بسم الله الرحمن الرحيم

Palestine Polytechnic University

College of Engineering &Technology

Mechanical Engineering department

Mechatronics Engineering

Graduation Project

Rehabilitation and Control of Articulated Hydraulic Manipulator

Project Team

Majid Alhroosh **Mujahed Masharqa**

Project Supervisor

Dr.Yousef Alswati

Hebron-Palestine

2012

CERTIFICATION

Palestine Polytechnic University

PPU

Hebron-Palestine

Rehabilitation and Control of Articulated Hydraulic Manipulator

Prepared By

Majid Alhroosh Mujahed Masharqa

In accordance with the recommendations of the project supervisor, and the acceptance of all examining committee members, this project has been submitted to the department of mechanical engineering in the college of engineering and technology in the partial fulfillment of the requirement of department for the degree of Bachelor of Science in engineering.

Project Supervisor signature

Committee Member signature

----------------- **--------------------** **-------------------**

Department Head Signature

Acknowledgment

We wish to thank our parents for their tremendous contributions and support both morally and financially towards the completion of this project.

We also grateful to our project supervisor Dr. Yousef Alswati who without his help and guidance this project would not have been completed.

I also show my gratitude to my friends and all who contributed in one way or the other in the course of the project.

Abstract

The articulated hydraulic manipulator is a manipulator with three revolute joints and attached gripper. Each joint in this robot is driven by a hydraulic actuator. This robot is belongs to the mechatronics laboratory of Palestine Polytechnic University (ppu), and it has a five degree of freedom.

We will use the logic controller (plc) to control the movement of the manipulator end-effectors in the working envelop, so by using the HMI touch screen the coordinate will be entered and the end effecter move to the desired location in the workspace of manipulator, hence we will design a robotic system to make the manipulator pick an object from one location and move it to another.

In order to make this, many challenging problems will be covered in this project; these problems are kinematic, dynamitic, actuation, motion planning, control and programming.

Table of Content

List of Tables

List of Figures

Chapter One

Introduction

1.1 Robotics overview

1.2 Robotic mechanical structure

1.3 Robotic systems

1.4 Classification of robots

1.5 project overview

1.6 Project Schedule

1.7 Report Content

1.1 Robotics overview

Robotics is concerned with the study of those machines that can replace human being in the execution of a task, as regards both physical activity and decision making.

At the present time, the industrial robots have a significant impact on the industry, such that the robot can improve the quality of life by freeing workers from dirty, boring, dangerous and heavy labor.

An official definition of such a robot comes from the Robotic Institute of America (RIA): a robot is a re-programmable multi-functional manipulator designed to move materials, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks.

1.2 Robotic mechanical structure

Robots are classified as those with fixed base "robot manipulators", and those with mobile base "mobile robots", in our project we have a robot with fixed base.

The mechanical structure of a robot manipulator consist of rigid link connected by a joint to form the kinematic chain, the joint can be revolute (rotary) or linear (prismatic).

In the case of revolute joint, this joint allows relative rotation between two links, these displacements are called joint angles .while prismatic joint allows a linear relative motion between tow links, which called the joint offset.

To construct the manipulator, the first link in a chain is connected to the base and the last link is connected to the end effecter, this end effecter can be anything from a welding device to a mechanical hand used to manipulate the environment. The kinematic chain of manipulator is characterized by number of degree of freedom (DOF).

1.3 Robotic systems

The basic component of robotic system is

- Manipulator (robotic arm)
- the end effecter (which is part of the manipulator)
- power supply
- the controller

And this can be viewed in figure (1.1)

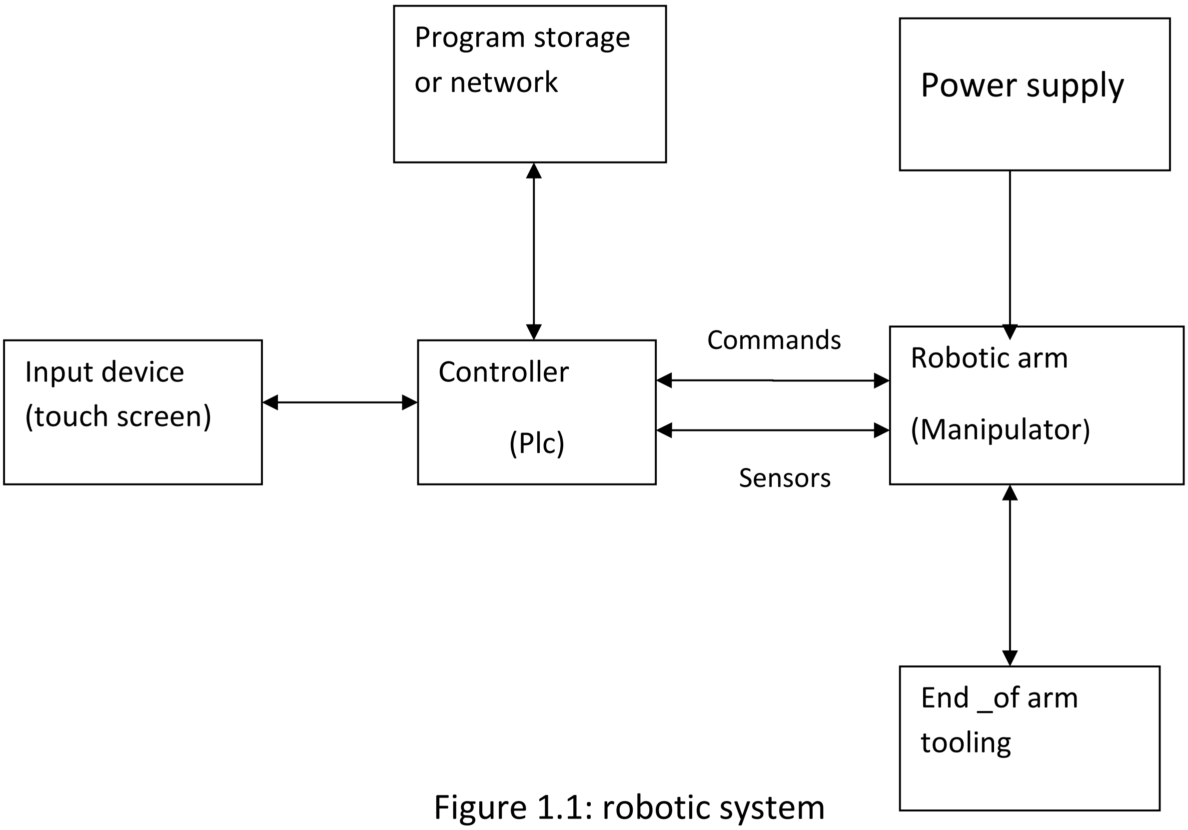


Figure 1.1: robotic system

The manipulator, which is the robotic arm, consists of segments joined together with axes capable of motion in various directions allowing the robot to perform work.

The end effecter, which is the gripper tool, a special device, or fixture attached to the robotic arm, actually performs the work.

The power supply provides and regulates the energy that is converted to motion by robotic actuator, and it may be either electric m pneumatic, or hydraulic.

The controller initiate, terminates, and coordinates the motion of sequences of a robot. Also it accepts the necessary input to the robot and provides the outputs to interface with the outside world.

1.4 Classification of robots

Robotic manipulators can be classified by several categories, such as their power source, geometry, application area, or their method control. Such classification is useful primarily in order to determine Which robot is righty for given task. For example, a hydraulic robot would not be suitable for food handling or clean room application.

Power source: Most robots are eclectically, hydraulically, or pneumatically powered. The advantage of use hydraulic power is that the hydraulic actuators are unrivaled in their speed of response and torque producing capability. These hydraulic robots are used primary for lifting heavy loads

Application area: Robots are often classified by application into assembly and non-assembly robots.

Method of control: Robots are classified by control method into servo (high technology) and non-servo (low technology) robots.

Geometry: Robot manipulators are usually classified cinematically on the basis of the first three joints of the arm. The majority of these manipulators fall into one of the five geometry types: articulated (RRR), spherical (RRP), SCARA (RRP), cylindrical (RPP), or Cartesian (PPP).

The common industrial manipulator is often referred to as a robot arm, with links and joints described in similar terms. Manipulators which emulate the characteristics of a human arm are called articulated arms (articulated manipulator). All their joints are rotary (or revolute).

The motion of articulated robot arms differs from the motion of the human arm. While robot joints have fewer degrees of freedom, they can move through greater angles. For example, the elbow of an articulated robot can bend up or down whereas a person can only bend their elbow in one direction with respect to the straight arm position

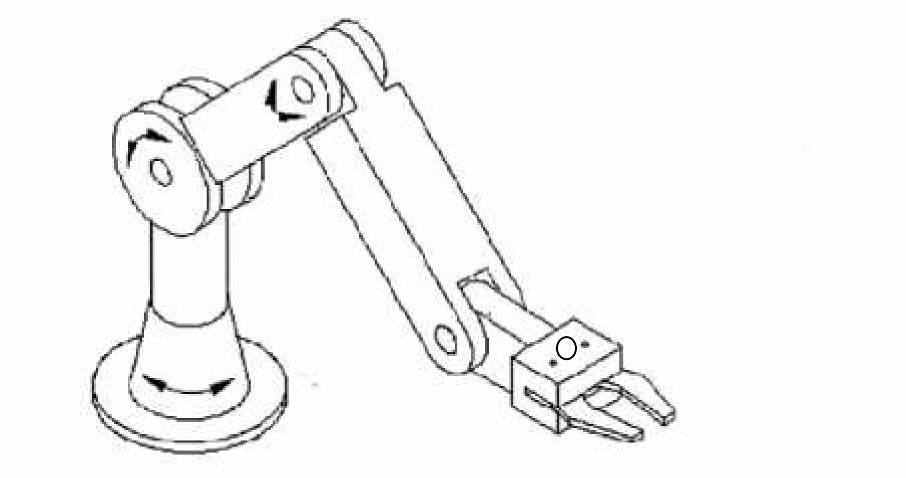


Figure 1.2: Articulated manipulator

1.5 project overview

Our work is concerned with rehabilitation and control of the articulated hydraulic robot that has five degree of freedom (figure 1.2). Each joint in this robot is driven by a hydraulic actuator. This robot is belongs to the mechatronics laboratory at Palestine Polytechnic University (ppu).

We will use a programmable logic controller (plc), specifically Schneider plc, to control the movement and orientation of the end-effecter in the working envelop, so by using the HMI touch screen the coordinate will be entered and the end effecter move to the desired location in the envelop.

This project also includes analysis, study of the kinematics, dynamics, trajectory planning, actuation and programming of the robot under consideration.

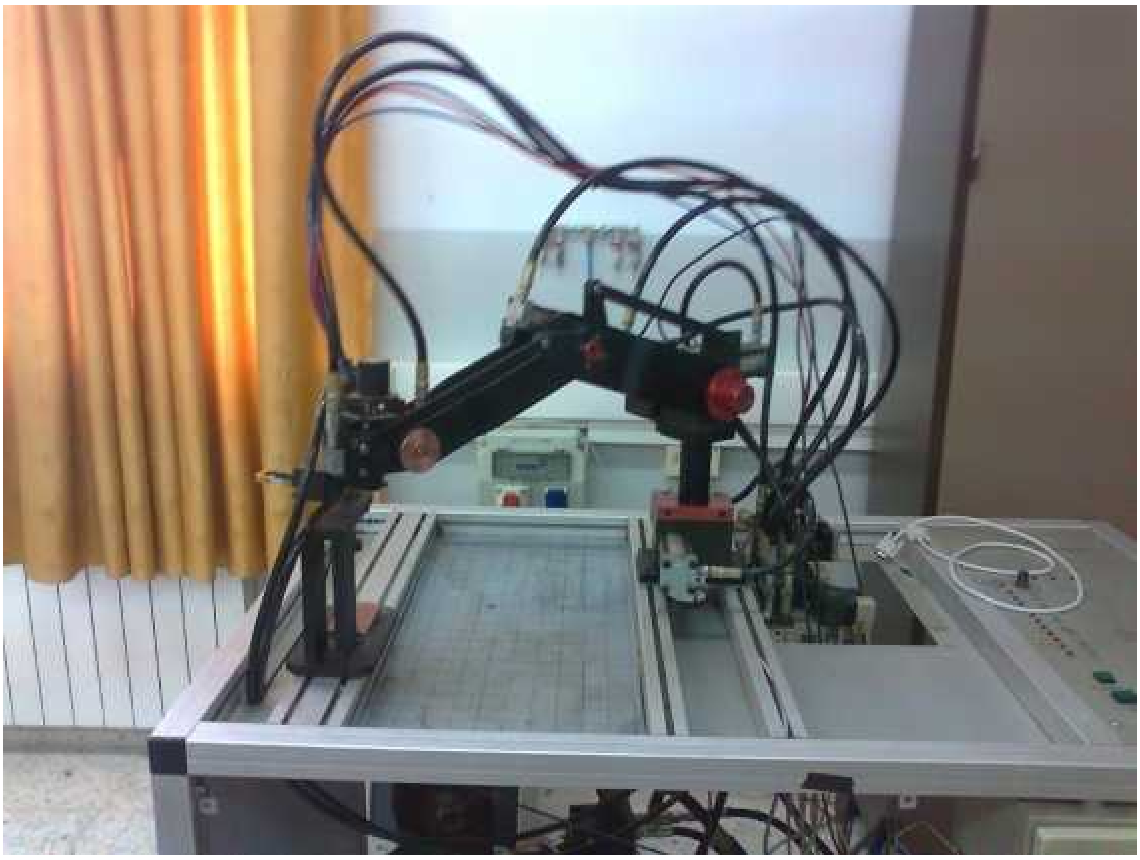


Figure 1.3: the articulated manipulator in the mechatronics lab (PPU)

1.6 Project Schedule

- **First Semester**

Process	**Week**															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Selected the project																
Collection of the needed data for the project																
Modeling of Articulated hydraulic robot																
Writing the Documentation																

Table1.1: first semester Project Schedule

- **Second Semester**

Process	**Week**															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Building Manual Control Circuits																
Programming																
Building PLC circuit																
Testing and Writing the Documentation																

Table1.2: seconded semester Project Schedule

1.7 Report Content

Now we provide a brief description of each chapter:

Chapter one: introduce an overview to robotic systems, classification of robots, and the project goal.

Chapter two: present solutions to the forward kinematics problems using Denevative-Haetenberg convention and to the inverse kinematics problem using Geometric Approach.

Chapter Three: introduce forward and inverse velocity kinematics using geometric Jacobean matrix, also this chapter provide a solution to singularities which is configurations that make the manipulator loss one or more degree of freedom.

Chapter Four: is concerned with describing motion of the manipulator in terms of trajectories through space.

Chapter Five: present we study the methods of controlling the manipulator (by digital controller) so that it will tack a desired position trajectory through space.

Chapter Two

Forward and Inverse Kinematics

2.1 Overview

Kinematics is the branch of classical mechanics that describes the motion of points or bodies without consideration of the causes of motion. To describe motion, kinematics studies the trajectories of points, lines and other geometric objects and their differential properties such as velocity and acceleration.

For the articulated manipulator we have to consider the forward and inverse kinematic. First consider the forward kinematic problem which is to determine the position and orientation of the end-effectors by given the values of joint variables of the robot. Then we solve the inverse kinematics problem which is to determine the values of the joint variables given the end-effectors position and orientation.

To perform the kinematics analysis, we must establish various coordinate frames to represent the position and orientations of rigid body objects, and with transformations among these coordinate frames.

2.2 Position and orientation representation

A rigid body (robot link) is completely described in space by its position and orientation with respect to reference frame. A coordinate reference frame i consist of an origin, denoted O_i , and a triad of mutually orthogonal bases vectors, denoted ($x_i y_i z_i$), that are all fixed within a particular body. The pose of a body will always be expressed relative to some other body, so it can be expressed as the pose of one coordinate frame relative to another. Similarly, rigid-body displacements can be expressed as displacements between two coordinate frames, one of which may be referred to as moving, while the other may be referred to as fixed. This indicates that the observer is located in a stationary position within the fixed reference frame, not that there exists any absolutely fixed frame.

2.2.1 Position and displacement

The position of body in coordinate frame denote by coordinate vector P and if we have two coordinate frames i and j , the position of the origin of coordinate frame i relative to coordinate frame $j$can be denoted by 3×1 vector

$$P_i^j = \begin{matrix} p_i^j\ x \\ p_i^j\ y \\ p_i^j\ z \end{matrix} \qquad (2.1)$$

The components of this vector are the Cartesian coordinates of O_i in the $j$frame.

A translation is a displacement in which no point in the rigid body remains in its initial position and all straight lines in the rigid body remains parallel to their initial orientations. The translation of a body in space can be represented by the combination of its positions prior to and following the translation. Conversely, the position of a body can be represented as a translation that takes the body from a position in which the coordinate frame fixed to the body coincides with the fixed coordinate frame to the current position in which the two frames are not coincident. Thus, any representation of position can be used to create a representation of displacement, and vice versa.

2.2.2 Orientation and Rotation matrix

In order to describe the orientation of body we will attach a coordinate frame to the body and then give a description of this coordinate system relative to the reference frame. In figure 2.1 coordinate frame **($x_i y_i z_i$)** haze been attached to the body in a known way. A description of frame **($x_i y_i z_i$)** relative to frame **($x_j y_j z_j$)** now suffices to give the orientation of the body. Thus, position of points is described with vectors and orientations of bodies are described with an attached coordinate frame. One way to describe the body-attached coordinate frame **($x_i y_i z_i$),** is to write the unit vectors of its three principal axes in terms of the coordinate frame ($x_j y_j z_j$**).**

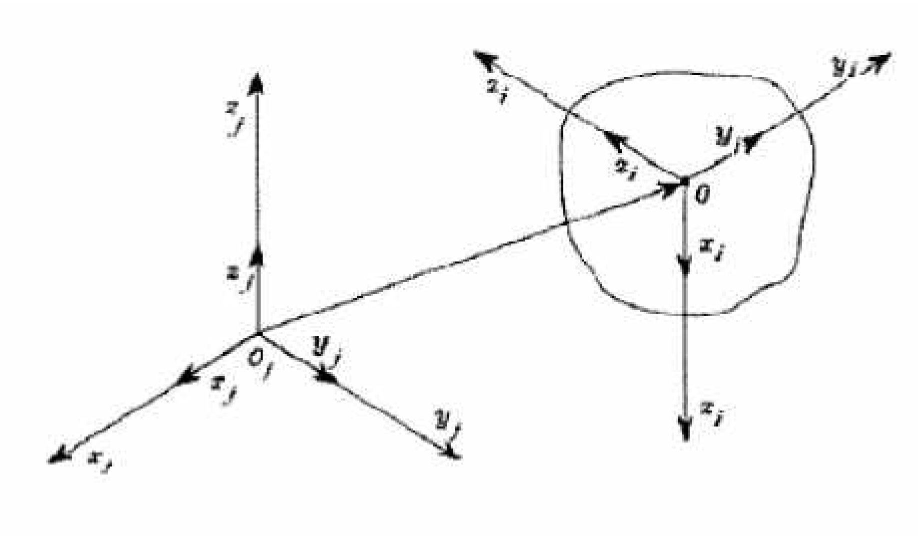


Figure 2.1: representing the rigid body in space as position and orientation

Rotation matrix

The orientation of coordinate frame i relative to coordinate frame j can bedenoted by expressing the bases vectors $(x_i y_i z_i)$in terms of the bases vectors $(x_j y_j z_j)$. This yields $(x_i^j y_i^j z_i^j)$, which can when written together as 3×3 matrix is known as the rotation matrix. The components of R_i^j are the dot products of basis vectors of the two coordinate frames.

$$R_i^j = \begin{matrix} x_i.x_j & y_i.x_j & z_i.x_j \\ x_i.y_j & y_i.y_j & z_i.y_j \\ x_i.z_j & y_i.z_j & z_i.z_j \end{matrix} \qquad (2.2)$$

Because the basis vectors are unit vectors and the dot product of any two unit vectors is the cosine of the angle between them, the components are commonly referred to as direction cosines. Thus, the columns of R_i^j specify the direction cosines of the coordinate axis of $(x_i y_i z_i)$ relative to coordinate axis of $(x_j y_j z_j)$

The set of n $\times$ n rotation matrices is known as the special orthogonal of order n, and is denoted by $\mathrm{SO(n)}$. for any $\mathrm{R} \in \mathrm{SO(3)}$ the following properties hold

- $\mathrm{R}^{\mathrm{T}} = \mathrm{R}^{-1}$
- The columns (and therefore the rows) of R are mutually
- Each column (and therefore each row) of R is a unit vector
- det $\mathrm{R} = 1$

Rotation matrices are combined through simple matrix multiplication such that the orientation of frame k relative to frame j can be expressed as

$$R_k^j = R_i^j R_k^i \tag{2.3}$$

2.2.3 Homogeneous Transformation

Homogeneous transformations combine rotation and translation onto one matrix. A homogeneous transformation has the form of

$$H = \begin{matrix} R & P \\ 0 & 1 \end{matrix}, \; R \in SO\,(3),\; P \in R^3,\; H \in R^{4x4} \tag{2.4}$$

Where R is the rotation matrix and P is the translational matrix.

Homogeneous transformation matrices can be used to perform coordinate Transformations between frames that differ in orientation and translation.

2.3 Parameterization of rotation: Euler Angles

A common method for specifying a rotation matrix in three independent quantities is to use Euler angles. Consider the fixed coordinate frame $o_0 x_0\, y_0\, z_0$ and the rotated frame $o_1\, x_1\, y_1\, z_1$ we can specify the orientation of frame $o_1\, x_1\, y_1\, z_1$ relative to the frame $o_0 x_0\, y_0\, z_0$ by three angles $(\emptyset, \theta, \psi)$ called Euler angles, and obtained by three successive rotation as follow. First rotate about the z-axis by an angle $\emptyset$, next rotate about the current y-axis by the angle θ,finally rotate about the current z-axis by an angle ψ. In terms of the basic rotation matrices the resulting rotational transformation can be generated as the product:

$$R_{zyz} = R_{z,\phi} R_{y,\theta} R_{z,\psi}$$

$$
= \begin{matrix} C_\Phi C_\theta C_\psi - S_\Phi S_\psi & -C_\Phi C_\theta S_\psi - S_\Phi C_\psi & C_\Phi S_\theta \\ S_\Phi C_\theta C_\psi + C_\Phi S_\psi & -S_\Phi C_\theta S_\psi + C_\Phi C_\psi & S_\Phi S_\theta \\ -S_\theta C_\psi & S_\theta S_\psi & C_\theta \end{matrix} \qquad (2.5)
$$

For the articulated manipulator the Euler angles are the angles of the wrist rotation so:

$\emptyset = \theta_5$, $\theta = \theta_4$, $\psi = 0$

The rotation matrix of frame $o_5\, x_5\, y_5\, z_5$ relative to wrist frame $o_3\, x_3\, y_1\, z_3$

$$
R_5^3 = \begin{matrix} C_\Phi C_\theta & -S_\Phi & C_\Phi S_\theta \\ S_\Phi C_\theta & C_\Phi & S_\Phi S_\theta \\ -S_\theta & 0 & C_\theta \end{matrix} \qquad (2.6)
$$

2.4 Forward kinematic:

A robot manipulator is composed of a set of links connected together by joints, the joints may be simple, such as a revolute joint, or a prismatic joint or they can be more complex such as a ball and socket joint.

A robot manipulator with n joints will have $n + 1$ links, since each joint connects two links, we number the joints from 1 to n, and we number the links from 0 to n, starting from the base. by convention joints i connects link $i - 1$ to link i.We will consider the location of joint i to be fixed with respect to $i - 1$.When joint i is actuated ,link i moves .Therefore ,link 0 (the first link)is fixed ,and does not move when the joints are actuated.

Joint 1 is called base, joint 2 is called shoulder, joint 3 is called elbow as shown in figure 2.2

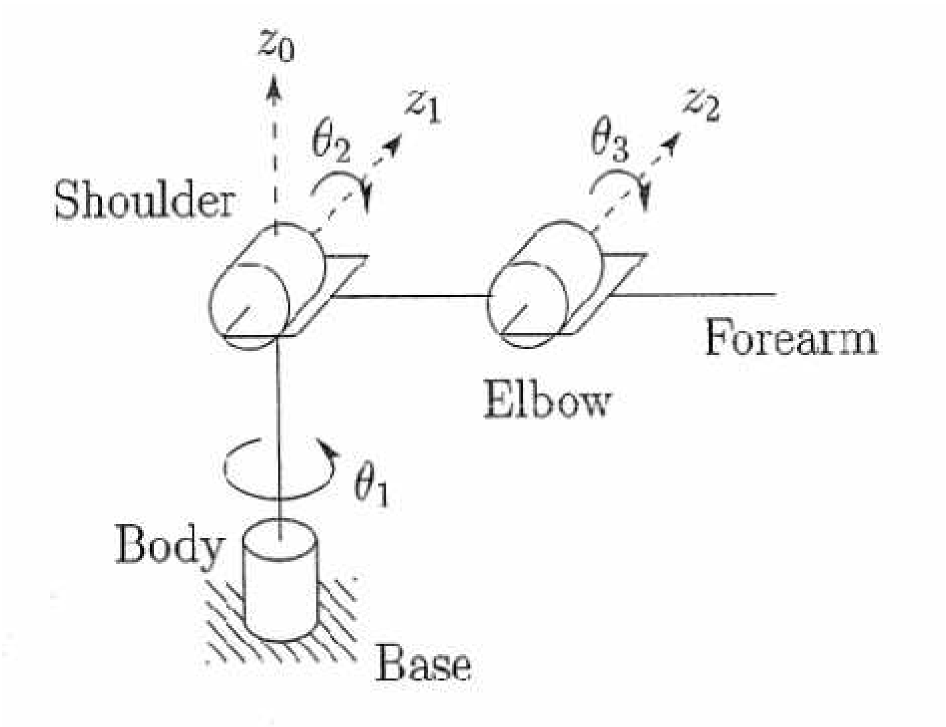


Figure 2.2: the symbolic representation of articulated manipulator

To perform the kinematic analysis, we attach the coordinate frame rigidly to each link .In particular, we attach $o_i\ x_i\ y_i\ z_i$ to link i.This means that, whatever motion the robot executes, the coordinates of each point on link i are constant when expressed in the i^{th} coordinate frame. Furthermore, when joint i is actuated, link i and its attached frame $o_i\ x_i\ y_i\ z_i$, experience a resulting motion. The frame $o_0 x_0\ y_0\ z_0$, which attached to the robot base, is referred to as the inertial frame. We will assign the coordinates frames that satisfy the Denative_Hartenberg convention.

The Denative_Hartenberg convention:

This convention is concerned about assigning a coordinate frame to each link in the manipulator .In this convention, each homogenous transformation A_i is representing as a product of four basic transformations.

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i}$$

$$= \begin{matrix} C_{\theta_i} & -S_{\theta_i}C_{\alpha_i} & S_{\theta_i}S_{\alpha_i} & a_i C_{\theta_i} \\ S_{\theta_i} & C_{\theta_i}C_{\alpha_i} & -C_{\theta_i}S_{\alpha_i} & a_i S_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{matrix} \qquad (2.7)$$

Where the four quantities θ_i,a_i, d_i and $\alpha_i$are parameters associated with link i and joint i .So if the DH convention is satisfied then the transformation matrix can be written as the above form.

The attached frame must have the following features according to the DH convention:

(DH1) the axis x_i is perpendicular to the axis z_{i-1}.

(DH2) the axis $x_i$intersects the axis z_{i-1}.

The tow features are shown in figure 2.3.

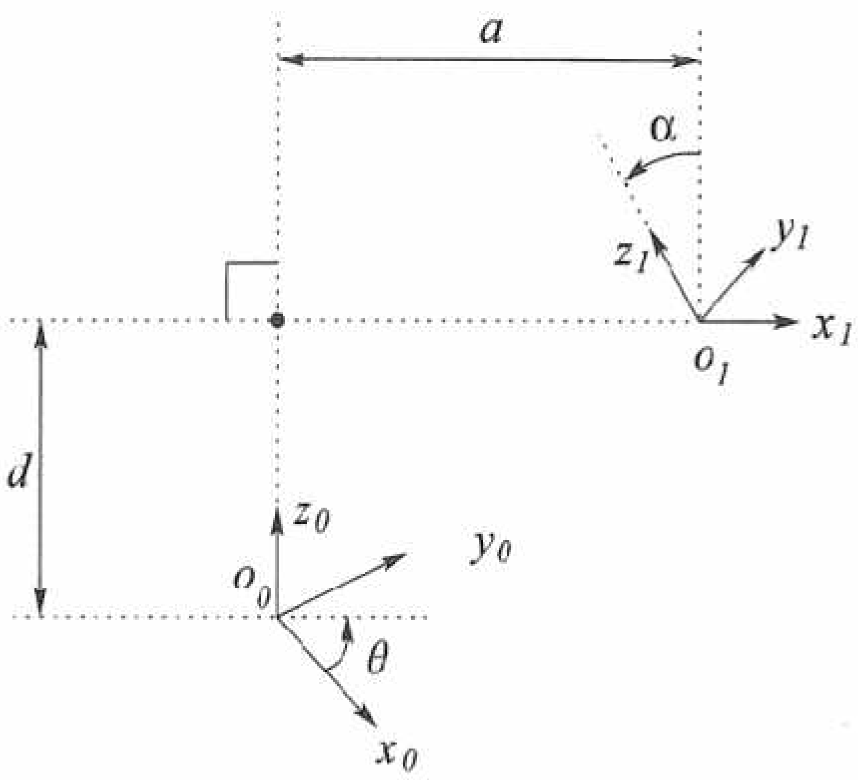


Figure 2.3: coordination frames showing DH1 and DH2.

Below we will describe the DH parameters that used in the transformation:

a_i : *is the distance between the axis* z_i *and* z_{i-1}*, and its measured along the axis* x_{i+1}

α_i: *is the angle between the axis* z_i *and* z_{i-1}*, measued in a plan normal to* x_{i+1}

θ_i: *is the angle from the axis* x_i *and* x_{i-1}*, measured in a plan normal to* z_i.

d_i: The distance from the origin o_{i-1} to the interconnection between x_i *ax*is and z_{i-1} *measured along the* z_i *axis.*

For the articulated manipulator we assign the coordinate frames that satisfy the DH convention and this is seen in figure 2.3.

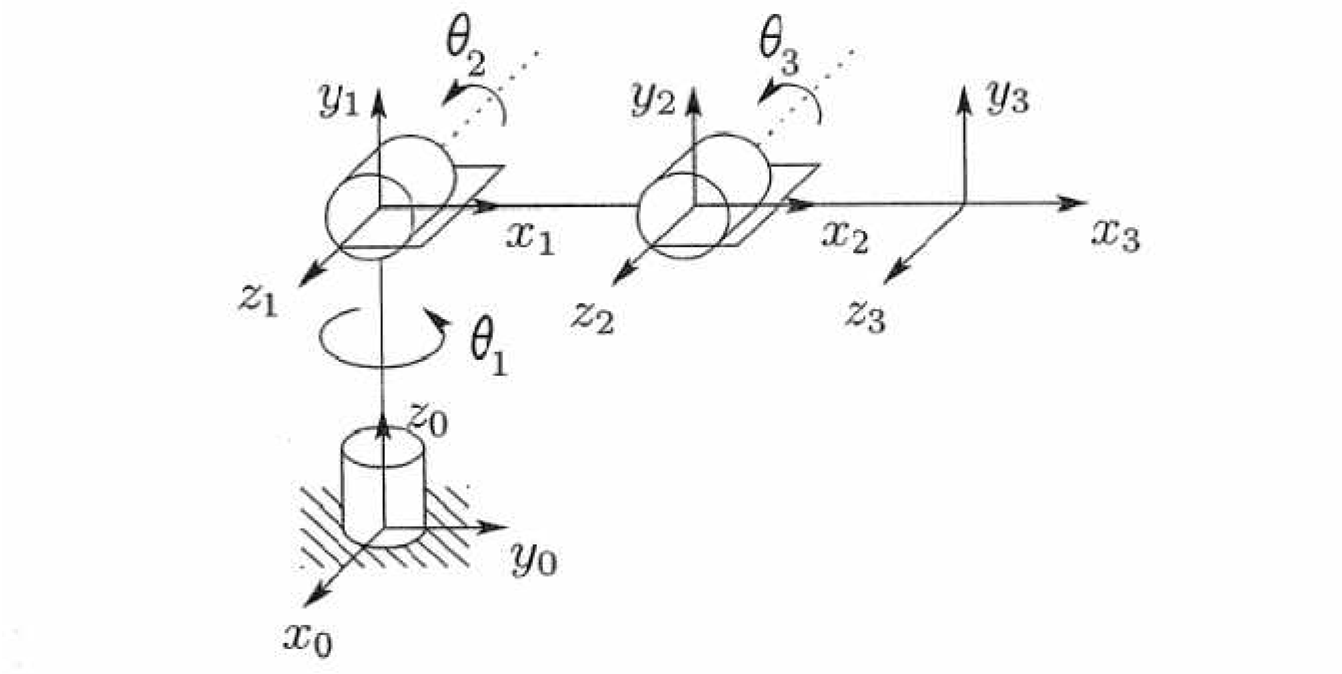


Figure 2.3: coordinates frame assigned for the articulated manipulator

The DH parameters for the articulated manipulator are shown in table 2.1:

Link	a_i	α_i	d_i	θ_i
1	0	$\pi/2$	a_1	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3

Table 2.1: DH parameters for the articulator manipulator.

The A-matrices are obtained using equation 2.5:

$$A_1 = \begin{matrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{matrix} \qquad A_2 = \begin{matrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$$A_3 == \begin{matrix} C_3 & -S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \qquad (2.8)$$

The homogenous transformation "T-matrices" are thus given by:

$$T_1^0 = A_1$$

$$T_2^0 = A_1A_2 = \begin{matrix} C_1C_2 & -S_2C_1 & S_1 & a_2C_1C_2 \\ S_1C_2 & S_1S_2 & -C_1 & a_2S_1C_2 \\ S_2 & C_1 & 0 & a_2S_2 + d_1 \\ 0 & 0 & 0 & 1 \end{matrix} \qquad (2.9)$$

$$T_3^0 = A_1A_2A_3 = \begin{matrix} C_1C_{23} & -S_{23}C_1 & S_1 & a_3C_1C_{23} + a_2C_1C_2 \\ S_1C_{23} & S_1S_{23} & -C_1 & a_3S_1C_{23} + a_2S_1C_2 \\ S_{12} & C_{23} & 0 & a_2S_2 + d_1 + a_3S_{23} \\ 0 & 0 & 0 & 1 \end{matrix}$$

Notice that the first three entries of the last column of T_3^0 are the x, y and z component of the origin $O_3$with respect to the base frame; that is,

$$x = a_3C_1C_{23} + a_2C_1C_2$$

$$y = a_3S_1C_{23} + a_2S_1C_2 \qquad (2.10)$$

$$z = a_2 S_2 + d_1 + a_3 S_{23}$$

are the coordinate of the end effecter with respect to the base frame. The rotational part of T_3^0gives the orientation of the frame O_3relative to the base frame.

2.5 Inverse kinematic:

The inverse kinematic problem is to find the joint variables in terms of the end effecter's position and orientation, the joint variable is the angle θ_i in the case of revolute joint, or d_i in the case of prismatic joint. Her we need a geometric approach to find $\theta_1, \theta_2, \theta_3$ that corresponds to a given position of the end effecter, called wrist center, and is represented by the point O_c.

The wrist center is the point between the arm and the end effecter and it has the coordinates of x_c, y_c, z_c, as shown in figure 2.4

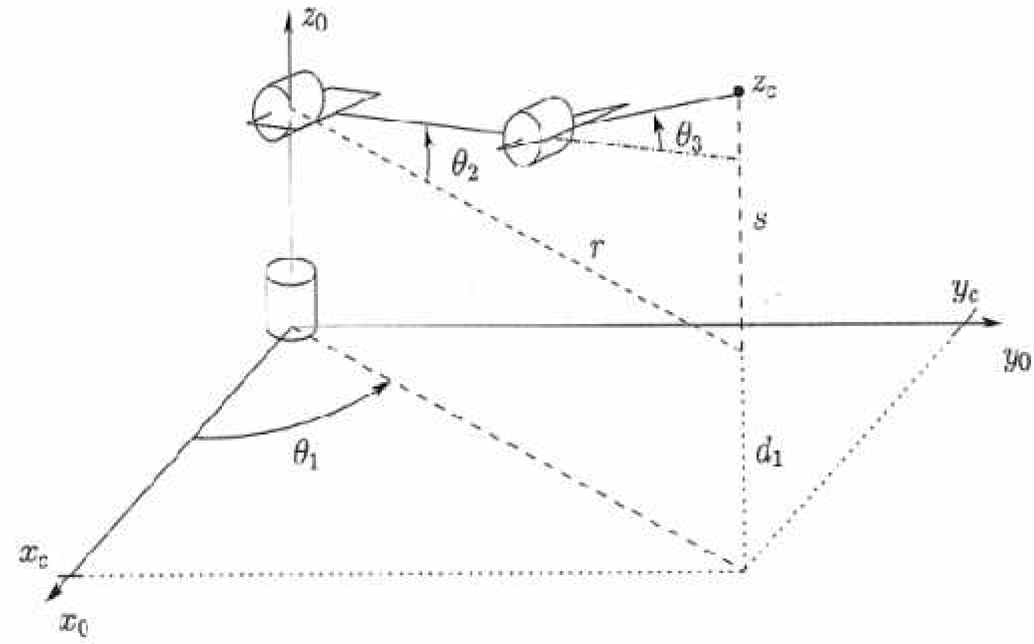


Figure 2.4: coordinates of O_c relative to the base frame

The general idea of the geometric approach is to solve for joint variable θ_i *for a revolute joint* by projecting the manipulator onto the $x_{i-1}-y_{i-1}$ plan and solving a simple trigonometry problem. For example, to solve for θ_1 we project the arm onto the x_0-y_0 plane and use trigonometry to find θ_1 as shown in figure 2.5.

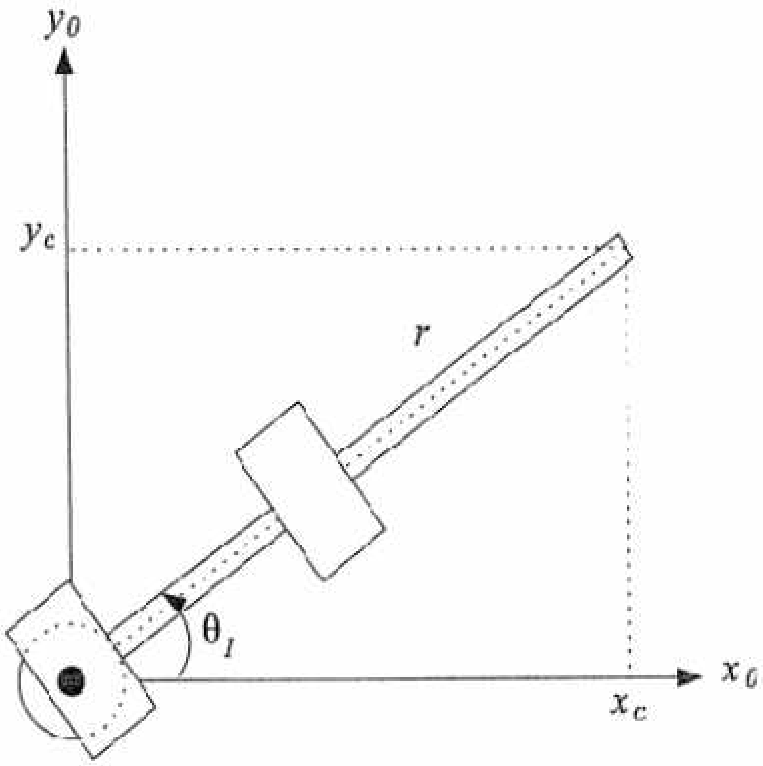


Figure 2.5: projection of the wrist center onto $x_0 - y_0$ plane.

We see from this projection that:

$$\theta_1 = Atan2\ (x_c, y_c) \tag{2.11}$$

In which $Atan2\ (x, y)$ denotes the two argument arctangent function.

To find the angles θ_2, θ_3 for the articulated manipulator given θ_1, we consider the plane formed by the second and the third links as shown in figure 2.6. Since the motion of the second and third link is planar:

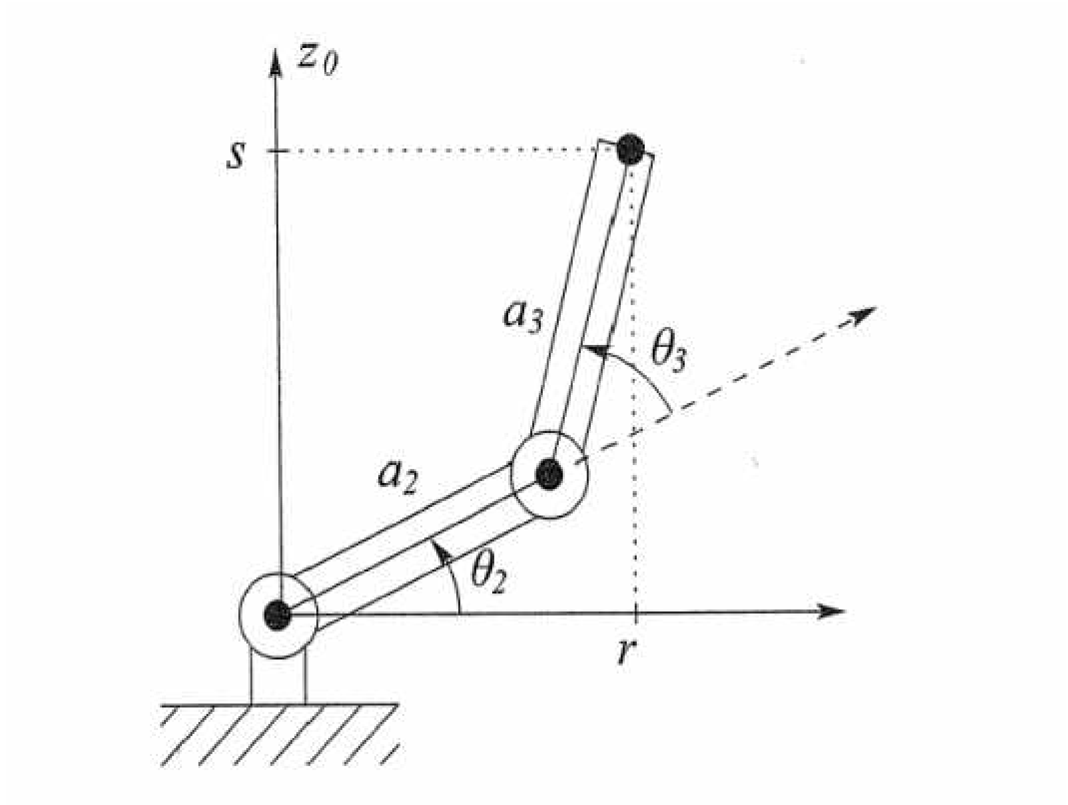


Figure 2.6: projecting onto the plan formed by links 2 and 3.

θ_2 Is given by:

$$\theta_2 = Atan2(\gamma, s) - Atan2(a_2 + a_3c_3, a_3s_3) \tag{2.12}$$

$$= Atan2(\sqrt{x_c^2 + y_c^2 - d^2}, z_c - d_1) - Atan2(a_2 + a_3c_3, a_3s_3)$$

Using the law of cosines we see that the angle θ_3 is given by:

$$\cos\theta_3 = \frac{\gamma^2 + S^2 - a_2^2 - a_3^2}{2a_2a_3}$$

$$= \frac{x_c^2 + y_c^2 - d^2 + (z_c - d_1)^2 - a_2^2 - a_3^2}{2a_2a_3} = D \tag{2.13}$$

Since $\gamma^2 = x_c^2 + y_c^2 - d^2$ and $s = (z_c - d_1)$. Hence, $\theta_3$is given by

$$\theta_3 = Atan2(D, \pm\sqrt{1 - D^2}) \tag{2.14}$$

The tow solutions for θ_3 is corresponds to the elbow-down position and elbow-up position, respectively, as shown in figure 2.7.

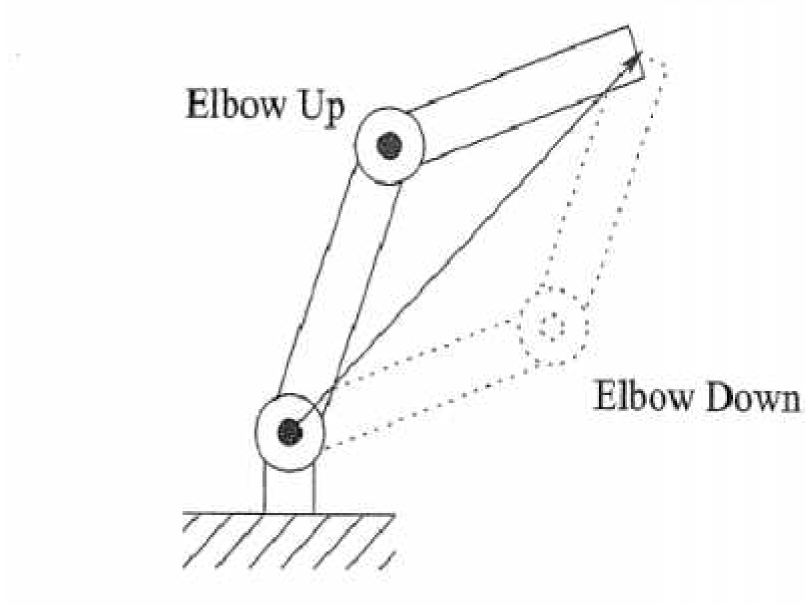


Figure 2.7: Elbow-up position and Elbow-down position

For the articulated manipulator we can use the inverse orientation techniques to find the final two joint variables using the rotation matrix described in equation 2.6:

$$\theta_4 = Atan2(c_1 c_{23} r_{13} + s_1 c_{23} r_{23} + s_{23} r_{33},$$

$$-c_1 s_{23} r_{13} - s_1 s_{23} r_{23} + c_{23} r_{33})$$

$$\theta_5 = Atan2\left(s_1 r_{13} - c_1 r_{23}, \pm\sqrt{1 - \left(s_1 r_{13} - c_1 r_{23}\right)^2}\right)$$

Where r_{ij} are the elements of Euler rotation matrix R_5^3.

2.6 The Workspace:

The workspace of a manipulator is the total volume swept out by the end effecter as the manipulator executes all possible motions. The workspace is constrained by the geometry of the manipulator as well as mechanical constraints of the joints. For example, a revolute joint may be limited to less than a full $360°$ of motion. The mechanical limits in the articulated manipulator limit the motion of a revolute joint to the values that appears in table 2.2. These values are measured experimentally.

Axis Movement	Axis Rang
Axis 1: base rotation	$-67 < \bar{\theta}_1 > 113$
Axis 2 : Shoulder rotation	$-45 < \bar{\theta}_2 > 45$
Axis 3 : Elbow rotation	$-135 < \bar{\theta}_3 > -45$

Table 2.2: the axis rang for each joint

The workspace for the articulated manipulator I shown in figure 2.8:

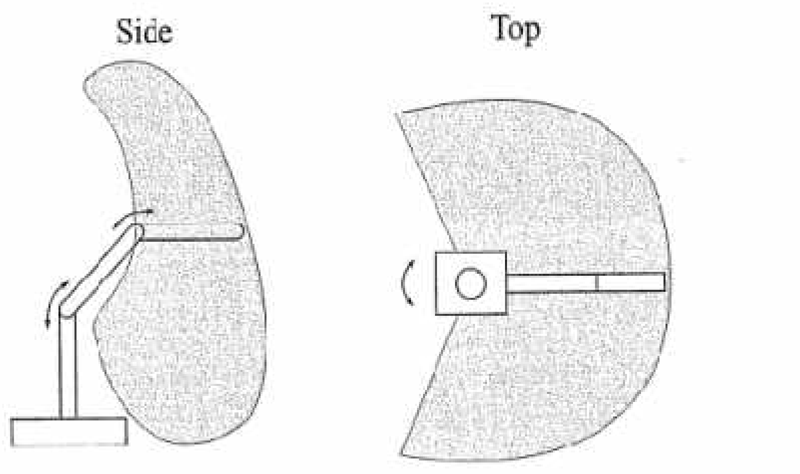


Figure 2.8: Workspace for the articulated manipulator

Chapter Three

Velocity Kinematic - The Jacobian

3.1 Overview

In this chapter we derive the velocity relations, relating the linear and angular velocities of the end effecter to the joint velocities. First we consider the forward kinematics of velocity which is to determine the linear and angular velocities of the end effectors by giving the joint velocities, and then we solve the inverse kinematic of velocity which is to determine the joint velocities that produce the desired end effectors velocities.

To determine the velocities relationships we need to attach coordinate frame rigidly to each link and find the forward kinematic equations that define a function between the space of Cartesian positions and orientations and the space of joint positions as we done in the previous chapter. Then the velocities relationships are determined by the Jacobian of this function.

The Jacobian is a matrix that generalizes the notion of the ordinary derivative of a scalar function. The derivative of kinematic equations is done with aid of skew symmetric matrices.

3.2 SKEW SYMMETRY MATRICES

This section derives the properties of rotation matrices that can be used to compute relative velocity transformations between coordinate frames.

An $n \times n$ matrix S is said to be skew symmetric if and only if

$$S^T + S = 0$$

We denote the set of all 3×3 skew symmetric matrices by $so(3)$.

The skew symmetric matrix contains only three independent entries and every 3×3 skew symmetric matrix has the form

$$S = \begin{bmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix} \tag{3.1}$$

So if $a = (a_x, a_y, a_z)^T$ is a 3-vector then we define the skew symmetric matrix as:

$$S(a) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{3.2}$$

Properties of skew symmetric matrices

1. The operator S in linear, that is

$$S(\alpha a + \beta b) = \alpha S(a) + \beta S(b)$$

For any vectors a and b belonging to R^3 and scalar α and β

2. For any vector a and p belonging to R^3

$$S(a)p = a \times p$$

3. For $R \epsilon so(3)$ and $a \epsilon R^3$

$$RS\ a\ R^T = S\ Ra$$

4. For an $n \times n$ skew symmetric matrix S and any vector $X \epsilon R^n$

$$X^T SX = 0$$

3.3 The derivative of a Rotation Matrix

If a rotation matrix R is a function of single variable θ.Hence,

$R = R\ \theta\ \epsilon so\ 3$ For every θ, the derivative of R is:

$$\frac{d}{d\theta} R = SR\ \theta \tag{3.3}$$

So for the basic rotation matrices:

If $R = R_{x,\theta}$, then $\frac{d}{d\theta} R_{x,\theta} = S\ i\ R_{x,\theta}$, $S\ i\ = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{matrix}$

If $R = R_{y,\theta}$,then $\frac{d}{d\theta} R_{y,\theta} = S\ j\ R_{y,\theta}$, $S\ j\ = \begin{matrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{matrix}$

If $R = R_{z,\theta}$, then $\frac{d}{d\theta} R_{z,\theta} = S\ j\ R_{z,\theta}$, $S\ k\ = \begin{matrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$

3.4 Angular Velocity : The Fixed Axis Case

When a body moves in pure rotation about fixed axis, and if k is a vector in the direction of the axis of rotation, then the angular velocity is given by

$$\omega = \dot{\theta} k \tag{3.4}$$

In which $\dot{\theta}$ is the time derivative of θ. And the linear velocity of any point on the body is given by

$$v = \omega \times r \tag{3.5}$$

In which r is a vector from the origin to the point.

3.5 Angular Velocity : the General Case

Suppose that a Rotational matrix R is time varying, so that
$R = R \;\; t \;\; \epsilon so \;\; 3$ For every t, the derivative of R is

$$\dot{R} \;\; t \;\; = S \;\; \omega \;\; t \;\;\; R(t) \qquad \textit{(3.6)}$$

Where the matrix $S \;\; \omega \;\; t$ is a skew symmetric. The vector $\omega \;\; t$ is the angular velocity of the rotating frame with respect to fixed frame at time t. The previous equation shows the relationship between angular velocity and the derivative of rotation matrix.

We are often interested in finding the resultant angular velocity due to the relative rotation of several coordinate frames. The angular velocities can be added once they expressed relative to the same coordinate frame, suppose that we are given

$$R_n^o = R_1^o R_2^1 \ldots R_n^{n-1}$$

Extending the above reasoning we obtain

$$\dot{R}_n^o = S \ \omega_{0,n}^0 \ R_n^0 \tag{3.7}$$

In which

$$\omega_{0,n}^0 = \omega_{0,1}^0 + R_1^0 \omega_{1,2}^1 + R_2^0 \omega_{2,3}^2 + \cdots + R_{n-1}^0 \omega_{n-1,n}^{n-1}$$

$$= \omega_{0,1}^0 + \omega_{1,2}^0 + \omega_{2,3}^0 + \cdots + \omega_{n-1,n}^0 \tag{3.8}$$

The symbol $\omega_{i,j}^k$ denoted the angular velocityvector corresponding to the derivative of R_j^i, expressed relative to frame k.

3.6 Linear velocity of a Point attached to moving Frame

We now consider the linear velocity of a point that is rigidly attached to a moving frame. Suppose we have two coordinate frames $o_0x_0y_0z_0$ and $o_1x_1y_1z_1$, and that the homogenous transformation relating the two frames is time dependent, so that

$$H_1^0(t) = \begin{matrix} R_1^0(t) & o_1^0(t) \\ 0 & 1 \end{matrix}$$

If a point p is rigidly attached to frame $o_1x_1y_1z_1$, and $o_1x_1y_1z_1$ is rotating relative to the frame $o_0x_0y_0z_0$ then

$$p^0 = Rp^1 + o$$

Differentiating the above expression gives

$$\begin{aligned} \dot{p}^0 &= \dot{R}P^1 + \dot{o} \\ &= s\ w\ RP^1 + o \\ &= \omega \times r + v \qquad (3.9) \end{aligned}$$

Where $r = Rp^1$ is the vector from $\boldsymbol{o_1}$to p expressed in the orientation of the frame $o_0x_0y_0z_0$, and $\boldsymbol{v}$ is the rate at which origin $\boldsymbol{o_1}$ is moving.

3.7 Derivation Of Jacobian

Consider an n- link manipulator with the joint variables $q_1, \dots\ , q_n$. Let

$$T_n^0 = \begin{matrix} R_n^0(q) & o_n^0(q) \\ 0 & 1 \end{matrix}$$

denote the transformation from the end effectors frame to the base frame, where $q = \ q_1, \dots, q_n\ ^T$ is the vector of joint variables. As the robot moves about, both the joint variables q_i and the end effectors position o_n^0 and the orientation R_n^0 will be function of time. The object of this section is to relate the linear and angular velocity of the end effectors to the vector of joint velocities $\dot{q}(t)$. Let

$$S\ \omega_n^0 = \dot{T}_n^0 (R_n^0)^T \tag{3.10}$$

define the angular velocity vector ω_n^0 of the end effectors, and let

$$v_n^0 = \dot{o}_n^0$$

denote the linear velocity of the end effectors. We seek expression of the form

$$v_n^0 = J_v \dot{q}$$

$$\omega_n^0 = J_\omega \dot{q}$$

WhereJ_v and J_ω are $3 \times n$ matrices, we can writing the previous equation together as

$$\xi = J\,\dot{q} \tag{3.11}$$

in which ζ and J are given by

$$\xi = \begin{matrix} v_n^0 \\ \omega_n^0 \end{matrix} \quad \text{and} \quad J = \begin{matrix} J_v \\ J_\omega \end{matrix}$$

The vector ζ is called a body velocity, and the matrix J is called Manipulator Jacobian or Jacobian for short and sometimes it's called the Geometric Jacobian. Note that J is a $6 \times n$ matrix where n is the number of manipulator links.

To derive the Jacobian for any manipulator, we find the terms J_ω, J_v for each link and then we combine the $6 \times n$ matrix. If the joint i is revolute, then the joint variable q_i equal to θ_i, and the axis of rotation is z_{i-1} and If the joint is prismatic, then the joint variable q_i equal to d_i.

For an n-link manipulator, the upper half of the jacobian J_v is given as

$$J_v = [\, J_v \dots \; J_{v_n}]$$

In which the i^{th} column of $J_{v_i}$is

$$J_{v_i} = \begin{matrix} z_{i-1} \times \; o_n - o_{i-1} & \text{for revolute joint } i \\ z_{i-1} & for\ presmatic\ joint\ i \end{matrix} \qquad (3.12)$$

The lower half f the Jacobian is given as

$$J_w = [\, J_{w_1} \cdots \; J_{w_n}]$$

In which the i^{th}column of is J_{w_i}

$$J_{w_i} = \begin{matrix} z_{i-1} & \text{for revolute joint } i \\ 0 & for\ presmatic\ joint\ i \end{matrix} \qquad (3.13)$$

Where $z_{i-1} = R^0_{i-1} k$, and $k = [0,0,1]^T$ (3.14)

The above formulas make the determination of the Jacobian of any manipulator simple since all the quantities needed are available once the forward kinematic worked out. The coordinate for z_i with respect to the base frame are given by the first three elements in the third column of T_i^0. While o_i is given by the first three elements in the fourth column of T_i^0 .

Thus only third and fourth columns of T matrices are needed in order to evaluate the Jacobian of the manipulator.

For the articulated manipulator: we can express the coordinates for each origin of coordinate frame by the vectors

$$O_0 = \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \quad , \quad O_1 = \begin{matrix} 0 \\ 0 \\ d_1 \end{matrix}$$

$$O_2 = \begin{matrix} a_2C_1C_2 \\ a_2S_1C_2 \\ a_2S_2 + d_1 \end{matrix} \quad , \quad O_3 \begin{matrix} a_3C_1C_{23} + a_2C_1C_2 \\ a_3S_1C_{23} + a_2S_1C_2 \\ a_2S_2 + d_1 + a_3S_{23} \end{matrix}$$

Where O_i^j is a vector that represents the coordinates of the origin of coordinate frame i relative to the base frame.

Also we find

$$z_{0=} \begin{matrix} 0 \\ 0 \\ 1 \end{matrix} \quad , \quad z_1 = \begin{matrix} 0 \\ 0 \\ 1 \end{matrix} \quad , \quad z_2 = \begin{matrix} S_1 \\ -C_1 \\ 1 \end{matrix}$$

Now we give the Jacobian components (J_ω, J_v) for each link

For link 1

$$J_{v1} = z_0 \times \; o_3 - o_0 \quad = \begin{matrix} -a_3S_1C_{23} - a_2S_1C_2 \\ a_3C_1C_{23} + a_2C_1C_2 \\ 0 \end{matrix}$$

$$J_{\omega 1} = z_0 = \begin{matrix} 0 \\ 0 \\ 1 \end{matrix}$$

For link 2

$$J_{v2} = z_1 \times \ o_3 - o_1 \ = \begin{matrix} -a_3S_1C_{23} - a_2S_1C_2 \\ a_3C_1C_{23} + a_2C_1C_2 \\ 0 \end{matrix}$$

$$J_{\omega 2} = z_1 = \begin{matrix} 0 \\ 0 \\ 1 \end{matrix}$$

For link 3

$$J_{v3} = z_2 \times \ o_3 - o_2 \ = \begin{matrix} -C_1a_3S_{23} - a_3S_1C_{23} \\ a_3C_1C_{23} - S_1a_3S_{23} \\ S_1a_3S_1C_{23} + C_1a_3C_1C_{23} \end{matrix}$$

$$J_{\omega 3} = z_2 = \begin{matrix} S_1 \\ -C_1 \\ 1 \end{matrix}$$

The Jacobian for the first three links of the articulated manipulator is:

$$\boldsymbol{J} = \begin{matrix} \boldsymbol{J_{v1}} & \boldsymbol{J_{v2}} & \boldsymbol{J_{v3}} \\ \boldsymbol{J_{\omega 1}} & \boldsymbol{J_{\omega 2}} & \boldsymbol{J_{\omega 3}} \end{matrix}$$

$$= \begin{matrix} -a_3S_1C_{23} - a_2S_1C_2 & -a_3S_1C_{23} - a_2S_1C_2 & -C_1a_3S_{23} - a_3S_1C_{23} \\ a_3C_1C_{23} + a_2C_1C_2 & a_3C_1C_{23} + a_2C_1C_2 & a_3C_1C_{23} - S_1a_3S_{23} \\ 0 & 0 & S_1a_3S_1C_{23} + C_1a_3C_1C_{23} \\ 0 & 0 & S_1 \\ 0 & 0 & -C_1 \\ 1 & 1 & 1 \end{matrix}$$

(3.15)

3.8 The Analytical Jacobian

The Analytical Jacobian denoted $J_a\ q$ is based on minimal representation for the orientation of the end effectors frame and it different from the Jacobian that we derived previously (its called Geometric Jacobian).

Geometric Jacobian $\zeta = \begin{matrix} v(q) \\ \omega(q) \end{matrix} = \begin{matrix} \dot{d}(q) \\ \omega(q) \end{matrix} = J\ q\ \dot{q}$

Analytical Jacobian $\dot{X} = \begin{matrix} \dot{d}(q) \\ \dot{\alpha}(q) \end{matrix} = J_a\ q\ \dot{q}$ (3.16)

By using Euler's angels for the parameterization of orientation, the analytical Jacobian relates the joint velocities to the time derivative of the pose parameter

$$X = \begin{matrix} d(q) \\ \alpha(q) \end{matrix} \quad , \quad \dot{X} = \begin{matrix} \dot{d} \\ \dot{\alpha} \end{matrix} = J_a\ q\ \dot{q} \tag{3.17}$$

in which $d(q)$ is the usual vector from the origin of the base frame to the origin of the end-effecter frame and α denotes the parameterization of the rotation matrix that specifies the orientation of the end-effecter frame relative to the base frame. For Euler angles parameterization, the analytical Jacobian is given by

$$J_a(q) = \begin{bmatrix} I & 0 \\ 0 & B^{-1}(\alpha) \end{bmatrix} J(q) \tag{3.18}$$

In which

$$B(\alpha) = \begin{bmatrix} c_\psi s_\theta & -s_\psi & 0 \\ s_\psi s_\theta & c_\psi & 0 \\ c_\theta & 0 & 1 \end{bmatrix}$$

3.9 Singularities

Singularities are configurations in which the manipulator loses one or more degree of freedom of motion. This idea can be made precise in terms of the rank of a Jacobian matrix relating the rates of change of input (joint position) and output (end-effectors position) variables.

The rank of a matrix is not necessary constant. Indeed, the rank of the manipulator Jacobian matrix will depend on the configuration q. Configurations for which rank $j(q)$ is less than its maximum value are called singularities or singular configuration.

In general: A configuration q is singular I and only if

$$\det J(q) = 0$$

It's difficult to solve this nonlinear equation, there for we use the method of decupling singularities, which is applicable whenever, for example, the manipulator is equipped with spherical manipulator, so we decouple the determination of singular configuration into two simpler problems .The first is to determine the singularities results from the motion of the arm (arm singularities) while the second is to determine the wrist singularities resulting from the motion of the wrist.

For spherical wrist manipulators, the Jacobian matrix has the block triangle form

$$j = \begin{matrix} j_{11} & 0 \\ j_{21} & j_{22} \end{matrix} \qquad (3.19)$$

With determinates

$$\det j = \det j_{11} \det j_{22}$$

The set of singular configurations of the manipulator is the union of the set of arm configurations satisfying $\det j_{11} = 0$ (arm configuration) and the set of wrist configurations satisfying $det j_{22} = 0 \quad wrist\ configuration$.

For the articulated manipulator with coordinate frames attached as shown in figure 3.1

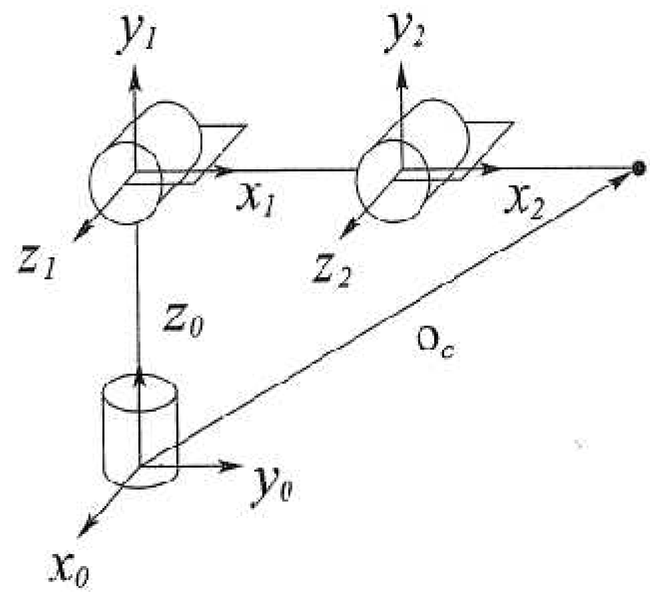


Figure 3.1: The articulated manipulator showing O_c

For this manipulator we see that

$$j_{11} = \begin{matrix} -a_2s_1c_2 - a_3s_1c_{23} & -a_2s_2c_1 - a_3s_{23}c_1 & -a_3c_1s_{23} \\ a_2c_1c_2 + a_3c_1c_{23} & -a_2s_1s_2 - a_3s_1s_{23} & -a_3s_1s_{23} \\ 0 & a_2c_2 + a_3a_3c_{23} & a_3c_{23} \end{matrix} \tag{3.20}$$

And the determinate of j_{11} is

$$det\, j_{11} = a_2a_3s_3(a_2c_2 + a_3c_{23}) \tag{3.21}$$

$$The\ articulated\ manipulator\ is\ in\ singular\ configuration\ when$$

$$s_3 = 0 \quad that\ is\ \ \theta_3 = 0\ \ or\ \ \pi$$

This situation is shown in figure 3.2 and it arises when the elbow is fully extended or fully retracted.

And whenever

$$a_2c_2 + a_3c_{23} = 0$$

This configuration occurs when the wrist center intersects the axis of base rotation z_0 and in this case there is infinity many singular configurations and infinity many solutions to the inverse position kinematics when the wrist center is along this axis. This configuration can be avoided by an offset in the elbow or shoulder.

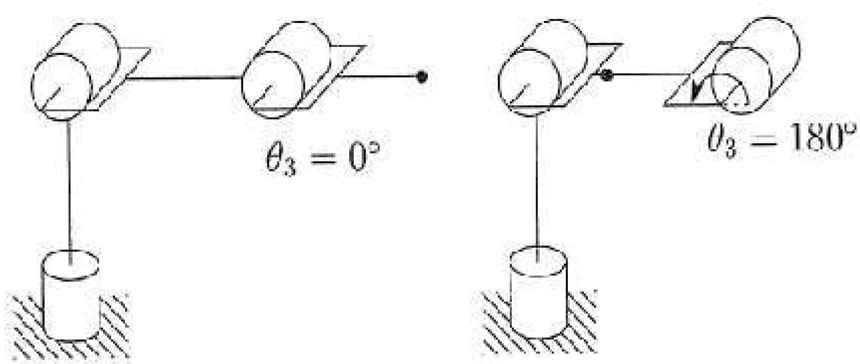


Figure 3.2: Elbow singularities of the articulated manipulator

3.10 Inverse Velocity And Acceleration

The inverse velocity problem is the problem of finding the joint velocities $\dot{q}$ that produce the desired end effector velocity $\dot{X}$ or acceleration $\ddot{X}$.

For manipulators that have six joints and the Jacobian matrix is square and nonsingular (det $J_a \; q \; \neq 0$), this problem can be solved by simply inverting the Jacobian matrix to give

$$\dot{q} = J^{-1}\zeta \tag{3.22}$$

For manipulators with $n > 6$ we can solve for $\dot{q}$ (joint velocities) using the right pseudoinverse of j. To construct this pseudoinverse, we use the fact that when $j \in R^{m \times n}$, if m < n and $rank\ j = m$, then $(jj^{+})^{-1}$ exist. in this case $(jj^{T}) \in R^{m \times n}$, and has a rank m . using this we can find that:

$$j^{+} = J^{T}(JJ^{T})^{-1} \tag{3.23}$$

$which\ is\ the\ right$ Pseudoinverse of the j , and if we multiply it by j this will give the identity matrix I, that to say $JJ^{+} = I$ but $J^{+}J \neq I$ since the matrix multiplication is not cumulative.

By using the right pseudoinverse we can find a solution for $\dot{q}\ (joint\ velocity)$ for nonsingular configurations as

$$\dot{q} = j^{+}\xi \tag{3.24}$$

We can apply a similar approach when the analytical Jacobian is used in place of manipulator Jacobian. The joint velocities and the end-effectors velocities are related by the analytical Jacobian as

$$\dot{X} = J_a(q)\dot{q} \tag{3.25}$$

Thus, the inverse velocity problem becomes one of solving the linear system given by the above equation.

Differentiating equation (3.23) yield an expression for the acceleration:

$$\ddot{X} = J_a\ \ q\ \ \ddot{q} + (\frac{d}{dt}J_a\ \ q\ \ )\dot{q} \tag{3.26}$$

Thus, given vector $\ddot{X}$ of end-effecter, the instantaneous joint acceleration vector is given as a solution of

$$J_a\ \ q\ \ \ddot{q} = \ddot{X} - (\frac{d}{dt}J_a\ \ q\ \ )\dot{q} \tag{3.27}$$

Chapter Four

Trajectory Planning

4.1 Introduction

Trajectory planning relates to the way a robot is moved from one location to another in a controlled manner. So that in this chapter we will plan a trajectory; a trajectory refers to the time history of position, velocity and acceleration for each joint in the manipulator. Trajectory planning requires the use of kinematic and dynamic equation of the manipulator.

When we dealing with trajectory there are many constants we expect to see in solving this problem, these constrain could be:

1. Spatial constrain, if we have on obstacle in the environment that we don’t want to collide with. We will neglect this constrain in our project and assume that there is no obstacle in the workplace of the robot arm.
2. Time constrain, if the motion ha to be done in particular time.
3. Smoothness, we want the manipulator to have a smooth motion because that uses less energy and easy to control.

The trajectory planning can be done in two main spaces, joint space and Cartesian space .In joint pace it easy to go through point, there is no problems with singularities and it requires less calculation. In the other hand, the actual end-effectors path of this approach can't be predicted and can't follow straight lines.

The trajectory planning in Cartesian space may involve problems difficult to solve. However, Cartesian space is more computationally expensive to execute since at run time, inverse kinematics must be solved at the path update rate. Other major problems that we may face in Cartesian space is singularity ;if there are some points on the path that the manipulator should follow are in singular configuration, but in this space we can specify the shape of the path between path points.

In Our project we will use the joint space trajectory planning for the articulated manipulator, because we want to move the end-effectors from initial position to final position regardless of the path to follow.

4.2 Joint space Trajectory

The joint space is a method of path generation in which the path shapes (in space and in time) are described in terms of function of joint angles.

Each path joint is usually specified in terms of a desired position and orientation of the tool frame, relative to the base frame, each of these points in converted into a set of desired joint angels by application of inverse kinematics .Then a smooth function is found for each of the n joints to describe the motion between the initial and final joint.

Through the remaining of this chapter we interest in establishing formulas for the angels of each DOF as a function of time in the case of the initial and final points on the path and traveling time are specified (point-to-point)

4.3 Point to point motion

In the trajectory planning, a complete description of all location of every point on the robot is referred to as a configuration. For our purpose, the vector of joint variables q provide a convenient representation of a configuration

The task of point to point motion is to plan a trajectory from an initial configuration $q\ t_o$ to a final configuration$q\ t_f$. In some cases, there may be constrains on the trajectory (for example, if the robot must start and end with zero velocity). Nevertheless, it's easy to realize that there are infinitely many trajectories that will satisfy a finite number of constrains on the end points.

It's a common practice therefore to choose trajectories from a finitely parameterizable family, for example, polynomials of degree, where $n$depends on the number of constrains to be satisfied. This is the approach that we will take in our project.

We will consider tow smooth functions for point to point motion, cubic polynomial and linear segments with parabolic blends, then these functions are substituted on the dynamic equation of the robot to see which function produce less torque and thus less power consumption.

4.3.1 Cubic Polynomials

Consider the problem of moving the end-effectors from its initial position to a final position in a particular time. The set of goal joint angles can be calculated using the inverse kinematic for particular values of end effectors position. The initial position of the manipulator is also known on the form of a set of joint angles. To make a smooth motion between the initial and final position of the manipulator, we first have to generate a polynomial joint trajectory between the two configurations, and specify the start and end velocities of the trajectory. This gives four constraints that the trajectory must satisfy, two constrains comes from the selection of initial and final values:

$$q(t_0) = q_0$$

$$q(t_f) = q_f$$

For the velocity constraints, if we want to have continuous velocity the final and initial velocities must be zero:

$$\dot{q}(t_0) = 0$$

$$\dot{q}(t_f) = 0$$

Thus we require polynomial with four independent coefficients to satisfy these constraints, so we can consider a cubic polynomial. The cubic polynomial will have this form

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{4.1}$$

For the manipulator we have the initial position q_0 is known in the form of a set of joint angle, and the final position q_f can be determined using inverse kinematic. The velocity and acceleration is given as

$$\dot{q}\ t\ = a_1 + 2a_2 t + 3a_3 t^2 \tag{4.2}$$

$$\ddot{q}\ t\ = 2a_2 + 6a_3 t \tag{4.3}$$

Combine equations 4.2 and 4.3 with four constraints yields four equation in four unknowns:

$$q_0 = a_0$$

$$q_f = a_0 + a_1 t_f + a_2 t_f{}^2 + a_3 t_f{}^3$$

$$\dot{q}_0 = a_1$$

$$\dot{q}_f = a_1 + a_2 t_f + 3a_3 t_f{}^2$$

Solving these equations for a_i we obtain

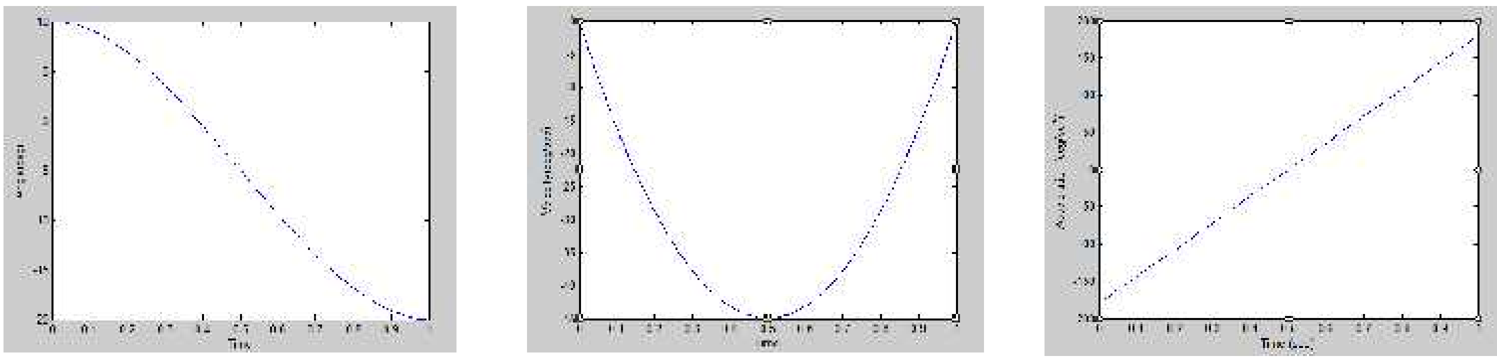
$$a_0 = q_0$$

$$a_1 = 0$$

$$a_2 = 3(q_f - q_0)$$

$$a_3 = 2(q_f - q_0)$$

Using these parameters we can calculate the cubic polynomial that connects any initial joint angle position with any desired final position. This solution is for the case when the joints starts and finishes at zero velocity. Figure 4.1 shows the cubic polynomial trajectory



(a) (b) (c)

Figure 4.1: a) Cubic Polynomial Trajectory b) Velocity profile for Cubic Polynomial Trajectory c) Acceleration profile for cubic polynomial Trajectory

4.3.2 Linear segment with parabolic blends (LSPB)

Another way to generate joint space trajectories is by using so-called linear segments with parabolic blends (LSPB). This type of trajectory has trapezoidal velocity profile, and is appropriate when a constant velocity is desired along portion of path.

This is a linear function but we add a parabolic blend region at the beginning and end of the path. These blend regions create a smooth path with continuous position and velocity. Thus, during the blend portion of the trajectory, constant acceleration is used to change velocity smoothly. Figure 4.2 shows a simple path constructed in this way.

In order to construct this single segment we will assume that the parabolic blend both have the same duration, and therefore, they have the same constant acceleration.

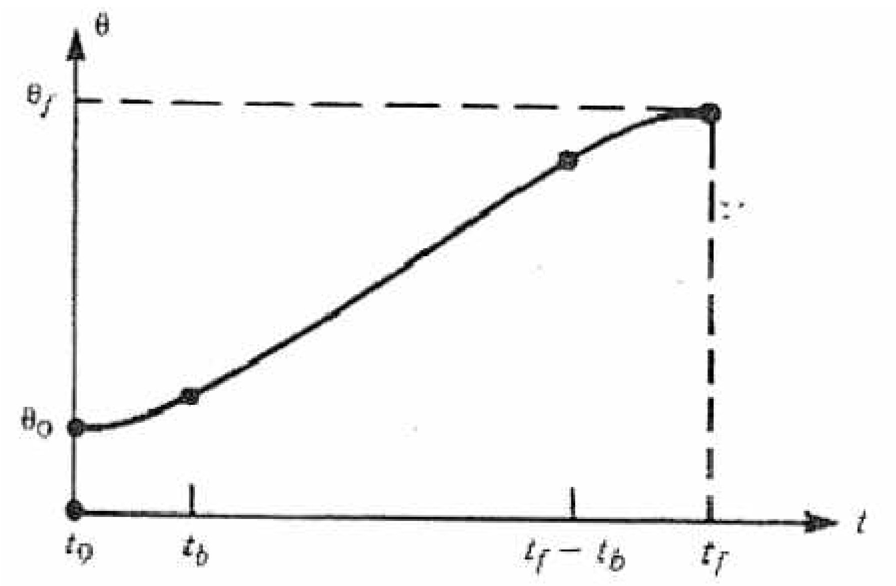


Figure 4.2: linear segments with parabolic Blends

For parabolic blends near the path points with the same duration t_b (blend time) the whole trajectory is symmetric about the halfway point in time t_b and about the halfway point in position q_m.

The velocity at the end of first blend or at the beginning of second blend must equal to linear segment, thus we have:

$$\mathrm{q} = \frac{\mathrm{q_m - q_b}}{\mathrm{t_m - t_b}} \tag{4.4}$$

Where q_b is the value of joint variable at the end of blend segment at time $\mathrm{t_b}$, $\ddot{q}$ is the acceleration during the blend segment, and the joint variable q_b is given by

$$q_b = q_0 + \frac{1}{2}\ddot{q}{t_b}^2 \tag{4.5}$$

Combining equations 3 and 4 and $t = 2t_h$

$$\ddot{q}{t_b}^2 - tt_b + \ q_f - q_0 \ \ = 0 \tag{4.6}$$

Where t is the desired duration of the motion**.** Usually equation 3.10 is solved for a corresponding t_b, and the acceleration $\ddot{q}$ is chosen. Solving equation (4.5) for t_b

$$t_b = \frac{t}{2} - \frac{\overline{\ddot{q}^2t^2 - 4\ddot{q}(q_f - q_0)}}{2\ddot{q}} \tag{4.7}$$

The constraint on the choice of acceleration used in blend segment is

$$\ddot{q} \geq \frac{4 \ \ q_f - q_0}{t^2}$$

The complete LSPB trajectory is given by

$$q(t) = \begin{cases} q_0 + \dfrac{\ddot{q}t^2}{2}, & 0 \le t \le t_b \\ q_0 + \ddot{q}t_b\left(t - \dfrac{t_b}{2}\right), & t_b < t \le t_f - t_b \\ q_f - \dfrac{1}{2}\ddot{q}\,(t_f - t)^2, & t_f - t_b < t \le t_f \end{cases} \qquad (4.8)$$

So the joint velocity and acceleration is given by

$$\dot{q}(t) = \begin{cases} \ddot{q}t, & 0 \le t \le t_b \\ \ddot{q}t_b t, & t_b < t \le t_f - t_b \\ \ddot{q}\,(t_f - t), & t_f - t_b < t \le t_f \end{cases} \qquad (4.9)$$

$$\ddot{q}(t) = \begin{cases} \ddot{q}, & 0 \le t \le t_b \\ \ddot{q}t_b, & t_b < t \le t_f - t_b \\ \ddot{q}, & t_f - t_b < t \le t_f \end{cases} \qquad (4.10)$$

Figure 4.3 shows the LSPB Trajectory, velocity profile and acceleration profile.

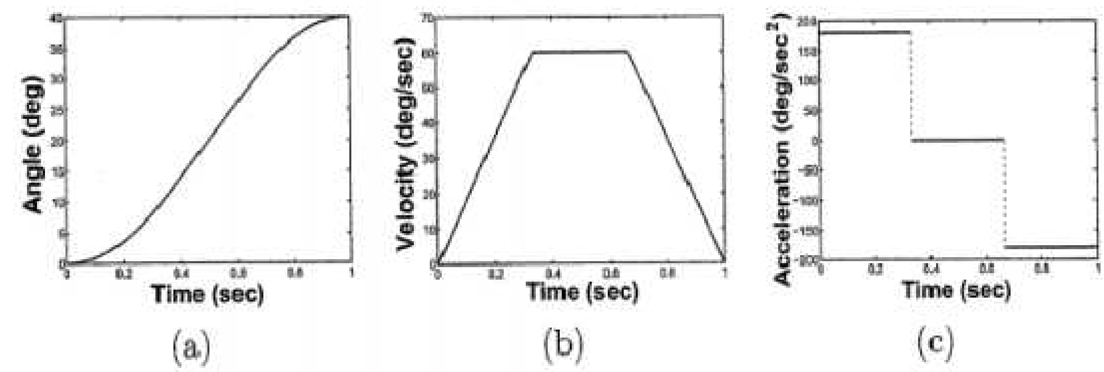


Figure 4.3: a) LSPB trajectory b) velocity profile for LSPB trajectory

c) Acceleration profile for LSPB trajectory.

In Our project we use Cubic Polynomial method since its easier to program and need less user specification than LSPB method

Chapter Five

Control Design

5.1 Introduction

5.2 System Architecture

5.2.1 Physical System Description

5.2.2 Functional description

5.2.3 Hydraulic Description

5.2.4 Electrical Description

5.3 Closed Loop Control

5.3.1 Feedback Sensors

5.3.2 Controller and software

5.1 Introduction

Robot control concerns with studying how to make a robot manipulator execute the desired task automatically. Typically, a robot takes the form of an equation or an algorithm which is realized via specialized computer program (TwidoSuite in our case). Robots controller's forms the so-called robot control system which is physically constitute of a computer, programmable logic controller, actuator, the robot itself and some extra electronics as illustrated in Figure 5.1.In this chapter we will describe how the overall system works.

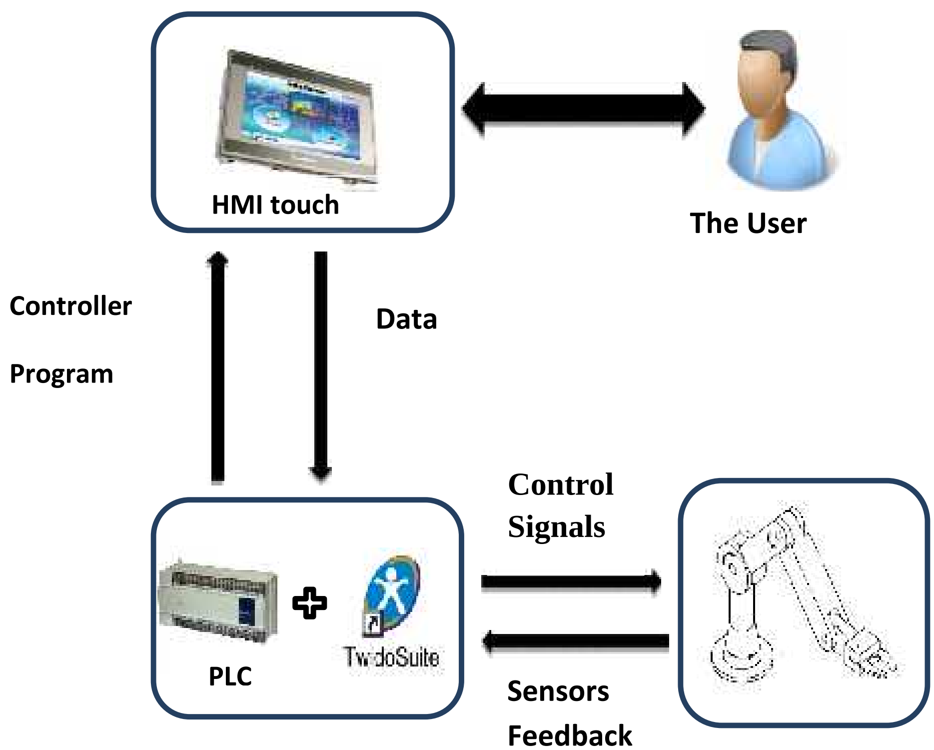


Figure 5.1: the robot control system

5.2 System Architecture

5.2.1 Physical System Description

The purpose of the articulated hydraulic manipulator to pick an object and move it to another location as we mentioned before the hydraulic manipulator consist from a three link arm and the end-effectors which is a gripper type. Each link is actuated with a hydraulic rotary actuator with limited rotation. Figure 5.1 shows this type of rotary actuator

For the first link, the rotary actuator provides a rotation of angle from -67° to 113° relative to the base frame. So this link will rotate a full 180° rotation.

The second link rotation affects the overall angle of the arm, and it permitted to rotate through an angle from -45° to 45° relative to the shoulder joint producing an angle of rotation of 90°.

The third link is rotate by angle of 90°, from 0° to 90° relative to the elbow frame.

Figure 5.2: Hydraulic rotary actuator for the joints

5.2.2 Functional description

The user is asked to enter the coordinates of the initial position and the final position of the end effectors through the touch screen, the manipulator go the initial position, pick an object and move to the final location. After the final position is reached the object is released, if there is no new coordinates are entered the manipulator may be programmed to move to an assigning suit which is the position where is no motion is executed.

5.2.3 Hydraulic Description

The system is operated by a fixed displacement hydraulic pump driven by an electric motor. The hydraulic pump provide a regulated flow to the main directional control valve which control the overall flow of the system by permitting a flow to the other parts of the hydraulic circuit or diverting the flow back to the reservoir. A proportional flow control valve is used to control the flow rate of the hydraulic actuators using a single solenoid.

each rotary actuator is controlled by a directional control valve that allow a clockwise and counterclockwise rotation ,these directional valves are actuated through tow solenoids at the ends of each valve, one solenoid for clockwise rotation and the other for counterclockwise rotation .

The system is protected against overload pressure by using a pressure relief valve and it's placed out of the hydraulic pump. Figure 5.2 shows the hydraulic circuit of the system.

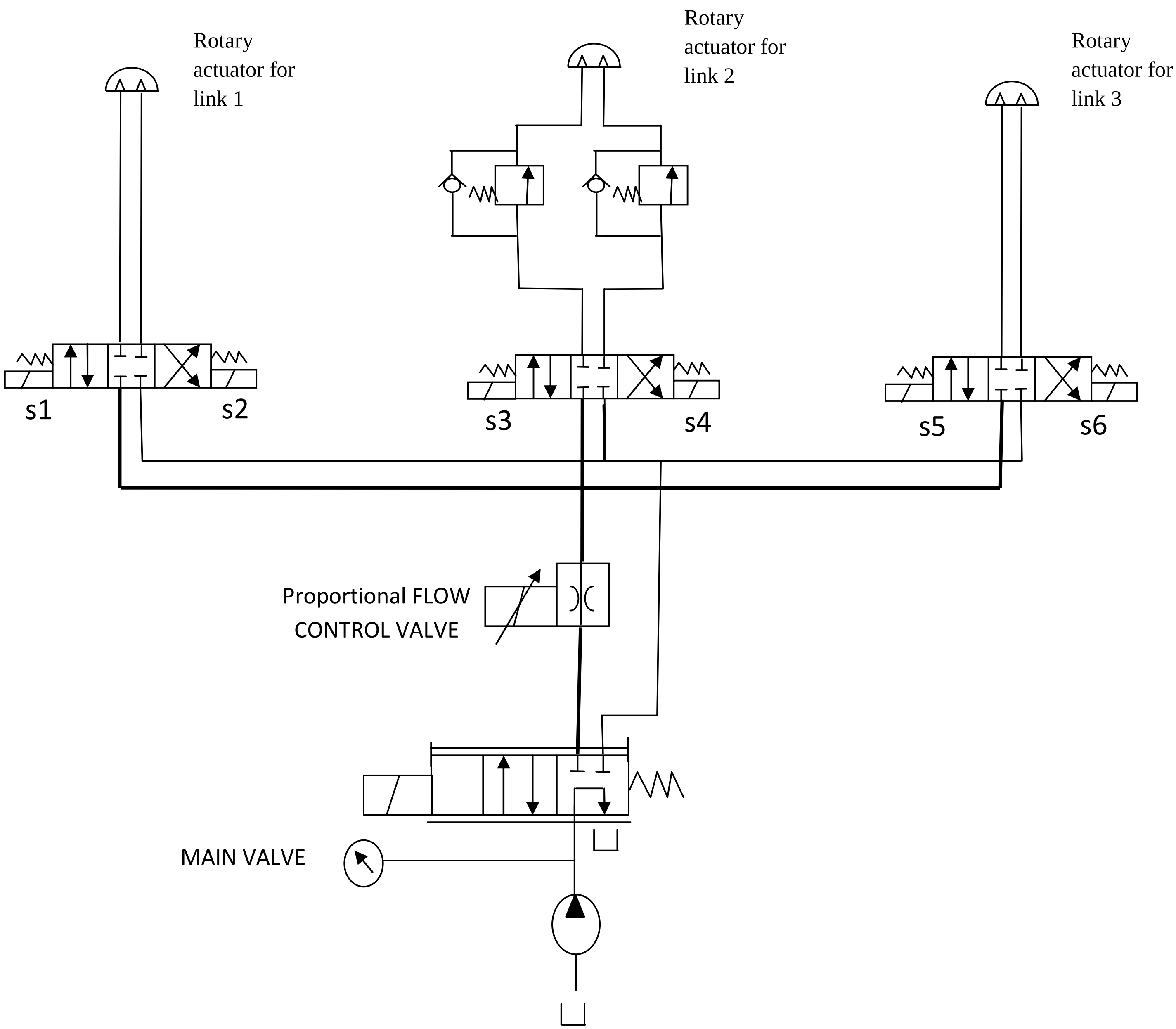


Figure 5.3: Hydraulic circuit for the system

5.2.4 Electrical Description

The machine is powered by 220 volt AC power .Incoming AC power is routed in parallel to the hydraulic pump electric motor, main valve and to the DC power supply .the power supply ensures that the system voltage is regulated to 24 volts DC. The system voltage is routed through an emergency power shutoff switch to DC power supply, toggle switches and system electronic unit. A simplifies electrical circuit for 220 volt AC power supply is shown in Figure 5.3

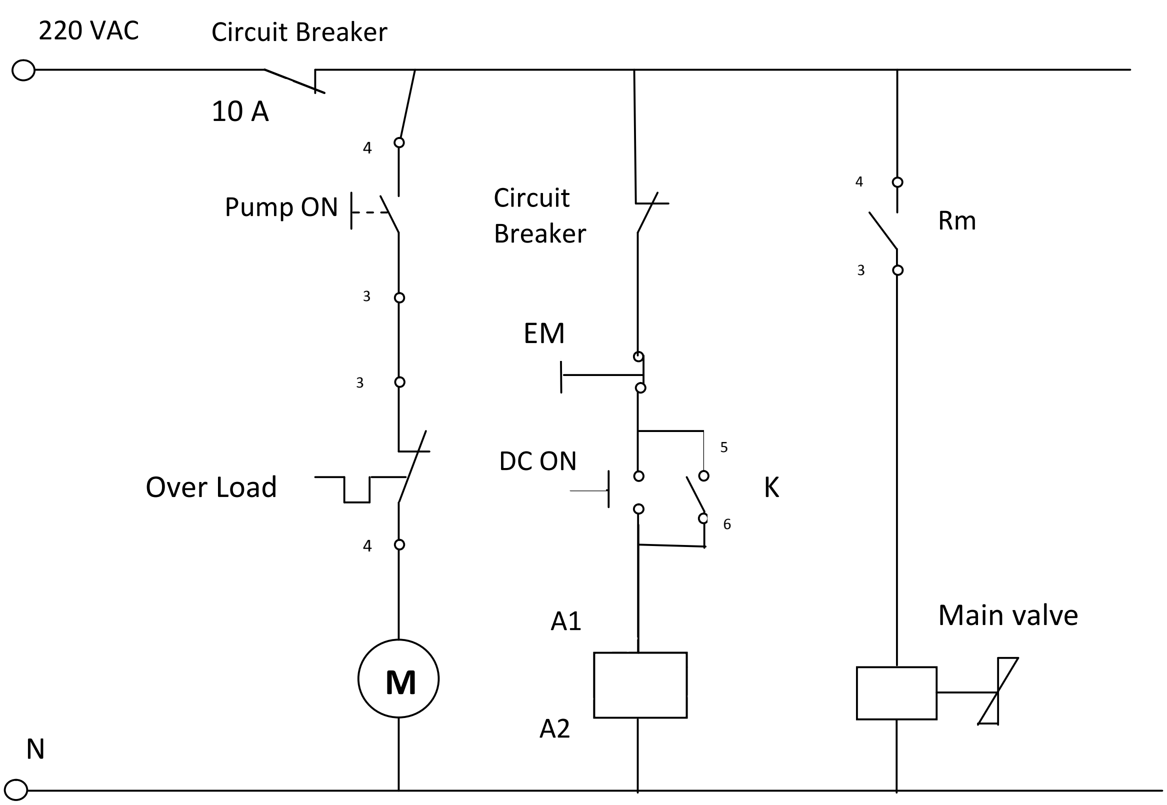


Figure 5.4: Simplifies electric circuit for 220 volt power supply

5.3 Closed Loop Control

The term closed-loop control refers to the robot system managing the flow demand by routing the valve system to achieve the desired movement or desired position with smooth motion, and using sensors that give a feedback read for the final position for each link angle. We used the PLC for this operation, this logic controller have the following specification:

1. Four analog inputs to read the position for each link through the potentiometer.
2. Seven digital output; each output is connected to each solenoid to give a signal for desired motion.
3. One analog output to control the flow through proportional valve.

To achieve this specification we choose Schneider PLC with HMI Touch screen to achieve this job. Figure 5.4 shows this type of plc and touch screen.

Figure5.5: Plc and HMI touch screen

As we mentioned before the amount of flow can be controlled by proportional valve through an analog signal (0-10 v), this signal can be changed through an analog output port of the controller.

5.3.1 Feedback Sensors

In a closed loop control system, four sensors monitor the system output (joint Angeles) and feed the data to a controller which adjusts the control (joint angle) as necessary to maintain the desired system output (match the desired position which is X, y and z coordinates of the end effectors). This robot uses potentiometers to determines where it and then controls their joints to match the desired position. The output of the potentiometer is an analog voltage that is proportional to the angle of rotation for each joint. These analog signals are connecting to the analog input of the PLC.

Potentiometer calibration

The calibration of potentiometer aims to find and represent the angles of the joints according to the output voltage of each potentiometer attached to its joint, this voltages inter to the module as analog input, the by some equations we determine the equivalent angle.

For the first link (A), the angle θ_1 is between [-30 _ 160] and the voltage v1 change between [6.63 _ 1.62], so the relationship between them described as shown in figure 5.6.

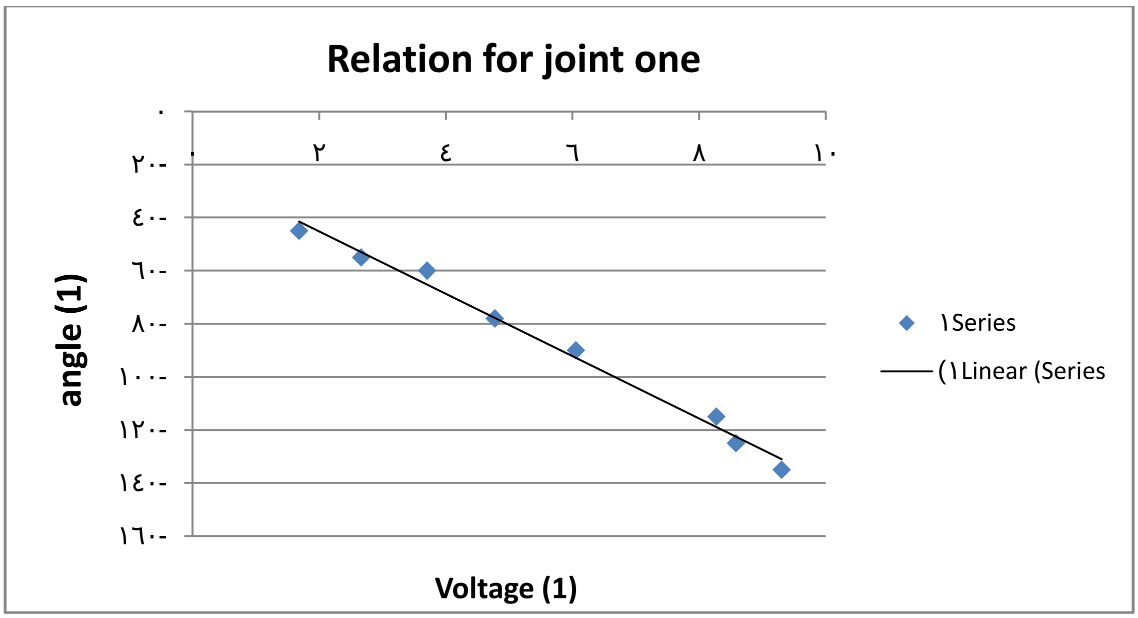


Figure 5.6 : relation between voltage and angle for the first joint

So the formula of represent angle to according voltage is

$$\theta_1 = -36.28 * v_1 + 214$$

For the second link (B), the angle θ_2 is between [45 _ - 6] and the voltage v2 change between [7.5 _ 2.78] , so the relationship between them described as shown in figure 5.7

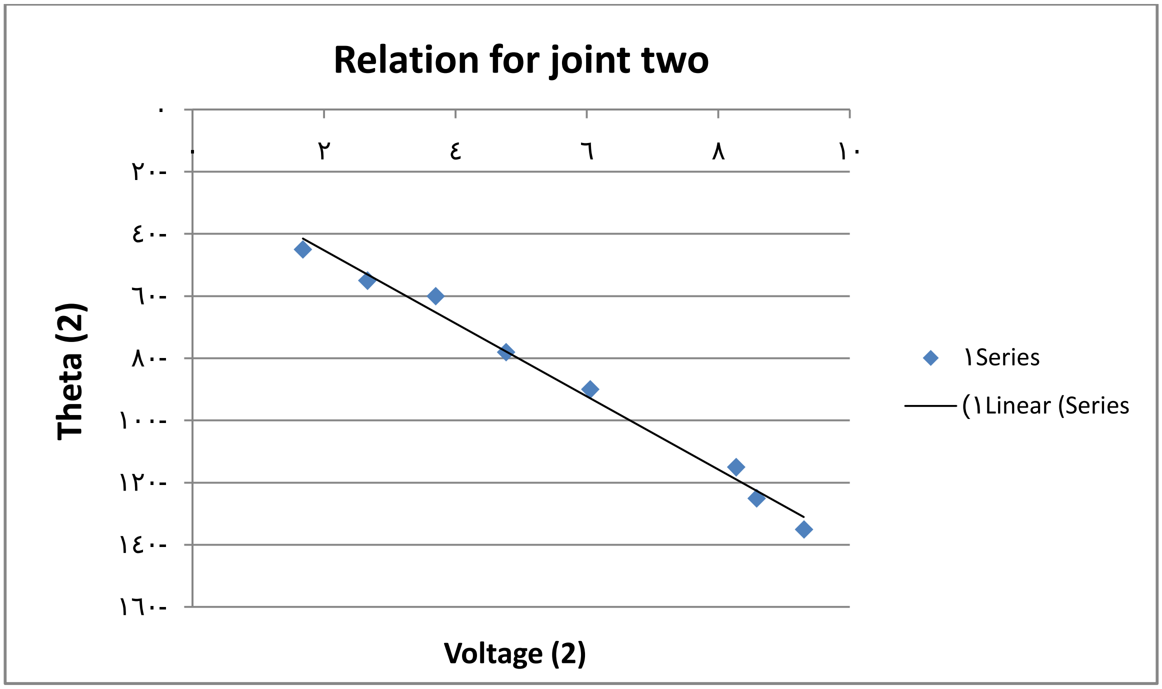


Figure 5.7: relation between voltage and angle for the first joint

So the formula of represent angle to according voltage is

$$\theta_2 = 11.94 * v_2 - 42.37$$

For the third link (C), the angle θ_3 is between [-90 _ - 135] and the voltage v3 change between [6.05 _ 9.37], so the relationship between them described as shown in figure 5.8.

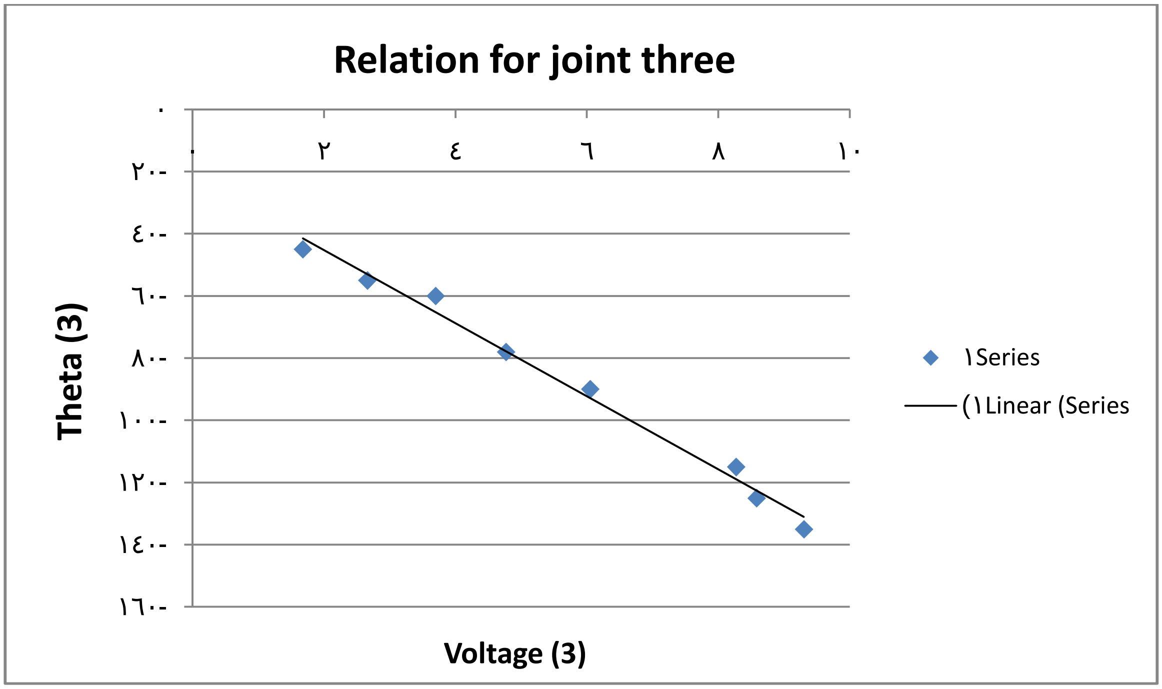


Figure 5.8 : relation between voltage and angle for the first joint

So the formula of represent angle to according voltage is

$$\theta_3 = -11.71 * v_3 - 21.9$$

5.3.2 Controllers and software

We are now interested in solving motion control problem .In motion control problem, the manipulator moves to a position to pick up an object, transport that project to another location, and deposit it. We treat this problem in the joint space.

Joint space control

The main goal of the joint space control is to design a feedback controller such that the joint coordinates track the desired motion as closely as possible .the control of robot manipulators is naturally achieved in the joint space. Since the control inputs are the joint torques

Figure 5.6 shows the basic outline of the joint space control methods. Firstly the desired motion, which is described in terms of end-effectors coordinates, is converted to a corresponding joint trajectory using the invest kinematics of the manipulator, Then the feedback controller determines the joint torque necessary to move the manipulator along the desired trajectory specified in joint coordinates starting from measurements of the current joint states.

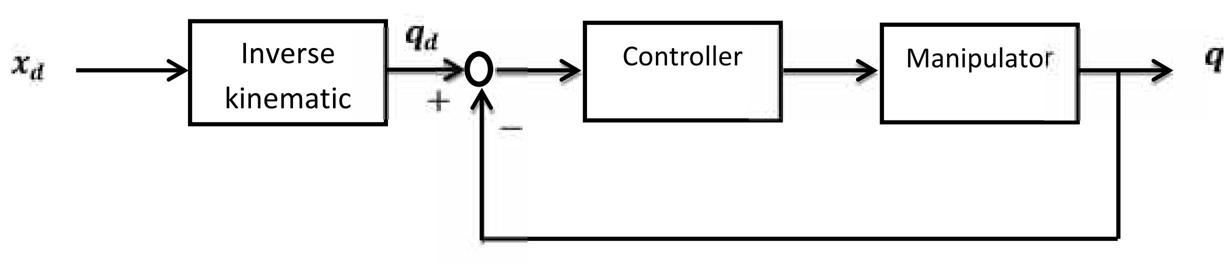


Figure 5.6 Generic concept of joint space control

Independent joint control

We adapt independent joint control to control the robot manipulator. By independent-joint control (i.e.., decentralized control) we mean that the control inputs of each joint only depends on the measurement of the corresponding joint displacement and velocity. Due to its simple structure, this kind of control schemes offers many advantages. For example by using independent- joint control, communication among different joint is saved. Moreover, since the computational load of controller may be reduced, only low-cost hardware is required in actual implementations. Finally, independent-joint control has the feature of scalability, since the controller on all joints has the same formulation.

The simplest independent-joint control strategy is to control each joint axis as single-input single-output (SISO) system; this type of control appears in figure 5.7. This figure is common for all links.

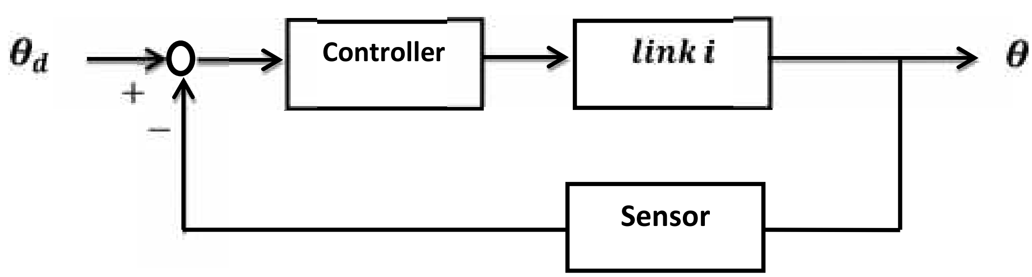


Figure 5.7: Concept of joint space control

Because of electrically actuated solenoid valves which using for control the actuator is (on /off control). Then we can’t control the velocity for each joint independently by this method, as mentioned before, there is one proportional valve to control the over flow and thus the velocity for all links, this leads to seek for a method to control all links velocity with common controller, this controller formulation depends on the hardware in the loop simulation .

In order to find the controller, and since we don’t knew the transfer function for the proportional valve, will use the experiments depends on Hardwar-in-Loop simulation as mentioned before to find the formulation of the controller.

We will use a PI controller to control the position of each joint and using experiments the best gain values for the controller is:

KP= 50

KI=1200

$$G(s)_{Controller} = 50 + \frac{1200}{s} = \frac{50s + 1200}{s}$$

Chapter six

Hardware and Software Description

6.1 Introduction

6.2 Hardware and Software Components

6.3 Sequence function chart –SFC (State Graph)

6.4 PID functions on Plc.

6.4.1 Introduction

6.4.2 The PID Controller Model

6.4.3 Operating Principles

6.4.4 Principal of the Regulation Loop

6.4.5 Role and Influence of PID Parameters

6.5 Touch Screen and Vijeo Designer

6.1 Introductions

In this chapter we will describe the hardware and software components that used in our project, and then we introduce Sequence function chart - SFC (or State graph) and the Concept of PID function, following by an introduction to HMI Software.

6.2Hardware and Software Components:

In our project we use these Types of Hardware:

1) TWDLCDE40DRF Twido Controller: which is a compact base controller, 24v DC, 40 points, 24v DC inputs, 12-2A relay output, 2-1A transistor output, Timer and Calendar and Ethernet 100Base Tx, Removable Battery and Non-removable terminal blocks.

2) TM2AMM6HT Analog Expansion Module: it's an Expansion Module with 4Analog inputs and 2 analog outputs (0 – 10v, 4– 20 mA), 12 bit resolution, removable screw terminals.

3) XBT OT 2210 Touch Screen: its 256 color and has a Supply Voltage Range 19.2V DC to 28.8V DC.

In this project we use TwidoSuit V2.31.04 for programming the PlC using sequence function chart (SFC) method.

And we used Vijeo Designer Opti for Programming the Xbt OT 2210 Touch screen.

6.3 Sequence function chart –SFC (State Graph)

Sequential Function Charts (SFCs) are a graphical technique for writing concurrent control programs or sequential control algorithms, and they are also known as Grafcet or IEC 848.

A sequence function chart is a pictorial representation of the system's individual operations, which when combined show the complete sequence of events. Once this diagram has been produced, then from it, the corresponding ladder diagram can be more easily designed. Figure 6.1 shows an example of the sequence function chart.

Description of sequence function chart

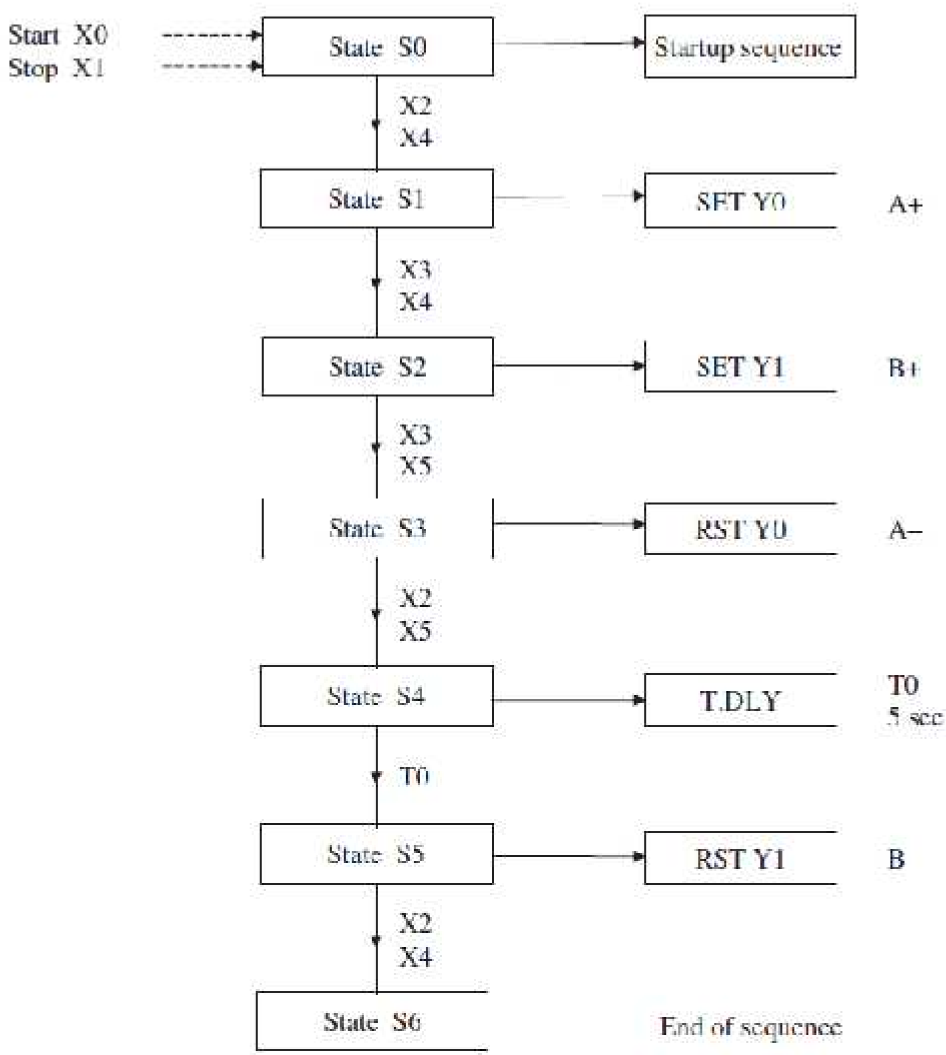


Figure 6.1 sequence function chart example

The above figure is an example of sequence function chart and we will describe it in the following points:

1) The sequence function chart consists, basically, of a number of separate sequentially connected states, which are the individual constituents of the complete machine cycle that controls the system. An analogy is that each state is like a piece of a jigsaw puzzle; on its own it does not show very much, but when all the pieces are correctly assembled, then the complete picture is revealed.

   Each state has the following:
   (a) An input condition.
   (b) An output condition.
   (c) A transfer condition.

   When the input condition into a state is correct, then that state will produce an output condition. That is, an output device or devices will be:
   (a) Turned ON and remain ON.
   (b) Turned OFF and remain OFF.

2) When the output or outputs are turned ON/OFF, then the system's input conditions will change to produce a transfer condition.

3) The transfer condition is now connected to the input condition of the next sequential state.

4) If the new input condition is correct, then the sequence moves to the next state.

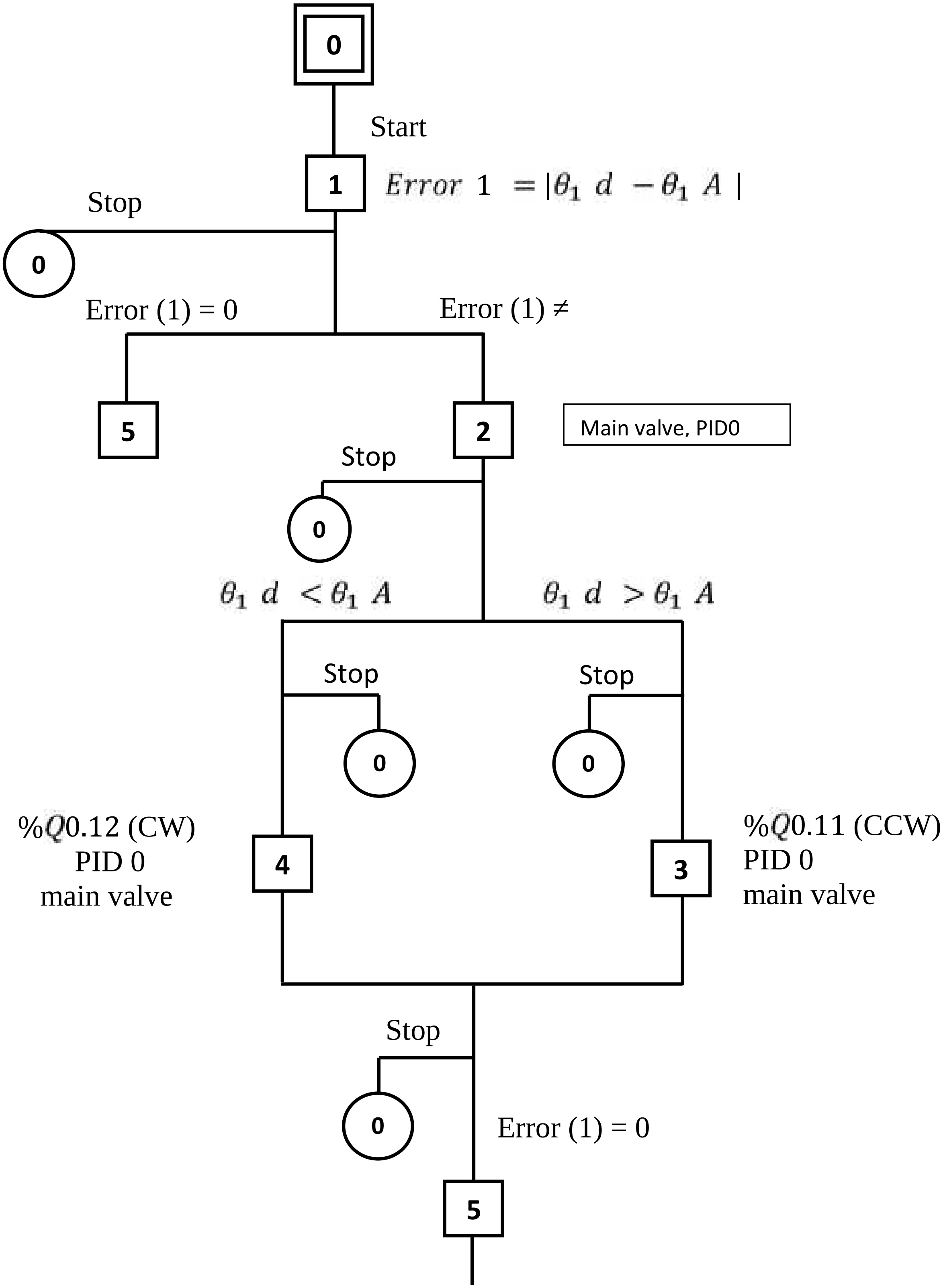
5) From the sequence function chart, it can be seen that when the start pushbutton is operated, this is the input condition for state 0.

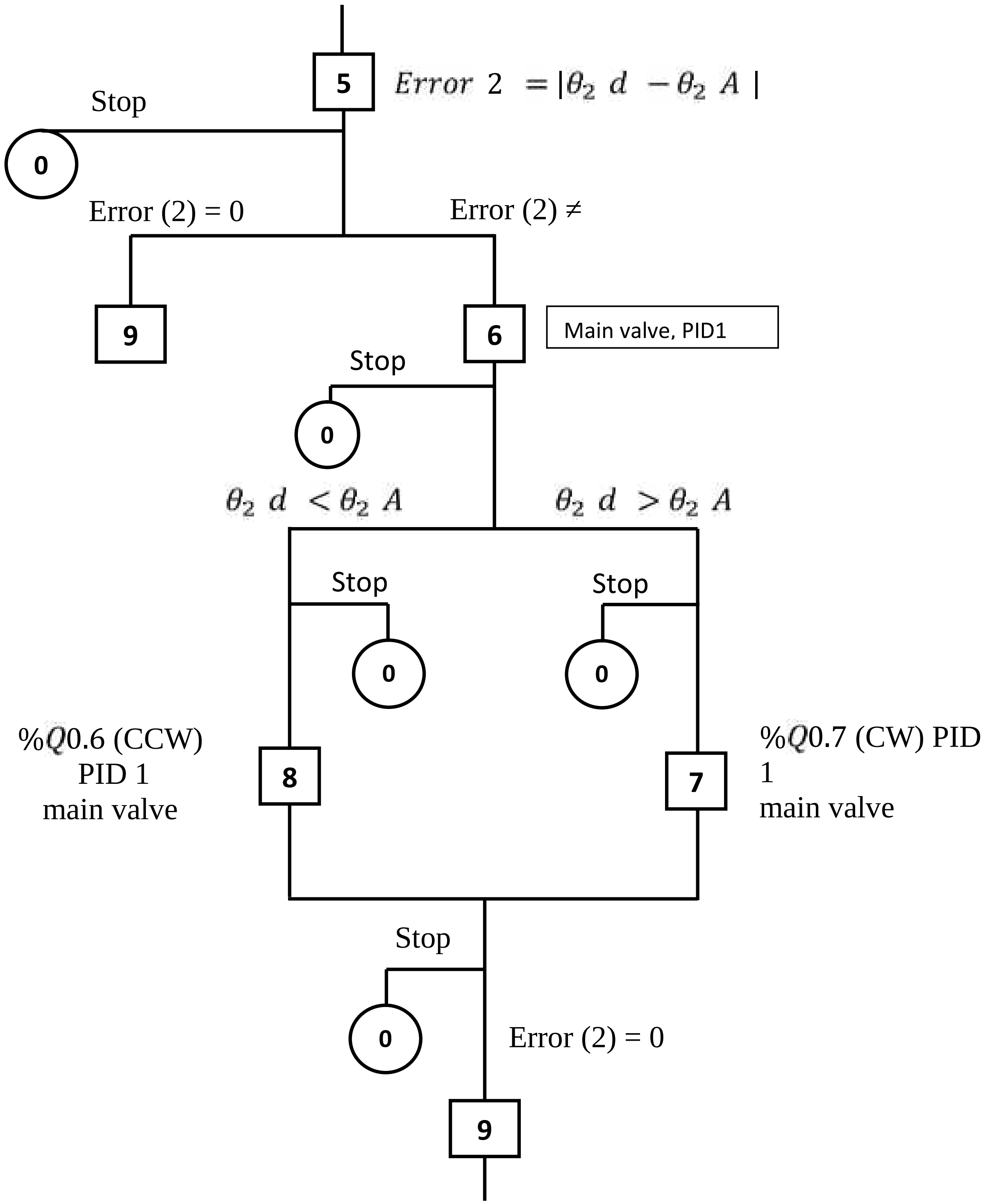
6) The output condition from State 0 is the startup sequence, which will reset both Solenoid A and Solenoid B. With Inputs X2 and X4 now made, the transfer from State 0 can take place.

7) The transfer conditions from State 0 are the correct input conditions for State 1, and hence the process now moves from State 0 to State 1.

8) The process will now continue from one state to the next, until the complete machine cycle is complete.
9) From the sequence function chart, the ladder diagram can now be produced

For the articulated manipulator the sequence function chart is shown in Figure 6.2




5
$Error\ 2\ = |\theta_2\ d\ - \theta_2\ A\ |$
Stop
0
Error (2) = 0
Error (2) ≠
9
6
Main valve, PID1
Stop
0
$\theta_2\ d\ < \theta_2\ A$
$\theta_2\ d\ > \theta_2\ A$
Stop
Stop
0
0
%Q0.6 (CCW) PID 1 main valve
8
7
%Q0.7 (CW) PID 1 main valve
Stop
0
Error (2) = 0
9

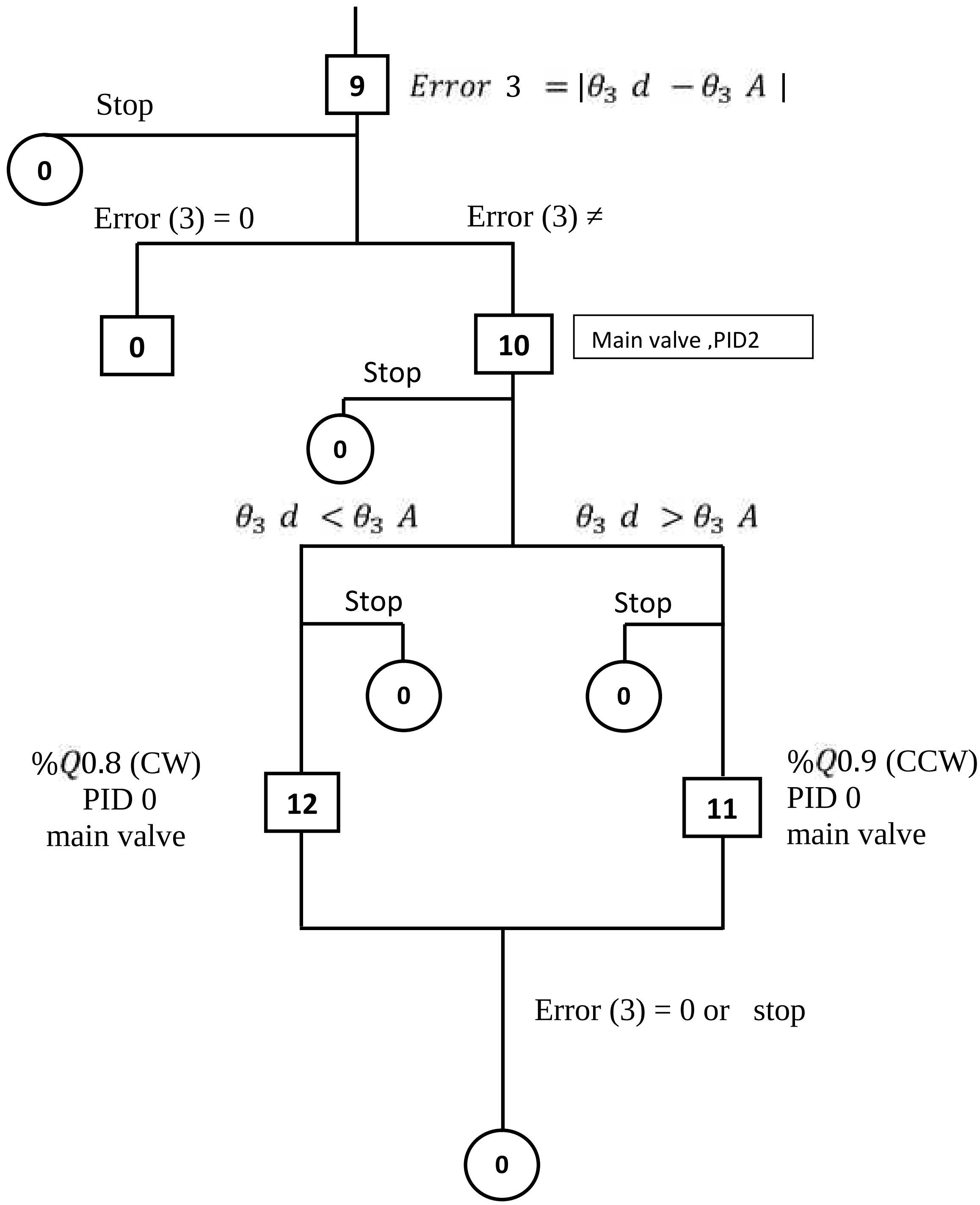


Figure 6.2: State graph for the articulated manipulator

This state graph is for three joint controls, the state graph sequence start at state zero and at this state nothing is active. And as we see there is Stop condition input at each state, this input reset all states and take the sequence to state 0, and any active device will turn OFF.

When the user push on start switch which is the input condition to next state, the state one is active but there is not outputs, it just take to two choices (two input conditions) to the next step.

The input conditions now are a comparison of the error (1) it equal zero or not, if the error is equal zero then sequence go to step five which to move joint two, if the error (1) is not equal zero then the sequence go to state two and there is another comparison between the desired and actual angle.

If the desired angle is more than the actual state three will activated and the main valve, PID 0 and $Q0.11$ is turned on, the result of this is move link A counter clock wise according to the desired angle. If the desired angle is less than the actual state four ($Q0.12$ turned on) which move link A clock wise to the desired angle. After link A reach desired angle the sequence move to state five.

The same thing do to the joint two and joint three until the end effectors of the Robot reach to the desired position.

Note: The Ladder Program is shown in Appendix A

6.4 PID functions on Plc

6.4.1 Introduction

The PID control function onboard all Twido controllers provides an efficient control to simple industrial processes that consist of one system stimulus (referred to as Set point in this Book) and one measurable property of the system (referred to as Measure or Process Variable).

The approach of PID controller used in this project is to achieve responsive and accurate positioning performance of the end effectors of the Robot, for each link (joint) we have a PID controller. By the PID controller the end effectors start moving at somewhat high speed then the speed decreased according to the error which is the difference between actual angle of joint and the desired angle.

This regulation function is particularly adapted to:

1) Answering the needs of the sequential process which need the auxiliary adjustment functions (examples: plastic film packaging machine, finishing treatment machine, presses, etc.)

2) Responding to the needs of the simple adjustment process (examples: metal furnaces, ceramic furnaces, small refrigerating groups, etc.)

It is very easy to install as it is carried out in the:

1. Configuration
2. and Debug

Screens associated with a program line (operation block in Ladder Language or by simply calling the PID in Instruction List) indicating the number of the PID used. The correct syntax when writing a PID instruction is: **PID<space>n**, when n is the PID number.

Example of a program line in Ladder Language:

NOTE: In any given Twido automation application, the maximum number of configurable PID functions is 14.

6.4.2 The PID Controller Model

The Twido PID controller implements a mixed (serial - parallel) PID correction (see PID Model Diagram below) via an analog measurement and set point in the [0-10000] format and provides an analog command to the controlled process in the same format.
The mixed form of the PID controller model is described in the following diagram:

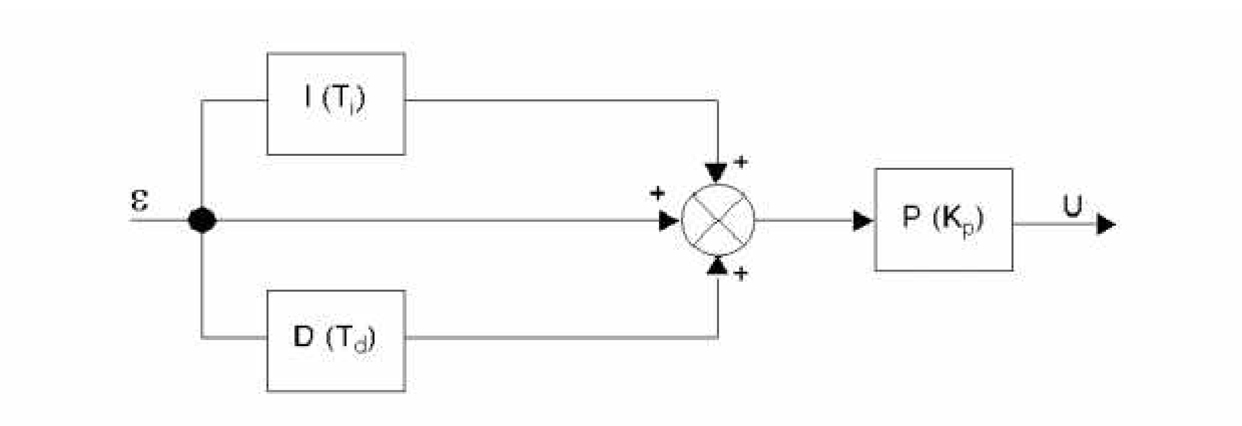


Figure 6.3 PID controller model

Where:

- I = the **integral** action (acting independently and parallel to the derivative action),
- D = the **derivative** action (acting independently and parallel to the integral action),
- P = the **proportional** action (acting serially on the combined output of the integral and derivative actions,
- U = the PID controller output (later fed as input into the controlled process.)

The PID controller is comprised of the mixed combination (serial - parallel) of the controller gain (Kp), and the integral (Ti) and derivative (Td) time constants. Thus the PID control law that is used by the Twido controller is of the following form:

$$u\ i\ = K_p.\ \ \varepsilon\ i\ + \frac{T_s}{T_i}\sum_{j=1}^{i}\varepsilon\ j\ + \frac{T_d}{T_s}\ \varepsilon\ i\ - \varepsilon\ i - 1 \tag{6.1}$$

Where

- Kp = the controller proportional gain,
- Ti = the integral time constant,
- Td = the derivative time constant,
- Ts = the sampling period,
- ε(i) = the deviation (ε(i) = set point - process variable.)

NOTE: Two different computational algorithms are used, depending on the value of the integral time constant (Ti):

- Ti ≠ 0: In this case, an incremental algorithm is used.
- Ti = 0: This is the case for non-integrating processes. In this case, a positional algorithm is used, along with a +5000 offset that is applied to the PID output variable.

6.4.3 Operating Principles

The following diagram presents the operating principle of the PID function

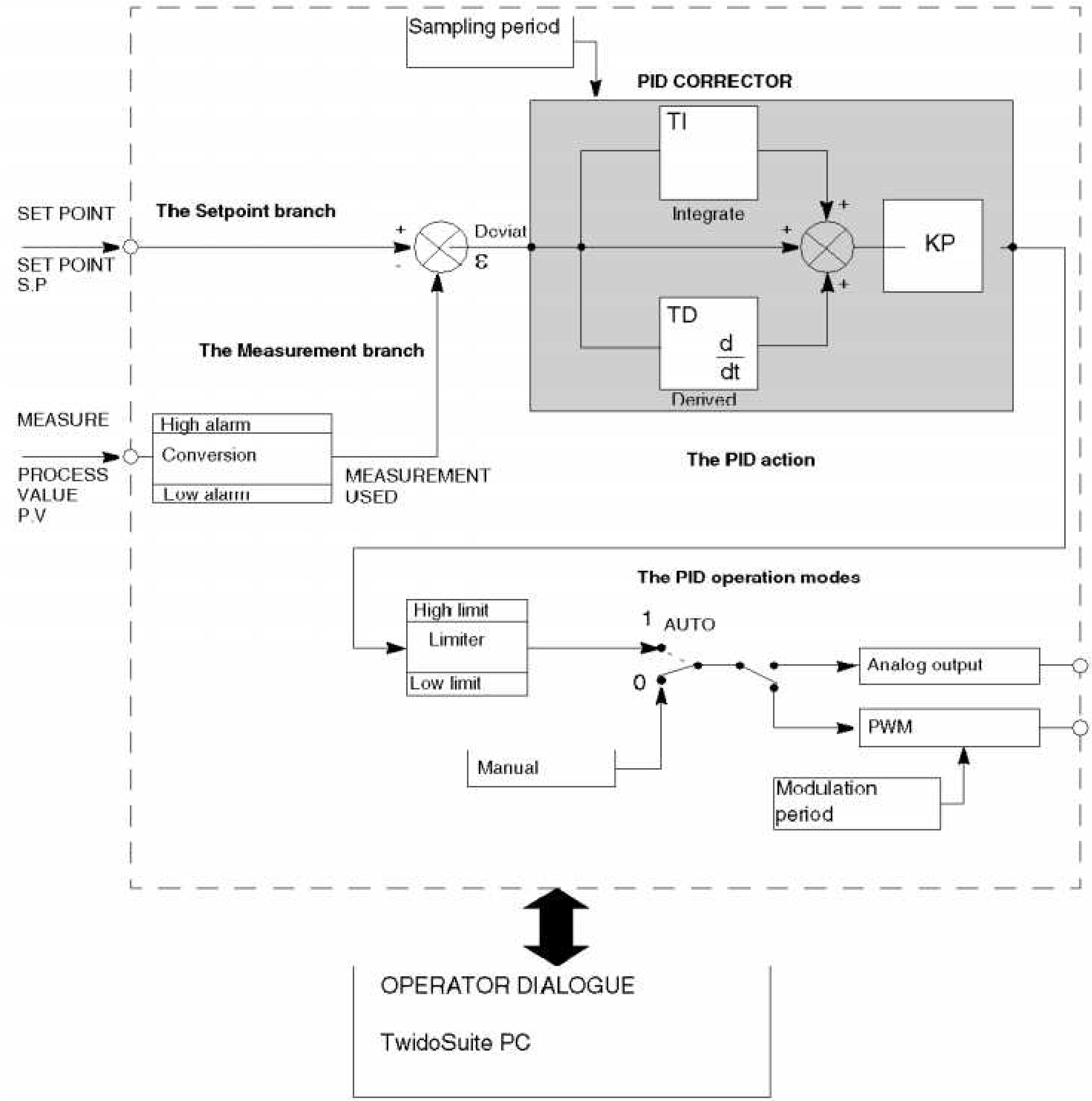


Figure 6.4: PID Operating Principals on PLC

Her the setpoint represent the desired angle that the joint should go and the measure point is the actual angle of the respective joint, the error of the summing point enters to the PID controller (PI in our case) .According to Controller gains there will be an output, this output can be limited using saturation or limiter as shown in the figure .In our project we use saturation for each joint with min value 3 v and max value 7 v ,since we don’t need a very high speed or very low speed .

The PID function has two modes: analog output or PWM output, in our project we use the analog output to control the proportional valve opening which control the flow of the system.

6.4.4 Principal of the Regulation Loop

At a Glance

The working of a regulation loop has three distinct phases:

- The acquisition of data:
    - Measurements from the process' sensors (analog, encoders)
    - Setpoint(s) generally from the controller's internal variables or from data from a TwidoSuit animation table
- Execution of the PID regulation algorithm.
- The sending of orders adapted to the characteristics of the actuators to be driven via the discrete (PWM) or analog outputs.

The PID algorithm generates the command signal from:

- The measurement sampled by the input module
- The setpoint value fixed by either the operator or the program
- The values of the different corrector parameters

The signal from the corrector is either directly handled by a controller analog output card linked to the actuator, or handled via a PWM adjustment on a discrete output of the controller.

Illustration

The following diagram schematizes the principal of a regulation loop

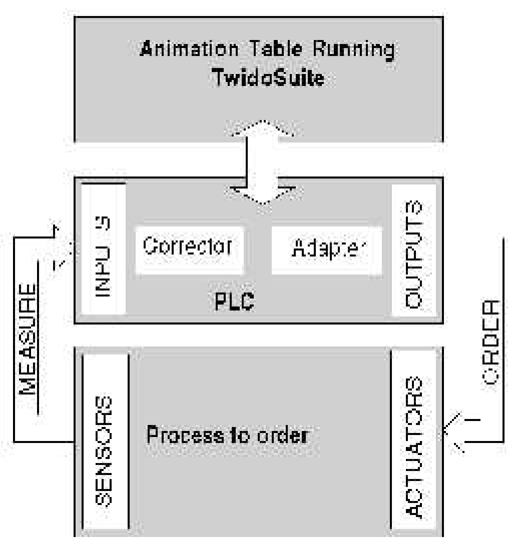


Figure 6.5: principal of a regulation loop

6.4.5 Role and Influence of PID Parameters

Influence of Proportional Action

Proportional action is used to influence the process response speed. The higher the gain, the faster the response and the lower the static error (in direct proportion), though the more stability deteriorates. A suitable compromise between speed and stability must be found. The influence of Proportional on process response to a Scale division is as follows:

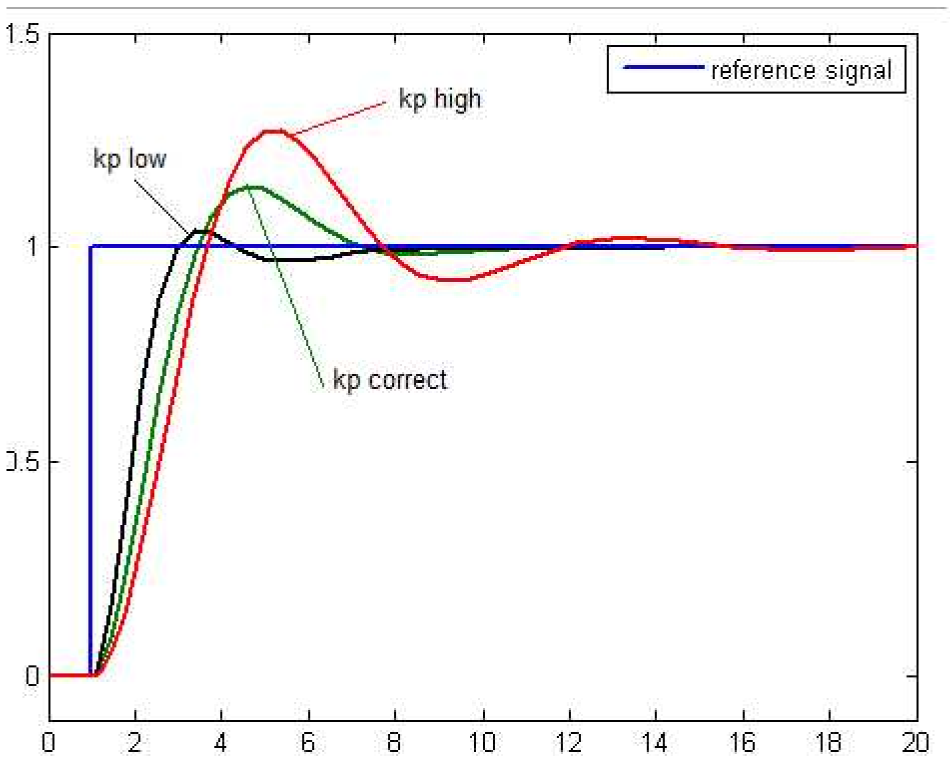


Figure 6.6: Influence of Proportional Action

Influence of Integral Action

Integral action is used to cancel out static error (deviation between the process value and the set point). The higher the level of integral action (low Ti), the faster the response and the more stability deteriorates. It is also necessary to find a suitable compromise between speed and stability. The influence of integral action on process response to a scale division is as follows:

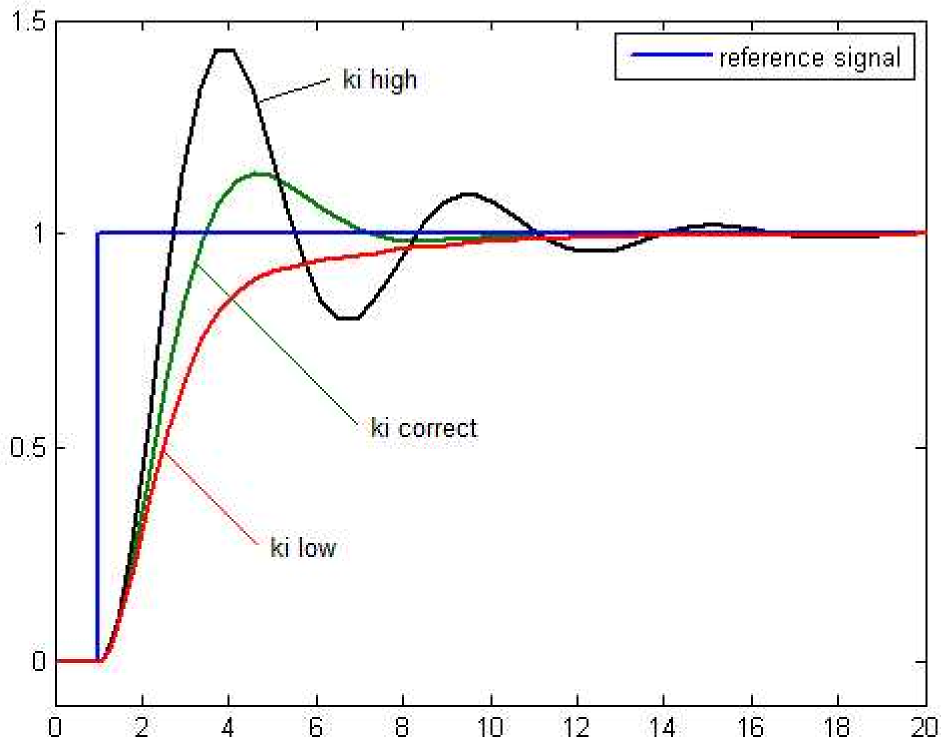


Figure 6.7: Influence of Integral Action

NOTE: A low Ti means a high level of integral action.

Where Kp = proportional gain, Ti = integration time and Td = derivative time.

Influence of Derivative Action

Derivative action is anticipatory. In practice, it adds a term which takes account of the speed of variation in the deviation, which makes it possible to anticipate changes by accelerating process response times when the deviation increases and by slowing them down when the deviation decreases. The higher the level of derivative action (high Td), the faster the response. A suitable compromise between speed and stability must be found. The influence of derivative action on process response to scale division is as follows:

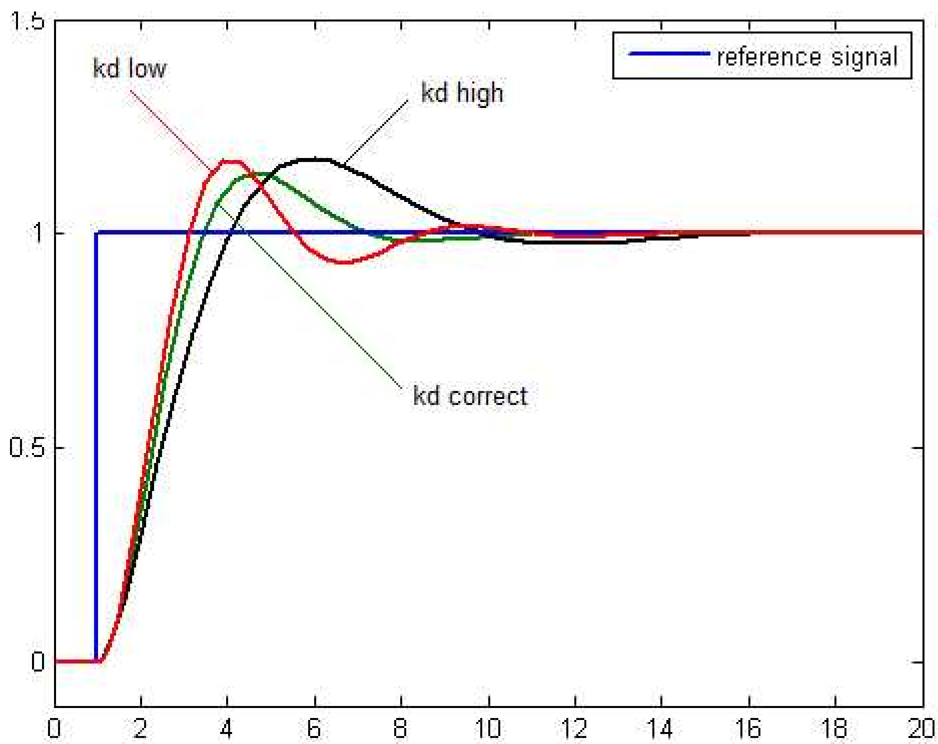


Figure 6.7: Influence of Derivative Action

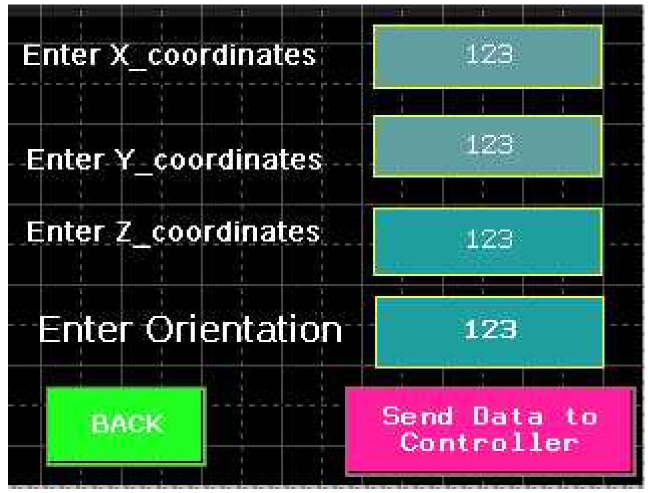
6.5 Touch Screen and Vijeo Designer

The touch screen that we use in this project is Magelis XPT OT 2110 Schneider Touch screen; we decide to choose this touch screen because it meet the specification required and hiveless cost and have the following specification:

Type	Advanced touch screen panel
Display type	Backlit monochrome STN LCD
Supply Voltage Range:	19.2V DC to 28.8V DC
Display color	16 levels of grey Blue and white
Display resolution	320 x 240 pixels QVGA
Display	size 5.7 inch
Software type	Configuration software
Software designation	Vijeo Designer
Operating system	Magelis
Processor name	CPU RISC
Processor frequency	133 MHz
Memory description	Back up of data SRAM 128 kB lithium battery Application memory flash EPROM 16 MB

We use Vijeo Designer Opti software to program the touch screen, and by touch screen the user can inter the desired coordinate of the end effectors (X, Y,Z).This coordinates are transferred to PLC via Modbus communication protocol which is a master/slave protocol that’s allow for one, and only one, master to request responses from slaves or to act based on the request.

The touch screen will be like this:



The user is asked to enter the desired position then these coordinates are transferred to plc to move to the desired position

Chapter Seven

Experiments and results

7.1 Introduction

7.2 Experimental results

7.3 Conclusion

7.4 Future Work

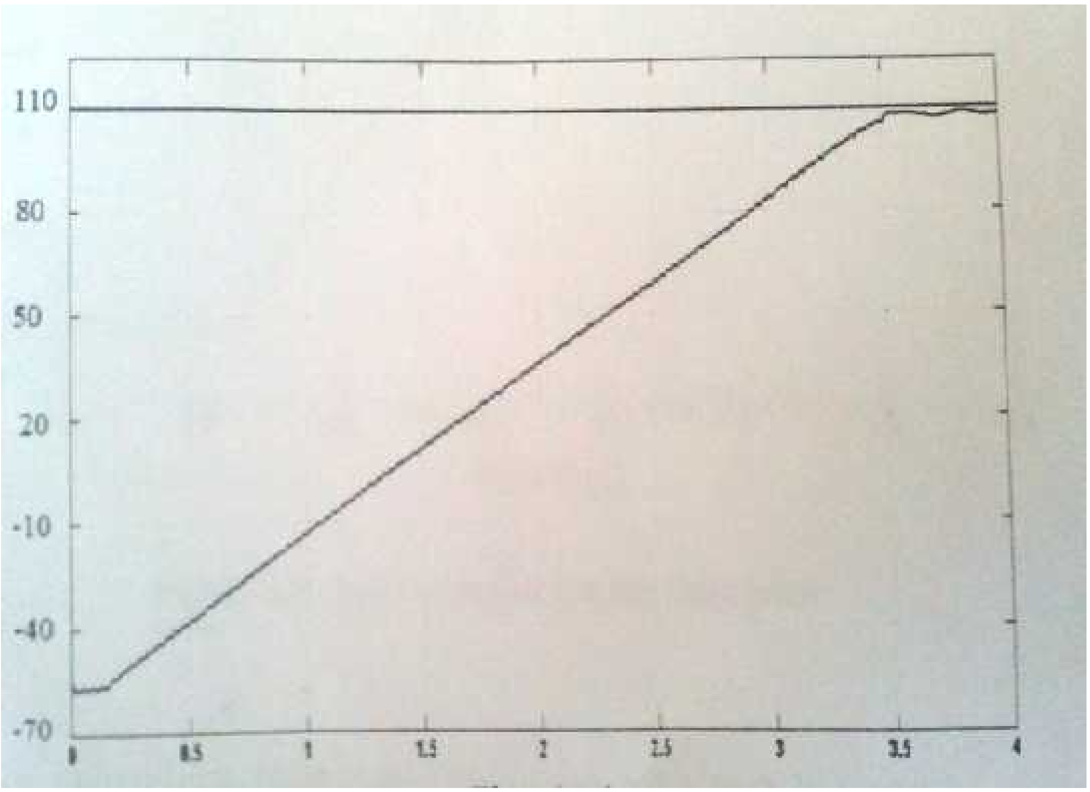
7.1 Introduction

This chapter contains the results that are obtained from the experiments which are done to verify the theoretical results that disused on the previous chapters. Then these results are discussed and explained from the experimental side.

7.2 Experimental results

In this section we mention the experimental results that we performed, here we show the step response of each joint as shown below:

First Joint



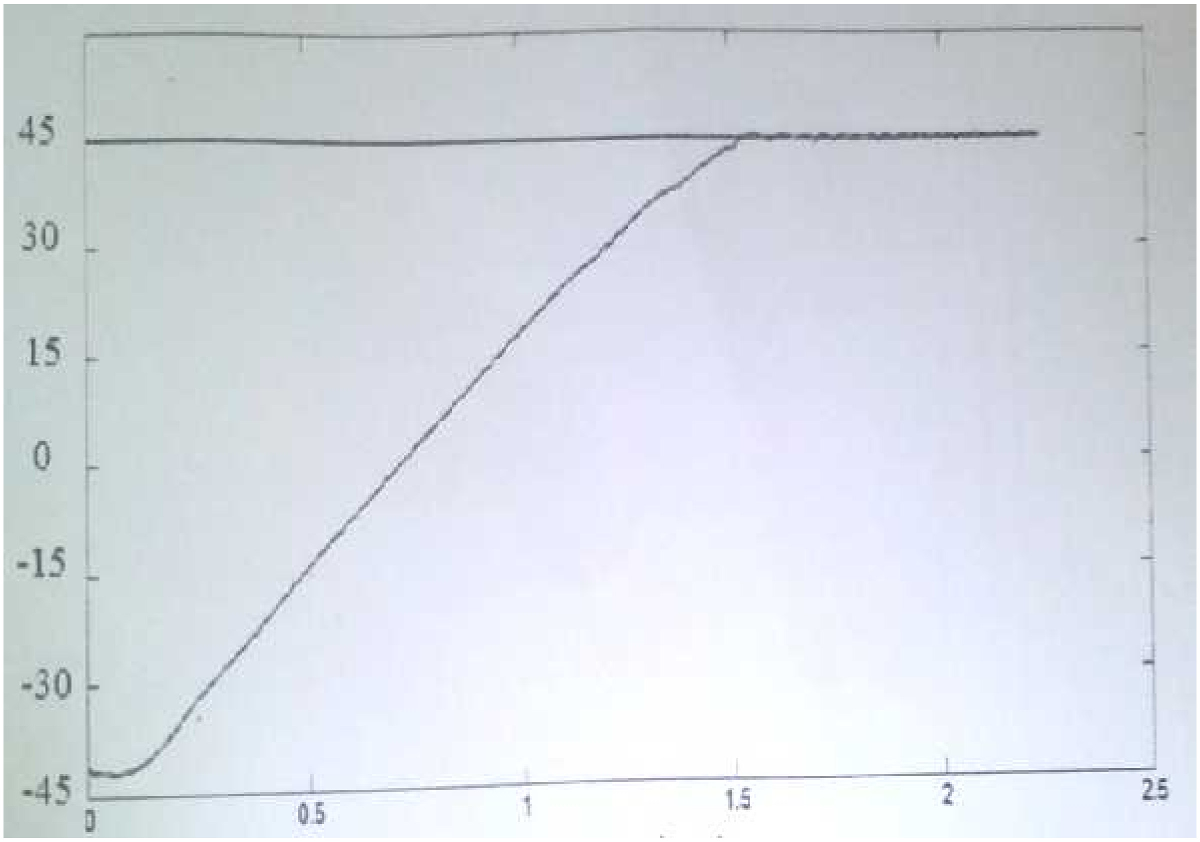
Position(degree)

Time(sec)

Figure7.1:step response of first joint

second Joint

Position(degree)



Time(sec)

Figure7.2:step response of second joint

Third Joint

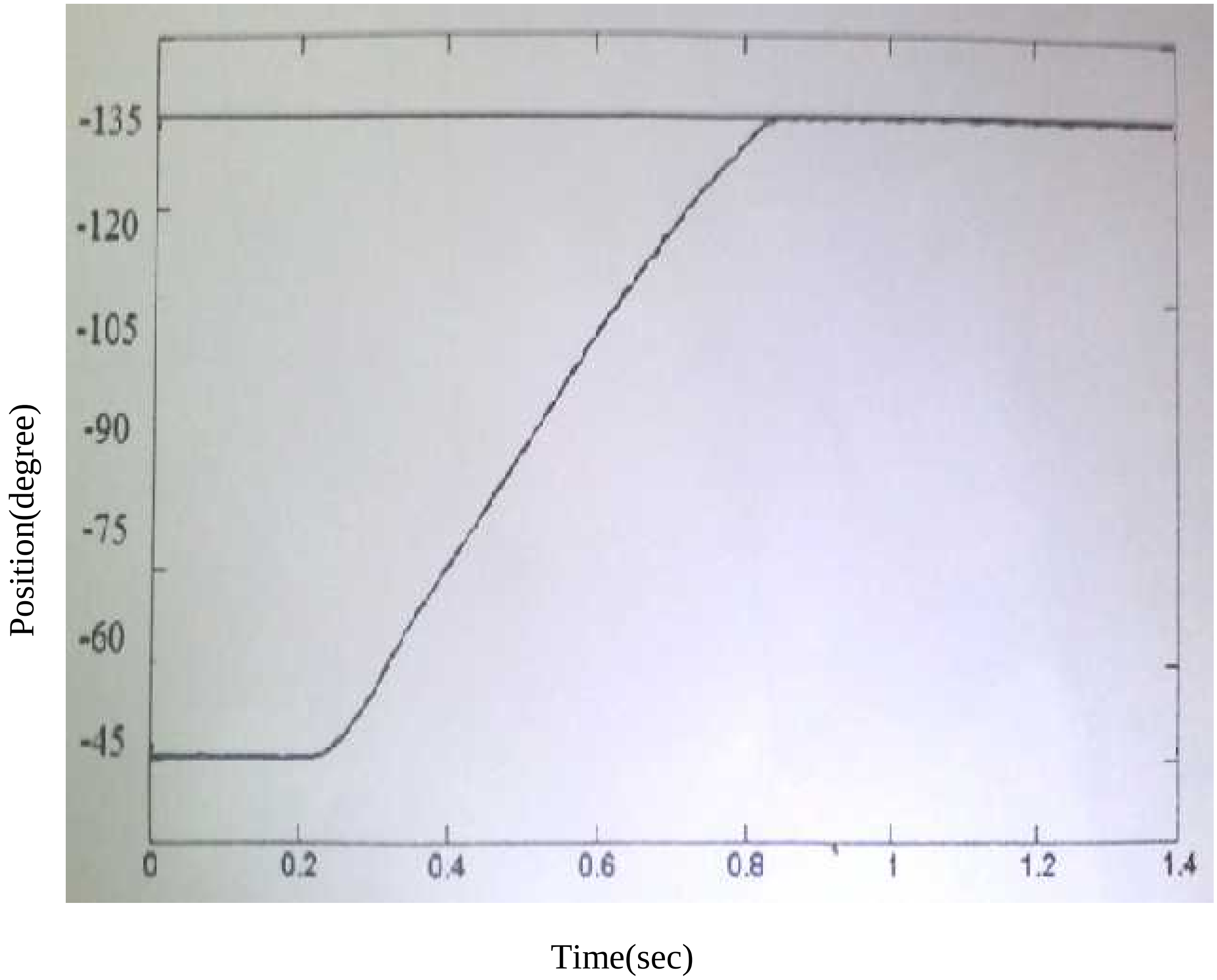


Figure7.3:step response of third joint

Each time we perform the experiments there is an error on each angle and this error appear due to several Reasons:

1) The hydraulic system is a non linear system so we can't model the system to find the appropriate controller, however we use tuning process to find the most suitable controller based on experimental computation.
2) The feedback sensors are potiometers so there must be an error associated with the mechanical part of the sensor, and as a result the robot will reach the desired position with some certain errors.

7.3 Conclusion

Robotics has become recently an interesting area of research. In our project we apply robotic theories, control techniques to accomplish the main goal of our project which is to move the robot to the desired position.

Moving the robot requires a control system, we used the Twido controller for this task, and based on some control algorithms we program our system. Moving all the joint together can't be accomplished using PLC so we design control technique to move the first joint followed by the second joint and then the third joint.

7.4 Future Work

The results of this work can be a basic point for the future studies .we can design different control algorithms with different hardware and servo system to produce a high performance robot arm. Another subject is that we could use another method of control; it may be microcontroller or DAQ system.

Appendices A

TwidoSuit Programs

TwidoSuite
Articulated Manipulator

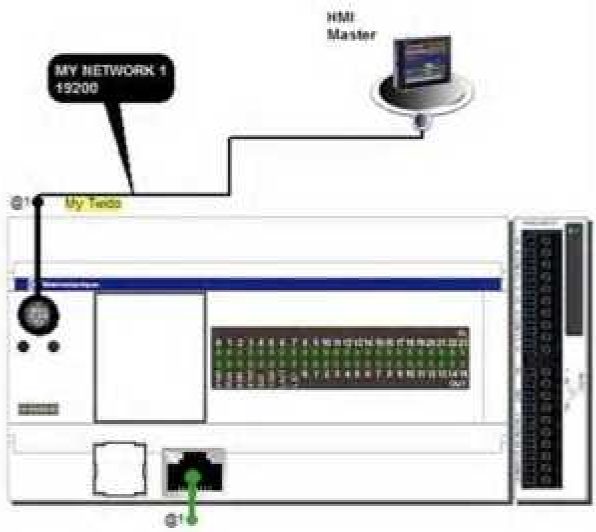
Project Information

Print date	7/31/2013
Author	
Department	
Index	
Industrial Property	

Comment

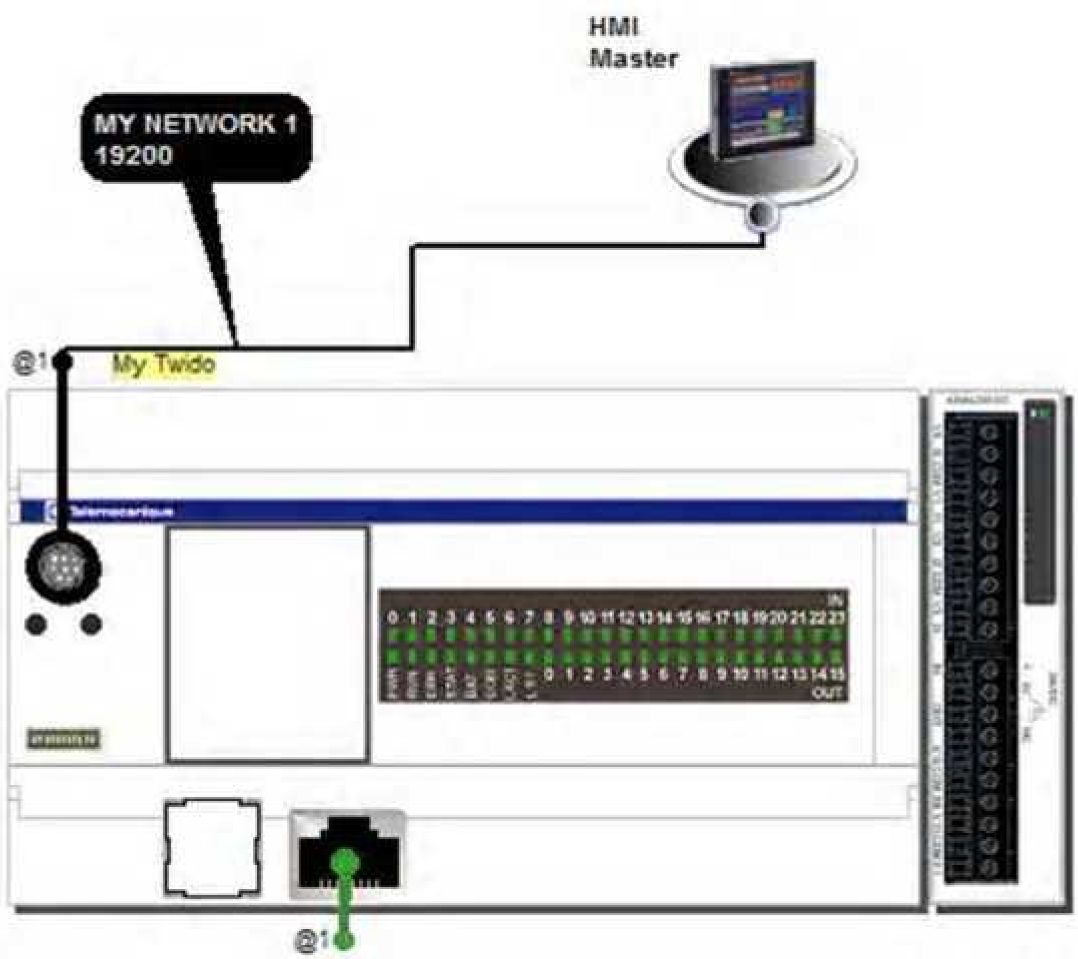
History

Contents



Page 1

Page 2



Page 1

Properties

Network MY NETWORK 1 :

Parameters

Type : Modbus
Address : 1
Baudrate : 19200
Data Bits : 8 (RTU)
Parity : even
Stop Bit : 1
Response Timeout: (x 100ms) : 10
Time between frames (ms) : 10

Element

Name	Type	Address
HMI	Modbus	Master

Ethernet Port :

IP address configuration

IP address :
Subnet mask :
Gateway address :

Marked IP

Marked IP address :

Idle time

Time (min) : 10

Remote devices

Index	IP	Unit ID	Timeout

Bill of material

Family	Reference number	Quantity
Twido	TWDLCDE40DRF	1
Twido	TM2AMM6HT	1
Modbus elements	Magelis	1

Hardware configuration

Base

TWDLCDE40DRF

Expansion bus modules

1 : TM2AMM6HT

Memory objects configuration

Timer configuration (%TM)

Counter configuration (%C)

Register configuration (%R)

Drum configuration (%DR)

Scheduler block configuration (%SCH)

Fast counters configuration (%FC)

Very fast counters configuration (%VFC)

Memory words (%MD)

Memory words (%MW)

Used	%MW	Symbol	Allocated
Yes	%MW0		Yes
Yes	%MW1		Yes
Yes	%MW20	Z1_ACT	Yes
Yes	%MW22	Z1_DIS	Yes
Yes	%MW30	Z2_DIS	Yes
Yes	%MW39	Z3_ACT	Yes
Yes	%MW76		Yes
Yes	%MW88		Yes
Yes	%MW90	Z2_ACT	Yes
Yes	%MW100		Yes
Yes	%MW156		Yes
Yes	%MW160		Yes
Yes	%MW193		Yes
Yes	%MW333		Yes
Yes	%MW350		Yes
Yes	%MW702	Z3_DIS	Yes

Memory words (%MF)

Used	%MF	Symbol	Allocated
Yes	%MF127		Yes
Yes	%MF130	X0	Yes
Yes	%MF132	Y0	Yes
Yes	%MF134	Z0	Yes
Yes	%MF200	Z_DIS1	Yes
Yes	%MF202	Z_ACT1	Yes
Yes	%MF203	E2	Yes
Yes	%MF204	E1	Yes
Yes	%MF206		Yes
Yes	%MF208	VOLTAGE_IN_CHANAL_1	Yes
Yes	%MF212	V2	Yes
Yes	%MF214	V3	Yes
Yes	%MF252		Yes
Yes	%MF290	E2_OO	Yes
Yes	%MF300		Yes

Used	%MF	Symbol	Allocated
Yes	%MF302	Z_ACT2	Yes
Yes	%MF306	Z_DIS2	Yes
Yes	%MF315		Yes
Yes	%MF317	VOLTAGE_IN_CH2	Yes
Yes	%MF320		Yes
Yes	%MF402	Z_ACT3	Yes
Yes	%MF407	Z_DIS3	Yes
Yes	%MF412		Yes
Yes	%MF420	X1	Yes
Yes	%MF422	Y1	Yes
Yes	%MF424	Z1	Yes
Yes	%MF430		Yes
Yes	%MF432		Yes
Yes	%MF434		Yes
Yes	%MF436		Yes
Yes	%MF563	E3	Yes
Yes	%MF566	E3_OO	Yes
Yes	%MF600		Yes
Yes	%MF623		Yes
Yes	%MF650	VOLTAGE_IN_CH0	Yes
Yes	%MF777	VOLTAGE__OUT_CH0	Yes
Yes	%MF1006		Yes
Yes	%MF1008		Yes
Yes	%MF1010		Yes
Yes	%MF1012		Yes
Yes	%MF1014		Yes
Yes	%MF1016		Yes
Yes	%MF1018		Yes
Yes	%MF1042		Yes
Yes	%MF1044		Yes
Yes	%MF1046		Yes
Yes	%MF1048		Yes
Yes	%MF1050		Yes
Yes	%MF1052		Yes
Yes	%MF1054		Yes
Yes	%MF1056		Yes
Yes	%MF1058		Yes
Yes	%MF1060		Yes
Yes	%MF1062		Yes
Yes	%MF1066		Yes
Yes	%MF1068		Yes
Yes	%MF1070		Yes
Yes	%MF1088		Yes
Yes	%MF1090		Yes
Yes	%MF1092		Yes
Yes	%MF1094		Yes
Yes	%MF1096		Yes
Yes	%MF1098		Yes
Yes	%MF1102		Yes
Yes	%MF1104		Yes
Yes	%MF1106		Yes
Yes	%MF1200		Yes
Yes	%MF1202		Yes
Yes	%MF1204		Yes
Yes	%MF1208		Yes
Yes	%MF1210		Yes
Yes	%MF1212		Yes
Yes	%MF1214		Yes
Yes	%MF1216		Yes
Yes	%MF1218		Yes

Used	%MF	Symbol	Allocated
Yes	%MF1800	XC	Yes
Yes	%MF1802	YC	Yes
Yes	%MF1804	ZC	Yes
Yes	%MF2008		Yes
Yes	%MF2013	THETA2	Yes
Yes	%MF2025	THETA3	Yes
Yes	%MF2156		Yes

Memory bits (%M)

Used	%M	Symbol	Allocated
Yes	%M0	T0_1	Yes
Yes	%M1	T1_2	Yes
Yes	%M2	T2_3	Yes
Yes	%M3	T3_4	Yes
Yes	%M4	T4_5	Yes
Yes	%M5	T4_6	Yes
Yes	%M6	T1_0	Yes
Yes	%M14	T5_7	Yes
Yes	%M15	T6_7	Yes
Yes	%M16	T7_8	Yes
Yes	%M17	T8_9	Yes
Yes	%M18	T9_10	Yes
Yes	%M19	T10_11	Yes
Yes	%M20	T11_12	Yes
Yes	%M21	T11_13	Yes
Yes	%M22	T12_14	Yes
Yes	%M23	T13_14	Yes
Yes	%M31	T14_15	Yes
Yes	%M32	T15_16	Yes
Yes	%M40	T21_0	Yes
Yes	%M41	T9_8	Yes
Yes	%M42	T16_15	Yes
Yes	%M43	T16_17	Yes
Yes	%M44	T17_18	Yes
Yes	%M45	T18_19	Yes
Yes	%M46	T18_20	Yes
Yes	%M47	T19_21	Yes
Yes	%M48	T20_21	Yes
Yes	%M57	START	Yes
Yes	%M59	STOP	Yes
Yes	%M77		Yes
Yes	%M88		Yes
Yes	%M90		Yes
Yes	%M91		Yes
Yes	%M92		Yes
Yes	%M99		Yes
Yes	%M100	ANGLE1_REACHED	Yes
Yes	%M101	ANGLE2_REACHED	Yes
Yes	%M233		Yes
Yes	%M244		Yes

PID configuration (PID)

PID 0 : configured

General					
Operating mode	: Word address				
PID Status	: Allow				
Input					
Measure	: %MW20				
Conversion	: Inhibit	Min	:	Max	:

Alarms	: Inhibit	Low	:	Output	:
		High	:	Output	:
		PID			
Setpoint	: %MW22				
Kp	: 5000	Ti	: 12000	Td	: 0
Sampling |Period	: 2				
		AT			
AT mode	: Inhibit	Limit	:	Output	:
		Output			
Action	: %M77				
Limits	: Allow	Min	: 1023	Max	: 1440
Manual mode	: Inhibit	Output	:		
Output analog	: %MW88				
PWM	: Inhibit	Period	:	Output	:
PID 1	: configured				
		General			
Operating mode	: Word address				
PID Status	: Inhibit				
		Input			
Measure	: %MW90				
Conversion	: Inhibit	Min	:	Max	:
Alarms	: Inhibit	Low	:	Output	:
		High	:	Output	:
		PID			
Setpoint	: %MW30				
Kp	: 5000	Ti	: 12000	Td	: 0
Sampling |Period	: 2				
		AT			
AT mode	: Inhibit	Limit	:	Output	:
		Output			
Action	: %M88				
Limits	: Allow	Min	: 1280	Max	: 2000
Manual mode	: Inhibit	Output	:		
Output analog	: %MW76				
PWM	: Inhibit	Period	:	Output	:
PID 2	: configured				
		General			
Operating mode	: Word address				
PID Status	: Inhibit				
		Input			
Measure	: %MW39				
Conversion	: Inhibit	Min	:	Max	:
Alarms	: Inhibit	Low	:	Output	:
		High	:	Output	:
		PID			
Setpoint	: %MW702				
Kp	: 5000	Ti	: 12000	Td	: 0
Sampling |Period	: 2				
		AT			
AT mode	: Inhibit	Limit	:	Output	:
		Output			
Action	: %M99				
Limits	: Allow	Min	: 1230	Max	: 1640
Manual mode	: Inhibit	Output	:		
Output analog	: %MW193				
PWM	: Inhibit	Period	:	Output	:

Constant configuration (%KD)

Constant configuration (%KW)

Constant configuration (%KF)

Used	%KF	Symbol	Float
Yes	%KF50	ZERO	0.0
Yes	%KF60	C1	0.0024414
Yes	%KF90	G_L	16.5
Yes	%KF233	D1	0.27235
Yes	%KF240	A2	0.2604
Yes	%KF244	A3	0.25

PLS/PWM configuration (%PLS/%PWM)

Configuration of external objects Comm

Configuration of external objects Drive

Configuration of external objects Tesys

Configuration of external objects Advantys OTB

Memory

Memory usage statistic

User data			
Memory bits	:	245 Bits	0.5%
Memory words	:	2158 Words	62.6%
Backed up	:	715 Words	
RAM = EEPROM	:	???	
Constants	:	246 Words	7.2%
Configuration	:	873 Words	25.3%
Avail. mem. data	:	25 Words	0.6%

User program			
Executable code	:	2560 Words	16.6%
Prog. data	:	4 Words	0.1%
Online modif.	:	0 Words	0.0%
Avail. code mem.	:	12161 Words	78.7%

Other			
Execution data	:	127 Words	3.7%
:	127 Words	3.7%	

Configure the behavior

Functional levels

Functional levels management

Management :	Automatic
Level :	The highest possible

Scan mode

Scan mode

Mode :	Periodic
Duration (ms) :	150

Watchdog

Duration (ms) :	20

Periodic event

Not used :	Yes

Startup

Parameters

Automatic start in Run :	No
Run/Stop Input:	None

Autosave

Parameters

Autosave RAM=>EEPROM :	Yes

Program lists and diagrams

1 LD

Rung 0

SHORT

X1 := 27.0 - X0

%MF420 := 27.0 - %MF130

Rung 1

SHORT

Y1 := 9.3 + Y0

%MF422 := 9.3 + %MF132

Rung 2

SHORT

Z1 := Z0

%MF424 := %MF134

Rung 3

SHORT

%MF430 := COS(%MF2008)

%MF430 := COS(%MF2008)

Rung 4

SHORT

%MF432 := SIN(%MF2008)

%MF432 := SIN(%MF2008)

Rung 5

SHORT

%MF434 := G_L * %MF430

%MF434 := %KF90 * %MF430

Rung 6

SHORT

%MF436 := G_L * %MF432

%MF436 := %KF90 * %MF432

Rung 7

SHORT

XC := X1 + %MF434

%MF1800 := %MF420 + %MF434

Rung 8

SHORT

YC := Y1 - %MF436

%MF1802 := %MF422 - %MF436

Rung 9

SHORT

ZC := Z0
%MF1804 := %MF134

2 LD Transitions - Sequential Function Chart [SFC]

YC/XC

Rung 0

SHORT

%MF1006 := YC / XC
%MF1006 := %MF1802 / %MF1800

Rung 1

SHORT

%MF1008 := ATAN(%MF1006)
%MF1008 := ATAN(%MF1006)

THETA 1 FOR C1 : : : : : : : : : :: : :: : :: : : :: : : : :: : : :: : :: : : :: : : :: : : : :: : : :

Rung 2

SHORT

%MF2156 := RAD_TO_DEG(%MF1008)
%MF2156 := RAD_TO_DEG(%MF1008)

Rung 3

XC > 0.0
%MF1800 > 0.0
<

%MF2008 := %MF2156
%MF2008 := %MF2156

Rung 4

XC < 0.0
%MF1800 < 0.0
<

%MF1106 := -1.0 * %MF2156
%MF1106 := -1.0 * %MF2156

Rung 5

XC < 0.0
%MF1800 < 0.0
<

%MF2008 := 180.0 - %MF1106
%MF2008 := 180.0 - %MF1106

Rung 6

XC = 0.0
%MF1800 = 0.0
<

%MF2008 := 90.0
%MF2008 := 90.0

XC^2

Rung 7

SHORT

%MF1010 := XC * XC
%MF1010 := %MF1800 * %MF1800

YC^2

Rung 8

SHORT

%MF1012 := YC * YC
%MF1012 := %MF1802 * %MF1802

R1^2 = XC^2+YC^2

Rung 9

SHORT

%MF1014 := %MF1010 + %MF1012
%MF1014 := %MF1010 + %MF1012

R1=SQRT(XC^2+YC^2)

Rung 10

SHORT

%MF1016 := SQRT(%MF1014)
%MF1016 := SQRT(%MF1014)

S1=ZC-D1

Rung 11

SHORT

%MF1018 := ZC - 27.235
%MF1018 := %MF1804 - 27.235

(ZC-D1)^2

Rung 12

SHORT

%MF1042 := %MF1018 * %MF1018
%MF1042 := %MF1018 * %MF1018

R1^2 + S1^2 = (XC^2+YC^2)+(ZC-D1)^2

Rung 13

SHORT

%MF1044 := %MF1014 + %MF1042
%MF1044 := %MF1014 + %MF1042

A2^2

Rung 14

SHORT

%MF1046 := A2 * A2
%MF1046 := %KF240 * %KF240

A3^2

Rung 15

SHORT

%MF1048 := A3 * A3
%MF1048 := %KF244 * %KF244

(R1^2 + S1^2) - A2^2

Rung 16

SHORT

%MF1050 := %MF1044 - %MF1046
%MF1050 := %MF1044 - %MF1046

(R1^2 + S1^2) - A2^2 - A3^2

Rung 17

SHORT

%MF1052 := %MF1050 - %MF1048
%MF1052 := %MF1050 - %MF1048

A2*A3

Rung 18

SHORT

%MF1054 := A2 * A3
%MF1054 := %KF240 * %KF244

2*A2*A3

Rung 19

SHORT

%MF1056 := 2.0 * %MF1054
%MF1056 := 2.0 * %MF1054

D

Rung 20

SHORT

%MF1058 := %MF1052 / %MF1056
%MF1058 := %MF1052 / %MF1056

D < 0.0

Rung 21

%MF1058 >= 0.0
%MF1058 >= 0.0
<

%MF1102 := %MF1058
%MF1102 := %MF1058

Rung 22

%MF1058 < 0.0
%MF1058 < 0.0
<

%MF1102 := -1.0 * %MF1058
%MF1102 := -1.0 * %MF1058

Rung 23

SHORT

%MF1060 := ACOS(%MF1102)
%MF1060 := ACOS(%MF1102)

Rung 24

SHORT

%MF1104 := RAD_TO_DEG(%MF1060)
%MF1104 := RAD_TO_DEG(%MF1060)

THETA 3 FOR D > 0

Rung 25

%MF1058 > 0.0
%MF1058 > 0.0
<

%MF1062 := %MF1104
%MF1062 := %MF1104

THETA 3 FOR D < 0

Rung 26

%MF1058 < 0.0
%MF1058 < 0.0
<

%MF1062 := 180.0 - %MF1104
%MF1062 := 180.0 - %MF1104

THETA 3 FOR D = 0

Rung 27

%MF1058 = 0.0
%MF1058 = 0.0
<

%MF1062 := 90.0
%MF1062 := 90.0

Rung 28
SHORT
%MF1066 := %MF1018 / %MF1016
%MF1066 := %MF1018 / %MF1016

Rung 29
SHORT
%MF1068 := ATAN(%MF1066)
%MF1068 := ATAN(%MF1066)

SECONED PART OF THETA 2 FOR ZC >= D1 CASE 1

Rung 30
SHORT
%MF1070 := RAD_TO_DEG(%MF1068)
%MF1070 := RAD_TO_DEG(%MF1068)

ZC < D1

Rung 31
SHORT
%MF1088 := 27.235 - ZC
%MF1088 := 27.235 - %MF1804

Rung 32
SHORT
%MF1090 := %MF1088 / %MF1016
%MF1090 := %MF1088 / %MF1016

Rung 33
SHORT
%MF1092 := ATAN(%MF1090)
%MF1092 := ATAN(%MF1090)

S2^2

Rung 34
ZC >= D1
%MF1804 >= %KF233
<
%MF1096 := %MF1018 * %MF1018
%MF1096 := %MF1018 * %MF1018

Rung 35
ZC < D1
%MF1804 < %KF233
<
%MF1096 := %MF1088 * %MF1088
%MF1096 := %MF1088 * %MF1088

S2^2+R1^2

Rung 36
SHORT
%MF1098 := %MF1096 + %MF1014
%MF1098 := %MF1096 + %MF1014

S2^2+R1^2+A2^2

Rung 37
SHORT
%MF1200 := %MF1098 + %MF1046
%MF1200 := %MF1098 + %MF1046

S2^2+R1^2+A2^2-A3^2

Rung 38

SHORT

%MF1202 := %MF1200 - %MF1048
%MF1202 := %MF1200 - %MF1048

SQRT(R1^2+S2^2)

Rung 39

SHORT

%MF1204 := SQRT(%MF1098)
%MF1204 := SQRT(%MF1098)

A2 * (SQRT (R1^2+S2^2))

Rung 40

SHORT

%MF1208 := %MF1204 * A2
%MF1208 := %MF1204 * %KF240

2.0 * A2 * (SQRT (R1^2+S2^2))

Rung 41

SHORT

%MF1210 := 2.0 * %MF1208
%MF1210 := 2.0 * %MF1208

(S2^2+R1^2+A2^2-A3^2) / [2.0 * A2 * (SQRT (R1^2+S2^2))]

Rung 42

SHORT

%MF1212 := %MF1202 / %MF1210
%MF1212 := %MF1202 / %MF1210

ACOS [S2^2+R1^2+A2^2-A3^2 / SQRT(R1^2+S2^2)*A2*2.0]

Rung 43

SHORT

%MF1214 := ACOS(%MF1212)
%MF1214 := ACOS(%MF1212)

FIRST PART OF THETA 2

Rung 44

SHORT

%MF1216 := RAD_TO_DEG(%MF1214)
%MF1216 := RAD_TO_DEG(%MF1214)

THETA 2 FOR C1 :: : : :: : : : : :: : :: : : :: : : :: : :: : : : :: : :: : : : :: : :

Rung 45

ZC >= D1
%MF1804 >=
%KF233
<

%MF1218 := %MF1216 + %MF1070
%MF1218 := %MF1216 + %MF1070

THETA 2 FOR C2 :: : : :: : : : : :: : :: : : :: : : :: : :: : : : :: : :: : : : :: : :

Rung 46

ZC < D1
%MF1804 < %KF233
<

%MF1218 := %MF1216 - %MF1094
%MF1218 := %MF1216 - %MF1094

Rung 47

SHORT

Z_DIS1 := %MF2008
%MF200 := %MF2008

Rung 48

SHORT

Z_DIS2 := THETA2
%MF306 := %MF2013

Rung 49

SHORT

Z_DIS3 := THETA3
%MF407 := %MF2025

SECONED PART OF THETA 2 FOR ZC < D1 CASE 2

Rung 50

SHORT

%MF1094 := RAD_TO_DEG(%MF1092)
%MF1094 := RAD_TO_DEG(%MF1092)

S2^2

Rung 51

ZC >= D1
%MF1804 >= %KF233
<

%MF1096 := %MF1018 * %MF1018
%MF1096 := %MF1018 * %MF1018

Rung 52

ZC < D1
%MF1804 < %KF233
<

%MF1096 := %MF1088 * %MF1088
%MF1096 := %MF1088 * %MF1088

S2^2+R1^2

Rung 53

SHORT

%MF1098 := %MF1096 + %MF1014
%MF1098 := %MF1096 + %MF1014

S2^2+R1^2+A2^2

Rung 54

SHORT

%MF1200 := %MF1098 + %MF1046
%MF1200 := %MF1098 + %MF1046

S2^2+R1^2+A2^2-A3^2

Rung 55

SHORT

%MF1202 := %MF1200 - %MF1048
%MF1202 := %MF1200 - %MF1048

SQRT(R1^2+S2^2)

Rung 56

SHORT

%MF1204 := SQRT(%MF1098)
%MF1204 := SQRT(%MF1098)

A2 * (SQRT (R1^2+S2^2))

Rung 57

SHORT

%MF1208 := %MF1204 * A2
%MF1208 := %MF1204 * %KF240

2.0 * A2 * (SQRT (R1^2+S2^2))

Rung 58 — SHORT — %MF1210 := 2.0 * %MF1208
%MF1210 := 2.0 * %MF1208

(S2^2+R1^2+A2^2-A3^2) / [2.0 * A2 * (SQRT (R1^2+S2^2))]

Rung 59 — SHORT — %MF1212 := %MF1202 / %MF1210
%MF1212 := %MF1202 / %MF1210

ACOS [S2^2+R1^2+A2^2-A3^2 / SQRT(R1^2+S2^2)*A2*2.0]

Rung 60 — SHORT — %MF1214 := ACOS(%MF1212)
%MF1214 := ACOS(%MF1212)

FIRST PART OF THETA 2

Rung 61 — SHORT — %MF1216 := RAD_TO_DEG(%MF1214)
%MF1216 := RAD_TO_DEG(%MF1214)

THETA 2 FOR C1 :: : : : :: : : : : : :: : :: : :: : : : :: : :: : : : :: : :: : : : :: : :

Rung 62 — ZC >= D1 / %MF1804 >= %KF233 (<) — %MF1218 := %MF1216 + %MF1070
%MF1218 := %MF1216 + %MF1070

THETA 2 FOR C2 :: : : : :: : : : : : :: : :: : :: : : : :: : :: : : : :: : :: : : : :: : :

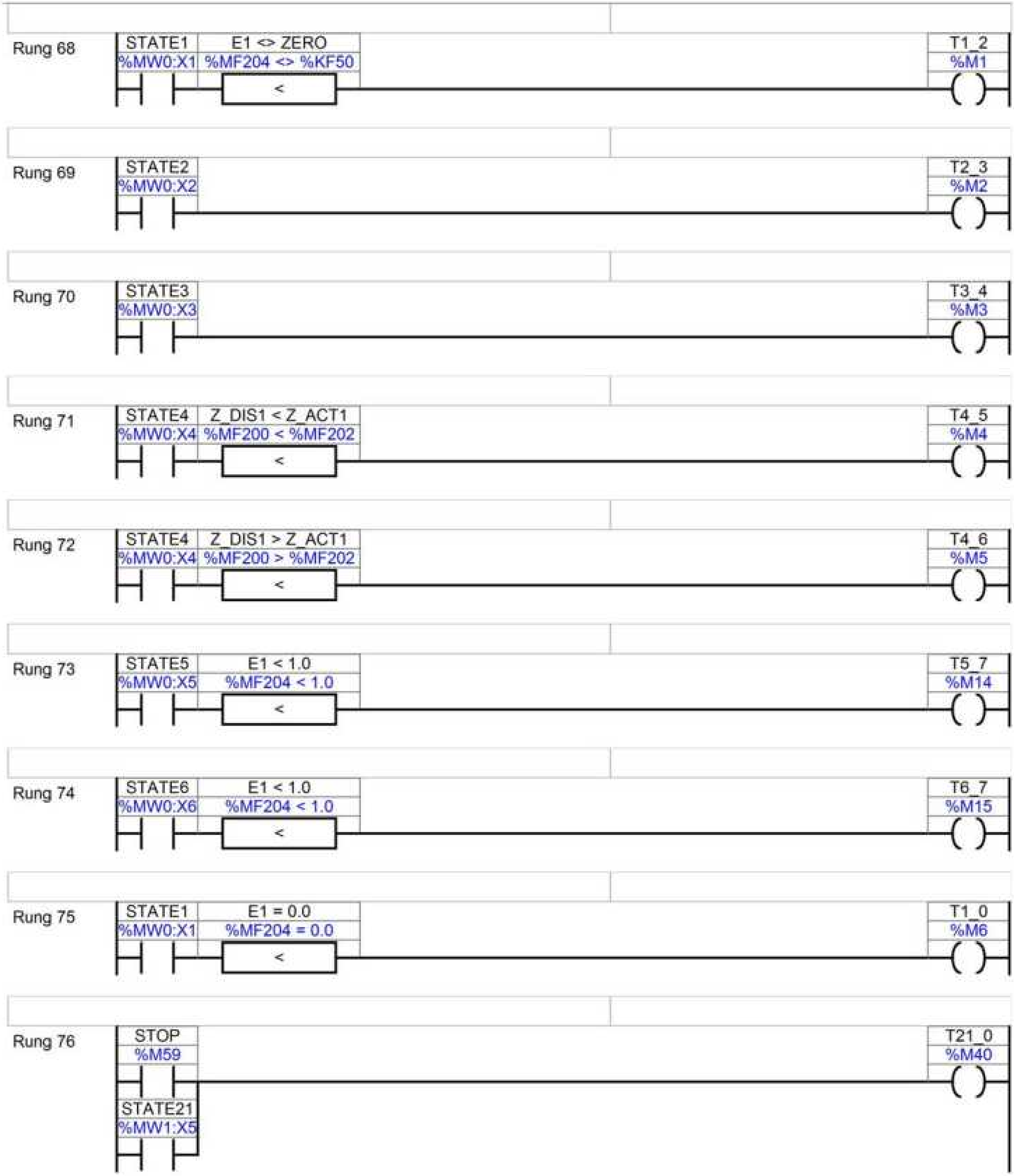
Rung 63 — ZC < D1 / %MF1804 < %KF233 (<) — %MF1218 := %MF1216 - %MF1094
%MF1218 := %MF1216 - %MF1094

Rung 64 — SHORT — Z_DIS1 := %MF2008
%MF200 := %MF2008

Rung 65 — SHORT — %MF1008 := ATAN(%MF1006)
%MF1008 := ATAN(%MF1006)

Rung 66 — SHORT — Z_DIS3 := %MF1218
%MF407 := %MF1218

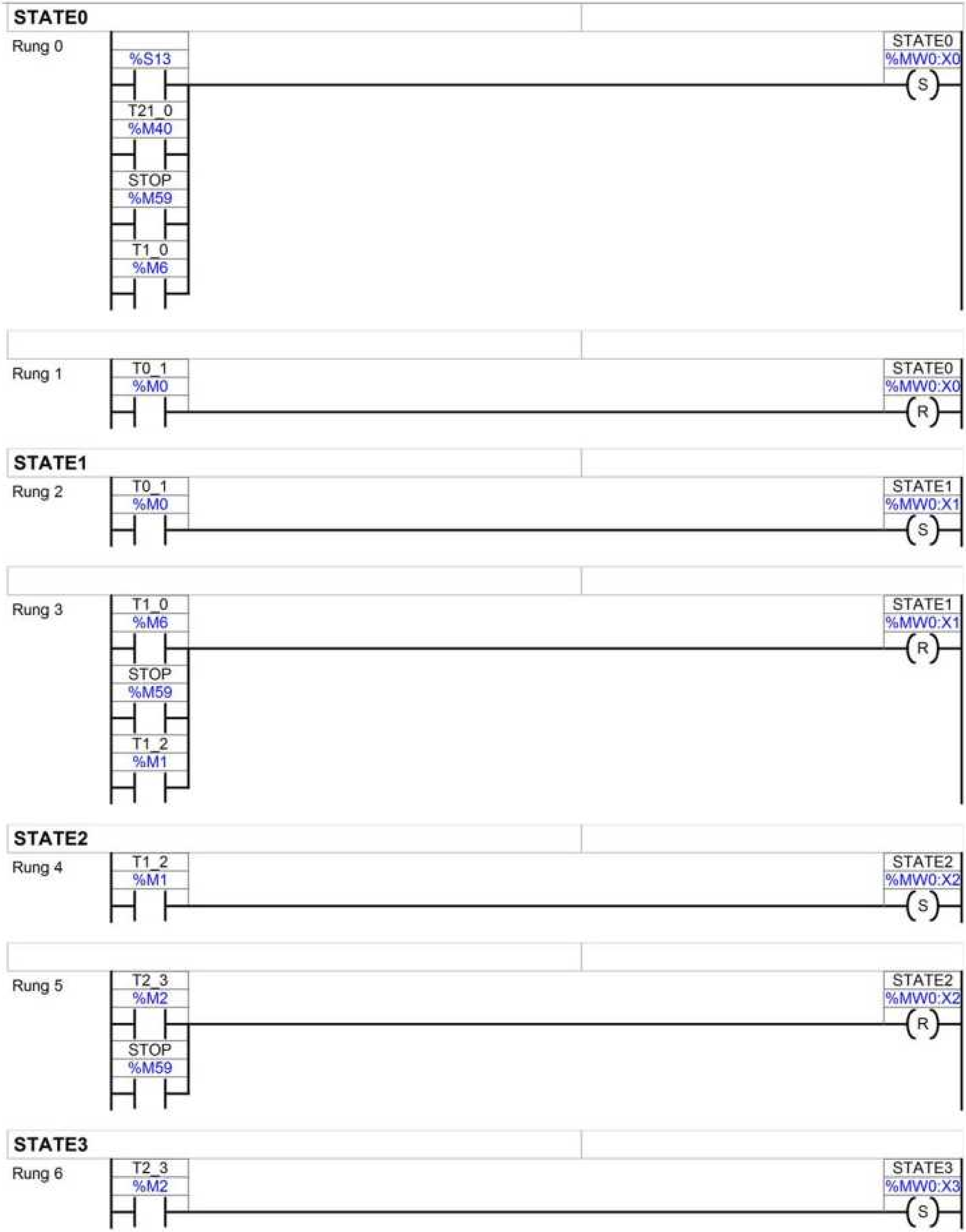
Rung 67 — STATE0 %MW0:X0 — START %M57 — ( ) T0_1 %M0


Rung 68
STATE1
%MW0:X1
E1 <> ZERO
%MF204 <> %KF50
T1_2
%M1
Rung 69
STATE2
%MW0:X2
T2_3
%M2
Rung 70
STATE3
%MW0:X3
T3_4
%M3
Rung 71
STATE4
%MW0:X4
Z_DIS1 < Z_ACT1
%MF200 < %MF202
T4_5
%M4
Rung 72
STATE4
%MW0:X4
Z_DIS1 > Z_ACT1
%MF200 > %MF202
T4_6
%M5
Rung 73
STATE5
%MW0:X5
E1 < 1.0
%MF204 < 1.0
T5_7
%M14
Rung 74
STATE6
%MW0:X6
E1 < 1.0
%MF204 < 1.0
T6_7
%M15
Rung 75
STATE1
%MW0:X1
E1 = 0.0
%MF204 = 0.0
T1_0
%M6
Rung 76
STOP
%M59
STATE21
%MW1:X5
T21_0
%M40

Rung	Contact	Comparison	Coil
Rung 77	STATE7 %MW0:X7		T7_8 %M16
Rung 78	STATE8 %MW0:X8		T8_9 %M17
Rung 79	STATE9 %MW0:X9	E2 <> 0.0 %MF203 <> 0.0 (<)	T9_10 %M18
Rung 80	STATE9 %MW0:X9	E2 = 0.0 %MF203 = 0.0 (<)	T9_8 %M41
Rung 81	STATE10 %MW0:X10		T10_11 %M19
Rung 82	STATE11 %MW0:X11	Z_DIS2 < Z_ACT2 %MF306 < %MF302 (<)	T11_12 %M20
Rung 83	STATE11 %MW0:X11	Z_DIS2 > Z_ACT2 %MF306 > %MF302 (<)	T11_13 %M21
Rung 84	STATE12 %MW0:X12	E2 < 2.0 %MF203 < 2.0 (<)	T12_14 %M22
Rung 85	STATE13 %MW0:X13	E2 < 2.0 %MF203 < 2.0 (<)	T13_14 %M23

Rung	Contact	Comparison	Coil
Rung 86	STATE14 %MW0:X14		T14_15 %M31
Rung 87	STATE15 %MW0:X15		T15_16 %M32
Rung 88	STATE16 %MW1:X0	E3 = 0.0 %MF563 = 0.0	T16_15 %M42
Rung 89	STATE16 %MW1:X0	E3 <> 0.0 %MF563 <> 0.0	T16_17 %M43
Rung 90	STATE17 %MW1:X1		T17_18 %M44
Rung 91	STATE18 %MW1:X2	Z_DIS3 < Z_ACT3 %MF407 < %MF402	T18_19 %M45
Rung 92	STATE18 %MW1:X2	Z_DIS3 > Z_ACT3 %MF407 > %MF402	T18_20 %M46
Rung 93	STATE19 %MW1:X3	E3 < 2.0 %MF563 < 2.0	T19_21 %M47
Rung 94	STATE20 %MW1:X4	E3 < 2.0 %MF563 < 2.0	T20_21 %M48

3 LD States - Sequential Function Chart [SFC]


STATE0
Rung 0
%S13
T21_0
%M40
STOP
%M59
T1_0
%M6
STATE0
%MW0:X0
S
Rung 1
T0_1
%M0
STATE0
%MW0:X0
R
STATE1
Rung 2
T0_1
%M0
STATE1
%MW0:X1
S
Rung 3
T1_0
%M6
STOP
%M59
T1_2
%M1
STATE1
%MW0:X1
R
STATE2
Rung 4
T1_2
%M1
STATE2
%MW0:X2
S
Rung 5
T2_3
%M2
STOP
%M59
STATE2
%MW0:X2
R
STATE3
Rung 6
T2_3
%M2
STATE3
%MW0:X3
S

Rung 7
T3_4 %M3
STOP %M59
STATE3 %MW0:X3 (R)

STATE4 [CHICK THIS] 1

Rung 8
T3_4 %M3
STATE4 %MW0:X4 (S)

Rung 9
T4_5 %M4
T4_6 %M5
STOP %M59
STATE4 %MW0:X4 (R)

STATE5

Rung 10
T4_5 %M4
STATE5 %MW0:X5 (S)

Rung 11
T5_7 %M14
STOP %M59
STATE5 %MW0:X5 (R)

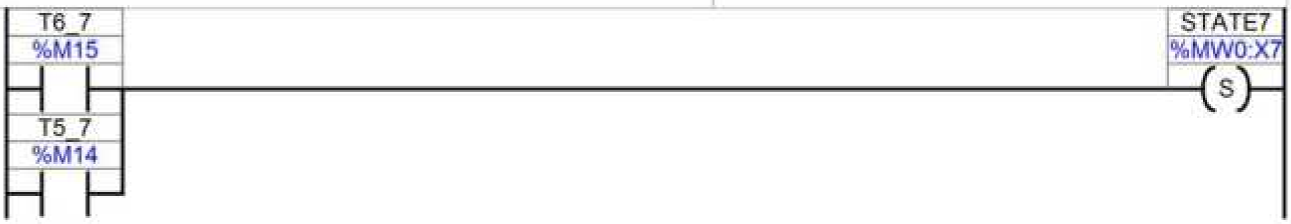
STATE6

Rung 12
T4_6 %M5
STATE6 %MW0:X6 (S)

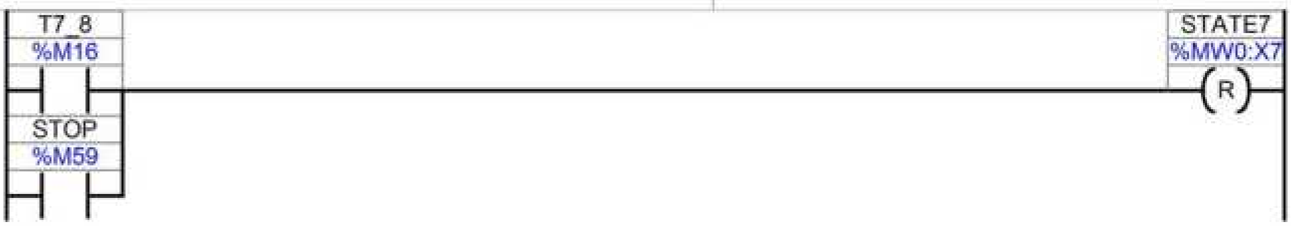
Rung 13
STOP %M59
T6_7 %M15
STATE6 %MW0:X6 (R)

STATE7

Rung 14



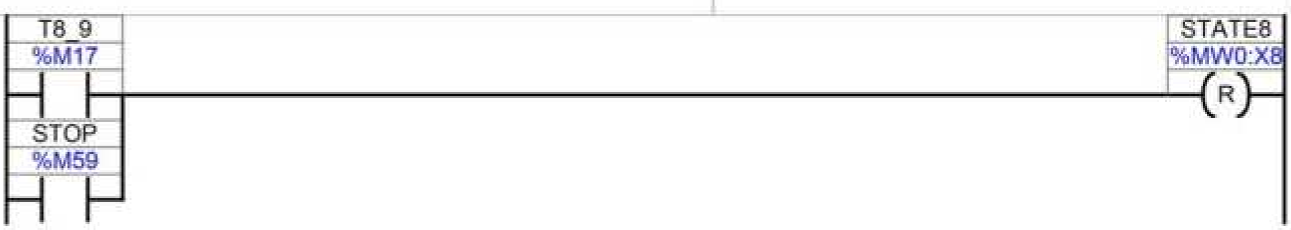
Rung 15



STATE8

Rung 16

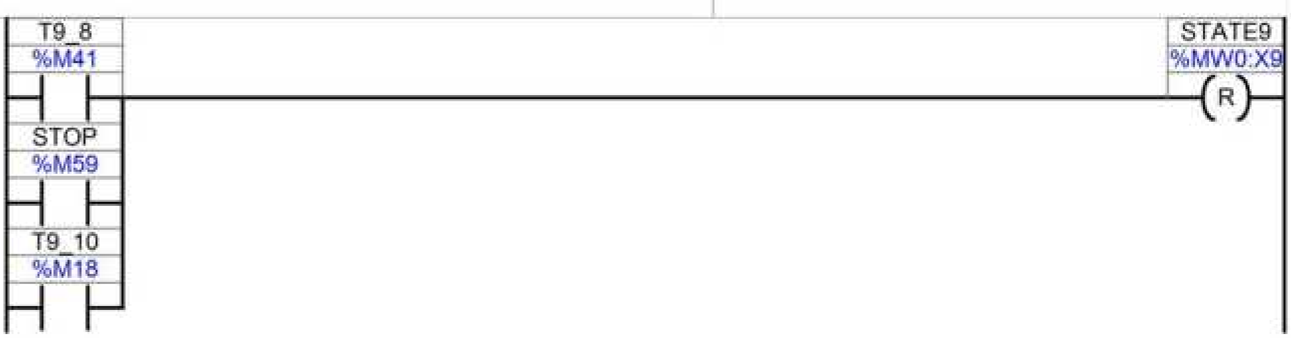
Rung 17



STATE9

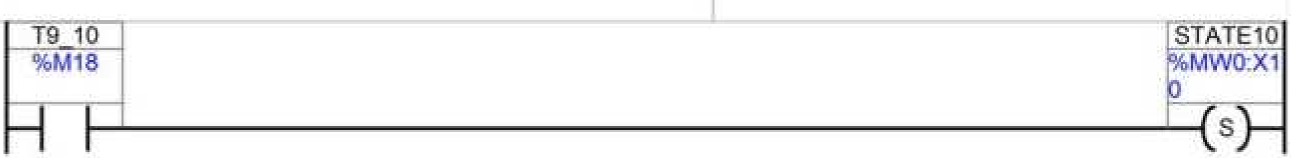
Rung 18

Rung 19

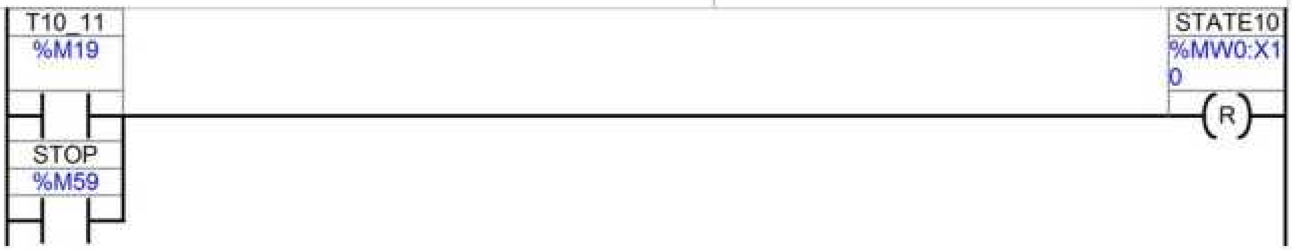


STATE10

Rung 20



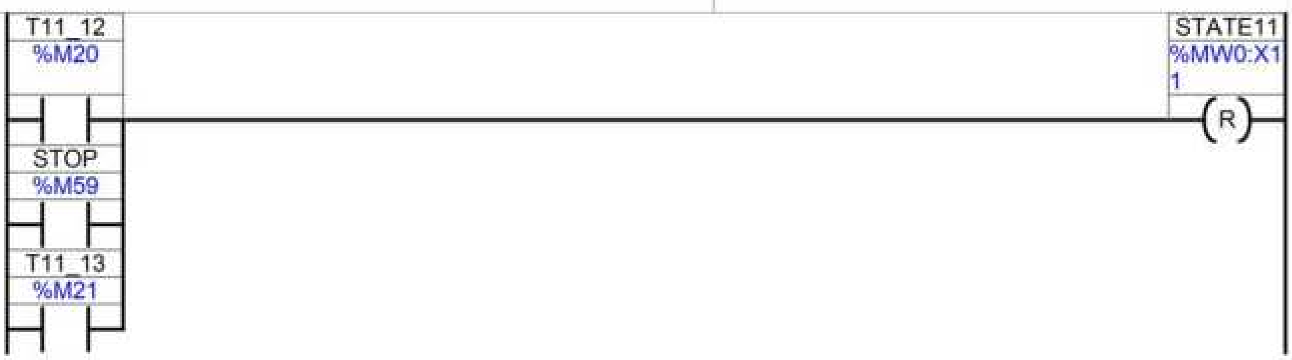
Rung 21



STATE11 [CHECK THIS] 2

Rung 22

Rung 23



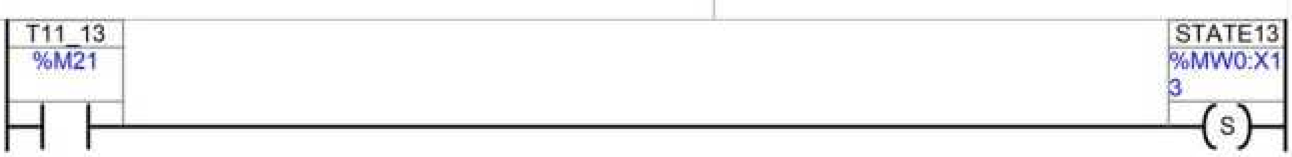
STATE12

Rung 24

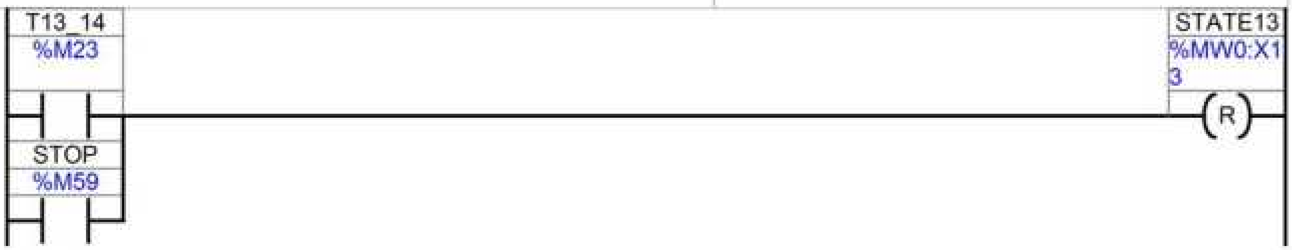
Rung 25

STATE13

Rung 26

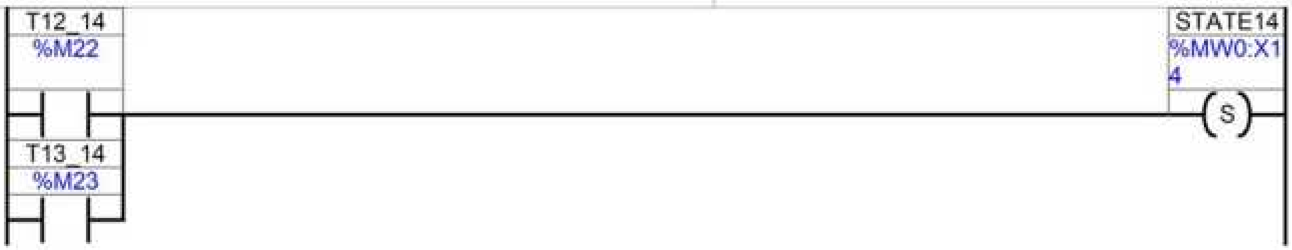


Rung 27

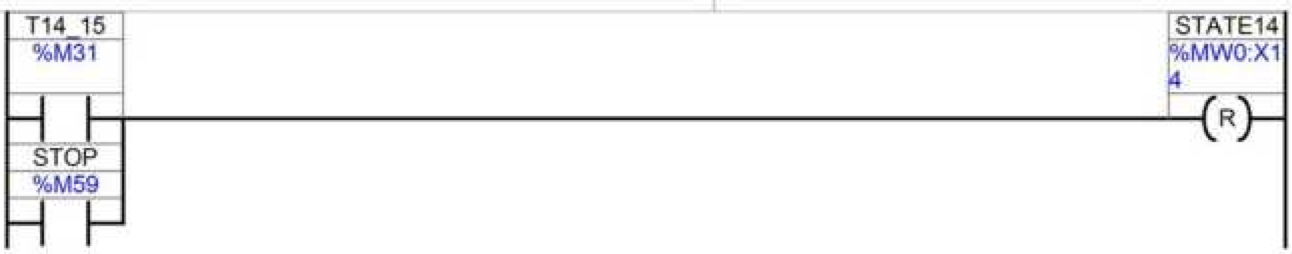


STATE14

Rung 28

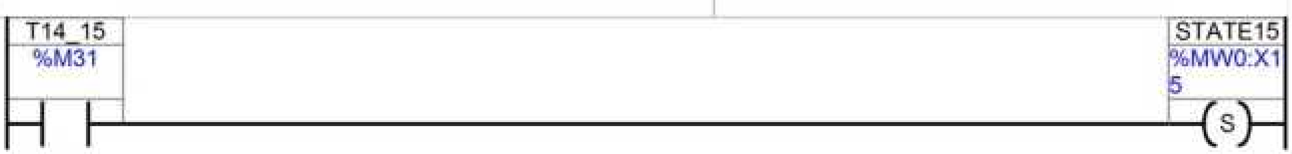


Rung 29



STATE15

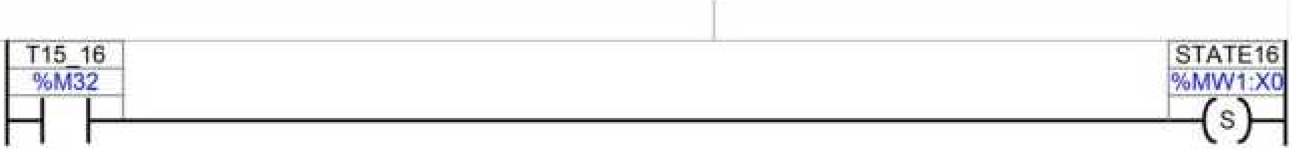
Rung 30



Rung 31

STATE16

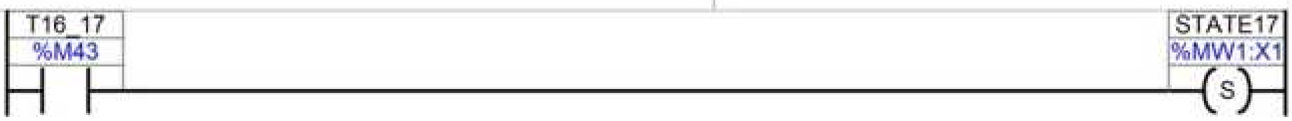
Rung 32



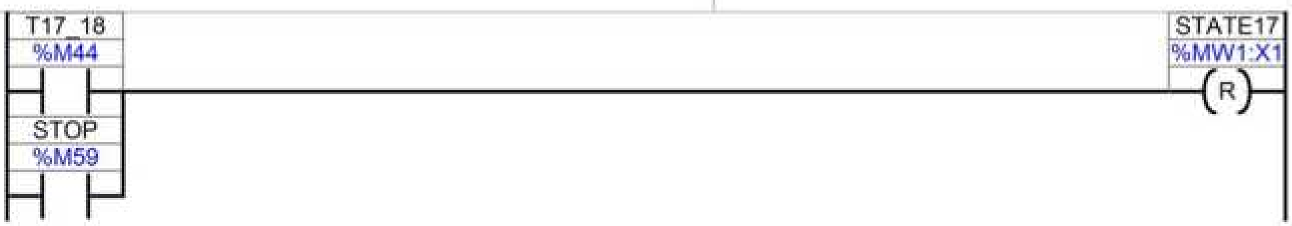
Rung 33

STATE17

Rung 34



Rung 35



STATE18 [CHECK THIS] 3

Rung 36

STATE18

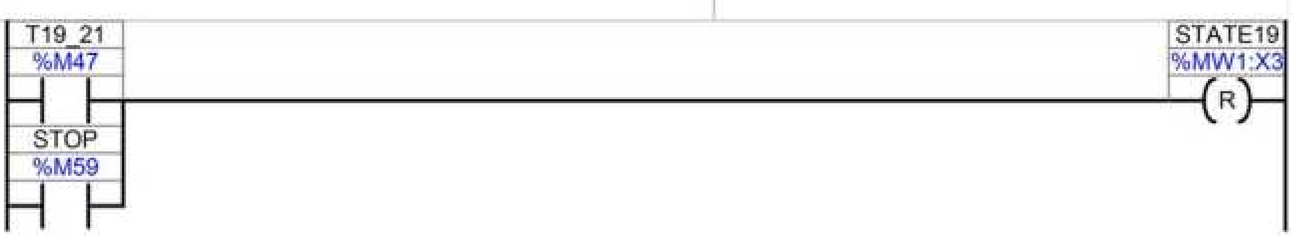
Rung 37

STATE19

Rung 38

STATE19

Rung 39



STATE20

Rung 40

T18_20 %M46

STATE20 %MW1:X4 (S)

STATE20

Rung 41

T20_21 %M48

STOP %M59

STATE20 %MW1:X4 (R)

STATE21

Rung 42

T19_21 %M47

T20_21 %M48

STATE21 %MW1:X5 (S)

Rung 43

T21_0 %M40

STATE21 %MW1:X5 (R)

4 LD

Rung 0

STATE1 %MW0:X1

STATE2 %MW0:X2

STATE7 %MW0:X7

STATE6 %MW0:X6

STATE5 %MW0:X5

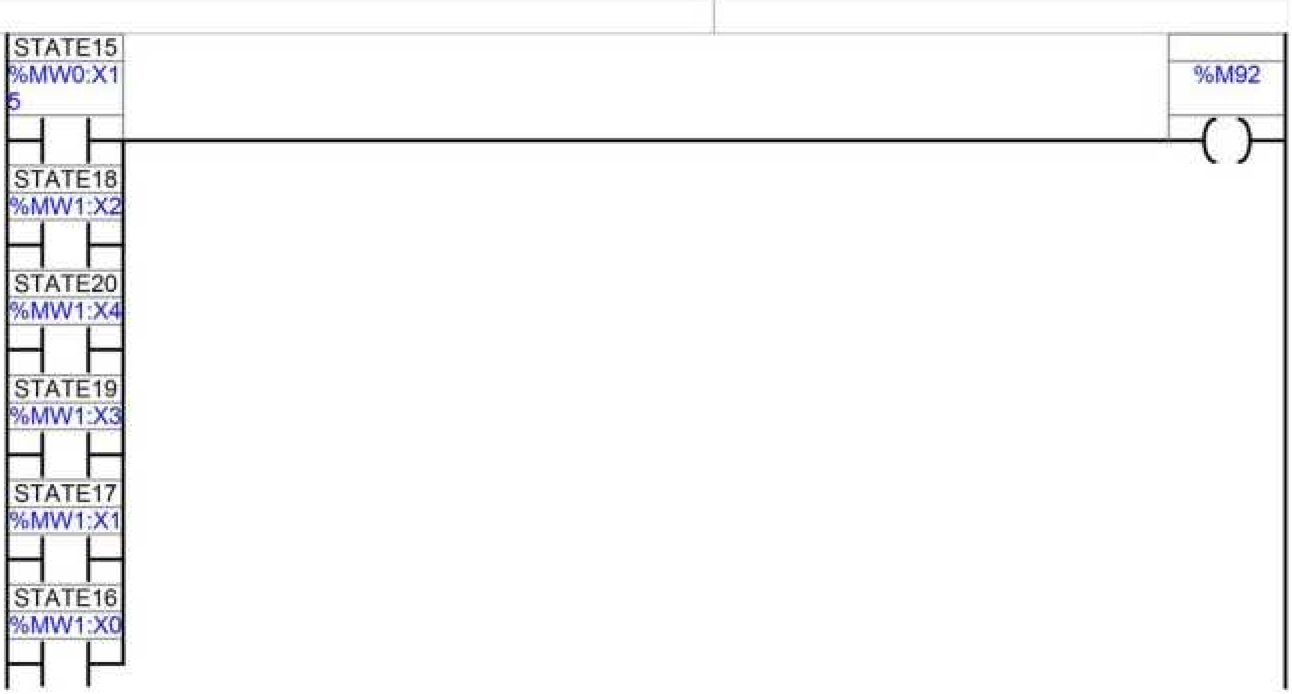
STATE4 %MW0:X4

STATE3 %MW0:X3

%M90 ( )

Rung 1

Rung 2



5 LD	Calculation of the first Joint	

VOLTAGE IN CHANAL 0

Rung 0
%M90
%MW350 := %IW0.1.0
%MW350 := %IW0.1.0

Rung 1
%M90
%MF600 := INT_TO_REAL(%MW350)
%MF600 := INT_TO_REAL(%MW350)

VOLTAGE IN CHANAL 0 [FINAL VALUE]

VOLTAGE IN FOE CHANEL 0

Rung 2
%M90
VOLTAGE_IN_CH0 := %MF600 * C1
%MF650 := %MF600 * %KF60

ANGLE 1 [Z_ACT 1] CALCULATION

Rung 3
%M90
V2 := VOLTAGE_IN_CH0 * -34.152
%MF212 := %MF650 * -34.152

Rung 4
%M90
V3 := V2 + 194.0
%MF214 := %MF212 + 194.0

ANGLE 1 [FINAL VALUE]

Rung 5
%M90
Z_ACT1 := V3
%MF202 := %MF214

Rung 6
%M90
Z1_ACT := REAL_TO_INT(Z_ACT1)
%MW20 := REAL_TO_INT(%MF202)

DISERED ANGLE OF JOINT 1

Rung 7
%M90
Z1_DIS := REAL_TO_INT(Z_DIS1)
%MW22 := REAL_TO_INT(%MF200)

ERORE OF THE ANGLE

Rung 8
%M90
%MF252 := Z_DIS1 - Z_ACT1
%MF252 := %MF200 - %MF202

TO THE ABSOLUTE VALUE OF THE ERRORE

Rung 9
%M90
E1 := ABS(%MF252)
%MF204 := ABS(%MF252)

ANALOG OUTPUT CALCULATION [PI OUTPUT]

Rung 10

%M90

%MW333 := AO
%MW333 := %QW0.1.0

Rung 11

%M90

%MF623 := INT_TO_REAL(%MW333)
%MF623 := INT_TO_REAL(%MW333)

ANALOG OUTPUT [FINAL VALUE]

Rung 12

%M90

VOLTAGE__OUT_CH0 := %MF623 * C1
%MF777 := %MF623 * %KF60

6 LD Seconed Joint Calculations

VOLATGE IN CALCULATION :

Rung 0

%M91

%MW100 := POT2
%MW100 := %IW0.1.1

Rung 1

%M91

%MF206 := INT_TO_REAL(%MW100)
%MF206 := INT_TO_REAL(%MW100)

VOLTAGE IN CHANAL 1 :[FINAL VALUE]

Rung 2

%M91

VOLTAGE_IN_CHANAL_1 := %MF206 * C1
%MF208 := %MF206 * %KF60

ANGLE 2 [Z_ACT2] CALCULATION FROM VOLTAGE IN 1

Rung 3

%M91

%MF300 := 12.238 * VOLTAGE_IN_CHANAL_1
%MF300 := 12.238 * %MF208

Z_ACT2 [FINAL VALUE]

Rung 4

%M91

Z_ACT2 := %MF300 - 43.167
%MF302 := %MF300 - 43.167

Z2_ACT [FINAL VALUE]

Rung 5

%M91

Z2_ACT := REAL_TO_INT(Z_ACT2)
%MW90 := REAL_TO_INT(%MF302)

Z2_DIS [FINAL VALUE]

Rung 6 — %M91 — Z2_DIS := REAL_TO_INT(Z_DIS2)
%MW30 := REAL_TO_INT(%MF306)

ERRORE OF THE SECONED ANGLE

Rung 7 — %M91 — E2_OO := Z_DIS2 - Z_ACT2
%MF290 := %MF306 - %MF302

Rung 8 — %M91 — E2 := ABS(E2_OO)
%MF203 := ABS(%MF290)

7 LD Thired Joint Calculations

VOLTAGE IN CHANAL 2 CALCLATION

Rung 0 — %M92 — %MW160 := POT3
%MW160 := %IW0.1.2

Rung 1 — %M92 — %MF315 := INT_TO_REAL(%MW160)
%MF315 := INT_TO_REAL(%MW160)

VOLTAGE IN CHANAL 2 [FINAL VALUE]

Rung 2 — %M92 — VOLTAGE_IN_CH2 := %MF315 * C1
%MF317 := %MF315 * %KF60

ACTUAL ANGLE 3 CALUALATION

Rung 3 — %M92 — %MF320 := VOLTAGE_IN_CH2 * -10.564
%MF320 := %MF317 * -10.564

ACTUAL ANGLE 3 [FINAL VALUE]

Rung 4 — %M92 — %MF127 := %MF320 - 30.77
%MF127 := %MF320 - 30.77

Rung 5 — %M92 — Z_ACT3 := ABS(%MF127)
%MF402 := ABS(%MF127)

Rung 6 — %M92 — Z3_ACT := REAL_TO_INT(Z_ACT3)
%MW39 := REAL_TO_INT(%MF402)

DISIRED ACTUAL (ABSOLUTE VALUE)

Rung 7

%M92

%MF412 := ABS(Z_DIS3)
%MF412 := ABS(%MF407)

DISERED ANGLE 3 [FINAL VALUE]

Rung 8

%M92

Z3_DIS := REAL_TO_INT(%MF412)
%MW702 := REAL_TO_INT(%MF412)

ERROR OF THE THIRED ANGLE

Rung 9

%M92

E3_OO := %MF412 - Z_ACT3
%MF566 := %MF412 - %MF402

ABSOLUTE VALUE OF THE ERROR

Rung 10

%M92

E3 := ABS(E3_OO)
%MF563 := ABS(%MF566)

8 LD PID Output Selection

Rung 0

STATE3
%MW0:X3

STATE2
%MW0:X2

STATE1
%MW0:X1

STATE5
%MW0:X5

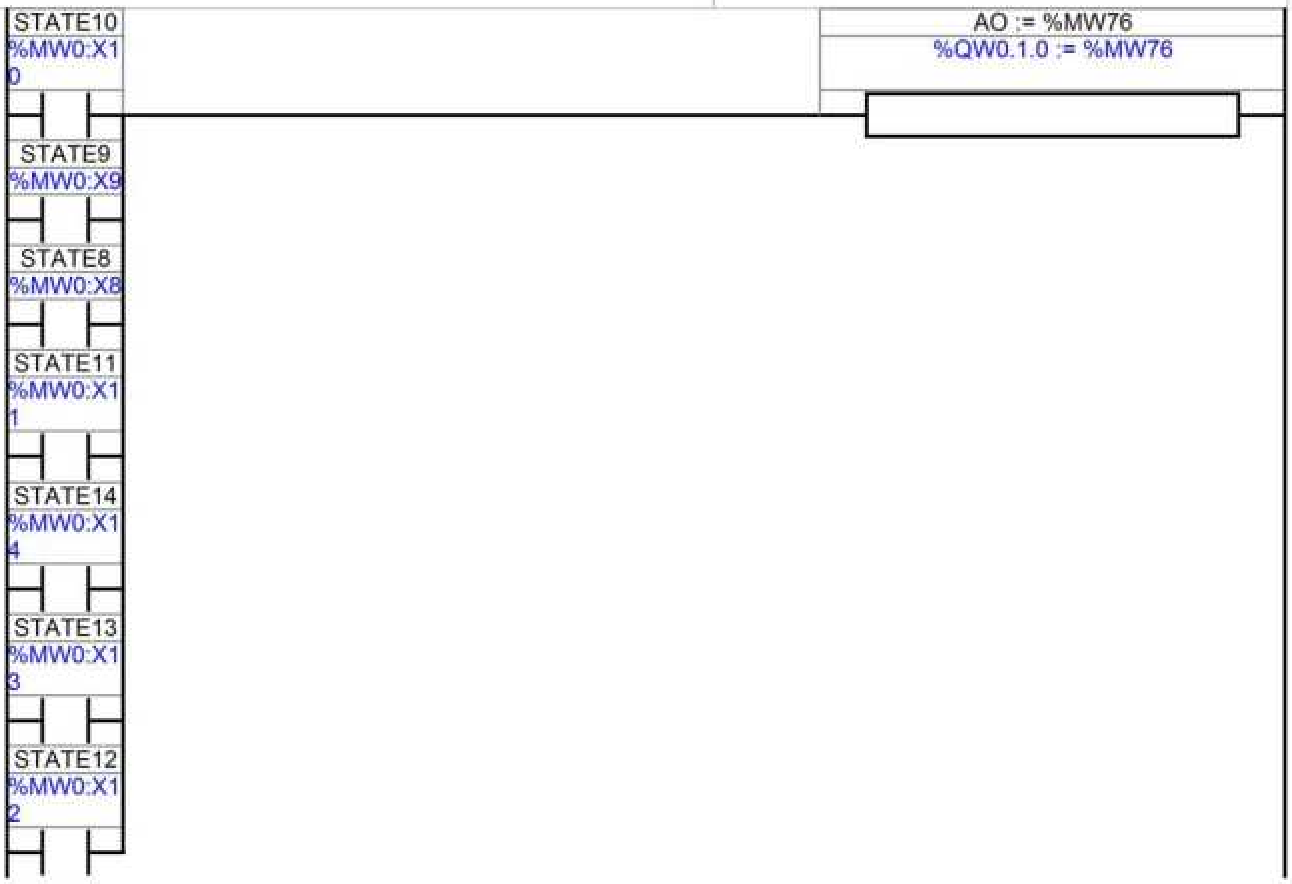
STATE4
%MW0:X4

STATE7
%MW0:X7

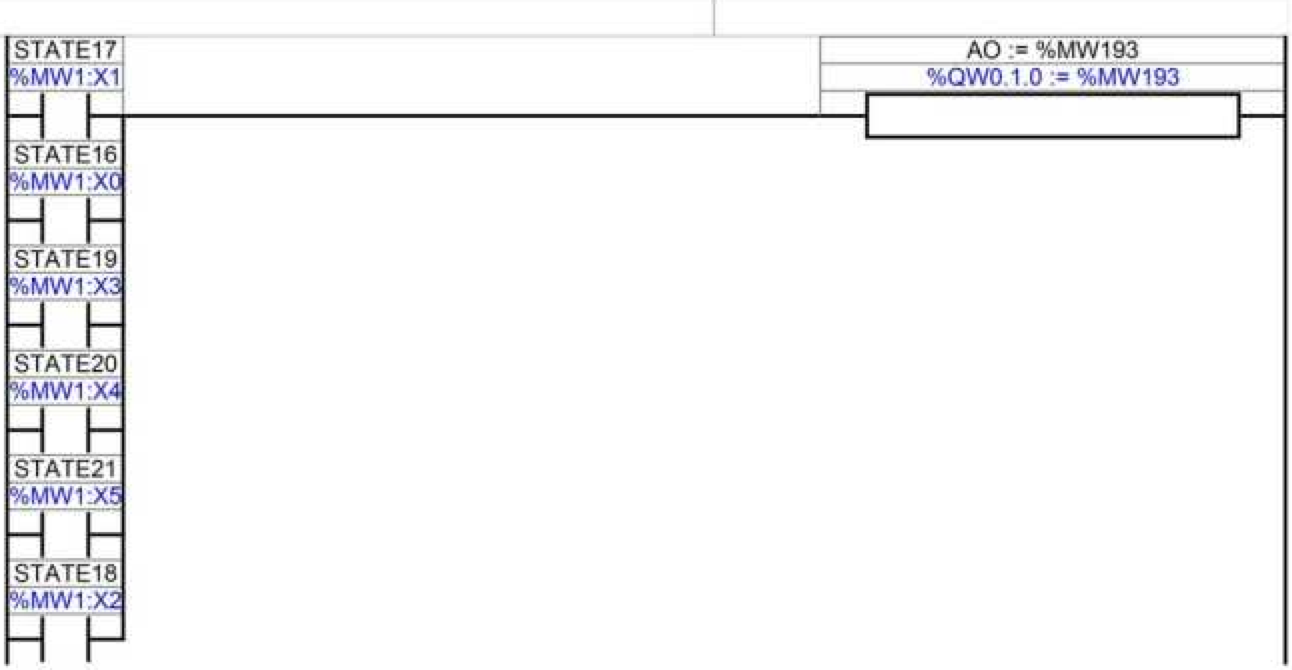
STATE6
%MW0:X6

AO := %MW88
%QW0.1.0 := %MW88

Rung 1



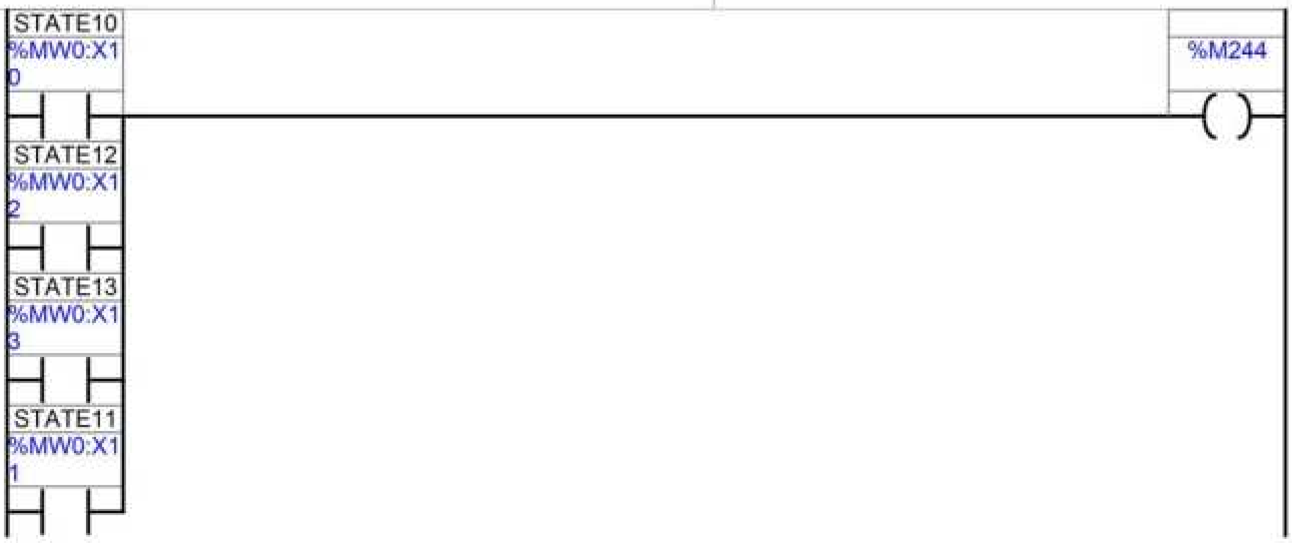
Rung 2



9 LD Actions - Sequential Function Chart [SFC]

Rung 0

Rung 1



Rung 2

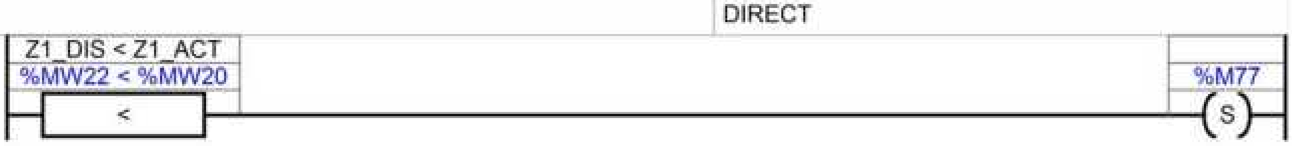
Rung 3

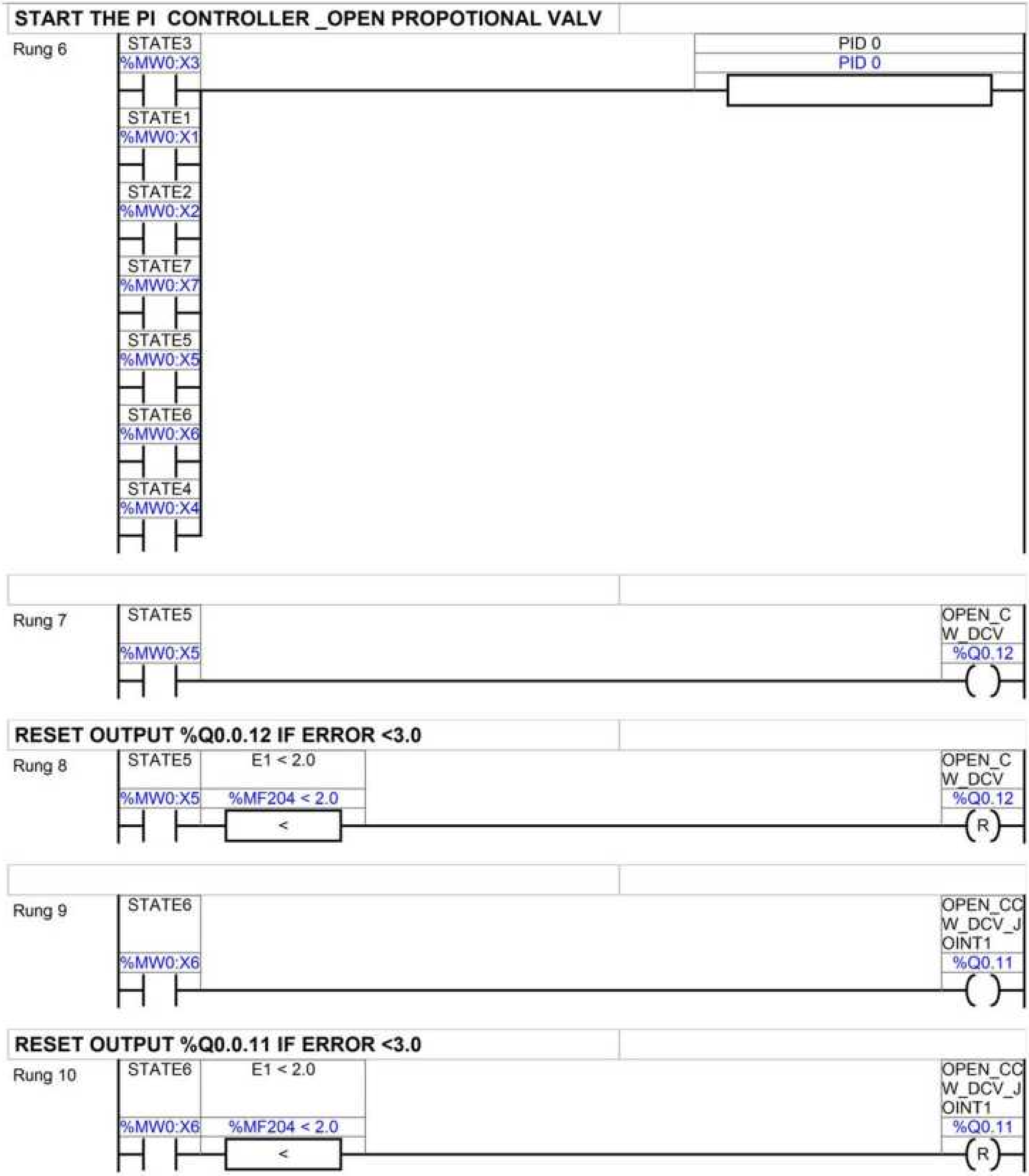
FOR DIRECT AND REVERSE OF PID0 | REVERSE

Rung 4

DIRECT | DIRECT

Rung 5




START THE PI CONTROLLER _OPEN PROPOTIONAL VALV
Rung 6
STATE3
%MW0:X3
STATE1
%MW0:X1
STATE2
%MW0:X2
STATE7
%MW0:X7
STATE5
%MW0:X5
STATE6
%MW0:X6
STATE4
%MW0:X4
PID 0
PID 0
Rung 7
STATE5
%MW0:X5
OPEN_CW_DCV
%Q0.12
RESET OUTPUT %Q0.0.12 IF ERROR <3.0
Rung 8
STATE5
%MW0:X5
E1 < 2.0
%MF204 < 2.0
<
OPEN_CW_DCV
%Q0.12
R
Rung 9
STATE6
%MW0:X6
OPEN_CCW_DCV_JOINT1
%Q0.11
RESET OUTPUT %Q0.0.11 IF ERROR <3.0
Rung 10
STATE6
%MW0:X6
E1 < 2.0
%MF204 < 2.0
<
OPEN_CCW_DCV_JOINT1
%Q0.11
R

Rung 11
STATE7
%MW0:X7
ANGLE1_REACHED
%M100

Rung 12
STATE8
%MW0:X8
E2 := ABS(E2_OO)
%MF203 := ABS(%MF290)

Rung 13
Z2_DIS < Z2_ACT
%MW30 < %MW90
<
%M88
S

FOR DIRECT AND REVERSE OF PID 1

Rung 14
Z2_DIS > Z2_ACT
%MW30 > %MW90
<
%M88
R

Rung 15
STATE10
%MW0:X10
STATE12
%MW0:X12
STATE11
%MW0:X11
STATE9
%MW0:X9
STATE8
%MW0:X8
STATE14
%MW0:X14
STATE13
%MW0:X13
PID 1
PID 1

Rung 16

STATE12 %MW0:X12 — OPEN_CCW_DCV_JOINT2 %Q0.6 ( )

RESET OUTPUT %Q0.0.6 IF ERRORE <3.0

Rung 17

STATE12 %MW0:X12 — E2 < 2.0 %MF203 < 2.0 [<] — OPEN_CCW_DCV_JOINT2 %Q0.6 (R)

Rung 18

STATE13 %MW0:X13 — OPEN_CW_DCV_JOINT2 %Q0.7 ( )

RESET OUTPUT %Q0.0.7 IF ERROR <3.0

Rung 19

STATE13 %MW0:X13 — E2 < 2.0 %MF203 < 2.0 [<] — OPEN_CW_DCV_JOINT2 %Q0.7 (R)

Rung 20

STATE14 %MW0:X14 — ANGLE2_REACHED %M101 ( )

Rung 21

STATE15 %MW0:X15 — E3 := ABS(E3_OO) %MF563 := ABS(%MF566)

[%MF412] IS THE ABSOLUTE VALUE OF Z_DIS3

Rung 22

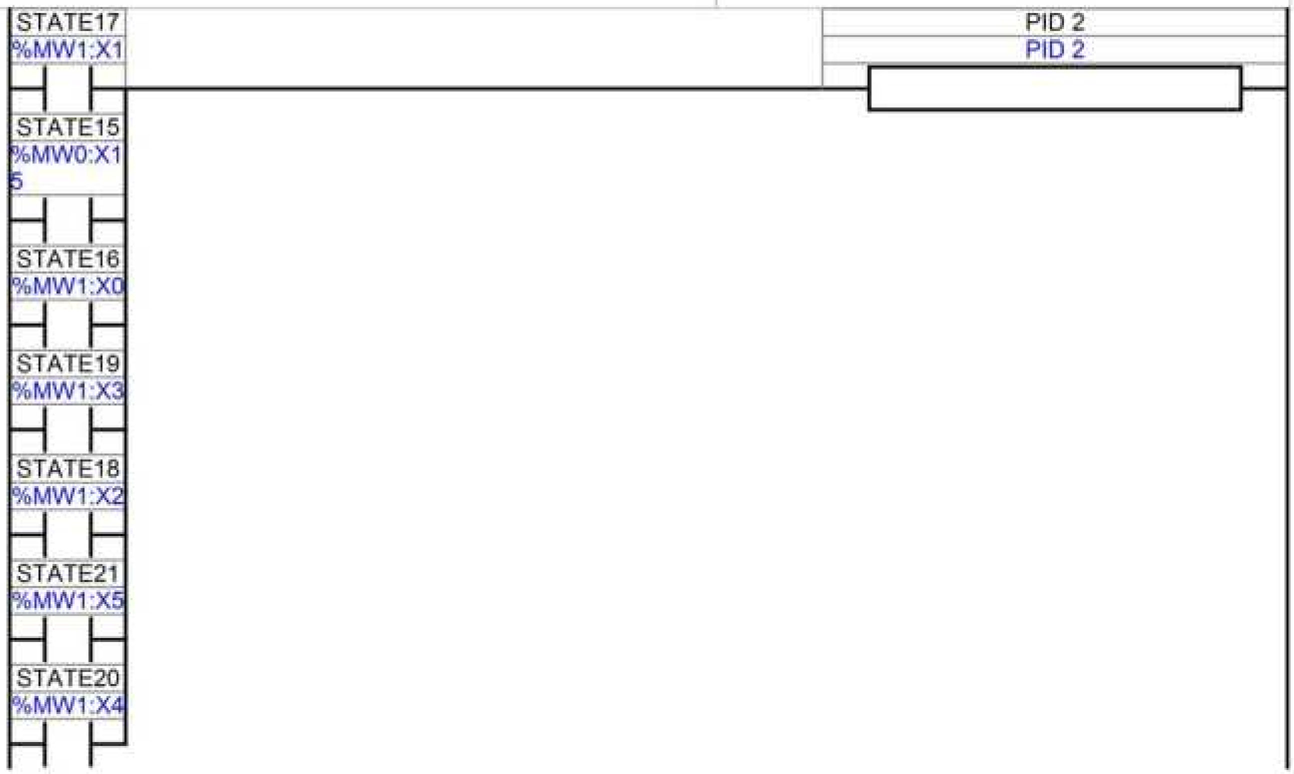
%MF412 < Z_ACT3 %MF412 < %MF402 [<] — %M99 (S)

FOR DIRECT AND REVERSE OF PID2

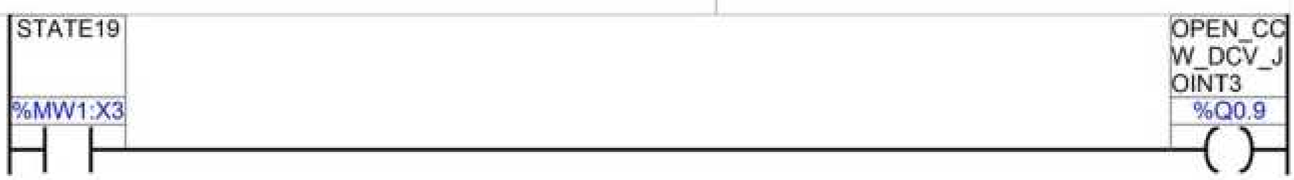
Rung 23

%MF412 > Z_ACT3 %MF412 > %MF402 [<] — %M99 (R)

Rung 24



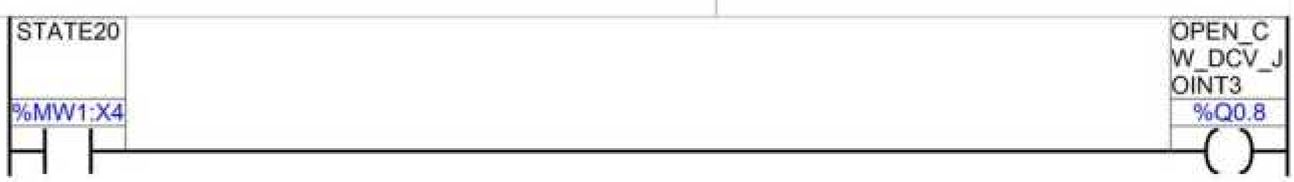
Rung 25



RESET OUTPUT %Q0.0.0 IF ERRPR <3.0

Rung 26

Rung 27



RESET OUTPUT %Q0.0.9 IF ERROR < 3.0

Rung 28

Symbols

Used	Address	Symbol	Comment
Yes	%KF240	A2	
Yes	%KF244	A3	
Yes	%M100	ANGLE1_REACHED	
Yes	%M101	ANGLE2_REACHED	
Yes	%QW1.0	AO	
Yes	%KF60	C1	
No	%KF70	C2	
No	%KF80	C3	
Yes	%KF233	D1	
Yes	%MF204	E1	
No	%MW95	E1_ABS	
Yes	%MF203	E2	
No	%MW320	E2_ABSS	
Yes	%MF290	E2_OO	
Yes	%MF563	E3	
Yes	%MF566	E3_OO	
Yes	%KF90	G_L	
No	%MW371	KI0	
No	%MW311	KI1	
No	%MW18	KI2	
No	%MW370	KP0	
No	%MW310	KP1	
No	%MW17	KP2	
Yes	%Q0.11	OPEN_CCW_DCV_JOINT1	
Yes	%Q0.6	OPEN_CCW_DCV_JOINT2	
Yes	%Q0.9	OPEN_CCW_DCV_JOINT3	
Yes	%Q0.12	OPEN_CW_DCV	
Yes	%Q0.7	OPEN_CW_DCV_JOINT2	
Yes	%Q0.8	OPEN_CW_DCV_JOINT3	
Yes	%Q0.10	OPEN_MAIN_VALVE	
Yes	%IW1.1	POT2	
Yes	%IW1.2	POT3	
No	%IW1.3	POT4	
Yes	%M57	START	
Yes	%MW0:X0	STATE0	
Yes	%MW0:X1	STATE1	
Yes	%MW0:X2	STATE2	
Yes	%MW0:X3	STATE3	
Yes	%MW0:X4	STATE4	
Yes	%MW0:X5	STATE5	
Yes	%MW0:X6	STATE6	
Yes	%MW0:X7	STATE7	
Yes	%MW0:X8	STATE8	
Yes	%MW0:X9	STATE9	
Yes	%MW0:X10	STATE10	
Yes	%MW0:X11	STATE11	
Yes	%MW0:X12	STATE12	
Yes	%MW0:X13	STATE13	
Yes	%MW0:X14	STATE14	
Yes	%MW0:X15	STATE15	
Yes	%MW1:X0	STATE16	
Yes	%MW1:X1	STATE17	
Yes	%MW1:X2	STATE18	
Yes	%MW1:X3	STATE19	
Yes	%MW1:X4	STATE20	

Used	Address	Symbol	Comment
Yes	%MW1:X5	STATE21	
Yes	%M59	STOP	
Yes	%M0	T0_1	
No	%M11	T1	
Yes	%M6	T1_0	
Yes	%M1	T1_2	
No	%M7	T2_0	
Yes	%M2	T2_3	
No	%M8	T3_0	
Yes	%M3	T3_4	
No	%M9	T4_0	
Yes	%M4	T4_5	
Yes	%M5	T4_6	
No	%M10	T5_0	
No	%M12	T5_4	
Yes	%M14	T5_7	
No	%M13	T6_4	
Yes	%M15	T6_7	
No	%M24	T7_0	
Yes	%M16	T7_8	
No	%M25	T8_0	
Yes	%M17	T8_9	
No	%M26	T9_0	
Yes	%M41	T9_8	
Yes	%M18	T9_10	
No	%M27	T10_0	
Yes	%M19	T10_11	
No	%M28	T11_0	
Yes	%M20	T11_12	
Yes	%M21	T11_13	
No	%M29	T12_0	
No	%M50	T12_11	
Yes	%M22	T12_14	
No	%M30	T13_0	
No	%M49	T13_11	
Yes	%M23	T13_14	
No	%M33	T14_0	
Yes	%M31	T14_15	
No	%M34	T15_0	
Yes	%M32	T15_16	
No	%M35	T16_0	
Yes	%M42	T16_15	
Yes	%M43	T16_17	
No	%M36	T17_0	
Yes	%M44	T17_18	
No	%M37	T18_0	
Yes	%M45	T18_19	
Yes	%M46	T18_20	
No	%M38	T19_0	
No	%M52	T19_18	
Yes	%M47	T19_21	
No	%M39	T20_0	
No	%M51	T20_18	
Yes	%M48	T20_21	
Yes	%M40	T21_0	
Yes	%MF2013	THETA2	
Yes	%MF2025	THETA3	
No	%MF210	V1	
Yes	%MF212	V2	
Yes	%MF214	V3	

Used	Address	Symbol	Comment
Yes	%MF650	VOLTAGE_IN_CH0	
Yes	%MF317	VOLTAGE_IN_CH2	
Yes	%MF208	VOLTAGE_IN_CHANAL_1	
Yes	%MF777	VOLTAGE__OUT_CH0	
Yes	%MF130	X0	
Yes	%MF420	X1	
Yes	%MF1800	XC	
Yes	%MF132	Y0	
Yes	%MF422	Y1	
Yes	%MF1802	YC	
Yes	%MF134	Z0	
Yes	%MF424	Z1	
Yes	%MW20	Z1_ACT	
Yes	%MW22	Z1_DIS	
No	%MW2	Z2	
Yes	%MW90	Z2_ACT	
Yes	%MW30	Z2_DIS	
Yes	%MW39	Z3_ACT	
Yes	%MW702	Z3_DIS	
Yes	%MF1804	ZC	
Yes	%KF50	ZERO	
Yes	%MF202	Z_ACT1	
Yes	%MF302	Z_ACT2	
Yes	%MF402	Z_ACT3	
Yes	%MF200	Z_DIS1	
Yes	%MF306	Z_DIS2	
Yes	%MF407	Z_DIS3	

Cross references

Address	Symbol	Section	Lines/Networks	Operator
%KF90	G_L	1	5	[:= *]
	G_L	1	6	[:= *]
%MF132	Y0	1	1	[:= +]
%MF134	Z0	1	2	[:=]
	Z0	1	9	[:=]
%MF420	X1	1	7	[:= +]
%MF422	Y1	1	1	[:= +]
	Y1	1	8	[:= -]
%MF424	Z1	1	2	[:=]
%MF430		1	3	[:= COS]
		1	5	[:= *]
%MF432		1	4	[:= SIN]
		1	6	[:= *]
%MF434		1	5	[:= *]
		1	7	[:= +]
%MF436		1	6	[:= *]
		1	8	[:= -]
%MF1800	XC	1	7	[:= +]
%MF1802	YC	1	8	[:= -]
%MF1804	ZC	1	9	[:=]
%MF2008		1	3	[:= COS]
		1	4	[:= SIN]
1		1	1	LD
		1	2	LD
		1	3	LD
		1	4	LD
		1	5	LD
		1	6	LD
		1	7	LD
		1	8	LD
		1	9	LD
9.3		1	1	[:= +]

Animation table

Active states

Used	Address	Symbol	Units
Yes	%MW0:X0	STATE0	Decimal
Yes	%MW0:X1	STATE1	Decimal
Yes	%MW0:X2	STATE2	Decimal
Yes	%MW0:X3	STATE3	Decimal
Yes	%MW0:X4	STATE4	Decimal
Yes	%MW0:X5	STATE5	Decimal
Yes	%MW0:X6	STATE6	Decimal
Yes	%MW0:X7	STATE7	Decimal
Yes	%MW0:X8	STATE8	Decimal
Yes	%MW0:X9	STATE9	Decimal
Yes	%MW0:X10	STATE10	Decimal
Yes	%MW0:X11	STATE11	Decimal
Yes	%MW0:X12	STATE12	Decimal
Yes	%MW0:X13	STATE13	Decimal
Yes	%MW0:X14	STATE14	Decimal
Yes	%MW0:X15	STATE15	Decimal
Yes	%MW1:X1	STATE17	Decimal
Yes	%MW1:X2	STATE18	Decimal
Yes	%MW1:X3	STATE19	Decimal
Yes	%MW1:X4	STATE20	Decimal
Yes	%MW1:X5	STATE21	Decimal

Input

Used	Address	Symbol	Units
Yes	%M57	START	Decimal
Yes	%M59	STOP	Decimal
Yes	%MF130	X0	Floating Point
Yes	%MF132	Y0	Floating Point
Yes	%MF134	Z0	Floating Point
Yes	%MF200	Z_DIS1	Floating Point
Yes	%MF306	Z_DIS2	Floating Point
Yes	%MF407	Z_DIS3	Floating Point

Output for the first joint

Used	Address	Symbol	Units
Yes	%Q0.0.11	OPEN_CCW_DCV_JOINT1	Decimal
Yes	%Q0.0.12	OPEN_CW_DCV	Decimal
Yes	%Q0.0.10	OPEN_MAIN_VALVE	Decimal

Output for the Seconed Joint

Used	Address	Symbol	Units
Yes	%Q0.0.10	OPEN_MAIN_VALVE	Decimal
Yes	%Q0.0.6	OPEN_CCW_DCV_JOINT2	Decimal
Yes	%Q0.0.7	OPEN_CW_DCV_JOINT2	Decimal

Output for the Thired Joint

Used	Address	Symbol	Units
Yes	%Q0.0.10	OPEN_MAIN_VALVE	Decimal
Yes	%Q0.0.7	OPEN_CW_DCV_JOINT2	Decimal
Yes	%Q0.0.8	OPEN_CW_DCV_JOINT3	Decimal

PID 0

Used	Address	Symbol	Units
Yes	%MF200	Z_DIS1	Floating Point
Yes	%MW22	Z1_DIS	Decimal

Yes	%MF202	Z_ACT1	Floating Point
Yes	%MW20	Z1_ACT	Decimal
Yes	%QW0.1.0	AO	Decimal
Yes	%MF777	VOLTAGE__OUT_CH0	Floating Point
Yes	%MF204	E1	Floating Point
Yes	%MF650	VOLTAGE_IN_CH0	Floating Point
Yes	%M77		Decimal

PID 1

Used	Address	Symbol	Units
Yes	%MF306	Z_DIS2	Floating Point
Yes	%MW30	Z2_DIS	Decimal
Yes	%MF302	Z_ACT2	Floating Point
Yes	%MW90	Z2_ACT	Decimal
Yes	%QW0.1.0	AO	Decimal
Yes	%MF777	VOLTAGE__OUT_CH0	Floating Point
Yes	%MF290	E2_OO	Floating Point
Yes	%MF208	VOLTAGE_IN_CHANAL_1	Floating Point
Yes	%M88		Decimal
Yes	%MF203	E2	Floating Point

PId 2

Used	Address	Symbol	Units
Yes	%MF407	Z_DIS3	Floating Point
Yes	%MW702	Z3_DIS	Decimal
Yes	%QW0.1.0	AO	Decimal
Yes	%QW0.1.1		Decimal
Yes	%MF777	VOLTAGE__OUT_CH0	Floating Point
Yes	%MF317	VOLTAGE_IN_CH2	Floating Point
Yes	%M99		Decimal
Yes	%MF402	Z_ACT3	Floating Point
Yes	%MW39	Z3_ACT	Decimal
Yes	%MF563	E3	Floating Point

List of preferences to print

Directory:

Parameters

Path:	C:\Program Files\Schneider Electric\TwidoSuite\My projects

Image:

Parameters

Image:	Default image
Path:	

Functional levels:

Parameters

Type:	Automatic
Level:	The very highest

Connections management:

Connection

Name	COM1
Connection type	COM
IP / Phone	COM1
Punit / Address	Punit
Baud rate	
Parity	None
Stop bits	
Timeout	5000
Break timeout	5

Connection

Name	COM2
Connection type	COM
IP / Phone	COM2
Punit / Address	Punit
Baud rate	
Parity	None
Stop bits	
Timeout	5000

Break timeout	5

About

License:

```
Company: hroosh co.
User First Name -
User Last Name -
State: Test version
Number of test days: 79
```

References

1) M. W. Song, S.Hutchinsons, and M. Vidyasagar, “Robot Modeling and control”, John Wiley & Sons, Inc, 2006

2) B. Siciliana, L. Sciavicoo, L. Villain, and G. Oriolo, “Robotics modeling, planning and control”, Springer- Verlag London Limited, 2009

3) J.J Craig, “Introduction to Robotics Mechanics and Control”, second edition, Silma. Inc, 1989

4) R. Kelly, V. Santibanes and A. Loria “Control of Robot Manipulator in Joint Space”, Spring- Verlag London Limited, 2005

5) Andrew J. smith and Van Batavia, “Electronic Control for Hydraulic Applications”, Eaton Corporation, 2008