MANIPULATING REAL-TIME ROBOTIC ARM MOVEMENTS USING PC BASED USER INPUT

K. Y. R. Kahawala[1], W. M. H. C. Wickramasinghe[1], P. P. Galhena[1] , L. Y. D. P. Wickramasinghe[1], P. C. Niresh[1], R. M. T. P. Rajakaruna[1]

1 Department of Mechatronics, Faculty of Engineering, South Asian Institute of Technology and Medicine (SAITM), Sri Lanka.
E-mail: thilini.r@saitm.edu.lk

ABSTRACT

Training a robotic hand to complete a sequence of movements with precision, particularly in time sensitive applications, is a complex task. As an alternative, this paper presents a real-time robotic arm control system, driven through PC based user input. Controlling the movements of a robotic hand with four degrees of freedom was tested using a mouse input captured via a PC in real-time. The user registers the required position of the robotic arm end-effecter by continuous motion of mouse, which is free to move inside a 2D frame. Mouse pointer locations in 2D plane are continually recorded and transferred to arm control unit to drive the actuators. A third mouse input is used to position the robotic arm in a semi-spherical space.

Key Words: Robotic arm, Motion control, Real-time maneuvering

1. INTRODUCTION

Robots are increasingly being used to enhance the efficiency of production and to substitute humans from risky and repetitive tasks. Some hazardous, meticulous and time consuming work such as, welding in hazardous conditions, arranging goods in shelves in storages can be undertaken by robots [1,2].

There are several types of robotic arms currently employed in a wide range of industrial applications [3]; namely - Cartesian, cylindrical, polar and articulated robotic arms. An articulated robotic arm is the most popular type and has a higher DOF (Degree of Freedom), small size and wide operation range, as well as capability to avoid obstacles within a small space.

There are robotic arms which could do these particular activities such as arc welding robots, pick and place items, painting, assembling and testing in car manufacturing companies. Robots have many benefits in healthcare, surgical and medical industries. Robot arms are used both for drugs carrying, manufacturing medicines or carrying out simple tasks in certain surgeries. Robotic arms are developed with a modular gripper (so that the robot is not limited to a particular task but to any situation whether it's to weld, lift etc, depending on the gripper attached), which can operate autonomously with an executable code or manually by processing the input given by a human, at a given time to move in a 3D plane. Due to the rising weight toward the complexity and precision of the robot arm movements, it is easier to deploy robotic arms by controlling its motion using a pc interface and training the robot to follow a complex motion [4,5,6]. This is particularly useful in training a robot to follow a complex motion sequence or for remote operations.

This paper proposes a motion control system for a robotic arm using real time control inputs given via a computer mouse. A robotic arm with four degrees of freedom was used to test the control system for a sequence of activities.

The rest of the paper is organized as follows: Section 2 presents the proposed robotic arm control system and the results and specifications are outlines in Section 3. Limitations and applications are discussed in Section 4.

2. BACKGROUND

The robotic arm with four degrees of freedom was used to test the real-time implementation of the control system. Figure 01 illustrates the side view of the arm, while the top view is given in Figure 02. It is driven by five motors as outlined below.

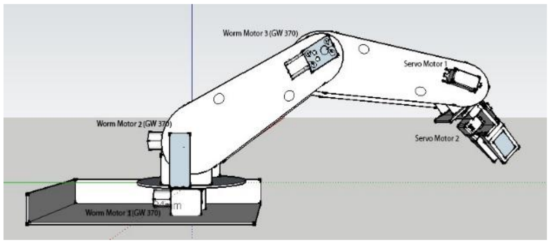

Figure 1: Side View of the Robotic arm used to test the control system

The positions of the 5 motors are at (Figure 1):

- Axis 1: for horizontal movement of the arm.
- Axis 2: for vertical movement of the lower arm.
- Axis 3: for vertical movement of the upper arm.
- Axis 4: for moving the wrist in clockwise and anticlockwise direction.
- Axis 5: for opening and closing the palm of the gripper.

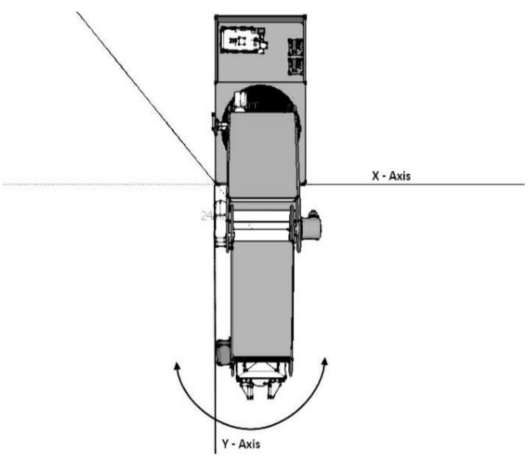

Figure2: Top View of the arm, indicating the X-Y plane.

Based on the arm layout, the positioning of the arm in a 2D frame was studied to identify inverse kinematic equations to manipulate each segment of the arm such that the end-effector reaches the desired x-y coordinate point. Figure 03 outlines the geometry of the arm in the 2D plane and the key parameters used in the inverse kinematic equations.

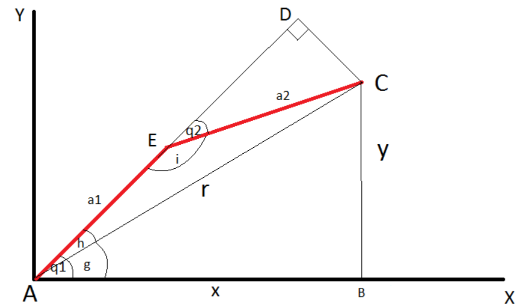

Figure 3: Robotic arm reach in X-Y Plane for a given coordinate point

As illustrate in Figure 3, q1 is the angle of the back arm against x - axis, while q2 is the angle measured between the back arm and the forearm. The red line represents the current position of the robot arm. Given the coordinate point of the destination (x,y) q1 and q2 were calculated as follows:

$$\mathbf{q1} = \tan^{-1}\frac{\mathbf{y}}{\mathbf{x}} + \tan^{-1}\left(\frac{\mathbf{a2.\,sin(q2)}}{\mathbf{a1 + a2.\,cos(q2)}}\right) \qquad (01)$$

$$\mathbf{q2} = -\cos^{-1}\left(\frac{\mathbf{x^2+y^2-a1^2-a2^2}}{\mathbf{2.a1.a2}}\right) \qquad (02)$$

Axis 1 is controlled by the scroller or the mouse buttons:

$$\mathbf{\tan q3} = \frac{\mathbf{z}}{\mathbf{x}} \qquad (03)$$

Accordingly the reach of the arm in 2D space was calculated and presented below in Figure 4.

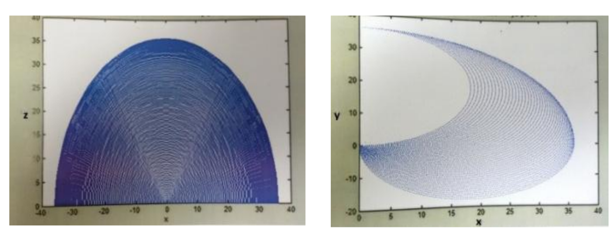

Figure 4: Movements in the Z-X Plane and X-Y Plane.

3. MOTION CONTROL

The robotic arm control system is presented in Figure 5 below. The user input is given through the PC and sent to the control unit, which will update the positioning of the robotic arm accordingly and the feedback system is sued to verify the movements of the robotic arm.

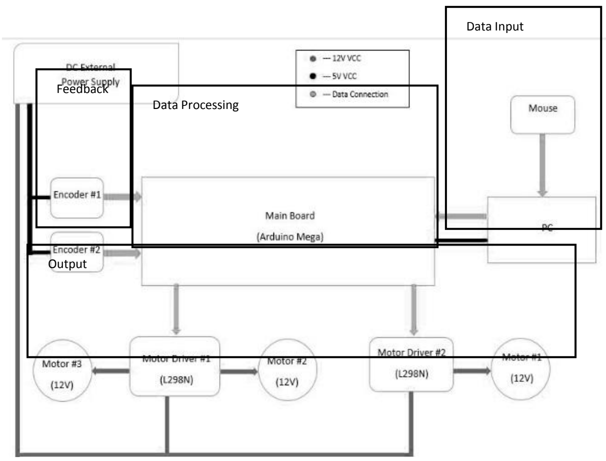

Figure 5: Block Diagram of the Hardware

Initially we had to find a way to capture data in the x, y and z directions from the mouse movements, as this concept works on 3D scale. Mouse movements were captured using software written on Java IDE Processing® [7]. Mouse movements in the horizontal plane were mapped to coordinates in X-Y plane and mouse scroller was used to indicate the desired Z axis coordinate.

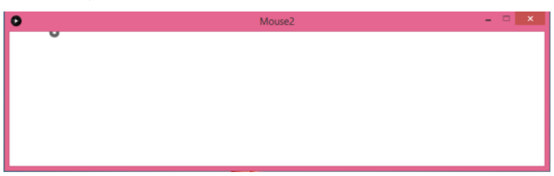

Figure 7: GUI Interface – Data Capture

For example raw data taken from mouse positions for the above path is mentioned below.

These data is order of (x : y : z);

(2 : 52:0,4 : 51:0,7 : 48:0,12 : 47:0,14 : 45:0,17 : 45:020 : 44:0,21 : 43:022 : 43:0,23 : 42:0,24 : 42:0,25 : 42:0,25 : 41:0,27 : 41:0,28 : 41:0,29 : 39:0,30 : 39:0,31 : 38:0,34 : 38:0,36 : 37:0,39 : 34:0,42 : 31:0,45 : 28:0,48 : 26:0,52 : 21:0,55 : 18:0,58 : 17:0,59 : 16:0,60 : 15:0,61 : 13:0,62 : 12:1,64 : 11:1,65 : 10:1,67 : 7:1,68 : 6:1,71 : 4:1,72 : 1:10)

The control system translates the input given via the PC interface to corresponding values of q1and q2, in order to identify the next position of the robotic arm.

Throughout the robotic arm we have used worm wheel motors instead of servo motors and stepper motors because worm motors are self-locking and they can be used to reduce speed and increasing torque smoothly. We had to ensure that the robot is steady and it can handle its weight.

4. RESULTS AND CONCLUSION

In this paper, we present a real-time robotic arm control system, driven through PC based user input. A robotic hand with four degrees of freedom was used to test the mouse input given in real-time to manipulate the robotic arm. Encoder based feedback mechanism was implemented to verify the positioning of each joint and the control unit continually updates the arm orientation according to user input. The sequence of motion given by a user could be recorded to train the robotic arm to carry out a certain task repeatedly.

5. REFERENCES

[1]. N.L.S. Verma . International Journal of Engineering and Technical Research (IJETR) ISSN: 2321-0869, vol.2, no.11, 2014.

[2]. M. Blackmore, J. Furniss, and S. Ochsner," *5 DOF Robotic Arm"*, ECE 478/578 Embedded Robotics, Portland State University, Fall 2009.

[3]. K. Kosuge, M. Koga, K. Furuta and K. Nosaki, *"Coordinated motion control of robot arm based on virtual internal model," Robotics and Automation*, Proceedings IEEE International Conference on, Scottsdale, AZ, vol.2, pp. 1097-1102. 1989

[4]. T. Kawamura, K. Hisatsune, M.U Matsusaka, and S. Kawamura, "*Adaptive motion control of a robotic arm with movable counterweights,"* 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Besacon, pp. 882-887, 2014.

[5]. D. Bassily, C. Georgoulas, J. Guettler, T. Linner and T. Bock, "*Intuitive and Adaptive Robotic Arm*

Manipulation using the Leap Motion Controller," ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of, Munich, Germany, pp. 1-7, 2014.

[6]. J. Choi and E. Amir, *"Factor-guided motion planning for a robot arm,"* IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, pp. 27-32, 2007.

[7]. Processing ide- https://processing.org/