Jurnal
Teknologi

Full Paper

Leap Motion Underwater Robotic Arm Control

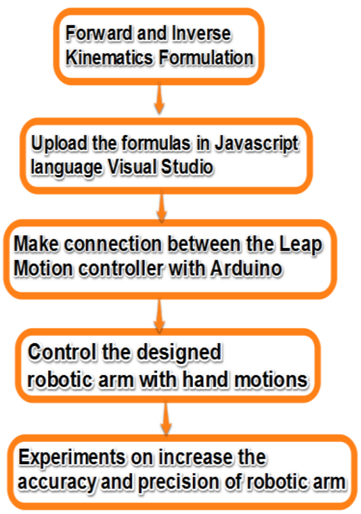
Lee Jun Wei*, Loi Wei Sen, Zamani Md. Sani

Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Malaysia

Article history
Received
13 March 2015
Received in revised form
14 April 2015
Accepted
15 June 2015

*Corresponding author
louis911_librazone@hotmail.com

Graphical abstract



Abstract

The robotic arm structure and control algorithm are designed for a purpose, to pick and place an object task at underwater which is attached to a ROV (Remotely Operated Underwater Vehicle). It is controlled by an innovated gesture control system, Leap Motion controller. The arm structure of pick and place is controlled by Arduino as microcontroller to control the angles and displacements of the servomotor precisely. The detection of position and orientation of the fingers and hands processed by develop control algorithm in Javascript language and sent to the Arduino. Meanwhile, a detailed 3D drawing is drawn precisely by using SolidWorks for the fabrication. After the platform is completed, kinematic and inverse kinematic equations and calculations are programed into JavaScript language for the control algorithm. Lastly, the hardware and software combined all together. With developed control algorithm, the robotic arm mimics human's fingers and arm movements which more user friendly interface especially underwater scavenging and salvaging. Since it designed for underwater, the accuracy and precision are crucial for robotic arms, it undergo several experiments and tests for investigate reliability performance of developed robotic arm.

Keywords: Leap motion, robotic arm, inverse kinematic, javascrip, precision, accuracy

Abstrak

Struktur lengan robot dan algoritma kawalan direka bentuk bagi sesuatu maksud, untuk mengangkat dan meletakkan satu objek di dalam air yang disertakan kepada ROV (Remotely Operated Underwater Vehicle). Ia dikawal oleh inovasi sistem kawalan isyarat menggunakan pengawal Leap Motion. Struktur lengan mengangkat dan tempat dikawal oleh mikropengawal Arduino sebagai untuk mengawal sudut dan anjakan daripada servomotor yang tepat. Mengesan kedudukan dan orientasi pada jari dan tangan diproses oleh membangunkan kawalan algoritma dalam bahasa Javascript dan dihantar ke Arduino. Sementara itu, lukisan 3D terperinci dilukis dengan tepat dengan menggunakan SolidWorks untuk fabrikasi. Selepas platform itu selesai, persamaan kinematik dan kinematik songsang dan pengiraan yang diprogramkan ke dalam bahasa JavaScript untuk algoritma kawalan. Akhir sekali, perkakasan dan perisian digabungkan semua bersama-sama. Dengan algoritma kawalan maju, jari robotik meniru lengan manusia dan pergerakan lengan yang lebih antara muka mesra pengguna memerangkap terutama di dalam air dan penyelamat. Oleh kerana ia direka untuk air, ketepatan adalah penting untuk robotik, ia menjalani beberapa eksperimen dan ujian untuk menyiasat prestasi kebolehpercayaan lengan robot.

Kata kunci: Leap motion, lengan robot, kinematik songsang, javascript, ketepatan

1.0 INTRODUCTION

Remotely operated underwater vehicle has been used for deep underwater task for few decades. Underwater robot plays a very important role in probing, salvaging, checking and repairing on-service pipeline [1]. Oil and gas industries have been created work-class ROV to perform task at offshore fields while the navy crews have been using it for rescue and recovery operation on the ocean floor. However, the ROV need one or a pair of robotic arms to perform pick and place task during on the sea bed. Most of robotic arms are controlled by joysticks or button which are conventional controller in the market. Further, the conventional robotic arm controlled by complicated control panel with numbers of buttons, joysticks or touchscreen panel which need time and manuals for fully operate. The most likely issue that operator of the controller found out difficulties in using those control panels and complex interface.

Alternative control mechanisms are developed to perform same task underwater. The combination on advantages in human perception and recognition skills with consistent and accuracy robots as well as a human-robot collaboration system can enhance target identification rate and reduce the complexity of robotic systems [2, 3].

Recently, researches have proposed better mechanism on controlling underwater robotic arm. One of the mechanism is black glove which combine the hand positioning and arm gestures using data glove [4]. This provide better user friendly interface but it come with some limitations like wear the glove that limits it freedom even though precision is high. Moreover, it required proper light and camera angle for capturing better hand gesture.

Next, electromyography (EMG) signals are alternative way for controlling arm by collect the data from human arm motion by the movement of muscles [5]. A specialize technique is needed to handle the light source and viewing angle to capture hand gestures efficiently [4].

In reality, controlling underwater ROV robotic arm required a reconfigurable control station design for robotic operation with graphical user interfaces is used to control the telerobotics arm. The station is huge and used eight devices controllers which are two hand controllers, two forceballs, two I/O simulators, a head tracker, and a 6-dof Mouse [6]. Despite having all these available assisting tools, improvement for the aids assisting controller are still much needed.

Leap Motion controller, one of acquisition of 3D optical sensors is newly invented technology in last few years. It detects 10 fingers with hand gestures with precise and accurately with minimum error 0.2mm [7]. Due to it high accurate and precise detection, it becomes one of the popular research in control mechanism in robotic arm in industry such as KUKA Roboter GmbH [7]. Moreover, it allowed to control in more intuitive with human movement instinct to control robotic arms by wave over one or two hands over the controller.

Above all, there are a range of dynamic effects that imposed on the robotic arm during underwater operations such as buoyancy and viscous drag will produce different dynamic response than in air. Thus a self-tuning and continually update gain PID controller must be implemented in the underwater robotic arm. This implementation able to provide real time adaptive robotic arm controller [8].

2.0 DEVELOPMENT OF MOTION CONTROL UNDERWATER ROBOTIC ARM CONTROL SYSTEM

To begin, Leap Motion proposed as the controller to control underwater robotic arms with the features on high accuracy and precise output. To test on accuracy and precise of underwater robotic arms, it requires to design a prototype where it needed a good model of the robotic arm with 5 degree of freedoms (DOF) that simulated human's hand movement in underwater scavenging and salvaging. It required a control algorithm to ensure robotic arm in pick and place tasks. Hence, the prototype is developed to test precision and accuracy on performance such as pick and place tasks in underwater which resemble condition in the air. It is carried out by experiments and analysis to review accuracy and precision.

2.1 Connection between the Leap Motion Controller and Robotic Arm

Generally, the control mechanism started with a wave of a hand or lift of a finger over wide open space of 24 cm^2 on Leap Motion controller which sensed hands and fingers and also followed every moves that have make. The Figure 2.1 below shows Cartesian coordinate system of x, y and z axes of the Leap Motion controller [9]. The origin is at the midpoint on top of the Leap Motion controller.

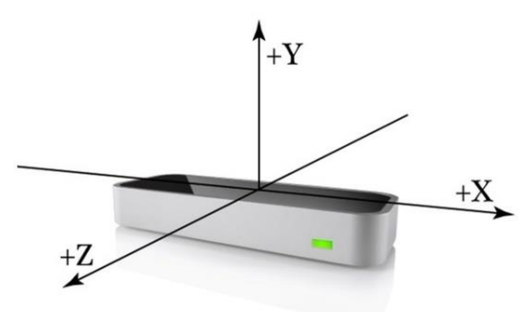


Figure 2.1 Coordinate system of Leap Motion controller

Figure 2.2 Coordination data of a palm

Firstly, the Leap Motion control panel must allow send tracking information to web apps that requested it. Then data of the hands and fingers that the Leap Motion controller captured is observed from the JSON Viewer as shown in Figure 2.2. These extracted data of position and orientation from hand and fingers from controller and sent to host computer. In addition, the robotic arm is controlled by the microcontroller called Arduino Uno Rev which has the Standard Firmata protocol to communicate with the host computer. While in host, node.js is acted as a library which allowed the Leap Motion controller to communicate with the Arduino Uno board and further processed the commands to perform the desire task like pick and place. Lastly, the programming is executed through Node.js command prompt.

2.2 Mathematical Modeling

As the Leap Motion controller is giving the position of x, y and z and orientation for each fingers when the fingers is on top and in the range of the Leap motion controller. The position of the hand is acted as the tool at the end effector of the robotic arm, gripper. Thus the inverse kinematics able to find the required joint angles to place the tool as the tool is the same position and orientation as the hand.

The forward kinematics calculation is started with affixing the frames at the robotic arm links as shown Figure 2.3 below. Only two degree of freedom to be determined in the forward and inverse kinematics because this two degree of freedom is related to each other. As the position data of the hand is collected from Leap Motion, the data are sent to use in inverse kinematics of the robotic arm. While, the waist and wrist parts are independent to shoulder and elbow of the robotic arm because the data are able to obtain from Leap Motion's JSON Viewer.

For the waist, the x-axis of the Leap Motion data can be collected to represent the angle of the waist should turn. Nevertheless, the motion data of the wrist can be get from "palmNormal". Specifically, "palmNormal[0]" is for wrist's supination and pronation while "palmNormal[2]" is for wrist's flexion and extension. Lastly, the gripper, or as the end effector is not count as a degree of freedom. Figure 2.3 shows the two degree of freedom of the shoulder and the elbow [10]. The Leap Motion controller fed the position of the hands and fingers in the x, y and z-axis, thus the angle of the shoulder and elbow is calculated as below [10]:

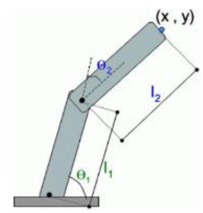


Figure 2.3 Two degree of freedom of the shoulder and the elbow

To acquire the equations for solving the angle of each joint after getting the coordination positions, the formulation for the inverse kinematics derived initially according to the designed robotic arm prototype. The limitation range for Leap Motion controller is recorded.

$$x = l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) \quad (2.1)$$

$$y = l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2) \quad (2.2)$$

Let (x, y) is target position for the end-effector of a robot with only two degree of freedom θ_1 and θ_2.

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1l_2\cos(\theta_2) \quad (2.3)$$

$$\cos(\theta_2) = \frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2} \quad (2.4)$$

where θ_2 angle is obtained from the Equation [2.4] by using simultaneous method.

$$x = l_1\cos(\theta_1) + \left(l_2(\cos(\theta_1)\cos(\theta_2) - \sin(\theta_1)\sin(\theta_2)\right) \quad (2.5)$$

$$x = \cos(\theta_1)\left(l_1 + l_2\cos(\theta_2)\right) - \sin(\theta_1)\left(l_2\sin(\theta_2)\right) \quad (2.6)$$

$$y = \cos(\theta_1)\left(l_2\sin(\theta_2)\right) + \sin(\theta_1)\left(l_1 + l_2\cos(\theta_2)\right) \quad (2.7)$$

$$\cos(\theta_1) = \frac{x+\sin(\theta_1)\,l_2\sin(\theta_2)}{l_1 + l^2\cos(\theta_2)} \quad (2.8)$$

$$\sin(\theta_1) = \frac{(l_1+l_2\cos(\theta_1))y - l_2\sin(\theta_2)\,x}{l_1^2+l_2^2+2l_1l_2\cos(\theta_2)} \quad (2.9)$$

and θ_1 can be obtained from Equation [2.8] or Equation [2.9] by using another simultaneous method.

2.3 Robotic Arm Prototype

The designed robotic arm has five degree of freedoms (DOF) which consisted of waist, shoulder, elbow, wrist and fingers where wrist part has two degree of freedoms. It able to do human's wrist movements which are flexion, extension, pronation and supination. Flexion and extension are a linear movement at the wrist while pronation and supination are the rotation movement at the wrist. The structure of the gripper showed the number of fingers, means the position of the two fingers determined the position of the clamps of the gripper. By developing 5 DOF, the robotic arm able to mimic the motion of a human arm while the operator controlling the robotic arm for pick and place objects task. Based on the formulation of the inverse kinematics equations that are modeled, the underwater robotic arm is developed.

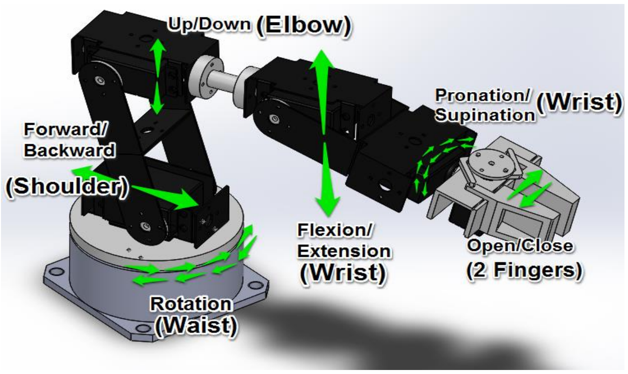


Figure 2.4 Movement of each part of the robotic arm

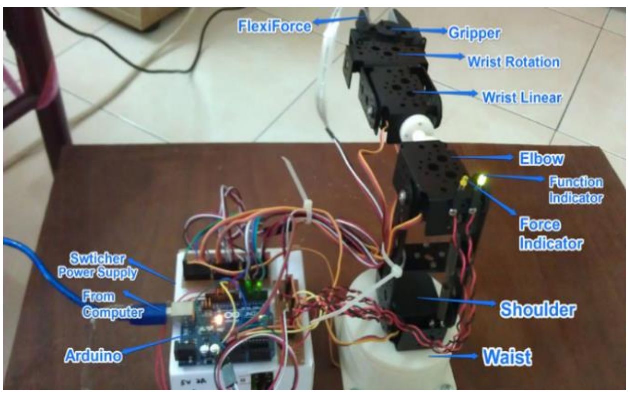


Figure 2.5 Prototype of robotic arm

3.0 EXPERIMENTAL RESULTS

Underwater task needed to withstand for a very long period of time and perform accurately. Hence, accuracy and precision of the robotic arm performance is crucial where control algorithm with the prototype is undergo tests to ensure it is reliable. There are few experiments are carried out to investigate the performance of control algorithm and its hardware in the accuracy and precision for the pick and place task.

3.1 Experimental 1: Error of the servomotor and improvement in accuracy

Objective of this experiment is to analyze the resulted error by servomotors. Servomotors used in each joint of the robotic arm are RC servomotors without encoder. Thus it has no feedback to the controller. In normal condition, the servomotor gives the angle of rotation not as programmed. In the performance underwater robotic arm, accuracy is very crucial. Errors should be eliminated in the measuring process, else the end effector of the robotic arm is unable to reach desire.

To perform this experiment, the robotic arm's waist, shoulder and elbow servomotor are focused for accuracy test. However, the data for wrist rotation and wrist linear given by JSON Viewer are in the number -1 to1. Thus, servomotor at wrist and is not focused in this experiment due to accuracy for these parts are less important and are adjustable according to the operator. Likewise, the distance between 2 claws in gripper is less important in accuracy test as is depended from the user's fingers. User able to correct the distance of the gripper's claws by observed the gripper's claws and altered by the user's finger.

As the waist is tested, the result is expected identical with shoulder and elbow. The errors of each angle are recorded for the analysis. Next, the tested servomotor undergoes the calibration process based on the error data that had obtained. The calibration process of servomotor by the follow formulation:

Calibrated Value
=Programmed Angles × [Average Error] (3.1)

Total Programmed Value
= Programmed Angles + Programmed Angles
× [Average Error] (3.2)

Figure 3.1 below shows the experiment results for both uncalibrated errors and calibrated error for the servomotor in in robotic arm's waist.

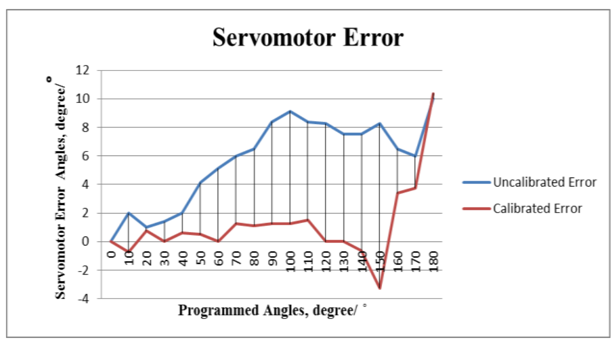


Figure 3.1 Error of the servomotor

Based on results in Figure 3.1, error of the uncalibrated servomotor increase as the programmed angle increase due the nature of servomotor. The average percentage of error for servomotor is 6.88%. This is also exist a random error that caused from the servomotor or any circuit itself.

Due to errors are varied, calibrated value is needed to reduce the percentage of error. Based on calculation that had completed before, the servomotor lack an average of 6.88% away from the actual value. Thus, the calculation needs to add another averaged of 0.0688 of programmed value in

Equation [3.1] and Equation [3.2]. Figure 3.2(a) below shows the uncalibrated and calibrated value of the servomotor, notice the needle pointed the angle of the servomotor.

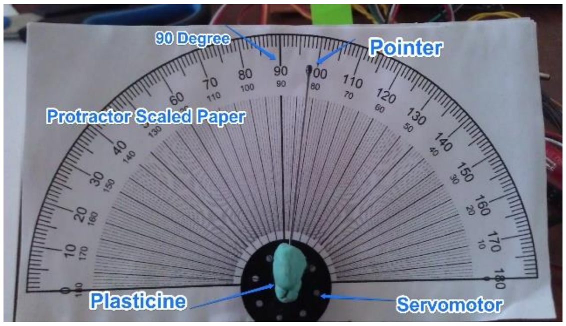


Figure 3.2(a) Servomotor angle of rotation (a) before calibrated value added

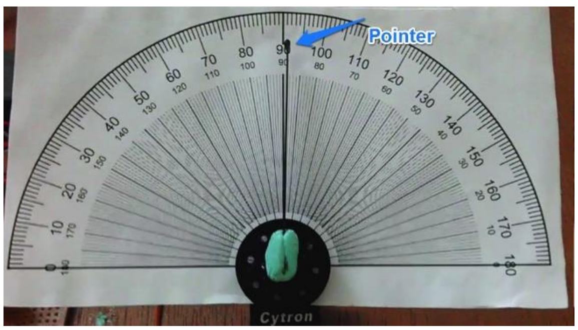


Figure 3.2(b) After calibrated value added

After the calibrated value is introduced to the calculation, the error is almost in linear while the percentage of error was decreased as shown in Figure 3.2(b) and the average percentage of error decreased from 6.88% to 1.77%. There is a total of 74.27% reduction error to increase accuracy in the servomotor. Thus, the accuracy is reliable for the servomotor after calibrated value is added to the calculation. Moreover, the joint movement at the waist are much more accurate. This calibrated value also added to the shoulder and elbow joints as it also using the same RC servomotor.

However, the percentage of error expected lower that 1.77% after the calibration process. Since the servomotor has a problem in getting to 180̊, it results greater error at the end of angle 160̊ to 180̊. This case happened because of the limitation of the servomotor itself where it never rotate to or near180 degree after programmed.

3.2 Experiment 2: Accuracy's Reliability on the Waist Experiment

After the calibrated value is added into account of the programming calculation of the servomotor, there is a decreased of error of the rotation angle of the servomotor, and this eventually increased the accuracy of the servomotor. To test the accuracy's reliability, another analysis is tested with dummy hand as an example of hand and moved on the Leap Motion controller. The Leap Motion controller also can sense the dummy hand as a real hand. Dummy hand is used to avoid hand sore and tired also easy to get measurements. Thus the coordination of the palm position's data from the JSON Viewer is fed to the programming for calculation as refer to Figure 2.2. This is to use the real time coordination of the palm position's data from the JSON Viewer and check the error of the rotation angle of the servomotor with and without the calibrated value that done previously.

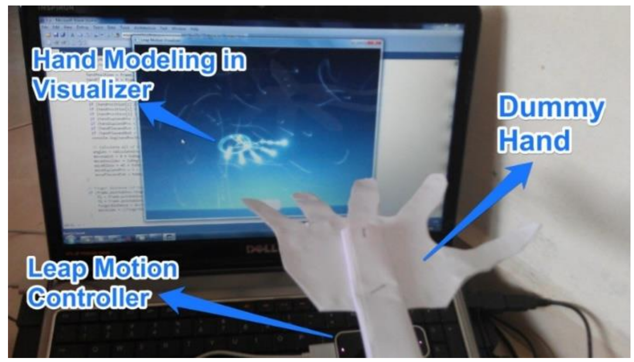


Figure 3.3 Dummy hand on leap motion visualizer

Figure 3.3 recorded the average of servomotor angles when the dummy hand and is compared with the programmed angles to find out the error of the servomotor and calculated the calibrated value to correct the error and increased the accuracy of the servomotor.

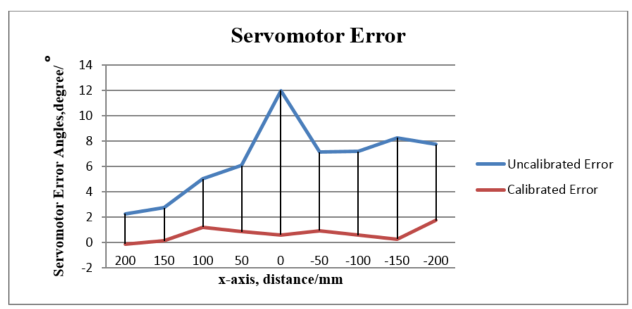


Figure 3.4 Servomotor error angles from -200 mm to 200 mm

The percentage of error of the servomotor decreased from 7.24% to 0.78%. Thus, there is a 90.19% improvement in the accuracy in the servomotor. The percentage of error is lower than 1.77% because in this experiment, the angle did not reach to angle more than 160 ° which bring more error to the calculation. Thus, there is a total of 90.19% improvement in the accuracy in the servomotor. From the Figure 3.4, the errors of the servomotor are almost in straight line. Thus, the accuracy is reliable for the servomotor after calibrated value is added to the calculation. Moreover, the joint movement at the waist are much more accurate. This calibrated value also added to the shoulder and elbow joints as it also using the same RC servomotor. Since the standard deviation of the servomotor for this experiment is 0.40 which is very low, thus is almost precise. As the conclusion, such accuracy of a robotic arm is acceptable for a human in using Leap Motion controller to control and of course, the higher the accuracy the better.

3.3 Experimental 3: Precision of Robotic Arm Repeatability Test

Second experiment is used to validate the precision of robotic arm. It designed a specific repeatability task for robotic arm to hit target coordinates, (0, 0) with 15 times while all the joints must set at fix value to reach the target coordinate and to prevent data updates to these joint. Data of robotic arm recorded and data shown as Figure 3.5 below. Based on Figure 3.5 below, the data distribution is in nonlinear form since R^2 relatively close to 0. As computed valued of standard deviation for x and z coordinate are 0.47056 and 0.4629 respectively. Furthermore, the average error shortest distance to target coordinate is 1.2527 mm. It implied that the variation of each data set relatively small and negligible. Hence, it is provides more precise output of robotic arms in perform some specific task in underwater.

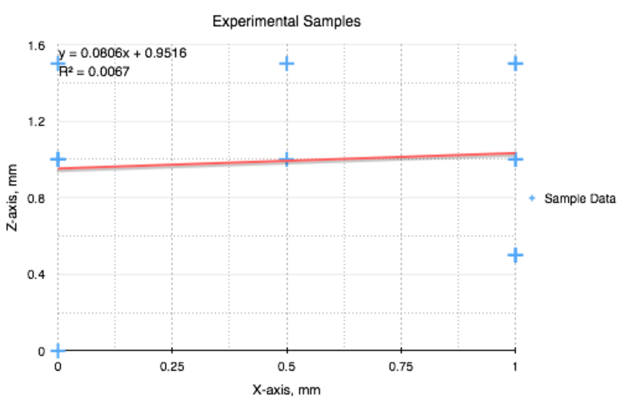


Figure 3.5 Result of repeatability test

Throughout the Experiment 3.1 to Experiment 3.3, random error can be avoided as much as possible by taking measurement repeatedly and calculate the average data taken. Moreover, the common influential factors that interfaces the experiments is systematic error which consists of human error and parallax error. As the systematic error reduced, the accuracy of the measurements increased. Another factor is the random error. This error usually occurs in electronic noise in the circuit of the servomotor or the controller. This error can be avoided by taking measurement repeatedly and calculate the average data taken.

However, the Leap Motion controller performance is light dependent. Bright light sources or reflective sources will impact performance. A few precautions are taken to avoid the effect on the Leap Motion performance as mentioned above. During all experiments, the LED indicator of the Leap Motion controller must be always in green color to avoid any unnecessary and unwanted data taken. For best performance, there few things that need to be concerned which is listed below [11]:

- Maintain clear area of view between the Leap Motion controller and the hands and fingers.
- Avoid loose sleeves, jewelry and non-transparent objects near the device.
- Avoid wear dark glove, or dark or transparent instruments to use the controller.

4.0 CONCLUSION AND RECOMMENDATION

As conclusions, the robotic arm which controlled by Leap Motion controller able to perform pick and place as part of task on the ocean floor by just using hand and fingers gestures for underwater scavenging and salvaging. Accuracy and performance of servomotor improved with calibrated experiment data. Further, accuracy and precision can further improve if the Leap Motion controller is placed in adequate light source. Last but not least, as this prototype is made of RC servomotors which without encoder, thus no feedback is available. The feedback is important as it can use feedback controller with PID controller or fuzzy logic controller to not just increase the accuracy of the robotic arm, but also improve overall performance. Likewise, robotic arm's joint should use a motor with encoder. Nevertheless, this project can be improved and developed into a telerobotics system which the robotic arm can controlled from a distance, using wireless connections which most possibly using Wi-Fi or radio frequency as the medium to transfer data to the hardware. Thus, this Leap Motion controlled robotic arm is able to compete with existing robotic arm in industrial field.

Acknowledgement

The Authors wish to express their thanks to the Universiti Teknikal Malaysia Melaka (UTeM) under the UTeM PJP Grant (PJP/2014/FKE(15D)/S01357).

References

[1] Wang H., Huang, X., Qi, X., Meng, Q. 2007. Development of Underwater Robot Hand and Its Finger Tracking Control. *IEEE International Conference on Automation and Logistics*, 19-27 August 2007. 2973-2977.

[2] Kidd. P.T. 1992. Design of Human-centered Robotic Systems. *IEEE International Conference on Robotics and Automation*, France. 225-241.

[3] Parasuraman R. Sheridan, T. B., Wickens, C. D. 2000. A Model for Types and Levels of Human Interaction with Automation. *IEEE Trans. Syst., Man, Cybern. C.* 297(3): 286-297.

[4] Raheja J. L. Shyam, R., Kumar, U., Prasad, P. B. 2010. Real-Time Robotic Hand Control using Hand Gestures. 2010 Second International Conference on Machine Learning and Computing (ICMLC), Bangalore, 9-11 February 2010. 12-16.

[5] Fukuda O. Tsuji, T., Kaneko, M. Otsuka, A. 2003. A Human-Assisting Manipulator Teleoperated by EMG Signals and Arm

Motions. *IEEE Transactions on Robotics and Automation.* 19(2): 210-222.

[6] Lane J. C. Carignan, C., Akin, D.L. 1997. Reconfigurable Control Station Design for Robotic Operations. *1997 IEEE International Conference on Computational Cybernetics and Simulation*, Orlando, 12-15 October 1997. 4: 3722-3727.

[7] Weichert F. Bachmann, D., Rudak, B., Fisseler, D. 2013. Analysis of Accuracy and Robustness of the Leap Motion Controller. *Sensors, MDPI, Switzerland.* 13(5):6380-6393.

[8] Broome D. and Wang, Q. 1991. Adaptive Control of Underwater Robotic Manipulators. 91 ICAR, Fifth International Conference Robots in Unstructured Environments, Pisa, Italy, 19-22 June 1991. 2: 1321-1326.

[9] Leap Motion Developer 2013. API Overview [Online]. (From: https://developer.leapmotion.com/documentation/javascript/devguide/Leap_Overview.html?highlight=controller%20cartesian. [Accessed on 202 September 2013].

[10] Kavraki. L. E. Protein Inverse Kinematics and the Loop Closure Problem. (2013) [Online]. From: http://cnx.org/content/m11613/latest/. [Accessed on 31 October 2013].

[11] Leap Motion Developer 2013 (2014, May 20). Leap Motion Support [Online]. From: https://support.leapmotion.com/entries/43733443-Operating-Environment. [Accessed on 20 May 2014].