# Proportional Derivative Control Based Robot Arm System Using Microsoft Kinect

Rizqa Afthoni[1], Achmad Rizal[2] and Erwin Susanto[3]
[1,2,3]Engineering School, Telkom University
Bandung 40257, Indonesia
[1]rizqathemaster1@ gmail.com, [2]arl@ittelkom.ac.id,[3]ews@ittelkom.ac.id

*Abstract-* **Today, the development of manipulator robot technology is growing fast and giving good impact for human life. Application of the manipulator robot technology can help many human's tasks effectively. However, most publications in this field of interest are based on complex computation, by using mathematical model from the actual robot. This paper proposes a control system for manipulator robot using Microsoft Kinect based on proportional-derivative control algorithm (PD-control). The function of Microsoft Kinect is as a sensor that detects position of Kinect user joints. The data from Microsoft Kinect will be processed by using inverse kinematics for mapping the position of user joints to the manipulator robot. After that, position controlling for manipulator robot will be processed based on PD-control algorithm in order to obtain the position of the manipulator robot in accordance with the movement of the user. The experiments have already been successfully implemented for movement of manipulator robot with some movement patterns directed by user.**

***Keywords— proportional-derivative control; Microsoft Kinect; inverse kinematic; manipulator robot***

## I. INTRODUCTION

Nowadays, the development of science and technology is growing very rapid. It gives a positive impact on people's lives. One of the results of this technology development is robotic that can help people to work effectively. The use of robotic technology especially manipulator robot has touched in various fields, such as in the industry fields, medical, nuclear, defense and many other research interests. This is be caused by many jobs must be done with a satisfied level of accuracy, robustness and stability.

To develop a sequence of manipulator robot movement generally is using mathematical modeling approached from the kinematics chain of the robot, for instance see [1]-[4].

In the mathematical approach, the robot manipulator must be modeled by involving several parameters such as mechanical and system dynamic. It takes a longer time relatively to obtain the mathematical analysis.

There are several techniques that can be used to solve the problem, for example by using a camera to detect the joint configuration of the human body so that it can be used as a flexible panel to move the manipulator robot. However, the technique needs complex initial preparation.

According to description above, this paper proposes a control system for manipulator robot using Microsoft Kinect [5], [6] to detect the position of the user joints that will be used as a panel to move the robot. Proportional-derivative control scheme is applied to track the reference directed by user. Three steps are provided to control the manipulator robot.

First step is tracking process carried out by Microsoft Kinect to detect the joints of user. There are 15 points of user joints that will be detected. Those points are head, neck, torso, left-shoulder, left-elbow, left-hand, right-shoulder, right-elbow, right-hand, left-hip, left-knee, left-foot, right-hip, right-knee, right-foot. In this paper we only use 3 joints point, namely right-shoulder, right-elbow and right-hand. These points will produce Lse vector, Leh vector and Lsh vector.

Lse vector is a vector that is formed by right shoulder joint and right elbow joint. Leh vector is a vector formed by right elbow joint and right hand joint. Lsh vector is a vector that is formed by right-shoulder joint and right-hand joint. Those vectors will be used to make inverse kinematics (IK) formula.

Second step is an initiation process. In this step, the mapping between IK models of user joint and the manipulator are carried out. Manipulator model uses 3 degree-of-freedom (DOF). Every movement of DOF will be adjusted to the movement of the flexion-extension shoulder user joint, abduction-adduction shoulder user joint and flexion-extension elbow user joint.

Third step is a position control process using proportional-derivative (PD)-control algorithm. This algorithm is selected for handling error position of manipulator robot in order to obtain the sequence movement of the robot in accordance with the movement of the user. The values of proportional and derivative constants based on tuning of Ziegher-Nichols oscillation method [7]. Manipulator model has 3 servo motors, each of servo motor has own PD value.

This paper focuses on implementation of IK [8] and PD-control algorithm for manipulator robot using Microsoft Kinect. The goal is to simulate the task of movement in accordance with the movement of the user and specify the position and orientation of its end-effector in task space coordinates.

## II. TRACKING PROCESS

Process tracking performed by the Microsoft Kinect aims to detect 15 joints. Before this sensor detects those joints, Microsoft Kinect must be calibrated by doing psi pose. User stands upright with arms raised as high as head. Microsoft

978-1-4799-1208-7/13/$31.00 ©2013 IEEE

Kinect will automatically calibrate until those joints are detected, see figure 1.

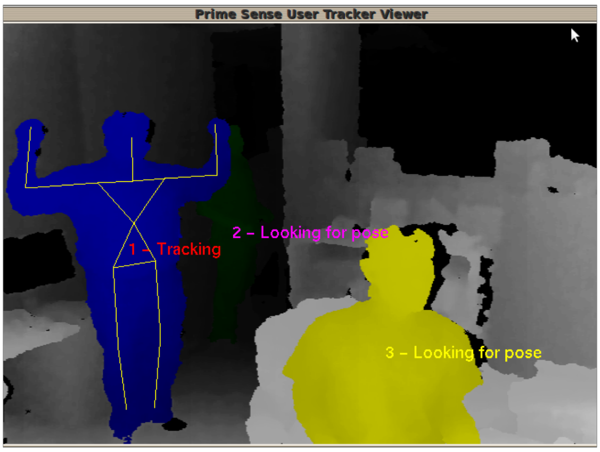


Fig. 1. Psi pose for calibration of Microsoft Kinect sensing

From figure 1, it is shown that the Microsoft Kinect is successful for detecting user. There are 15 points of joints user that will be detected. Those points are head, neck, torso, left-shoulder, left-elbow, left-hand, right-shoulder, right-elbow, right-hand, left-hip, left-knee, left-foot, right-hip, right-knee, and right-foot.

Data from the Microsoft Kinect sensor is a 3-dimensional coordinate data. The data is fed to the inverse-kinematics formula. The coordinate of the center point is on the kinect sensor.

Data from the Microsoft Kinect sensor is a 3-dimensional coordinate data. The data is input to the inverse-kinematics formula. Coordinates of the center point is on the kinect sensor.

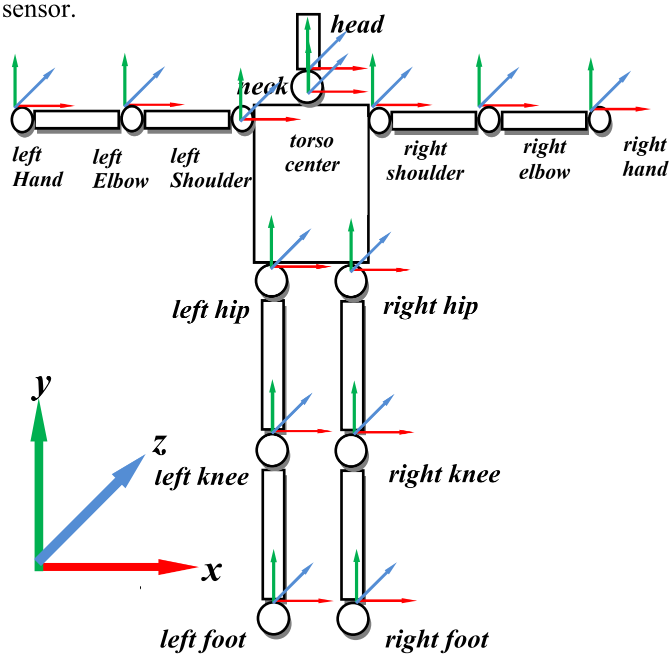


Fig. 2. Fifteen joints were detected by Microsoft Kinect

In this paper will only use 3 joints, Those joints are right shoulder, right elbow and right hand. Those points will produce Lse (shoulder-elbow) Vector, Leh (elbow-hand) Vector and Lsh (shoulder-hand) Vector. Those vectors will be used to make IK formula for right arm user.

## III. Initiation Process

Initiation process will be carried out the mapping between IK models of joint user to the robot manipulator. Manipulator robot model uses 3 degree-of-freedom (DOF). Every movement of DOF will be adjusted to the movement of the flexion-extension right shoulder joint user, abduction-adduction right shoulder joint user and flexion-extension right elbow joint user. Configuration of manipulator robot is shown in the figure 2.

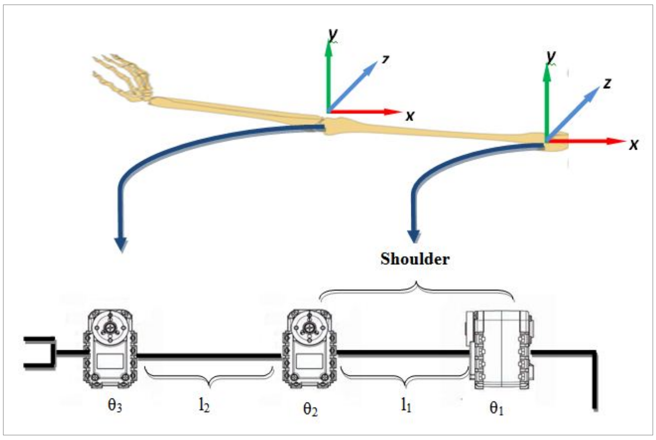


Fig. 3. Fifteen mapping of joints user to servo motors

Figure 3. shows the mapping of right arm of user to kinematic chain of manipulator robot. Those servo motor angles are $\theta_1$, $\theta_2$ and $\theta_3$. Angle $\theta_3$ is the angle of rotation of flexion-extension elbow joint user joint user. Meanwhile, angles $\theta_1$ and $\theta_2$ are the rotation angle for the user's right shoulder. Angle $\theta_1$ follow the movement of the user towards the flexion-extension right shoulder joint user, while the angle $\theta_2$ is to follow the movement of abduction-adduction right shoulder joint.

There are several vectors that must be considered to get the angles of $\theta_1$, $\theta_2$ and $\theta_3$. Lse vector, Leh vector and Lsh vector. As already explained, Lse vector is the difference between the position of the right shoulder joint and right elbow joint. Leh vector represents the difference between the position of the right elbow joint and right hand joints. Meanwhile, Lsh vector represents the difference between the position of the right shoulder joint and right hand joints. Solution to find the angles would have to use IK.

Inverse kinematics is a method to find the joint angles, given the end effector position and orientation. Inverse kinematics is much harder than forward kinematics. One of the problems in IK is multiple solutions. To overcome that problem, IK model is not from the kinematics chain of the manipulator robot, but the IK model is from the user's right arm, so that the servo motors on the manipulator robot will make the movement of the IK calculations based on the right arm user. These have already been successfully implemented for movement of manipulator robot with some movement patterns from user.

*A. IK solution for the angle $\theta_3$*

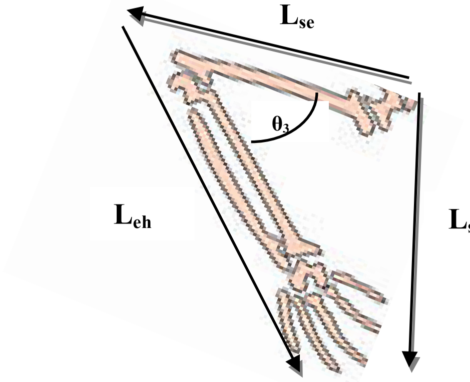


Fig. 4. Kinematics of $\theta_3$

$$\overline{L}_{se} \bullet \overline{L}_{ek} = \left\|\overline{L}_{se}\right\|\left\|\overline{L}_{ek}\right\| \cos\theta_3 \tag{1}$$

$$\theta_3 = \cos^{-1} \frac{\overline{L}_{se} \bullet \overline{L}_{ek}}{\left\|\overline{L}_{se}\right\|\left\|\overline{L}_{ek}\right\|} \tag{2}$$

By using $L_{eh}$ vector and $L_{se}$ vector, angle $\theta_3$ can be obtained as shown in figure 4., whereas dot product multiplication is used to get the angle $\theta_3$.

*B. IK solution for the angle $\theta_2$*

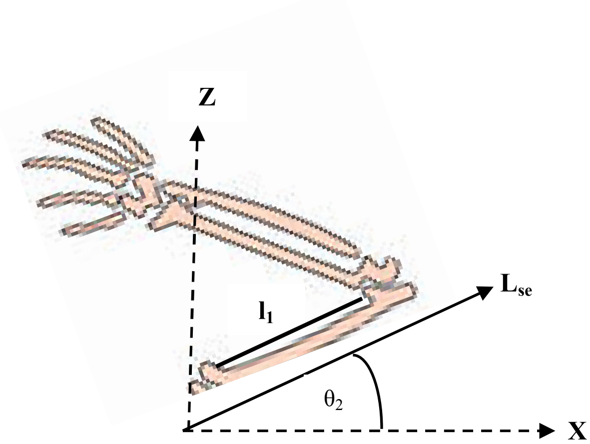


Fig. 5. Kinematics of $\theta_2$

$$\overline{L}_{se_x} = l_1 \cos\theta_2 \tag{3}$$

$$\theta_2 = \cos^{-1}\left(\frac{\overline{L}_{se_x}}{l_1}\right) \tag{4}$$

The angle of $\theta_2$ can be obtained through the Lse vectors as shown in figure 5. By using the rules of the triangle trigonometry to the axis x and the length of $l_1$, IK solution for the angle $\theta_2$ can be obtained as shown in (3)-(4). Similarly, we can get the angle $\theta_1$ based on figure 6.

*C. IK solution for the angle $\theta_1$*

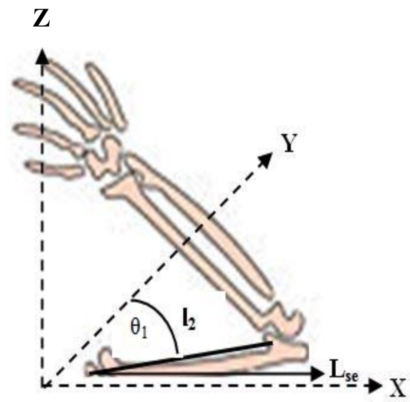


Fig. 6. Kinematics of $\theta_1$

$$\overline{L}_{se_y} = l_2 \cos\theta_1 \tag{5}$$

$$\theta_1 = \sin^{-1}\left(\frac{\overline{L}_{se_y}}{l_2}\right) \tag{6}$$

## IV. CONTROL PROCESS

After the initiation process, then the next step is the control process. This step aims to correct the error value of the position of manipulator robot. Error value is the difference between the process value and the calculation of IK (set points). This is processed so that the output from the control system becomes reference of correction values of servo motor position

PD-control algorithm is selected for handling error position of manipulator robot in order to obtain the sequence movement of the robot in accordance with the movement of the user. The implementation of the control method for the manipulator robot does not use integral control; because it has response to highly accumulative error that will cause the system response oscillates.

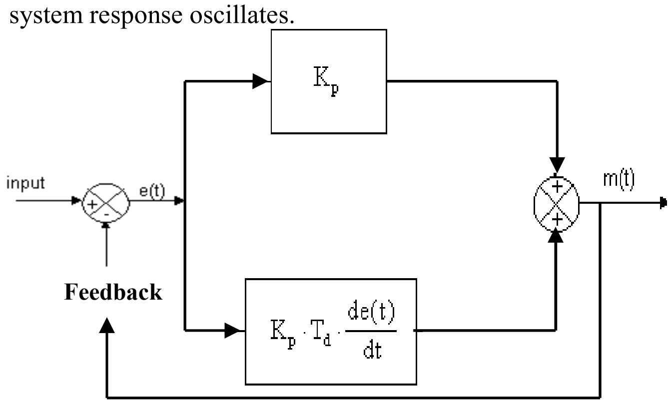


Fig. 7. Diagram block of PD-control

The values of proportional and derivative constants are derived based on tuning ziegher-nichols oscillation method. Manipulator robot model has 3 servo motors, each of servo motor has own PD value.

### *A. PD-Control Tuning for $\theta_1$*

TABLE I. Performance parameters of the system at some constants of P

| Experiment | P | Parameter Performance | | | |
|---|---|---|---|---|---|
| | | ***Delay Time ($t_d$)*** | ***Rise Time ($t_r$)*** | ***Peak Time ($t_p$)*** | ***Overshoot ($M_p$)*** |
| 1 | 0,19 | 1,5 s | 3 s | - | - |
| 2 | 0,38 | 1,55 s | 3 s | - | - |
| 3 | 0,57 | 1,5 s | 3 s | - | - |
| 4 | 0,76 | 1,5 s | 3 s | - | - |
| 5 | 0,95 | 1,5 s | 2,99 s | - | - |
| 6 | 1,14 | 1,81 s | 2,99 s | 3,1 s | 0.752 % |
| 7 | 1,13 | 1,475 s | 2,90 s | 3 s | 0.752 % |
| 8 | 1,52 | 1,475 s | 2,90 s | 3 s | 1.055 % |
| 9 | 1,71 | 1,45 s | 2,90 s | 3 s | 1.357 % |
| 10 | 1,9 | 1,45 s | 2,90 s | 3 s | 1.660 % |

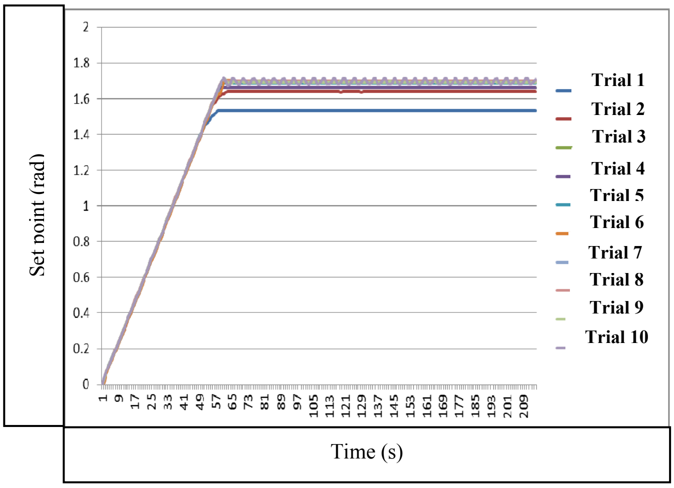


Fig. 8. Graph the effect of proportional constant for servo motor $\theta_1$

In accordance with the rules of ziegler-nichols oscillation method, the critical gain (Kcr) must be obtained first. Based on the above data was obtained the value of Kcr and Pcr. Both of these data can be obtained proportional and derivative constants using ziegler nichols table.

TABLE II. Ziegler-Nichols Table

| Type of Controller | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| PID | 0.6Kcr | 0.5Pcr | 0.125Pcr |

$$Kp = 0.6 \times Kcr = 0.6 \times 1.9 = 1.14 \quad (7)$$

$$Kd = 0.125 \times Pcr = 0.125 \times 0.3 = 0.0375 \quad (8)$$

From the calculation above, it can be concluded that the best performance based on the ziegher-nichols oscillation method for servo motor $\theta_1$ when the values are Kp = 1.14 and Kd = 0.0375.

### *B. PD-Control Tuning for $\theta_2$*

TABLE III. Performance parameters of the system at some constants of P

| Experiment | P | Parameter Performance | | | |
|---|---|---|---|---|---|
| | | ***Delay Time ($t_d$)*** | ***Rise Time ($t_r$)*** | ***Peak Time ($t_p$)*** | ***Overshoot ($M_p$)*** |
| 1 | 0,21 | 1,40 s | 3,55 s | - | - |
| 2 | 0,42 | 1,35 s | 2,98 s | - | - |
| 3 | 0,63 | 1,29 s | 2,66 s | - | - |
| 4 | 0,84 | 1,29 s | 2,54 s | - | - |
| 5 | 1,05 | 0,50 s | 1,70 s | - | - |
| 6 | 1,26 | 1,29 s | 2,49 s | 2,54 s | 1,90 % |
| 7 | 1,47 | 1,49 s | 2,49 s | 2,59 s | 1,16 % |
| 8 | 1,68 | 1,05 s | 2,10 s | 2,15 s | 1,90 % |
| 9 | 1,89 | 1,31 s | 2,52 s | 2,59 s | 2,26 % |
| 10 | 2,1 | 1,20 s | 2,20 s | 2,55 s | 2,30 % |

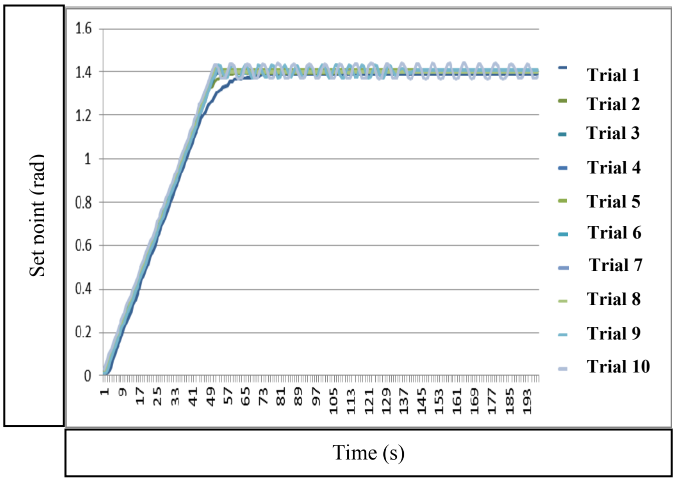


Fig. 9. Graph the effect of proportional constant for servo motor $\theta_2$

In accordance with the rules of ziegler-nichols oscillation method, the critical gain (Kcr) must be obtained first. Based on the above data was obtained the value of Kcr and Pcr.

$$Kp = 0.6 \times Kcr = 0.6 \times 2.1 = 1.26 \quad (9)$$

$$Kd = 0.125 \times Pcr = 0.125 \times 0.299 = 0,0373 \quad (10)$$

Similar with the previous Ziegler-nichols oscilation method, it can be concluded that the best performance for servo motor $\theta_2$ when the values are Kp = 1.26 and Kd = 0.0373.

### C. PD-Control Tuning for $\theta_3$

TABLE IV. Performance parameters of the system at some constants of P

| Experiment | P | Parameter Performance | | | |
|---|---|---|---|---|---|
| | | *Delay Time ($t_d$)* | *Rise Time ($t_r$)* | *Peak Time ($t_p$)* | *Overshoot ($M_p$)* |
| 1 | 0,26 | 1,17 s | 2,94 s | - | - |
| 2 | 0,52 | 1,15 s | 2,44 s | - | - |
| 3 | 0,78 | 1,15 s | 2,44 s | - | - |
| 4 | 1,04 | 1,14 s | 2,24 s | - | - |
| 5 | 1,30 | 1,14 s | 2,34 s | - | - |
| 6 | 1,56 | 1,14 s | 2,24 s | 2,34 s | 1,14 % |
| 7 | 1,82 | 1,14 s | 2,24 s | 2,34 s | 1,14 % |
| 8 | 2,08 | 1,14 s | 2,24 s | 2,34 s | 2,65 % |
| 9 | 2,34 | 1,14 s | 2,24 s | 2,34 s | 3,05 % |
| 10 | 2,60 | 1,10 s | 2,20 s | 2,34 s | 1,87 % |

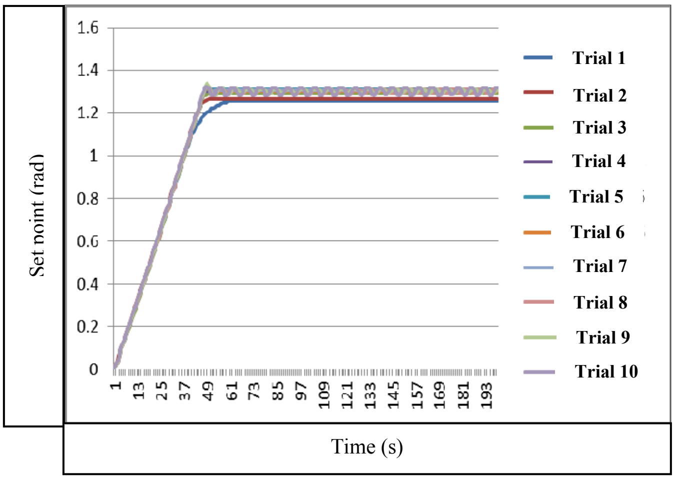


Fig. 10. Graph the effect of proportional constant for servo motor $\theta_3$

In accordance with the rules of ziegher-nichols oscillation method, the critical gain (Kcr) must be obtained first. Based on the above data was obtained the value of Kcr and Pcr.

$$Kp = 0.6 \times Kcr = 0.6 \times 2.6 = 1.56 \quad (11)$$

$$Kd = 0.125 \times Pcr = 0.125 \times 0.3 = 0{,}0375 \quad (12)$$

It can be concluded that the best performance based on the ziegher-nichols oscillation method for servo motor $\theta_3$ when the values are Kp = 1.56 and Kd = 0.0375.

## V. RESULT

This testing is done by running the manipulator robot to a particular position based on the position of the joints of the user is detected by the Microsoft Kinect. This test aims to determine the reliability of the movement of manipulator robot.

### A. Testing the movement of the manipulator robot

Fig. 11. The movement of the manipulator robot

- Result of the testing on the servo motor $\theta_1$

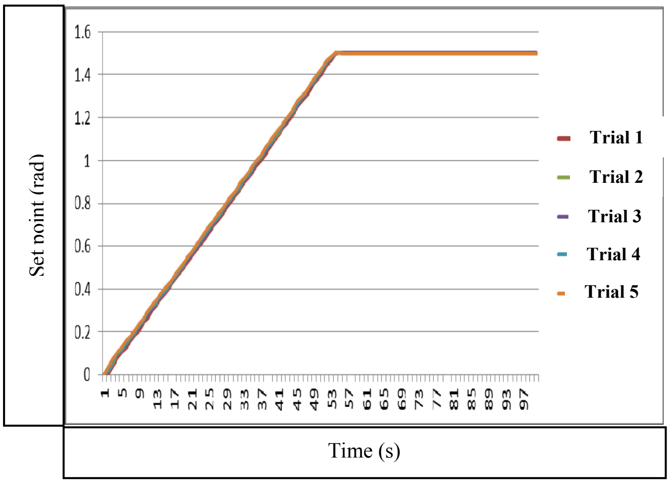


Fig. 12. Plot the process value of servo motor $\theta_1$

TABLE V. System performance of servo motor $\theta_1$

| Experiment | Parameter Performance | | | |
|---|---|---|---|---|
| | *Delay Time ($t_d$)* | *Rise Time ($t_r$)* | *Peak Time ($t_p$)* | *Overshoot (Mp)* |
| 1 | 1,81 s | 2,99 s | 3,1 s | 0,2 % |
| 2 | 1,80 s | 2,99 s | 3,1 s | 0,2 % |
| 3 | 1,81 s | 3,00 s | 3,1 s | 0,2 % |
| 4 | 1,82 s | 3,00 s | 3,2 s | 0,2 % |
| 5 | 1,80 s | 2,99 s | 3,1 s | 0,2 % |

According to the table V and Fig. 12 above, it can be concluded that servo motor $\theta_1$ system running quite stable, where the value of the parameter does not change significantly. It can be concluded that the system will remain stable when tested repeatedly.

- Result of the testing on the servo motor $\theta_2$

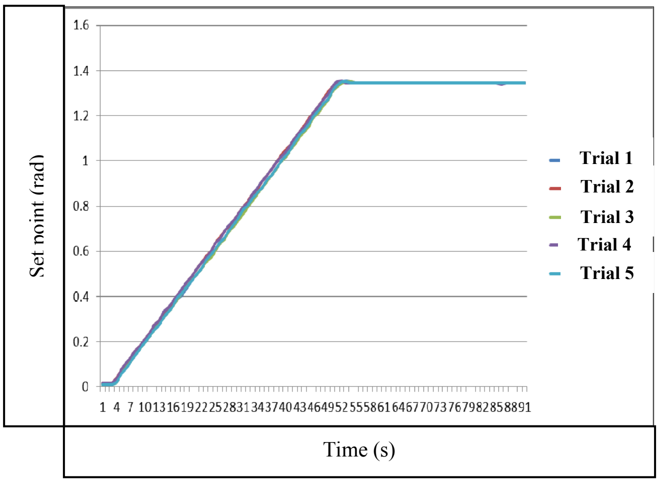


Fig. 13. Plot the process value of servo motor $\theta_2$

TABLE VI. System performance of servo motor $\theta_2$

| Experiment | Parameter Performance | | | |
|---|---|---|---|---|
| | *Delay Time ($t_d$)* | *Rise Time ($t_r$)* | *Peak Time ($t_p$)* | *Overshoot (Mp)* |
| 1 | 1,27 s | 2,49 s | 2,54 s | 1,16 % |
| 2 | 1,30 s | 2,55 s | 2,50 s | 1,16 % |
| 3 | 1,29 s | 2,49 s | 2,54s | 1,16 % |
| 4 | 1,35 s | 2,35 s | 2,50 s | 1,16 % |
| 5 | 1,29 s | 2,49 s | 2,54 s | 1,16 % |

From the table VI and figure 13., it can be concluded that servo motor $\theta_2$ system running quite stable, where the value of the parameter does not change significantly. It can be concluded that the system will remain stable when tested repeatedly.

- Result of the testing on the servo motor $\theta_3$

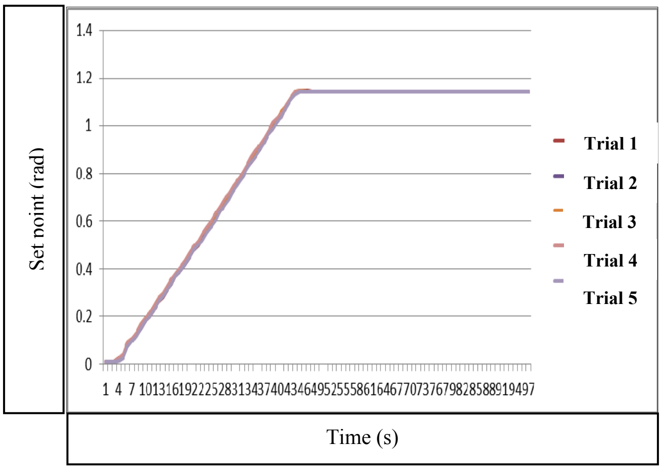


Fig. 14. Plot the process value of servo motor $\theta_3$

TABLE VII. System performance of servo motor $\theta_1$

| Experiment | Parameter Performance | | | |
|---|---|---|---|---|
| | *Delay Time ($t_d$)* | *Rise Time ($t_r$)* | *Peak Time ($t_p$)* | *Overshoot (Mp)* |
| 1 | 1,41 s | 2,24 s | 2,34 s | 1,14 % |
| 2 | 1,40 s | 2,25 s | 2,37 s | 1,14 % |
| 3 | 1,41 s | 2,24 s | 2,34 s | 1,14 % |
| 4 | 1,41 s | 2,24 s | 2,34 s | 1,14 % |
| 5 | 1,41 s | 2,24 s | 2,34 s | 1,14 % |

Hence, it can be concluded that servo motor $\theta_3$ system running quite stable, where the value of the parameter does not change significantly. It can be concluded that the system will remain stable when tested repeatedly.

## VI. CONCLUSION

This paper developed a methodology for controlling a manipulator robot. A control system for manipulator robot using Microsoft Kinect based on Proportional-Derivative control algorithm (PD-control). The function of Microsoft Kinect is as a sensor that detects position of joints kinect user. Position controlling for manipulator robot will be processed by using PD-control algorithm in order to obtain the position of the manipulator robot in accordance with the movement of the user. These have already been successfully implemented for movement of manipulator robot with some movement patterns from user.

## REFERENCES

[1] K. Ogata, Modern Control Engineering. Jakarta, Erlangga, 1997.
[2] C. John, Introduction to Robotics Mechanics & Control, Stanford University, 1986.
[3] A. Muis and W. Indrajit.. Realistic Human Motion Preservation-Imitation Development on Robot with Kinect : Indonesia University, 2012.
[4] J.M. I. Zannatha, R. C. Lim'on, “Forward and inverse kinematics for a small-sized humanoid robot,” 2009 International Conference on Electrical, Communications, and Computers, pp. 111-118.
[5] M.A. Ali, H. A. Park, and C.S.G. Lee, “Closed-form inverse kinematic joint solution for humanoid robots,” The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 704-709.
[6] P. Falco and C. Natale, “On the stability of closed-loop Inverse kinematics algorithms for redundant robots,” IEEE Trans. on Robotics, vol. 27, no. 4, 2011, pp. 780-784.
[7] G. Antonelli, “Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems,” IEEE Trans. on Robotics, vol. 25, no. 5, 2009, pp 985-994