Proc. of the 5th WSEAS/IASME Int. Conf. on Electric Power Systems, High Voltages, Electric Machines, Tenerife, Spain, December 16-18, 2005 (pp645-655)

Modeling and Implementation of Robot Based Control by using Programmable Logic Controller

Ashraf Salah El Din ZEIN EL DIN
Electrical Engineering Department, Faculty of Engineering, Minoufiya University, Shebin El-Kom, Egypt
Tel. 00-20-48-2312420, Fax. 00-20-48-2235695

Abstract
A novel scheme of robot based control using Programmable Logic Controller (PLC) is proposed, in which the PLC performs control of robot arms via a novel interface. This scheme is simple and minimum drive control unit for robot. A mathematical model and software algorithm are proposed for controlling signals, switching and monitoring the system behavior. The results obtained by laboratory implementation are presented to verify the effectiveness of the proposed on-line scheme.

1. Introduction

The robots are used in various fields of application including the direct contact with human being in future, the systematic study of such robots which have multi-degree of freedom and deform its form softly is considered to be very important [1]. The theoretical design of a N axes robot arm dynamic controller [2], called computed voltage controller, which includes the current dynamics of a synchronous driver (brushless DC motor). The method is implemented on an experimental one axe robot joint. Generally, the advanced control laws of robot manipulators, such as computed torque control, consider the joint drive chain as a constant gain, that is to say the controllers are designed at the 'torque input' level [3][4]. But in the case of high speed motion/force control of robot manipulators the dynamics associated with the actuators can't be neglected [5]. Including the robot actuator dynamics into the robot equations typically makes the latter a system of third order non linear differential Equations. A few authors have discussed this problem, but with complicated control law such as Freud 's nonlinear control theory or with the Riccati equation applied to a robot driven with a DC motor. Based on the dynamic control technique [6][7]. Compared with the computed torque control law which needs the desired joint position, velocity and acceleration, the global computed voltage control law requires the desired joint position, velocity, acceleration and jerk to track. [8]. The present technological trends in microelectronic, sensor and computing techniques make possible intensive research and development in the large area of applications. One of the most progressive areas is the development of autonomous mobile robots. Recently, there is an effort to convert the academic development of the individual prototypes into the serially produced mobile systems, which are widely used in practice. These systems are more and more used in different areas of infrastructure (cleaning mobile systems, mobile systems in hospitals, etc.), in the civil engineering and the machine-building [9]. Robots are essential components of automation to achieve such objectives. For this reason, robots are increasingly deployed in several production processes to increase efficiency. 3D robot placement controlled by programmable logic controller (PLC) is an interesting automation device for the thesis proposal because 3D robots, operating along axes x, y and z, are widely used in grabbing and placement maneuvers such as holding steel sheet for car body molding and removing objects from plastic injection molds. Furthermore, 3D robot's working principle is similar to CNC machines and 3D measuring instruments. By undertaking this thesis project, it is possible to more clearly understand the working of such machines and instruments as well. Axes x and y employ ball screw with step motor driving grabbing and holding unit of the robot arm to desired position. Encoder unit determines exact location controlled by feedback or close-loop signals. Axis z, moving vertically, employs pneumatic system for the action of grabbing and holding. The robot is designed systematically that various parts function in harmony. It can be controlled automatically or manually. Robot arm can be moved to various positions through numeric switching. It can also be taught to move in a particular way. The error margin ranges from 0.22-0.042 mm which is within 0.5 mm. The robot has fulfilled all prescribed thesis objectives [10].

2. Modeling and Simulation of a PLC based control of Robot

The robot is designed systematically that various parts function in harmony. It can be controlled automatically or manually. Robot arm can be moved to various positions through numeric switching.

The schematic diagram of a proposed system shown in Figure 1, consists of a computer which may be use in conjunction with a Programmable Logic Controller (PLC) for writing and storing a program for controlling the arms of the robot. The robot catches an object from a specified position and transports it to another location. The interface is a circuit required for connection between the PLC and Robot and it may be use for controlling the direction of movement of robot arms. The D-C power supply is used for distributing a power to the d-c motors which are connected to joints of links terminals in both sides. **Robot control** is the theory of how to model and control robots. A simplistic model of a robot is to look at as a collection of *links* connected by *joints*. The tip of the robot is commonly referred to as the *tool center point* or TCP. As the joints rotate and the links contract and expand, the TCP will change position. It is of great importance to know the position of the TCP in world coordinates. For example, for a robot to weld in a straight line, the actuators in the joints of the robot have to be controlled in complex manner.

2.1 Denavit-Hartenberg (D-H) coordinates transformation of robot [11]

In order to find a transformation from tool tip to the base of a manipulator, we have to define link frames and to derive a systematical technique, which allows describing the kinematics of a robot with n degrees of freedom in a unique way. A set of 4 X n parameter will turn out to be sufficient for that purpose. Figure 2 presents the first links of a kinematics chain. The base and each link i of the chain are assigned to a specific frame K_i, which is fixed to the link. So the position and orientation of a link frame changes with respect to a neighboring link frame according to the motion of their connecting joint. Therefore coordinate frame K_i can be described from its precedent link frame K_{i-1} by means of a homogeneous transformation. This homogeneous transformation includes the joint angle (for rotary joints) or the joint offset (for prismatic joints). Finally the tool frame K_n can be transformed to the base frame by multiplication of all the link transformations through the kinematics chain. So in order to transform any position/orientation relative to the tool frame (e.g. by sensors attached there) towards the manipulators base frame (e.g. where it is fixed to the floor), the sorted sequence of homogeneous transformations from tip to toe via K_{n-1}, K_{n-2}, -----, K_0 has to be processed. The remaining task is to set up all the homogeneous transformation matrices for a particular type of kinematics chain, considering geometrical attributes of link arrangements and types of joints. To facilitate calculations, engineers use the Denavit-Hartenberg convention to help them describe the positions of links and joints unambiguously. Every link gets its own coordinate system. There are a few rules to consider in choosing the coordinate system:

1-the -axis is in the direction of the joint axis

2-the -axis is parallel to the common normal or if there is no common normal,

3-the -axis follows from the - and -axis by choosing it to be a right handed coordinate system

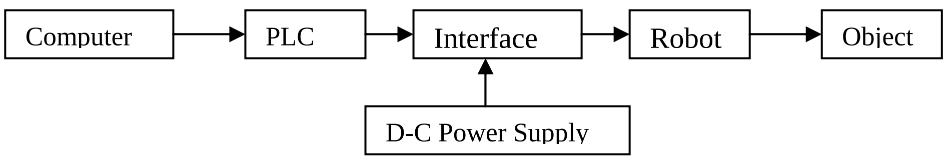

Figure 1 A Schematic diagram of a proposed system

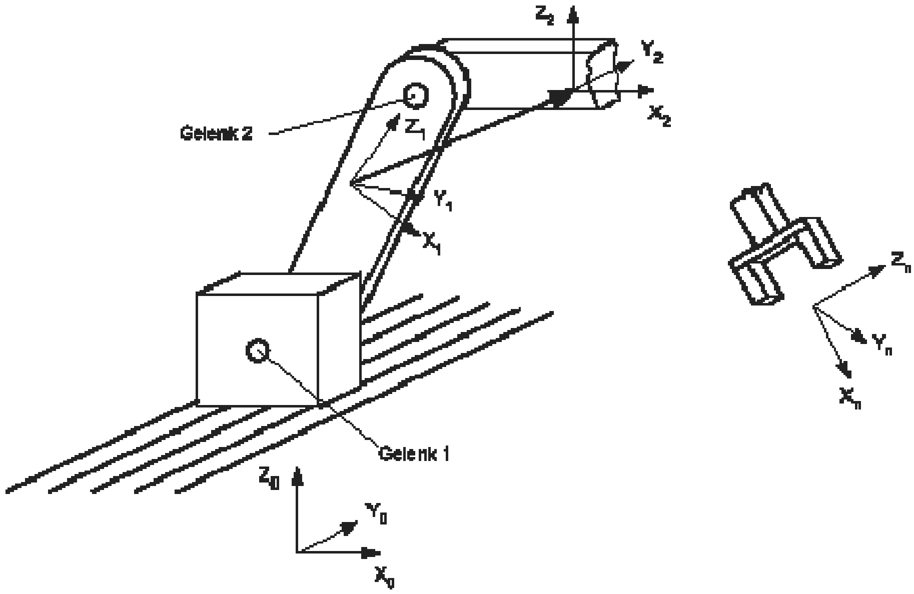

Figure 2 Links of a kinematics chain

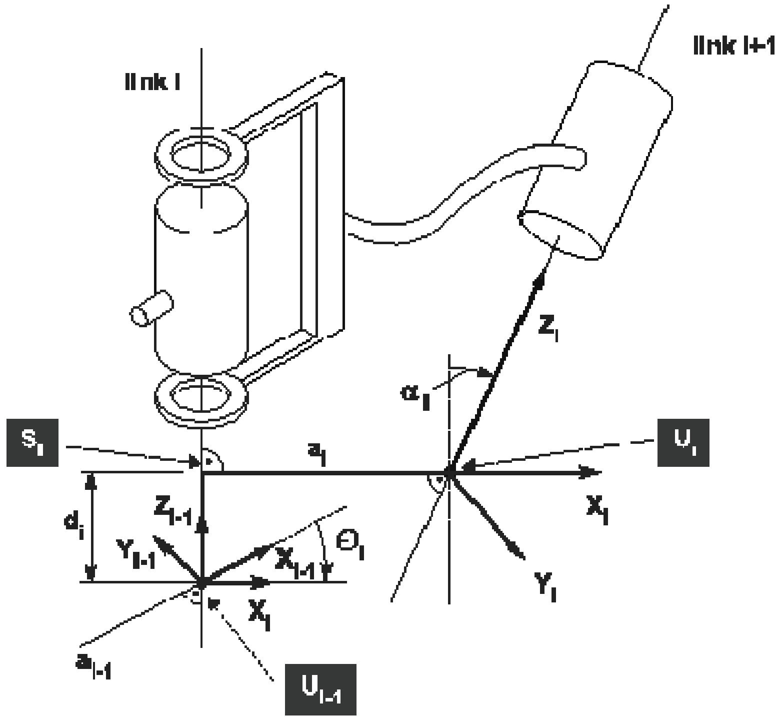

Figure 3 Assignment of DH-parameters

Every link/joint pair can be described as a coordinate transformation from the previous coordinate system to the next coordinate system. Figure 3 shows a detail of a kinematics chain, where two links are connected via a rotary joint. It is used to show, how invariant parameters are obtained which describe a link. The axes of rotation for link i and i+1 are extended towards straight lines G_i , G_{i+1} which in general are warped with respect to each others. The straight lines G_i and G_{i+1} have a common normal line a_i, which is perpendicular to both of them. The point of intersection U_i between straight line G_{i+1} and normal line a_i is defined as origin of link frame K_i. Further on, the base vector x_i is defined as extension of the normal line, whereas the base vector Z_i of this link frame is assumed to be matching the straight line G_i and thereby describing the axis of rotation for link i+1. Now given the vectors x_i and Z_i, the remaining base vector of this frame y_i is chosen appropriately in order to create a right-hand coordinate frame. Let us assume that the same operation has already been carried out for the precedent links. In this case, coordinate frame K_{i-1} is determined with its origin U_{i-1}located on straight line G_i. We are now able to derive some necessary parameters for link description. Parameter a_i is already introduced, whereas the second characteristic parameter d_i is defined here as distance between origin U_{i-1} of frame i-1 and the intersection S_i of normal line a_i (but this time) with straight line G_i. Obviously the origin of frame i is located in the plane, which is stretched out by the pair of joint rotation vectors Z_{i-1} and Z_i. One more characteristic link parameter is the angle α_i, which occurs between both joint rotation vector Z_{i-1} and Z_i. For link i, the rotation of its rigid body i (with respect to its rigid body i-1) is given by the rotation angle Θ_i . In other words, angle Θ_i is located between X_{i-1} -axis and X_i -axis. For revolute joints, joint parameters d_i , α_i and a_i are constant. They depend on joint design only and do not include any joint motion. The only variant parameter for revolute joints is the angle Θ_i, which describes variable joint positions. The situation is different for prismatic joints. Here parameter d_i becomes variable and describes translational joint positions, whereas parameters Θ_i , a_i and α_i depend on joint design only. However, in principle prismatic and revolute joints can be completely described with just 4 parameters:

a_i : Length of normal line $\overline{S_i U_i}$

α_i : Angle between Z_{i-1} and Z_i

d_i : Length of line $\overline{U_{i-1} S_i}$

Θ_i: Angle between X_{i-1} and X_i

A homogeneous transformation ${}^{i-1}T_i$ mapping frame K_i towards K_{i-1} via the actual link, can be derived now from the following geometrical transformations via the link under consideration:

1. Rotation around Z_{i-1} with angle Θ_i
2. Translation along Z_{i-1} with displacement d_i
3. Translation along X_{i-1} with displacement a_i
4. Rotation around X_i with angle α_i

or, in a more formal description with the help of homogeneous transformation matrices for each of the 4 actions mentioned above:

$$ {}^{i-1}T_i := \begin{pmatrix} \cos\Theta_i & -\sin\Theta_i & 0 & 0 \\ \sin\Theta_i & \cos\Theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

This leads to the following general description of transformations via prismatic or revolute joints:

$$ {}^{i-1}T_i = \begin{pmatrix} \cos\Theta_i & -\sin\Theta_i \cos\alpha_i & \sin\Theta_1 \sin\alpha_i & a_i \cos\Theta_i \\ \sin\Theta_i & \cos\Theta_i \cos\alpha_i & -\cos\Theta_i \sin\alpha_i & a_i \sin\Theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1) $$

Apparently homogeneous link transformations ${}^{i-1}T_i$ become especially simple, if the angular parameter α_i is set to some specific values, which are multiplicities of $\underline{+}\pi/2$. As parameter α_i is always constant, no matter whether a prismatic or a revolute joint is considered, simplifying the link transformation can be done through joint construction. From the physical viewpoint, α_i becomes a multiplicities of $\underline{+}\pi/2$, if neighboring axis are in parallel or perpendicular to each others. In fact, most of the industrial robots are designed according to these considerations. Observing this construction rule does in no way restrict any practical demands on the robot. Thereby some necessary calculations are facilitated for the robot controller, for example transformations from Cartesian task descriptions to joint motion interpolation, and vice versa. One more practical demand for robot design has to be mentioned here: There is one posture of the robot arm (of its open kinematics chain), where all the joints are in 'zero' position. This posture itself is called the arms' zero position. Usually it is a reference position for measurement of joint positions by internal sensors. Therefore the posture should be physically accessible for the robot.

2.2 The ABB industrial robot IRB 7600 [12,13]

As an example, we will consider, a robot IRB 7600 shown in Figure 4. For the DH-algorithm we will need some measurements of the different links. It consists of six links, six d-c motor, sensors and gripper. A new world of possibilities opens up with ABB's new power robot family. The IRB 7600 is ideal for weighty applications, regardless of industry. Typical areas can be handling of heavy fixtures, turning car bodies, lifting engines, handling heavy parts in foundries or forges, loading and unloading of machine cells, alternatively handling large and heavy pallet layers. With the IRB 7600 you can redefine the laws of automation, think big. We have set up three concepts to get your ideas started...

- Flex Framer - a highly flexible framing station
- Flex Positioner - IRB 7600 replaces the positioner
- Heavy handling - loads of up to 650kg, e.g. palletizing kegs

A major concern with robots handling payloads of up to 500 kg is to safeguard your personnel in the unlikely event of an accident, as well as to protect the robot itself and your investment. Therefore we have a range of software products called Active Safety.

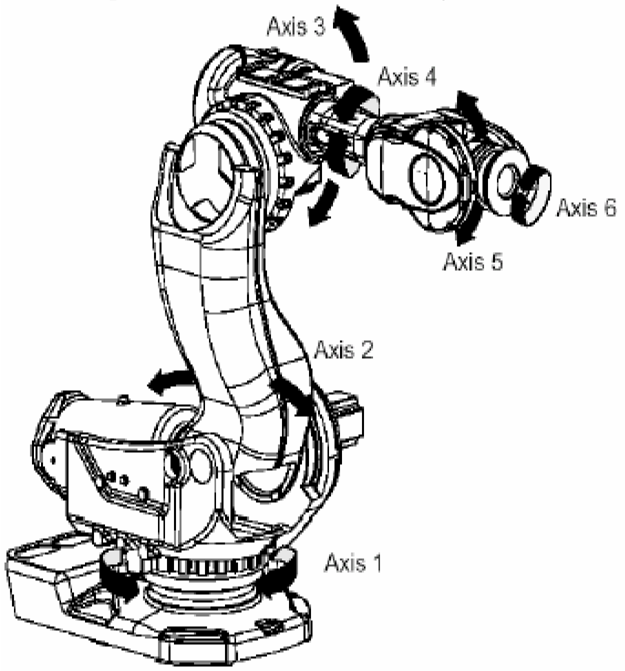

Figure 4 The ABB IRB 7600 manipulator model.

Apply the link parameters method (Denavit-Hartenberg motion), for model of robot exists in our Lab., the parameters of the robot are shown in table 2

Link i	α_i (rad.)	a_i (m.)	d_i (m.)	θ_i (rad.)
1	0	0	0	0.8028
2	$-\pi/2^o$	0	0	0
3	0	0.2	0	$-\pi/6^o$
4	$-\pi/2^o$	0	0.2	0
5	$\pi/2^o$	0	0	0
6	$-\pi/2^o$	0	0	0

Table 2 Link Parameters of the ABB IRB 7600 robot model

Apply equation (1), to determination the coordinate systems, e.g. according to Denavit-Hartenberg.

$$ {}^{0}_{1}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}_{2}T = \begin{bmatrix} 0.6946 & -3.34e-5 & -0.7193 & -1.0906 \\ 0.7193 & 3.226e-5 & 0.6946 & -1.1293 \\ 0 & -0.9999 & 7.962e-4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{3}_{4}T = \begin{bmatrix} 0.866 & -2.3e-5 & -0.5 & -1.36 \\ 0.5 & 4.02e-5 & 0.866 & -0.785 \\ 0 & -1 & 7.9e-4 & 0 \\ 0 & 0 & & 1 \end{bmatrix} \qquad (2) $$

$$ {}^{4}_{5}T = \begin{bmatrix} 1 & 0 & 0 & 1.57 \\ 0 & 4.6447e-5 & -1 & 0 \\ 0 & 0.9999 & 7.9627 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{5}_{6}\mathbf{T} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{-1.57} \\ \mathbf{0} & \mathbf{4.644e-5} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{-0.9999} & \mathbf{7.9627e-4} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} $$

We can obtain the product of six link D-H transform matrix for adjacent coordinate from frame 0 to frame 6

$$ {}^{0}_{6}\mathbf{T} = \begin{bmatrix} \mathbf{0.6016} & \mathbf{0.719} & \mathbf{-0.3484} & \mathbf{-2.1047} \\ \mathbf{0.6229} & \mathbf{-0.6949} & \mathbf{-0.358} & \mathbf{-2.1792} \\ \mathbf{-0.4999} & \mathbf{-1.4857} & \mathbf{-0.8659} & \mathbf{0.6117} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} $$

or $ {}^{0}\mathbf{P} = {}^{0}_{6}\mathbf{T}\ {}^{6}\mathbf{P} $ Where:

${}^{0}\mathbf{P}$ is initial position of robot, ${}^{6}\mathbf{P}$ is final position of the object and ${}^{0}_{6}\mathbf{T}$ Is the transformation matrix

$$ {}^{0}\mathbf{P} = \begin{bmatrix} \mathbf{P_{x0}} \\ \mathbf{P_{y0}} \\ \mathbf{P_{z0}} \\ \mathbf{1} \end{bmatrix}, \quad {}^{6}\mathbf{P} = \begin{bmatrix} \mathbf{P_{x6}} \\ \mathbf{P_{y6}} \\ \mathbf{P_{z6}} \\ \mathbf{1} \end{bmatrix}, \quad {}^{6}\mathbf{P} = [\mathbf{T}^{-1}]\ {}^{0}\mathbf{P} $$

and $\mathbf{T}^{-1}$=

$$ \begin{bmatrix} \mathbf{0.0697} & \mathbf{1.138} & \mathbf{-0.4985} & \mathbf{2.9315} \\ \mathbf{0.7169} & \mathbf{-0.6937} & \mathbf{-0.0016} & \mathbf{-0.0019} \\ \mathbf{-1.2704} & \mathbf{0.5333} & \mathbf{-0.8642} & \mathbf{-0.9828} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \quad \mathbf{(3)} $$

2.3 Path Trajectory Generation Based on the Kinetics in Robot

The environment the robots are moving in, have plane surface with absolutely no elevations. A frame of reference is considered in the plane. With respect to the origin of the reference frame, the initial and the final poses are defined. The path the robot has to trace is computed as a cubic polynomial. For smooth movement of the robot, the jerks have to be avoided or minimized, at particular points. The bicycle model with Ackerman steering type is used to model the robot. Considering a Cartesian co-ordinate frame, the pose of the robot is given as (x, y, θ) where x and y give the abscissa and ordinate of the centric of the robot, and θ gives the orientation i.e. angle of longitudinal axis of robot with respect to x-axis (See Fig. 5). The homogenous transform gives the translation and rotation of the robot body frame with respect to the reference frame and then T_{robot} can be obtained as follows:

$$ \mathbf{T_{Robot}} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & \mathbf{X} \\ \sin(\theta) & \cos(\theta) & \mathbf{Y} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} $$

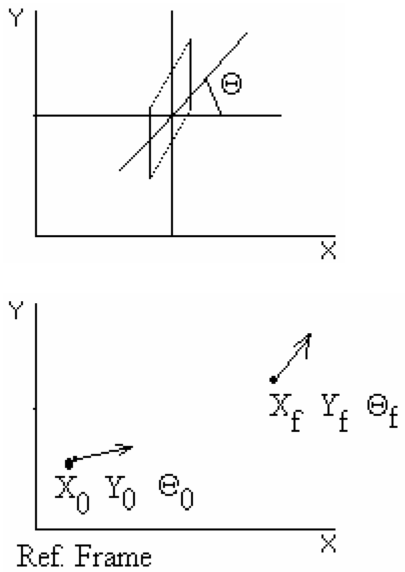

Fig. 5 Angle of longitudinal axis of robot with respect to x-axis

2.3.1 Cubic path for the robot

Curvature polynomials of cubic order are ideal primitive trajectories for robots. Unlike the clothoids, which are linear curvature polynomials, cubic curves can be used to determine a unique trajectory to an arbitrary target posture using a single continuous primitive. Such curves are also the lowest order curves which are continuous in the torque applied to steering mechanisms, so they generate trajectories which are relatively easily tracked by a real vehicle. Like the clothoids, cubic curvature polynomials are relatively difficult to compute but are easy to execute. There is a necessity for the smooth movement of the robot from one position to the other position in a plane. The cubic polynomials give a very smooth and kinetically flexible curve [14-15].

$$ Y = C_0 + C_1 X + C_2 X^2 + C_3 X^3 \quad (4) $$

Considering the initial position (X_0, Y_0, θ_0) and the final position as (X_f, Y_f, θ_f), .four different Equations are obtained as follows:

$$Y_0 = C_0 + C_1 X_0 + C_2 X_0^{2} + C_3 X_0^{3} \quad (5)$$
$$Y_f = C_f + C_1 X_f + C_2 X_f^{2} + C_3 X_f^{3} \quad (6)$$
$$\tan \Theta_0 = C_1 + 2 C_2 X_0 + 3 C_3 X_0^{2} \quad (7)$$
$$\tan \Theta_f = C_1 + 2 C_2 X_f + 3 C_3 X_f^{2} \quad (8)$$

Solving the equations. 5, 6, 7 and 8. We get C_0, C_1, C_2, and C_3 and hence the cubic path for the robot. The length of the cubic curve is very necessary to compute the time constraints for generating the position trajectories. It is given by S_t as follows:

$$S_t = \int_{X_0}^{X_f} \sqrt{1+(C_1+2C_2X+3C_3X^2)^2} \quad (9)$$

2.4 A Programmable Logic Controller

The Programmable Logic Controller (PLC), like a computer, employs a microprocessor chip to do the processing and memory chips to store the program. The PLC consists of a processor, input/output devices or I/O and a programmer. The input devices are sensors that monitor the machine or the process being controlled. The status of these sensors (ON or OFF) is fed to the PLC controller. Depending upon the input status of these sensors the outputs of the PLC may be switched on to energize motors, relays to control the robot.

3. Experimental Setup and Results

All control experiments were conducted at Electrical Engineering Department, Faculty of Engineering, Minoufiya University. The experiments were performed on a ABB IRB 7600 manipulator model. The experimental setup is shown in Fig.6. To evaluate the control and planning algorithms developed in the previous sections. The robot gripper catches an object from a certain position then transports it to another position. The robot links was controlled via a PLC. D-C Power Supply Voltage is used for delivering motors of the robots. Extensive experiments have been performed in different cases of the object (light load).

3.1 Interfaces between PLC and Robot

As shown in Fig.7, the interface board, it consists of a socket 25 pin (female) , cable 25/25 pin male/female connected to robot and connectors of supplying power to motors of links of robot. This interface is used for controlling the movement direction of motors of the joints of links by using a PLC output relay contactors. The Type of PLC used is Allen Bradely SLC 100, the number of inputs is 10 and number of output is 6.

Fig.6 Diagram of experimental setup for control experiments

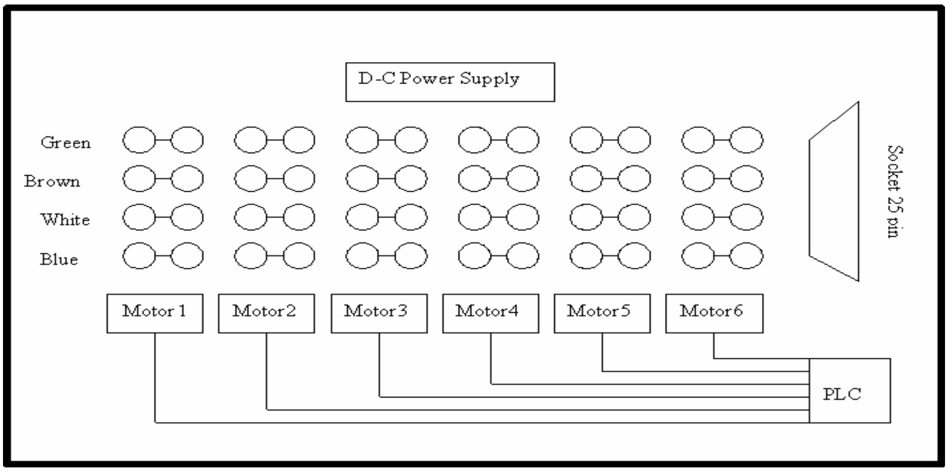

Figure 7 Interfaces to PLC

The description of 25 bin connector lies in the interface board which represents the motor number, wire color and bin number is shown in Table 3

No.	Color	Motor No.	Notes
1	blue	Motor 1, Motor 2, Motor 3, Motor 4, Motor 5 and Motor 6	Common
2			Not used
3	white	Motor 1	Monitoring
4	green	Motor 1	-ve dc supply
5	brown	Motor 1	+ve dc supply
6			Not used
7	white	Motor 2	monitoring
8	green	Motor 2	-ve dc supply
9	brown	Motor 2	+ve dc supply
10			
11	white	Motor 3	Monitoring
12	green	Motor 3	-ve dc supply
13	Red	Motor 1, Motor 2, Motor 3, Motor 4, Motor 5 and Motor 6	common
14	brown	Motor 6	+ve dc supply
15	green	Motor 6	-ve dc supply
16	white	Motor 6	Monitoring
17	brown	Motor 5	+ve dc supply
18	green	Motor 5	-ve dc supply
19	white	Motor 5	Monitoring
20			Not used
21	green	Motor 4	-ve dc supply
22	brown	Motor 4	+ve dc supply
23	white	Motor 4	Monitoring
24			Not used
25	brown	Motor 3	-ve dc supply

Table 3 Description of 25 bin connector

Note that Green wire and brown wire are the two terminals of the motor shown in Fig.8 which are connected to d-c supply. Study of each motor (direction) when brown wire is connected to +ve, green wire is connected to GND and vice versa.

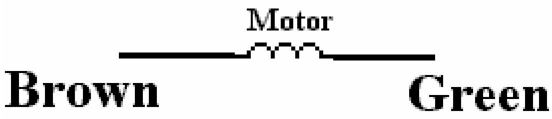

Figure 8 Color of d-c motor terminal

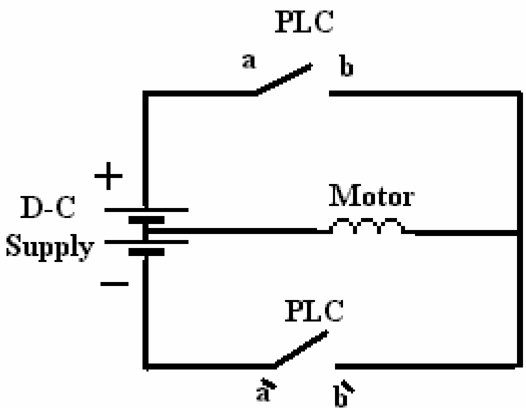

Figure 9 a, b, a' and b' are relay contactor of output terminal of PLC

By using a PLC, Connecting points a and b (output relay contactor of PLC) shown in Fing.9, means that the motor of one joint of robot rotates in a certain direction and on the other hand, Connecting points a' and b' means that rotate the motor of this joint in the other direction. So, it may be control the direction of the links of the robot in clock wise or anti-clock wise direction. In our experiment, six outputs relay address of the PLC are used 011, 012, 013, 014, 015, and 016. To control the rotational motion of robot base clockwise and anticlockwise, addresses 011, 012 can be used. Moving link of robot up and down can be achieved by addresses 013, 014 and Open and close gripper can be controlled by using address 015, 016. So the steps (rungs) of the program can be written to rotate the robot base, move the link of the robot, open gripper in order to catch an object and return back of the robot to its initial condition and release the object The program is written in ladder diagram shown in Fig. 10, to move the links of the robot from initial condition, open gripper, then catch an object from the final position, return back to the initial position of the robot and release the object. This cycle of movement of robot is repeated to transport another object. In this program, Only two statements for controlling the robot movement, first statement is an output Time driven sequencer (SQ0) with address 901 and is active if input switch 1 is on. The second statement is the reset of the sequencer by switching on switch 2. The sequencer time driven consists of sequential steps. Each step is executed for a certain time with output relay of PLC from 011 to 018 are either on or off. In other words, a sequence of steps is shown in table 4. The flow chart of the controlling links in order to catch an object (from initial position to another final position) is represented in Fig.11. It gives time intervals for controlling motors in joints of links. This time intervals may be changed according to the initial position of links and gripper of a robot and the position of the object. So, it is clear that the control of robot link is achieved by PLC with a minimum hardware and small length of a program. It is easy to change the time intervals of the steps in table 4 for a sequencer, if the object is moved in another position.

3-2 Trajectory Generation for Robot Using Curvature Polynomials

In this section, we present results for a cubic polynomial reference trajectory. The reference consists of two consecutive trajectories; the first trajectory of movement of robot is for the rotation of the base of the robot for 4 sec.

From initial position $X_0 = 0$, $Y_0 = 0$ and $\Theta_0 = 0$ to final position $X_f = 20$ cm. $Y_f = 20$ cm. and $\Theta_f = 45$ deg. To evaluate the first trajectory, solving the equations. 5, 6, 7 and 8. We get $C_0 = C_1 = 0$, $C_2 = 10$, and $C_3 = -25$ and hence the first cubic path for the robot is $Y = 10 X^2 - 25 X^3$, and $\tan \Theta = 20 X - 75 X^2$, By Solving Eq. 9, the length of the first cubic curve is 29.51 cm. The second trajectory is for the movement of the arm down from $X0 = 0$, $\underline{Y}_0 = 0$ and $\underline{\Theta}_0 = 0$ to final position $\underline{X}_f = 25$ cm. , $\underline{Y}_f = 25$ cm. and $\underline{\Theta}_f = 0$, We get $C_0 = C_1 = 0$, $C_2 = 12$, and $C_3 = -32$ and hence the second cubic path for the robot is $Y = 12 X^2 - 32 X^3$, and $\tan \Theta = 24 X - 96 X^2$
By Solving Eq. 9, the length of the second cubic curve is 30.66 cm.

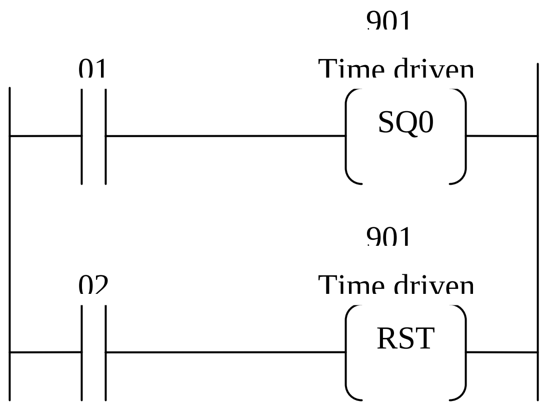

Fig.10 The ladder diagram program for controlling robot

4. Conclusion

The problem was to control links of robot with a minimum hardware; simple program is designed for catching an object from a certain position and transporting it to final position by using a Programmable Logic Controller (PLC). As a result, this paper presents a powerful control with very satisfying results for system. The result is a high dynamic control with robust stability and robust performance, which might be impossible with standard controller structures. In addition, the control compensates disturbances with a good performance. A simulation and modeling of the robot based control by using PLC are obtained. The control was tested in an experimental set-up with successful results. The PLC is very interesting for controlling different links of robot.

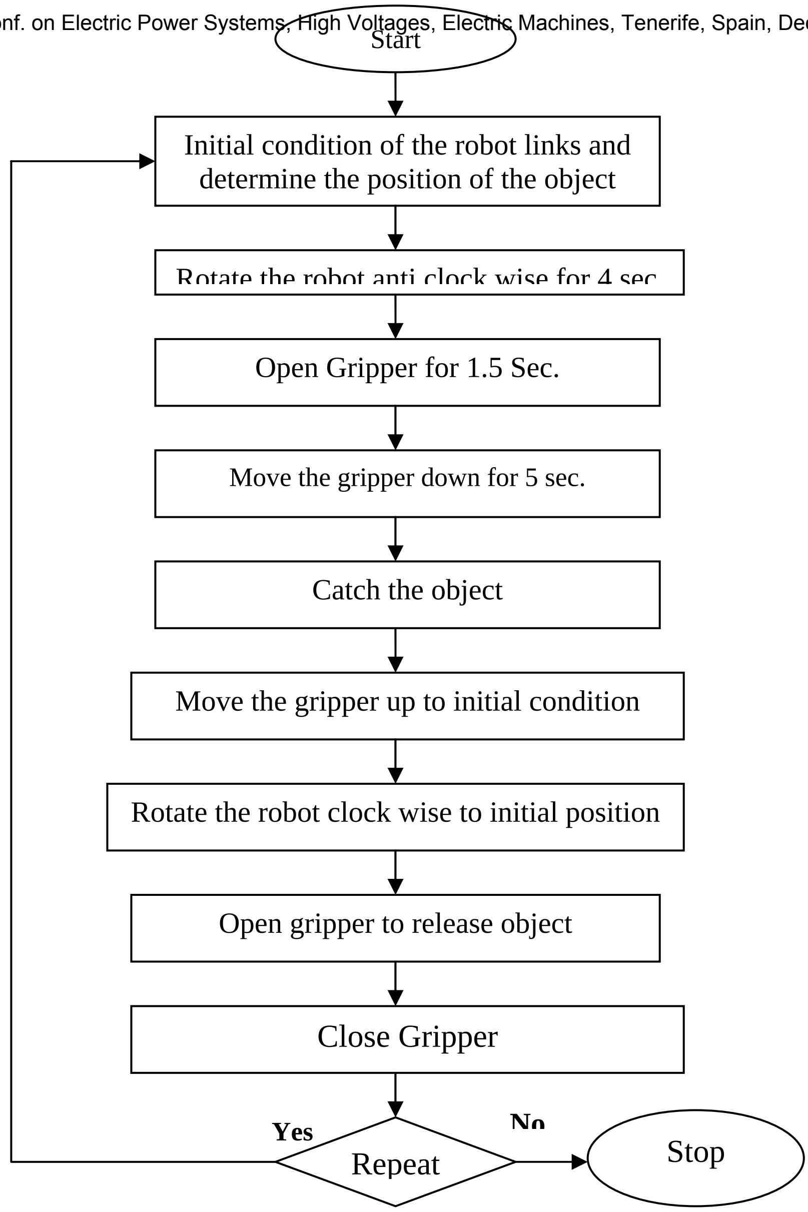

Fig.11 The flow chart of the controlling links in order to catch an object

Bit Address		Output Address								Time in sec.
		018	017	016	015	014	013	012	011	
Mask data		0	0	1	1	1	1	1	1	
Step	Hex. Data	Binary Data								
0	00			0	0	0	0	0	0	2
1	01			0	0	0	0	0	1	4
2	00			0	0	0	0	0	0	2
3	10			0	1	0	0	0	0	1.5
4	00			0	0	0	0	0	0	2
5	04			0	0	0	1	0	0	5
6	00			0	0	0	0	0	0	2
7	20			1	0	0	0	0	0	3
8	00			0	0	0	0	0	0	2
9	08			0	0	1	0	0	0	2.5
10	00			0	0	0	0	0	0	2
11	02			0	0	0	0	1	0	4
12	00			0	0	0	0	0	0	2
13	10			0	1	0	0	0	0	5
14	00			0	0	0	0	0	0	2
15	20			1	0	0	0	0	0	3

Table4 The different intervals time for sequencer used to control a robot link

5. References

[1] Hidetaka Ohno and Shigeo Hirose, "Study on Slime Robot", IEEE/RSJ International Conference on intelligent Robots and Systems, pp.2218-2222, 2002

[2]Lewis F.L., Abdallah C.T., Dawson D.M., "Control of Robot Manipulators", Macmillan Publishing Company, New York 1993.

[3] Samson, Le Borgne,Espiau, C.Samson, M.Le Borgne, B.Espiau, "Robot Control" , Oxford University Press, Oxford,1991.

[4] Eppinger S., Seering W., "Introduction to Dynamic models for robot force control", IEEE Control Syst. Mag., Vol.7, no.2,pp.48-52,Apr.1987.

[5] Tzyh- Jong Tarn, Antal K. Bejczy, Xiaoping Yun, Zuofeng Li, "Effect of motor Dynamics on Nonlinear Feedback Robot Arm", Control, IEEE Transactions on Robotics and Automation, vol.7, No.1,733-742, February 1991.

[6] R. Beekman and K .Lee, "Nonlinear robotic control including drive motor interactions" , Proc. Amer. Control Conf.p.1333-1338,Atlanta USA,1988.

[7] M.Renaud, "Contribution à la modélisation et à la commande dynamique des robots manipulateurs", Thèse d'Etat, Université P.Sabatier, Toulouse, 1980

[8]P.Ph.Robet, and M.Gautier, "Nonlinear Feedback Robot Arm Control", EPE-PEMC2002, Cavtat & Dubrovnik, Croatia, 9-11 September 2002.

[9] Štefan SZABÓ and, Vladislav SINGULE, " Implementation of CAN communication protocol in autonomous mobile robot control", EPE-PEMC2002, Cavtat & Dubrovnik, Croatia, 9-11 September 2002.

[10] Suthep Jamjai, "The Design of 3D Robot Control by PLC", Thesis, http://library.kmitnb.ac.th, 1996

[11]Andreas Jochheim, Michael Gerke and Andreas Bischoff, "Modeling and simulation of robotic systems", Control Systems Engineering group, Department of Electrical Engineering, University of Hagen , D-58084 Hagen, Germany

[12] ABB. Product speci_cation irb 7600, m2000 rev 2. www.abb.com.

[13] M. W. Spong and M. Vidyasagar. Robot Dynamics and Control. Wiley, 1989.

[14]Abhinav A.Kalamdani, "Path and actuator trajectory generation, based on the kinetics, in mobile robots", Instrumentation Systems and Automation Society (ISA) Bangalore Section St, India, October 2002. www.contrib.andrew.cmu.edu/~akalamda/abhinav_trajectory.pdf

[15] Bryan Nagy and Alonzo Kelly, "Trajectory Generation for Car-Like Robots Using Cubic Curvature Polynomials", Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3890, in Field and Service Robots 2001, url: http://www.ri.cmu.edu/people/nagy_bryan_ext.html, http://www.frc.ri.cmu.edu/~alonzo

6. Symbols

x_0: Initial position of robot.
x_f: final position of robot.
a_i : Length of normal line $\overline{S_i U_i}$
α_i: Angle between Z_{i-1} and Z_i
d_i : Length of line $\overline{U_{i-1} S_i}$
Θ_i: Angle between X_{i-1} and X_i
S_i: The intersection of normal line a_i with straight line G_i
z_i The base vector of link frame.
U_{i-1}: Origin position.
C_0, C_1, C_2 and C_3: The cubic polynomials constants.
S_f: The length of the cubic curve.
SQ0: Output sequencer
Rst: Reset