Design and Development of a Cost-efficiency Robot Arm with a PLC-based Robot Controller

Vo Duy Cong
Industrial Maintenance Training Center
Ho Chi Minh City University of Technology
(HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

To develop a cost-efficient robot arm for a typical pick and place application that can applied in industry, this paper deployed a programmable logic controller (PLC) to control the rotation motion of the robot joints. The main tasks of the PLC controller are to calculate the kinematics, create high-speed pulse outputs for stepper motors, and implement sequence operations for a certain application. Functions are written into subprogram segments. When needed, the main program only turns on the corresponding flag for executing the subprogram. Using the pre-written subprograms, a logical sequence to implement the Pick and Place application is easily implemented and described in this paper. The PLC program is developed to control a SCARA robot with three rotation joins. Stepper motors drive the robot joints. The Delta DVPSV2 PLC is utilized to design the robot controller. This PLC series has four high-speed pulse output pins, which is suitable for this project. Synchronous motion of stepper motors is easily peformed using high-speed pulse output commands built into the PLC program. Experimental results of robot arm control have demonstrated the efficiency and accuracy of the developed program. The robot arm's forward and inverse kinematics problems are verified using the simulator on the software. The robot's joints move synchronously as required to perform pick-and-place applications.

Keywords: *Industrial Robot, Kinematics equation, Pick and Place, Programmable Logic Controller, Robot control, SCARA robot.*

1. INTRODUCTION

Starting from the need to reduce production costs and increase productivity, industrial robots have been invented to perform a number of production tasks that require precision and repeatability. Robots are gradually replacing humans in manufacturing industries because of a number of outstanding attributes, such as reliability, predictability, accuracy, repeatability, and the ability to operate in toxic and unsafe working environments [1-3]. Robot technology is also increasingly developing with the development of science and technology. The application scope of robots has been expanded in different areas of life. Therefore, robots have been de–signed and developed to meet the different requirements of each field.

Designing and developing a robot arm is a challenging task. It requires a wide range of knowledge about mechanical design, structural analysis [4-8], electronics and control [9-10], kinematic modeling and analysis, path planning, and trajectory tracking [11-13]. There have been many studies to develop robot arms used for many different purposes, from education, healthcare, and service to industry. Finite element analysis (FEA) is commonly used for structural optimization and analysis of robotic arms. Stress analysis of robot arm components and optimization structure is performed before manufacturing [14-17].Analyzing the robot's dynamics and kinematics is also necessary for optimizing robot performance [18-19].The forward kinematics problem is used to calculate the position and orientation of the robot end-effector from the values of joint angles. The Denavit-Hartenberg (DH) parameters are employed to establish the transform matrix between coordinate frames of robot joints.The solution of the inverse kinematics problem is the values of the robot joint angles corresponding to a certain pose of the robot end-effector [20-22]. Traje–ctory planning is the basis of robot motion control and involves the stability of robot motion, working effi–ciency, energy consumption, etc. [23-25]. Robot trajec–tory planning determines the final motion trajectory of the robot arm to meet processing requirements and ensure high performance by linking the spatial and temporal domains of the robot during movement. Optimized trajectory planning can help meet process requirements, reduce jitter, and improve structural quality to enable smooth motion [26-27].

The robot controller is a critical component of the robot system. It has a great influence on the perfor–mance, accuracy, and working ability of the robot. Robot controllers can be developed based on microcon–trollers, DSP chips, CNC controllers, embedded com–puters, FPGAs, or PLCs. In low-cost designs, micro controllers are commonly used. In [28], a 6-DOF robot arm is designed and controlled by a combination of a

Received: January 2024, Accepted: February 2024
Correspondence to: Vo Duy Cong
Industrial Maintenance Training Center,
Ho Chi Minh City University of Technology, Vietnam
E-mail: congvd@hcmut.edu.vn
doi: 10.5937/fme2402226C

computer and an Arduino board. In this study, the function of Arduino is only to control the rotation of DC servo motors for the robot joints. The motion planning of the robot is performed on the ROS. Arduino com–municates with ROS through a USB interface. Through a Bluetooth module, Arduino is connected to a mobile phone to control the motion of a robot arm [29]. An application is written on the mobile phone to send the rotation angles to the Arduino. Victor et al. [30] used the ESP32 microcontroller to develop an affordable IoT open-source robot arm for online teaching of robotics courses. ESP32 controls the robot's motion based on data received through Wi-Fi from an HMI interface deployed on a smart phone. In [31], the STM 32F407VGT6 microcontroller was chosen to control a 4-DOF parallel robot arm. The STM32 microcontroller created the PWM signal to control the speed and direction of DC motors. The STM32 communicated with the computer through a USB↔UART converter to receive the position and orientation of the robot end-effector. It can be seen that the microcontroller only performs simple tasks such as controlling the rotation of motors. Other complex tasks, such as kinematics and motion control, are performed on computers.

To develop a high-performance controller, DSP or FPGA technology is commonly used. In [32], a robot controller based on a DSP processor was designed and fabricated. The DSP processor calculates the rotation angles of DC motors from the 3D coordinate and creates signals to control the motion of DC motors. Lee and Jung [33] provided a tutorial for the control imple–mentation of a collaborative robot using digital signal processors(DSPs). The DSP processor performs the task of calculating kinematics and controlling motion by using the time-delayed control (TDC) method. In robo–tic systems with a large number of degrees of freedom, the kinematic calculation processes performed by DSP also become more complex and use large resources, thus reducing the robot's response speed. With the ability to calculate and process data in parallel, FPGAs are widely used in the field of robot control. FPGA helps improve hardware processing power and real-time information processing speed [34-40].Chen et al. [34] implemented the forward/inverse kinematics for a SCRARA robot on FPGA. The parameterized function is utilized to increase the code flexibility in the design, and then the Finite state machine is employed to reduce the hardware resource. FPGA-based motion control was developed by Chand et al. [38] for a dual-arm robot. The motion planning and control problem was solved by implementing Lyapunov-based acceleration controllers on FPGA. Gürsoy and Efe [40] proposed a proportion integral differential and sliding mode control scheme for robot manipulators based on FPGA, achieving better trajectory tracking performance.

Implementing FPGA for robot control is a difficult task that requires more knowledge about computation algorithms, hardware description language, and digital systems. Therefore, it is costly in terms of time and cost to develop a robot controller with FPGA. In this paper, a programmable logic controller (PLC) is employed for the purpose of developing a cost-effective robotic arm. PLCs are robust industrial electronic systems used for controlling a wide variety of mechanical systems and applications. PLC systems are found everywhere, inclu–ding in factories, office buildings, and even controlling the traffic on our streets. PLCs are at the very heart of the control of many critical technologies. Initially, PLC was designed to replace traditional mechanical relays and timers. However, with the development of electronic technology, the functions of PLCs have been expanded, and they can perform many different comp–lex tasks in the industry. Several studies have used PLCs to control the robot arm. In [41], PLC controls two servo motors of a gantry robot system. The PLC received the X and Y coordinates from the computer and calculated to transform them to the rotation angles of servo motors. In [42], PLC was employed to control a linear delta robot arm. The rotation angles obtained from the inverse kinematic analysis are transferred to PLC using OPC. The necessary coding in the TIA Portal was done to control the motor drivers so that the three stepper motors could ensure the delta robot moved to the desired coordinate position.

In this paper, a PLC program is developed to control a SCARA robot with three rotation joins. Stepper mo–tors drive the robot joints. The Delta DVPSV2 PLC isutilized to design the robot controller. This PLC series has four high-speed pulse output pins, so it is suitable for this project. PLC performs kinematics calculations to find the rotation angle of the robot joints and output high-speed pulse to stepper motor drives for rotating stepper motors to reach the required angles. Functions for kinematics calculation and controlling stepper mo–tors are written into subprogram segments. When needed, the main program only turns on the corres–ponding flag for executing the subprogram. A logical sequence to implement the Pick and Place application is easily implemented using the pre-written subprograms.

2. KINEMATIC EQUATIONS OF THE ROBOT ARM

Figure 1 shows the kinematic diagram of the robot arm that is used in this research. The robot consists of three rotation joints connected by three links with lengths of l_2, l_4 and l_6, respectively. To establish the homogeneous transformation matrix that expresses the position and orientation of the end effector frame in terms of the base frame, the Denavit-Hartenberg (DH) parameters are used. The DH parameter tables consist of four variables: two variables, θ and α, used for rotation; two variables, r, and d, used for displacement. From the DH parameters, the transformation matrix between frame n and frame n-1 is defined as:

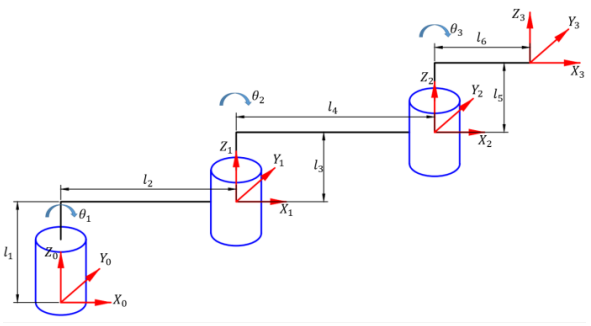


Figure 1. The kinematic diagram of the SCARA robot

$$T_n^{n-1} = \begin{bmatrix} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n\cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n\cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Table 1. The D-H parameters

Joint	θ_i(°)	α_i(°)	r_i(mm)	d_i(mm)
1	θ_1	0	l_2	l_1
2	θ_2	0	l_4	l_3
3	θ_3	0	l_6	l_5

Transform from frame 1 to frame 0:

$$T_1^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & l_2\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & l_2\sin\theta_1 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Transform from frame 2 to frame 1:

$$T_2^1 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & l_4\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & l_4\sin\theta_2 \\ 0 & 0 & 1 & l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Transform from frame 3 to frame 2:

$$T_3^2 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & l_6\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & l_6\sin\theta_3 \\ 0 & 0 & 1 & l_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

By multiplying the matrices together, we obtain the transformation matrix that expresses the relation between the end-effector frame and the base frame:

$$T_3^0 = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_6c_{123} + l_4c_{12} + l_2c_1 \\ s_{123} & c_{123} & 0 & l_6s_{123} + l_4s_{12} + l_2s_1 \\ 0 & 0 & 1 & l_1 + l_3 + l_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

with:

$$\begin{aligned} c_{123} &= \cos(\theta_1 + \theta_2 + \theta_3) \\ s_{123} &= \sin(\theta_1 + \theta_2 + \theta_3) \\ c_{12} &= \cos(\theta_1 + \theta_2); s_{12} = \sin(\theta_1 + \theta_2) \\ c_1 &= \cos\theta_1; s_1 = \sin\theta_1 \end{aligned} \quad (6)$$

From matrix T_3^0, the position and orientation of the end-effector can be extracted as follows:

$$\begin{aligned} x_p &= l_6c_{123} + l_4c_{12} + l_2c_1 \\ y_p &= l_6s_{123} + l_4s_{12} + l_2s_1 \\ z_p &= l_1 + l_3 + l_5 \\ \theta_p &= \theta_1 + \theta_2 + \theta_3 \end{aligned} \quad (7)$$

where (x_p, y_p, z_p) is the 3D coordinate and θ_p is the rotation angle around the Z-axis.

Forward Equation (7) determines the position and orientation of the robot end-effector from the robot joints angles. Contrarily, the inverse kinematic equations help calculate the joints' angles when we know the position and orientation of the end-effector. We use geometrical relations among links to find the inverse kinematic equations, as shown in Figure 2.

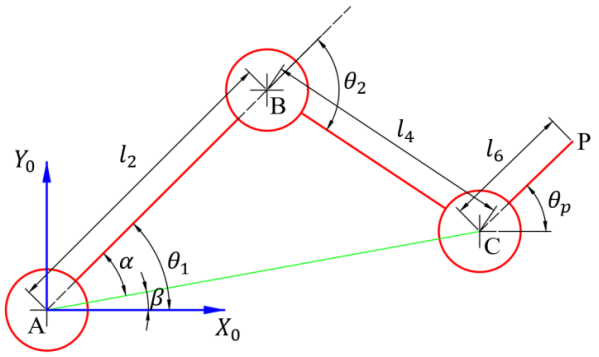


Figure 2. The kinematic diagram of the SCARA robot

The coordinate of point C is:

$$\begin{aligned} x_c &= x_p - l_6\cos\theta_p \\ y_c &= x_p - l_6\sin\theta_p \end{aligned} \quad (8)$$

Consider the triangular ABC; the angle θ_2 can be calculated:

$$\theta_2 = -a\cos\frac{x_c^2 + y_c^2 - l_2^2 - l_4^2}{2l_2l_4} \text{ q} \quad (9)$$

The angle θ_1 is the sum of two angles α and β:

$$\theta_1 = \alpha + \beta \quad (10)$$

where:

$$\cos\alpha = \frac{l_2^2 + x_c^2 + y_c^2 - l_4^2}{2l_2\sqrt{x_c^2 + y_c^2}} \quad (11)$$

$$\tan\beta = \frac{y_c}{x_c} \quad (12)$$

The inverse kinematic problem has two solutions. Consider another configuration of the robot; the values of the angles θ_1 and θ_2 are:

$$\theta_2 = a\cos\frac{x_c^2 + y_c^2 - l_2^2 - l_4^2}{2l_2l_4} \quad (13)$$

$$\theta_1 = -\alpha + \beta \quad (14)$$

3. PLC DESIGN AND IMPLEMENTATION

3.1 Electricaldesign with PLC

In this project, a Programmable Logic Controller (PLC) is employed as the robot controller to control the motion of the robot joints. This section presents the electrical devices used in the system and describes the functions, technical specifications, and connection diagram between devices. The devices and components of the electrical circuit are shown in Figure 2.

The PLC lies in the center of the diagram, demonstrating the importance of this device in the electrical circuit. PLC has the function of controlling the operation of other devices in the system. It reads the input module's signal states, executes the user program's logic operations, and controls the outputs. PLCs range from small non-modular devices used for processing simple binary signals up to large-rack-mounted modular devices, which can perform analog processing, PID control, motor control, networking, and more. Our design employs the small non-modular PLC DVP28SV11T2 to develop a cost-efficient system. This PLC belongs to the DVP-SV2 series of Delta Company. DVP-SV2 PLC is the high-end model of the Delta DVP-S series with larger program capacities and data registers. DVP-SV2 has excellent motion control. It consists of 4 high-speed pulse outputs up to 200kHz.

Many motion control instructions are integrated into the programming software to meet the applications that require high-speed and high-precision positioning control. Therefore, DVP-SV2 PLC is suitable for robot motion control applications with 4-DOF. The specifications of DVP28SV11T2 PLC are listed in Table 2.

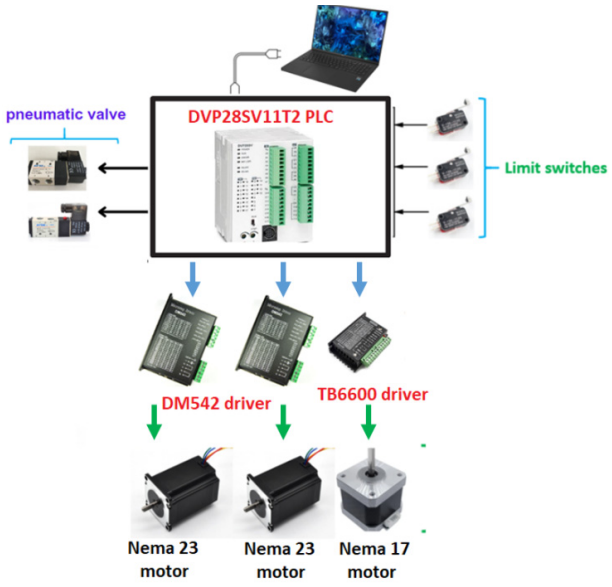


Figure 3. Components in electrical system

Specifically, the PLC functions in this project consist of 1) output pulse and direction signals for controlling the three stepper motors to rotate at the required rotation angle, 2) output digital signals to change the state of the pneumatic valves, 3) reading the signal of the limit switches to halt the motors, 4) execute the program to solve the kinematic equations of the robot, 5) communication with another device via RS232/RS485 port, 6) executes the logic operations of the user program for a specific application.

The pulse/direction signals are sent to motor drivers to convert to voltage signals that apply to the motor coils to create the stepper motors' precision motion. Two NEMA 23 motors are installed in the two first joints to rotate the robot links, and the NEMA 17 is used for the last joint. The specifications of motors are illustrated in Table 3 and Table 4, respectively.

Table 2. Specifications of DVP28SV11T2 PLC

Specifications	Value
Power supply	24V
Program capacity	30k step
Communications	RS232/RS485
Total number of I/O pins	28 pins
Number of input pins	16 pins
Number of output pins	12 pins
Type of output pins	Transistor

Table 3. Specifications of NEMA 23 motor

Specifications	Value
Step angle	1.8^0
Motor length	76mm
Rate current	3.0A
Holding torque	1.8Nm
Weight	1050g
Rotor Inertia	440 $g.cm^2$
Lead Wires	4
Shaft Diameter	8mm

Table 4. Specifications of NEMA 17 motor

Specifications	Value
Step angle	1.8^0
Motor length	48mm
Rate current	1.5A
Holding torque	0.55Nm
Weight	0.4kg
Lead Wires	4
Shaft Diameter	5mm

Table 5. Pin function of the DM542 driver

Pin	Details
PUL+	Pulse signal: receive the pulses from the controller with the voltage of 5-24V when PUL-HIGH and 0-0.5V when PUL-LOW
PUL-	
DIR+	Direction signal: representing two directions of motor rotation; 5-24V when DIR-HIG, 0-0.5V when DIR-LOW
DIR-	
ENA+	Enable signal: This signal is used for enabling/disabling the driver
ENA-	
+V	Power supply, 20-50VDC
GND	Power ground
A+, A-	Motor phase A
B+, B-	Motor phase B

The NEMA 23 motor is controlled through the Leadshine DM542 digital stepper motor drive. The DM542 is a fully digital drive developed with an advanced DSP control algorithm based on the latest motion control technology. The main features of DM542 drive are auto-identification motor and auto-configuration parameter; include 15 selectable micro-step resolutions: 400, 800, 1600, 3200, 6400, 12800, 25600, 1000, 2000, 4000, 5000, 8000, 10000, 20000, 250000; input voltage from 18-50VDC; 8 selectable

peak current: 1A, 1.46A, 1.91A, 2.37A, 2.84A, 3.31A, 3.76A, 4.2A; pulse input frequency up to 200kHz with optical isolate; over voltage, over current protections. Table 5 details the pins' functions of the DM542 drive.

The NEMA 17 motor is controlled through a TB6600 digital stepper motor drive. This drive has 6 selectable micro-step resolutions: 200, 400, 800, 1600, 3200, 6400; 8 selectable peak currents: 0.7A, 1.2A, 1.7A, 2.2A, 2.7A, 2.9A, 3.2A, 4.0A. The supply voltage is from 9V to 42VDC. The pins' function of TB6600 is the same as the DM542 drive.

Three limit switches are arranged at the end of the stroke of the rotating joints to limit the rotating joints' rotation angle and determine the joints' initial position when starting. When powered on, the robot joints rotate to touch limit switches. Then, the motors stop reverse rotation. The joints rotate with a certain rotation angle to return to zero origin. In this position, the robot arm is straightened.

Figure 4 shows the wiring diagram of the electrical circuit. It must be noted that the PLC's DC input has two operating modes: SINK and SOURCE. It depends on the current going into or out of the S/S pin of the PLC. The diagram in Figure 4 uses the SINK mode. The S/S pin is connected to the 24V source, so current flows from the 24V source to the S/S pin. The input pins X0, X1, and X2 are connected with the limit switches of robot joints. The input pins X3 and X4 are connected with the start and stop buttons. The high-speed pulse output pins Y0, Y2, and Y4 of the PLC are respectively connected to the PUL pin of the stepper motor drives.The output pins Y1, Y3, and Y5 are used as direction pins, which are connected to the DIR- pins of the stepper motor drives. Switching power is used to convert the 220VAC to the 24VDC power. 24VDC power is distributed to other devices such as PLCs, stepper motor drives, and coils of pneumatic valves. A circuit breaker (CB) is installed at the input 220VAC to protect all devices in the circuit. The output of the CB is connected in series with an EMI/RFI filter device. The purpose is to stop noise from entering and disrupting the operation of electrical equipment and stop electrical equipment in the circuit from putting EMI/RFI noise onto the power lines.

3.2 PLC programming

To write a program for the Delta DVP-PLC series, the software WPLSoft is used. In addition to basic commands, WPLSoft provides application commands to generate high-speed pulses to the output pins and conduct the arithmetic operations applied in our application. The API 57 PLSY is the command for pulse output with the structure as follows:

PLSY	S1	S2	D

- S1: Pulse output frequency
- S2: Pulse output number
- D: External output (Y0, Y2, Y4, Y6)

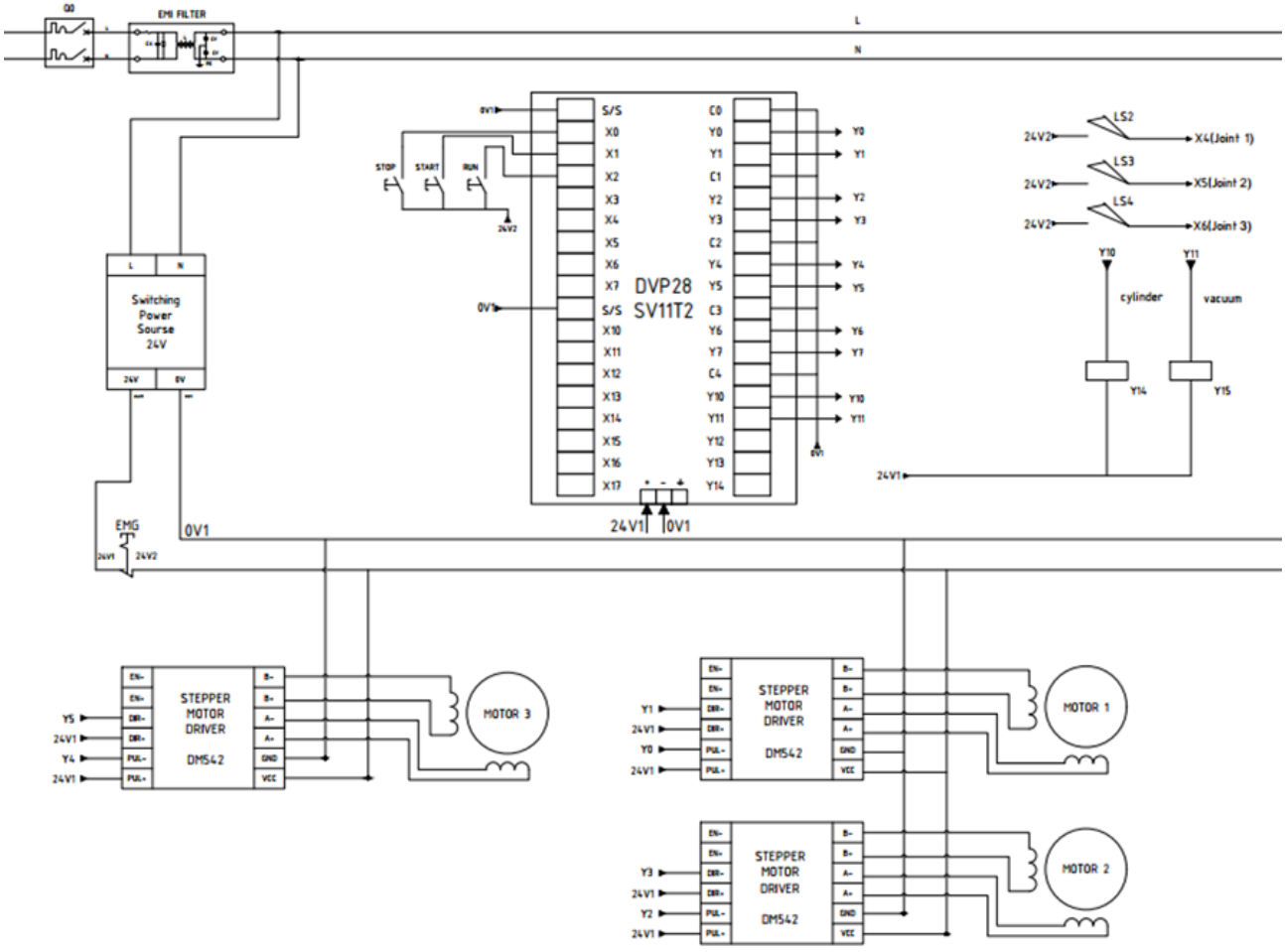


Figure 4. The wiring diagram of the electrical system

PLSY instruction assigns the number of output pulses S2 from output device D at the frequency S1. Before using this command, the direction signal pin of the motor drive must be assigned so that the motor rotates in the correct direction. The PLSR command performs the same function as PLSY; however, it allows the generation of acceleration pulses, which helps the motor accelerate and decelerate when starting and stopping, minimizing vibrations. The structure of the PLSR command is:

PLSR	S1	S2	S3	D

- S1: Pulse output frequency
- S2: Pulse output number
- S3: acceleration/deceleration time (ms)
- D: External output (Y0, Y2, Y4, Y6)

Table 6 lists some arithmetic operations that are used to solve the forward kinematic and inverse kine–matics equations.

The robot control task is divided into many prob–lems, from simple to complex. The first requirement to perform is to return the robot arms to zero.

Table 6. The arithmetic operations

Command	Functions
DADDR	Floating point number addition
DSUBR	Floating point number subtraction
DMULR	Floating point number multiplication
DDIVR	Floating point number division
DCOS	Cosine operation of binary floating point
DSIN	Sine operation of binary floating point
DTAN	Tangent operation of binary floating point
DACOS	Arccosine operation of binary floating point
DATAN	Arctangent operation of binary floating point
DESQR	Square root of binary floating point

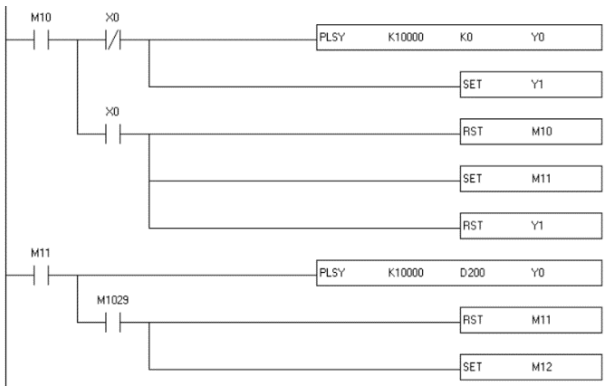


Figure 5. Program to rotate the first joint to the home position.

Figure 5 shows an example of returning a motor to zero.The pulse signal is the Y0 pin, Y1 is the direction signal, and X0 is the limit switch. Register D200 is used to contain the number of pulses to rotate the motor when it touches the limit switch to the zero position. The flag M1029 will be On when the CH0 pulse output is com–pleted.

To conduct the kinematic calculations, the kinematic parameters of the robot are stored in the data registers for convenience in calculations and command usage. The data registers are shown in Table 7.

Table 7. The register to store the kinematic parameters

Variables	Registers
l_2	D212
l_4	D214
l_6	D216
x_p	D218
y_p	D220
θ_p	D222
θ_1	D224
θ_2	D226
θ_3	D228
x_c	D14
y_c	D16
$x_c^2 + y_c^2$	D22
$x_c^2 + y_c^2 - l_4^2$	D24
π	D20
l_2^2	D26
α	D30
β	D32

Figure 6 is the ladder diagram to calculate the angle θ_2 according to Equation (9). This program segment will be executed when contact M0 is activated. Each command in the PLC ladder program performs the calculation of one or two terms. Therefore, the equation is broken down into multiple statements, each statement performing a calculation. The previous calculation results are stored and used for subsequent calculations until the equation is completed. For example, to calculate the value x_c, first, the cosθp is calculated by the command DCOS and stored in register D10. Then, multiply two registers, D216 (store the value of l_6) and D10, by command DMULR corresponding to the expression $l_6\cos\theta_p$; the result is stored in register D10. Finally, subtract register D218 from register D10 by the command DSUBR and store the result in register D14, which is the value x_c. The first six lines of the program calculate the coordinate ((x_c,y_c)) according to Equation (10). These values are contained in registers D14 and D16 for calculating θ_1 and θ_2 angles. The program to calculate the angles θ_1 and θ_3 are illustrated in Figure 7. Some values that are used to calculate the angle θ_2 are re-used to calculate the angle θ_1, such as $x_c^2 + y_c^2$ stored in D22 and $x_c^2 + y_c^2 - l_4^2$ stored in D24. Add two re–gisters, D26 and D24, by the DADDR command to cal–culate the value of the expression $l_2^2 + x_c^2 + y_c^2 - l_4^2$; the result is stored in D18. Using the DSQR command to

calculate $\sqrt{x_c^2 + y_c^2}$, the input is D22 (store the value of $x_c^2 + y_c^2$), and the output is stored in the D28 register. Then, DDIVR instructions are used consecutively to calculate the value of cosα according to Equation (11); the value of cosα is stored in D18. Next, the DACOS command is employed to calculate the value of angle α; the result is stored in the D30 register. Divide two registers, D16 and D14, to calculate the value of tanβ and use the DATAN command to get the value of the angle β; this value is stored in the D32 register. Finally, two registers, D30 and D32, are added together by the DADDR command to obtain the angle θ_1 (stored in the D224 register).

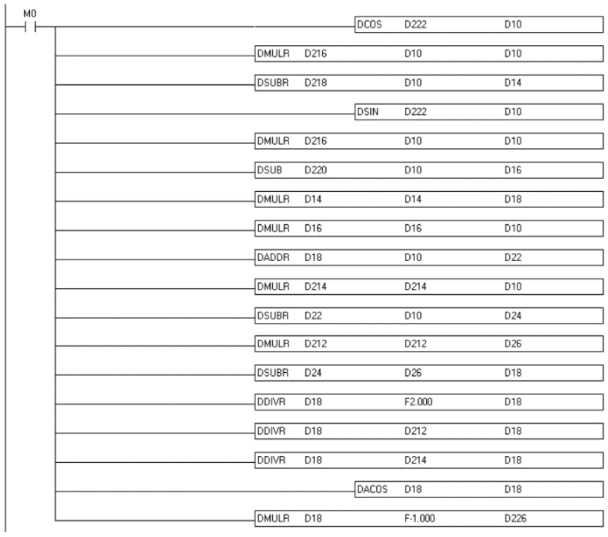


Figure 6. Program to calculate the angle θ_2

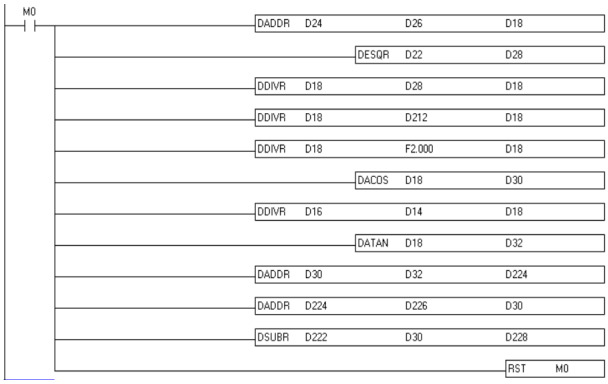


Figure 7. Program to calculate the angles θ_1 and θ_3

The main application developed in this study is Pick-and-Place tasks. To perform this task, a program segment is written to rotate three joints from the current position to the desired position (Move Joint Instruction). The contact M2 is used to activate this sub-program. The desired absolute angles of three joints are stored in registers D230, D232, and D234, respectively. Then, the relative angles are calculated by subtracting the desired angles from the current angles. The relative angles are used to calculate the number of pulses for the stepper motors. The INT command is used to convert the floating point numbers to the integer values. The DDRVI command has the function of creating pulses to the output pins. The structure of the DDRVI is as follows:

DDRVI	S1	S2	D1	D2

- S1: Number of output pulses (relative designation)
- S2: Pulse output frequency
- D1: Pulse output device (Y0, Y2, Y4, Y6)
- D2: Output device for the signal of rotation direction

Figure 8 shows the program segment for the Move Joint Instruction. The data registers D242, D244, and D242 store the pulse frequency of the first, second, and third joints, respectively. The flags M1029, M1030, and M1036 are On when pulse outputs are completed. So, the flag M2 is set to OFFF when all three flags are ON, ending the process. At the same time, M6 is also set ON to indicate that the program has finished executing. This program segment is used as a sub-function. When we want to move the robot from one point to another, we just need to put the rotation angle values into registers D242, D244, and D242 and turn on the M2 motor to execute the program. When the program finishes, the M6 flag is turned ON; we use this mechanism to continue executing the next commands.

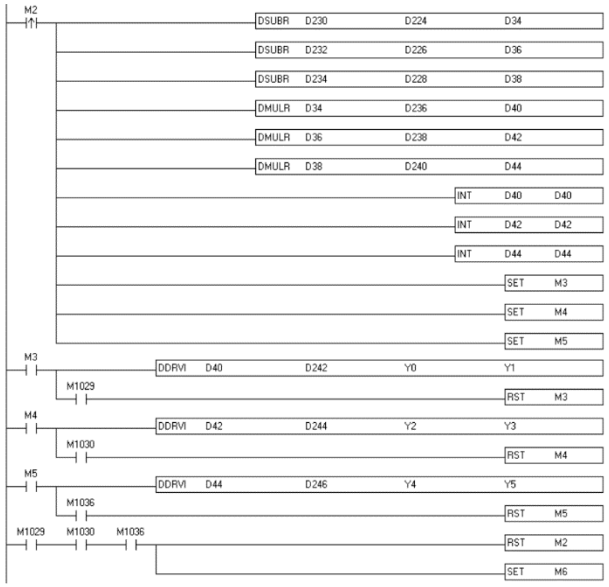
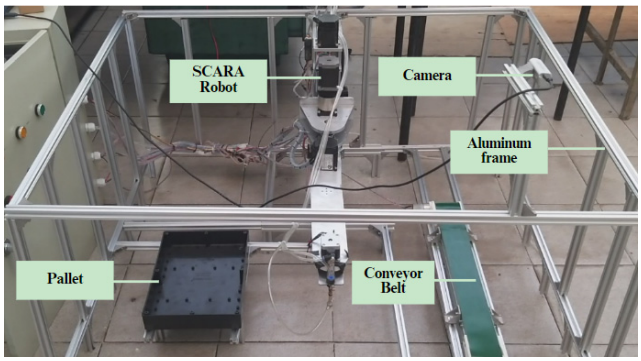


Figure 8. Program to for the Move Joint Instruction

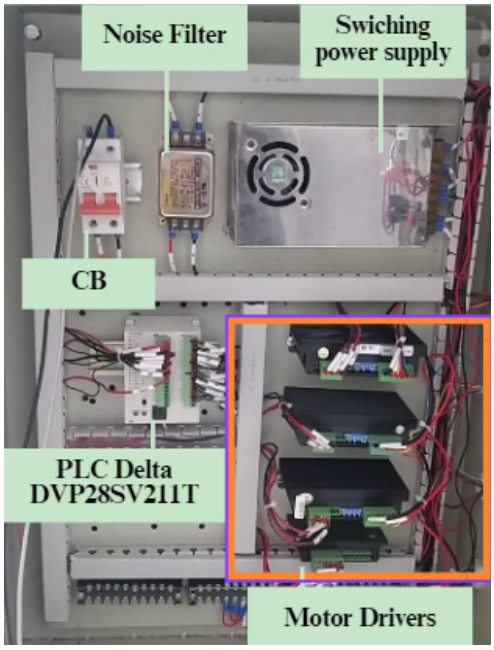
4. EXPERIMENT RESULTS AND DISCUSSION OF THE ROBOT ARM CONTROLLED BY PLC

Figure 9 shows the experimental robot arm and elec–trical cabinet. The links of the robot arm are machined on a CNC machine with aluminum alloy. Mechanical parts are precisely machined on the CNC machine to ensure that the assembly meets technical requirements. The lengths of the robot's links are l_1 = 260 mm, l_2 = 240 mm and l_3 = 60 mm. The robot control cabinet is constructed based on the electrical diagram in Figure 4. The size of the cabinet is 800x600 mm. The NEMA 23 motors are set up with a resolution of 10000 pulses/ revolution, and the NEMA 17 motor is set up with a resolution of 6400 pulses/revolution. Because stepper motors are used, joint speeds are limited to values of 360 degree/s.

Before uploading code to the PLC, we need to test the correctness of the programs. One of the most error-prone programs is the inverse kinematics problem because it involves complex arithmetic calculations that a PLC cannot easily handle. Using the simulator function of the WPLsoft software, we run the simulation to calculate the angles θ_1, θ_2, and θ_3 by loading different values of the end-effector position and orientation to the registers D218, D220, and D222, respectively. After running the program, open the "monitor devices" window on the software and add the data registers to the Device Name column to see the floating point values (see Figure 10).Because the inverse kinematics problem is solved using geometric methods, AutoCAD software is used to draw the robot's configuration and measure the values of joint angles. Comparing the rotation angle values obtained by simulation and the measured values on AutoCAD software, the results show that the program written on WPLsoft gives the correct solution. Some simulation cases are shown in Figure 11. Similarly, the simulator function on WPLsoft also tests other program segments.



(a) SCARA robot arm



(b) Electrical cabinet

Figure 9. Experimental system

Figure 11 shows the flow chart to control the robot arm for a Pick and Place application. When turning on the CB on the electrical cabinet, the operation sequence of the robot is as follows: First, the robot moves to the home position. At this position, the values of the robot's joints are zeros. Second, the robot moves to the object's position to grasp the object by using the Move Joint Instruction described in this paper. When the end-effector has approached the object, the PLC outputs a signal to an output pin to close the gripper; the object is attached to the gripper. Then, the robot brings the object to another position and de-attaches the object to place it in a certain position. The above process will repeat until the last object is picked up. Finally, the robot returns to its home position to wait for another task.

Device Name	Comment	Statu	T/C	Pres	Prese	Floating Point	Format	T/C Set V
D218	Xp value			K0	K113	F460.000	Signed Decimal	
D220	Yp value			K0	K-10	F-100.000	Signed Decimal	
D222	thetaP value			K0	K0	F0.000	Signed Decimal	
D224	theta 1 angle			K-23	K105	F0.329	Signed Decimal	
D226	theta 2 angle			K42	K-10	F-1.203	Signed Decimal	
D228	theta 3 angle			K-12	K106	F0.874	Signed Decimal	

Device Name	Comment	Statu	T/C	Pres	Prese	Floating Point	Format	T/C Set V
D218	Xp value			K0	K113	F300.000	Signed Decimal	
D220	Yp value			K0	K112	F115.000	Signed Decimal	
D222	thetaP value			K-73	K105	F0.523	Signed Decimal	
D224	theta 1 angle			K-22	K106	F0.983	Signed Decimal	
D226	theta 2 angle			K-12	K-10	F-2.106	Signed Decimal	
D228	theta 3 angle			K-17	K107	F1.646	Signed Decimal	

Device Name	Comment	Statu	T/C	Pres	Prese	Floating Point	Format	T/C Set V
D218	Xp value			K-32	K113	F273.000	Signed Decimal	
D220	Yp value			K0	K-10	F-204.000	Signed Decimal	
D222	thetaP value			K-10	K106	F0.890	Signed Decimal	
D224	theta 1 angle			K-41	K106	F1.007	Signed Decimal	
D226	theta 2 angle			K-25	K-10	F-2.165	Signed Decimal	
D228	theta 3 angle			K31	K107	F2.047	Signed Decimal	

Figure 10. Some simulation results

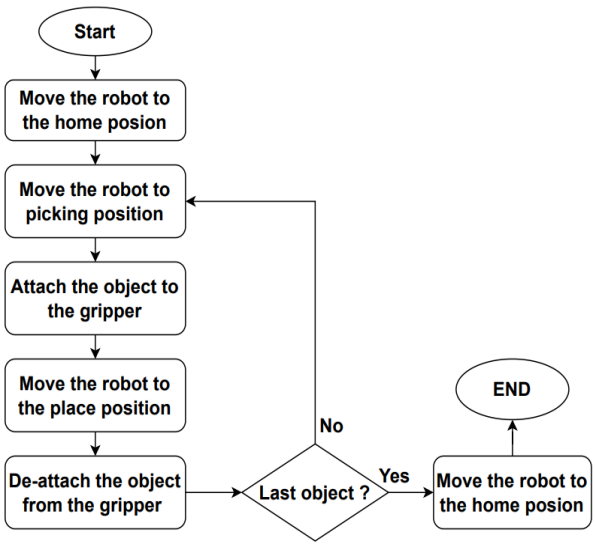
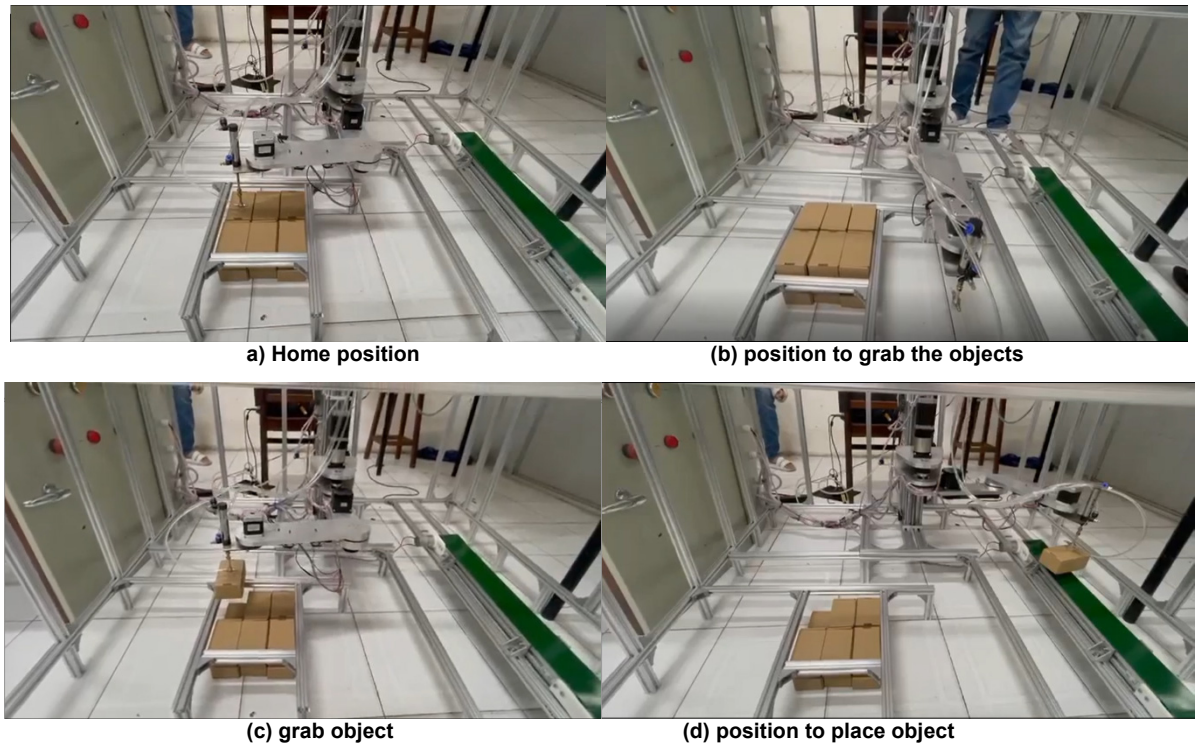


Figure 11. Flow chart for Pick and Place applications

Figure 12 shows the states of the robot in the operation process: (a) shows the configuration of the robot at the home position, (b) shows the robot at the position to grab the objects, (c) the object is grabbed by the gripper, (d) shows the robot at the position to place the object. Experimental tests have shown that the robot operates correctly, and the programs do not have calculation and logic errors. The stepper motor can reach the speed of 360 degrees/s without losing steps.



a) Home position (b) position to grab the objects

(c) grab object (d) position to place object

Figure 12. Test the operation of the robot in a Pick and Place task

As a result, the robot takes about 1.2 to 2.5 seconds to pick and place a product. Therefore, the robot can meet the requirements of pick-and-place applications in production lines. Using PLC to design the controller helps the robot operate stably and durably; the signal from the PLC is less affected by noise from the surrounding environment.

5. CONCLUSIONS

This paper has designed a controller based on PLC for a 3-DOF SCARA robot arm. The devices and components of the electrical circuit are analyzed and connected together to control the robot arm. PLC plays a key role in the electrical system. It controls the operation of other devices, such as motors, valves, and sensors. The kine–matics equations of the robot arm have been analyzed. A PLC subprogram of the kinematics equations is written to calculate the rotation angles of the robot joints.The si–mulator function of WPLsoft validates the kinematic solution.

Using commands to perform the high-speed pulse generation task, a sub-program is programmed to perform the point-to-point control task. A Pick and Place application is implemented to verify the accuracy of the program. The results show that the robot can grasp objects without errors. The stepper motors can reach the speed of 360 degrees/s without losing steps. As a result, the robot takes about 1.2 to 2.5 seconds to pick and place a product. Using PLC to design the controller helps the robot operate stably and durably. Therefore, the designed controller can be applied in industrial environments.

ACKNOWLEDGMENT

We acknowledge Ho Chi Minh City University of Technology (HCMUT) and VNU-HCM for supporting this study.

REFERENCES

[1] Cong, V.D., Hanh, L.D.: Visual servoing control schemes of 4 DOF robot manipulators. Int J Intell Robot Appl, Vol. 6, pp. 804–813, 2022. https://doi.org/10.1007/s41315-022-00259-7

[2] Thurow, K.: System Concepts for Robots in Life Science Applications, Applied Sciences, Vol. 12, No. 7, 2022. https://doi.org/10.3390/app12073257

[3] Cong, V.D.: Extraction and classification of moving objects in robot applications using GMM-based background subtraction and SVMs, J Braz. Soc. Mech. Sci. Eng., Vol. 45, 317, 2023. https://doi.org/10.1007/s40430-023-04234-6

[4] Xu, Z., Xia, J., Zhong, F.: Adaptive chicken swarm optimization algorithm for identifying structural parameters of 6-DOF mechanical arm. J Braz. Soc. Mech. Sci. Eng.,Vol. 46, No. 17, 2024. https:// doi.org/10.1007/s40430-023-04585-0

[5] Kim, H.S., Kim, G.S.: 5□Axis Robot Design for Loading and Unloading Workpieces, International Journal of Precision Engineering and Manu–facturing, 2023.

[6] Kouritem, S. A. et al.: A multi-objective optimi–zation design of industrial robot arms. Alexandria Engineering Journal, Vol. 61, No. 12, 12847-12867, 2022. https://doi.org/10.1016/j.aej.2022.06.052

[7] Li, K., Huo, Y., Liu, Y., Shi, Y., He, Z., and Cui, Y.: Design of a lightweight robotic arm for kiwifruit pollination. Computers and Electronics in Agriculture, Vol. 198, 2022. https://doi.org/10. 1016/j.compag.2022.107114

[8] Anurag, S., Rashmi, A., Yashpal, S.: Design and Static Analysis of Robotic Arm using Ansys, International Journal of Recent Technology and Engineering, Vol. 9, No. 1, 2020.

[9] Phuong, L.H., Cong, V.D., Hiep, T.T.: Design a Low-cost Delta Robot Arm for Pick and Place Applications Based on Computer Vision, FME Transactions, Vol. 51, pp. 99-108, 2023.

[10] Cong, V.D., Duy, A.D., Phuong, L.H.: Development of Multi-Robotic Arm System for Sorting System Using Computer Vision, Journal of Robotics and Control (JRC), Vol. 3, No. 5, 2022.

[11] Singh, G., Banga, V.K.: Kinematics and trajectory planning analysis based on hybrid optimization algorithms for an industrial robotic manipulators. Soft Comput,Vol. 26, pp. 11339–11372, 2022) https://doi.org/10.1007/s00500-022-07423-y

[12] Rout, A., Mahanta, G.B., Bbvl, D. et al.: Kinematic and Dynamic Optimal Trajectory Planning of Industrial Robot Using Improved Multi-objective Ant Lion Optimizer. J. Inst. Eng. India Ser. C 101, pp. 559–569, 2020. https://doi.org/10.1007/s40032-020-00557-8

[13] Raza, K., Khan, T. A., and Abbas, N.: Kinematic analysis and geometrical improvement of an industrial robotic arm. Journal of King Saud University - Engineering Sciences, Vol. 30, No. 3, pp. 218-223, 2018. https://doi.org/10.1016/j.jksues.2018.03.005

[14] Yang, H., Shen, M., Zhang, M., et al.: Structural design and performance analysis of a self-driven articulated arm coordinate measuring machine. Meas Sci Technol, Vol. 33, No. 1, 2022.

[15] Ali, Z., Sheikh, M. F., Al Rashid, A., Arif, Z. U., Khalid, M. Y., Umer, R., and Koç, M.: Design and development of a low-cost 5-DOF robotic arm for lightweight material handling and sorting applications: A case study for small manufacturing industries of Pakistan, Results in Engineering, Vol. 19, 2013.

[16] Li, X., Sun, K., Guo, C., Liu, T., Liu, H.: Design, modeling and characterization of a joint for inflatable robotic arms, Mechatronics, Vol. 65, 2020.

[17] Tung, T. T., Van Tinh, N., Phuong Thao, D. T., and Minh, T. V.: Development of a prototype 6 degree of freedom robot arm, Results in Engineering, 18, 2023.

[18] Huynh, H. N., Assadi, H., Rivière-Lorphèvre, E., Verlinden, O., and Ahmadi, K.: Modelling the dynamics of industrial robots for milling operations, Robotics and Computer-Integrated Manufacturing, Vol. 61, 2020.

[19] Zhang, B., Wu, J., Wang, L., & Yu, Z.: Accurate dynamic modeling and control parameters design of an industrial hybrid spray-painting robot, Robotics and Computer-Integrated Manufacturing, Vol. 63, 2020.

[20] Nguyen, H.V., Cong, V.D., Trung, P.X.: Development of a SCARA Robot Arm for Palletizing Applications Based on Computer Vision, FME Transactions, Vol. 51, No. 4, pp. 541-549, 2023.

[21] Slavković, N., Živanović, S., Vorkapić, N., Dimić, Z.: Development of the Programming and Simu–lation System of 4-axis Robot with Hybrid Kine–matic, FME Transactions, Volume 50 No.3, 2022.

[22] Phuong, L.H., Cong, V.D.: Control the Robot Arm through VisionBased Human Hand Tracking, FME Transactions, Vol. 52, No. 1, pp. 37-44, 2024.

[23] Xu Z, Wang W, Chi Y, Li K, He L.: Optimal Trajectory Planning for Manipulators with Efficiency and Smoothness Constraint. Electronics. Vol. 12, No. 13, 2023.

[24] Wang, X., Zhou, X., Xia, Z., Gu, X.: A survey of welding robot intelligent path optimization, J. Manuf. Process., Vol. 63, pp. 14–23, 2021.

[25] Xue, Z., Zhang, X., Liu, J.: Trajectory planning of a dual−arm space robot for target capturing with minimizing base disturbance, Adv. Space Res, 2023.

[26] Tamizi, M.G., Yaghoubi, M., Najjaran, H.: A review of recent trend in motion planning of industrial robots, Int. J. Intell. Robot. Appl. Vol. 7, pp. 253–274, 2023.

[27] Wang, T., Xue, Z.,Dong, X., Xie, S.: Autonomous Intelligent Planning Method for Welding Path of Complex Ship Components, Robotica, Vol. 39, pp. 428–437, 2020.

[28] Vyas, D. R., Markana, A., and Padhiyar, N.: Economic 6-DOF robotic manipulator hardware design for research and education, Materials Today: Proceedings, Vol. 62, pp. 7179-7184, 2021. https://doi.org/10.1016/j.matpr.2022.03.140

[29] Hussein M.A., Yasir H., Ghadah A.A.: Design and implementation of Arduino based robotic arm, International Journal of Electrical and Computer Engineering (IJECE), Vol. 12, No. 2, pp. 1411-1418, 2022.

[30] Victor, H.B.B., Rodrigo, S., David E.E.C.: Design of an affordable IoT open-source robot arm for online teaching of robotics courses during the pandemic contingency, HardwareX, Vol. 8, 2020.

[31] Laski, P.A., Smykowski, M.: Using a Development Platform with an STM32 Processor to Prototype an Inexpensive 4-DoF Delta Parallel Robot, Sensors. Vol. 21, No. 23, 2021.

[32] Li, N., Chen, Y., Shi, Y., Gao, F., Che, J. and Chen, J: Development of a DSP based control system for a parallel high-quality tea plucking robot, in American Society of Agricultural and Biological Engineers Annual International Meeting 2015, pp. 1188–1195, Curran Associates, Inc., 2015.

[33] Lee, J.W. and Jung, S.: A tutorial on control implementation for a collaborative robot in joint space using DSPs, Journal of Institute of Control, Robotics and Systems, Vol. 28, No. 1, pp. 28–38, 2022.

[34] Cong, V.D.: Industrial Robot Arm Controller Based on Programmable System-On-Chip Device, FME Transactions, Vol. 49, No. 4, pp. 1025-1034, 2021.

[35] Deli, Z., Shaohua, J., Liu, Z.: The Parallel Solving Method of Robot Kinematic Equations Based on FPGA, Journal of Robotics, pp. 1-8, 2023.

[36] Fan, X., Yan, H. and He, D.: Inverse kinematics of robot based on Zynq platform, Microcontroller and Embedded System Applications, Vol. 17, No. 2, pp. 18–22, 2017.

[37] Chen, W.C., Chen, C.S., Lee, F.C. and Kung, Y.S.: Digital hardware implementation of the forward/inverse kinematics for a SCARA robot manipulator, 2018 IEEE International Conference on Applied System Invention (ICASI), Chiba, Japan, pp. 54-57, 2018.

[38] Chand, R., Chand, R.P., Assaf, M., Naicker, P.R., Narayan, S.V. and Hussain, A.F.: Embedded FPGA-based motion planning and control of a dual-arm car-like robot, in 2022 IEEE 7th Southern Power Electronics Conference (SPEC), pp. 1–6, IEEE, Nadi, Fiji, 2022.

[39] Gürsoy, H. and Efe, M.Ö.: Control system implementation on an FPGA platform, IFAC-PapersOnline, Vol. 49, No. 25, pp. 425–430, 2016.

[40] Wu, K., Krewet, C., and Kuhlenkötter, B.: Dynamic performance of industrial robot in corner path with CNC controller. Robotics and Computer-Integrated Manufacturing, Vol. 54, pp. 156-161, 2018. https://doi.org/10.1016/j.rcim.2017.11.008

[41] Köse, I., Öztürk, S., and Kuncan, M.: Pantography application with real-time plc based on image processing in gantry robot system, European journal of technique (EJT), Vol. 9, No. 2, pp. 219-229, 2019. https://doi.org/10.36222/ejt.621558

[42] Öztürk, S., and Kuncan, F.: Linear Delta Robot Controlled with PLC Based On Image Processing. Kocaeli Journal of Science and Engineering, Vol. 5, No. 2, pp. 150-158, 2022. https://doi.org/10.34088/kojose.1058559

ДИЗАЈН И РАЗВОЈ ТРОШКОВНО-ЕФИКАСНЕ РОБОТСКЕ РУКЕ СА РОБОТСКИМ КОНТРОЛЕРОМ БАЗИРАНИМ НА ПЛЦ-У

В.Д. Конг

Да би се развила исплатива роботска рука за типичну примену пицк анд плаце која се може применити у индустрији, овај рад је применио програмабилни логички контролер (ПЛЦ) за контролу ротационог кретања зглобова робота. Главни задаци ПЛЦ контролера су израчунавање кинематике, креирање импулсних излаза велике брзине за корачне моторе и имплементација операција секвенце за одређену примену. Функције се уписују у сегменте потпрограма. Када је потребно, главни програм укључује само одговарајућу заставицу за извршавање потпрограма. Коришћењем унапред написаних потпрограма, логички низ за имплементацију апликације Пицк анд Плаце се лако имплементира и описује у овом раду. ПЛЦ програм је развијен за контролу СЦАРА робота са три ротационе спојнице. Корачни мотори покрећу зглобове робота. Делта ДВПСВ2 ПЛЦ се користи за дизајнирање роботског контролера. Ова серија ПЛЦ-а има четири брза импулсна излазна пина, што је погодно за овај пројекат. Синхроно кретање корачних мотора се лако изводи коришћењем команди за брзи импулсни излаз уграђених у ПЛЦ програм. Експериментални резултати управљања роботском руком су показали ефикасност и тачност развијеног програма. Проблеми кинематике напред и инверзне кинематике роботске руке су верификовани коришћењем симулатора у софтверу. Зглобови робота се померају синхроно како је потребно за обављање апликација за бирање и постављање.