Human Machine Interface Design for a 3 DoF Robot Manipulator

Fatih Cemal Can*, Önder Lapçin, Burak Ayan, Mehmet Çevik

Faculty of Engineering and Architecture, Mechatronics Engineering, İzmir Kâtip Çelebi University, İzmir, 35620, Çiğli, Turkey

Copyright©2018 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract This paper represents a Human Machine Interface (HMI) design to control a 3 DoF robot manipulator. This manipulator has two parallelograms to make the moving platform always parallel to the ground. We used inverse kinematic analysis of the robot manipulator to control the end point location. Inverse kinematic results are verified using design parameters and end effector location. According to our algorithm, user defines the end point location from HMI, and then program solves inverse kinematics of the robot manipulator. The angles are sent to Arduino microcontroller to set the position of the servo motors. Using this HMI, the user picks and places the object in real time. The user can also give command to draw linear, circular and rectangular paths on the HMI.

Keywords Arduino, 3 DoF Robot Manipulator, Inverse Kinematics, Human Machine Interface, Visual C#.

1. Introduction

The world industry has developed rapidly since the beginning of the first industrial revolution. This evolution still continues and now 4th industrial revolution is started. In the last revolution, automation lines must be designed to manufacture flexible products which are specified by customer needs. In planning of automated production lines, engineers need various designs of robot manipulators for picking and placing, painting, welding, assembling and similar tasks. Therefore, the robot manipulators are the most important part in flexible automated lines. According to task of the robot manipulator, motion of the end effector can be spatial or planar.

Huang et. al [1] designed and synthesized a two DoF parallel robot manipulator for pick-and-place operations. This parallel manipulator consists of two parallelograms to make moving platform always parallel to the ground link.

Five bar two DoF parallel manipulator is investigated for performance evaluation by Feng et. al [2]. The

performance evaluation is realized by using global conditioning and velocity indices.

Other two DoF parallel manipulator structure using two sliding joint is researched by Xin-Jun Liu et. al [3]. These sliding joints are vertically aligned to move platform of the manipulator.

The Diamond robot which is a kind of two DoF parallel robot manipulator structure is represented by Huang et. al [4]. In this study, the moving platform path generation is planned considering joint torque and velocity constraints.

Well shaped workspace optimization of two DoF parallel robot manipulator is carried out by designing link parameters in study of Huang et. al [5]. Optimal manipulator configuration in terms of force-transmission behavior and isotropy is the first step of their approach. The second step consists of determining independent kinematic parameters by optimizing of global index.

Dynamic analysis of two DoF parallel manipulator is used to investigate joint clearances of the manipulator joints in investigation of XuLi-xin and LiYong-gang[6]. Trajectories of the end effector for both ideal and clearance joints are calculated in computer and then the results are plotted to observe difference on the end effector trajectory.

A new measuring mechanism is designed and tested for zero offset calibration of a similar 2-DoF parallel robot manipulator by Jiangping et. al[7]. Results of calibration mechanism are illustrated and discussed in plots.

In this study, we developed a HMI for a 3 DoF robot manipulator which was previously investigated 2 DoF kinematic structures in [1,4,5,6,7]. All links of our manipulator mechanism is manufactured by using 3D printer. Here, our purpose is to control the end of the robot manipulator from a graphical user interface by giving positions. In order to do this, the inverse kinematic of the robot manipulator is solved analytically. The inverse kinematic analysis is verified by using some specific positions for the end effector. Using inverse kinematics, the interface redraws the manipulator links for each end effector position and robot manipulator moves simultaneously. The user of the manipulator can select different paths such as linear, circular and rectangular paths by clicking buttons in the graphical user interface. Using

our interface, one can determine the path of the end effector and easily make pick-and-place operations.

2. Robot Manipulator Prototype Structure and HMI Design

Our design of the robot manipulator is shown in Figure1. Manipulator moves linearly along z axis using a stepper motor connected to a trapezoidal lead screw linear mechanical element. Other two translational movements in x-y plane are obtained by closed kinematic chain which is depicted in Figure2. The links for this two degree of freedom closed mechanism are manufactured from 3D printer.

Figure 1. Three DoF Robot Manipulator

Figure 2. Two DoF closed kinematic chain (a part of the manipulator)

2.1. Inverse Kinematics of the Manipulator

Inverse kinematics of the manipulator must be solved to control motion of the end effector. Our algorithm based on inverse kinematic solution in Visual C# and then sends the data from HMI to arduino that controls two servo motors of the manipulator.

Figure 3. Kinematic diagram of the closed kinematic chain

Vector loop closure equation of one side is derived as follows,

$$\mathbf{O}_0\mathbf{O}_2 + \mathbf{O}_2\mathbf{O}_4 + \mathbf{O}_4\mathbf{K} + \mathbf{K}\mathbf{P} = \mathbf{O}_0\mathbf{P} \quad (1)$$

If Equation (1) is described using Euler notation, vector equation will be obtained in (2).

$$L_1 e^{i\theta_1} + L_3 e^{i\theta_3} + k/2 + is = P_x + iP_y \quad (2)$$

Rearranging (2), we get (3) and conjugate of this equation (4) as follows,

$$L_3 e^{i\theta_3} = P_x + iP_y - L_1 e^{i\theta_1} - (k/2) - is \quad (3)$$

$$L_3 e^{-i\theta_3} = P_x - iP_y - L_1 e^{-i\theta_1} - (k/2) + is \quad (4)$$

Multiplying (3) and (4), the angle θ_3 is eliminated. After multiplying these equations, (5) is obtained as follows,

$$e_3 + e_2 \cos \theta_1 + e_1 \sin \theta_1 = 0 \quad (5)$$

where $e_1 = 2L_1(-P_y + s)$, $e_2 = L_1(k - 2P_x)$ and

$$e_3 = L_1^2 + \frac{1}{4}(k - 2P_x)^2 + (P_y - s)^2 - L_3^2.$$

Using half tangent relations, $\cos \theta_1 = \frac{1 - t_1^2}{1 + t_1^2}$,

$\sin \theta_1 = \frac{2t_1}{1 + t_1^2}$ and $\tan \theta_1 = t_1/2$, trigonometric

equation in (5) can be converted to a single unknown polynomial equation given in (6).

$$t_1^2(e_3 - e_2) + t_1(2e_1) + (e_2 + e_3) = 0 \quad (6)$$

Solution of this polynomial equation is described in (7). Two real or imaginary results can be obtained. Imaginary results are unusable. Therefore, $discr_1 = e_1^2 - e_3^2 + e_2^2$ must be always positive for physically realizable mechanism.

$$t_{1,2} = \frac{-e_1 \pm \sqrt{e_1^2 - e_3^2 + e_2^2}}{(e_3 - e_2)} \quad (7)$$

Lastly, the first actuation angle is calculated as follows,

$$\theta_{1,2} = 2\arctan(t_{1,2}) \quad (8)$$

On the other side of closed kinematic chain, the second actuator position is solved by using similar procedures and similar equations. The second equation is given in (9).

$$\mathbf{O}_0\mathbf{O}_1 + \mathbf{O}_1\mathbf{O}_3 + \mathbf{O}_3\mathbf{O}_5 + \mathbf{O}_5\mathbf{K} + \mathbf{K}\mathbf{P} = \mathbf{O}_0\mathbf{P} \quad (9)$$

$$e_6 + e_5 \cos \theta_2 + e_4 \sin \theta_2 = 0 \quad (10)$$

where $e_4 = 2L_2(-P_y + s)$, $e_5 = L_2(2d - k - 2P_x)$ and $e_6 = L_2^2 + \frac{1}{4}(-2d + k + 2P_x)^2 + (P_y - s)^2 - L_4^2$. Similarly, using half tangent rules, polynomial is written as follows,

$$t_2^2(e_6 - e_5) + t_2(2e_4) + (e_6 + e_5) = 0 \quad (11)$$

Solution of polynomial equation is given in (12). Again, we have two results for the manipulator. For this loop, $discr_2 = e_4^2 - e_6^2 + e_5^2$ must be positive.

$$t_{2,2} = \frac{-e_4 \pm \sqrt{e_4^2 - e_6^2 + e_5^2}}{(e_6 - e_5)} \quad (12)$$

The second actuator angle of the robot manipulator is calculated in (13).

$$\theta_{2,2} = 2\arctan(t_{2,2}) \quad (13)$$

From kinematic analysis of the manipulator, four modes are available using results. However, we are using always the mode which is depicted in Figures 2 and 3 for our robot manipulator.

2.2. Verification of the Inverse Kinematics

Before programming the graphical user interface, the verification of the kinematic analysis is needed to be sure that our analysis is correct and usable. We assumed the links of the manipulator to be $L_1 = L_2 = 70\text{ mm}$, $L_3 = L_4 = 140\text{ mm}$, $d = 48\text{ mm}$, $k = 24\text{ mm}$, and $s = 1.5\text{ mm}$. The position of the end effector is considered to be $P_x = 100\text{ mm}$ and $P_y = 150\text{ mm}$. Substituting these values into kinematic equations, the parameters in (7 and

12) are calculated as $e_1 = -20790$, $e_2 = -12320$, $e_3 = -15096.3$, $e_4 = -20790$, $e_5 = -5600$ and $e_6 = 8952.25$. Using these parameters, we calculated polynomial parameters as $t_{1,1} = 1.44662$, $t_{1,2} = 0.06999$, $t_{2,1} = 1.9757$ and $t_{2,2} = 0.061711$. The results for actuation angles are calculated as follows, $\theta_{1,1} = 110.69^\circ$, $\theta_{1,2} = 8.01^\circ$, $\theta_{2,1} = 126.31^\circ$ and $\theta_{2,2} = 7.06^\circ$.

Figure 4. Four modes of the manipulator

2.3. HMI Design for the Manipulator

We designed our HMI (human machine interface) in Visual C#. NET environment. Before using this program, user must select COM port and baud rate of Arduino Microcontroller from all available ports listed in ComboBox. Then, Connect Arduino button must be clicked to start communication between HMI and Arduino. If the communication is successfully established, the progress bar will be full green color. If the communication fails, the program gives a message to connect Arduino correctly. Then user can send command from interface to Arduino. Using inverse kinematics in previous section, program calculates actuation angles and then draws the manipulator on the form. User changes the end effector location using mouse pointer. The manipulator moves simultaneously according to this mouse position change in inverse kinematics. Our interface design is shown in Figure 5.

Figure 5. Human Machine Interface Program prepared using Visual C#

Disconnect button can be used any time to give up communication between HMI and Arduino. Visual C# is event based programming language. Therefore, we used events in Visual C# as follows,

Start: Form1_Load event is working. In this event, link lengths and other constant parameters are defined.

Communication: Select appropriate COM port and baud rate from ComboBox and then click Connect button.

Input: Form1_Mousedown, Form1_MouseMove and Form1_MouseUp events are working. These events wait for a new input on HMI.

Calculate: discr1 and discr2 are calculated according to inverse kinematic analysis of the robot.

Are roots of discr1 and discr2 real?: Program must decide that the results are physically realizable or not. Non real results cannot be used in real environment. Therefore, they are unnecessary.

Calculate: If roots are real, angles are calculated using inverse kinematics equations (7, 8, 12 and 13).

Prepare and send data packages to Arduino: When two angles are calculated, they are packaged in order for sending. The data package includes plus and minus symbols which show the direction of the rotation, and two digit numbers which indicate magnitude of the rotation.

One Data Package:

+	1	7	9	-	0	5
---	---	---	---	---	---	---

Figure 6. Flowchart for Visual C#

Fig. 6 shows the flowchart of inverse kinematic based control procedure. Furthermore, Arduino code working procedure is depicted in Fig.7. According to these procedures, data package is generated in Visual C# code and then this package is evaluated in Arduino code. Type of data package is string containing 7 characters. Arduino code divides this string into two parts for servo motor angles. Then, the code converts string variables into

integer.

Figure 7. Flowchart for Arduino

Figure 8. Robot Manipulator and Electronic Control Circuit setup

3. Three DoF Robot Manipulator Setup and Test Results

The setup of our 3 DoF robot manipulator is shown in Figure 6. Electro magnet end effector is connected to the moving platform of the manipulator to pick a metal object and place a position in workspace of the manipulator. Electro magnet is controlled by using a button to grasp or release a metal object. Two Arduino microcontrollers are used to command servo and step motors simultaneously. Arduino micro is chosen to control stepper motor whereas Arduino Uno is used to control servo motors and electro magnet. The links lengths of the manipulator are designed to be $L_1 = L_2 = 70\text{ mm}$, $L_3 = L_4 = 140\text{ mm}$, $d = 48\text{ mm}$, $k = 24\text{ mm}$, and $s = 20\text{ mm}$.

Firstly, we tested our robot manipulator using HMI for circular and linear paths. Our test results are depicted in Figure 9. Figure 9 (a),(b) and (c) shows linear path which is drawn by the robot manipulator setup whereas Figure 9 (a1),(b1) and (c1) illustrates linear path which is generated by HMI program. As seen from these figures, our manipulator end effector moves very close to given linear path from HMI. The second path is circular for manipulator test. Similarly, the manipulator movements are shown in Figure 9 (d), (e) and (f), simulation of the movement is shown in (d1), (e1) and (f1). Consequently, our manipulator end effector can follow circular path.

Other test for our manipulator is to pick and place a plastic solid object which is assembled with a metal bolt for grasping. Electro magnet was activated by using button. Nearly a rectangular path is defined by user. The starting position of the solid object is seen in Figure 10 (a). The object was moved up in horizontal direction illustrated in Figure 10 (b). Then, it was translated horizontally as shown in Figure 10 (c). Finally, the object was moved down vertically to ground in Figure 10 (d).

All these four positions and path are depicted on HMI in Figure 10 (f).

4. Conclusions

3 DoF robot manipulator is designed and manufactured in this study. Inverse Kinematic analysis is analytically solved. Inverse kinematic analysis is verified using some design parameters. Human machine interface program is developed based on inverse kinematic analysis. Our manipulator cost is low due to 3D printing prototyping method in this project (all cost is nearly 120 \$). Therefore,

the manipulator can be manufactured by engineering students to program similar type of manipulators.

The manipulator will be manufactured from metal for industrial pick-and-place robot in the future. Of course, the cost will be increased due to industrial components. However, our procedure can be applied to industrial tasks using robot manipulator. This industrial robot will be programmed by using same control idea. Training and learning algorithms will be added to working HMI procedure. User will be able to plan and manage motion of the manipulator for a desired industrial task. Accuracy and precision of the manipulator will be measured in its workspace.

5. Supplements

You can watch working the video of the manipulator which is given in link: <https://www.youtube.com/watch?v=hVkBj0UeEIA>

Figure 9. Linear and Circular Path Test (a),(b) ,(c) real and corresponding (a1), (b1),(c1) HMI positions for a line, (d),(e), (f) real and corresponding (d1), (e1),(f1) HMI positions for a circle.

Figure 10. Pick and Place Task completed by using robot manipulator. (a) Starting position (b) The first middle position (c) The second middle position (d) End position (f) Path in HMI

Acknowledgements

This work is supported by Scientific and Technological Research Council of Turkey (TUBITAK) under 2209 Support for graduation project toward industrial application with 1139B411701814 project number.

REFERENCES

- [1] Huang, T., et al. Optimal design of a 2-DOF pick-and-place parallel robot using dynamic performance indices and angular constraints. *Mechanism and Machine Theory*, 2013, 70: 246-253.
- [2] Gao, Feng; Liu, Xinjun; Gruver, William A. Performance evaluation of two-degree-of-freedom planar parallel robots. *Mechanism and Machine Theory*, 1998, 33.6: 661-668.
- [3] Liu, Xin-Jun; Wang, Qi-Ming; Wang, Jinsong. Kinematics, dynamics and dimensional synthesis of a novel 2-DoF translational manipulator. *Journal of Intelligent and Robotic Systems*, 2005, 41.4: 205-224.
- [4] Huang, T., et al. Time minimum trajectory planning of a 2-DOF translational parallel robot for pick-and-place operations. *CIRP Annals-Manufacturing Technology*, 2007, 56.1: 365-368.
- [5] Huang, T., et al. Optimal kinematic design of 2-DOF parallel manipulators with well-shaped workspace bounded by a specified conditioning index. *IEEE Transactions on Robotics and Automation*, 2004, 20.3: 538-543.
- [6] Li-Xin, Xu; Yong-Gang, Li. Investigation of joint clearance effects on the dynamic performance of a planar 2-DOF pick-and-place parallel manipulator. *Robotics and Computer-Integrated Manufacturing*, 2014, 30.1: 62-73.
- [7] MEI, Jiangping, et al. Rapid and Automatic Zero-Offset Calibration of a 2-DOF Parallel Robot Based on a New Measuring Mechanism. *Strojnicki Vestnik/Journal of Mechanical Engineering*, 2017, 63.12.