International Journal of Advanced Computational Engineering and Networking, ISSN(p): 2320-2106, ISSN(e): 2321-2063
Volume-6, Issue-9, Sep.-2018, http://iraj.in

PROGRAMMED CONTROLLER FOR A RIGID SCARA ROBOT LINKS

[1]MARWAN ALI, [2]MILAN SIMIC

[1,2]School of Engineering, RMIT University, Melbourne, VIC, 3083, Australia
E-mail: [1]Marwan@rmit.edu.au, [2]Milan@rmit.edu.au

Abstract - The Selective Compliance Assembly Robot Arm (SCARA) robot arm has a complex mechanism as it tries to imitate the human arm for performing tasks. In this paper, the implementation of a PID controller using visual C# environment is presented. Hardware and software solutions are chosen is to keep the costs at a minimum. A GUI is designed to monitor and control the robot arm in real-time. The final controller performances are monitored and analyzed. This opens a new, innovative, inexpensive and flexible environment for controller implementation.

Index Terms - SCARA robot; control; PID; visual C#

I. INTRODUCTION

Industrial robots perform a fundamental role in modern day manufacturing systems. The utilization of robots in industry has improved the production capacity and quality of manufactured products. SCARA robots offer certain advantages on account of their unique structures. Their performances are faster, cleaner and more precise operations, compared to other robot types. In order to design a complex control system, utilization and expansion of hardware electronic circuit is obligatory. However, increase in hardware will cause the cost to increase. In order to economize on financial resources, visual programming and software interfacing techniques are employed to replace hardware electronic circuits.

The main objective of this research is to develop simple, yet efficient control method and end-user software for SCARA robot manipulators. Further, it should be possible to monitor and, if needed, modify parameters on-line. This operation was implemented in visual C#.net which is a flexible and economical software environment. Finally, the design incorporates a simple and popular controller with visual end-user program which can be adapted to other SCARA robots. The specific research objectives are

• To design a digital PID controller to control the position and motion of the SCARA robot,

• To develop a graphical user interface (GUI) using C#.Net software to control and monitor online parameters of the SCARA robot, and

• To allow on-line modifications to the proposed controller for different situations and operating conditions.

II. BACKGROUND

Sankyo Seiki et al. suggested a new concept for assembly robots. The robot was expanded under the guidance of Hiroshi Makino, and the robot was named SCARA [1].The term Degrees of Freedom (DOF) has been adopted to describe the number of ways a robot can move. There are many types of mechanisms used in the industry such as SCARA, Articulating, Spherical, Cylindrical and Cartesian configurations and some other types employed in industry. An articulated robot is a robot with rotary joints. Articulated robots can range from simple two-jointed structures to systems with 10 or more interacting joints. Spherical robots are composed of an arm with two rotary joints and one prismatic joint. The axes of a spherical robot form a polar coordinate system. They are used to handle machine tools and arc or spot welding. Cylindrical robots operate in a cylinder-shaped space or coordinate system and have at least one rotary joint and at least one prismatic joint. Cartesian robots have three prismatic joints whose axes are coincident with a Cartesian (x, and y) coordinate system [2].

In a SCARA Robot, the manipulator arm can basically move within two major axes of x and y, referring to base movement, vertical height and horizontal addition. Different manipulator configurations are available as rectangular, cylindrical, spherical, revolute and horizontal jointed [3, 4].

III. SCARA ROBOT CONTROL

Karan attempted to design a graphical user interface for PC-based control of a robot arm. The task undertaken in this study was to redevelop the computer interface of a Selective Compliance Robotic Arm through Visual Basic [5].

PID controller is the most popular controller used in process industries. The algorithm is simple; however, it can provide excellent control performance despite variation in the dynamic characteristics of a process plant. The PID controller was first placed on the market in 1939 and has remained the most widely used controller in process control until today [6].

A. Original Prototype

Fig. 1 shows the SCARA robot used in this project. It consists of two links, a bracket, a base and an end-effector. The end-effector consists of two DC motors, and its close-up view is shown in Fig. 1. The SCARA robot needs a base for holding the body. In total, SCARA robot consists of 4 DC motors (M1 – M4) and an aluminum frame. Two DC motors, M1 and M2, are used to control link 1 and link 2. The other two DC motors, M3 and M4, control the end effector.

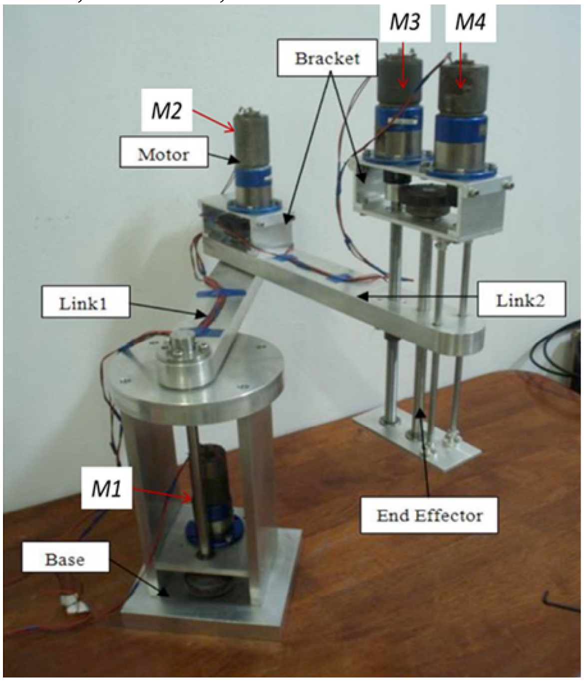

Fig 1 The SCARA Robot [7]

B. Hardware

There are few hardware components employed in this research. Data Acquisition (DAQ) is employed as interface between robot and the host controller, which is a laptop in this case. Signals from sensors mounted on the SCARA robot are feed back to the DAQ/USB6009, bus-powered multifunction DAQ USB Device, capable of performing analog and digital inputs and outputs. Microcontroller, PIC16F877A, is employed to generate Pulse Wide Modulation (PWM) signal for the motor driver.

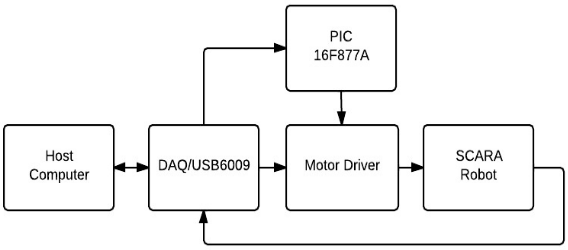

Fig 2 System Hardware Block Diagram

C. Operation

Task, defined for this robot, is moving along coordinate axes, according to user's desired points. GUI can take the desired point from the user actions.

PID controller is implemented in four steps. The first step is to get the present angles and position of the robot. The second step is to obtain the error and apply PID terms into controller, while the third step involves getting feedback control and making double correction of position in order to have precise movement. Finally, the fourth step is to plot the PID-Error chart using the angle in the time domain. PID controller code is written as a method (function) in C#. After performing data acquisition by using NI-DAQ device USB6009, and calculating the kinematics of both links, the bulk of the program calls a PID method, i.e. function. The error variable passes to the PID method and the result of the PID calculation is returned to the main program. There is another method which is also called by the main body of program; it is PLOT method which runs concurrently with the PID method for plotting the error base in time and angle.

D. Forward Kinematics

Forward kinematics analysis of SCARA robot was performed in order to find the mapping between the joint displacements and the end-effector position with respect to the base border. Kinematic parameters are determined in accordance with Denavit-Hartenberg conventions method. The position vector $\bar{P}$ is given by Equation (1), [8, 9].

$$\bar{P} = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \begin{bmatrix} L_1 c\theta_1 + L_2 c\theta_2 \\ L_1 s\theta_1 + L_2 s\theta_2 \\ -d_4 + d_1 + d_3 \\ 1 \end{bmatrix}$$

(1)

Where P_x, P_y and P_z denote the elements of the position vector in x and y axes, $d_{1,\ 3,\ 4}= 0$ distant from zero position of z axes. L_1 and L_2 are the lengths of the elbows, as fully detailed in reference [9] and shown in Fig 3.

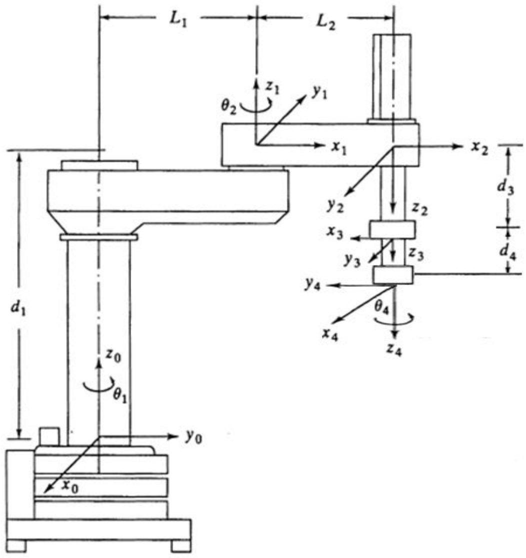

Fig 3 Free Body Diagrams of the SCARA Robot Links [10]

E. Inverse Kinematics

While direct kinematics sets up a relationship between the joint space and Cartesian space, inverse kinematics determine the entire possible joint variables which will bring the end-effector to the desired position and orientation [10]. Inverse kinematics is concerned with the inverse problem of finding the joint variables in terms of the end-effector position and orientation. It is, in general, more difficult than the forward kinematics problem. Shown in Fig. 3 is the free body diagram of links 1 and 2 on the XY plane.

The inverse kinematics equations of the SCARA robot for the angles of link 1 and link 2 were found to be [10]

$$\theta_1 = atan2\left(\frac{(l_1+l_2 cos\theta_2)P_y-(l_2 sin\theta_2)P_x}{l_1^2+l_2^2+2l_1l_2cos\theta_2} \; \frac{(l_1+l_2cos\theta_2)P_x+(l_2sin\theta_2)P_y}{l_1^2+l_2^2+2l_1l_2cos\theta_2}\right) \quad (2)$$

$$\theta_2 = cos^{-1}\left(\frac{P_x^2+P_y^2-l_1^2-l_2^2}{2l_1l_2}\right) \quad (3)$$

F. Transfer Function of SCARA Robot

Transfer function is derived and the rest of the response curves are already analyzed and presented in [7]. The discrete time Transfer function from torque input "u" to position output "y" for link 1 and link 2 are given by equations (4) and (5)

$$G_1(z) = \frac{-0.00024\,z+0.003799}{z^2-0.9009\,z-0.09921} \quad (4)$$

$$G_2(z) = \frac{-9.641e-0.005z+0.002362}{z^2-0.9502\,z-0.04981} \quad (5)$$

G. PID Controller

PID control is based on three actions, Proportional, Integral, and Derivative. All PID functions are activated by changes in the process error. The block diagram with PID control is shown in Fig. 4.

The proportional, integral and derivative terms are summed to calculate the output of the PID controller. Defining $u(t)$ as the controller output, the final form of the PID algorithm is, [11].

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt}e(t) \quad (6)$$

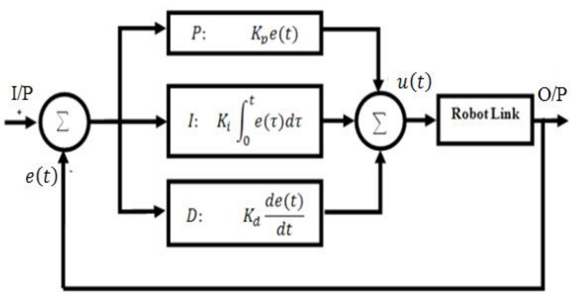

Fig 4 The Control Loop with the PID controller

Where K_p is the proportional gain, K_i is the integral gain, K_d is the derivative gain and e is the error. The values of the PID controller parameters used are presented in table I.

Table 1 PID controller Gains

Parameters	Link 1	Link 2
Kp	5.3089	5.1045
Ki	0.9514	0.9773
Kd	1.6250	1.7104

IV. DEVELOPING PROGRAM USING C#

Visual C# is an object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. The mathematical models were introduced and the kinematic problems were solved in previous section. In addition to that, PID terms and equations were given. It should be noted that the theoretical issues will not be useful unless they are altered to executable parameters for digital processing. Calculations of SCARA position were done and the values of θ_1 and θ_2 are extracted from the Equations (2) and (3) respectively. Flowchart for calculation of θ_1 and θ_2 is shown in Fig. 5.

A. PID Implementation

The output of the PID that is generated from the PID subroutine in C# goes into the PWM generator. The PWM is generated by PIC16F877A microcontroller and then the output of the PWM is written to the motor driver to generate another PWM of the proper power levels to drive DC motors.

The terms of the PID controller are implemented in programming language visual C#. The PID calculation is derived using Equation (6). As already mentioned, PID code is written as a subroutine program and is called from the main program to proceed. The flow chart of PID execution in visual C# code editor is shown in Fig. 6.

In order to calculate the actual angles θ_1 and θ_2 of the SCARA robot, Equations (7) and (8) are employed.

$$M_{1,2} = \frac{V_0 - V_{90}}{90^\circ - 0^\circ} \quad (7)$$

$$\theta_{1,2} = \frac{V_i - V_0}{m} \quad (8)$$

Where V_i is the angle voltage input, V_0 is the voltage value when θ is 0° and V_{90} is the voltage value when θ is 90°. The GUI is shown in Fig. 7.

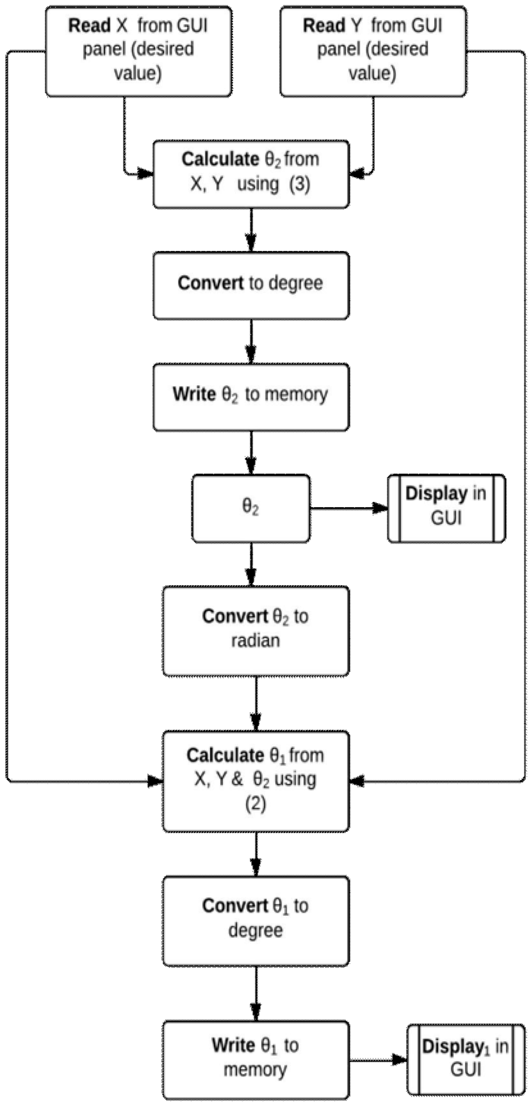

Fig 5 Flow chart for calculating desired θ_1 and θ_2

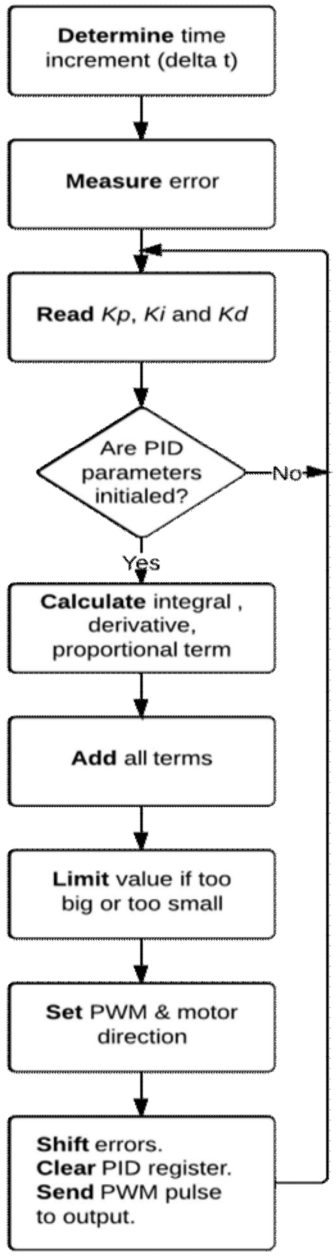

Fig 6 PID controller implementation

B. Real-Time Simulation

The responses of link 1 and link 2 with a PID controller are shown in Fig. 8 and Fig. 9. It is evident that the PID controller provides reliable control with reasonable accuracy. As it can be seen from the table II, the comparison between given graphs, which the overshoot takes, place when the link is close to the desired position. The time when the output reaches requested value, and does not leave a defined angle, is known as settling time which is given in the Table.

CONCLUSION

An open source GUI based on visual C#.Net has been developed for real-time control of a SCARA robot. A PID controller has been implemented without tuning. Testing demonstrates successful implementation and verification of the real time control system.

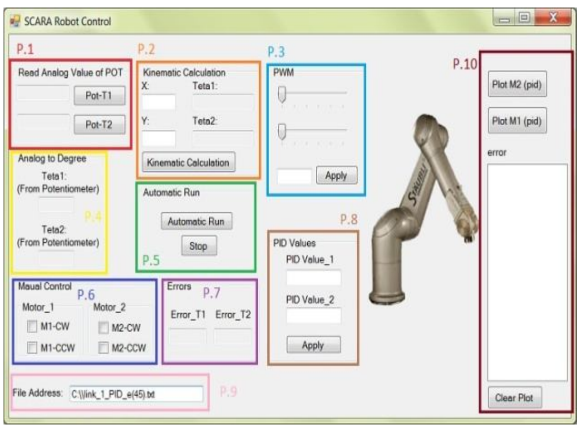

Fig 7 The GUI from C# Program window

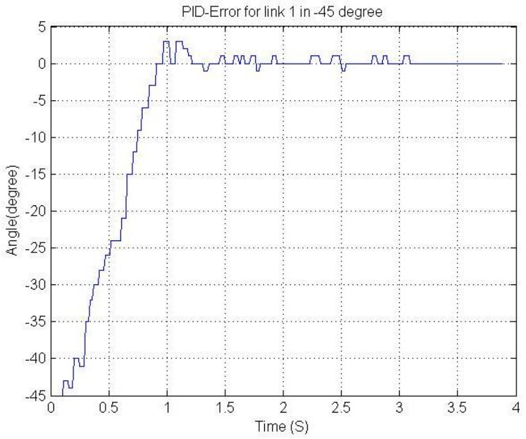

Fig 8 Link 1 error signal – initial angle of -45°

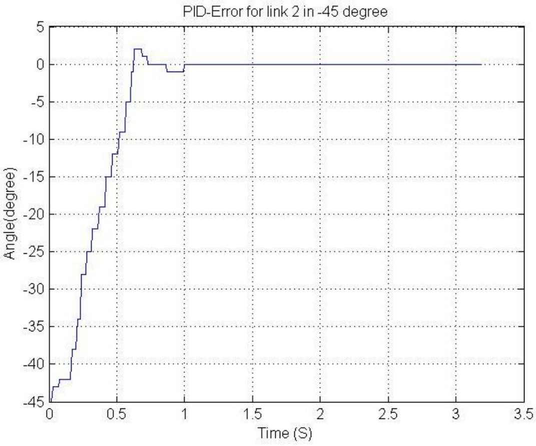

Fig 9 Link 2 error signal – initial angle of -45°

Table 2 Peak Overshoot and settling time for several initial positions

	Peak Overshoot (deg)		Settling Time (sec)	
Initial Position	Link 1	Link 2	Link 1	Link 2
45	-2	-3	2.15	0.88
90	-3	-2	2.37	1.71
161	-	-2	-	3.38
180	-3	-	3.32	-
-45	3	2	1.23	0.74
-90	4	1	1.74	1.76
-161	-	2	-	2.46
-180	4	-	2.81	-

REFERENCES

[1] Lars Westerlund, the Extended Arm of Man - A History of the Industrial Robot. Vol. 9177364678. 2000: Informationsförlaget.

[2] Marwan Ali, M.S., Fadi Imad, Calibration method for articulated industrial robots. Procedia Computer Science, 2017(122): p. 1610.

[3] Weria Khaksar, K.S.b.M.S., Firas B. Ismail, Moslem Yousefi, Marwan A. Ali. Runtime reduction in optimal multi-query sampling-based motion planning. in Robotics and Manufacturing Automation (ROMA), 2014 IEEE International Symposium on. 2014. Kuala Lumpur, Malaysia.

[4] A Marwan, F.N., KSM Sahari, S Hanim, Robust fuzzy MIMO bang-bang controller for two links robot manipulators. Australian Journal of Basic and Applied Sciences, 2012. 5: p. 2071-2083.

[5] Karan Desai, Development Of A Graphical User Interface For Control Of A Robotic Manipulatior With Sample Acquisition Capability, in Aerospace Engineering. 2012, Ryerson University: Toronto, Ontario, Canada,. p. 153.

[6] Shreeraman, H.O.B.R.S.P.R., PID Controller Tuning Techniques: A Review. Journal of Control Engineering and Technology (JCET), 2012. 2(4): p. 168-176.

[7] A Marwan, N.F., KSM Sahari, S Hanim, Real-time on line tuning of fuzzy controller for two-link rigid–flexible robot manipulators. Transactions of the Institute of Measurement and Control, 2011. 6(35): p. 730-741.

[8] A Marwan, F.N., KSM Sahari, S Hanim, I Fadi, On-line adaptive fuzzy switching controller for SCARA robot. WSEAS Trans. Syst. Control, 2011. 6: p. 404-416.

[9] Cubero, S., Industrial Robotics: Theory, Modelling and Control. 2006, ProGermany / ARS, Austria. Literatur Verlag, .MA Ali, I.F., KSM Sahari, K Weria, Y Moslem, R Izaizi. Trajectory Tracking Controller for flexible robot arm,. in Robotics and Manufacturing Automation (ROMA). 2014. Kuala Lumpur, Malaysia: IEEE International

[10] Claudio U, Juan C, Jose P, 'Design, construction and control of a SCARA manipulator with 6 degrees of freedom', Journal of Applied Research and Technology

[11] Submit your manuscript electronically for review.

★★★