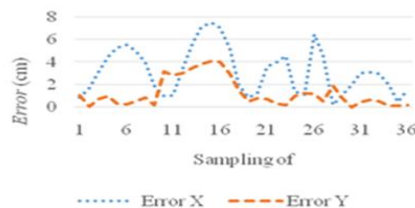
### 3.5 Position of end effector testing

This test is performed to determine whether the position of the robot arm in accordance with the expected. The test is done by object detection method using the camera as the object color identifier. Objects detected in the form of an orange ball placed at the end of the arm as an end-effector marker of the robot. Fig. 10 shows an example of its implementation.



**Fig. 10.** Using of object markers of the end effector.

The fig. 11 below shows the test data of the end effector position that has been collected from several experiments with all the possible points obtained from theuserposition.



**Fig. 11.** Result of positon accuracy of the end effector testing.

From the above data it can be concluded that the forward kinematics equation for the shoulders and elbows is valid, with an absolute average of 3.2 cm on the x axis and 1.28 cm on the y-axis. The deviation of the position does not show a certain pattern. The cause of the deviation is quite large due to the mechanical construction of the robot that is not rigid and there is a large backlash on each joint, especially because in humans, from hand to elbow there are actually four joints, while the robot model used only two.

## 4 SUMMARY

### 4.1 Conclusions

From the research that has been done, it can be concluded several things as follows:

- The system can read skeleton gesture data with an average error of 1.69 cm and successfully translate the data into information movement of humanoid robot.
- The absolute error of the average end effector on the x axis of 3.2 cm and the y-axis of 1.28 cm.
- Delay robot motion to follow the average human motion for 1.3 seconds.
- The system successfully makes the operation of humanoid robot easier. The statement is evidenced by the percentage of quality gained from the questionnaire of 88% and reduced learning time.

### 4.2 Suggestion

After doing research, suggestion for further research as follows:

- Use more advanced actuators, such as humanoid robots that have a scale and degree of freedom that are closer to the original person.
- Using a better image processing device using 2 or more devices simultaneously.
- Conducting the process of imitation of all parts of human to humanoid robot.

## References

1. Takuya, I. *Behavior Control of Robot by Human Finger Direction*. (Elsevier, Nagoya, 2012).
2. Asfahl, R. *Robot and Manufacturing Automation*. (John Wiley & Sons, New Jersey, 1992).

3. Deepan, R. Hand Gesture Based Control of Robotic Hand using Raspberry Pi Processor. *Asian Journal of Scientific Research*, 392-402. (2015).
4. Hernandez-Lopez, J.-J. *Detecting Objects Using Color and Depth Segmentation with Kinect Sensor*. (Elsevier, Guanajuato, 2012).
5. Indrajit, W. *Rancang Bangun Sistem Motion Capture dan Database Motion Untuk Robot Humanoid dengan Perangkat Microsoft Kinect Berbasis ROS (Robot Operating System)*. (UI Diglib, Depok, 2012).
6. Du, G.L., Zhang, P., Mai, J.H. and Li, Z.L. (2012) *Markerless Kinect-Based Hand Tracking for Robot Teleoperation*. *I. J. of Advanced Robotic Systems*, **9**, 36 Spong, (2012).