MOTION CONTROL OF A SCARA ROBOT WITH A PLC UNIT

M.Sc Thesis

in

Mechanical Engineering

University of Gaziantep

By

M. Taylan DAŞ

August 2003

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Bülent GÖNÜL
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Abdullah AKPOLAT
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. L. Canan DÜLGER
Supervisor

Examining Committee Members

Prof. Dr. Sedat BAYSEÇ(Chairman)

Assoc. Prof. Dr. Sadettin KAPUCU

Assoc. Prof. Dr. L. Canan DÜLGER

Assist. Prof. Dr. İlyas EKER

Assist. Prof. Dr. Vedat KARSLI

ABSTRACT

MOTION CONTROL OF A SCARA ROBOT WITH A PLC UNIT

DAŞ, M. Taylan
M.S. in Mechanical Engineering
Supervisor: Assoc. Prof. Dr. L. Canan DÜLGER
August 2003, 49 pages

SCARA (Selectively Compliant Articulated Robot Arm) robot is a small industrial robot, which is commonly used for cutting, handling and assembling of small components for continuous manufacturing lines. Programmable Logic Controllers (PLC) can be defined as industrial control devices containing hardware and software to be used for performing control functions, for example; CNC machines, for any kind of Electro-Pneumatic and hydraulic systems, and also in textile machinery. Generally input-output elements of PLCs include timers, counters and some basic mathematical instructions for control applications, which have an important role for today's automation technology.

The purpose of this study is to achieve the motion control of a SCARA robot having four degree's of freedom Serpent 1(Practical Electronics) robot with a PLC based control system. Siemens S7-200 series PLC and Serpent 1 robot are used for this study. Serpent 1 robot system has been studied mathematically. Available system is operated with dc servo motors and pneumatic actuators. Initially, Serpent 1 robot control system is activated. Partial adaptation of the PLC to the robot system is then performed. Three different sized objects are picked from one point and placed to the target.

Keywords: SCARA robot, PLC, Mathematical modelling, Simulation.

ÖZ

SCARA TİPİ ROBOTUN PROGRAMLANABİLİR MANTIK DENETLEYİCİSİYLE (PLC) HAREKET DENETİMİ

DAŞ, M. Taylan

Yüksek Lisans Tezi, Makine Mühendisliği
Tez Danışmanı: Doç. Dr. L. Canan DÜLGER
August 2003, 49 Sayfa

SCARA(Seçici Serbest Esnemeli Robot Kolu) robot özellikle kesme, yükleme, montaj ve üretim hatları gibi sürekliliği gerektiren ortamlarda kullanılan küçük bir endüstriyel robottur. Programlanabilir mantık denetleyicileri (PLC) ise esnek bir mantık stratejisiyle farklı endüstriyel uygulamalarda; CNC tezgahları, taşıma bantları (konveyörler) ve tekstil makinelerinde bulunan elektriksel ve pnömatik mantık devrelerin birlikte kullanımını sağlayan elemanlardır. PLC' ler temel giriş-çıkış olanaklarının yanı sıra zamanlayıcıları, sayıcıları ve diğer denetim elemanları ile günümüz otomatik denetim uygulamalarında sanayide önemli bir yer tutmaktadır.

Bu çalışmanın amacı SCARA, dört serbestlik dereceli Serpent 1 tipi robotun programlanabilir mantık denetleyicisiyle (PLC) hareket denetimini sağlamaktır. Çalışmada Siemens S7-200 serisi PLC ve Serpent 1(Practical Electronics) robot kullanılmaktadır. Mevcut sistem doğru akım servo motorları ve pnömatik sürücülerle hareket etmektedir. Öncelikle robotun orijinal konumlandırması çözümlenmiş daha sonra PLC kısmen SCARA robota uyarlanmıştır. Örnek olarak ise montaj hattında olduğu düşünülen üç farklı boyut ve ağırlıktaki parçaların bir noktadan hedef olarak belirlenen başka bir noktaya taşınarak hareket denetiminin sağlanması üzerinde çalışılmıştır

Anahtar kelimeler: SCARA robot, PLC, Matematiksel modelleme, Simulasyon.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Assoc. Prof. Dr. L. Canan Dülger, for her encouragement, guidance, help and suggestions throughout the course of this study.

My thanks go to Mr. Basar Küçükparmak and Technicians of workshop in Mechanical Engineering Department who helped me for the preparation of system set-up.

Special thanks to my wife and collegue Res. Ass. G. Sena Daş and my family, for their motivation, support, and patience during the preparation of this thesis.

LIST OF CONTENTS

ABSTRACT....................iii

ÖZ....................iv

ACKNOWLEDGEMENTS....................v

LIST OF CONTENTS....................vi

LIST OF TABLES viii

LIST OF FIGURES....................ix

1.INTRODUCTION

- 1.1 SCARA Robot Anatomy....................1
- 1.2 Literature Survey on SCARA Robot....................3
- 1.3 PLC in Automation....................5
- 1.4 Literature Survey on PLCs in Automation....................6
- 1.5 Thesis Structure....................7

2. SERPENT 1 ROBOT STRUCTURE

- 2.1 Introduction....................8
- 2.2 Control Structure of Serpent 1 Robot....................9
  - 2.2.1 Programming Techniques of SCARA Robots....................10
  - 2.2.2 Methods of Robot Programming....................11
- 2.3 Applications of Serpent 1 Robot....................13

3. MATHEMATICAL MODELLING AND SIMULATION OF SERPENT 1

- 3.1 Introduction....................16
- 3.2 System Modelling....................16
  - 3.2.1 Serpent 1 Configuration....................16
  - 3.2.2 DC Servo Motor Modelling....................18
  - 3.2.3 Motor Controller Action....................20

3.3 Systems Equations of Motion....20
3.4 Numerical Integration....22
3.5 Simulation and Experimental Results....25

4. PROGRAMMABLE LOGIC CONTROLLER (PLC)

4.1 Introduction....30
4.2 PLC Operation....31
  4.2.1 Parts of PLC....31
  4.2.2 Inputs and outputs....32
  4.2.3 Scan Cycle....33
4.3 PLC Functions and Programming Techniques....34
  4.3.1 Logic Circuit....34
  4.3.2 Latches....35
  4.3.3 Timers....36
  4.3.4 Counters....37
  4.3.5 Subroutines....37
4.4 Routine Applications with PLC....38
4.5 Linear Positioning with PLC....40
4.6 SCARA Robot Control with PLC....42

5. CONCLUSIONS AND DISCUSSIONS

5.1 About the Present Work....45
5.2 Discussion on Present Work....46
5.3 Recommendations for the Future Work....47

REFERENCES....48

LIST OF TABLES

Table 2.1 Specifications of Serpent 1 robot 10

Table 2.2 Specifications of Objects 14

Table 3.1 Mechanical Data for the Main and Fore Arms and Electromechanical data for dc servo motors 19

Table 4.1 Timer numbers and resolutions 36

Table 4.2 Positions of Serpent 1 Robot Arms 43

Table 4.3 Sequence of Pneumatic control structure and PLC program 44

LIST of FIGURES

Figure 1.1 Base Anatomy of SCARA 2
Figure 1.2 The Photograph of Serpent 1 3
Figure 1.3 Control scheme of the PLC 5
Figure 2.1 Control Structure of Serpent 1 9
Figure 2.2 Working Envelope Of Serpent 1 13
Figure 2.3 Illustration of the working envelope of located objects 14
Figure 2.4 Photograph of Picking and Placing Objects 15
Figure 3.1 The Serpent 1 Configuration 17
Figure 3.2 An armature controlled pm dc motor model 18
Figure 3.3 Flow Chart of the Computer Simulation for Serpent 1 Robot-Actuator System 24
Figure 3.4 The Simulated and Experimental Results (1st Example) 27
Figure 3.5 The Simulated and Experimental Results (2nd Example) 28
Figure 3.6 The Simulated and Experimental Results (3rd Example) of Main Arm and Fore Arm 29
Figure 4.1 General configurations of PLC 32
Figure 4.2 Scan cycle of PLC 33
Figure 4.3 Illustration of logic functions 34
Figure 4.4 Ladder diagram of the example 35
Figure 4.5 Latch Circuit 35
Figure 4.6. Timer instructions for S7-200 PLC 36
Figure 4.7 Counter instructions for S7-200 PLC 37
Figure 4.8 A Pneumatic cylinder with PLC control 38
Figure 4.9 Two Pneumatic Cylinders with PLC control 39
Figure 4.10 A DC motor control with PLC 40
Figure 4.11 Photograph of Linear Motion System 40
Figure 4.12 General Structure of Linear Motion System 41
Figure 4.13 The photograph of Serpent 1 robot control with PLC 42
Figure 4.14 Schematic Representation of Serpent 1 with PLC 43

CHAPTER 1

INTRODUCTION

The scope of this study is to perform motion control of a Selectively Compliant Articulated Robot Arm (SCARA) with a Programmable Logic Controller (PLC). In this study, SCARA robot, which is available in Dynamic Systems Laboratory, Department of Mechanical Engineering and Siemens S7-200 CPU 216 PLC have been used.

1.1 SCARA (Selectively Compliant Articulated Robot Arm) Robot Anatomy:

Selectively Compliant Articulated Robot Arm (SCARA) is an industrial robot, which is firstly designed at Japan's Yamamachi University[1-5] for filling the gap between simple pneumatic pick-and-place and servo controlled robot modelled upon the human arm like Unimation Puma. It is engineered in this fashion in order to provide compliance in the horizontal directions; a characteristic is also essential for vertical insertion operations. This makes SCARA a good choice for limited assembly tasks where workpiece access is from above. The motion possibilities of the robot arms are revolute motions confined to the horizontal plane, together with translation of this plane. Gripper axis works on vertically. Such a robot is also known as "an articulated robot arm" or " an assembly robot arm". SCARA is basically an anthropomorphic or jointed-arm (RRR) structure. The base anatomy of a SCARA robot is given in Figure 1.1.

SCARA has four degree's of freedom (DOF); three rotational axes which operate on X-Y plane, and the vertical axis performs up and down movement on Z plane with the aid of a pneumatic cylinder. Three rotational motions are provided by theta1 (θ_1) is the main arm, the forearm is theta2 (θ_2), third rotary axis roll is the wrist, theta3(θ_3). The wrist is usually belt driven from a motor at the fixed end of the arm. The importance of this configuration is that it keeps the wrist, and hence the workpiece at a constant angle with respect to the bench independent from the arm movement. The vertical axis is also important for positioning of SCARA.

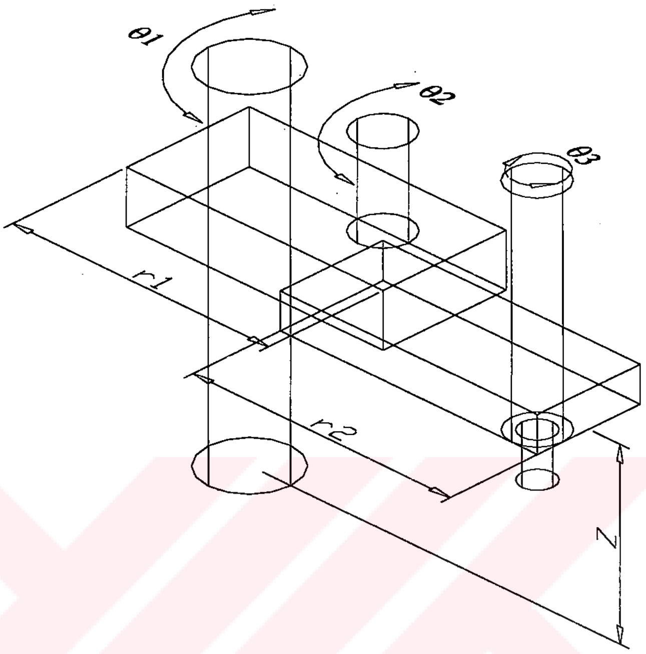


Fig 1.1 Base Anatomy of SCARA

In this study, Serpent 1 type SCARA robot is used. The system is driven with three main 24 V dc servo motors under closed loop control. The vertical axis and the gripper operated pneumatically, are also controlled by solenoid valves. The photograph of Serpent 1 robot system can be seen in Figure 1.2.

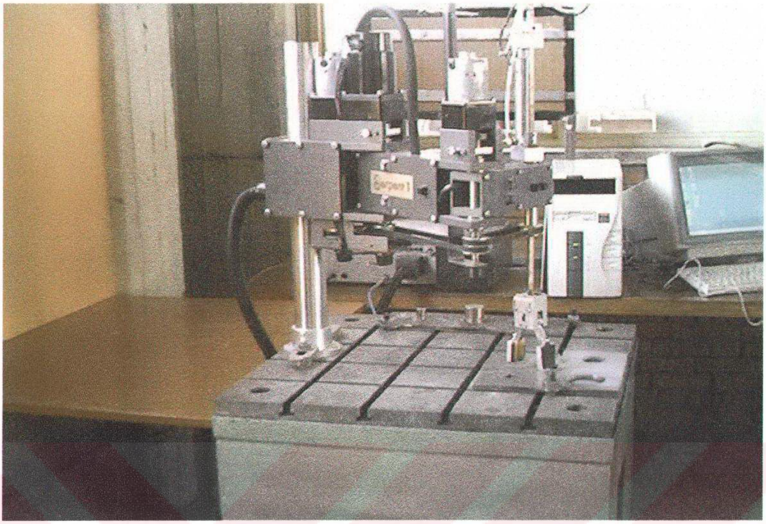


Figure 1.2 The Photograph of Serpent 1

1.2 Literature Survey on SCARA

Previous studies are searched on SCARA which is widely used and controlled industrial robot.

Ertürk (1997) [6], has studied on control of a SCARA robot. Robot programming details, programming languages and inverse kinematic analysis were presented in this thesis.

Bhatia et al. (1998)[7], have used an expert system for design of a SCARA robot to reduce the design cycle time of the current four-six months down to a few days for supplying a required customer demand optimisation. The duty has been estimated to make more difficult, due to the fact that many design parameters could not be predicted until the design was begun. Therefore design parameters were iteratively modified, and SCARA robot has been prepared with suitable changes. Additionally, some robot design parameters, like payload, mobility, workspace, agility, accuracy and repeatability of positioning can also be found in this study, that was useful for an implementation stage of SCARAs.

Ge et al. (1999) [8] have presented dynamic modelling and controller design for a SCARA Cartesian smart materials robot with piezoelectric actuators and sensors.

Omodei et al. (2000) [9] have presented, and compared three algorithms for the geometric parameter identification of industrial robots. Experimental results were obtained for a SCARA IBM 7535 robot.

Er et al. (2001) [10] have studied on the design, development and implementation of a Hybrid Adaptive Fuzzy Controller (HAFC) suitable for real-time industrial applications. The SEIKO D-TRAN 3000 series SCARA robot was controlled and analysed.

Hong et al. (2001) [11] have been investigated modular and object-oriented approach for the PC-based open control (PC-ORC) system, that technique has been applied on a SCARA robot. The PC-ORC has existed basic software programming, application tools, and some other hardware equipment on a PC. The basic advantage of the system, which was applicable for illustrating every motion of robots and taking feedback of the sensors that can also be seen on the computer screen. Owing to this technique, movements of the robot can be directly changed.

Nagchaudhuri et al. (2002) [12] have considered industrial robots as a perfect teaching equipment for demonstrating improvements in mechatronics, which basically included all application areas together, mechanism usage and analysis, soft computing and electronics. This study has been considered to illustrate flexible manufacturing systems to students by integrating imaging and action using an industrial SCARA robot. Therefore, the two-fingered gripper has been taken under the control of an overhead camera, which was used to find correct letters of "HELLO". Finally, manipulation of letters of the "HELLO" has been carried from existing point to another ones, and then all of them have been placed in arbitrary position and orientation in the robot working area.

Miro and White (2002)[13] have then presented modelling and simulation of an industrial manipulator as a case study. A complete model was provided with its mechanical and electrical equations of motion.

1.3 PLC in Automation

Programmable Logic Controllers (PLCs) are very popular in application of control systems in nowadays industry. Many of the workcells and factory plants are controlled by automatically with different PLC's that technologies are continuously developed. In this study, PLCs are chosen for observing their industrial applications.

PLCs are often defined as miniature industrial computers that contain hardware and software used to perform different control functions. A PLC consists of two basic sections: the central processing unit (CPU) and the input/output interface system. The CPU, which controls all PLC activity, can further be broken down into the processor and the memory system [14]. The input/output system is physically connected to field devices (e.g., switches, sensors, etc.) and provides the interface between the CPU and the information providers (inputs) and the controllable devices (outputs). Control scheme of the PLC can be seen in Figure 1.3.

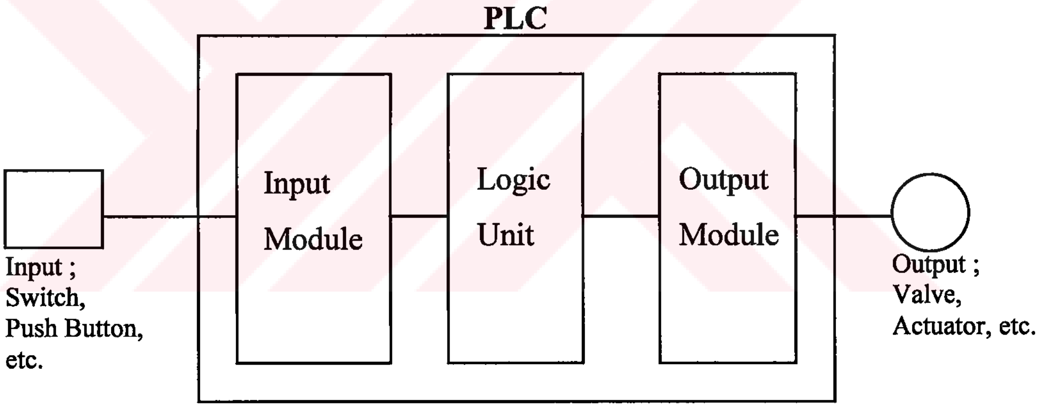


Fig.1.3 Control scheme of the PLC

Initially the CPU "reads" input data from connected field devices through the use of its input interfaces, and then "executes", or performs the control program that has been stored in its memory. Programs are typically created in a languages, which are named as ladder logic, list and functional block diagram. These languages closely resemble a relay-based wiring schematic, and are entered into the CPU's memory prior to operation. Finally, based on the program, the PLC "writes", or updates the output devices via the output interfaces. This process, also known as scanning, continues in the same sequence without interruption, and changes only when a change is made to the control program [15].

1.4 Literature Survey on PLCs in Automation

Lu et al. (1992) [16] have presented a basis of the PLC simulation system used for testing programs off-line. Also new method, named as "direct sequential method", has been tried for simulating PLC. Based on that method, the software has been implemented in C programming language and tested. Mathematical calculations were not required for this method.

Starr et al. (1999) [17] have presented comparison of two generic diagnostic models for PLC controlled flexible manufacturing systems (FMS). Descriptions of the models and applicable FMS examples, which have been selected to demonstrate the major operational faults of PLC controlled FMS's were given.

Figliolini and Ceccarelli (1999) [18] have studied on an Electro-pneumatic walking robot named as EP-WAR. Design and manufacturing parameters have been defined in this study. The main structure of the robot has been composed by a pantograph with a double articulated paralelogram, which were driven with pneumatic actuators. Also all pneumatic actuators and electronics of the system of the robot were controlled by a PLC. Additionally, three walking modules were programmed, and used on the robot with PLC programming techniques. This study was a good example for understanding PLC's and their working principles in robotic applications.

Pfeiffer (2000) [19] has presented concept of self-tuning PID controllers for temperature control, that can also be used for other industrial control systems. Modelling, control concept of the system and implementation were given. PID controllers of PLCs were used that have become a standard component of automation with PLCs.

Ravina (2001) [20] has studied design and description of an original mechatronic unit for acoustic characterisation of industrial machines and noisy devices. Therefore, automatic motions of probes have been supplied. They have been carried by a pneumatic three-axial gantry positioning unit, controlled by PLC.

1.5 Thesis Structure

In Chapter 1, the purpose of study, a brief introduction on SCARA robot, PLC and their industrial applications are described. The recent studies including literature survey, and historical descriptions are given for SCARA robot and PLC.

In Chapter 2, working principle of SCARA robot and application areas are given. All members of SCARA are indicated with all necessary drawings and photographs. The robot programming techniques, which type of control technique is applied on system and what is done with robot are tried to be answered one by one. Walli language used for Serpent 1 is described. Finally, an example of SCARA robot, Serpent 1 and related figures are included.

In Chapter 3, mathematical modelling and simulation of Serpent 1 robot can be seen. Formulations and modelling methods are written and given with general descriptions. The equations of motion are found by using Lagrange's equations. Simulation of Serpent 1 robot is performed by applying 4th order Runge Kutta method of integration. A comperative study is performed between the mathematical model involving trajectory of the Serpent 1 and the experimental results. Simulation and experimental results are then presented and explained briefly.

In Chapter 4, Programmable Logic Controller (PLC) is defined. Basic PLC models, application areas, general descriptions, basic illustrative samples of PLCs timers, counters, I/O modules are given. Subroutine programs, mathematical applications and comparative expressions are then included. Programming methods and their applications with demonstration kits, that are achieved in Dynamic Systems Laboratory with S7-200 Siemens PLC are presented together with PLC algorithms. Partial control of Serpent 1 with PLC is included.

In Chapter 5, Conclusion, discussion and general comments are given. Experimental results, problems observed during the thesis study, some important limitations are presented. Finally, recommendations for future studies are included in the conclusion stage.

CHAPTER 2

SERPENT 1 ROBOT STRUCTURE

2.1 Introduction

Robots have been classified according to the teaching methods used as manual manipulators, fixed sequence robots, variable sequence robots, playback robots, numerical control robots, and intelligent robots. Another commonly used classification scheme designates a robot as pick-and-place, which has two possible locations defined by mechanical stop. Point-to-point robots have repeated task cycles defined by a limited sequence of desired locations. Continuous path or assembly robots depending on the motion control algorithms that are classified as servo and playback. Continuous path robots have been similar to point-to-point robots except the programmed locations are closely spaced to achieve near total trajectory control, operating them a closed loop mode[2].

2.2 Control Structure of Serpent 1

In this study, Serpent 1 robot is used for applications. Serpent 1, an industrial robot which is a 4 DOF system, has 3 rotational and one translational axis. Three rotational motions are provided by theta1 (θ_1) is the main arm, which makes shoulder movement, the forearm is theta2 (θ_2) makes an elbow movement, a third rotary axis roll is the wrist, theta3 (θ_3). These can be seen in Figure 1.1 in Chapter 1.

The rotational axes are driven by 24 volt direct current (dc) servo motors and the translation axis is driven by a pneumatic cylinder. They are controlled by 3 servo motor amplifiers and solenoid driver. A solenoid driver for the pneumatic cylinder and an interface board connected to a computer. Control structure of Serpent 1 robot can be seen in Figure 2.1.

Here, data from the computer is passed to the interface board, which has been a multiplexed, digital to analogue converter (DAC) and analogue to digital converter (ADC) provides 6 outputs and 16 inputs of 12-bit resolution analogue voltages. The position of each rotary axis of the arm is sensed by angular potentiometers, which give an error voltage in the range of –10V to +10V. The power amplifiers drive motors with a voltage in the range of –24V to +24V [1].

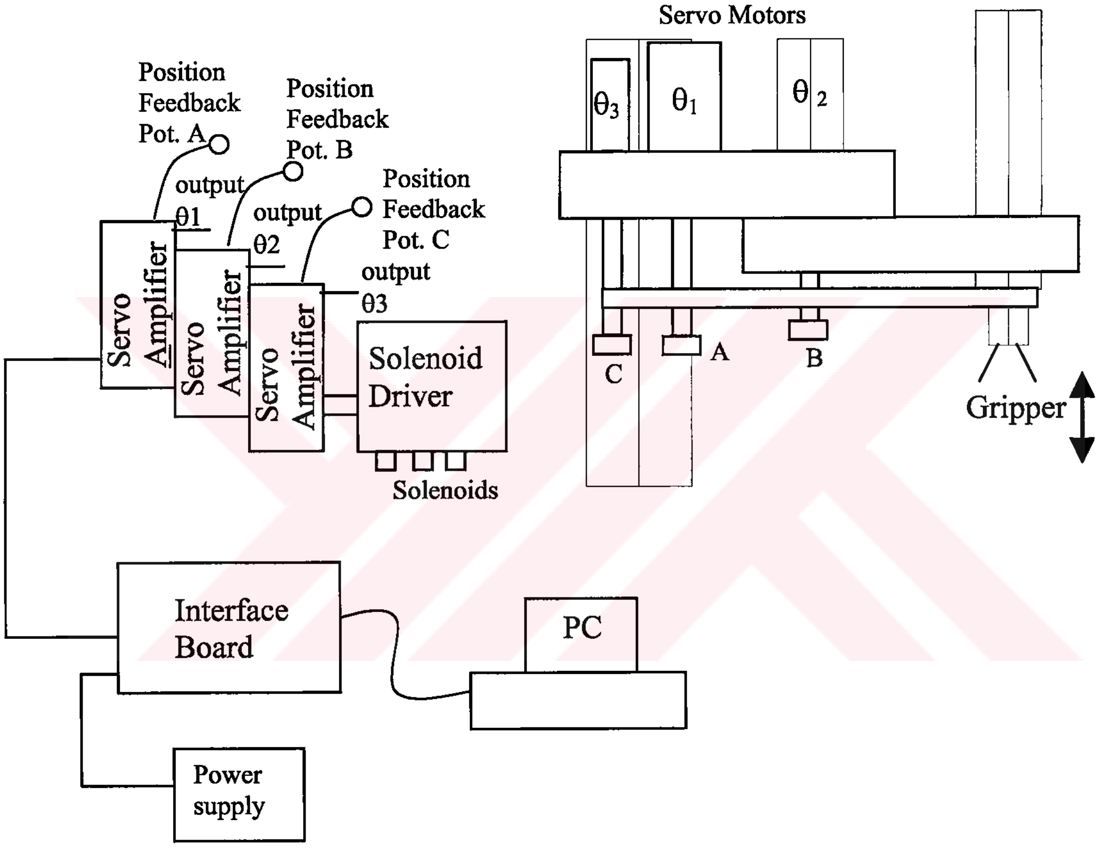


Figure 2.1 Control Structure of Serpent 1

Pneumatic actuators are very popular because of their simplicity of control and their beneficial economic characteristics. One of the important disadvantages of pneumatically actuated manipulators is its reduced stiffness and natural frequencies caused by the compressibility of air [3]. In pneumatic control of robot, three 24V solenoid valves have been used for Serpent 1 robot as the gripper and the vertical axis motion control. General specifications of Serpent 1 are given in Table 2.1.

Table 2.1 Specifications of Serpent 1 robot

Main Arm Length (r_1)	250mm
Fore Arm Length (r_2)	150mm
Shoulder movement (θ_1 axis)	200°
Elbow movement (θ_2 axis)	250°
Wrist rotation (θ_3 axis)	450°
Z axis (up-down)	75mm
Control System	12bit
Repeatability	1.0 mm
Maximum tip velocity	550mm/sec
Capacity	2.0 kg

2.2.1 Programming Techniques of Robots

Robot programming languages can be classified with different levels of which are joint control level(very low), motion level(low), structured programming level(medium), task-oriented level(high). Some examples can be given as AMBRASIC and RASP for joint control level programming languages. ANORAD, EMILY, RPL, SIGLA, VAL are motion level programming languages respectively. AL, HELP, MCL, PAL are structured programming level programs. Task-oriented level programming languages are AUTPASS, RAPT [5].

Workcell Amalgamated Logical Linguistic Instructions (WALLI) 2.5, which is a joint control level programming language has been used for main programming of Serpent 1 robot. Workcells consisting eight robots and expansion boards that chained together can be easily operated by WALLI 2.5 from a single controlling computer. In WALLI 2.5, the sequence files are used as the directories of positions. The program consists of 5 main parts; Editor, Debugger, Interpreter, WALLI program and Robot Position Data Files [1].

Editor is used for writing the WALLI program and as its name implies includes powerful editing facilities. These are called by only pressing a single key. The statements require only pairs of key letters to be entered followed by simple answers to on-screen prompts.

Debugger is used to check for ambiguities within the WALLI program which may not be picked up by the interpreter. It is run automatically during the start of each execution of the program.

Interpreter executes the finished WALLI program by setting flags and data which are used by an interrupt routine which reads from and writes to each devices every time when the routine is called. That is repeated in every 55ms. The interrupts come from the 8253 elapsed-time counter.

WALLI Program is a numbered sequence of instructions, for the robots, mills and expansion boards. WALLI statements are in intelligible English with minimal abbreviation.

Robot Position Data Files are numbered sets of data representing the robot axis positions that are to be used in the WALLI program. The files are created by the main control programs of the robots.

2.2.2 Methods of Robot Programming

The robot motion programming has been classified into two main techniques; teach-and-playback method, using an on-line teach pendant, and off-line high level programming languages [4,5]. The main difference between them teach-and-playback has required very small memory space, and that was simple to learn so that suitable for simple tasks. Off-line programming was related with the programs for complicated duties. It has required user's expertise and experience in the area of robot language.

Off-line program may be considered as the process by which robot programs are developed partially or completely, without requiring the use of the robot itself. This includes generating point coordinate data, function data, and cycle logic. Developments in both hardware and software robot technology make off-line programming techniques more feasible. Especially for SCARA type industrial robots which are used in welding, painting and assembling lines, where the reprogramming time required was either minimal or absent. The increasing complexity of robot applications, particularly regarding to assembly work, makes the advantages associated with the off-line programming. These are given as follows: [5]

- Reduction of robot downtime.
- Removal of programmer from potentially hazardous enviroments.
- Single programming system.
- Integration with existing CAD/CAM systems.
- Simplification of complex tasks.
- Verification of robot programs

Teach programming is a means of entering a desired control program into the robot controller. In teach programming the robot is manually led through a desired sequence of motions by an operator. The movement information is recorded by the robot controller as the robot is guided through the desired path during the teach process. Teach programming is the most natural way to program an industrial robot. In the study presented, CHARMER has been used for Serpent 1 robot programming. In the main menu, control options are available for teaching the robot and replaying the stored sequences.

Closed Loop Keyboard Control is used for direct use of robot and to save program for other applications. In the screen of computer θ_1, θ_2, θ_3, Up/Down and gripper (open/close) positions are adjusted. Feedback data, velocity replacement and delay time can be also changed in this section.

Open Loop Keyboard Control has been included as a method in which the position data can be introduced or changed manually on the robot. Switches are available for each motor control. However, pneumatic control is not included in this menu.

Replays of Sequence, In this manner, methods of replay sequences are single step, continuous replay for once, and repeat sequence (forever replay). Other topic of the menu is the editing of sequence, which is similar to a Microsoft office application all cut, paste, copy applications are available in this menu. Save and recall can be done in the replay of sequence menu.

List Sequence, the list of sequence in memory can be displayed on the computer screen.

2.3 Applications of Serpent 1 Robot

Point to point control, which moves from one position to the next with no consideration of the path taken by the manipulator, is done with Serpent 1 robot. Three different dimensional and weighed objects are located on working envelope of the robot to demonstrate the robot capacity clearly. Tip of the gripper is redesigned and produced for preventing against to the sliding effect, which is caused by different weights and surfaces of objects. New gripper also increases touching surface of the gripper to the object. Working envelope and dimensional limitations are shown in Figure 2.2 for Serpent1 robot.

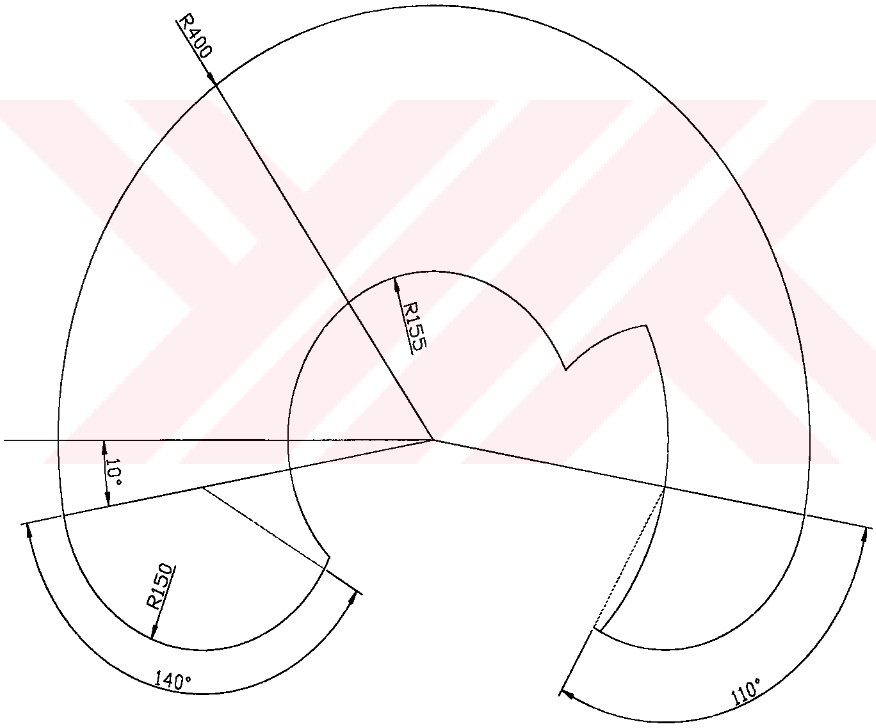
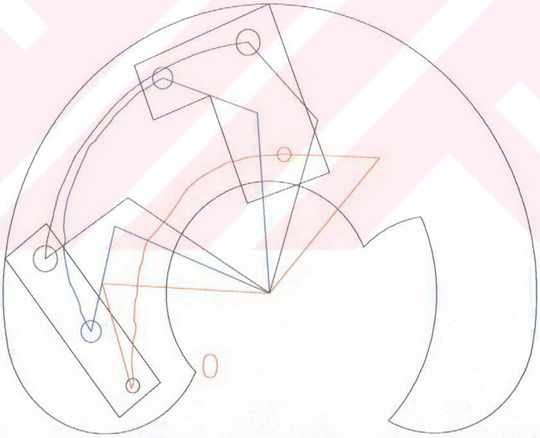


Figure 2.2 Working Envelope Of Serpent 1

In demonstration small, medium and large objects are carried respectively. Paths of the carried objects are drawn experimentally in Figure 2.3, which shows picking and placing of all. That is closely same with the model of Serpent 1 robot described in chapter 3. There is an offset difference between photograph and drawing because of the initial mounting of the Serpent 1 robot. Top view photograph of the system is also shown in Figure 2.4. Dimensions and weights of objects are shown in Table 2.2. Serpent 1 robot control and programming are achieved with different examples. Teach-and playback was only used because of the absence of software program of the robot, since more detailed was not sent from the company.

Table 2.2 Specifications of Objects

	Dimension(mm)	Mass (g)
Small	ϕ20 x 30	100
Medium	ϕ38 x 40	380
Big	ϕ58 x 50	820



Small object is represented by red colour

Medium Object is represented by blue colour

Big object is represented by brown colour

Figure 2.3 Illustration of the working envelope of located objects

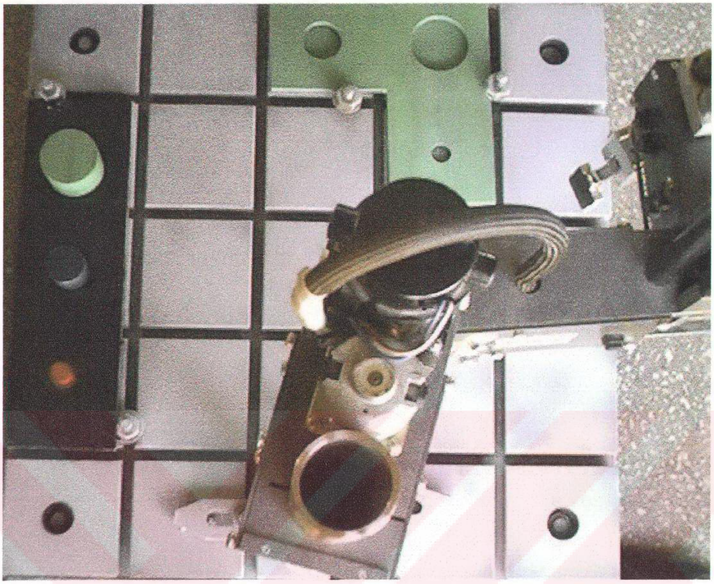


Figure 2.4 Photograph of Picking and Placing Objects

CHAPTER 3

MATHEMATICAL MODELLING AND SIMULATION OF SERPENT 1

3.1. Introduction

A complete mathematical model of Serpent 1 is developed including servo actuator dynamics and presented together with dynamic simulation in this chapter. A Serpent 1 robot is instructed to achieve pick and place operations of 3 different size cylindirical objects through the assigned holes. The dc servo motors driving each robot joint is studied. The performance of robot-actuator-control system is examined with numerical simulation and experimentally verified. The displacement of the robot is described analytically such that the robot can position a component on assembly line. So the position and orientation of the gripper is obtained.

3.2. System Modelling

The equations of motion relating joint torques, positions, velocities and accelerations are derived for each joint with Lagrangian formulation. Actuator dynamics is also included by considering a model for pm dc servomotor and its control structure.

3.2.1. Serpent 1 Configuration

The anatomy of the SCARA is shown in Figure 1.1, Chapter 1. It operates on the XY plane where $\theta_1$and θ_2 represent the main arm (shoulder) and the fore arm (elbow) respectively. The third rotary axis, roll is given as the wrist. Vertical movement is performed in the Z direction. The gripper has powered pneumatically. Working envelope and specifications of Serpent 1 are given in Figure 2.1 and Table 2.1 in Chapter 2. In the application, the gripper is positioned in x-y plane, not always required to rotate about a vertical axis. The gripper also works at a constant vertical displacement which is not taken while developing a mathematical model. So this dof is neglected while deriving the equations of motion. Motion control is implemented only for axes θ_1 and θ_2. Gravitational effects are not taken into consideration also.

The Serpent 1's schematic representation is shown in Figure 3.1. The robot kinematics is completely described by referring to Figure 3.1.The parameters of Serpent 1 used throughout the study are listed below as

r_1, r_2 - lengths of the main arm and the fore arm (m)
m_1, m_2 - masses of the main and the fore arm (kg)
J_1, J_2 - moment of inertias for the main arm and the fore arm (kg.m^2)
$\theta_1, \dot{\theta}_1, \ddot{\theta}_1$ -angular displacement, velocity and acceleration of the main arm
$\theta_2, \dot{\theta}_2, \ddot{\theta}_2$ -angular displacement, velocity and acceleration of the fore arm

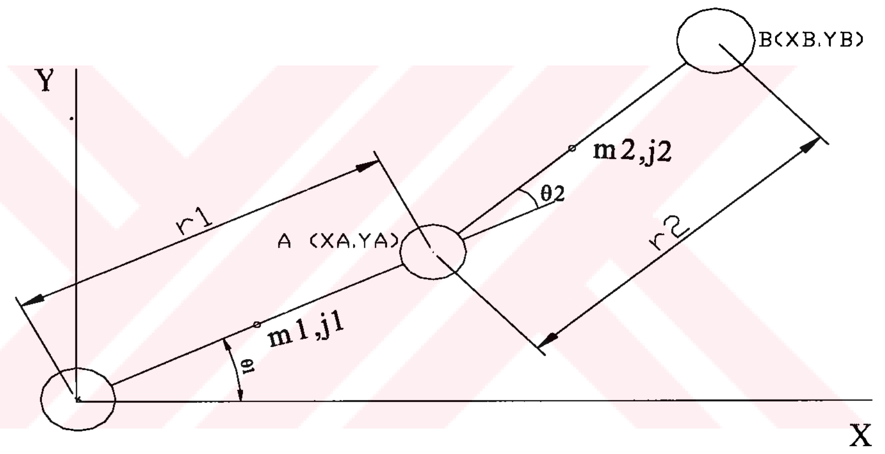


Figure 3.1. The Serpent 1 configuration (2 dof)

The inverse kinematics transforms the output position into the joint coordinate. The position of the arm end refers to the tip of the manipulator. So the coordinates of point B, x_B and y_B can be written as:

$$x_B = r_1 \cos\theta_1 + r_2 \cos(\theta_1 + \theta_2) \tag{3.1}$$

$$y_B = r_1 \sin\theta_1 + r_2 \sin(\theta_1 + \theta_2) \tag{3.2}$$

First and second derivatives of equations (3.1) and (3.2) give the velocity and acceleration of point B. Inverse mapping of linkage coordinates to joint coordinates to be performed. So a known inverse problem appears to be solved by using equations (3.1) and (3.2). Solution can be found in many text books [2, 3]. The main arm and the fore arm joint angles θ_1 and θ_2 can be expressed in terms of tip position of Serpent 1 as:

$$\cos\theta_2 = \frac{(x_B^2 + y_B^2) - (r_1^2 + r_2^2)}{2r_1r_2} \tag{3.3}$$

$$\tan\theta_1 = \frac{-(r_2\sin\theta_2)x_B + (r_1 + r_2\cos\theta_2)y_B}{(r_2\sin\theta_2)y_B + (r_1 + r_2\cos\theta_2)x_B} \tag{3.4}$$

By using equations (3.3) and (3.4), taking time derivatives of them, the main arm and the fore arm joint velocities and accelerations can also be calculated for Serpent 1.

3.2.2. DC Servo Motor Modelling

A dynamic model for a pm dc servo motor has an armature equation and a dynamic equation which must be considered together with robot equations [21-23]. Loads are driven by pm dc servo motors through reduction gearboxes and timing belts representing an indirect drive configuration.

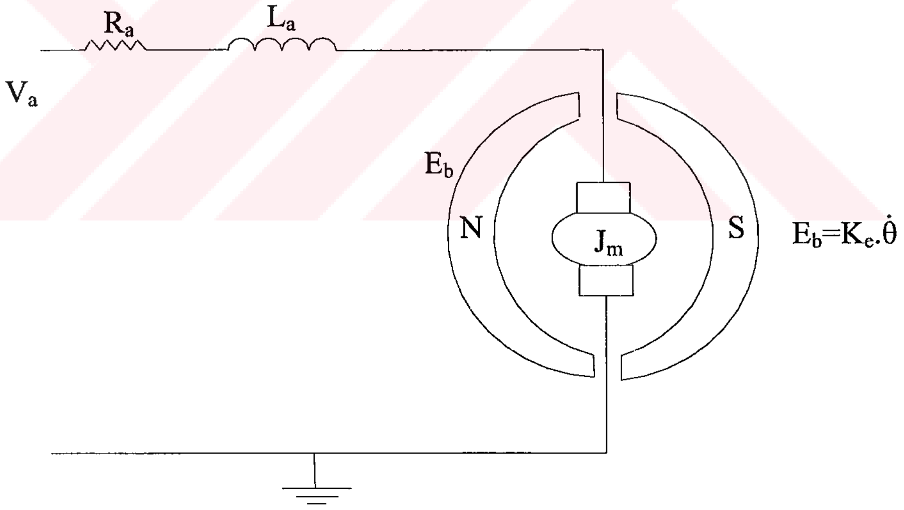


Figure 3.2. An armature controlled pm dc motor model

A circuit model includes a pm armature controlled dc motor, gearbox, belt-pulley and a mechanical load. The motor armature model can be seen in Figure 3.2 where E_b representing the back emf voltage across the motor. Kirchhoff's voltage law is applied around the armature windings.

The motor armature equation is given as:

$$L_{an}\frac{dI_{an}}{dt} + R_{an}I_{an} + K_{e_n}\dot{\theta}_{mn} = V_{an} \tag{3.5}$$

Where I_{an} is the current of motor armature circuit (A) , R_{an} is the resistance of motor armature winding (Ω), L_{an} is the inductance of motor armature (H), Ke_n is motor back emf constant (Vs/rad), V_{an} is the motor control voltage (V) and subscripts n=1,2 referrring to the main arm (shoulder) and the fore arm (elbow) movements respectively. By modeling pm dc motors applied, the control torques to be employed can be calculated. The motor current I_n also produces a torque T_n, proportional as

$$T_n=K_{Tn}I_n \tag{3.6}$$

Where K_{Tn} is the motor torque constant (Nm/A). Two identical 24 V pm dc servo motors are used for actuating the main arm and the fore arm joints, and electrical data for dc servo motors are listed in Table 3.1. The back arm movement is performed by a gearbox and belt-pulley arrangement.

A gear train introduces a relation,

$$N_n = \frac{\theta_{mn}}{\theta_n} \tag{3.7}$$

Where N_n is the coupling ratio introduced by gearbox 1 and 2 and θ_{mn} is the angular displacement of the servo motor. While calculating the reflected moment of inertias and motor angular displacement, velocity and acceleration, equation (3.7) is used in the coming sections. Equivalent moment of inertia is also taken as;

$$J_{m1}=J_m+J_{g1} \quad \text{and} \quad J_{m2}=J_m+J_{g2}$$

Table 3.1. Mechanical data for the main and fore arms and electromechanical data for dc servo motors

Moment of inertia of the main arm	J_1=0.0980 kg.m^2
Moment of inertia of the fore arm	J_2=0.0115 kg.m^2
Mass of the main arm	m_1=1.90 kg
Mass of the fore arm	m_2=0.93 kg
Moment of inertia of motor 1 and 2	Jm_1= Jm_2=3.3x10^{-6} kg.m^2
Motor voltage constant	K_{e1}= K_{e2}=0.047 V/rad/s
Motor torque constant	K_{t1}= K_{t2}=0.047 Nm/A
Motor armature resistance	R_{a1}= R_{a2}=3.5 Ω
Motor armature inductance	L_{a1}= L_{a2}= 1.3 mH
Gearbox ratio for the main arm	N_1=90
Gearbox ratio for the fore arm	N_2=220
Gearbox 1 moment of inertia	J_{g1}=0.0002 kg.m^2
Gearbox 2 moment of inertia	J_{g2}=0.0005 kg.m^2

3.2.3. Motor Controller Action

Many types of controller schemes; PD, PI, PID can possibly be employed. A lot of studies can be found on applications of PD and PID controllers on servo systems.[24, 25] In the study presented, PD action is chosen with the control software available. In this study, a PD controller is used for calculating the armature control voltage as

$$V_{an}(t) = K_{pn} e_n(t) + K_{dn} \frac{de_n(t)}{dt} \tag{3.8}$$

where the error

$$e_n(t) = \theta_{mn_c}(t) - \theta_{mn}(t) \tag{3.9}$$

$$\dot{e}_n(t) = \dot{\theta}_{mn\,c}(t) - \dot{\theta}_{mn}(t) \tag{3.10}$$

and $V_{an}(t)$ is the controller output voltage. K_{pn} and K_{dn} represent proportional and derivative control gains. Motor referance point is given by subindis nc. Subscript n becomes 1 and 2 while describing the main arm and the fore arm joints.

Experimental data points are measured while performing three different experiments; two axes positioning (θ_1 and θ_2 are changing), one axis movement where θ_1 is constant, one axis movement where θ_2 is constant. The numerical methods are used for curve fitting for the trajectories followed by Serpent 1 [17,18]. The experimental data for θ_1 and θ_2 are correlated with analytical expressions between time variable. The overall data are fit by using least-squares regression. The best function is found to be 2nd order polynomials for both axis. Thus having found the best fit, the angular velocity points are calculated, and then used in equation (3.10) for utilizing feedback information in velocity error during simulation.

3.3. Systems Equations of Motion

The dynamic equations of Serpent 1 are derived by using Lagrange's equations described detailed in given referances [21,22,24]. The mathematical model is developed with some simplifying assumptions; without friction in all joints, and gearbox, transmission losses. Since Serpent 1 operates in the horizontal plane, gravity effects are not taken into consideration. Lagrange's equations are expressed as:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right)-\frac{\partial L}{\partial q_i}=Q_i \quad \text{for i=1,2,....,n} \tag{3.11}$$

Where L is the Lagrangian function (L=T-V) which is an energy approach refers to system's total kinetic and potential energies as a whole. Q_i represent the generalized torques which characterizes the influence of all nonconservative forces for the main arm and the fore arm, and q_i is the generalized coordinate.

The system equations of motion include two second order ODEs. The two generalized coordinates are the joint angles of the main arm and the fore arm; defined as θ_1 and θ_2.

Serpent 1's total kinetic energy, T can be given by

$$T=\frac{1}{2}J_{m1}\dot{\theta}_{m1}^{2}+\frac{1}{2}m_1\left(\frac{r_1^2}{4}\right)\dot{\theta}_1^{2}+\frac{1}{2}J_1\dot{\theta}_1^{2}+\frac{1}{2}J_{m2}\dot{\theta}_{m2}^{2}+\frac{1}{2}J_2(\dot{\theta}_1^{2}+2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_2^2)+$$

$$\frac{1}{2}m_2\left\{r_1^2\dot{\theta}_1^{2}+\left(\frac{r_2^{2}}{4}\right)(\dot{\theta}_1+\dot{\theta}_2)^2+r_1r_2(\dot{\theta}_1+\dot{\theta}_1\dot{\theta}_2)[\cos(\theta_1+\theta_2)\cos\theta_1+\sin(\theta_1+\theta_2)\sin\theta_1]\right\} \tag{3.12}$$

The equations of motion of Serpent 1 are given by including the dynamics of dc servo motor as the following.

For the main arm (shoulder) movement,

$$L_{a1}\frac{dI_{a1}}{dt}+R_{a1}I_{a1}+K_{e1}\dot{\theta}_{m1}=V_{a1} \tag{3.13}$$

$$\left[J_{m1}+\frac{(J_1+J_2)}{N_1^2}+\frac{(m_1r_1^2+m_2r_2^2+4m_2r_1^2)}{4N_1^2}\right]\ddot{\theta}_{m1}+$$

$$\left[\frac{m_2r_1r_2}{N_1^2}\left(\cos\left(\frac{\theta_{m1}}{N_1}+\frac{\theta_{m2}}{N_2}\right)\cos\left(\frac{\theta_{m1}}{N_1}\right)+\sin\left(\frac{\theta_{m1}}{N_1}+\frac{\theta_{m2}}{N_2}\right)\sin\left(\frac{\theta_{m1}}{N_1}\right)\right)\right]\ddot{\theta}_{m1}$$

$$+\left[\frac{J_2}{N_1N_2}+\frac{m_2r_2^2}{4N_1N_2}+\frac{m_2r_1r_2}{2N_1N_2}\left(\cos\left(\frac{\theta_{m1}}{N_1}+\frac{\theta_{m2}}{N_2}\right)\cos\left(\frac{\theta_{m1}}{N_1}\right)+\sin\left(\frac{\theta_{m1}}{N_1}+\frac{\theta_{m2}}{N_2}\right)\sin\left(\frac{\theta_{m1}}{N_1}\right)\right)\right]\ddot{\theta}_{m2}$$

$$+\frac{m_2 r_1 r_2}{N_1 N_2}\left(\frac{\dot{\theta}_{m1}\dot{\theta}_{m2}}{N_1}+\frac{\dot{\theta}_{m2}^2}{2N_2}\right)\left[\left(\cos\left(\frac{\theta_{m1}}{N_1}+\frac{\theta_{m2}}{N_2}\right)\sin\left(\frac{\theta_{m1}}{N_1}\right)-\sin\left(\frac{\theta_{m1}}{N_1}+\frac{\theta_{m2}}{N_2}\right)\cos\left(\frac{\theta_{m1}}{N_1}\right)\right)\right]$$

$$= T_1(t,\theta_{m1},\dot{\theta}_{m1}) \tag{3.14}$$

For the fore arm (elbow) movement;

$$L_{a2}\frac{dI_{a2}}{dt}+R_{a2}I_{a2}+K_{e2}\dot{\theta}_{m2}=V_{a2} \tag{3.15}$$

$$\left[J_{m2}+\frac{J_2}{N_2^2}+\frac{m_2 r_2^2}{4N_1 N_2}\right]\ddot{\theta}_{m2}+$$

$$\left[\frac{J_2}{N_1 N_2}+\frac{m_2 r_2^2}{N_1 N_2}+\frac{m_2 r_1 r_2}{2N_1 N_2}\left(\cos\left(\frac{\theta_{m1}}{N_1}+\frac{\theta_{m2}}{N_2}\right)\cos\left(\frac{\theta_{m1}}{N_1}\right)+\sin\left(\frac{\theta_{m1}}{N_1}+\frac{\theta_{m2}}{N_2}\right)\sin\left(\frac{\theta_{m1}}{N_1}\right)\right)\right]\ddot{\theta}_{m1}$$

$$+\frac{m_2 r_1 r_2}{2N_1 N_2}\left(\frac{\dot{\theta}_{m1}^2}{N_1}\right)\left[\left(\sin\left(\frac{\theta_{m1}}{N_1}+\frac{\theta_{m2}}{N_2}\right)\cos\left(\frac{\theta_{m1}}{N_1}\right)-\cos\left(\frac{\theta_{m1}}{N_1}+\frac{\theta_{m2}}{N_2}\right)\sin\left(\frac{\theta_{m1}}{N_1}\right)\right)\right]=T_2(t,\theta_{m2},\dot{\theta}_{m2}) \tag{3.16}$$

where the torque functions to be applied to the actuation system of the robot for each joint.

The equations (3.13) to (3.16) are coupled, and have to be solved simultaneously. Each equation contains θ_{m1} and θ_{m2} as the motor's angular displacements in which the dynamic behavior will be effected by each other.

3.4. Numerical Integration

A system of simultaneous ordinary differential equations are solved by 4th order Runge Kutta method to perform a numerical simulation of Serpent 1 [22, 24, 27]. The mechanical and electrical system equations are coupled together with the motors torque equation. So the movement of the arm in a desired position requires simultaneous solutions of the motor and the dynamic system equations for each simulation interval. The method; 4th order Runge Kutta is a stable one, the change in time step can be implemented easily. So by choosing small time step, the integration can give more accurate results, but computational time certainly increases.

Equations (3.13) through (3.16) represent the systems equations. Four equations are written for the whole system. A set of state variables are introduced, and equations (3.13) and (3.16) are expressed as state-variable equations of nonlinear character. Six first order equations are resulted by taking dynamics of dc servo motor for the main arm and the fore arm. The 4th order Runge Kutta is applied step by step , and the value of the function can be computed. The solution can trace out the trajectory at the end.

A computer program is written in Pascal for simulation purposes. To obtain solutions for system ODEs available with numerical integration, the system constants; Serpent 1 link lengths, link masses, link moment of inertias and the initial conditions for the angular displacement, velocity and the motor currents are to be specified. Thus, the initial values; motor currents, motor angular displacements and velocities are set as inputs. The performance of integration scheme depends on the nature of the functions being applied. Whole motion (picking and placing) is performed in 3 seconds. Integration step size Δt is taken 0.731 milli seconds herein. The flow chart of the computer simulation based on equations (3.13) to (3.16) is shown in Figure 3.3.

The addition of the controller is possibly improved the transient and steady-state responses during simulation. It is observed that the proportional control (K_p) is very effective on the overall system stiffness and the derivative control (K_d) effects the overall system damping. So by properly choosing K_p and K_d, the decay of oscillations are controlled easily.

Three operations are performed successively during the experimental studies. A series of simulation runs representing different cases are performed to realize the model validity. Simulation results are presented in the next section.

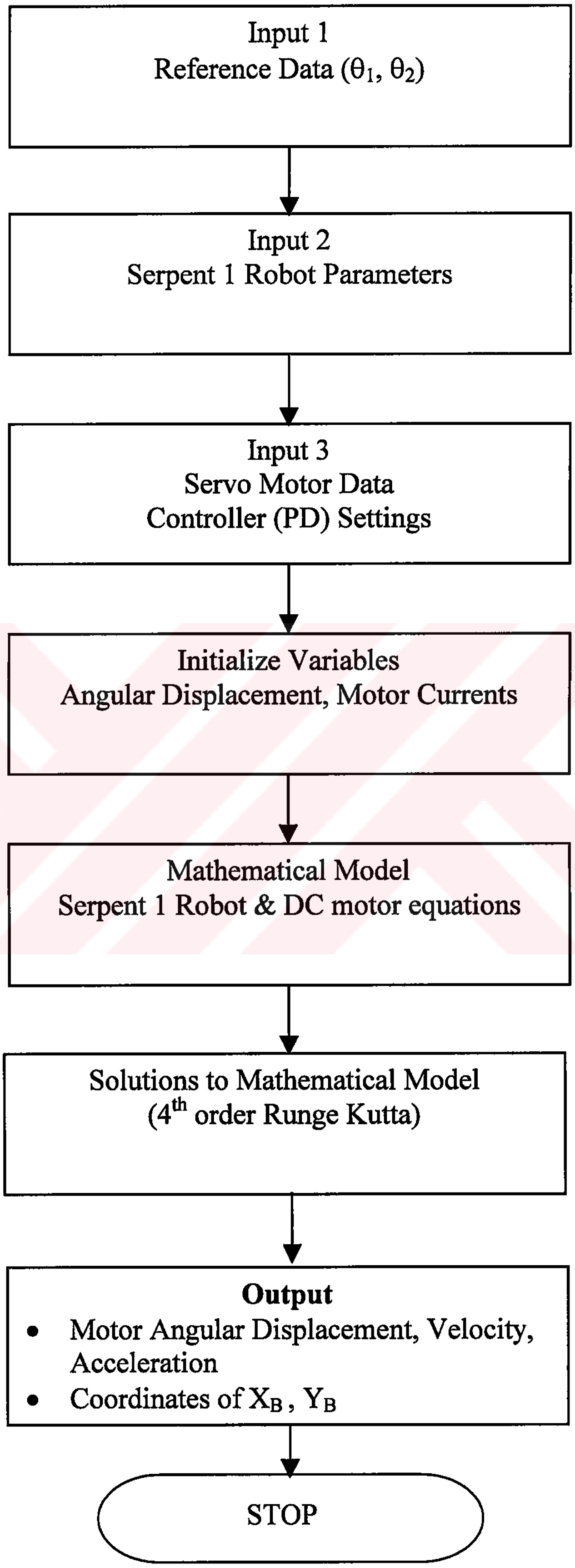


Figure 3.3. Flow Chart of The Computer Simulation for Serpent 1 Robot Actuator System

3.5. Simulation and Experimental Results

Here simulations are carried out together with experiments. Experimental data points are taken as motion reference points for angular positions of the main arm and the fore arm. The developed model is studied with different possible simulations. The solutions can show each possible case behaving differently in the responses. The controller gains; K_p and K_d are set by looking at dynamic behavior of the system. All angular position values used are converted into the assigned angular orientation by referring to the working envelope of Serpent 1 during simulation. Three simulation cases are presented here as the following.

Initially, the main arm is tried to be held constant and the motion of the fore arm is observed. Figure 3.4 represent the simulated and the experimental displacement results for this case. Figures 3.4(a) and (b) give the experimental and simulated results for the main arm and the fore arm respectively. Whole trajectory is shown in Figure 3.4(c). Experimental and simulated trajectories are given in thick and dotted lines. The vertical axis is given in radians and the horizontal axis in seconds for Figures 3.4(a) and (b). In Figure 3.4(c), both axes are given in metres representing the coordinates of the end effector as x_B and y_B.

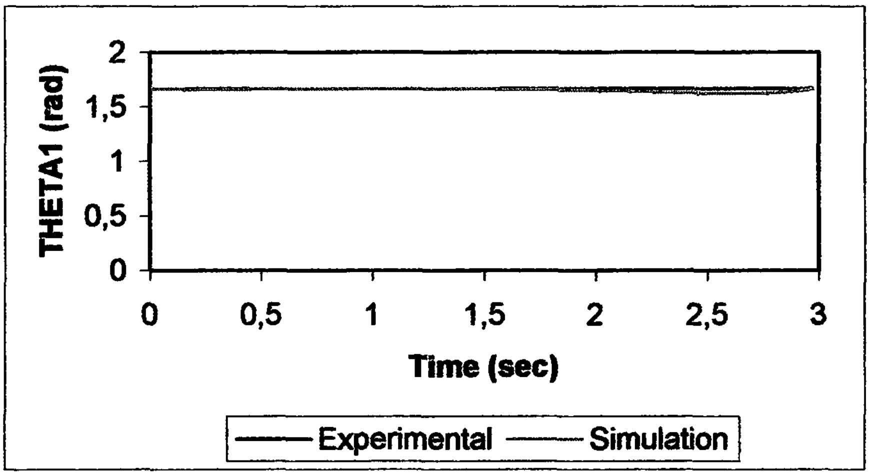
The main arm is kept constant at θ_1=94.5° by sending a constant armature voltage. The fore arm, θ_2 is continuously changing. Initial and final positions of the fore arm are θ_{2i}=151.84° and θ_{2f}=384.84°. The proportional and derivative control gains are found K_{p1}= 300, K_{d1}=50 for the main arm, and K_{p2}=30, K_{d2}=15 for the fore arm respectively. Different simulation runs are performed by changing controller gains K_p and K_d for both motors. Since simplifying assumptions are taken, quite good following characteristics are obtained at the end. This example can be considered as a swinging arm.

Secondly, the motion of the main arm is observed, and the fore arm is tried to be kept constant. Figure 3.5 represent the simulated and the experimental results for this case. Figure 3.5(a) and (b) give the simulated and experimental results for the main arm and the fore arm respectively. Whole trajectory performed for this example is shown in Figure 3.5(c). Thick and dotted lines represent the experimental and the simulated results respectively. The vertical axis is given in radians and the horizontal axis in seconds for Figures 3.5(a) and (b). In Figure 3.5(c), both axes are given in metres representing the coordinates of the end effector as x_B and y_B.

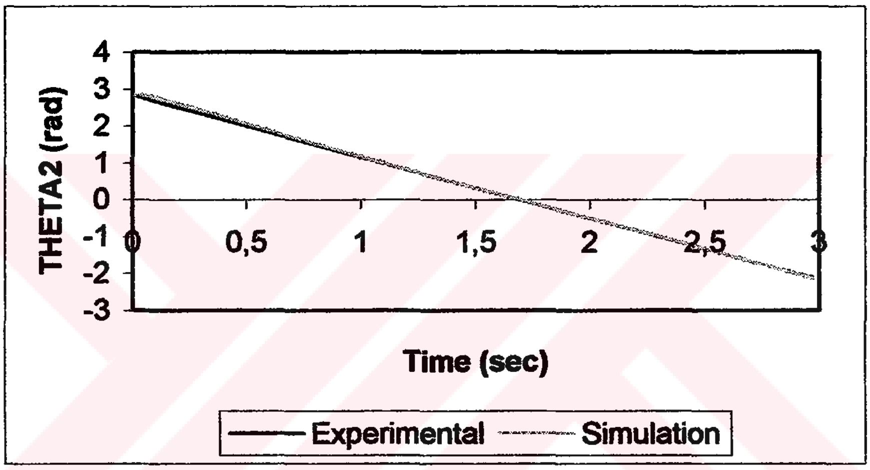
The main arm is continuously changing and the fore arm is kept constant at θ_2=140.35° by sending a constant armature voltage. Initial and final angles measured for the main arm are θ_{1i}=172.70° and θ_{1f}=45.50°. These values are converted into the orijinal positioning values by using the working envelope. In this example, having given initial conditions, nearly 1/3 of the whole motion oscillatory behavior is observed during simulation. By increasing the gain values respectively, transients are settled down. This is given to show trainsients in the response. Steady-state behavior is seen nearly in 1 seconds. This represents more difficult sitution compared to the 1st one. The proportional and derivative control gains are found K_{p1}=300 , K_{d1}=50 for the main arm, and K_{p2}=170, K_{d2}=30 for the fore arm respectively.

Finally, the motions of the main arm and the fore arm are observed, examplifying the case when Serpent 1 works during an assembly operation. This is considered when Serpent 1 to be used while picking and placing a large object referred during experiments, also performing two axis positioning. Figure 3.6 represent the experimental and the simulated results for this case. Figures 3.6(a) and (b) give the experimental and the simulated results for the main arm and the fore arm respectively. Whole trajectory performed is shown in Figure 3.6(c). Experimental and simulated trajectories are given in thick and dotted lines. The vertical and the horizontal axes are given in radians and seconds in Figures 3.6(a) and (b) respectively. In Figure 3.6(c), both axes are given in metres representing the coordinates of the end effector as x_B and y_B.

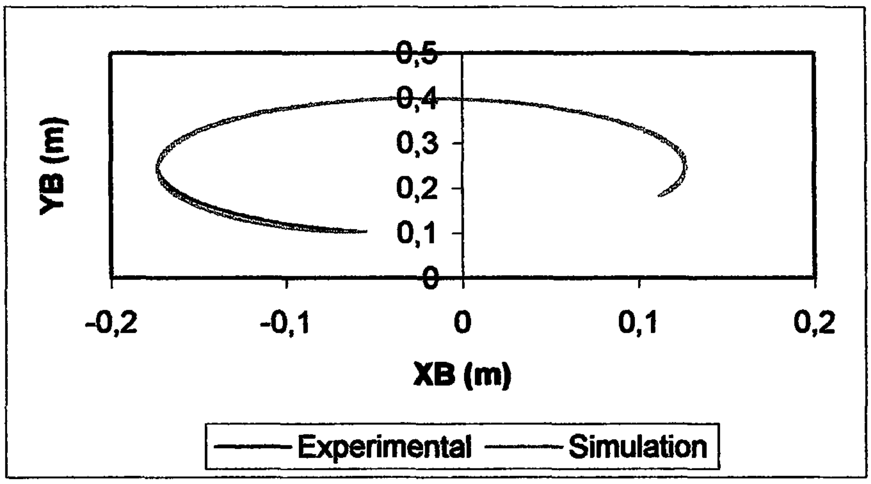
Initial and final positions for the main arm and the fore arm; θ_{1i}=13.28° ,θ_{1f}=141.47° and θ_{2i}=77.47° and θ_{2f} = 185.99°. The proportional and derivative control gains are found K_{p1}=400, K_{d1}= 25 for the main arm, and K_{p2}=60, K_{d2}=15 for the fore arm respectively. Both simulation and experimental responses of Serpent 1 match reasonably good by considering highly nonlinear characteristics of the robot arm. The difference between both results is simply caused by assumptions made while developing the mathematical model. Here coordinates of pick and place points are very important for exact positioning. This is certainly performed.



(a)



(b)



(c)

Figure 3.4. Experimental and Simulated Results(1st Example)

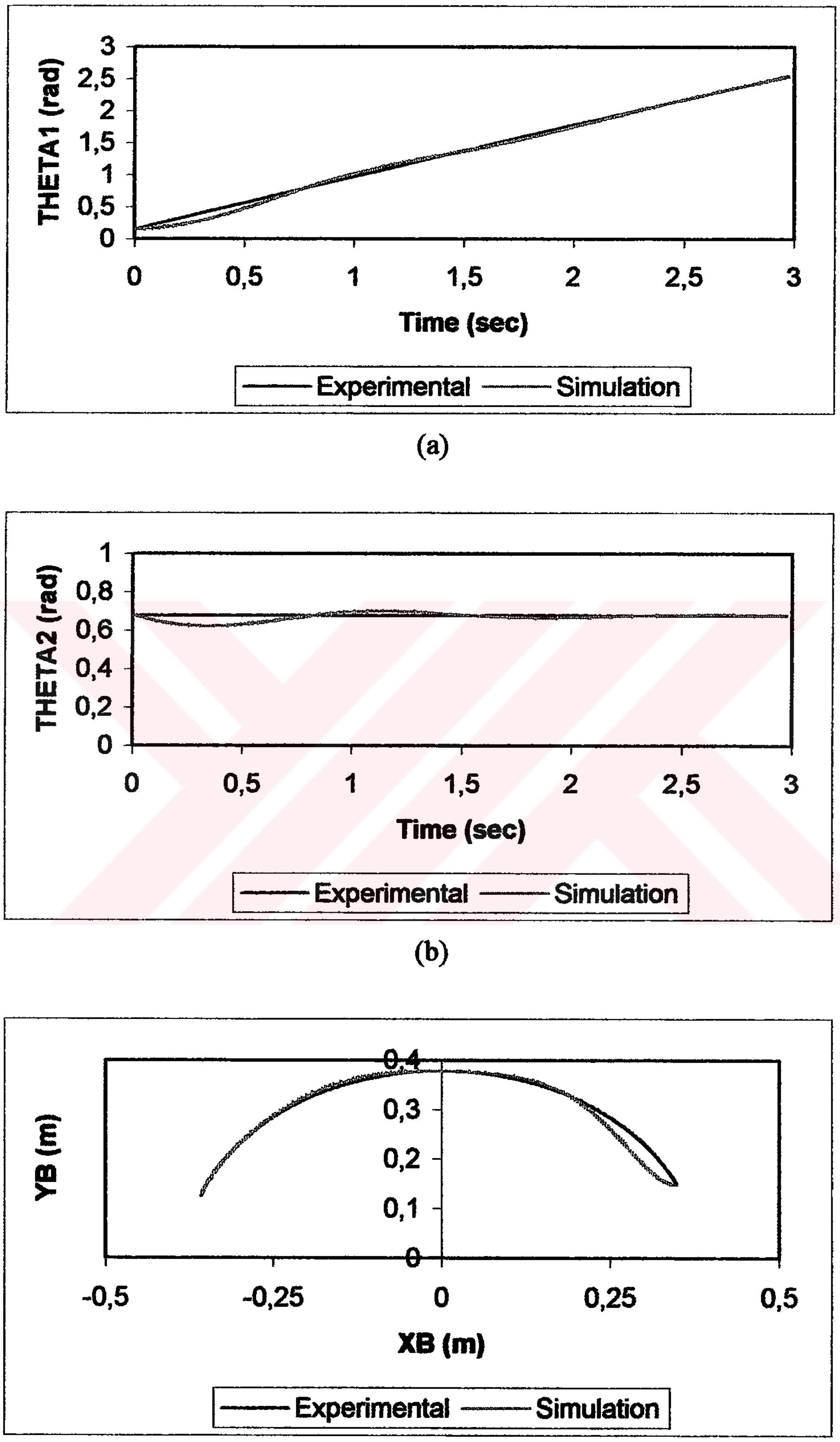


Figure 3.5. Experimental and Simulated Results(2nd Example)

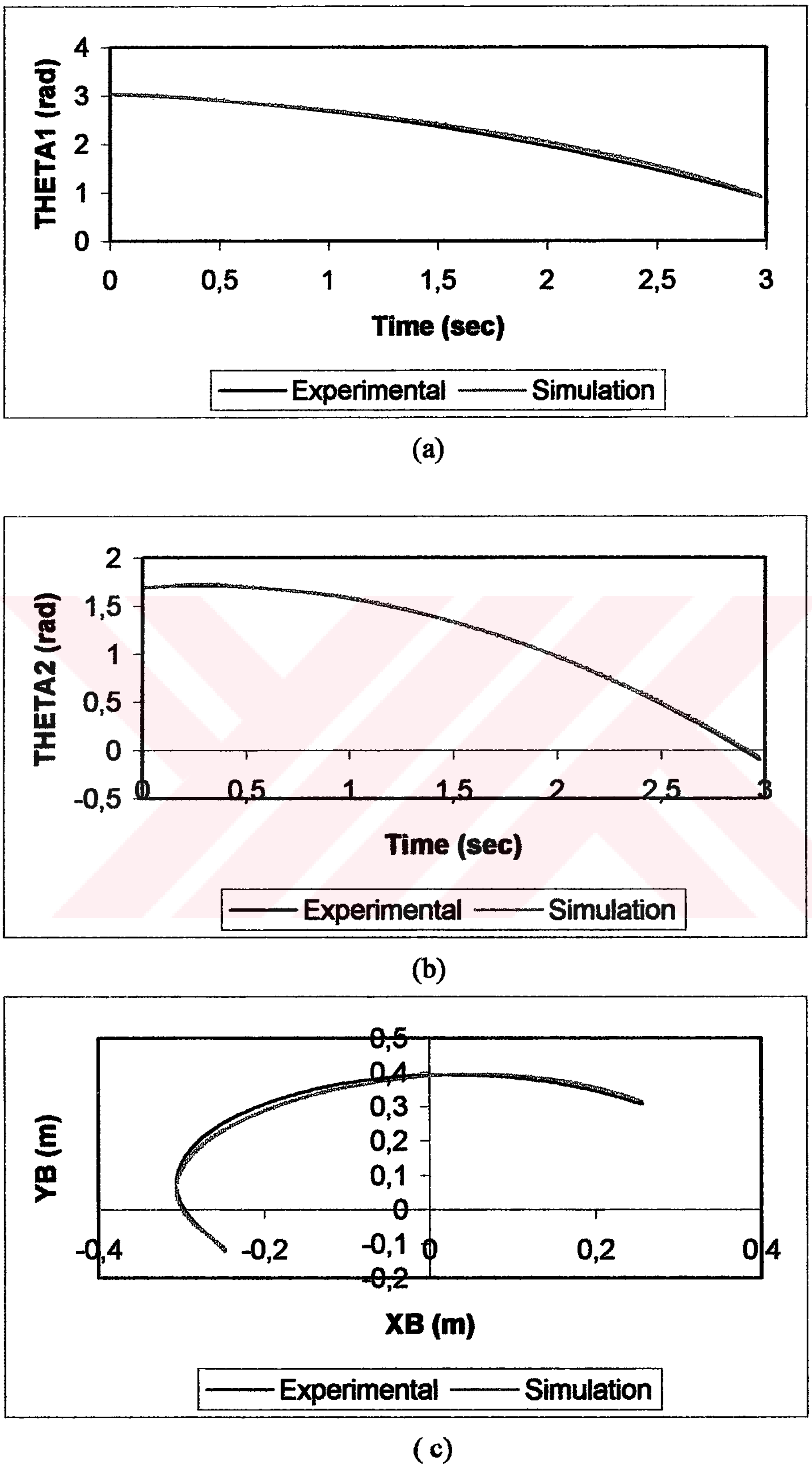


Figure 3.6. Experimental and Simulated Results(3rd Example)

CHAPTER 4

PROGRAMMABLE LOGIC CONTROLLER (PLC)

4.1 Introduction

Control engineering has evolved over time. In the past, humans were the main method for controlling a system. More recently electricity has been used for control and early electrical control was based on relays. These relays allow power to be switched on and off without a mechanical switch. It is common to use relays to make simple logical control decisions. The development of low cost computer has brought the most recent revolution, the Programmable Logic Controller (PLC). The advent of PLC began in the 1970s, and has become the most common choice for manufacturing controls [15].

The PLC is a device, which is especially designed to receive input signals and send output signals with respect to the program algorithm. PLCs come in many shapes and sizes from small, self-contained modules with limited input-output capacity to large, PLC units can be configured to provide hundreds or thousands of inputs/outputs for the required systems. In industrial applications, PLCs are often defined as miniature industrial computers that contain hardware and software that are used to perform control functions [29].

Nowadays, PLC technology has advanced, so have the programming languages and communication capabilities, along with many other important features. Today's PLCs offer faster scan times, space efficient high-density input/output systems, and special interfaces to allow non-traditional devices to be attached directly to the PLC. Not only can they communicate with other control systems, but they can also perform reporting functions and diagnose their own failures, as well as the failure of a machine or process. Size is typically used to categorise today's PLC, and is often an indication of the features and types of applications it will accommodate. Small, non-modular PLCs (also known as fixed I/O PLCs) generally have less memory and accommodate a small number of inputs and outputs in fixed configurations. Modular

PLCs have bases or racks that allow installation of multiple I/O modules, and will accommodate more complex applications [30, 31, 32].

Basic advantages of PLC control [15];

- Cost effective for controlling complex systems.
- Flexible and can be reapplied to control other systems quickly and easily.
- Computational abilities allow control that is more sophisticated.
- Trouble shooting aids make programming easier and reduce downtime.
- Reliable components make these likely to operate for years before failure.

4.2 PLC Operation

Parts of PLC including power supply, central processing unit, input and outputs, status lights and information on scan cycle are included in this section.

4.2.1 Parts of PLC

Many PLC configurations are available in a form of a unique vendor. The most general components of PLCs are power supply, CPU, I/O and status lights

Power Supply - This can be built into the PLC or be an external unit. Common voltage levels required by the PLC (with and without the power supply) are 24Vdc, 120Vac, 220Vac. For this study, 220V ac power supply and 24V dc inputs and outputs are used with Siemens S7-200.

CPU (Central Processing Unit) – executes the program that is stored and processed for controlling the automation task.

I/O (Input/Output) - A number of input/output terminals must be provided so that the PLC can monitor the process and initiate actions.

Status lights - These indicate the status of the PLC including modes of run, stop and fault of program. These are essential when diagnosing problems.

The configuration of the PLC refers to the packaging of the components. Typical configurations are listed as micro, mini, rack, shoebox and software. In Figure 4.1, the mostly used configurations of PLC packages as micro, mini and rack can be seen [14,15,29].

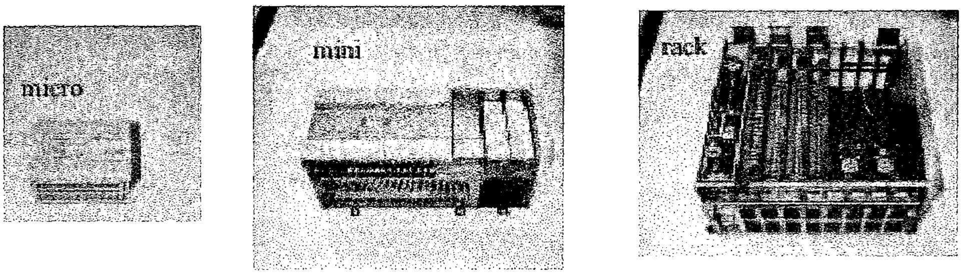


Figure 4.1.General configurations of PLC

Micro - These units can be as small as a deck of cards. They tend to have fixed quantities of I/O and limited abilities, but costs will be the lowest.

Mini - These are similar in function to PLC racks, but about half the size.

Rack - A rack is often large (up to 18” by 30” by 10”) and can hold multiple cards. When necessary, multiple racks can be connected together. These tend to be the highest cost, but also the most flexible and easy to maintain

Shoebox - A compact, all-in-one unit (about the size of a shoebox) that has limited expansion capabilities. Lower cost, and compactness make these ideal for small applications.

Software - A software based PLC requires a computer with an interface card, but allows the PLC to be connected to sensors and other PLCs across a network.

4.2.2 Inputs and Outputs

Inputs (sent to), and outputs (come from), a PLC are necessary for monitoring and controlling of a process. Both inputs and outputs can be categorized into two basic types; logical or continuous. An example of a light bulb, which is connected to a switch can be considered. If the light bulb can only be turned on or off, it is “logical control”. If the light can be adjusted to different levels, it is “continuous control”. Continuous values seem more intuitive, but logical values are preferred because they allow more certainty, and simplify control. Consequently, many control applications (and PLC’s) use logical inputs and outputs.

The most general input and output elements that are used in all industrial areas and applications are actuators, sensors, power supplies and control units. *Actuators* include electric motors, solenoids, pneumatic and hydraulic cylinders. *Switching devices* include valves, relays or contactors.

4.2.3 Scan Cycle

The PLC executes a series of tasks repetitively This cyclical execution of tasks is called the scan cycle. The PLC performs all of the following tasks during a scan cycle as shown in Figure 4.2.

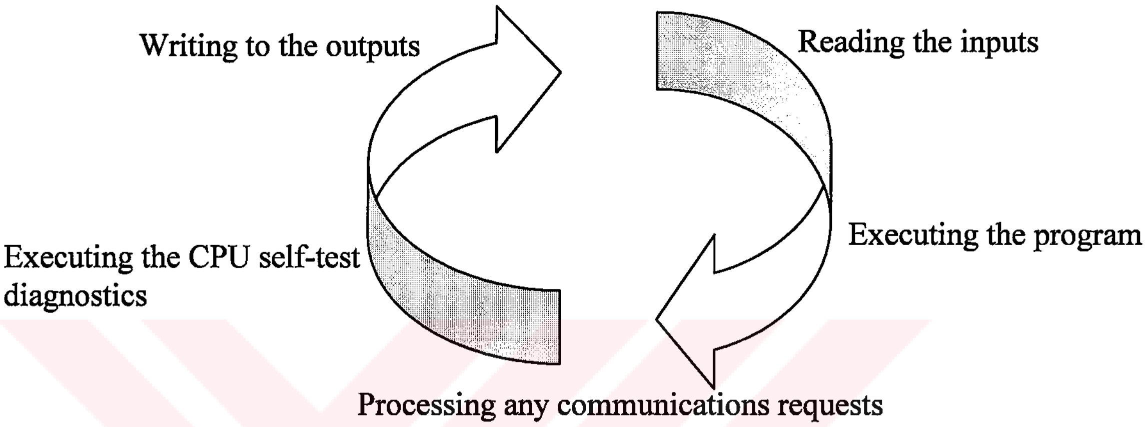


Figure 4.2. Scan cycle of PLC

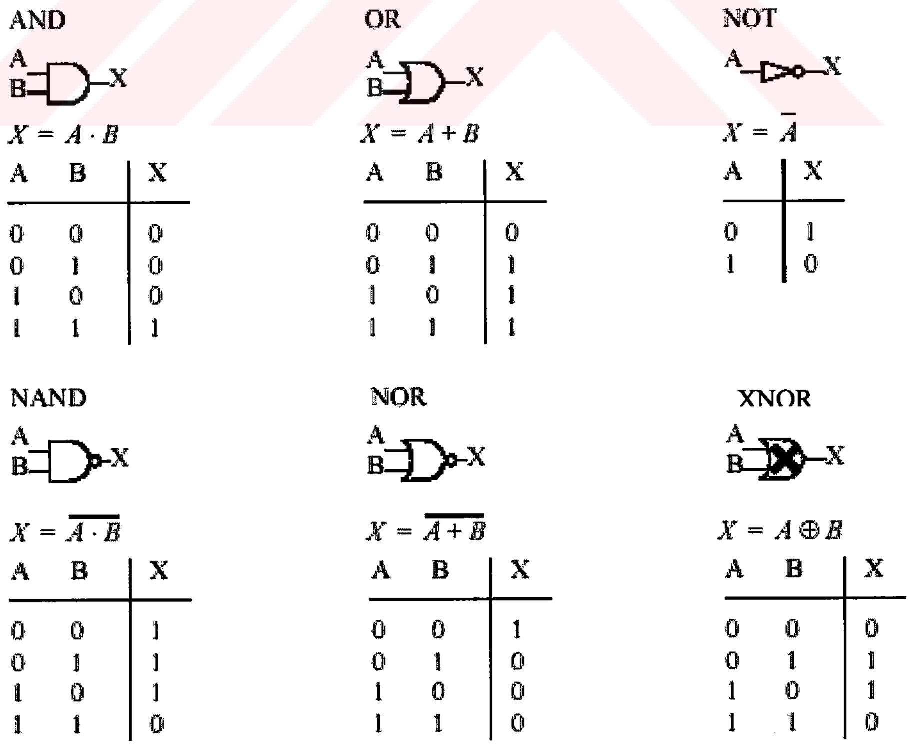
All PLC's have four basic stages of operations that are repeated in many times per second. First of all when turned on the first time it will check it's own hardware and software for faults. If there are no problems it will copy all the input and copy their values into memory, name of the process is "the input scan" that is stored on input process image. Using only the memory copies of the inputs, the ladder logic program will be solved once, this is called "the logic scan" that is execution of program. While solving the ladder logic the output values are only changed in the temporary memory. When the ladder scan is done the outputs will be updated by using the temporary values in memory, this is called "the output scan" that is stored as the output process image. The PLC now restarts the process by starting a self-check for faults. This process typically repeats 10 to 100 times per second [15,29].

4.3 PLC Functions and Programming Techniques

Functions of PLC, logic circuit, latches, timers, counters, subroutines, and programming techniques are included in this section.

4.3.1 Logic Circuit

Before defining some basic PLC functions and programming, first of all, logic circuits, relays and latches are introduced with some examples. Simple hardwired circuit contains very straightforward "logic". Switch and lamp is a primitive example; when the switch is on, the lamp is on and when the switch is off, the lamp is off, that is logic is 'AND'. When two switches are connected to parallel to a lamp. Which one or both switches are on, lamp is on, If two switches are off, then lamp is certainly off. That logic is 'OR'. Inputs can also be expressed in two types as normally open or normally closed. Number of examples can be increased for logic circuits, therefore, name and symbol of simply used ones are given. Logic functions, AND, OR, NOT, NAND, NOR, XNOR can be seen in Figure 4.3 [15]. The translations of these logic circuits to a ladder diagram are the base of PLC programming.



AND

$X = A \cdot B$

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

OR

$X = A + B$

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

NOT

$X = \bar{A}$

A	X
0	1
1	0

NAND

$X = \overline{A \cdot B}$

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

NOR

$X = \overline{A + B}$

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

XNOR

$X = A \oplus B$

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Figure 4.3 Illustration of logic functions

In the following example, translation of logic to the ladder diagram can be seen clearly. A program can be developed that will cause output D to go true when switch A and switch B are closed or when switch C is closed. The ladder diagram of the example can be seen in Figure 4.4. A and B is normally opened series contact that means AND logic. C is connected parallel to them, it is OR logic.

D= (A*B)+C

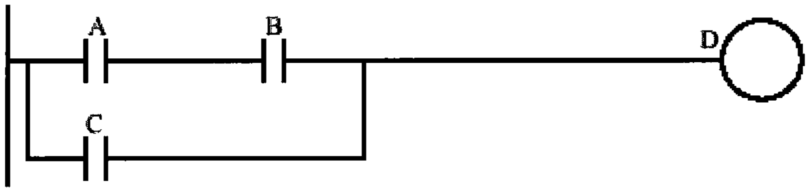


Figure 4.4 Ladder diagram of the example

More complex systems can not be controlled with combinatorial logic alone. The main reason for this is that it cannot be happened, or choose not to add sensors to detect all conditions. In these cases, events can be used to estimate the condition of the system. Typical events are used by a PLC include;

- First scan of the PLC - indicating the PLC has just been turned on
- Time since an input turned on/off - a delay
- Count of events - to wait until set number of events have occurred
- Latch on or unlatch - to lock something on or turn it off

4.3.2 Latches

Latch is like a sticky switch, also this circuit is a memory circuit. That is preferred to control the power fed to machine tools. Working principle is that the switch starts the motor that continues to motion when the switch is released. Other name of the circuit is "hold-on" circuit [14]. This circuit is used to save system from initial high voltage. Figure 4.5 shows an example of the latch circuit.

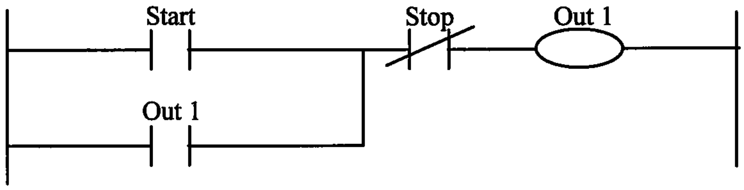


Figure 4.5 Latch Circuit

4.3.3 Timers

Timers are used to implement time-based counting functions. Siemens S7-200 series provide two different timer instructions; the On-Delay Timer (TON), and the Retentive On-Delay Timer(TONR). Both of them count time up while the given input is on. The timers do not count time up when the given input is off. Main difference between them can be seen when the input is off. TON timer is reset automatically, however TONR timer is not reset and saves its last recorded value. Therefore, the TON timer is the best to use when timing interval is single. The TONR timer is suitable for accumulating a number of intervals.

Timers are controlled with a single enabling input, and have a pre-set time value that is compared to the current value, each time the current value is updated when the timer initialisation is executed. Finally, timers become active when the current value is bigger than pre-set value. Timers can be reset by using reset instruction of timer. In S7-200 PLC, there are three types of resolutions of timing up which are 1-ms, 10-ms, 100-ms. In Table 4.1, numbers and resolutions of timers are illustrated for CPU 216 PLC between T0-255. The programming timer instructions, used in programming of PLC are given in Figure 4.6 [33].

Table 4.1 Timer numbers and resolutions

Timer	Resolution(ms)	Maximum value (second)	CPU 216
TON	1 ms	32.767 s	T32, T96
	10 ms	327.67 s	T33 to T36 T97 to T100
	100ms	3276.7 s	T37 to T63 T101 to T255
TONR	1 ms	32.767 s	T0, T64
	10 ms	327.67 s	T1 to T4 T65 to T68
	100 ms	3276.7 s	T5 to T31 T69 to T95

Figure 4.6. Timer instructions for S7-200 PLC

4.3.4 Counters

There are two types of counters; the *count up* (CTU) and the *count up-down* (CTUD) for general applications. The CTU instruction counts up to the maximum value of the count up input. Also, in similar with timers, the counter bit turns on when the current value become greater than or equal to the pre-set count value. For CTUD, there are two different input values, the first one is the necessary for counting up and the second input value is required for counting down. The counter bit becomes active when the current value is greater or equal to the pre-set value. Reset input is existed for both counters in order to inactivate them. Typical counter instructions are given in Figure 4.7 [33].

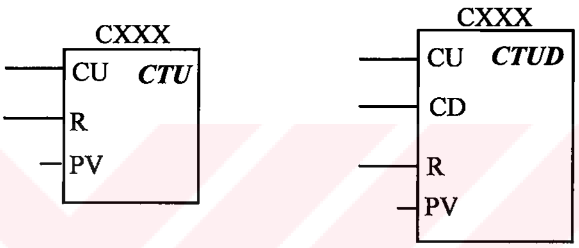


Figure 4.7 Counter instructions for S7-200 PLC

4.3.5 Subroutines

This optional element of PLC program is executed only when called; by the main program, by an interrupt routine, or by an another subroutine. Subroutines are useful in cases where a function is executed repeatedly. Instead of rewriting the logic for each place in the main program where the function is occured, the logic is once written in a subroutine, and called as many times as needed during the main program. Subroutines provide several advantages of reducing the overall size of program and decreasing scan time with removing the excess code from the main program, creating code that is portable.The S7-200 evaluates the code in the main program for every scan cycle, whether the code is executed or not. However, the S7-200 evaluates the code in the subroutine only when it is called, and does not evaluate the code during the scans in which the subroutine is not called [14,15] .

4.4 Routine Applications with PLC

In the following paragraphs, three different examples are presented to realize routine applications of PLC. First two examples represent a pneumatic system and last one utilizes PLC as a motor switching element. All PLC programs are written in Micro/Win.

In the first example one pneumatic cylinder, two limit switches and a solenoid valve are used. The cylinder is extended initially when input switch is opened and cylinder comes to A(+) position after touching to the limit switch S1, the cylinder retracts, which means A(-) and touching S2 limit switch, again the cylinder goes to the extend position. This cycle continues until the input switch is closed. The circuit is shown in Figure 4.8.

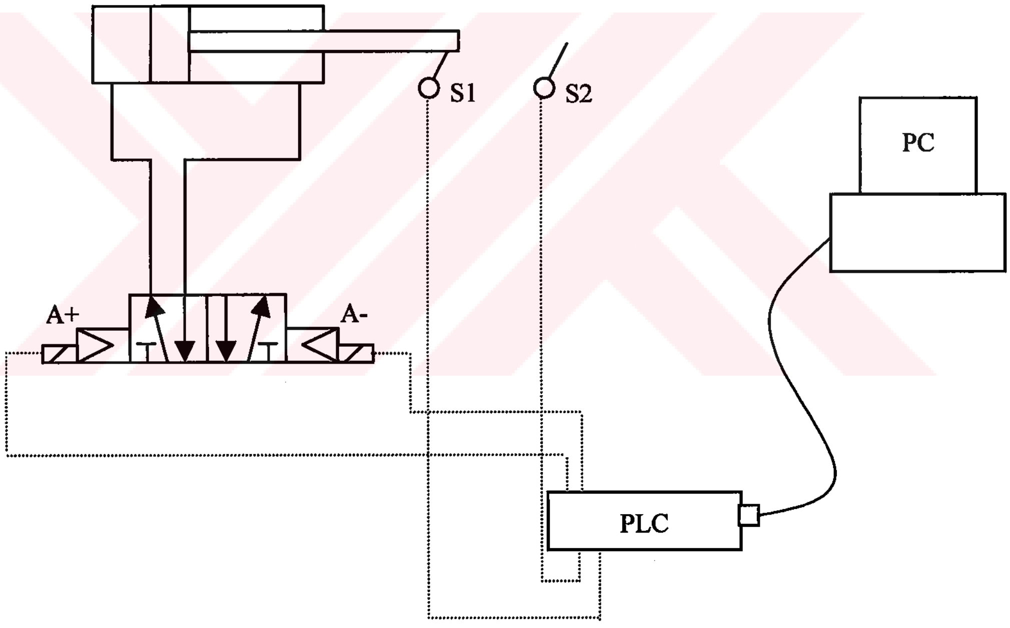


Figure 4.8 A Pneumatic cylinder with PLC control

In the second example, two pneumatic cylinders, two solenoid valves, a contactor and a switch are used in the circuit. The cylinders A and B are operated in a sequence of A(+) B(+) B(-) A(-) repetitively. Initial movement of circuit is provided by electrical switch and the cylinder A extends to A(+) then the cycle finished when switch is closed. This circuit can be seen in Figure 4.9.

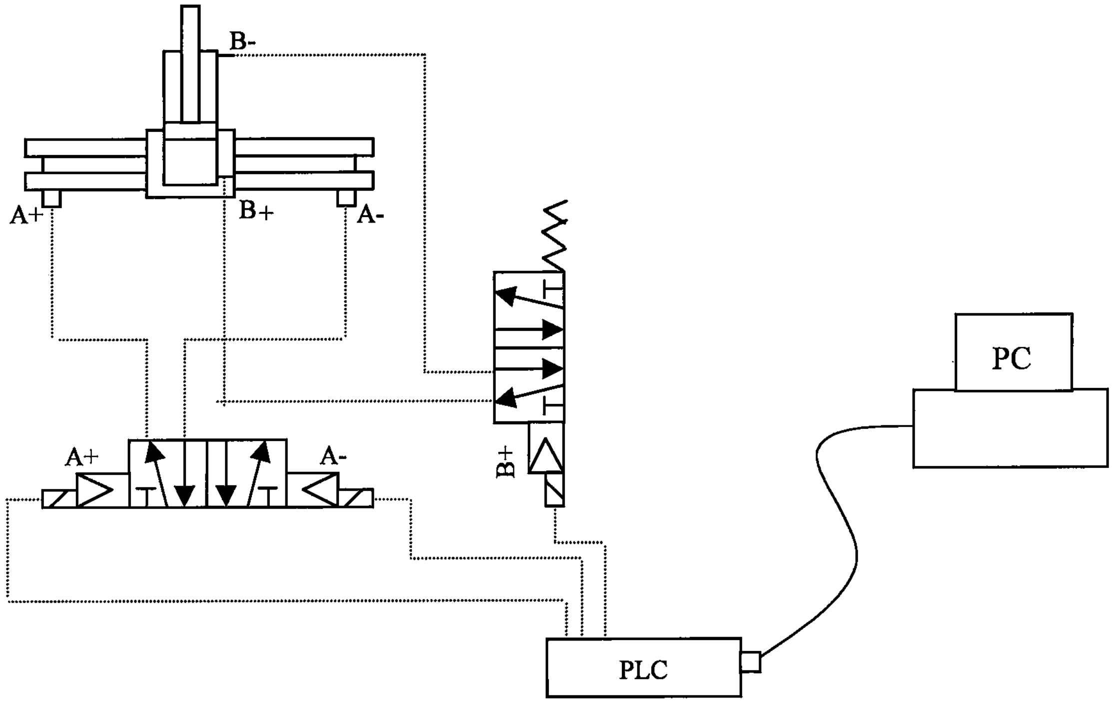


Figure 4.9 Two Pneumatic Cylinders with PLC control

In the above examples pneumatic systems are used, however in hydraulic systems or in great numbers of components for both of circuits can be also used with PLC control. Program for the pneumatic system is;

```
NETWORK 1 //NETWORK TITLE
(pneu)
//
//NETWORK COMMENTS
//
LD    I0.0
LPS
TON    T32, +5000
AN    T32
AN    Q0.2
=    Q0.0

LRD
A     T32
TON    T33, +600
LPP
A     T33
AN    Q0.0
=    Q0.2

NETWORK 2
MEND
```

PLC's can be also used for controlling an electric motor (CW and CCW). A direct current (DC) electrical motor can be driven in different ways with different time intervals. In Figure 4.10 schematic representation of the system is shown.

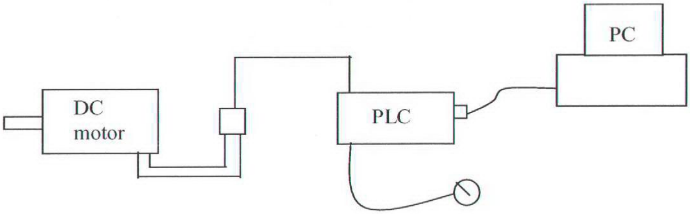


Figure 4.10 A DC motor control with PLC

4.5 Linear Positioning with PLC

Two different linear axes are driven by two stepper motors, which are controlled by PLC. A modular system is designed and produced for this application. The required position is written in PLC program and the driven cart has moved to exact position. Related PLC program is given in the following page. The photograph and the schematic representation of the system can be seen in Figure 4.11 and Figure 4.12.

Figure 4.11 Photograph of Linear Motion System

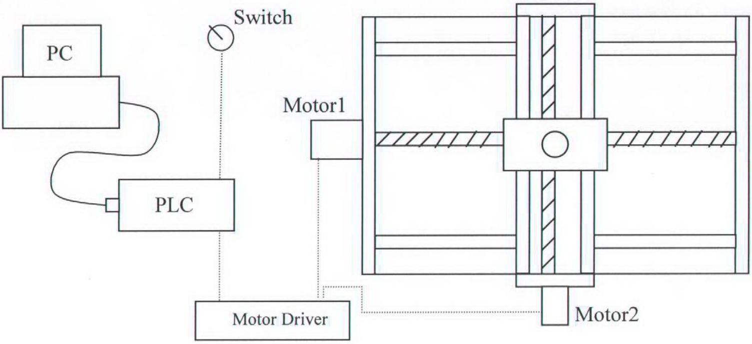


Figure 4.12 Schematic Representation of Linear Motion System

```
NETWORK 1 //Linear  Control
//
//NETWORK COMMENTS
//
LD    I0.0
AN    T33
AN    C0
TON   T32, +10
NETWORK 2
LD    T32
AN    T34
TON   T33, +5
NETWORK 3
LD    T33
TON   T34, +10
NETWORK 4
LD    T32
=     Q0.0
NETWORK 5
LD    Q0.0
LD    I0.1
CTU   C0, +150
NETWORK 6
LD    C0
AN    T39
AN    C1
TON   T36, +10
NETWORK 7
LD    T36
AN    T39
TON   T38, +1
NETWORK 8
LD    T38
AN    T40
TON   T39, +2
NETWORK 9
LD    T39
TON   T40, +3
NETWORK 10
LD    T38
=     Q0.2
NETWORK 11
LD    Q0.2
LD    I0.1
CTU   C1, +100
NETWORK 12
MEND
```

4.6 SCARA Robot Control with PLC

The photograph of Serpent 1 with PLC unit is given in Figure 4.13. Schematic representation of Serpent 1 with PLC connections is also given in Figure 4.14. Angular positions of Serpent 1 robot arms are given for three objects in Table 4.2.

Pneumatic system, which includes three solenoid controlled directional control valves, is controlled with PLC. Up-down and open-closed conditions are available in sequence, which is given in Table 4.3. These order becomes active when robot arms are driven and stopped in exact position while picking and placing an object. PLC program for the pneumatic part is given in the following pages.

Figure 4.13 The photograph of Serpent 1 with PLC

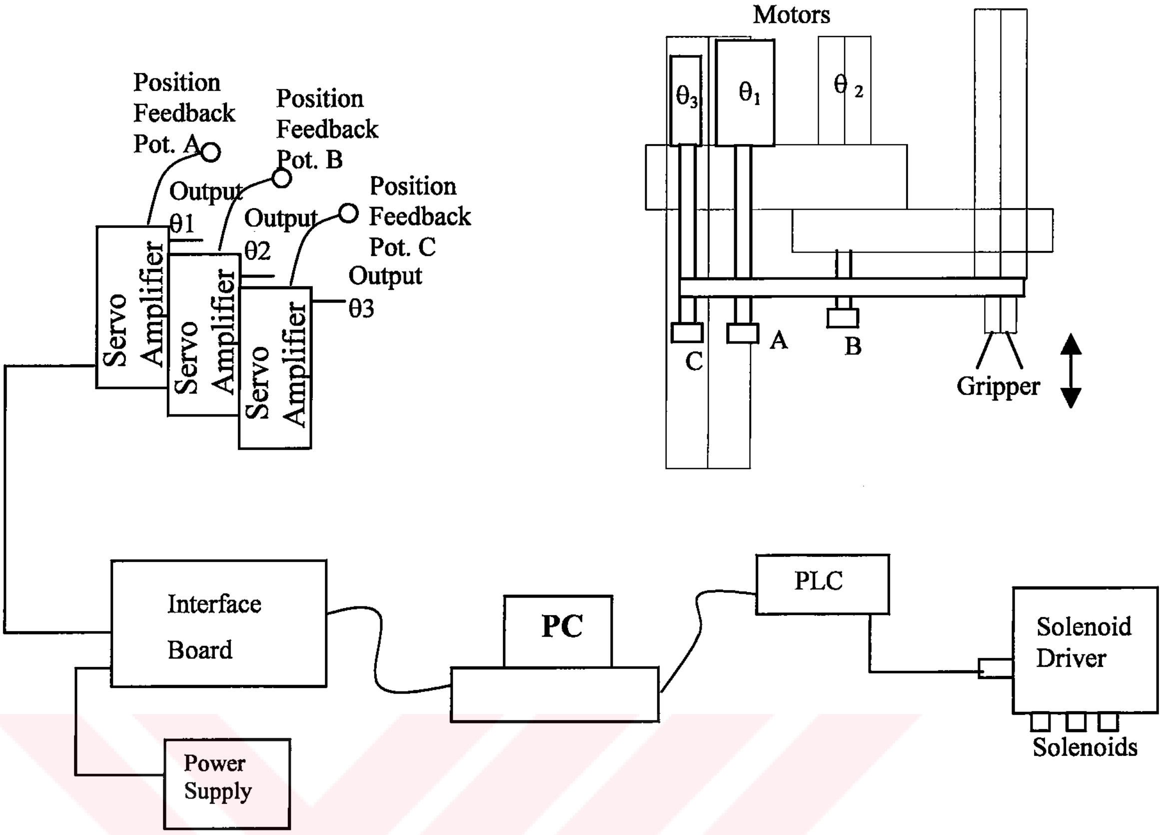


Figure 4.14 Schematic Representation of Serpent 1 with PLC

Table 4.2 Positions of Serpent 1 Robot Arms

	Initial		Final	
	θ_1	θ_2	θ_1	θ_2
Large	42.01°	146.98°	117.9°	226.37°
Medium	32.88°	104.15°	96.77°	200.15°
Small	13.28°	77.48°	141.47°	186°

Table 4.3 Sequence of Pneumatic control structure and PLC program

	1	2	3	4	5	6	7
Up	on	off	off	on	off	off	on
Down	off	on	on	off	on	on	off
Closed	off	off	on	on	on	off	off

```
NETWORK 1
LDN    I0.0
JMP    0

NETWORK 2 //Pneumatic
//
//NETWORK COMMENTS
//
LD     I0.0
AN     Q0.1
O      T34
=      Q0.0

NETWORK 3
LD     Q0.0
O      T33
AN     T35
TON    T33, +350

NETWORK 4
LD     T33
=      Q0.1

NETWORK 5
LD     Q0.1
TON    T37, +10

NETWORK 6
LBL    0
```

```
NETWORK 7
LD     T37
AN     T34
O      T36
ON     I0.0
=      Q0.2

NETWORK 8
LD     Q0.2
O      T34
AN     T36
TON    T34, +1000

NETWORK 9
LD     T34
AN     T36
=      Q0.3

NETWORK 10
LD     Q0.3
TON    T35, +350

NETWORK 11
LD     T35
TON    T36, +200

NETWORK 12
MEND
```

CHAPTER 5

CONCLUSIONS AND DISCUSSIONS

5.1 About the Present Work

This work has presented a study on motion control of a SCARA robot (Serpent 1) with PLC unit. The development of work was concluded with five chapters.

The work was started with introduction on SCARA robots and PLC in automation. A brief literature study was included together with basic information on industrial applications of PLC, programming methods etc. Serpent 1 robot was then studied with its kinematic issues, mechanical, electromechanical data available and possible trajectories for its use as pick and place robot.

A simplified mathematical model was developed as a representation of the system. Lagrange's method of formulation was used to derive the equations of motion. The system equations were solved by using fourth order Runge-Kutta method of integration. Simulation was prepared with Pascal programming language. Different transient and steady-state response characteristics were observed. Simulated and experimental results for Serpent 1 have shown that the mathematical model developed was good enough to understand and explain dynamics of the Serpent 1 with some simplifying assumptions.

Motion control issues were then given with Serpent 1 and S7-200 unit. Initially routine applications of PLC were studied as control of pneumatic cylinders, motor switching elements, and also linear positioning systems with stepper motors. Several examples were investigated for better understanding of PLC units with programs. In the learning of programmable logic controller, a lecture was taken from Electrical and Electronic Eng. Dept. and PLC programming was achieved with the aid of possible laboratory equipment. Programming was not only the main part of study, some technical electrician knowledge were learned while operating the system.

Necessary set up was designed and operated. Having performed routine examples for PLC, the partial adaptation of PLC to Serpent 1 robot system was achieved. Here partial means only the control of pneumatic cylinders. When the exact positioning was performed by dc servo motors, pneumatic action was started and “up-down”, “open-closed” comments were used while picking and placing cylindrical objects. Necessary PLC programming was given with its details.

5.2 Discussion on the Present Work

Serpent 1 robot system and Siemens S7-200 PLC unit are applied in the study. This study can be considered as an industrial application, because of commercial use of the pneumatic systems and industrial robot with PLC control. These systems are very popular in production and assembly lines.

Serpent 1 robot system is already available in Dynamic Systems Laboratory, is equipped with a controller unit, an interface board, a PC and related software for robot programming. By using the software given, it is possible to perform different pick and place operations. The intention of this study was to perform two axes positioning, and pneumatic action by using PLC unit. Here angular positionings of both axes are performed by pm dc servo motors.

However, when the control system is tried to be changed completely, it is not possible to change available connections for the servo system. Motor and controller connection details can not be found. Pneumatic parts (solenoids) and necessary connections are easier to change at this point. Pneumatic parts are completely taken out of the controller unit. Thus it is realized that the package program is already blocked to control any change in the servo motor drivers. So a partial control is performed by dismounting the pneumatic control unit from the main control board. Therefore, PLC is connected to pneumatic part to control “up-down”, “open-close” action. A X-Y linear positioning system has then been designed and produced for illustrating motion control, which is like a base of universal machine. This system is driven with stepper motors. Thus, it will be possible to show two axes positioning respectively by PLC unit with this set-up.

To deal with kinematics and dynamics of Serpent 1, modelling and simulation studies are carried out together. The intention of modelling and simulation is to understand behaviour of the original system Serpent 1 with its linkage configuration and controller details. To do this, some simplifications are considered, and a mathematical model is developed. Experimental data is taken as reference points while performing motor controller action. Thus verification of the model is performed with simulation. The effect of various controller settings on the system response is certainly seen. The best ones are given with final controller settings. Since the requirement is to pick and locate one of the cylindrical object with in tolerance limits, simulation is satisfied that requirement especially at pick and place points.

5.2 Recommendations for the Future Study

To manage Serpent 1 robot system completely, all control system; motion controller, software used and feedback elements (potentiometers) of Serpent 1 can be changed. The reason is that, the system can be fully understood with new programs, advanced control methods; self tuning, adaptive control can easily be applied to Serpent 1 robot system. The control algorithm can be modified. Different modifications can be done on the robot system dealing with controller gains and clearances causing the positioning error. An effective and faster gripper can be designed to the robot, when geometry of the object is changed, such as cubic, or irregular shape etc. Designing a different gripper will be a new research subject to be explored. Weight of the object will be another limitation to be considered during the design.

For PLC usage new generation PLC's and modules can be used like S7-400 series or Simotion, which include every modules itself. In that case, S7-200 units will not be used. Newer PLC modules will have to be bought and applied. This will certainly introduce higher cost for the future research. For the linear positioning system, more powerful motors can be adapted to the system for increasing the efficiency and carrying capacity. Advanced programming techniques can be learned for PLC programming. Control of hydraulic servo valves can be also performed by using PLC unit with available laboratory equipment. Lastly, mathematical model can be modified by inclusion of losses, real effects in the system, and a complete model can be obtained as a potential for future work.

REFERENCES

1) *WALLI Serpent Manual*, 1993

2) Dorf R. C., 1990. Concise International Encyclopedia of Robotics, John Wiley&Sons Canada. *Specifications.* Wiley Interscience.

3) Eugene R. I.,1988. *Mechanical Design of Robots, McGraw Hill.*

4) Dorf R. C. , 1990. Concise International Encyclopedia of Robotics, John Wiley&Sons Canada. *Programming, Off-Line Languages.* Wiley Interscience

5) Nof S. Y. , 1985. *Handbook of Industrial Robotics,* John Wiley&Sons

6) Ertürk M., 1997, *SCARA tipi robot kolunun hareketlerinin bilgisayar destekli programlarla kontrolü,* Y. Lisans Tezi, Çukurova Üniversitesi.

7) Bhatia P., Thirunarayanan J. , Dave N. , 1998, An expert system-based design of SCARA robot, *Expert Systems With Applications, 15,* 99-109.

8) Ge S.S. , Lee T. H., Gong J. Q. , 1999, A robust distributed controller of a single-link SCARA/Cartezian smart materials robot, *Mechatronics, 9,* 65-93.

9) Omodei A. , Legnani G. , Adamini R. , 2000, Three methodologies for the calibration of industrial manipulators: Experimental results on a SCARA Robot, *Journal of Robotic Systems, 17(6),* 291-307

10) Er M.J. , Lim M. T. , Lim H. S. , 2001, Real-time hybrid adaptive fuzzy control of a SCARA robot, *Microprocessors and Microsystems, 25,* 369-378.

11) Hong K.S. , Choi K. H. , Kim J. G. , Lee S. , 2001, A PC-based open robot control system; PC-ORC, *Robotics and Computer Integrated Manufacturing, 17,* 355-365.

12) Nagchaudhuri A. , Kuruganty S. , Shakur A. , 2002, Introduction of mechatronics concepts in a robotics course using an industrial SCARA robot equipped with a vision sensor, *Mechatronics, 12,* 183-193.

13) Miro J. V. , White A. S. , 2002, Modelling an industrial manipulator a case study, *Simulation Practise and Theory, 9,* 293-319.

14) Clements-Jewery K. , Jeffcoat W. , 1996. *The PLC WORKBOOK,* Prentice Hall Europe.

15) Jack H., 2002. *Automating Manufacturing Systems with PLCs,* Copyright By Hugh Jack

16) Lu H. , Ying Z. , Liao T. W. , 1992, Simulation of programmable controller, *Computers and Industrial Engineering, 23,* 351-354

17) Starr A. G. , Hu W. , Leung A. Y. T. , 1999, Two diagnostic models for PLC controlled flexible manufacturing systems, *International Journal of Machine Tools& Manufacture, 39,* 1979-1991.

18) Figliolini G. , Ceccarelli M. 1999, Walking programming for an electropneumatic biped robot, *Mechatronics, 9,* 941-964.

19) Pfeiffer B-M., 2000, Towards 'plug and control': Self-tuning temperature control controller for PLC, *International Journal of Adaptive Control and Signal Processing,14,* 519-532.

20) Ravina E. , 2001, A pneumotronic equipment for acoustic measurements, *Mechatronics, 11,* 185-197

21) Burton T. D. , !994, *Introduction to Dynamic System Analysis,* McGraw-Hill

22) Vu H. V. , Esfandiari R. S. , 1998, *Dynamic Systems Modelling and Analysis,* McGraw-Hill

23) Krause P. C. , Wasynczuk O. , *Electromechanical Motion Devices,* McGraw-Hill

24) D'Souza A. F. , 1984, *Advanced Dynamics Modeling and Analysis,* Prentice Hall

25) Ogata K. , 1997, *Modern Control Engineering,* Prentice Hall

26) Kuo B. C. , 1995, *Automatic Control Systems,* Prentice Hall

27) Chapra S. C. , Canale R. P. , 1998, *Numerical Methods for Engineers,* McGraw-Hill.

28) Holman P. , 2001, *Experimental Methods for Engineers,* McGraw-Hill.

29) Petruzella F. D., 1996, *Industrial Electronics,* McGraw-Hill,Glencoe

30) Fraser C. ,1994, *Integrated Electrical and Electronic Engineering for Mechanical Engineers,* McGraw Hill.

31) Histand M. B. , 1999, *Introduction to Mechatronics and Measurement Systems,* McGraw Hill.

32) www.automationdrive.com

33) Siemens 1998, *S7-200 Programmable Controller System Manual,*_Siemens AG.