International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 04 – Issue 06, November 2015

Experimental Study on Position Control Techniques of a 4-DOF Mechanical Arm based on Open-Source Software and Hardware

Zhen Gao* and Ishwar Singh
School of Engineering Technology
McMaster University,
Hamilton, Ontario, Canada
*Email: gaozhen [AT] mcmaster.ca

Abstract—**Robotic arm, which is a popular serial manipulator, has been studied for a couple of decides, especially in the areas of theoretical study including forward kinematics, inverse kinematics, singularity, workspace, various fancy control methods and simulations. In a word, a lot of research activities in academia focused on the robotic mechanism and mathematics study. Because teaching and research co-exist in most universities, current trend makes the university robot education too theoretical. In this situation, experimental study is much more important to boost both the short-term and long-term developments of the real robot technology. There is no obvious drawback of the experimental robot education, namely a lot of purchased robots are not built based on open source. Many industrial robots have self-carried closed-source software which is developed for end-users rather than developers. While for university robotics education the students are better to be treated as the developers while not end-users. Thus, the authors here propose a concept of Open-Source Robotics to explore theexperimental study for new kinematic control techniques of the 4DOF arm. The proposed idea is applicable for the generic robotic manipulators.**

Keywords-Experimental study; kinematic control techniques; mechanical arm;open-source robotics

I. INTRODUCTION

Robotic arm has been the most popular robot in the world for a couple of decades [1-4].In term of robotic mechanism, Robotic arm can be called as serial robot. Compared with parallel robot, serial robot has larger workspace and it is more convenient and flexible to be operated [5-8]. Besides, it is also more affordable.

Since robotics education is highly demanded for various programs including computer science, electrical and computing engineering, mechanical engineering, and automation engineering [9-12]. Many universities purchased industrial robots for robot technical education. The industrial robots play an important role to let the students get familiar with the robots which are used in industrial environment. Once they get sufficient training on it, they should have no big barriers to control and operate the similar machines in industry. There are two issues should be mentioned. One is the students usually have limited time to use the industrial robot due to the limited quantities. Because, an industrial robot needs a large safety zone, it is difficult to purchase many industrial robots and put them in a very spacious lab room. Another issue is that the self-carried software of industrial robots is not designed for open-source purpose, which largely limits the potentials for its functionality expansion.Since the university students, who learn robotics, should be regarded as the robot developer rather than the end-user, a robotic kit with open-source software and hardware can be introduced to train the student both hands-on and minds-on skills. In the way, they will understand how to design, build, analysis and test a robotic system from scratch.
Besides, for the idea about Open-Source Robotics, it means the robotic system architecture is open, including open algorithm in free IDE, open mechanical structure with modularity, open hardware and open HMI. Open source robot is an effective way to boost the robot technology.

II. KINEMATICS MODELLING

A. Forward Kinematics

Forward kinematics means if the input angles of the active joints are known, solve the position and orientation of the end-effector. For serial robot, the forward kinematics solution is unique. For parallel robot, it has multiple solutions [13-15].

The Denavit-Hartenberg (DH) table should be populated to calculate the forward kinematics of the robotic arm. The kinematic structure of the 4-DOF robotic arm is provided in the following figure. Note that the last DOF is about the rotation of gripper itself. It does not have to be considered for this DOF. Thus, the DH table can be represented as follows,

Table1: DH Table

i	$\alpha_{(i-1)}$	$a_{(i-1)}$	d_i	θ_i
1	0	0	0	θ_1
2	-90	l_1	0	θ_2
3	0	l_2	0	θ_3

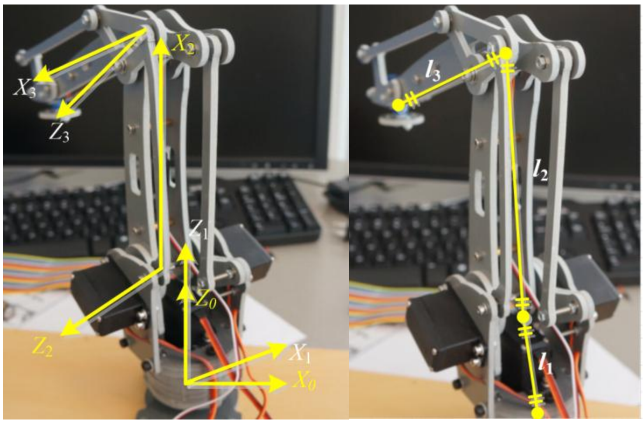

Figure 1: Kinematic structure of the robotic arm

The pose of the end-effector is derived as,

$$Pose\ of\ end\text{effector} = {}^{base}_{1}T \cdot {}^{1}_{2}\mathrm{T} \cdot {}^{2}_{3}T \cdot {}^{3}_{ef}T \qquad (1)$$

where,

$$ {}^{base}_{1}T = \begin{bmatrix} cos\theta_1 & -sin\theta_1 & 0 & 0 \\ sin\theta_1 & cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

$$ {}^{1}_{2}T = \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ -sin\theta_2 & -cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (3)$$

$$ {}^{2}_{3}T = \begin{bmatrix} cos\theta_3 & -sin\theta_3 & 0 & l_2 \\ sin\theta_3 & cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (4)$$

$$ {}^{3}_{ef}T = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (5)$$

B. InverseKinematics

The inverse kinematics means if the position of the end-effector is known, solve the input angles of the active joints. Generally, there are multiple solutions for inverse kinematics for serial robot [16-18]. Howerver, for this robotic arm, the solution of inverse kiematics is unique due to its hardware configuration.

Assumeming the base position is (0, 0, 0), and the position of the end- effector is (x, y, z). The input angles of the base, left, and right motors are theta_base, theta_left, and theta_right. According to the following figure,

$$\mathrm{theta}_{\mathrm{base}} = \tan^{-1}\frac{y}{x} (6)$$

The distance bewteeen the base and the end-effector is,

$$d = \sqrt{x^2+y^2+z^2} \qquad (7)$$

Once the coordinates of the end-effector are known, *d*is fixed. From the figure it can be found that the solid lines represent the links *l*2 and *l*3. Since the robot only can do elbow-down due to mechanicl constraints, the triangle made by the two solid lines and the dash line is fixed.

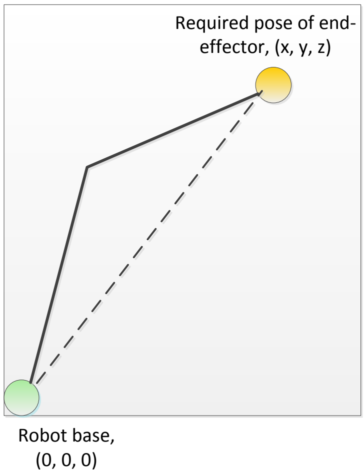

Figure 2: Relative position of end-effector with respect to the base

III. Kinematic Control via Computer Mouse/Touchpad Based on Hybrid Programming

A. Objective

It is somehow inconvenient for continuous manipulation directly based on microcontroller. There are various alternative methods to achieve continuous manipulation. One method is to integrate the joystick to control the 3 axis of this robot. Another way is to use Microsoft Kinect. However, these methods need extra devices which will make more suitable for larger project. Based on the limited resources, how to control the 3 axis of this robot by the computer mouse with the hybrid programmingis an open-ended question.

It is well known that when the mouse (cursor) is moving around the screen, use

MousePosition.X()
MousePosition.Y()

can obtain the mouse's position on the screen.Based on this truth, it can map mouse's x position to the range of [40, 90] for the 'left' servo motor. Similarly, Mouse's y position can be converted to the range of [40, 90] for the 'right' servo motor. For the 'base' servo motor,the 'MouseClick' event can be used to make the 'base' servo motor rotate in one direction.The 'MouseDoubleClick' event can be applied to make the 'base' servo motor rotate in the opposite direction.

As shown in the following figure, creating a HMI is necessary to observe the input values from the mouse and the real valueswhich are sent to the motors.The HMI can be developed with Visual Basic .Net. So when the operator moves the mouse (cursor), Label 1 and Label 2 show the real-time position of the mouse (the position is converted to the range of [40, 90] for the 'left' and 'right' servo motors). Meanwhile, the robot's 'left' and 'right' servo motors will go to that position. Label 3 and Label 4 show the actual positons of the 'left' and 'right' servo motors. In order to show the actual positons of the servo motors, on microcontroller side, it needs to use the command such as 'Serial.write' to send the position information back to Visual Basic.

For Label 3, since the range of the 'base' servo motor is [40 degrees - 110 degrees], set the initial position as 75. That means the initial text value for Label 3 is 75. When the end-user single-clicks the mouse, the value will increase; when the user double-clicks the mouse, this value will decrease. Since the range of the 'base' servo motor is [40 degrees - 110 degrees], so it may say,

```
    If PositionBase< 40 Then
PositionBase = 40
    End If

    If PositionBase> 110 Then
PositionBase = 110
    End If
```

Similar withLabel 3 and Label 4, Label 6 displays the actual positons of the 'base' servo motor, which means it will use the command 'Serial.write' again to send the related information back to Visual Basic.

Figure 3: HMI of motion control of a robotic arm based on computer mouse

B. Constraints:

The motion range of the 'base' servo motor: 40 degrees - 110 degrees.
The motion range of the 'left' servo motor: 40 degrees - 90 degrees.
The motion range of the 'right' servo motor: 40 degrees - 90 degrees.

IV. EXPERIMENTS

A. Pseudo Codeof Microcontroller

The pseudo code of the microcontroller is provided as follows,

Include the servo library;
Define leftservo;
Define rightservo;
Define baseservo;
Begin the serial communication;
Attach digital pins to the three servos;
Read the data which contains the coordinates from HMI;
Assign the related angle to each servo motor;
Control the left servo position according to the scaled value;
Control the right servo position according to the scaled value;
Control the base servo position according to the scaled value;
Time delay between two positions

B. Pseudo Codeof HMI

The pseudo code of the HMI is provided as follows,

Import the system library such as IO, ports and threading;
Define x position of mouse/cursor which is related to the left servo;
Define y position of mouse/cursor which is related to the right servo;
Define home position which is related to the base servo;
Close the serial communication port;
Setup the correct port number and the baud rate;
Setup the properties of the serial port;
Pass the x position of mouse to the predefined variable;
Map x position of mouse to the appropriate range of the left motor;
Pass the above value to the Label1's text property;
Pass the yposition of mouse to the predefined variable;
Map yposition of mouse to the appropriate range of the right motor;
Pass the above value to the Label2's text property;
Pass the value of the base motor to the Label3's text property;
Define a buffer which contains {variableX, variableY, variableBase}, which variableX means the required input of the left motor,variableY means the required input of the right motor, and variableBase means the required input of the base motor;
Open the serial port;
Write he buffer to the serial port;
Read the real values of three motors and pass the values to Label4, Label5 and Label6, respectively.
Close the serial port;

In the MouseClick event, it will say,

```
PositionBase = PositionBase + 10
    If PositionBase< 40 Then
PositionBase = 40
    End If

    If PositionBase> 110 Then
PositionBase = 110
    End If
```

For the MouseDoubleClick event, it will say,

```
PositionBase = PositionBase - 20
    If PositionBase< 40 Then
```

```
PositionBase = 40
    End If

    If PositionBase> 110 Then
PositionBase = 110
    End If
```

C. Experiments

The first step of the experimentsis to figure out the home position. As shown in the following figure, the home position is the nature position of the robot.

As show in figure 5, for the HMI status of the home position, it has,

Position of left motor: 90;

Position of right motor: 90;

Position of base motor: 75

Figure 4: Home position

Figure 5: The HMI status of the position

Stretch position means the case when the upper link is in thehorizontal level.It is also the edge of the workspace of this robot. In order to achievethis position, the mouse should be placed on the top left corner. Thus both the left motor and the right motor can reach the edges of the motion scopes, as shown in figures 6 and 7.

Figure 6: The stretch position

Figure 7: The HMI status of the stretchposition

For the approaching position, the end-effector comes towards the operator. The mouse can be moved from the top-left cornor of the screen to the right-bottom direction. Simulanetously, the mouse can be single-clicked to make the base motor rotate towards the operator.

As show in figures8 and 9, for the HMI status of theapproaching position, it has,

Position of left motor: 71;

Position of right motor: 69;

Position of base motor: 105

Figure 8: An approaching position

Figure 9: The HMI status of the approaching position

Similarly, for the moving-away position, it is required to double-click the mouse to make the base motor rotate away from the operator as show in figures10 and 11.
In this case, the required values and the real angles of the three servos are,

Required values for the position of left motor: 42;
Required values for the position of right motor: 86;
Required values for the position of base motor: 50

Real values for the position of left motor: 41;
Real values for the position of right motor: 88;
Real values for the position of base motor: 50

It can be found that there is some difference between the required values and the real values for the left motor and right motor. The time delay for serial communication between the microcontroller and the HMI is the main reason.

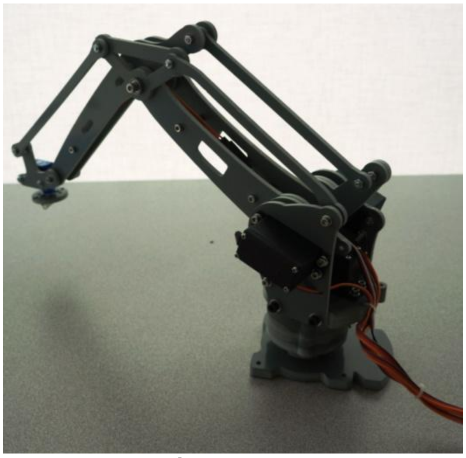
Figure 10: The moving-away position

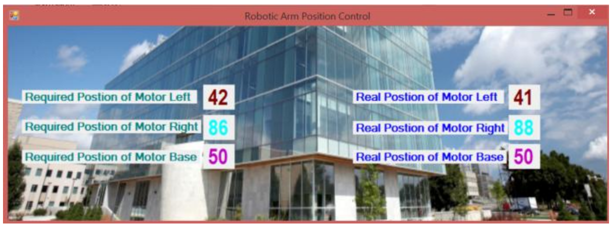

Figure 11: The HMI status of themoving-away position

D. Application

An application is setup to pick some target from the table. A target is randomly placed on the table. By controlling the mouse, the angles of the three motors will be correctly adjusted. Firstly of all, the robot is held with an initial position as shown in figure 12. Then, single click or double click the mouse to align the base motor with the line from the base to the target as shown in figure 13. After that, move the mouse on the screen to put the end-effector right above the target. As shown in figures 14 and 15, finally, approach the target to pick it up.

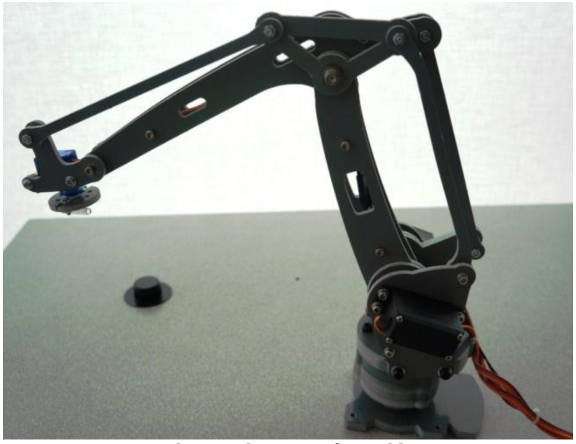
Figure 12: The initial position for grabbing a target

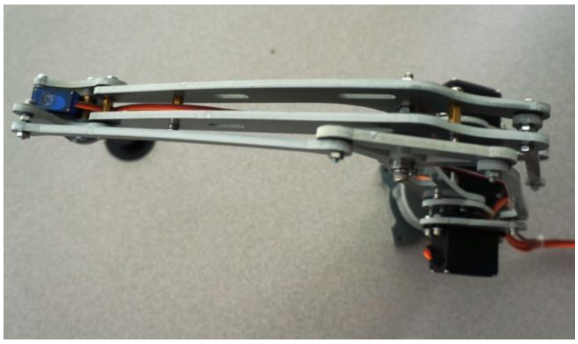
Figure 13: The topview of the first transit position

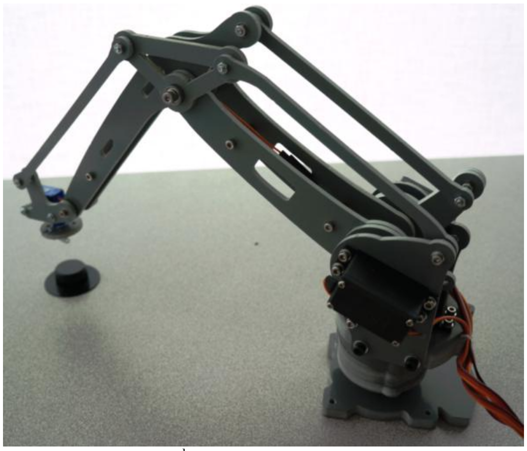
Figure 14: The 2nd transit position for grabbing a target

Figure 15: The final position for grabbing a target

V. CONCLUSIONS

Robot arm is extensively applied in industry and university education. The work proposes an idea which is called open-source robotics to fill the gap between the robotics education and robot development. It is believed that open-source robotics can boost the development of robotic technique, especially for the development of small scale and affordable robots. A case study about how to control the positionof a 4-DOF robotic arm based on open-source software and hardware is discussed. It is expected that, just like personal computer in late 20 century and this century, personal robot will become more popular and affordable with open-source concept.

REFERENCES

[1] Michiel Plooij, Wouter Wolfslag, Martijn Wisse,"Robust feedforward control of robotic arms with friction model uncertainty,"Robotics and Autonomous Systems, Volume 70, August 2015, pp. 83-91

[2] Salvador Dura-Bernal, George L. Chadderdon, Samuel A. Neymotin, Joseph T. Francis, William W. Lytton,"Towards a real-time interface between a biomimetic model of sensorimotor cortex and a robotic arm,"Pattern Recognition Letters, Volume 36, 15 January 2014, pp. 204-212

[3] Abdullah Özer, S. Eren Semercigil, R. Prasanth Kumar, Pulas Yowat,"Delaying tool chatter in turning with a two-link robotic arm,"Journal of Sound and Vibration, Volume 332, Issue 6, 18 March 2013, pp. 1405-1417

[4] Ankit Sharma, Mathew Mithra Noel,"Design of a low-cost five-finger anthropomorphic robotic arm with nine degrees of freedom," Robotics and Computer-Integrated Manufacturing, Volume 28, Issue 4, August 2012, pp. 551-558

[5] Chao Yuan, Lu-Ping Luo, Quan Yuan, Jing Wu, Rui-Jun Yan, Hyeonguk Kim, Kyoo-Sik Shin, Chang-Soo Han, "Development and evaluation of a compact 6-axis force/moment sensor with a serial structure for the humanoid robot foot," Measurement, Volume 70, June 2015, pp. 110-122

[6] Ionel Staretu, "Structural synthesis, work spaces and direct kinematic of the one serial kinematic chain with 8 axes for industrial robots," Procedia Technology, Volume 19, 2015, pp.207-214

[7] Rolland-Michel Assoumou Nzue, Jean-François Brethé, Eric Vasselin, Dimitri Lefebvre,"Comparison of serial and parallel robot repeatability based on different performance criteria," Mechanism and Machine Theory, Volume 61, March 2013, pp.136-155

[8] Donghun Lee, TaeWon Seo, Jongwon Kim,"Optimal design and workspace analysis of a mobile welding robot with a 3P3R serial manipulator," Robotics and Autonomous Systems, Volume 59, Issue 10, October 2011, pp. 813-826

[9] ChanMin Kim, Dongho Kim, Jiangmei Yuan, Roger B. Hill, Prashant Doshi, Chi N. Thai, 'Robotics to promote elementary education pre-service teachers' STEM engagement, learning, and teaching," Computers & Education, Volume 91, 15 December 2015, pp. 14-31

[10] Victor Andreev, Sergey Kuvshinov, Valentin Pryanichnikov, Yury Poduraev, "Education on the Basis of Virtual Learning Robotics Laboratory and Group-controlled Robots," Procedia Engineering, Volume 69, 2014, pp. 35-40

[11] Hani H. Rashid, Yuk-Yuen M. Leung, Megan J. Rashid, Gregory Oleyourryk, John R. Valvo, Louis Eichel, "Robotic surgical education: A systematic approach to training urology residents to perform robotic-assisted laparoscopic radical prostatectomy," Urology, Volume 68, Issue 1, July 2006, pp. 75-79

[12] Jacob Nielsen, Henrik Hautop Lund,"Modular robotics as a tool for education and entertainment," Computers in Human Behavior, Volume 24, Issue 2, March 2008, pp. 234-248

[13] Shirin Fartash Toloue, Mohammad-R. Akbarzadeh, Alireza Akbarzadeh, Mohsen Jalaeian-F,"Position tracking of a 3-PSP parallel robot using dynamic growing interval type-2 fuzzy neural control," Applied Soft Computing, Volume 37, December 2015, pp. 1-14

[14] Mats Isaksson, Luke Nyhof, Saeid Nahavandi,"On the feasibility of utilising gearing to extend the rotational workspace of a class of parallel robots," Robotics and Computer-Integrated Manufacturing, Volume 35, October 2015, pp. 126-136

[15] Xin Li, Felix T.S. Chan, S.H. Chung, "Optimal multi-degree cyclic scheduling of multiple robots without overlapping in robotic flowshops with parallel machines, Journal of Manufacturing Systems," Volume 36, July 2015, pp. 62-75

[16] Hariharan Ananthanarayanan, Raúl Ordóñez, "Real-time Inverse Kinematics of (2n+1) DOF hyper-redundant manipulator arm via a combined numerical and analytical approach,"Mechanism and Machine Theory, Volume 91, September 2015, pp. 209-226

[17] Hamid Toshani, Mohammad Farrokhi, "Real-time inverse kinematics of redundant manipulators using neural networks and quadratic programming: A Lyapunov-based approach,"Robotics and Autonomous Systems, Volume 62, Issue 6, June 2014, pp. 766-781

[18] Tamar Flash, Yaron Meirovitch, Avi Barliya, "Models of human movement: Trajectory planning and inverse kinematics studies,"Robotics and Autonomous Systems, Volume 61, Issue 4, April 2013, pp.330-339