1-1-2000

Control of multiple tele-operated Robotic Bridge Transporters for remote handling of hazardous material

T. R Venkatesh
University of Nevada, Las Vegas

Repository Citation

Venkatesh, T. R, "Control of multiple tele-operated Robotic Bridge Transporters for remote handling of hazardous material" (2000). *UNLV Retrospective Theses & Dissertations*. 3136.
http://dx.doi.org/10.25669/vnh4-o134

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

Control of Multiple Tele-operated Robotic Bridge Transporters for Remote handling of Hazardous Material

by

T. R. Venkatesh

A thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Mechanical Engineering

Department of Mechanical Engineering
University of Nevada, Las Vegas
May, 1996

The thesis of T. R. Venkatesh for the degree of Master of Science in Mechanical Engineering is approved.

Chairperson, Woosoon Yim, Ph.D

M B Trabia

Examining Committee Member, Mohamed Trabia, Ph.D

BJ O'Toole

Examining Committee Member, Brendan O'Toole, Ph.D

Graduate Faculty Representative, Sahjendra Singh, Ph.D

Cheryl L Bowles

Interim Graduate Dean, Cheryl Bowles, Ed.D

University of Nevada, Las Vegas

May, 1996

ABSTRACT

The objective of this research is to develop control of a multiple telerobot system based on Direct Numerical Processing Technology and propose a practically implementable collision avoidance algorithm for two gantry type robots sharing a common workspace.

The tele-operatic set-up consists of two Robotic Bridge Transporters which are X-Y-Z positioning overhead bridge type cranes. These cranes consisting of three sub-assemblies for the X, Y and Z motions respectively are actuated by brushless servo drive motors. The DC motors are controlled by Modicon FA3240 automation controllers from a supervisory control station equipped with computer graphics based human-machine interface. Teleoperation is achieved through a programmable logic controller which acts as a command arbiter and data interface between the automation controllers and the supervisory control station computer.

The collision avoidance algorithm proposes a collision free approach for a given path, by means of a minimum delay time technique for the two robots. Two examples, one each for a single segment and a two segment path for the two robots were tried out and found to work satisfactorily. The same can be extended to several segments as may be needed. The implementation methodology has been discussed.

Contents

ABSTRACT . . . iii
LIST OF FIGURES . . . vii
LIST OF TABLES . . . x
ACKNOWLEDGMENTS . . . xi

1 Introduction **1**
1.1 The experimental set-up . . . 2
1.2 Objective and Scope of Research . . . 3
1.3 Literature Survey . . . 4
1.4 Organization of Thesis . . . 6

2 System Configuration **7**
2.1 Robotic Bridge Transporters I and II . . . 7
  2.1.1 Bridge assembly . . . 8
  2.1.2 Trolley assembly . . . 10
  2.1.3 Telescoping tube assembly . . . 12
  2.1.4 Limit Switches . . . 13
  2.1.5 Cable management system . . . 13
2.2 Electrical Distribution System . . . 14
2.3 Control System . . . 16
  2.3.1 FA-3240 Automation Controller . . . 16
  2.3.2 Isolation transformers . . . 17
  2.3.3 Servo-drive Power Supplies . . . 18
  2.3.4 Servo-drive Amplifiers . . . 19
  2.3.5 Programmable Logic Controller (PLC) . . . 19
  2.3.6 Inductors . . . 19
  2.3.7 Motor Contactors . . . 20
  2.3.8 Clutch/Brake Controllers . . . 21
  2.3.9 Cooling Fans . . . 21
  2.3.10 TR-130 Control Station . . . 21
  2.3.11 Emergency Stop Contactor . . . 23
  2.3.12 Miscellaneous Power Supplies . . . 23
  2.3.13 DC Motor Starters . . . 24

2.3.14 Current Monitors/Sensors 25
2.4 Video Control Cabinet 25
2.5 RBT/MT Transfer Switch Panel 27
2.6 Supervisory Control Station 27
2.6.1 System Software 32
2.7 Cables 33

3 Servo System **35**
3.1 Operational principles of brushless servomotors 35
3.2 Control circuit of Brushless servo motors 37
3.2.1 Rotor position detector 38
3.2.2 Sine wave generation circuit 39
3.2.3 DC-SIN conversion circuit 40
3.2.4 Sine wave PWM circuit 42
3.2.5 Speed detector 44
3.2.6 Detection of Position signal 44
3.2.7 Detection of Speed signal 46
3.2.8 Speed control circuit 46
3.2.9 Direct Numerical Processing 48
3.2.10 Speed control loop 49
3.2.11 Position control loop 52
3.3 Automation Controller 59
3.3.1 Program module 61
3.3.2 Breakout Modules 61
3.3.3 Control Inputs and Outputs 62
3.4 3240 Firmware - C/ROS 62
3.4.1 C/ROS remote I/O processor mapping 63
3.5 Actual Setting of servo parameters 64

4 Programmable Logic Controller **65**
4.1 Introduction 65
4.2 Components of a Programmable Controller 66
4.3 Modicon 984 series control system 67
4.3.1 System Memory 68
4.3.2 Input/Output Systems 69
4.3.3 Communications 69
4.3.4 Theory Of Operation 70
4.3.5 984B PLC 72
4.3.6 PLC I/O racks 74
4.4 Configuring the PLC 78
4.4.1 I/O configuration and Traffic copping 79
4.4.2 Virtual Inputs and Outputs 81
4.5 Software 81
4.5.1 Networks 82

4.5.2 Segments 82
4.5.3 References 83
4.5.4 Relay contacts 84
4.5.5 Vertical and Horizontal Shorts 84
4.5.6 Coils 85
4.6 Translating logic elements in System memory database 85

5 Human Machine Interface 87
5.1 Introduction 87
5.2 Supervisory Control Station 88
5.3 Computer graphics based software interface 88
5.3.1 Intouch 89
5.3.2 Tagname listing in Intouch 89
5.3.3 Communication between Intouch and PLC 90
5.3.4 Using scripts in Intouch 91
5.4 Implementation of Intouch based Teleoperation 92
5.5 Joystick Control 94

6 Collision-Free Motion planning 98
6.1 Introduction 98
6.2 The Collision problem 99
6.3 Solution Methodology 103
6.3.1 Basis for solution 103
6.3.2 Assumptions 103
6.3.3 Defining Coordination space and Collision region 104
6.3.4 Collision criteria for two Gantry Robots 105
6.4 Defining collision region boundary 108
6.4.1 Constructing the Coordination curve 112
6.4.2 Collision mapping in t-s space 112
6.5 Collision avoidance 115
6.6 Summary of algorithm 117
6.7 Application example 118
6.8 Implementation methodology 129

7 Conclusion 130
7.1 Conclusions 130
7.2 Future research 133

Appendix A 135
A. 1 Setting of Resolution 135
A. 2 Axis Parameters 136

Appendix B 137
B. 1 Trapezoidal velocity Profile 137
BIBLIOGRAPHY 139

List of Figures

1.1 Control System - Hardware/Software Configuration 3

2.1 Robotic Bridge Transporter - I 8
2.2 Robotic Bridge Transporter - II 9
2.3 Block diagram of front elevation of Bridge assembly 10
2.4 Block diagram of front elevation of Trolley and tele-tube assembly 11
2.5 Block diagram of front elevation of limit switch assembly 13
2.6 Block diagram of Main electrical distribution system 15
2.7 Process Control Cabinet 17
2.8 Exploded view of FA-3240 Automation Controller 18
2.9 Programmable Logic Controller 20
2.10 TR-130 control Station 22
2.11 Supervisory Control Station 28

3.1 Principle of a rotating field 36
3.2 Principle of a rotating field 36
3.3 Principle of a rotating field 37
3.4 Block diagram of the system for brushless servomotors 38
3.5 Wire connection for ROM 40
3.6 Objective of DC-SIN conversion 41
3.7 D.C. to Sinusoidal conversion circuit 42
3.8 Principle of sine wave PWM 43
3.9 Connection diagram and relational expression for output of resolvers of single-phase output 45
3.10 Conversion circuit implementing an R/D convereter 45
3.11 PID amplifier 47
3.12 Block diagram of a DNP servo loop 48
3.13 Velocity profile for different ATC values 54
3.14 Block diagram of brushless servomotor control 55
3.15 Effect of changing RPG on velocity response 56
3.16 Effect of changing RIG on velocity response 57
3.17 Effect of changing ATC on velocity and position response 58
3.18 Block diagram of FA3240 system 60

4.1 A Typical Programmable Controller System 66
4.2 Block diagram of Modicon 984 Controller 67
4.3 Functional block diagram of Modicon 984 Control system 68
4.4 Block diagram of 984 controller system with drops configured 75
4.5 Database format for different nodes 85

5.1 Block diagram showing communication between Intouch and PLC 90
5.2 Graphic display of Front view of the system 92
5.3 Graphic display of Top view of the system 93
5.4 A typical joystick with axis direction indicated 95
5.5 Schematic of electrical connection of joystick 96

6.1 Case 1: Top view of collision between left end of RBT-II trolley frame and telescoping tube assembly of RBT-I 100
6.2 Case 2: Top view of collision between right end of RBT-II trolley frame and telescoping tube assembly of RBT-I 100
6.3 Case 3: Top view of collision between front end of RBT-II trolley frame and telescoping tube assembly of RBT-I 101
6.4 Case 4: Top view of collision between front end of RBT-II trolley motor and extended frame of RBT-I bridge assembly 102
6.5 Concept of Coordination Space and Coordination curve 104
6.6 Intersection between the line segment moving along a straight line path and the circle 109
6.7 Possible cases of intersection between circle and line segment 109
6.8 Flow chart to determine the collision boundary 110
6.9 Collision region and coordination curve in $s^1 - s^2$ space 114
6.10 Transformed collision region in $t^1 - s^2$ space and trajectory of robot 2 . . . 114
6.11 Possibilities of delaying or advancing robot 2 115
6.12 Two types of collision free coordination curve 116
6.13 Special cases of Collision region 117
6.14 Example:1 Paths of RBT-I and RBT-II 119
6.15 Example:1 Individual collision region in $s^1 - s^2$ space - case 1 119
6.16 Example:1 Individual collision region in $s^1 - s^2$ space - case 2 120
6.17 Example:1 Individual collision region in $s^1 - s^2$ space - case 3 120
6.18 Example:1 Individual collision region in $s^1 - s^2$ space - case 4 121
6.19 Example:1 Coordination curve and collective collision region in $s^1 - s^2$ space 122
6.20 Example:1 Collision region and path of RBT-I in $t^1 - s^2$ space 123
6.21 Example:1 Collision free path of RBT-I in $t^1 - s^2$ space 123
6.22 Example:2 Paths of RBT-I and RBT-II 124
6.23 Example:2 Individual collision region in $s^1 - s^2$ space - case 1 125
6.24 Example:2 Individual collision region in $s^1 - s^2$ space - case 2 125
6.25 Example:2 Individual collision region in $s^1 - s^2$ space - case 3 126
6.26 Example:2 Individual collision region in $s^1 - s^2$ space - case 4 126
6.27 Example:2 Coordination curve and collective collision region in $s^1 - s^2$ space 127

6.28 Example:2 Collision region and path of RBT-I in $t^1 - s^2$ space 128
6.29 Example:2 Collision free path of RBT-I in $t^1 - s^2$ space 128

B. 1 Trapezoidal velocity profile 137

List of Tables

3.1 Relation of each binary digit to the angle of rotation of the rotor 39
3.2 Contents of ROM 40

ACKNOWLEDGMENTS

This thesis represents the colloborative efforts of a number of people. Although it would not be possible to enumerate everyone who contributed, I certainly wish to acknowledge a few of those important individuals who played a major role in my completing this thesis. At the outset, I would like to thank my advisors Dr. Woosoon Yim and Dr. Mohamed Trabia for the help extended to me during the course of my graduate program. Their constant encouragement, guidance, and support, ensured that this thesis would be more meaningful to future researchers on this project. My thanks are also due to Dr. Brendan O'Toole and Dr. Sahjendra Singh for consenting to be on my thesis committee.

A very special thanks to Meng King at Modicon Technical Support group who went out of the way on several occasions to help me trouble shoot various problems encountered with their equipment during the course of this research work.

Last, but not the least, I wish to acknowledge the work of fellow students Mark Khiley, who built the skeleton setup and Murali Kathari, who greatly helped me in making the system operational.

I dedicate this humble effort to my mother whose motivation brought me this far and my entire family for their admirable support and backing.

Chapter 1

Introduction

The science and technology of robotics have developed to a great extent in recent years and many forms of robots have been built and put to use. The application of robotic systems in controlled surroundings is well known. However, it becomes difficult to predict the environments in some surroundings where the working area is large and the environment complex and calls for recognition of the environment and decision making. Human beings have the ability to adapt and react to changes in these environments which are still lacked by today's fully automated manipulators. It is in such cases where **Teleoperation** comes into action providing the opportunity to use both human supervision and robotics technology, especially in hazardous areas and for human decision oriented tasks.

Teleoperation which literally means "operating at a distance", can also be referred generically to direct and continuous control of a machine by a human through some form of barrier in a way that naturally extends the human capabilities of sensing and actuation beyond that barrier, as defined by Sheridan [1] and later by Jamshidi [2]. In remotely operated systems the distance is referred to as the barrier. Teleoperatic control could be established either by a human operated master arm which is geometrically similar to the remotely located slave arm often referred to as master-slave positioning device or through more advanced concepts like a keyboard of a micro-computer or a joystick. The endpoint positioning of the slave system is commanded by the operator from a supervisory control station.

1.1 The experimental set-up

The control system software hardware configuration of the set-up is as shown in figure 1.1. The tele-operatic setup consists of two Robotic Bridge Transporters which are X-Y-Z positioning overhead bridge type cranes. The end effector is a 14 axes human like slave arm capable of performing complex tasks. These cranes consisting of three sub-assemblies for the X, Y, and Z motions respectively are actuated by brushless servo drive motors. These DC motors are controlled by 4-axes servo controllers using Modicon's [3] Direct Numerical Processing (DNP) technology from a supervisory control station equipped with computer graphics based human-machine interface. A programmable logic controller acts as a command arbiter and data interface between the automation controllers and the supervisory control station computer. The supervisory control computer which is a pentium 90 MHZ PC, communicates with the PLC by means of a graphic programming language. It is also equipped with a PC based Man-Machine-Interface software which assists the human operator in teleoperation of the bridge transporters. The operator can also pass motion commands using a three axis joystick which provides proportional speed control of the transporters. The joystick serves as an implement for human intervention in the event of machine malfunctioning.

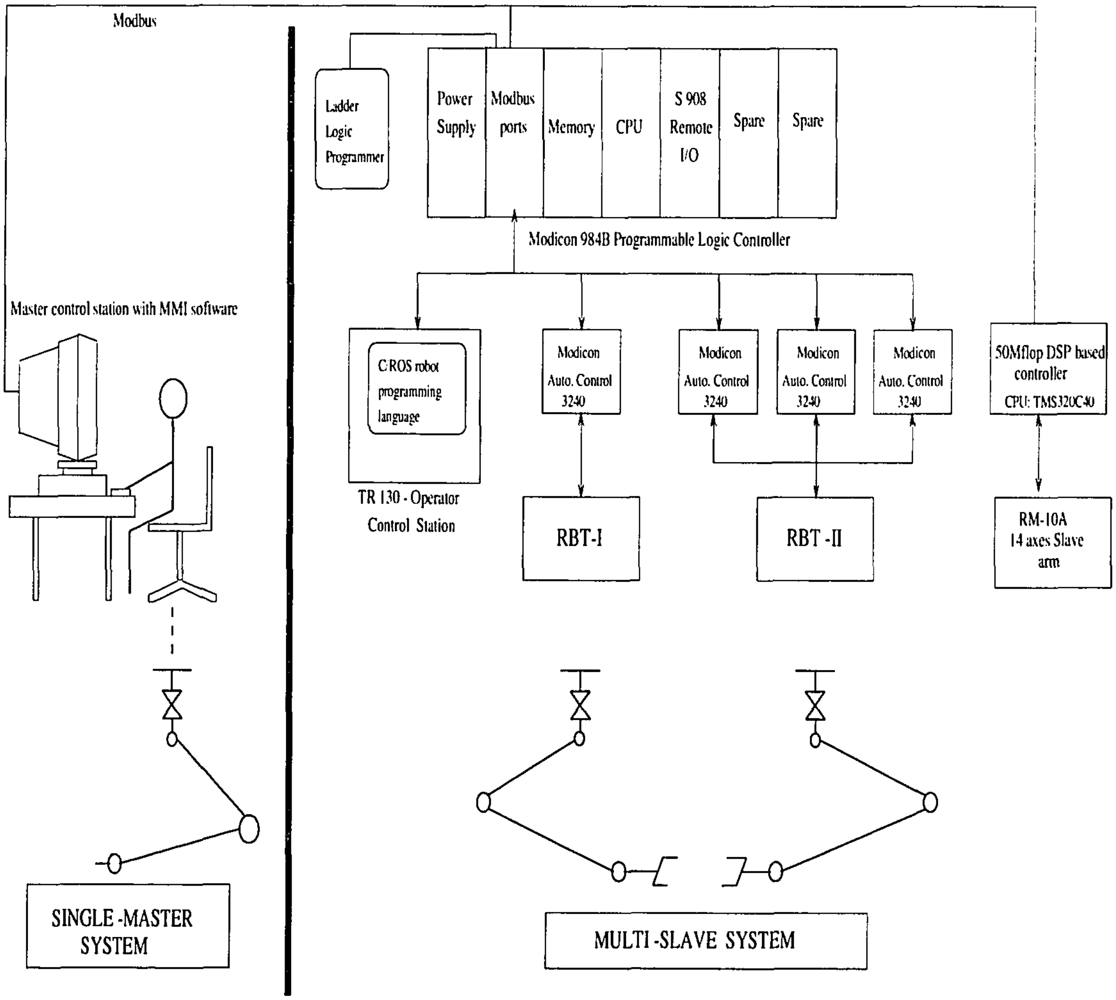


Figure 1.1: Control System - Hardware/Software Configuration

1.2 Objective and Scope of Research

The two fundamental tasks that an operator is confronted with, in teleoperated systems are those of moving the robots to a desired location and avoiding obstacles that can obstruct the motion while doing so.

The objective of this research is to develop collision-free teleoperatic control of the multiple robotic system consisting of two robotic bridge transporters for remote handling purposes. The scope of this study is divided into two parts. The first part of this experimental nature of research includes setting up of the whole system which involves a portion of mechanical installation, complete electrical installation and establishing supervisory control from a control station equipped with computer graphics based Man-Machine Interface software. The second portion of this research focusses on collision-free motion planning of the bridge transporters. A practically applicable collision-free path planning technique is being proposed and can be implemented in real time for the two gantry type robots sharing a common workspace.

1.3 Literature Survey

Telerobotics has recently become an active area of research primarily due to its flexibility in allowing human to be present as part of the control loop which is very essential in performing complex tasks in hazardous environments.

Teleoperated robotic systems are often very complex and difficult to operate especially when it comes to multiple robots sharing a common workspace and operating on a common task. This high level coordination problem according to Sheu [4], calls for highly precise control of the robots for position, velocity, acceleration, force, etc., The need for collision-free motion planning of the two robots thus becomes inevitable to accomplish complex coordinated tasks.

Collision-free motion planning for a single robot with stationary or moving obstacles has been handled in many works. However relatively lesser research has taken place for collision-free motion planning for multiple-robot systems. Freund and Hoyer [5] proposed a real-time collision avoidance planning method by introducing the concept of hierarchical coordinator while a similar problem was handled by J.Lee and Bien [6] by proposing an algorithm that minimizes the energy defining the degree of collision and the accomplishment of given tasks.

Chang et al. [7] proposed a collision free motion planning approach for two articulated robot arms by locally modifying the trajectory of the secondary robot by treating it as a constrained nonlinear minimization problem. B.H. Lee and C.S.G. Lee [8] in their approach for collision free motion planning for two robots, proposed an algorithm modifying the velocity profile of the secondary robot by time scheduling with the notions of a collision map.

The concept of "coordination space" commonly used when dealing with multiple robots, which is constructed with scalar variables defining the positions along prescribed paths was developed by Shin and Bien [9] and they also defined collision region in coordination space. Zeungnam Bien and J. Lee [10] proposed a method of generating time-optimal motions for two robot manipulators in a common workspace. The time-optimal trajectory of each robot is planned based on the algorithm of Shiller and Lu [11] under the constraints on actuator torques and velocities and the collision region defined in coordination space is transformed into time-versus-travelled length space with the preplaned velocity profile of one robot.

Subsequently, Jihong Lee et al. [12] proposed a practically collision free trajectory planning for two robot systems. This method decomposes the problem into two subproblems, path planning and velocity planning as discussed by K. Kant and S. W. Zucker [13]. This method is based on planning coordination decomposition technique in which planning finds trajectory of each robot independently according to their tasks and coordination finds velocity modification to avoid collision with each other. A delay time is introduced between path segments for collision avoidance and the delay time minimized. The method applies to any velocity profile in general.

This research extends the above technique of J. Lee [12] to two gantry type robots with suitable modifications. The main advantage of this technique lies in its simplicity and conservation of velocity of the robots as they move.

1.4 Organization of Thesis

This thesis is arranged into seven chapters - Introduction, System configuration, Servo system, Programmable Logic controller, Human Machine Interface, Collision avoidance algorithm and Conclusion. Chapter 1 has given a brief outline of the experimental setup and defined the objective and scope of the thesis. Chapter 2 describes the system elaborating on the equipment and major components used in the experimental set-up. In Chapter 3, the principle of operation of brushless servo motors, direct numerical processing control technique and 4 axis servo controller used in the system are discussed at length. Chapter 4 explains the Programmable logic controller, its theory of operation and communication aspects. Chapter 5 on Human machine interface explains how tele-operation is achieved from a supervisory control station and human presence in the control loop. In Chapter 6, a practically applicable collision avoidance algorithm is presented and the implementation methodology discussed. Chapter 7 concludes the study and also outlines the future research that is in progress. Appendix A lists the axis parameters used for the automation controller. Appendix B provides information on trapezoidal velocity profile and computation of distance in each velocity region.

Chapter 2

System Configuration

Chapter 2 describes the system configuration in detail. The system consists of two gantry type robots, hereinafter referred to as Robotic Bridge Transporters RBT-I and RBT-II, with rated capacities of 2 tons and 0.5 tons respectively. In this section, mechanical and electrical system configuration of RBT-I and II and the associated control cabinets are given in detail.

2.1 Robotic Bridge Transporters I and II

The RBT-I and RBT-II are X-Y-Z positioning devices as shown in figure 2.1 and 2.2 respectively, consisting of a bridge assembly (travelling jib crane assembly in case of RBT-II) for X-axis movement, a top running trolley assembly for Y-axis movement and a telescoping tube assembly for up-down (Z-axis) movement. [14]

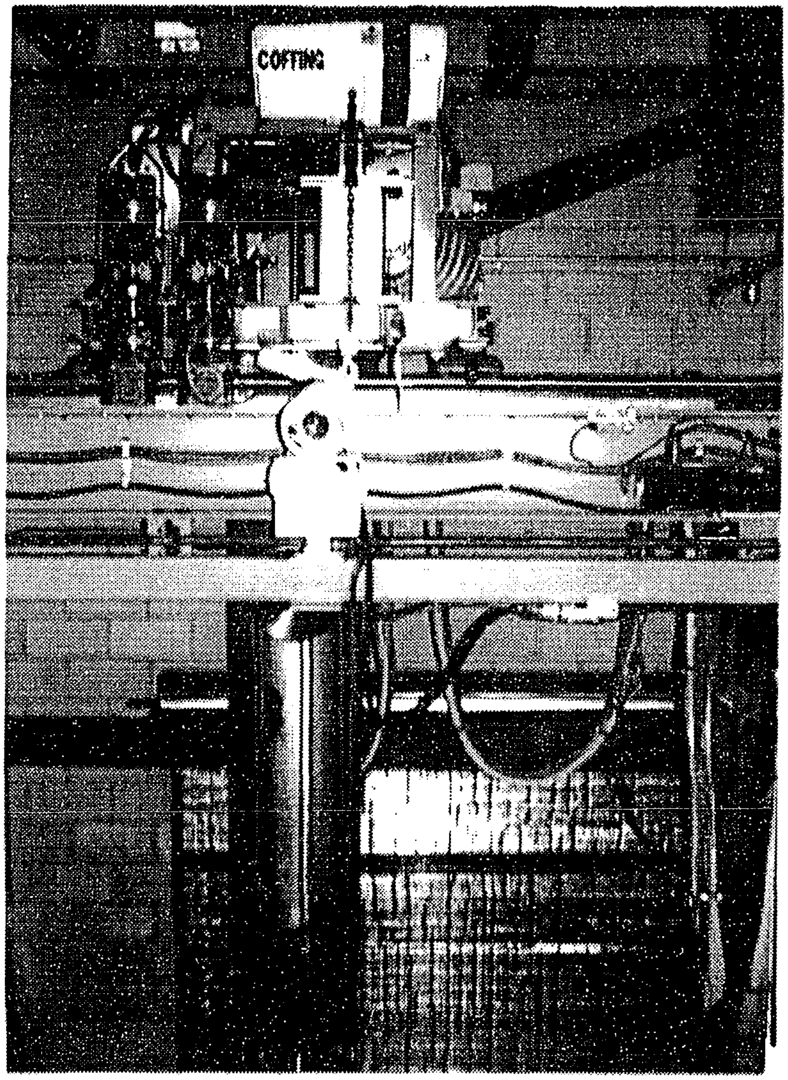


Figure 2.1: Robotic Bridge Transporter - I

2.1.1 Bridge assembly

The bridge rides on roundway rails mounted on steel I beams, with a top of rail elevation of approximately 15 feet. The bridge assembly consists of a double girder structure bolted to a pair of end trucks as shown in figure 2.3.

Each end has a drive motor assembly for redundancy. Each motor assembly is identical and consists of a brushless servo-drive motor with built-in brake, resolver, speed

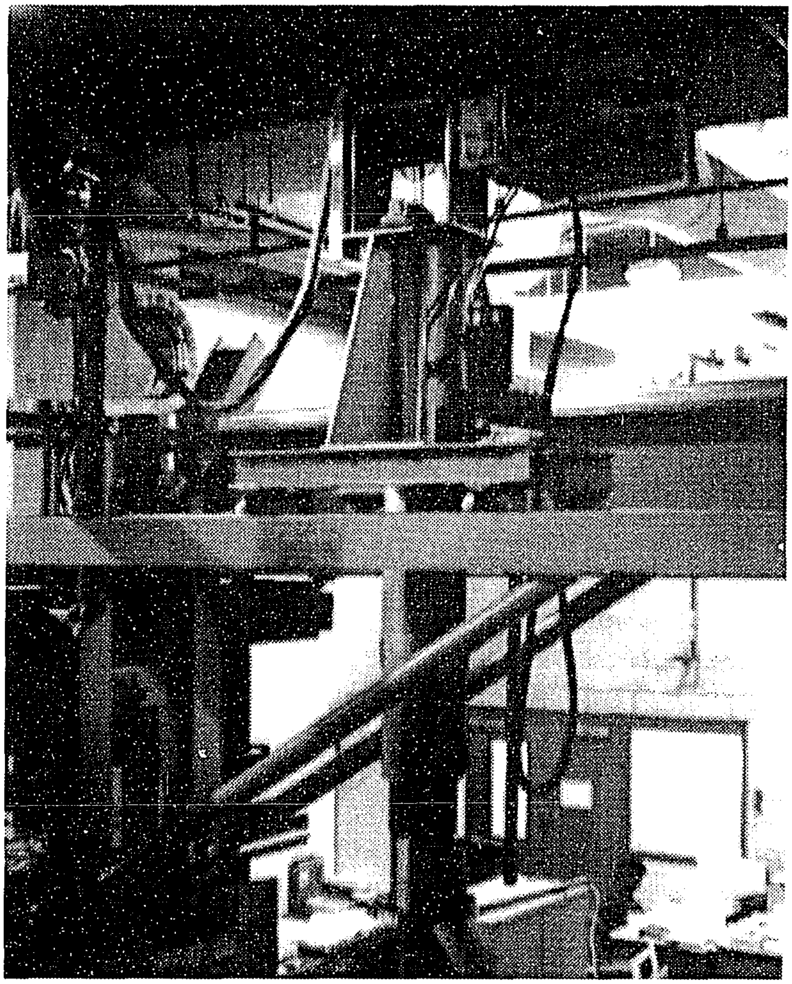


Figure 2.2: Robotic Bridge Transporter - II

reducer, electric clutch, and drive gear. Both drive gears are connected through a shaft so that each end truck is driven simultaneously along a track to prevent crabbing. During normal operation, the end trucks are driven by a single motor. The bridge can be driven by either one of the motors if the other fails. The clutch allows a failed motor to be disengaged and the other motor can position the RBT for repair or continue with process operations. Each servo-drive motor has a built-in resolver which provides position feedback signals to the servo controller in the control cabinet. The true position of the bridge is measured

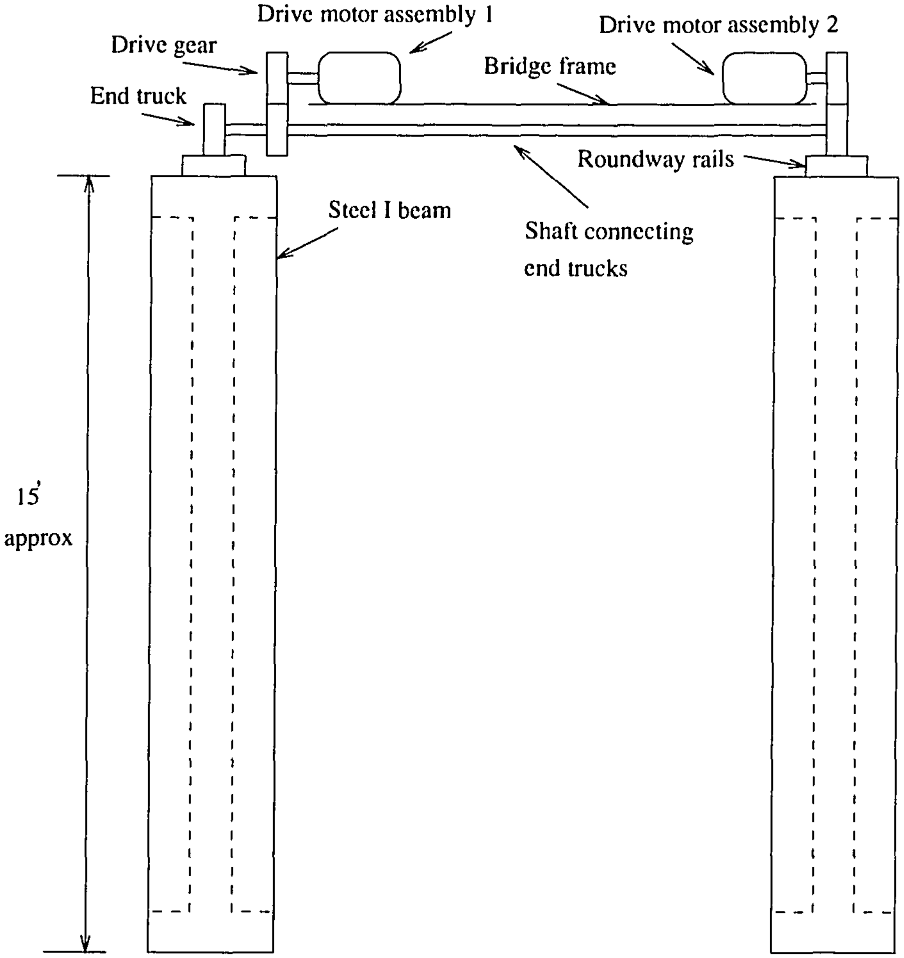


Figure 2.3: Block diagram of front elevation of Bridge assembly

directly by an external absolute resolver mounted on one of the trucks. This resolver is equipped with a pinion gear that mates with a rack attached to the rail.

The travelling jib crane for RBT-II is similar in operation to the bridge assembly mentioned above except that the drive system consists of a single motor only.

2.1.2 Trolley assembly

The trolley assembly of RBT-I consists of several components and assemblies. These components include the telescoping tube assembly, trolley frame, and two drive motor assemblies as shown in figure 2.4.

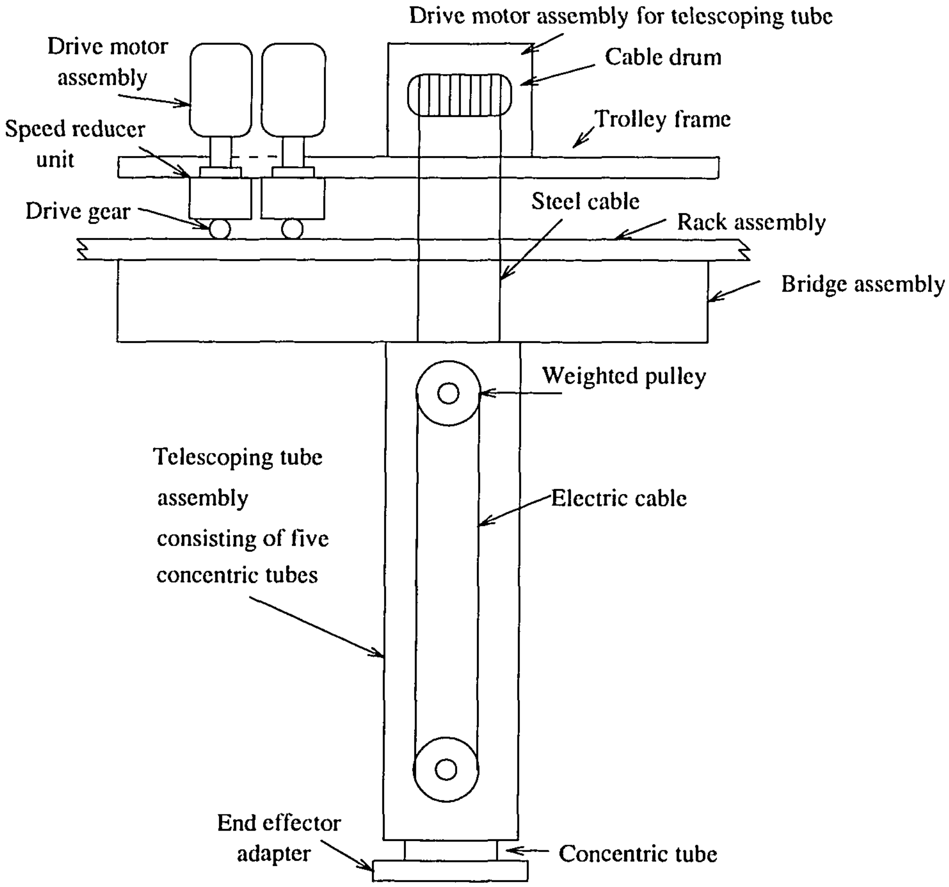


Figure 2.4: Block diagram of front elevation of Trolley and tele-tube assembly

The trolley assembly rides on roundway rails mounted to the bridge frame. The trolley assembly is driven along the rails by a drive gear, which is driven along a rack. The drive gear is powered by one of two servo-drive motor assemblies which are essentially identical to the bridge's servo-drive motors, with the single exception that a different speed reducer is used on the trolley. All other components of the drive motor assembly are identical. The trolley can be driven by either one of the motors in the event that one fails. The true position of the trolley can be measured directly by an external absolute resolver attached to one end of the trolley.

The trolley assembly of RBT-II is essentially identical to that of RBT-I except that it has a shorter travel length, different speed reducer and is driven by a single drive motor assembly.

2.1.3 Telescoping tube assembly

The telescoping tube assembly consists of five concentric tubes. The outer tube is bolted to the trolley frame as shown in figure 2.4 and the four inner tubes are free to move vertically along a cam follower and track assembly. The tubes are raised and lowered by a redundant cabling system which is connected from a cable drum on the trolley assembly to the lower innermost tube. The cable drum is powered through a right angle gearbox by two redundant servo-drive motor assemblies. Each motor assembly is identical and consists of a servo-drive motor with built-in brake, resolver, electric clutch, and a modified coupling. This modified coupling allows the motor assembly to be lifted off the right angle gearbox in a simple vertical motion for replacement operations. The telescoping tube can be driven by either one of the motors in the event that one fails. The true positioning of the tube is measured directly by an external absolute resolver. The external resolver is chain driven by the cable drum drive system which is mounted to the trolley frame. On the lower end of the innermost telescoping tube is the RBT-I end effector adapter. The end effector adapter provides both the mechanical and electrical connections to the end effector. The end effector is mounted with an identical flange that has four guide pins which fit into alignment holes on the RBT-I adapter. When the adapter and flange are fully engaged, two DC motors drive a locking collar to latch the end effector to the interface adapter.

The telescoping tube assembly of RBT-II is similar to that of RBT-I, except that it has four concentric tubes as against five of the latter.

2.1.4 Limit Switches

The bridge, trolley, and telescoping tube are provided with limit switches to prevent the assemblies from being overdriven into the hardstops. A simple block diagram shown in figure 2.5 explains the arrangement better.

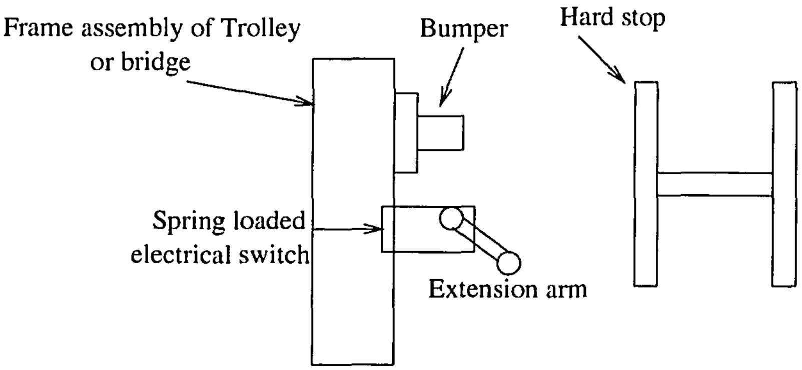


Figure 2.5: Block diagram of front elevation of limit switch assembly

The bridge and trolley limit switches consist of a spring loaded electrical switch with an extension arm which contacts the hardstop prior to the truck assembly. The switch is set to disengage the drive prior to the drive assembly making contact. In the event of a limit switch failure, bumpers are provided to cushion the impact for the bridge and trolley. The telescoping tube employs cam-type limit switches which are driven off the cable drum drive. Two limit switches at each end of tube travel are used. The first limit switch contacted at either end of travel, is the overtravel switch. If either of these switches fails, a final overtravel switch is contacted to deactivate the drive motor.

2.1.5 Cable management system

Power and signal transmission for the bridge, trolley, and telescoping tube assembly occur through a series of cables. To avoid entanglement or damage to the cables, a cable management system is utilized with each of the RBT subassemblies. The bridge and trolley have similar cable management systems which consist of cable trolleys that ride along a track. The cable bundle is attached to the cable trolleys which are pulled or pushed along the bridge and trolley. The telescoping tube cable management system consists of a series of weighted pulleys which the cable bundle rides in. The pulleys move away from each other to take up the slack when the telescoping tube is retracted. When the telescoping tube is extended, the pulleys move toward each other to allow the cable to be payed out.

The RBT-I is rated as a 2-ton capacity bridge crane per ANSI/ASME NOG-1 requirements.

2.2 Electrical Distribution System

The Electrical Distribution System distributes, transforms, converts, or regulates the electrical power supplied by others for use in the operation, monitoring, and control of the equipment. The process control equipment interfaces with the electrical distribution system through vital and non-vital distribution systems. The non-vital distribution system provides operating power to motors and motor controls. The vital distribution system provides power to the supervisory control station, all FA-3240 Automation Controllers, the PLC, and all power supplies necessary to support these items. The electrical distribution system receives 3-Phase, 480 volt, 60 hertz power from an external source. The external power is fed to the main distribution Panel DP-I which is equipped with a three pole, 400 amp main circuit breaker. Circuit breakers in DP-I then distribute power to the remainder of the electrical distribution system. Figure 2.6 shows a schematic of the electrical distribution system.

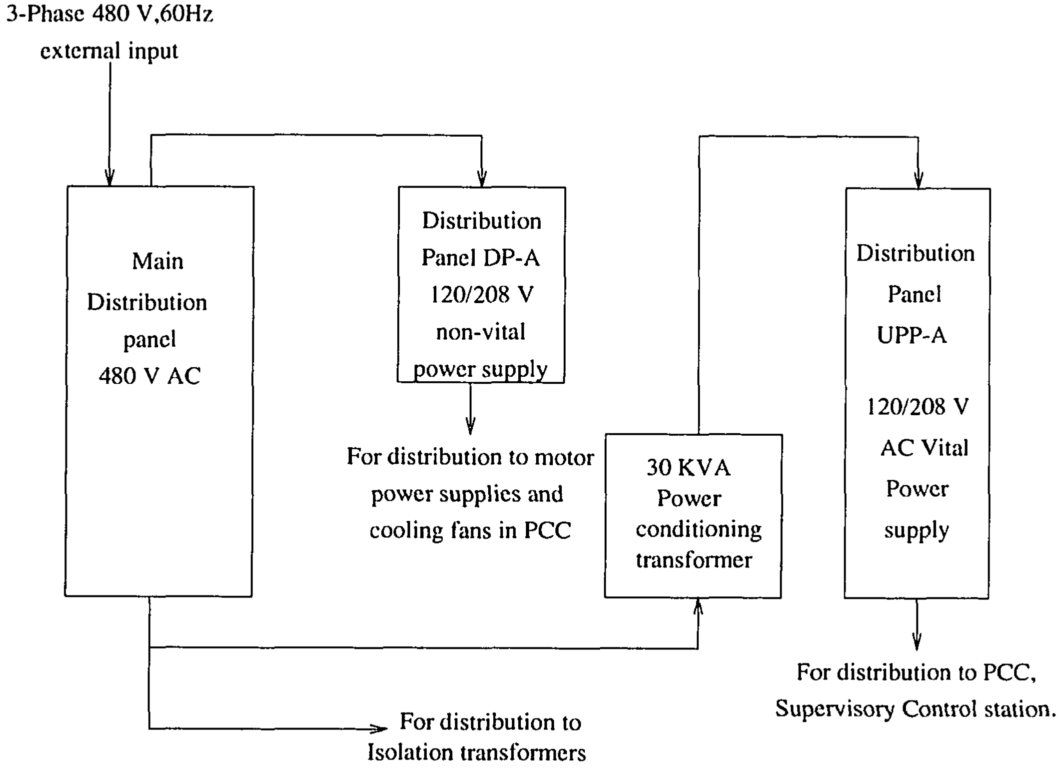


Figure 2.6: Block diagram of Main electrical distribution system

Distribution Panel DP-I supplies power to the distribution panels DP-A and UPP-A, 30KVA power conditioning isolation transformer, isolation transformers 1, 2, 3 and 4 located in PCC-06, and RBT auxiliary hoist motor starters. Distribution Panel DP-A which contains a 15KVA transformer supplies 120/208 volt non-vital power to 115 VAC motor power supply, 50VDC motor power supply and 12VDC motor power supply located in PCC-05 and the cooling fans located in all the cabinets. Distribution panel UPP-A supplies 120/208 volt vital power to process control cabinets 01, 03, 12, 14, video control cabinet, supervisory control station, RM-10A Master control station, video control receivers and overhead CCTV monitors.

2.3 Control System

Control System consists of all the electrical instrumentation cabinets, control equipment, process control stations, and software necessary to operate and maintain the equipment.

There are 5 process control cabinets (PCC) located in the control room. This room also contains one video control cabinet and other wall mounted controllers and panels.

The primary function of the control cabinets is to house all the various power, control, instrumentation, and monitoring equipment required to support the funcion of the equipment. The cabinets are also to provide a safe, clean, ventilated environment for the equipment while keeping it organised in groups according to its use.

Each PCC is designed as a NEMA-1 enclosure and is ventilated with filtered cooling fans. A typical Process control cabinet with major components assembled is as shown in figure 2.7

The size of the fans are based on the amount of heat dissipated within an individual cabinet and the amount of room temperature air required to keep the components below their maximum operating temperatures. Each of the Process control cabinets have two doors, one front, and one rear. With the exception of PCC-1, both the doors provide access to control components. The following are some of the major components that can be found in the PCCs.

2.3.1 FA-3240 Automation Controller

The FA-3240 Automation Controllers are responsible for operation of servo-drives in all modes. Each 3240 controller is autonomous and can operate independently of other machines. An exploded view of a Modicon 3240 automation controller is shown in figure 2.8.

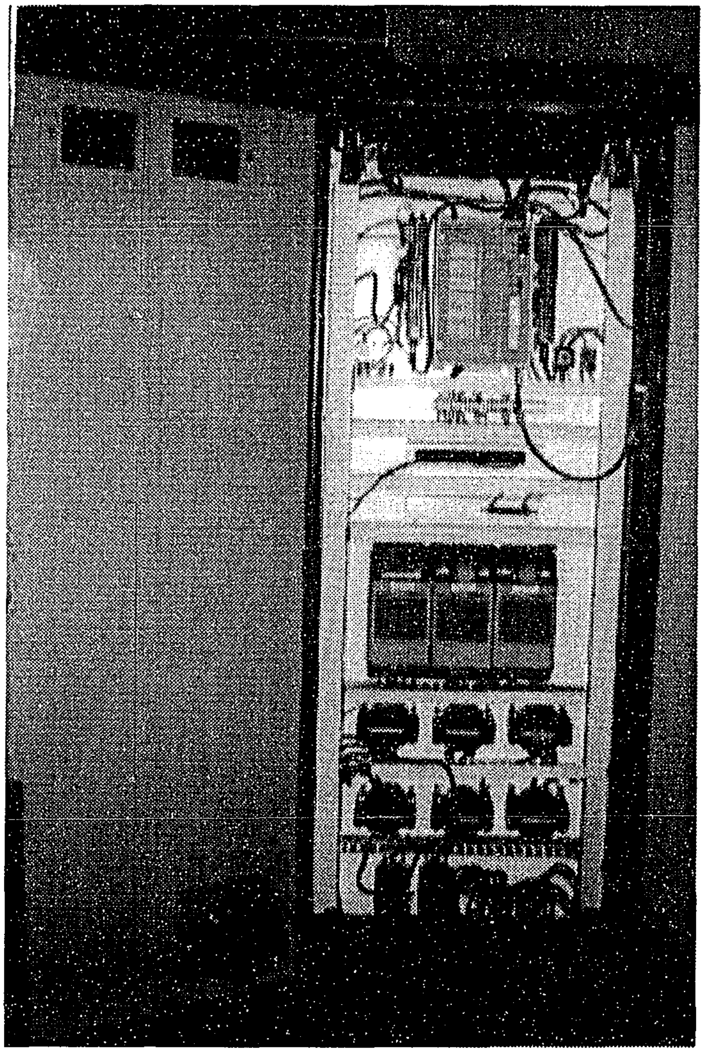


Figure 2.7: Process Control Cabinet

The automation controller is dealt with in more detail in chapter 3.

2.3.2 Isolation transformers

The isolation transformers provide both the isolation and the voltages needed by the servo-drive power supplies and amplifiers. Each transformer has two 3-phase outputs, one at 199 VAC and the other at 19VAC. The transformers used are rated 13 KVA and are capable of supplying power for more than one motor depending on the motor size. Each transformer is

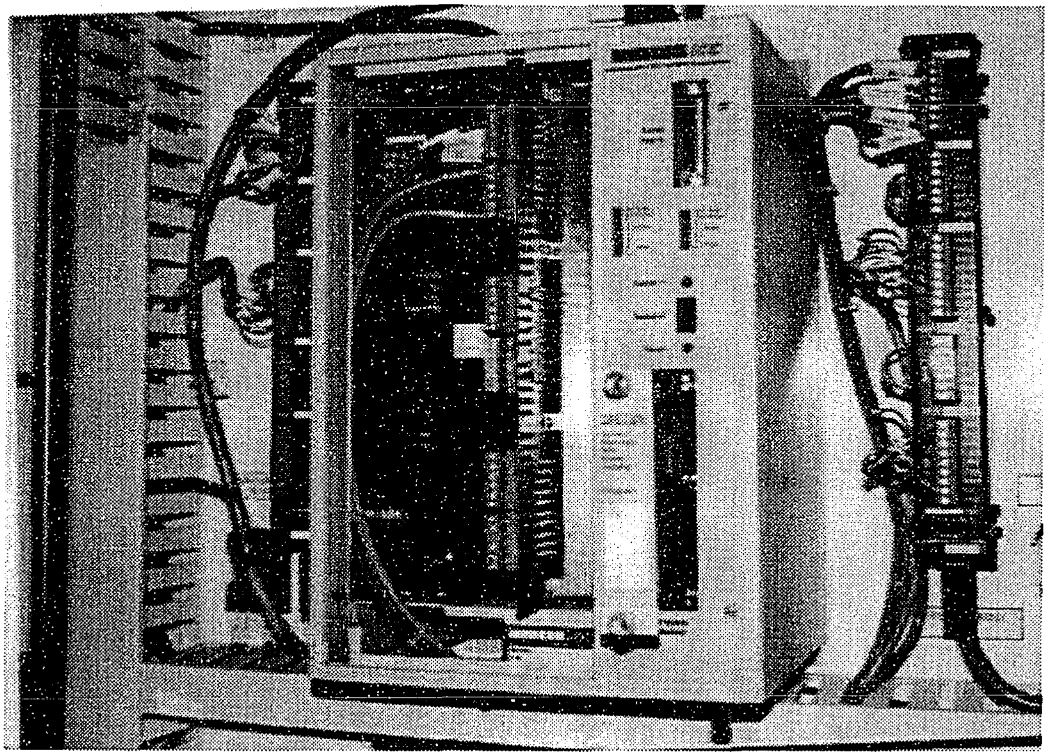


Figure 2.8: Exploded view of FA-3240 Automation Controller

associated with a group of fuses or circuit breakers for protection and distribution of power to the various components they feed. The isolation transformers receive 3-phase, 480 volt, 60 hz power from main non-vital distribution panel DP-1.

2.3.3 Servo-drive Power Supplies

These power supplies are provided in two different size configurations. Each provides nominal DC bus voltage of 280 VDC and maximum bus voltage of 325 VDC. These power supplies also supply low-level DC voltages (+12, -12 and +20 VDC) to the drive amplifiers. The power supplies receive 3-phase AC input power from the isolation transformers. Each power supply can power one or more drive amplifiers depending on the size of the motors involved.

2.3.4 Servo-drive Amplifiers

The drive amplifiers are 3-phase, pulse width modulated (PWM) current amplifier/motor drives which receive a maximum bus voltage of 325 VDC from the drive power supplies. The motor's resolver provides commutation, speed, and position feedback to the FA-3240 automation controller, which generates 3-phase, low level torque command signals. The drives then amplify the 3-phase torque command signals from the FA-3240 automation controller and send revised 3-phase power to the motors. Each drive amplifier can drive one motor only.

2.3.5 Programmable Logic Controller (PLC)

The Programmable logic controller is an essential part of the control system. It serves as the interface between the supervisory control station and the automation controllers. The setup uses Modicon 984B PLC which is shown in figure 2.9. The PLC has been dealt with in detail in Chapter 4.

2.3.6 Inductors

Inductors are provided for low inductance motors to prevent a mismatch with the drive amplifiers. One inductor is required for each line of the 3-phase motor.

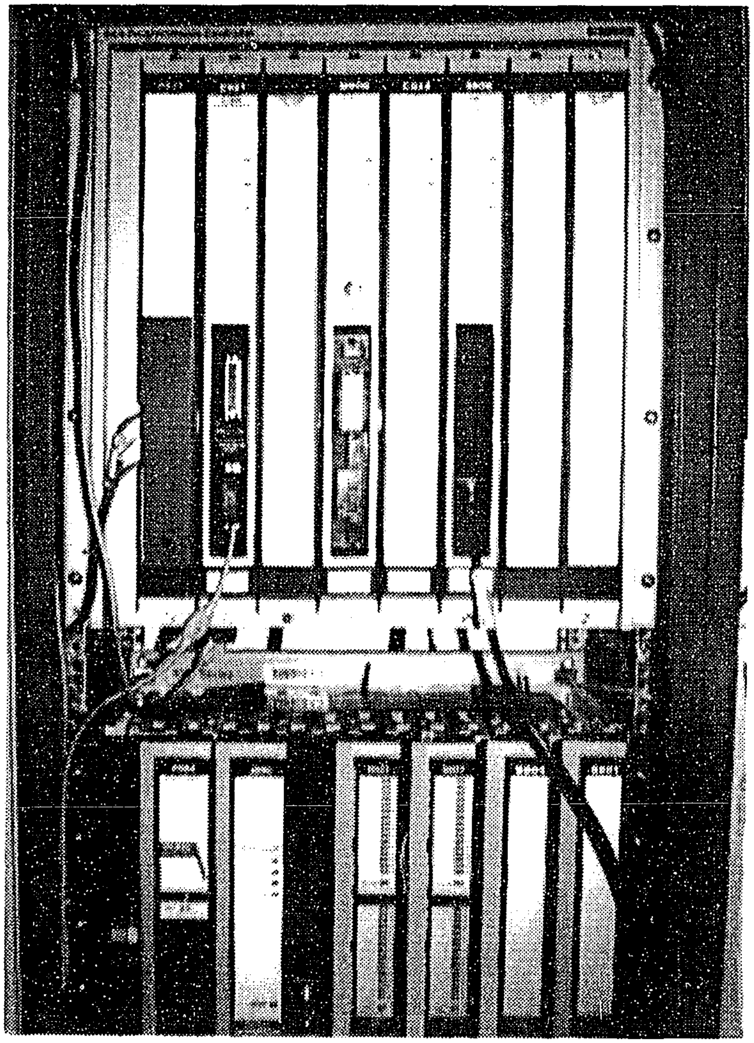


Figure 2.9: Programmable Logic Controller

2.3.7 Motor Contactors

A motor contactor is provided for each servo-drive and is wired between the amplifier and the motor. The contactors are provided for emergency conditions to immediately remove power from the motors. The contactors operating coils are wired into the 120VAC emergency stop control circuit. When an emergency stop device is activated, these contactors will all deenergise and their contacts open and remove power from the motors. Dynamic braking resistors are wired into contactors whose motors may need them. These resistors get switched across the motor leads when the contactors deenergise.

2.3.8 Clutch/Brake Controllers

A clutch/brake controller is provided for each clutch and or brake associated with a servo-drive. These controllers are supplied 120VAC and provide 90VDC for clutch/brake operation. The 90 VDC supply is switched on and off as required by the control system through output modules in the various 3240 Automation Controllers.

2.3.9 Cooling Fans

All control cabinets are provided with one or more cooling fans depending on the amount of air required to keep the internal components below their maximum operating temperatures. The fans are mounted in the lower part of the front and /or rear door, and are provided with filters to keep dust and dirt out of the cabinets. Ventilation openings are provided at the top of the doors. The air flow pattern is designed for cooler air to be brought in to the cabinet at the bottom and pressurize the cabinet. The warmer rising air is then forced out through the ventilation openings at the top of the cabinets. The size and quantity of fans is based on the amount of ambient room temperature air (68-72 deg F) required to keep the internal components below their maximum operating temperature of 95 deg F.

2.3.10 TR-130 Control Station

The TR-130 Control Station is a combination video screen, keyboard, and keypad as shown in figure 2.10. The video screen displays messages from an FA-3240 automation controller concerning equipment status or default conditions. The screen can also be used to display menus, diagnostic data, and prompts from the controller. The keyboard and keypad are a sealed membrane type to keep out dust and liquids. The keyboard and keypad are used to enter alphanumeric data, select functions, choose menu options, respond to prompts, and in general communicate with the automation controller.

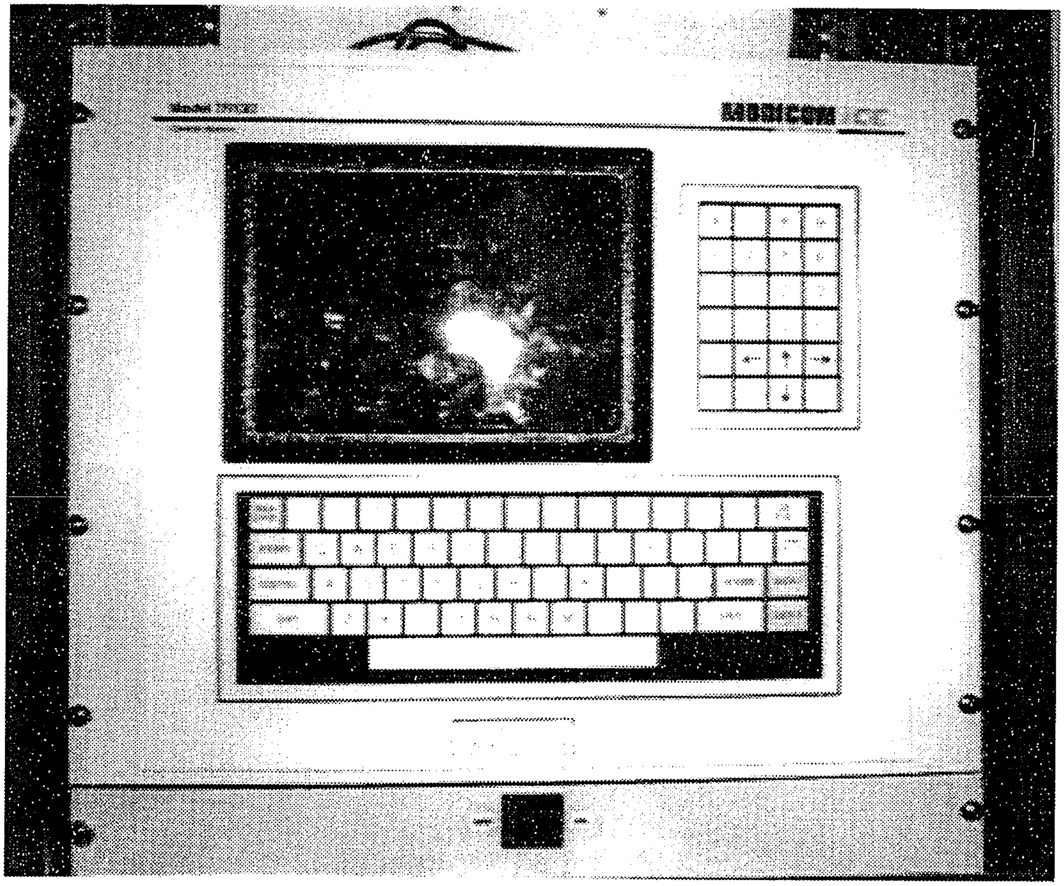


Figure 2.10: TR-130 control Station

The TR-130 has been provided for use as a start-up and maintenance tool. The TR-130 can only communicate with one of the automation controllers at a given time. The controller it communicates with, is selected by the operator at the patch panel located below it. The communications is through a RS-232 serial port on the controller. The TR-130 is powered from distribution panel UPP-A.

2.3.11 Emergency Stop Contactor

The emergency stop contactor when de-energised will remove power from all motors and cause all brakes to be set. This contactor removes power from all the motors by removing power from the operating coils of all motor starters and all servo motor contactors. When the emergency stop is activated a signal will be given to each 3240 and the PLC so that the controlling devices will know why motion has stopped and will prevent the sending of numerous error messages. To reset from an emergency stop, the operator must physically reset the stopping device (button).

2.3.12 Miscellaneous Power Supplies

Miscellaneous power supplies are provided with the system for operating the DC motors and providing power for various instrumentation and control components. Each power supply is associated with a group of fuses for protection and distribution of power to the various components they feed.

Power supplies for DC motors receive non-vital power from distribution panel DP-A. These power supplies are found in PCC-05 and are described below:

- 50VDC Power Supply

  The power supply for the 50 VDC motors is a 50 amp adjustable 0-60 VDC supply. The supply has a digital readout of selected voltage. The power supply also has an adjustable current limiting feature.

- 12 VDC Power Supply

The power supply for the 12 VDC motors which are used for end-effector latching and unlatching is a 7-8 amp, 12 VDC supply.

Power supplies for instrumentation and control purposes receive vital power from distribution panel UPP-A. These supplies are found in PCC-03 and are described below:

- 24 VDC Power Supply

  This power supply is rated at 6.5 amperes and is used to provide power to operate discrete inputs to the system processors.

- +6 to -6 VDC Power Supply

  This power supply is made of two, 3.4 amp, adjustable 0-7 VDC supplies which have been adjusted to 6 volts and have been wired to provide +6 to -6 volts. This power supply is used to provide power to the console joy sticks.

- 28 VDC Power Supply

  This power supply is rated at 3.5 amperes and is used to provide power to operate the current monitors.

- 12 VDC Power Supply

  This power supply is rated at 7.8 amperes.

2.3.13 DC Motor Starters

These motor starters are full voltage reversing type with 115 VAC operating coils, NEMA size 00, three pole, for polyphase motors. Each starter is provided with the required overload relays and heater elements. These starters are used to operate the low current 50 and 25 VDC motors. Each starter has been wired to place three contacts in series with the motor to decrease the arcing effect when the motor is de-energised. Each starter is associated with two fuses, one in the positive leg and one in the negative leg.

2.3.14 Current Monitors/Sensors

A DC current sensor is provided for each DC motor starter. Each sensor has a range of 0-10 amperes. Smaller motors with low currents have their motor lead passed through the sensor several times in order to proportionately multiply the amount of current seen, thereby effectively dividing the range accordingly. Therefore, if the wire passed through the sensor two times, the effective range would be 0-5 amperes and if it passed through five times, the range would be 0-2 amperes and so on. The sensor operates on 28 VDC from a regulated power supply. The sensor output signal is a 0-5 VDC analog signal which is proportional to the current being monitored. The sensor is wired to the PLC. The PLC converts this signal to a 0-4,096 count scale and monitors a calibrated setpoint for determining proper clamp force and/or fault current. The supervisory control station reads the PLC's 0-4,096 count for each current monitor and rescales it to current, and then multiplies it times the motor manufacturer's "K" factor for that motor to calculate torque.

2.4 Video Control Cabinet

Video control cabinet VCC-02 contains most of the video processing equipment. A description of the video processing equipment in this cabinet is as follows:

- Camera Control Units

  Two camera control units (CCUs), one for each camera, are provided. The CCUs contain all the remote electronics, controls, and indicators necessary for the proper operation, adjustment and alignment of the system cameras.

- Video Matrix Switcher Assemblies

  Two video matrix switcher assemblies are provided as the heart of the system. These switchers control the routing of all video input and output signals in the system as well as all pan, tilt, zoom and auxiliary control inputs and outputs. The switcher allows monitoring of any camera at any time on any monitor. Video input channels are provided with identification numbers that are superimposed on the picture.

- Camera Control Receivers

  Two camera control receivers, one for each camera, are provided. The receivers, which are mounted on the equipment room wall, are associated with the VCC. These receivers along with the control console video transmitters are an interface between the operator and camera and its pan/tilt unit.

- Video Multiplexer

  One video multiplexer is provided to interface the two video control transmitters located on the operator control station, with the video switchers and the individual video control receivers.

- Microphone Preamplifier

  Three microphone preamplifiers with adjustable gains, each capable of handling two microphone inputs, are provided for preamplification of the systems microphones. These preamplifiers allow for signal matching and an increase in gain to permit signal splitting of each output. which goes to supervisory control station.

2.5 RBT/MT Transfer Switch Panel

The purpose of this panel is to provide the necessary terminals and relays to allow the RBTs to share certain wires within their cables. This is required to reduce the number of wires that travel with the RBT's telescopic tube. The relays consist of three multi-pole control relays and two multi-pole coax relays for switching of the RM-10A's camera video signal.

2.6 Supervisory Control Station

The primary function of the supervisory control station is to provide an operator interface to allow for the control and monitoring of the setup in both automatic and manual modes of operation. It also controls the start-up and shut-down of the equipment, processes all alarm and hold point functions, and produces reports on general system data as and when required.

This control station is networked to the PLC and to all of the 3240 automation controllers which control the two RBTs. This communication network provides the supervisory control station with the capabilities to monitor all system parameters, download programs, receive messages, and respond to messages.

The supervisory control station which is shown in figure 2.11 is a wing shaped two bay console and is designed as a NEMA-1 enclosure and is ventilated with filtered cooling fans. It has two doors in the rear, one in the lower left front, and one drawer in the lower right. Each door provides access to control components and the drawer is for general storage. The major components associated with the supervisory control station are as follows:

- Supervisory Control Station Computer

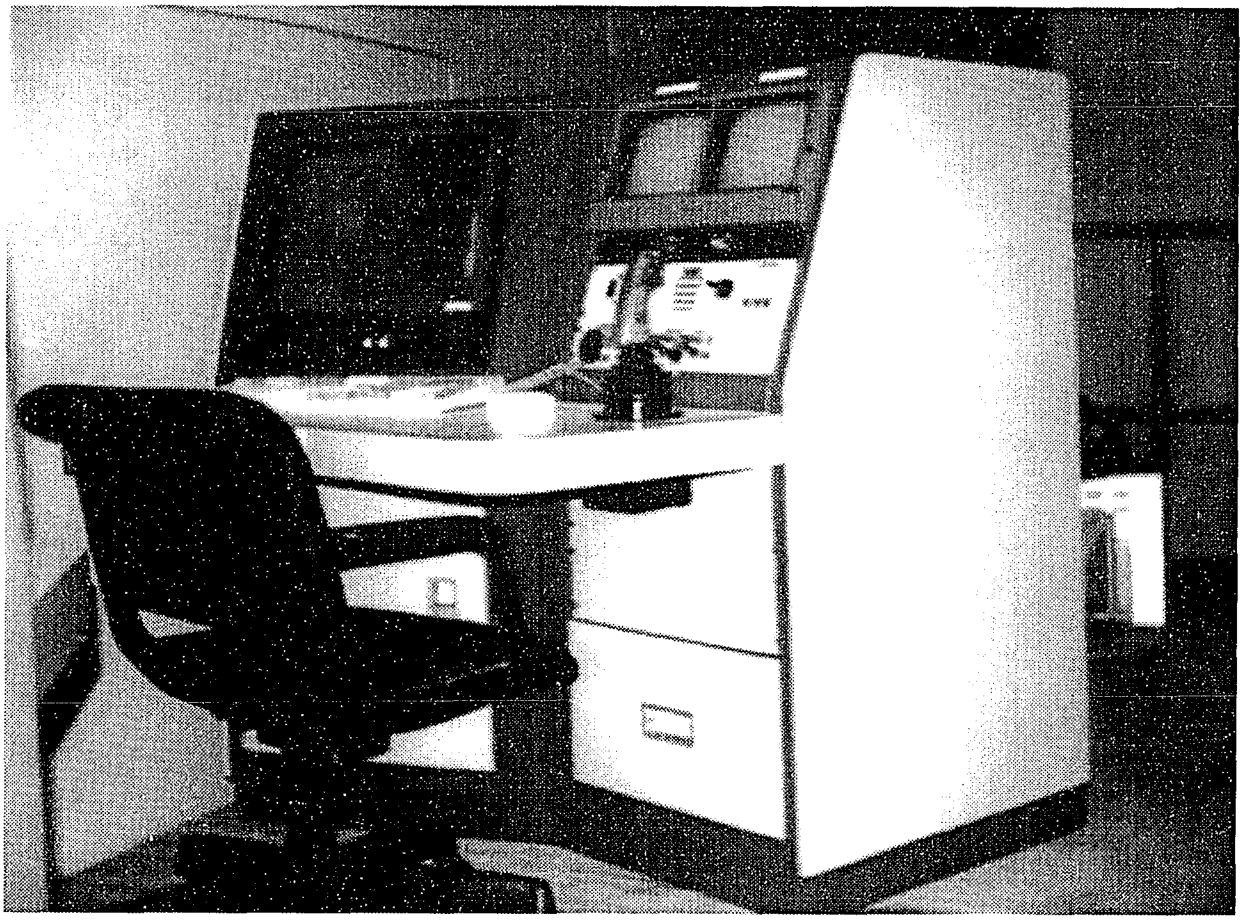


Figure 2.11: Supervisory Control Station

The supervisory control station computer consists of a Pentium 90 MHZ PC with a 731 MB hard drive and a 1.44 MB 3.5″ floppy drive; a 19-inch rack mounted color monitor with a touch screen; all necessary software and communications interface boards; and a keyboard and mouse for operator input.

- Audio Amplifier/Mixer

The amplifier/mixer provides the necessary outputs and preamp input modules to interface with speakers, video tape recorders, and microphones. The amplifier/mixer also provides a master volume and tone control for the station as well as individual preamp gain controls for the six inputs. These gain controls allow the operator to select and/or mix the inputs to allow audio monitoring as needed.

- Video Monitor Panel

  This panel contains two 9-inch black and white CCTV monitors provided for operational and general viewing, if the operator selects the cameras he wishes to view. Each monitor allows for video in and a video out signal. Each monitor also provides all necessary video adjustment controls as well as a speaker and volume control.

- Audio Panel

  The audio panel contains a remote volume control and a headset jack. The volume control allows the operator to adjust the volume of the overhead speaker associated with that station without going to the amplifier. A pair of headphones are provided so that the operator may listen to the sounds without interferences from ambient noise in the control room. When the headphones are used, the speaker is cut off and the volume control then adjusts their volume.

- Video Control Panel

  The Video control panel allows the operator to interface with the CCTV system by transmitting all the operators commands and selections to a video receiver, a video switcher, and a video alarm processor. The video control panel provides the operator with the following major capabilities:

  - Select a camera and a monitor to view it on.

  - Select a camera to control.
  - Control a camera pan, tilt, and zoom.
  - Turn camera lights off and on and switch them between bright and dim settings.
  - Display the camera and monitor most recently selected.

- Manual Control Devices

  The supervisory control station has several manual control devices available for use during manual procedures or emergency conditions. These manual control devices consist of the following:

  - Two three-axis joysticks which provide independent jog mode motion control for RBT-I and RBT-II. A trigger on the joystick acts as a deadman switch to prevent accidental actuation. A neutral switch in the joystick unit prevents vertical motion while performing transverse motions and vice versa when the joysticks are used with the RBTs. The joysticks provide proportional speed control for servo motors. Control of RBTs using joysticks is dealt with in detail in Chapter 5 on Human-machine Interface.
  - A JOG pushbutton on the control station which provides an alternate method to the joystick for controlling equipment motion when in the JOG mode. The use of the JOG pushbutton operates an individual motor in the direction selected and at the maximum speed selected.
  - An "RBT ON-OFF " selector keyswitch is provided as an interlock for switching onto manual operation using joysticks. The supervisor must insert his key and turn the key switch in the "ON" position. This allows the operator to JOG the

RBTs.

- An "RBT-I/RBT-II HOIST FAULT - NORMAL/BYPASS" keyswitch is provided for operation of the primary hoist on the two RBTs. If the hoist faults with the slack cable or drum level wind problem, the contactors for the hoist motors are opened, removing motor power. In order for the hoist motors to be reactivated, the supervisor must insert his key and turn and hold the keyswitch in the "BYPASS" position. This reapplies power to the hoist motor and allows the operator to JOG the hoist away from the fault condition.
- An "EMERGENCY STOP" pushbutton is provided to stop all related motion. The button is maintained when pushed and drops out the emergency stop contactor which causes power to be removed from all motors. To restart the system following an emergency stop, the operator must pull out the "EMERGENCY STOP" button.

The communications interface boards consist of several cards dedicated to system communications and are described below:

- An asynchronous real-time interface co-processor (ARTIC) card is used to provide an intelligent communications interface to the 984B PLC via the PLC's MODBUS ports.
- A microcomm 422 card is used to provide an interface to each of the 3240 automation controllers controlling the equipment. This communication path allows the supervisory control station computer to download part-programs to each of the 3240 automation controllers.
- A digital Ethernet Plus/Microchannel card is used to provide a local area network

(LAN) to connect the supervisory control station computer to the network.

2.6.1 System Software

The whole setup is comprised of the following separate intelligent devices which require software:

- A Supervisory control station computer
- A 984B PLC
- Four 3240 Automation controllers

From an external standpoint, the control setup acts as a single entity, providing total control. This is accomplished through a software architecture design which distributes the motion processing functions to the automation controllers while retaining the central control within the 984B PLC, in turn controlled by the supervisory control computer.

The function of the software at the Supervisory Control station consists of the following:

- Storage and development of 3240 automation controller programs.
- Communications with 3240 automation controllers over the 422 network for up-loading and down-loading the programs.
- Start and shut-down equipment operations.
- Communicate with the PLC through the MODBUS port for up-loading and down-loading programs.
- Off-line development and storage of the 984 PLC ladder logic program.

- Establish teleoperation using Intouch, a PC based Man-Machine Interface software.

The function of the software at the PLC is to control the overall equipment and be the central interfacing point for all other computers and processors as well as support all of the systems I/O points.

The function of the software at the 3240 automation controllers is to provide specific motion control for individual servo motors as well as non-motion functions for both individual motors as well as for the overall equipment it controls.

2.7 Cables

The primary function of the cables is to serve as the electrical interface between the control cabinets/boxes and the equipment for all power instrumentation and control.

The basic type of cable designs utilized for the control system are power, instrument, control, and CCTV cables. The power, instrument, and control cables have a crosslinked -polyethylene insulation and jacket and utilize multi-stranded copper conductor wire for flexibility. The cables are flame retardant and have been designed for a radiation environment of 2×10^8 rads, total integrated dose. All cable/conductor sizes and requirements have been selected to perform their intended functions within the design limits. Cables are qualified to applicable standards including IEEE-323 and IEEE-383. The cables that connect to the junction boxes are fabricated with quick disconnect push-pull connectors to permit easy replacement. Cables that terminate in the control cabinets with the exception of communication and video cables are typically terminated on terminal strips with full ring lug connectors whenever possible.

All the generic facility cables are routed via a cable tray system. Cables leaving the tray system will terminate in a junction box at the appropriate connection points where they will connect with an extension cable that extends to the equipment junction boxes. From these junction boxes, these cables are then secured to the equipment and routed to the appropriate component viz motor, limit switch etc.,

In Chapter 3, the principle of operation of brushless servomotors and direct numerical processing and control of servomotors using Modicon 3240 automation controllers are explained.

Chapter 3

Servo System

The operational principles of brushless servomotors in general and the technique of **direct numerical processing technology (DNP)** of servo control are discussed in this chapter. This chapter also explains about Modicon FA3240 automation controller used in this setup, C/ROS, the firmware used for the controller and three important software settable servo parameters [3] used in DNP based brushless servomotor control in detail.

3.1 Operational principles of brushless servomotors

The brushless servomotor [15] as the name signifies lacks the brushes and commutator which act as the rectifier in a conventional D.C. motor and instead has a device for making the current flow to fit the rotor position by controlling the power source. In the d.c. motor, torque variation is reduced by increasing the number of commutator segments whereas in the brushless motor, torque variation is reduced by making the coil three-phase and by transforming the current of each phase into a sine wave.

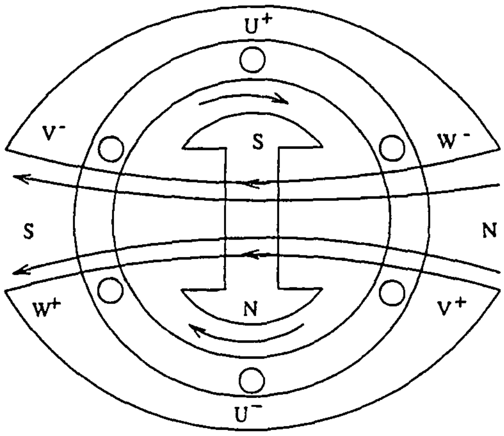


Figure 3.1: Principle of a rotating field

Figures 3.1 and 3.2 are cross-sectional views of a three-phase synchronous motor. with U^+, U^-, V^+, V^-, $W^+$and W^- indicating the beginning or the end of the coil of each phase. When a motor is

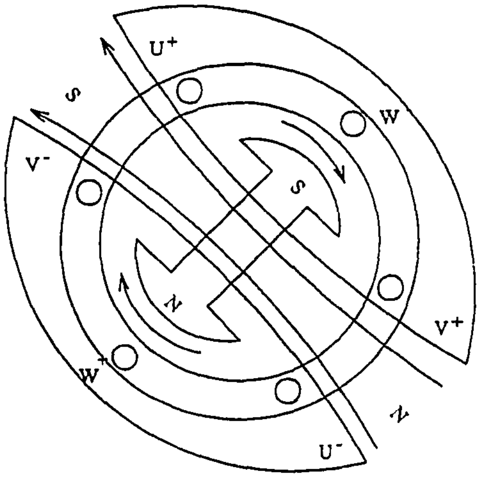


Figure 3.2: Principle of a rotating field

energised by three-phase alternating currents as shown in figure 3.3, only phase U is positive at point A, while phases V and W are both negative. Therefore, the direction of

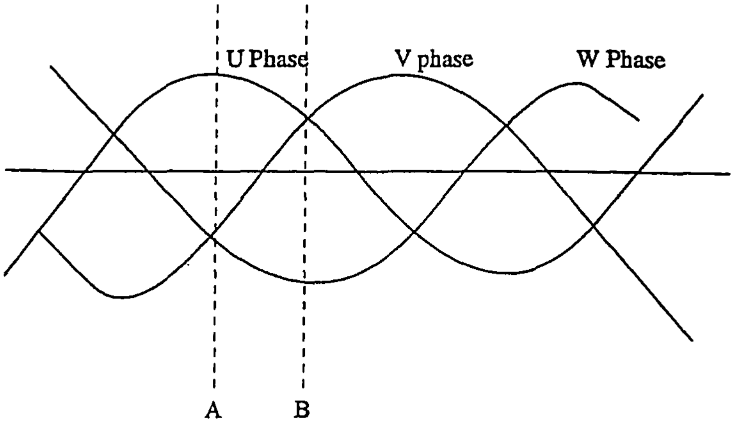


Figure 3.3: Principle of a rotating field

the current at each coil is as shown in figure 3.1 and the composite vector of the magnetic flux induced by the current is generated in the direction from N towards S. If there is a rotor field intersecting the magnetic flux at right angles at that time, torque is generated to rotate the rotor clockwise owing to the repulsive and attractive forces between the magnets. At point B, magnetic flux is generated 60 degrees further clockwise. Thus a continuously rotating field can be obtained by making three-phase currents flow in the stator coil. If the sine wave phase and the rotational position can be made to be always at right angles, it becomes possible to make a highly efficient motor of smooth torque without using brushes.

3.2 Control circuit of Brushless servo motors

In order to control a brushless servomotor, the control device has to perform the task of making the magnetic flux perpendicular to the current. This should essentially be performed by the motor, in addition to the task of controlling the voltage applied to the armature by the d.c. servomotor. An absolute encoder or a resolver is used for the control circuit. Fig 3.4 shows the basic system block diagram of brushless servomotors.

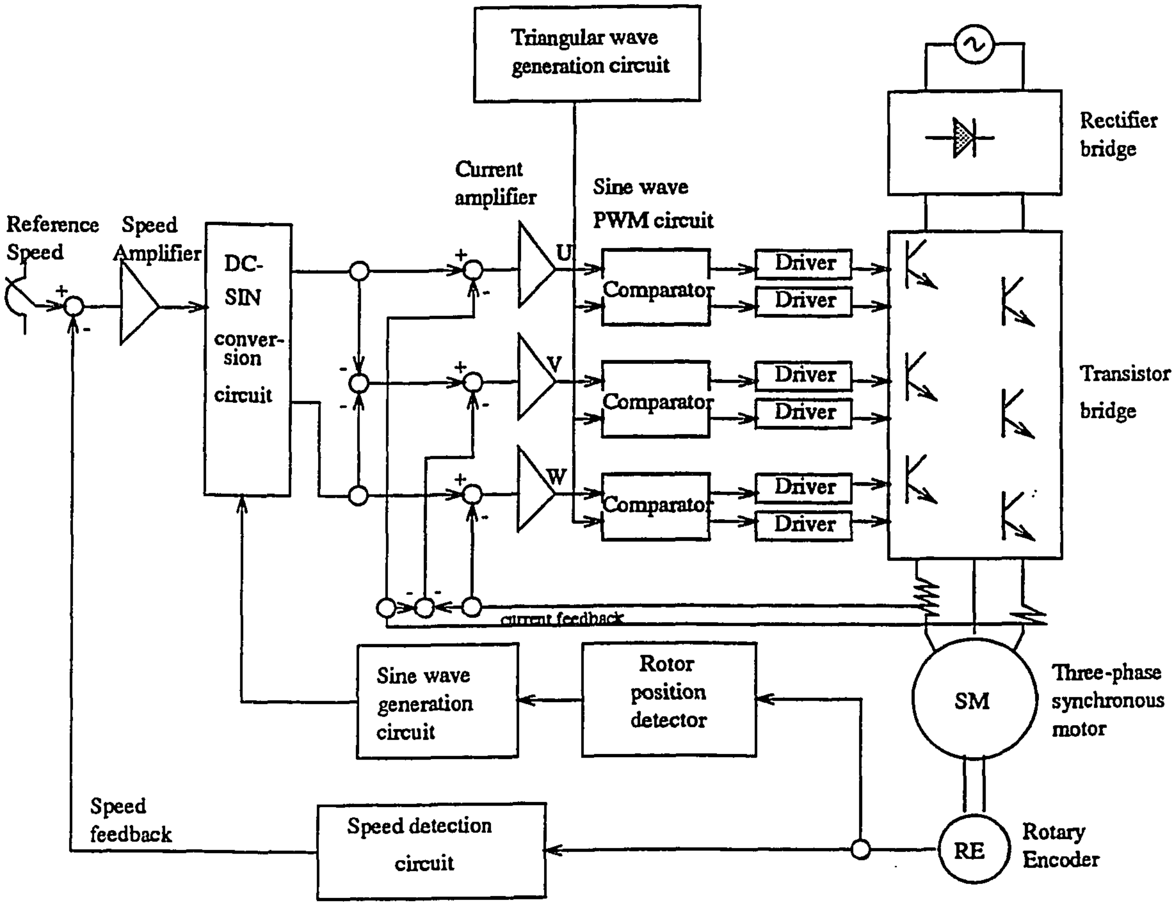


Figure 3.4: Block diagram of the system for brushless servomotors

The brushless servomotor has a rotor position detector, a sine wave generation circuit, a DC-SIN conversion circuit, and a speed detector which do not form part of a D.C. motor.

3.2.1 Rotor position detector

The position of the magnets should be known accurately in order to make the direction of current and the magnetic flux at right angles. The rotor position detector circuit is used for receiving a rotor position signal from the encoder and converting it to a form that can be read by the sine wave generation circuit that follows the rotor position detector. If the encoder or the resolver is eight-bit absolute, a code signal generated by dividing one rotation by 256 is sent to the rotor position detector circuit from the resolver. The code is then converted into pure binary as shown in table 3.1. When each binary digit is properly ordered, the signal is extracted.

3.2.2 Sine wave generation circuit

This circuit is for generating sine waves with the rotor position serving as their phase. It is basically composed of ROM (read only memory). Table 3.2 and figure 3.5 show the contents of ROM and the wire connection for ROM.

As shown, the necessary data, each item of which corresponds to an address, are written on a ROM. Next, when the binary numbers designating the addresses are fed to the address bus connected to the ROM, the data corresponding to the addresses are conveyed close to the data bus. The data signals are then sent to the data bus when read signals are fed to the ROM. By making use of this characteristic, the pattern of the sine wave

Binary digit signal **Rotation of rotor**	**1/256**	**2/256**	**3/356**	**254/256**	**255/256**	**1**
2^7	0	0	0	1	1	0
2^6	0	0	0	1	1	0
2^5	0	0	0	1	1	0
2^4	0	0	0	1	1	0
2^3	0	0	0	1	1	0
2^2	0	0	0	1	1	0
2^1	0	1	1	1	1	0
2^0	1	0	1	0	1	0

Table 3.1: Relation of each binary digit to the angle of rotation of the rotor

Address	Contents
0	data 0
1	data 1
2	data 2
3	data 3
:	:
:	:
254	data 254
255	data 255

Table 3.2: Contents of ROM

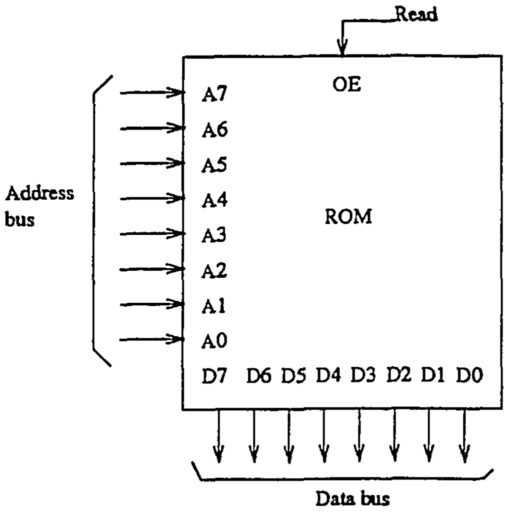


Figure 3.5: Wire connection for ROM

is memorized by one cycle per full rotation in a bipolar motor. and by one cycle per half rotation in a tetrapolar motor. As the brushless servomotor is a three-phase motor. there are three phases with phase differences of 120 degrees. In practice phase V can be estimated by a simple analog operation through the equation V = -(U+V). Therefore. only phases U and W have to be memorized by ROM.

3.2.3 DC-SIN conversion circuit

Two-phase sine waves synchronised with the rotor position are generated in the sine wave generation circuit. However the sine waves are expressed as variations of the amplitude from -1 to +1 through 0, then storing them in 00H-FFH. Hence, these factors have to be converted into current values for practical use.

In this circuit, the sine wave reference current is estimated by multiplying the reference current, which is the output of the speed amplifier, by the amplitude factor of the sine wave. Since, the speed reference signals are fed as direct currents in both the d.c.servomotor and the brushless servomotor, the speed feedback signal to be compared with them should also be a direct current. Accordingly, the speed amplifier output, which is the result of the comparison, is also a direct current. In the case of d.c. servomotor, the speed amplifier output can be used as the reference current without any modification, because the current of the motor to be controlled is a direct current. However, in the case of brushless servomotors, the speed amplifier output has to be converted and used as the reference current as shown in figure 3.6.

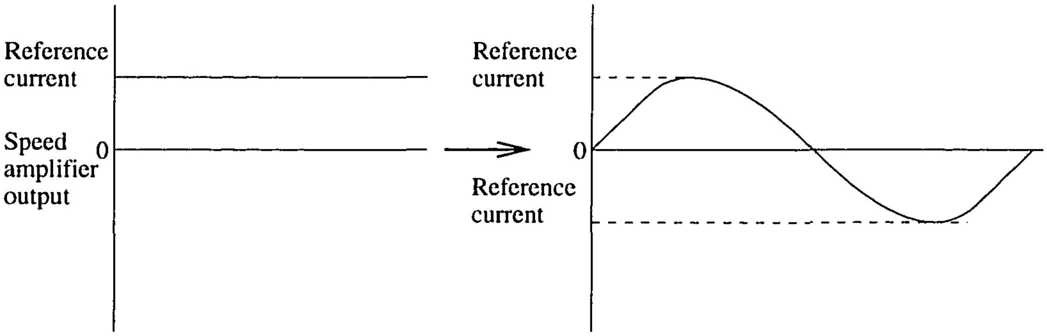


Figure 3.6: Objective of DC-SIN conversion

This circuit consists of the combination of a D/A converter, which converts digital signals at the output of the sine wave generation circuit into analog signals, and a multiplier as shown in figure 3.7.

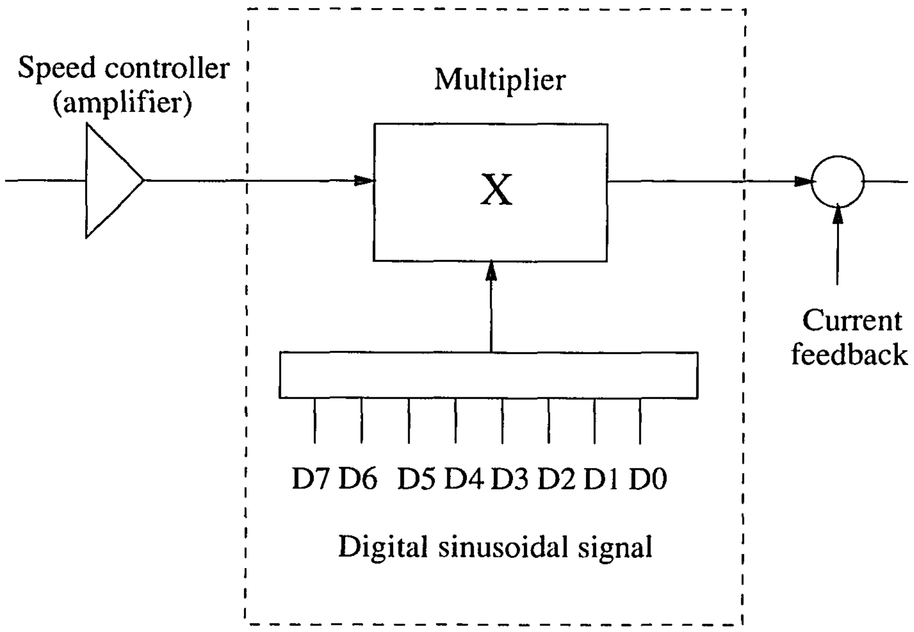


Figure 3.7: D.C. to Sinusoidal conversion circuit

3.2.4 Sine wave PWM circuit

The aim of the brushless servomotor is to make sine wave currents flow in the motor. For this reason it is ideal for a brushless servomotor to have the output of the current amplifier of the sine waves applied directly to the motor after amplifying the power. However, amplification of the sine waves is not practical, as it calls for a power transistor to be used in the proportional region. This makes it difficult to solve the problem of high temperature due to power loss. Consequently, the power loss is reduced by switching the power transistor. This method is called PWM (pulse width modulation). In this method, the current of a motor is converted into a controlled pulse of width proportional to the amplitude of the sine wave so that it may become a sine wave on the average. The principle of PWM is shown in figure 3.8.

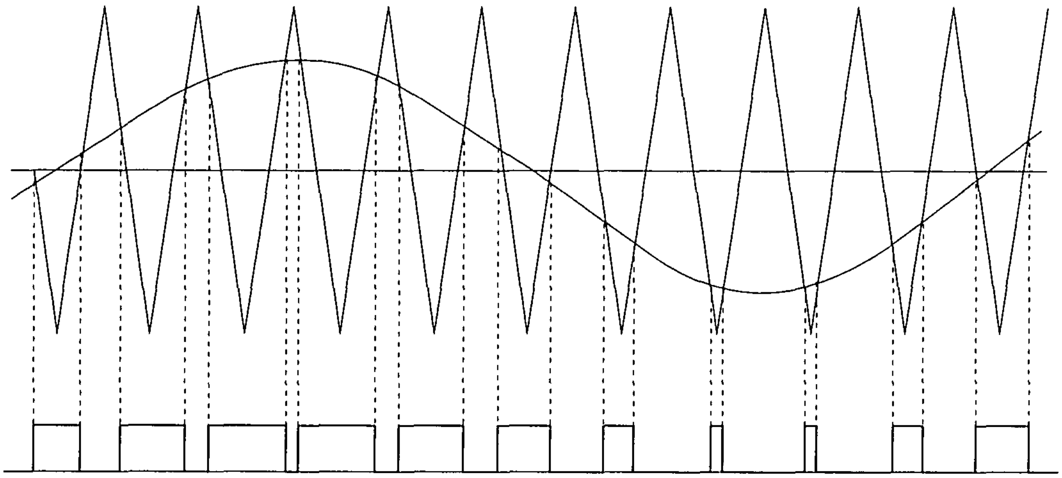


Figure 3.8: Principle of sine wave PWM

A triangular carrier wave oscillating with constant frequency and amplitude, and the sine wave output from the current amplifier are compared by a comparator. As shown in the figure, pulses of unequal widths are output by extracting the portions where the values of the sine wave exceed those of the carrier wave. The duty ratio of the pulse width is increased or decreased, centring around 50%, by the sine wave and modulated to make a sine wave on the average as the inverter output is 0 V when the duty ratio is 50%.

The method of selecting the oscillating frequency of the chopping wave is important; as the carrier frequency equals the switching frequency of the power transistor, it increases the switching loss proportionally as it is made higher and reduces the speed of response of the servomotor as it is made lower. Furthermore, ripples appear more frequently, and the torque change and core losses are increased.

Generally, a carrier frequency of 1-3 kHz is selected when the inverter consists of bipolar transistors; 5-20 kHz is selected when the inverter consists of FETs. The current ripples developed at these levels of frequency make the iron core of the motor vibrate and this generates unpleasant noises when the frequency is within audio range. To resolve this problem, the carrier frequency is made 16 kHz or more by using FETs. Alternatively, the generation of noise is prevented by moulding the iron core and the coil of the motor into one body.

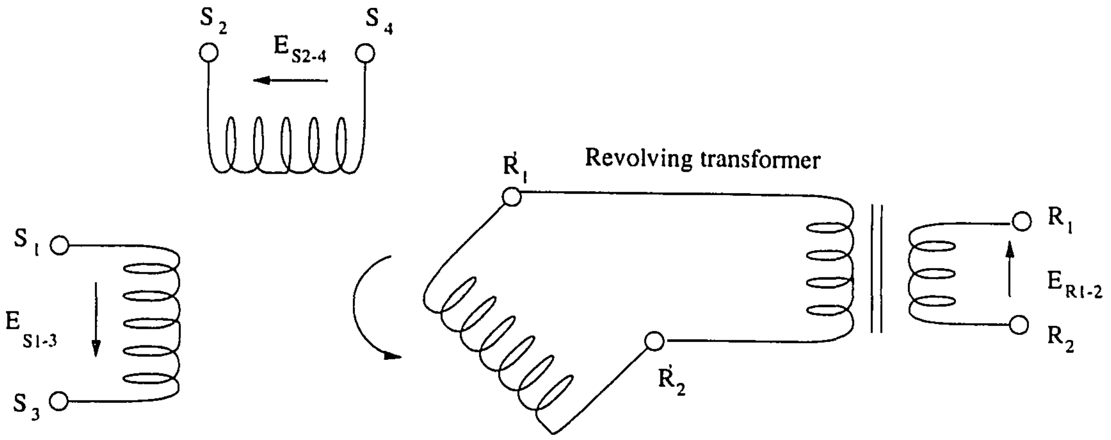
3.2.5 Speed detector

The brushless motor uses a rotary encoder or resolver for position and speed detection. Encoders or resolvers can be classified as absolute or incremental. While the encoder converts the amount of displacement into digital form, a resolver converts it into an analog form. A resolver equipped with a rotary transformer is called a brushless resolver.

The brushless resolver consists of a stator, a rotor, and a rotary transformer. The windings of the stator and the rotor are distributed so that the magnetic flux is distributed in the form of a sine wave to the angle of rotation. The stator winding, which is the excitation winding, is of two-phase structure, with an electrical phase difference of 90 degrees. The rotor winding, which is the output winding, is either one-phase or two-phase winding depending on use. Figure 3.9 shows the connection diagram and the relational expression for a resolver of single-phase output.

3.2.6 Detection of Position signal

Figure 3.10 shows an example of a circuit converting the signal of a resolver into a digital signal.



$E_{R1\text{-}2} = K_1\,(E_{S1\text{-}2})\cos\theta - E_{S2\text{-}4}\sin\theta$

K_1 - ratio of voltage transformation

θ - rotational angle of reslover

Figure 3.9: Connection diagram and relational expression for output of resolvers of single-phase output

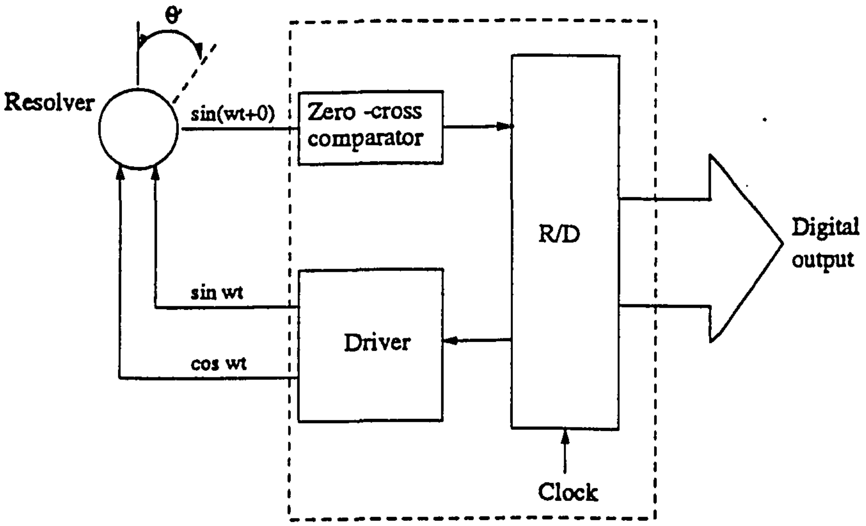


Figure 3.10: Conversion circuit implementing an R/D converter

The conversion circuit uses a dedicated converter capable of exciting the stator and processing the output of the resolver. The output signal of the converter is widely used as the signal for detecting absolute values, incremental values, and the input signal to a computer.

3.2.7 Detection of Speed signal

Speed detection using resolver are done by two methods. In the first method, the incremental signals obtained by the circuit are counted per unit period. The second method called the differential method calculates the difference of position signals per unit period for detecting the absolute value.

In this setup, resolvers are being used in place of rotary encoders for the following reasons.

- They are highly resistant to environmental conditions such as vibration and impact.
- They have lower sensitivity to environment.
- They are usable over a wide temperature range.
- They are capable of long-distance signal transmission.

3.2.8 Speed control circuit

The servo-driver, which is the heart of the control circuit of a servomotor, is called the servoamplifier because the deviation-amplifying function of this circuit is responsible for the performance of the whole system. When the circuit receives analog signals as inputs, the difference between the reference speed and the speed feedback signal is amplified. The structure of the circuit is as shown in figure 3.11.

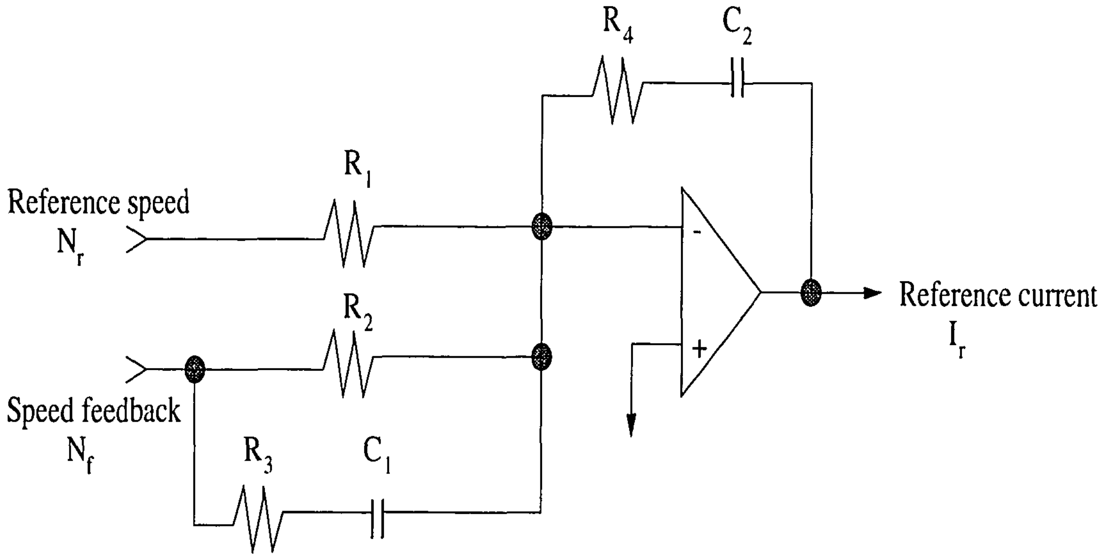


Figure 3.11: PID amplifier

This simple circuit essentially consists of one operational amplifier. The ratio of R_1 to R_4 represents the proportional gain, and R_4 and C_2 decide the differential time constant. The operation of this PID controller can be expressed by the following equation

$$I_r = K_p(N_r - N_f) + K_i \int (N_r - N_f)dt + K_D \frac{d(N_r - N_f)}{dt} \tag{3.1}$$

where K_p, K_i and K_d are the constants of the PID. The error is calculated once and the result is stored in memory and is used for calculation performed the next time. The above equation can hence be transformed into equation (3.2) where $\Delta N_n = N_r - N_f$; at the time one interval before is ΔN_{n-1} and at the time two intervals before is ΔN_{n-2} and so on.

$$I_{rn} = K_p(\Delta N_n - \Delta N_{n-1}) + K_i(\Delta N_n) + K_D(\Delta N_{n-2} \times \Delta N_{n-1} + \Delta N_{n-2}) + I_{rn-1} \tag{3.2}$$

In general the sampling time which is the time required by the CPU to perform the same calculations in regular periods should be 1/10 of the step response or less.

3.2.9 Direct Numerical Processing

Direct Numerical Processing technology (DNP) [3] is a higher performance method of servo control developed by Modicon. In this technology, a single brushless resolver generates all commutation, speed and position information for the controller, eliminating the individual feedback devices required in a conventional servo control system. By using the resolver's position information, software closes both the position and velocity loops, including all servo compensation and gains. The software parameters can be set repeatedly in a digital controller which simplifies maintenance.

The block diagram of a DNP servo loop is as shown in figure 3.12.

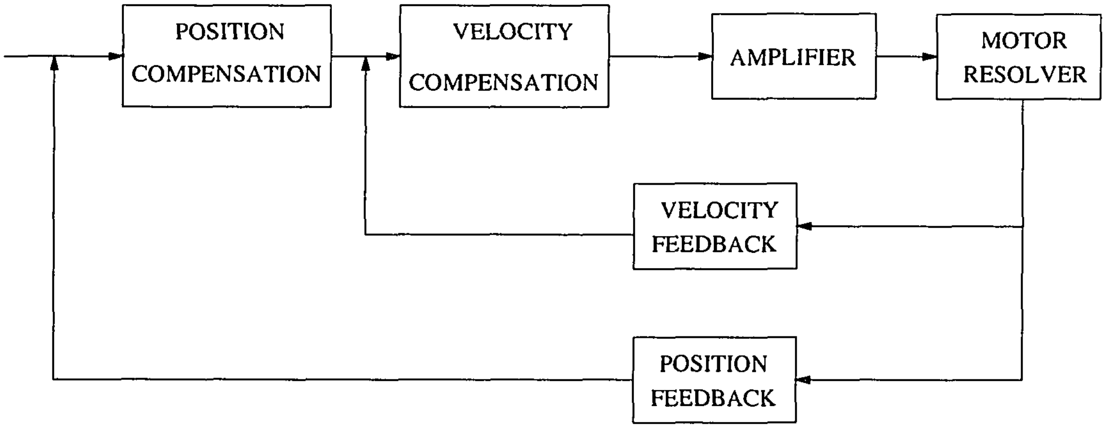


Figure 3.12: Block diagram of a DNP servo loop

The inner loop is for speed control and the outer one for position control. The speed loop must be stabilized before stabilizing the position loop. The behaviour of the system is split into two major blocks and analysed individually.

3.2.10 Speed control loop

- Servo Amplifier

  A PWM transistor amplifier with a DC gain K_a of 12 amps per volt and break frequency F_a of 333 Hz is used in the control loop. The amplifier has the following transfer function

$$G(s) = K_a/(1 + s/F_a) \qquad (3.3)$$

- Motor and Load

  A DC motor produces an acceration proportional to current, and speed is equal to acceleration multiplied by time. For a constant current, speed is therefore given by

$$V(t) = I_m \times K_m \times t \qquad (3.4)$$

where V(t) is the speed at time t, I_m is the motor current and K_m is the acceleration per amp. For a sinusoidal input, the speed is given by the following relationship

$$V = I_m \times K_m/f \qquad (3.5)$$

where I_m is the amplitude of the applied sinusoidal current waveform, V is the amplitude of the resulting sinusoidal velocity waveform, K_m is the motor acceleration per amp and f is the frequency of the applied current waveform. But the motor acceleration per amp K_m can be expressed in terms of motor torque K_t and the total system moment of inertia J, as $K_m = K_t/J$. The transfer function of motor and load is thus given by

$$G(s) = K_t/(J \times s) = K_m/s \tag{3.6}$$

- Velocity feedback

The DNP servo measures speed by monitoring motor position every millisecond and calculating velocity. The fact that this is a sample based system limits the frequency response to about 1/15 of the sample frequency of 1KHz, thus F_v is equal to 66.7 Hz or 420 rad/sec. The gain is measured in counts per millisecond per radian per second, where each count is equal to 1/16384 revolutions. Here it may be noted that the axis controller divides every resolver (motor) revolution into 16384 parts. DNP allows setting up any measurement system opted by the user and the axis controller card automatically does the number conversions. Termed as 'resolution' in DNP technology, this sets the smallest increment of position that can be commanded. Setting up of resolution is explained in greater detail in appendix A. Since for internal reasons the velocity feedback block includes a multiplication by 4, the resulting gain at one radian per second is $K_v = 16384/(2 \times \pi) \times 4/1000 = 10.43$.

The transfer function for the velocity feedback is given by

$$G(s) = K_v/(1 + s/F_v) \tag{3.7}$$

and the combined system transfer function without compensation is given by

$$G(s) = K_a/(1 + s/F_a) \times K_m/s \times K_v/(1 + s/F_v) \tag{3.8}$$

- Lag compensation algorithm

The stiffness of the system is the amount of steady state torque that will produce one radian per second of velocity error, which is proportional to the DC gain. The system stiffness can be increased either by accomplishing the desired DC gain by selecting amplifier motor and load with appropriate values of K_a, K_t and J respectively or by incorporating a compensation block with a desired DC gain. The lag compensation algorithm increases the stiffness of the system without decreasing the system phase margin. This algorithm has a DC gain of RPG × 10/127 volts per pulse per millisecond and the break frequency of the system is RIG radians per second. Here RPG and RIG are DNP servo parameters which are software settable. Rate loop proportional gain (RPG) compensates for the variation in acceleration capability between motor and drive combinations of various types. RPG is inversely proportional to the speed at which the motor and drive can accelerate the load. Rate loop integral gain (RIG) compensates for the bandwidth of the servo drive amplifier and the mechanical load bandwidth. The transfer function of the lag compensation block is given by

$$G(s) = K_c \times (1 + s/F_c)/s \tag{3.9}$$

where $K_c = RPG \times 10/127$ is the DC gain and $F_c = RIG/2 \times \pi$, the break frequency. Adding lag compensation provides for a major increase in stiffness though at the expense of bandwidth. At higher gains, the system bandwidth will be about 63% of the bandwidth without lag compensation. Since the overall gain is proportional to the system inertia, in applications where the inertia varies considerably, reducing RIG provides for better stability over the range of inertia.

3.2.11 Position control loop

As with the velocity feedback system, the frequency response of the position loop feedback is limited by the sample frequency of the DNP system. The gain is measured in counts per radian and is given by $K_p = 16384/(2 \times \pi)$ units per radian. The overall DC open loop gain of the position system is given by $K = K_h \times K_p$ where K_h is the closed loop gain of the speed control loop. The variation of this gain to make the system stable is accomplished by another servo parameter called Acceleration time constant (ATC) which is also software settable and forms part of the position control loop. The acceleration time constant (ATC) sets the acceleration and deceleration rates for the basic exponential-shaped motion profile by controlling the position loop gain in the DNP servo system. Thus the position loop compensation DC gain K_{pc} is given by 4/ATC where 4 is the factor by which the overall DC open loop gain of the position system needs to be reduced to bring the gain at crossover frequency to 1 and hence make the system stable.

The command to the speed loop is always equal to 1000/ATC times the position error.

$$E(t) = P(t) - Pc(t) \tag{3.10}$$

where E(t) is the position error at time t, P(t) is the actual position at time t and Pc(t) is the position cammand at time t. Hence, the velocity V(t) at time t is given by

$$V(t) = (P(t) - Pc(t)) \times 1000/ATC \tag{3.11}$$

assuming a negligible velocity error which is valid as long as the amplifier does not enter current limit. Thus when the motor is at rest and a command is given to run at constant speed, the position command at time t is given by $Pc(t) = V \times t$ where V is the speed command. By integrating and substituting into the equation for V(t), we have

$$V(t) = V \times (1 - e^{t}/k) \tag{3.12}$$

where k = ATC/1000 and e is the exponential operator. ATC is the parameter which also allows synchronised motion of mutiple motors.

Criteria for setting ATC

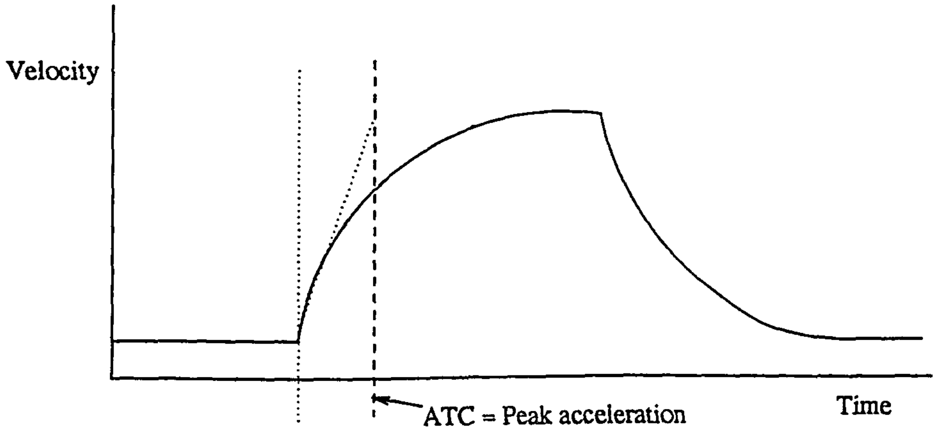
The correct value is controlled by three considerations:

- The minimum value that is consistent with stablity in the servo loop.
- The minimum value that will allow contouring.
- The minimum value that will allow point to point positioning without overshoot.

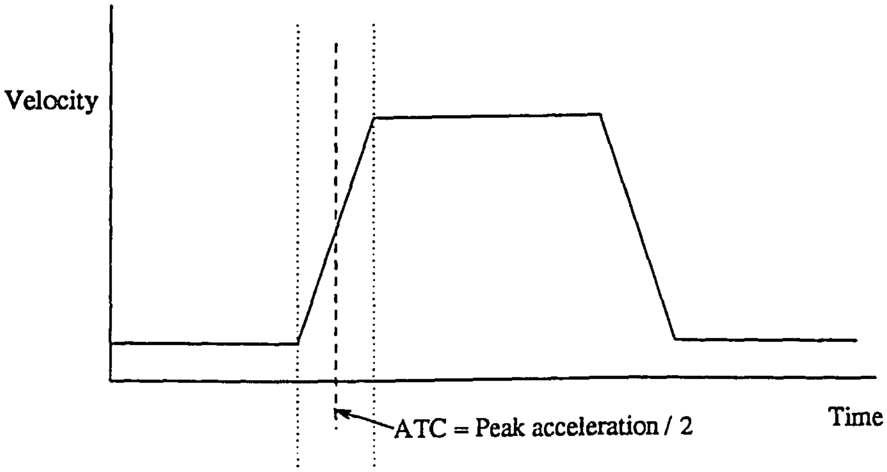
There is however a limit for a low value that can be used. When ATC is set to half the motor's maximum possible acceleration, drive saturation limits the profile shape and gives a trapezoidal-shaped profile as shown in figure 3.13.

Further decrease in ATC would result in overshoot and possible low frequency oscillation. The contouring and overshoot constraints are however removed by another software settable parameter, Linear acceleration time constant (LATC), which is essentially the time taken by the motor to accelerate to full speed. The servo system is now modelled as shown in the block diagram 3.14 incorporating these servo parameters.

Figures 3.15, 3.16 and 3.17 show the change in velocity responses simulated for changes in RPG, RIG and ATC respectively keeping the other servo parameters unchanged.



Exponential velocity profile



Trapezoidal velocity profile

Figure 3.13: Velocity profile for different ATC values

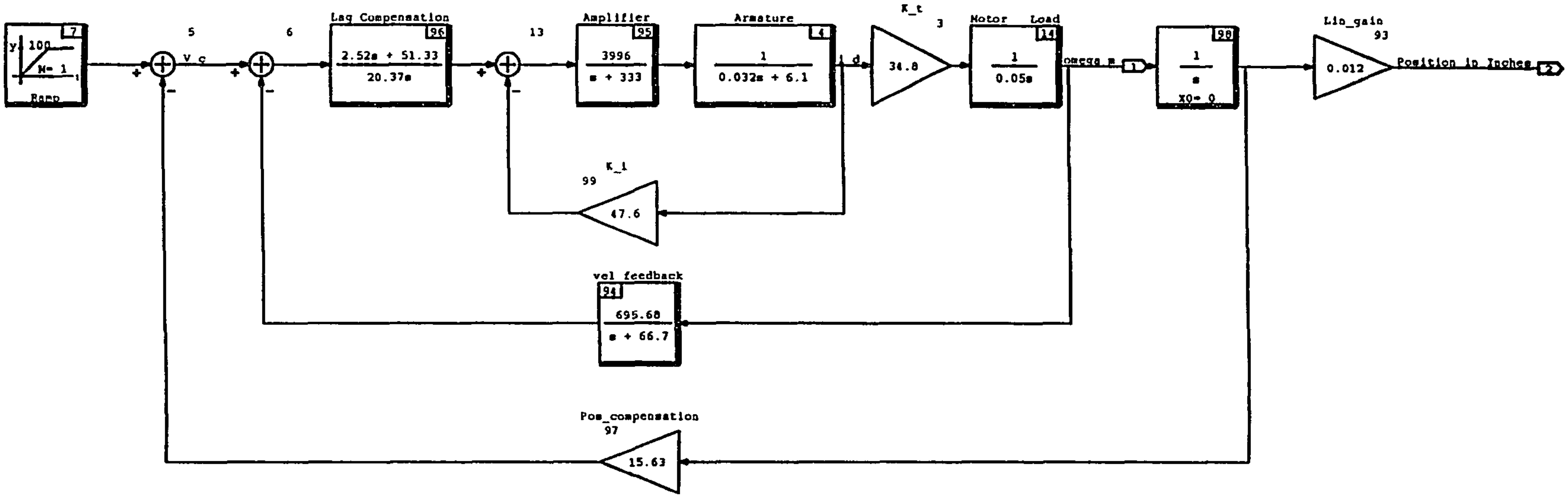


Figure 3.14: Block diagram of brushless servomotor control

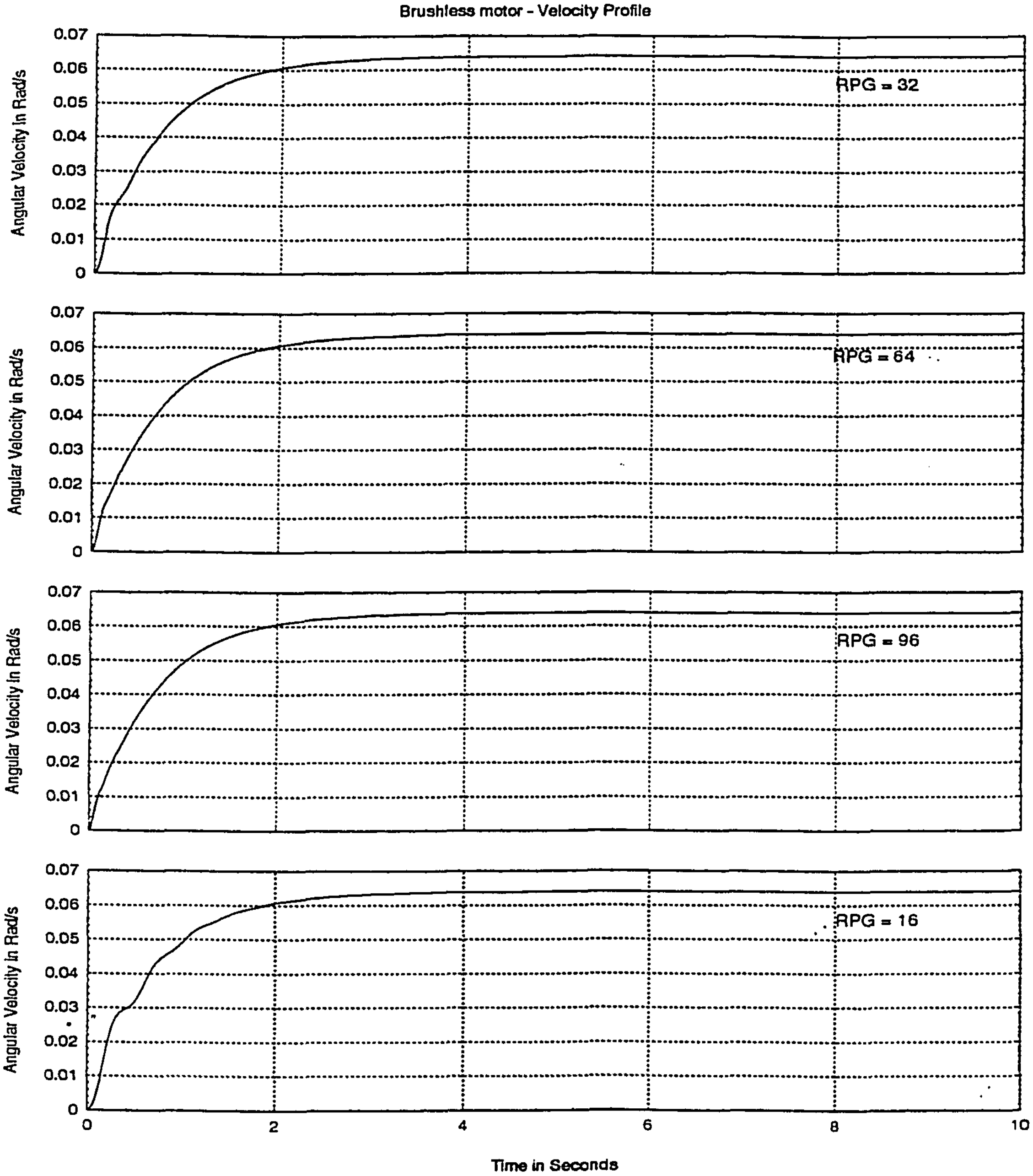


Figure 3.15: Effect of changing RPG on velocity response

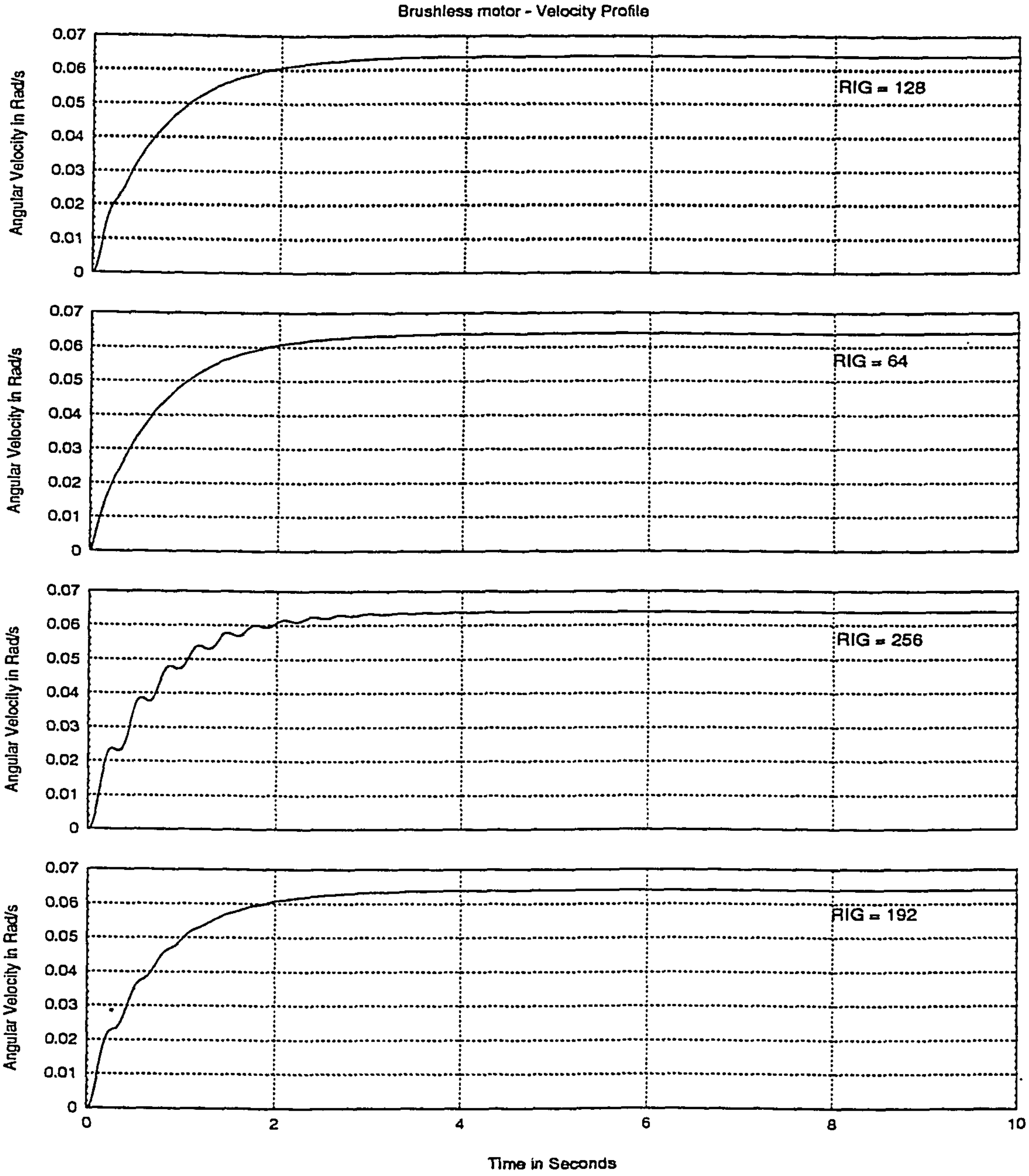


Figure 3.16: Effect of changing RIG on velocity response

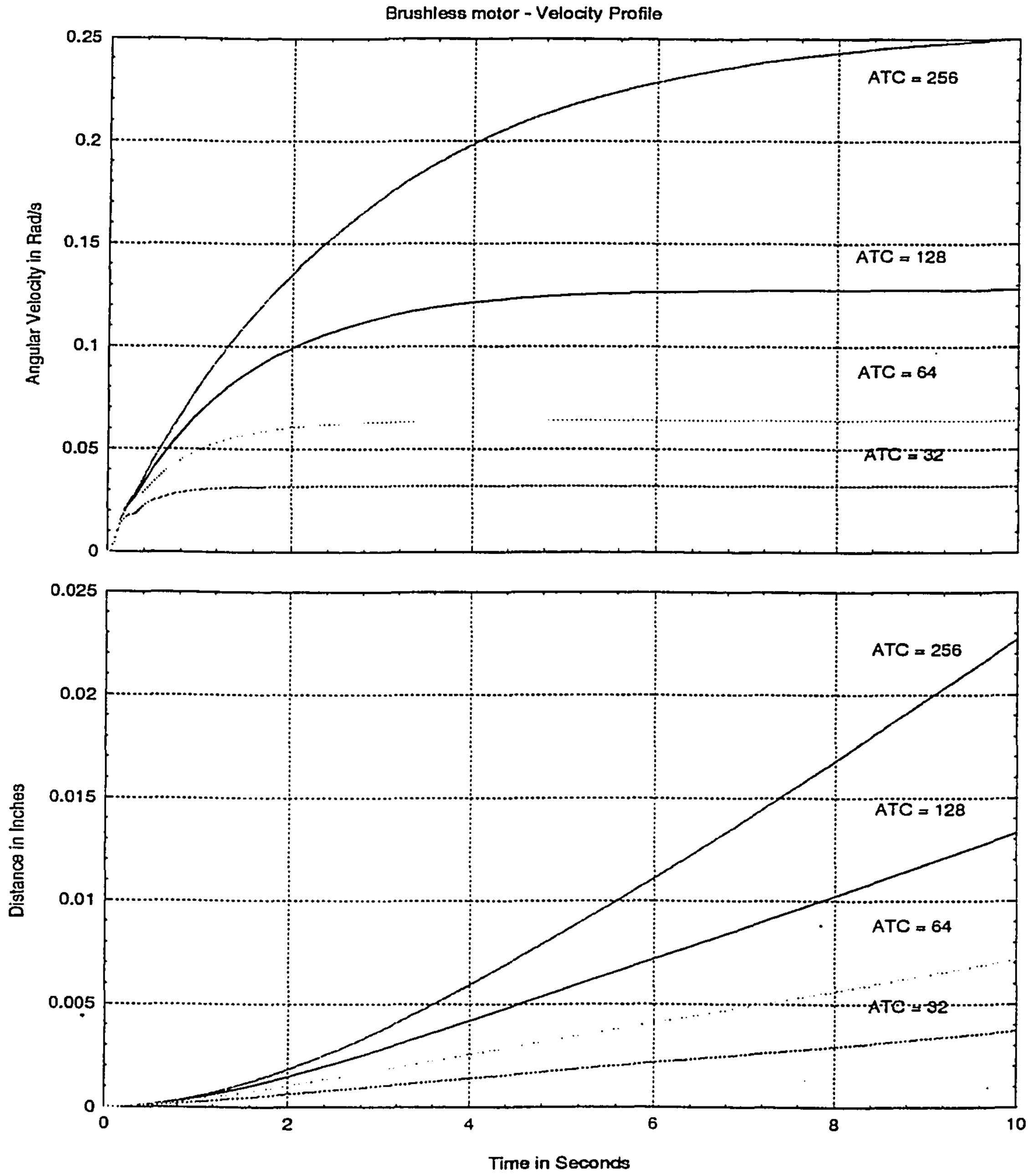


Figure 3.17: Effect of changing ATC on velocity and position response

3.3 Automation Controller

Modicon FA3240, the automation controller used in the system is capable of controlling upto four axes either independently or in a coordinated fashion. This controller offers a plug-in memory cartridge and an easy interface to a Modicon 984 PLC via the S908 remote I/O processor. There are four automation controllers in all, the first one controlling all the three axes of RBT-II, the third controlling the bridge and trolley motions and the fourth, the telescoping tube motion of RBT-I respectively. The second controller can perform the same function as the third controller except that it controls the second motor of the bridge and tube. Since the X-Y-Z assemblies of RBT-I are operated using one motor, the second controller is essentially used as a spare controller in the event of a breakdown to the third controller. Each 3240 controlled equipment is autonomous and can operate independently of other machines.

The block diagram of a typical 3240 system is as shown in figure 3.18.

As already stated, the main unit of each 3240 controls up to four drives and their end-of-travel and home limit switches. The main unit consists of the following components:

- **The central processor unit** which performs the logic processing in accordance with the current programming.
- **The power supply** which receives 120 VAC input supply and provides the required operating voltages for the remaining components.
- **The Axis controller** which is capable of controlling up to four separate axes (servo motors) to produce independent or coordinated motion.
- **Four serial communication ports** which are jumper configurable for either RS-232

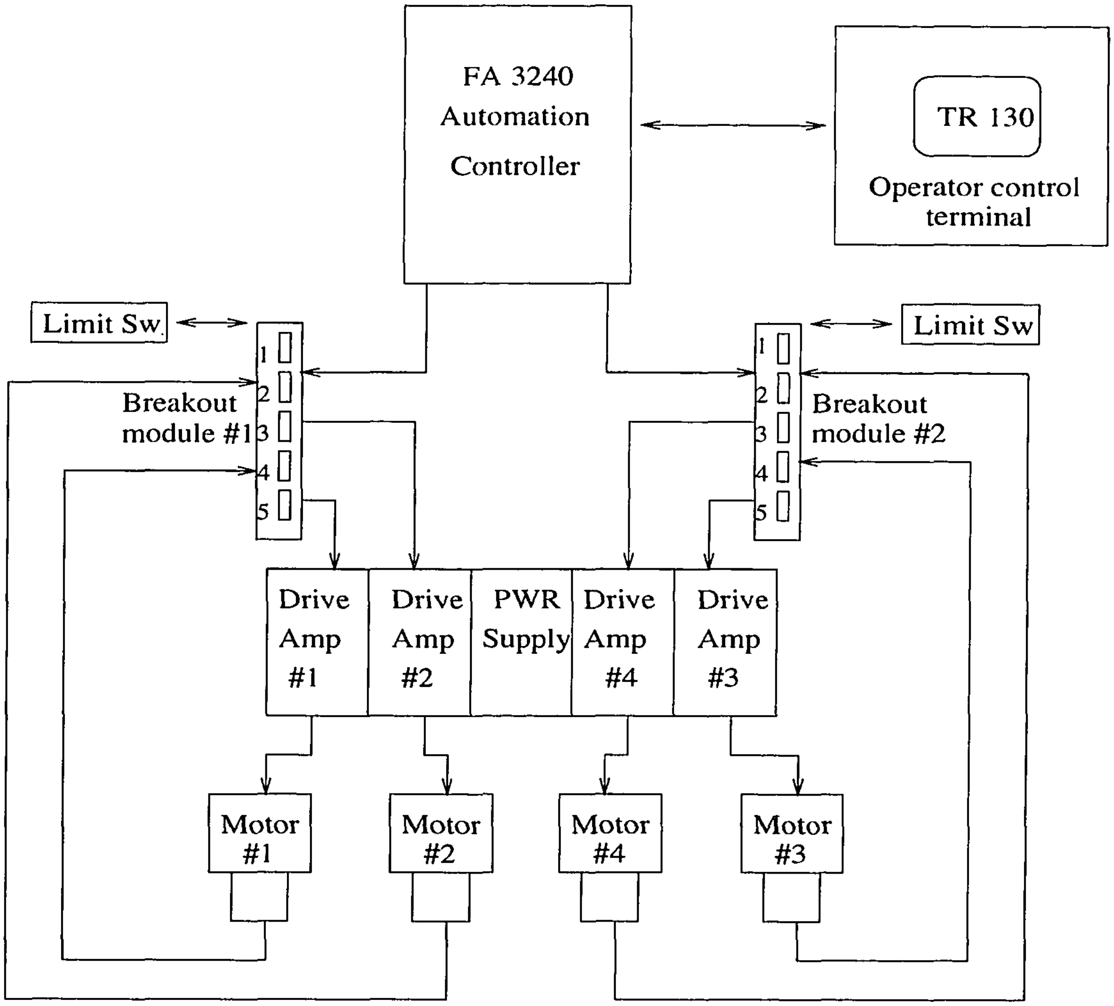


Figure 3.18: Block diagram of FA3240 system

or RS-422 operation.

In this system, one port on each 3240 is configured for RS-422 while the remaining three support RS-232. The RS-422 interface connects the 3240 controller to the supervisory station computer to allow downloading of part-programs. One RS-232 port is used for communications with the TR-130 operator control station. Up to two RS-232 interfaces may be used to connect multiple 3240s into a cluster configuration of a single master with upto two slaves dedicated to a single machine. The S908 remote I/O interface provides the high speed link between each 3240 and 984B PLC. The 984B acts as a command arbiter for the 3240s and a data interface between the 3240s and the supervisory control station computer.

3.3.1 Program module

The program module which plugs into the CPU board of the 3240, holds the firmware for the controller which is basically an application specific program. The program is in either EPROM or EEPROM (electrically erasable programmable read only memory) depending on the system configuration. The ROM portion of the controller's memory stores the main program while the RAM stores the part program which the controller uses to drive the axes. The contents of RAM are protected while power is turned off. If power is removed and the program module remains plugged into the 3240, the contents of RAM are protected for about 30 days.

3.3.2 Breakout Modules

Breakout modules are used to link the 3240 to the drives, resolvers, and end of travel limit switches. Each breakout module has five terminal blocks. Terminal block 1 is connected to the clockwise (CW) and counterclockwise (CCW) limit switches and 24 VDC supply to them is routed through this terminal block. The sine and cosine feedback signals from the resolver to the controller can be measured at terminal block 2 of the breakout module. Terminal block 4 performs a similar function but for a different axis.

The torque command signals for the three phases supplied from the 3240 controller to the drive amplifier can be measured at terminal block 3 of the breakout module. The enable command signal issued from the 3240 to the drive amplifier is routed through this terminal block. The fault signal, controlled by the drive amplifier, in the event of a over temperature, improper bus voltage or improper torque commands, can also be measured at this terminal block. Terminal block 5 of the breakout module is identical to terminal block 3 and performs the same function but for a different axis. Each 3240 unit can connect to one or two breakout modules, with each module supporting one or two drives each, for a total of up to four drives.

3.3.3 Control Inputs and Outputs

The FA3240 can interface with up to 30 control inputs and outputs in a mixed configuration. Control inputs send signals to the FA3240 unit to let it know the status of a portion of the system. Switches, relay contacts, temperature sensors and float levels are some examples of control inputs. The 3240 also controls outouts such as lights, buzzers relays etc., The 3240 is connected to these inputs and outputs through isolator modules located on the I/O board which can transform signals for use by the controller and also isolate them for protection and noise immunity. Each I/O connection is protected by a 1 -amp fuse located on the axis board.

3.4 3240 Firmware - C/ROS

Cyber Robot Operating System, C/ROS, [16] as it is called in practice, is the program module that resides in the controller. C/ROS is a complete operating environment for a flexible robot system and allows for creating, maintaining and executing robotic functions of any type and complexity. C/ROS also provides the following features:

- Teaching of robotic motion points for upto four axes with coordinated, tandem or spindle motion with facility of storing of over 500 taught points and 2000 lines of program.
- Simplifies input of machine parameters for the robotic system.
- Supports serial communication between the 3240, robotic equipment and external equipment such as computers and vision systems.

3.4.1 C/ROS remote I/O processor mapping

C/ROS software recognizes 48 logical inputs and 48 logical outputs [17]. The input and output setup menus enable mapping these logical I/0 points to any of the 96 points in the controller's state table. A single physical point in the state table may be mapped to more than one C/ROS logical input or output and all such mapped logical inputs/outputs will receive the same inputs or outputs respectively. I/O points 0 to 29 can be used either as an input or an output by plugging in the appropriate opto isolator module at the designated location on the I/O board. Since these inputs/outputs are hardwired, they are referred to as physical I/O. I/O points 30 to 95 are the virtual I/O points and can be mapped to either 3240 logical outputs or inputs using the respective configuration menu. It is however not necessary that all I/O points be mapped. The virtual input/output points are useful in performing various operations such as homing, switching between setup and run modes etc., from the supervisory control computer without having to hard wire the inputs.

3.5 Actual Setting of servo parameters

Based on the results obtained from simulating the model of brushless servomotor, the three software servo parameters have been set as follows.

- Rate loop proportional gain **(RPG)** = 32
- Rate loop integral gain **(RIG)** = 64
- Acceleration Time Constant **(ATC)** = 64

Basic operational principles of brushless servo motors and control of brushless servo motors using Modicon FA3240 automation controller have been detailed in this chapter. Chapter 4 discusses about the programmable logic controller used and how communication between the PLC and the automation controller is effected. Appendix A contains a listing of other important axis parameters in C/ROS.

Chapter 4

Programmable Logic Controller

This Chapter discusses in detail about Modicon 984 control system, theory of operation of 984B Programmable Logic Controller used in the system and other aspects such as components, memory and communication of the controller. It also explains how the PLC has been configured to suit system requirements and throws light on programming aspects of PLC.

4.1 Introduction

Programmable controllers were originally designed to replace relay-based logic systems and solid-state hard-wired logic control panels. Their advantages over conventional logic systems are that they are easily programmed, highly reliable, flexible, relatively inepensive, and able to communicate with other plant computers.

A programmable controller examines the status of inputs and, in response, controls some process or machine through outputs. Combinations of input and output data are referred to as logic. Several logic combinations are usually required to carry out a control plan or program. This control plan is stored in memory using a programming device to input the program into the system. The control plan in memory is periodically scanned by the processor, usually a microprocessor, in a predetermined sequential order. The period required to evaluate the programmable controller program is called the "scan time".

4.2 Components of a Programmable Controller

All programmable controllers share the same basic components and functional characteristics regardless of their size, cost and complexity. A programmable controller consists of a processor, an input/output system, a memory unit, a programming language and device, and a power supply. A block diagram of a typical programmable controller system is shown in figure 4.1 [18].

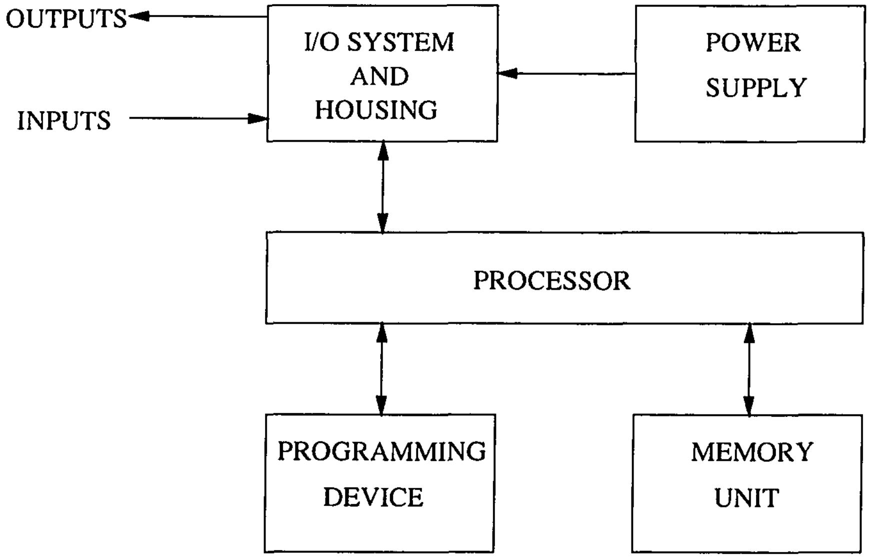


Figure 4.1: A Typical Programmable Controller System

4.3 Modicon 984 series control system

The setup makes use of Modicon 984 series control system for total control. The 984 controls the application based on data received from the input/output modules connected to devices. Figure 4.2 represents block diagram of a 984 controller [19].

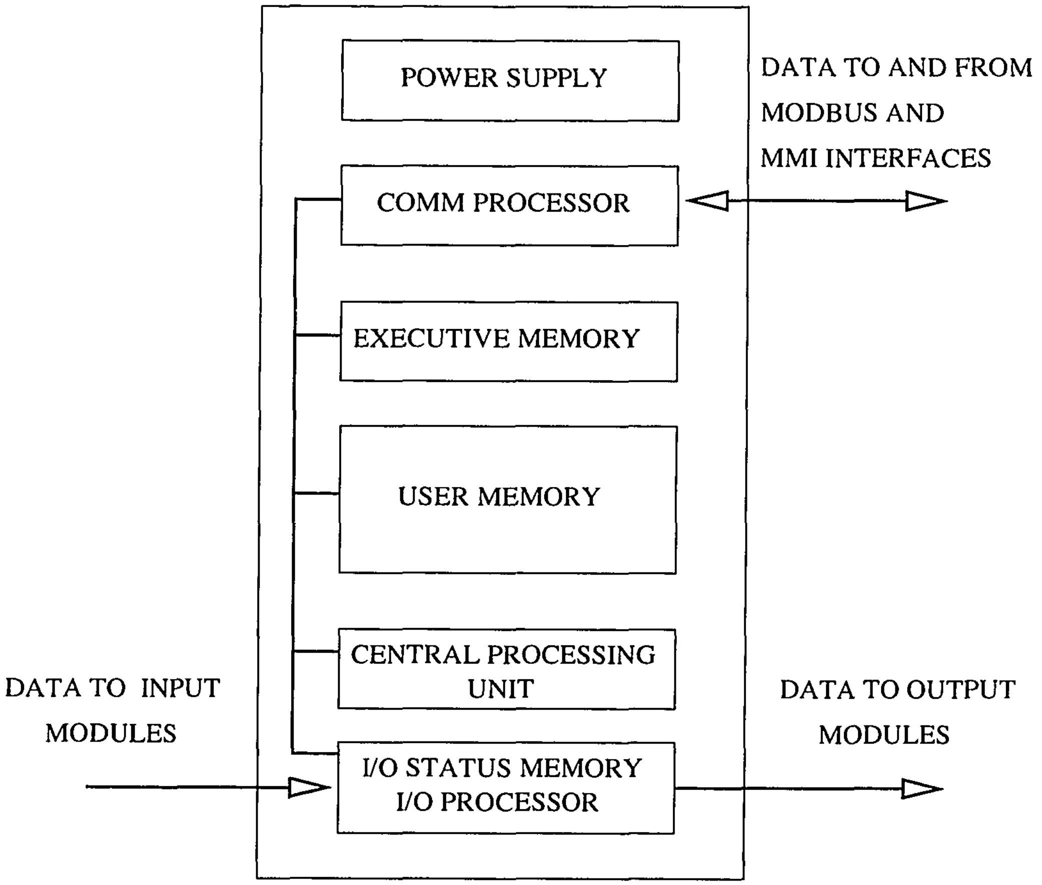


Figure 4.2: Block diagram of Modicon 984 Controller

Input modules accept electrical signals from field sensing devices and convert them into acceptable voltage levels for CPU processing. Output modules receive electrical signals from the CPU and convert them into voltage or current levels necessary to activate the switching devices. The 984's central processing unit solves user logic at very fast, regular intervals, making control predictable. The logic determines what actions to take, based on data received from the input modules. The resulting changes in output states are then forwarded to the field. A functional block diagram of a 984 control system is shown in figure 4.3 [19].

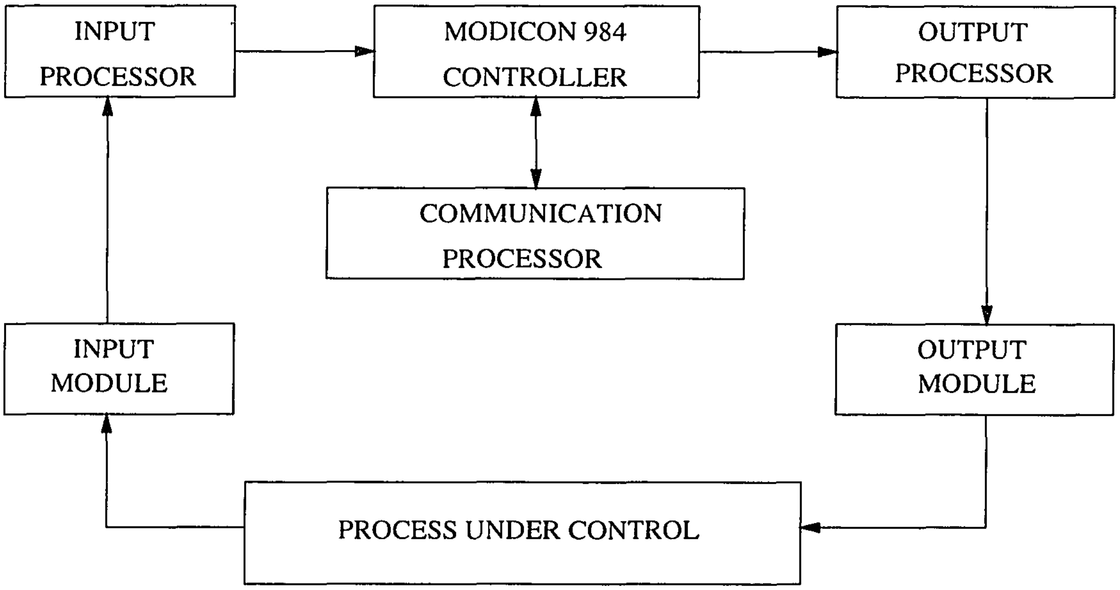


Figure 4.3: Functional block diagram of Modicon 984 Control system

4.3.1 System Memory

The 984 system memory is based on CMOS technology [19] with battery backup to maintain integrity during power loss. A Memory Protect switch, on the controller prevents unauthorised alteration of the user program. 984 system memory consists of two types namely

- **Executive Memory** which resides in nonvolatile memory and

- **User logic, configuration data and system status data** which resides in volatile Random Access Memory (RAM), backed up by batteries. User logic is the control logic used to program the application while configuration data is used to define system set-up and resides in a data table. The input/output status, whose current electrical status is updated each time the PLC scans the user logic, is stored in state RAM.

4.3.2 Input/Output Systems

The application logic that is stored in and solved by the PLC is implemented by input and output modules. I/O modules are field-wired to sensing or switching devices and linked to the PLC over an I/O bus to create a complete control system. I/O subsystems may be local, being located together with or in close proximity to the PLC or remote, being located at far-off distances of upto a maximum of 4.5 km from the PLC, depending on the cable type. Local I/O communicates to the PLC across a housing backplane or local cables. Upto 5 housings may be connected to any remote I/O drop. Remote I/O communicates through a remote I/O interface installed at each I/O location.

4.3.3 Communications

Peripheral devices, such as programming panels or computers are directly connected to the 984 PLC through built-in Modbus ports. Modbus Plus is a peer-to-peer, token-bus network with a speed of one megabit per second used for data acquisition and program editing. The 984 PLC also has ports to support ASCII communications.

4.3.4 Theory Of Operation

- Scan

The 984 controller scan consists of reading inputs, solving logic, and servicing outputs, which is not a random process. The logic program is divided into segments which control specific channels of I/O. Before solving the logic for a segment, the inputs for that segment are read. While the logic is being solved, the controller reads the inputs for the next segment. While the logic solve is complete for a segment, the controller services the I/O for that segment and begins solving the logic in the next segment. The order of solve for segments can be varied using the Segment scheduler.

The scan is further broken down within each segment. Segments are divided into networks. Networks contain a combination of programming elements which reside in an eleven column by seven row format with the eleventh column meant exclusively for coils.

The network solve begins in column one, row one and proceeds from the top row to the bottom row. The solving continues in column two, row one. Each column is solved from the top to the bottom starting with column one and ending with column ten. After completing one network, the solving continues in the same manner in the next network.

At the end of each scan various internal functions are performed. The Modbus communication ports are serviced, programming changes can be made, data can be provided to the CRT display, information can be sent to a computer, and registers can be loaded from external devices. Also, if Memory Protect is on, a portion of the retentive memory is examined with a checksum function to insure that the memory has not changed.

At least twice each scan the PC provides the new status to the output modules. Each output module compares each set of received data and if they are the same, drives its outputs with the data. If the data is different, the outputs are not driven and the processor retransmits the data. The output module then compares the new data with the previous most-recent data. If the data is the same, the outputs are driven and if not, the data is retransmitted up to four times. If after five comparisons there is no valid compare, the outputs retain their previous states and scanning continues. The watch dog timer is not triggered until a valid compare is achieved. If the output module does not receive valid data within 250 milliseconds, its active light goes out and all its outputs are shut off.

- Memory Utilization

Networks of user logic are stored successively in memory. The first network in the program is stored at the beginning of user logic memory and all consecutive networks follow. As networks are added to the program, they are placed after the existing networks. Networks inserted in the program are placed appropriately in memory and the following networks are moved down. This applies for additions to existing networks as well.

Each network stored in user logic memory is stored in columns; the entire first column is stored, followed by the second column, third column etc., Column eleven, although holds the coil symbols and references, is not stored in memory as a column. Each coil is stored in the column in which it is activated. For example, if the logic controlling a certain coil activates the coil in column seven, it is stored in column seven and so on.

4.3.5 984B PLC

Modicon 984 control system makes use of a 984B chassis mount PLC. The 984B PLC is designed for applications that need a high performance PLC with large memory requirements. The controller's user logic memory capacity can be enhanced with upto 96k of extended data memory.

The 984B PLC consists of the following modules:

- P933 Power Supply

  The P933 Power supply provides the required operating voltages for the operating components.

- C921 Communications Processor

  The C921 communications processor supports the RS-232 communications between the supervisory station and the 984B PLC.

- M909 Memory Module

  The M909 memory module contains a 24 bit, 32 kilobyte(kb), CMOS RAM memory board which houses the user logic program. This program controls the overall process. The contents of this CMOS RAM memory board are protected during power failure or power-down for upto one year by long-life lithium batteries.

- C924 Central Processor Unit

  The C924 central processor unit performs the logic processing in accordance with current programming.

- S908 Remote Input/Output (I/O) Processor

  The 984B communicates with a remote I/O, a feature facilitated by the S908 communications protocol. This protocol is built into the mainframe of the PLC. The S908 Remote Input/Output Processor supports upto 32 drops of remote I/O, depending on the upper limit of the 984 PLC to which it is applied. Each drop can support upto two ASCII devices, depending on the remote input/output interface device at the drop. S908 processor uses a single coaxiable connector. The S908 remote I/O processor supports the S908 communications between the 984B PLC, the three PLC I/O racks and the four 3240 Automation Controllers.

- RG-6/U cable

  RG-6/U cable is a recommended type of trunk cable which runs from the PLC to the remote I/O subsystems [20]. Remote I/O communications operate at 1.544 MHz. The RG-6/U coaxial cable is a 5/16 inch flexible cable with moderate noise immunity and signal loss. The shield type is bonded foil quad shield, with a minimum bend radius of 2.0 inch. It has a capacitance of 16.2 pfd/ft and an attenuation of 0.41 dB/100 ft at 1.544 MHz.

- Cable Connectors and Terminators

  The remote I/O system are connected using suitable BNC connectors having an impedance of 75 Ohms. Modicon F-type line tap connectors are used for routing the I/O cables. These taps have a tap loss of 14 dB and a return loss of 18 dB and an impedance of 75 Ohms. Modicon terminators with an impedance of 75 Ohms are used for the trunk and drop cables.

The 984B PLC is an essential part of the system. It performs major functions of: operational sequencing; interlock, permissive, and safety control; inter-system communications; and interface between the supervisory control station and the 3240 automation controllers. The 984B PLC performs the following major tasks:

- Constantly tests the overall system I/O status and maintains this database for normal operation and system startup, restore and shutdown purposes.
- Acts as the communications and data interface between the supervisory control station computer and the 3240s.
- Monitors the 3240s through the S908 remote I/O process module and maintains the status of each controller. The 984B PLC passes the information to any device requesting the status of a particular controller.
- Acts as the communications interface between each of the 3240s and arbitrates requests by the 3240s for operational control of the respective equipments.
- Handles delayed motor tasks which take several seconds to complete. This function increases the speed of the command processor.

4.3.6 PLC I/O racks

There are five PLC I/O racks combined into two groups which are referred to, as drops. Each drop is connected to the PLC and through the S908 communications system. The five I/O racks provide an interface with the equipment and instrumentation. The main I/O rack of each drop is provided with Modicon P884-001 main input power supply modules, Modicon J890 remote I/O processor module, and Modicon input/output modules as required for the application. The I/O rack drop numbering reflects the S908s drop numbering scheme. The first four drops are dedicated to FA-3240 Automation Controllers and numbers (15 , 16) are assigned to the PLC I/O rack groups. These racks contain the following:

- Drop (15) contains three I/O racks which house two B827, 24 VDC input modules, six B804, 16 point, 115 VAC output modules, three B814, dry contact relay output modules, and four AMCI 1844 intelligent absolute resolver interface modules.

- Drop (16) contains two I/O racks which house seven B875, high speed 0-10 VDC analog input modules.

Figure 4.4 represents the 984 controller system with the various configured I/O drops.

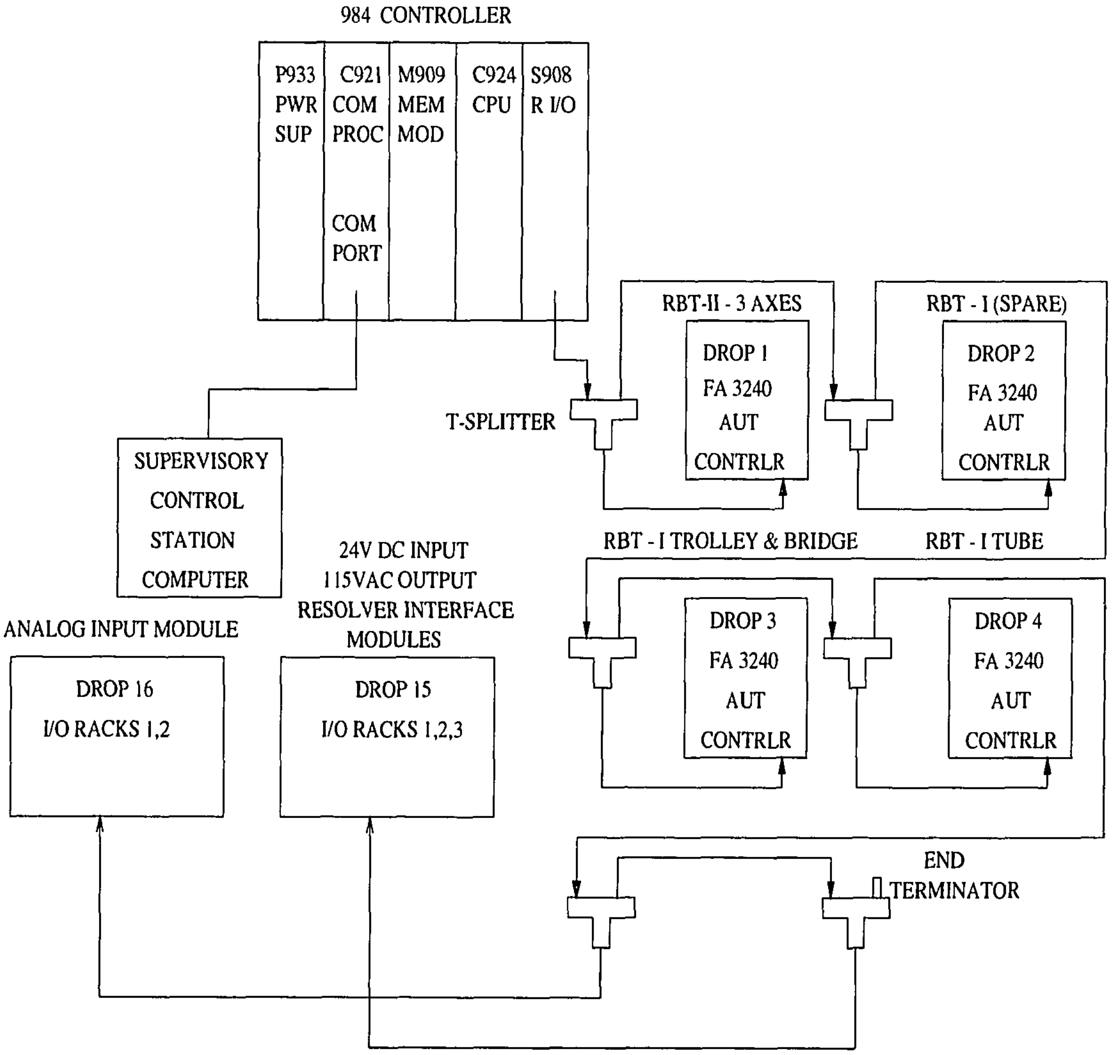


Figure 4.4: Block diagram of 984 controller system with drops configured

B827-24VDC, 32 Point, Input Module

This true high input module consists of a single group of thirty-two independent inputs. The B827 24VDC true high input module senses and converts switched input signals into logic voltage levels used by the programmable controller. Inputs can be received from push buttons, limit and proximity switches, and other 24VDC sources. The module provides thirty-two inputs that share an external supply voltage. Input voltages are sensed by comparing the incoming voltage against a fixed threshold which is function of the user-supplied field voltage.

B804-16 point 115 VAC Output Module

This output module converts logic signals used within the programmable logic controller into 16 independent 115 VAC outputs. Each output is capable of driving a relay, pilot lamp, motor starter, solenoid, or any other load of upto 2.0 amperes. This module is capable of handling a total continuous current of 6 amperes per group of eight points and 12 amperes per module.

B814-108 Relay Output Module

This selectable relay NO/NC output module consists of eight independent outputs. The B814 relay output module converts logic signals used within the programmable controller into eight independent relay outputs. Outputs are capable of driving relays, pilot lamps, and other loads rated at 5 amperes. Eight high reliability relay contacts are used to control these loads. When an output is enabled in the NO output module, the relay contacts conduct current flow from the output A terminal to the output B terminal, whereas, when an output is enabled in the NC module, the relay contacts stop current flow from the output

A terminal to the output B terminal.

1844 AMCI Intelligent Absolute Resolver Interface Module

This is a four axes, single turn encoder module that plugs into directly a Modicon series 800 I/O housing. This module requires a + 5 VDC supply and draws a current of 0.90 Amps (4.50 W total). This encoder performs two operations also called functions namely

- Position function which gives information on the position of the transducer's shaft relative to a zero point. Position resolution is programmable to 1 part in 8192.
- Tachometer function which gives information on the angular velocity of the transducer's shaft in RPM.

Each function is defined by one or more inputs. One input is the input from the resolver while other inputs called parameters, are programmable from the keyboard. The Parameters are

- Scale Factor - sets the resolution to which the position of the transducer's shaft is determined.
- Circular Offset - changes the value of the position function without mechanically rotating the transducer's shaft.
- Linear Offset - adds a fixed number to the value of the position function.
- Tach response - sets the resolution and update time of the built-in tachometer.

B875-101, High Speed Analog Input Module

This module is a field point input device which accepts four or eight analog voltage or current signal inputs from its field circuitry, converting them to numerical data values proportional to the analog input. This module is configured to eight input circuit configurations having the same input signal range of 0-10 VDC, by setting the DIP switch accordingly. The module accepts input current signals through a built-in precision resistor which converts the input current to a voltage drop. The module digitizes the input signals with a 12 bit resolution; i.e., 1 part in 4096. This implies that the least significant bit of the output code corresponds to slightly more than 0.024 of full scale. The resultant digital word for each input circuit is processed and shipped out to the PC for use by the system controller. The module is also capable of presenting its digitized data output to the PC in raw or converted binary format. The exact numerical value is calculated using the following proportional equation, Analog input/FSR = Output count/FCR, where FSR is full scale range for analog input and FCR is full count range in a raw binary format.

4.4 Configuring the PLC

Before the PLC is programmed, it is configured to meet the system's requirements. This configuration describes the way one wishes the controller's memory to be used and specifies the communication parameters and several other hardware settings. Configuring the PLC sets maximum values for discrete inputs and outputs, registers, I/O channels, extended memory and RS-232-C ports. The number of discretes and analogs that may be used in the logic program of the user is limited by the size of state RAM in the controller. Each discrete output (0xxxx) or discrete input (1xxxx) point requires 1 bit of memory space. These are grouped into 16 bit words and hence discretes are allocated in groups of 16 (1 word). Each Input register (3xxxx) or holding register (4xxxx) requires 16 bits or 1 word of space.

The 984B/S908 controller which has a device address of 001 has been configured with 6000 discrete inputs(0xxxx), 2048 discrete outputs (1xxxx), 999 input registers (3xxxx) and 9000 output or holding registers (4xxxx). A total number of 16 segments and 16 I/O drops and I/O modules have been setup. In order to communicate with a computer or other device, the communication ports of the controller must be configured to be compatible with the device. When the database is downloaded to the controller, and the controller started, this configuration is used for communications. The three modbus ports of the controller are configured as RTU (remote terminal mode), a data transmission mode used for modbus communications, with even parity, 1 stop/data bit and a baud rate of 9600. An ASCII port with even parity, 2/8 stop/data bits and a baud rate of 7600 has also been configured. The ASCII port is used for pulling force data transmission to the control system for the pull force software. The extended memory area of the controller has not been configured.

4.4.1 I/O configuration and Traffic copping

Drops 1 through 4 are configured as 900 series type as they are assigned to the four, four axis servo controllers (FA 3240) used in the system as ahown in figure 4.4. The first controller controls the motion of Robotic bridge transporter II. The second and third controllers independently control the X-axis (trolley) and Y-axis (bridge) of Robotic bridge transporter I. As mentioned in Chapter 2, each axis of RBT-I is driven by either one of the two motors provided. Thus the second controller acts as a standby in the event of a failure of the third one which is being normally used. The fourth controller is dedicated to control the telescoping tube assembly (Z-axis) motion of RBT-I. Drop 15 is configured as 800 series type and houses 2 B827 24VDC input modules, 6 B804 16 point 115VAC output modules, 3 B814 dry contact relay output modules and 4 AMCI intelligent absoluteresolver interface modules. Drop 16 is also configured as 800 series type and is made of seven B875 analog input modules in two racks.

The traffic cop directs the flow of data between the various I/O modules and the user logic. It maps the element references used in the logic program and the physical I/O module connection points. The traffic copping also tells the controller how an input signal should be used in user logic, and where an output signal should be sent. The 3240 remote I/O processor appears as a J902 module, binary data type, with a four-axis servo module description in the traffic cop menu. Input registers 30001 - 30008 and holding registers 40001 - 40008 have been traffic copped to controller 1, 30009 - 30016 and 40009 - 40016 to controller 2, 30017 - 30024 and 40017 - 40024 to controller 3 and 30025 - 30032 and 40025 - 40032 to controller 4. Input, output commands and data exchange between 984 PLC and the respective 3240 controller take place in the corresponding set of eight registers only. Here the the first 4xxxx register is called the command register as the read, write or I/O commands are sent from the 984 controller to the 3240 automation controller through this register. The corresponding 3xxxx register is called command echo register as the 3240 controller echoes the command it received from the PLC in this register. The remaining 6 4xxxx and 3xxxx registers are the data registers. The eighth holding register is not used while the eighth input register reveals the status from 3240 controller. The three Opcodes used to handle communication between the 3240 and 984 controller are 1xxx (decimal) for reading from a 3240 register, 2xxx (decimal) to write to a 3240 register and 3xxx (decimal) for I/O data exchange.

4.4.2 Virtual Inputs and Outputs

The 3240 controller can write to the virtual output table and the 984 controller can read from it. Points in the virtual output table are mapped to 3240 logical outputs using the output configuration menu. Similarly, the 984 controller can write to the virtual input table and the 3240 controller can read from it. Points in the virtual input table are mapped to 3240 logical inputs using the input configuration menu. To send and receive the 96 point I/O state table, the first of the designated 984 holding registers is set the opcode 3000 (decimal). The next six registers should contain the I/O information that needs to be sent to 3240. Since each of these registers is a 16 bit register, the most significant bit (msb) of the first data register is assigned the I/O point number 0 and the least significant bit (lsb) is assigned 15 and the same continues till the sixth data register. Thus the lsb of the sixth register will have an I/O point number 95. If a bit is set to "1", the corresponding I/O point in the 3240 controller's state table will be turned on and if it is set to "0", the state of the corresponding point in the table will be either "0" or "1", depending on whether or not one of the control sources is trying to turn the point on. For points in the table controlled exclusively by other sources, the corresponding register bit is set to "0".

4.5 Software

The 984B PLC is programmed with ladder logic, a simple, intuitive, and graphic programming technique. Taylor Industrial Software's Pro**WORX**PLUS, [21] a complete PLC software package, that supports Modicon family of 984/584 processors has been made use of, to program the PLC. Logic programming language uses logic elements like relay contacts, coils, references, registers, shorts and function blocks. As explained earlier in this Chapter, the controller's logic is structured into networks and segments. The controller scans each network and solves the logic which can either control the input to other logic in the program or directly control an output. The main structural elements of a logic program and some basic logic elements used are briefly explained below:

4.5.1 Networks

A network is a set of interconnected logic elements which represents all or part of the logic program. A network is a ladder diagram bounded on the left and right by power rails. By convention only the left rail is shown. Seven rungs or rows, each of eleven columns in width, run from left to right between the two power rails. The 77 regions formed by the intersection of rows and columns are called nodes. Logic elements and instructions can be programmed into these nodes. All 77 nodes may be used to store ladder logic elements and instructions, which are the fundamental building blocks of the logic program. The eleventh column is used exclusively for coils. A network can contain any combination of relay contacts, coils, counters, timers and arithmetic, data transfer and special function blocks.

4.5.2 Segments

The fixed structure of the network results in the logic program overlapping into multiple networks. A group of contiguous networks performing a task or subtask in the application program is called a segment. The number of networks that can be placed in a segment is restricted only by the amount of user memory available and by the maximum amount of PLC scan time (250 ms). For larger ladder logic applications wherein multi drop remote I/Os are used, several segments may be programmed. As a rule in RIO configurations, the number of segments in the program equals the number of I/O drops. However, there could be more segments than drops but never fewer. Segments are numbered 1 ... n, upto maximum of 32, in the order they are created by the programmer.

Each segment of logic generally controls two I/O channels; each channel has a maximum number of 128 inputs and 128 outputs. The number of segments available for the program depends on the number of channels being used, one segment for every two I/O channels. The logic contained in each segment should control the I/O channels it represents. For instance, if logic in segment three controls an output in channel one, the output would not have been activated until the controller starts its next scan and scans segment one, which otherwise would have taken place in the same scan, had it been placed in segment one.

4.5.3 References

Reference numbers are used to identify relay contacts, coils, inputs, outputs, and registers. Each reference has a different code digit, which is the first of five consecutive digits, which identifies the reference type. The following references are in use and explained briefly [22].

Coils or discrete outputs are essentially on/off signals controlled by logic. A coil is usually referred by a five digit code starting with a leading zero (0xxxx). A discrete output can either be used to drive a real output through an output module or can be used internally to drive one or more contacts in user logic.

Discrete Inputs whose status is controlled by an input module, are referred by reference type 1xxxx. They are used to drive contacts in user logic and can be used repeatedly in the logic program, unlike coils, whose usage is restricted to once only.

Input registers referenced by type 3xxxx, hold numerical inputs from an external source, for example, a thumbwheel entry, an analog signal, data from a high speed counter etc., These can also be used to store 16 contiguous discrete signals, which may be entered into the register in either binary or binary coded decimal format.

Output or Holding registers referenced by type 4xxxx, can be used to store numerical information in state RAM or to send the information to an output module.

Extended memory registers referenced by type 6xxxx, store binary information in controller's extended memory area. Extended memory is addressed as 6xxxx registers organised as a group of files, each file containing up to 10,000 registers numbered from 0 to 9999. These registers can be addressed either as a individual or as a group of registers.

4.5.4 Relay contacts

The relay contact is the basic programming element referenced to either a discrete output (0xxxx) or a discrete input (1xxxx). Relay contacts can be normally open, normally closed, or transitional.

Normally Open (NO) contacts pass power when a coil or discrete input is "ON" and do not pass power when the coil is "OFF". **Normally Closed (NC)** contacts do not pass power when a coil is "ON" and pass power when it is "OFF". **Transitional contacts** are turned on by the transition, "OFF" to "ON" (Positive) or "ON" to "OFF", (Negative) of the coil or discrete input to which it is referenced. These contacts are not affected by the state of the logic coil after the transition.

4.5.5 Vertical and Horizontal Shorts

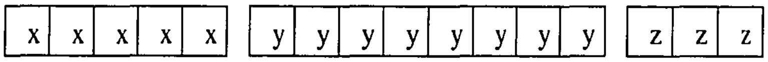
Vertical and horizontal shorts are straight line connections between contacts which allow for expansion and combination of logic. Vertical shorts are used to connect contact and function blocks one above the other in a network. They can also be used to connect inputs or outputs in a function block to create either/or conditions. Horizontal shorts are used in combination with vertical shorts to expand logic within a network without breaking the power flow. They can also be used to create either/or conditions using basic relay contacts.

4.5.6 Coils

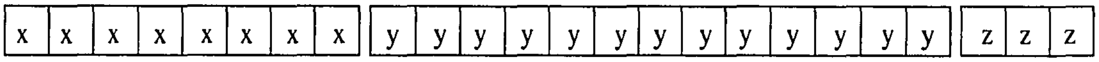
Coils are used to respond to logic and/or to control an output circuit. They are located in the far right of a network and are permitted to be used only once in a program. To continue the logic flow, a contact which is referenced to the coil is used. **Normal coils** are turned off when power is removed and later restored whereas **Latched coils** retain their previous state when power is removed and later restored.

4.6 Translating logic elements in System memory database

The PLC translates symbolic ladder logic elements and function blocks into database nodes that are stored on page 0 in system memory. A node in ladder logic is a 16- or 24- bit word - an element such as a contact translates into one database node, while an instruction such as an ADD block translates into three database nodes. The database format for 16-bit node and 24-bit node are shown in figure 4.5 [23].



16 -BIT NODE FORMAT



24 BIT NODE FORMAT

Figure 4.5: Database format for different nodes

The five most significant bits in a 16-bit node and the eight most significant bits in a 24-bit node - x bits are reserved for opcodes. An opcode defines the type of functional element associated with the node.

When standard ladder logic elements and non-DX function blocks are translated, the system uses the remaining y and z bits as pointers to register or bit locations in state RAM associated with the discretes or registers used in the logic program. With a 16-bit node, 11 bits are available as state RAM pointers resulting in a total addressing capability of 2048 words. The maximum number of configurable registers is 1920 and the balance of 128 words are occupied by discrete reference, disable and history bits. 984B PLC has an extended register option that supports upto 4096 registers in state RAM. In 24-bit machines, 16 bits are available as state RAM pointers and the total number of configurable registers is 9999.

Basic aspects of ladder logic programming software for the PLC have been covered in this chapter. Chapter 5 discusses in detail about Man-Machine-Interface (MMI) and how PLC assists in human intervention.

Chapter 5

Human Machine Interface

This chapter discusses how tele-operation is achieved using supervisory controller equipped with computer graphics based human-machine interface and human role in tele-operation. It also explains how human intervention has been provided by way of joysticks.

5.1 Introduction

In teleoperator theory, man and machine seem analytically irreconcilable; and to make the schism seem more complete, few men doubt that they are superior to machines in many important ways. Yet, man and machine must be integrated , especially in the teleoperator where the partnership is closer than it is in most human-machine systems [24]. Human and machine meet at two hardware interfaces in the teleoperator; the controls and the displays.

Given the attributes of man and machine, one can draw the interface between them by allocating tasks or portions of tasks to each of them. In tele-operator control, the problem of task allocation is rather simple because today's teleoperator normally works with the human operator in full real-time control of all activity. The most important tool in tele-operator control is the supervisory control computer. According to Edwin [24], today's tendency is unquestionably to let the machine portion of the teleoperator do the hard, repetitious work, while the human thinks, plans and explores. The more "intelligent" the machine, the more abstract the controls and the less often man would enter the control loop to operate the controls directly.

5.2 Supervisory Control Station

The primary function of the supervisory control station is to provide an operator interface to allow for the control and monitoring of the setup in both automatic and manual modes of operation.

This control station is networked to the PLC and to all of the 3240 automation controllers which control the two RBTs. This communication network provides the supervisory control station with the capabilities to monitor all system parameters, download programs, receive messages, and respond to messages.

The supervisory control station computer consists of a Pentium 90 MHZ PC with a 731 MB hard drive and a 1.44 MB 3.5″ floppy drive; a 19-inch rack mounted color monitor with a touch screen; all necessary software and communication interface boards; and a keyboard and mouse for operator input.

5.3 Computer graphics based software interface

The supervisory computer is equipped with Intouch, a PC based Man-Machine Interface software through which tele-operation is effected.

5.3.1 Intouch

Intouch [25] is a software package used to create PC based human-machine interfaces. Intouch uses Microsoft Windows Version 3.1 as its operating environment. This package has two major elements namely **WindowMaker**, the development environment where object-oriented graphics are used to create animated touch-sensitive display windows which can be connected to industrial I/O systems and other Microsoft Windows applications and **WindowViewer**, the runtime environment used to display the graphic windows created in WindowMaker.

5.3.2 Tagname listing in Intouch

Tagname data dictionary is the heart of Intouch. Each object created in the WindowMaker menu of Intouch can be associated to a tagname if it needs to be dynamic. Each tagname can be specified a minimum and maximum value, a range within which it is supposed to vary in runtime. These tagnames can be analog or discrete type and can be further classified as Memory or DDE (Dynamic data exchange) type tagnames depending on what purpose they are being used for. Memory type tagnames exist internally within the Intouch application and can be used for creating system constants and simulations. They are also useful in creating calculated variables to be accessed by other Windows programs. Those tagnames that read or write their values from or to another Windows program are classified as DDE type tagnames. At runtime, Intouch contains the current value of all the items in its database. This is the underlying principle of Intouch operation under which it constantly updates those tagnames which are linked to output or input registers of the Programmable controller and other window applications as the case may be.

5.3.3 Communication between Intouch and PLC

Dynamic Data Exchange (DDE) is the communication protocol used by Intouch to communicate to other windows program and to other DDE servers communicating to the **real world**. It implements a client-server relationship between two concurrently running programs. The server application provides the data and accepts requests from other applications interested in its data. The requesting applications are called clients. InTouch can act as both the client and the server. The DDE communication protocol requires three pieces of information namely the application name, the topic name and the item name to be configured and to transfer and receive data. Communication between Intouch and the PLC is established through a RS-232 cable which connects the serial port of the supervisory control station and one of the Modbus ports as shown in figure 5.1.

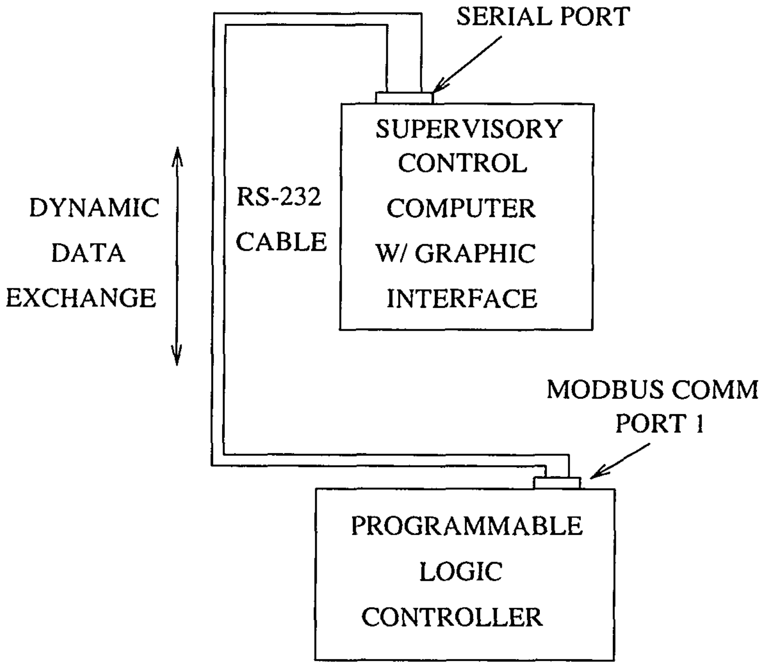


Figure 5.1: Block diagram showing communication between Intouch and PLC

In this set-up, Intouch uses **Modbus** as the DDE server to communicate with

Modicon 984B PLC. When WindowViewer is running, Intouch updates its DDE tagname database as the value in the PLC register to which it is linked, changes.

5.3.4 Using scripts in Intouch

Intouch has the facility to allow for scripts to be written in it which are then executed in runtime to meet specified criteria. These windows can be classified as

- Application scripts which are applicable to the entire application and can be used to create simulations, calculate variables etc.,
- Window scripts which are similar to the application script but apply only to a specific window. Other windows are not affected by this script.
- Key scripts which are linked to a specific key or key combination on the keyboard and can be used for creating global keys for the application.
- Condition scripts which are linked to a discrete tagname or expression and are executed either continuously when the condition is true or false or one time when the condition transitions to true or false depending on how they are applied.
- Data Change scripts which are linked to a tagname and/or a tagname.**field** and are executed once when the value of the tagname or tagname.field changes by a value greater than the dead band defined for the tagname in the tagname dictionary.
- Touch push button action scripts are linked to objects that will be used as a touchlink action pushbutton and are executed whenever the pushbutton is pressed.

5.4 Implementation of Intouch based Teleoperation

The front view and top view of the system are created as object oriented touch sensitive graphics display in the Windowmaker environment as shown in figure 5.2 and 5.3 respectively.

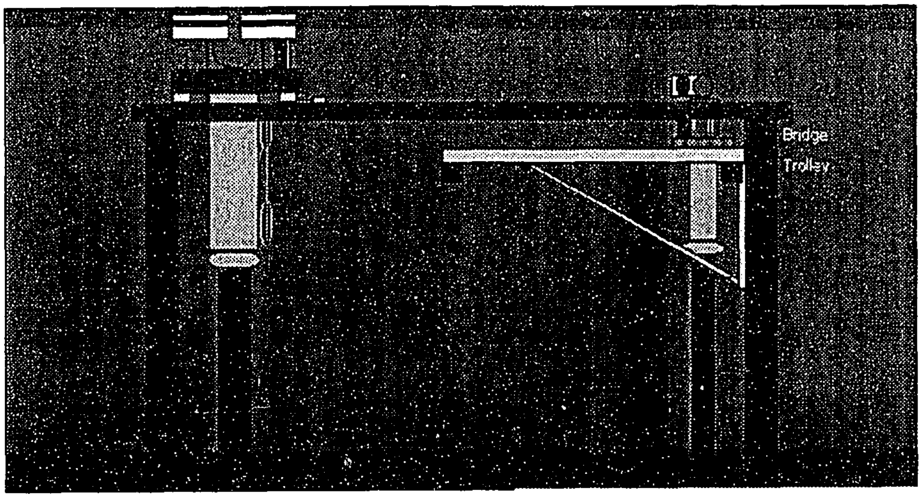


Figure 5.2: Graphic display of Front view of the system

As discussed earlier, these are associated to both Memory and DDE type tag names depending on the need and purpose for which they have been created. The present position of the robot obtained from absolute resolvers in the case of RBT - I is read through an absolute resolver interface module which is configured as an 8 channel input module (B865). The position information is thus available in designated input registers (3xxxx) and is obtained by Intouch through Modbus, its DDE server. In case of RBT -II, position information from the motor resolvers are written into designated C/ROS registers which can then be read into the PLC output registers through a ladder logic program. This information is now available to the graphic based software, Intouch, through its DDE server on a continuous basis. The x, y and z position information thus obtained is displayed on the graphic screen

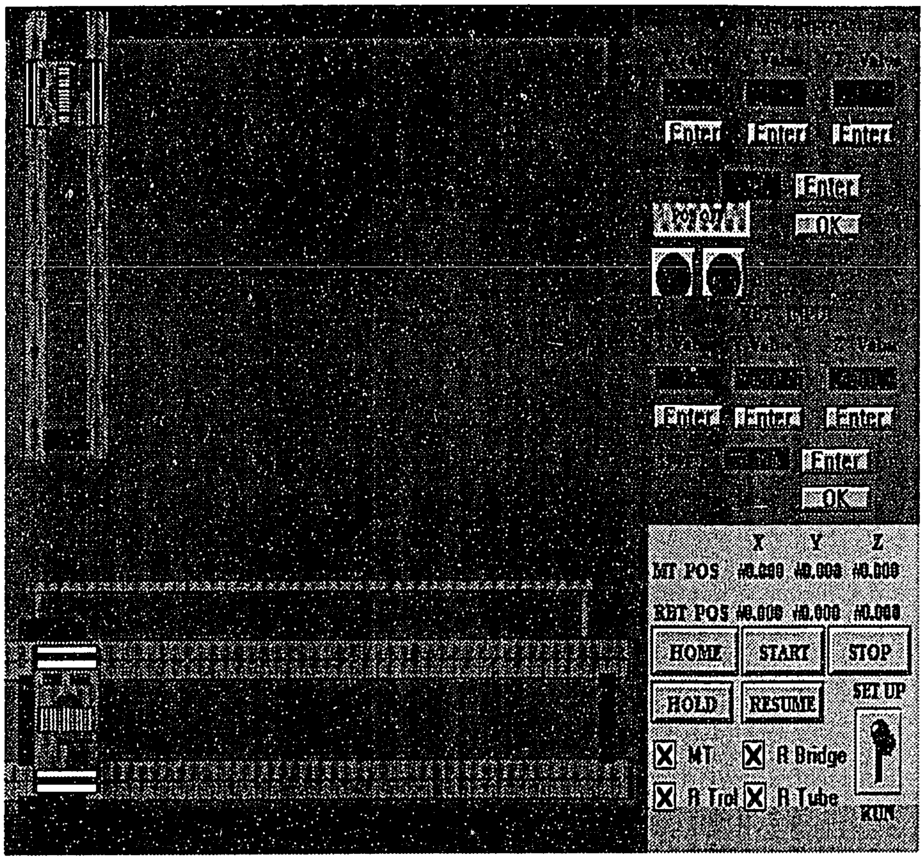


Figure 5.3: Graphic display of Top view of the system

continuously on an absolute scale.

The operator has an input window where he can enter the x, y and z coordinates of the destination point, the robots need to be moved to. The operator also has the provision to enter the speed information at the input window. However, the maximum speed is limited to 150 inches per minute and the maximum point location in the x, y and z direction limited by the available workspace area that the robots can move. Provision has been made to inform the operator if he/she enters a value higher than the maximum permissible through a warning display. The application script in Intouch then converts this information into controller recognizable information. This information is then transferred to the Programmable controller through a logic program in Proworx Plus which the controller scans on a continuous basis. As already mentioned under I/O configuration and Traffic copping in Chapter 4, the controller recognizes these information and then commands the respective automation controller by passing these values to their designated C/ROS registers. The C/ROS module which has the part program loaded into it starts execution once it gets the command input. The movement of the robot to the desired location is monitored by the respective automation controllers which are position feedback controlled.

5.5 Joystick Control

A joystick is a stick- like control that may be tilted forward, backward and sideways; it may also be twisted or pushed in and out along its axis. A joystick consolidates several degrees of freedom into a single piece of hardware. Two important features of a joystick are proportionality (motion/speed control can be made proportional to the joystick displacement) and directionality (motion can be reversed by simply reversing the joystick polarity). A joystick is often a better control device than an array of switches because the operator identifies it better with the task as its construction is usually along anthropomorphic lines.

As already explained in Chapter 2, the supervisory control station has several manual control devices available for use during manual procedures or emergency conditions of which two three-axis joysticks also form a part. These provide independent jog mode motion control for RBT-I and RBT-II. A trigger on the joystick acts as a deadman switch to prevent accidental actuation. A neutral switch in the joystick unit prevents vertical motion while performing transverse motions and vice versa when the joysticks are used with the RBTs. The joysticks provide proportional speed control for servo motors. An "ON-OFF " selector key switch is provided as an interlock for switching onto manual operation using joysticks. The operator must insert his key and turn the key switch in the "ON" position to allow jogging of the RBTs.

Figure 5.4 shows a joystick with the axis direction indicated while figure 5.5 shows the schematic of the electrical connection. A DC supply of +6 to -6 volts is supplied through the joystick to a high speed analog input module (B875-101), the function of which was explained in detail in Chapter 4.

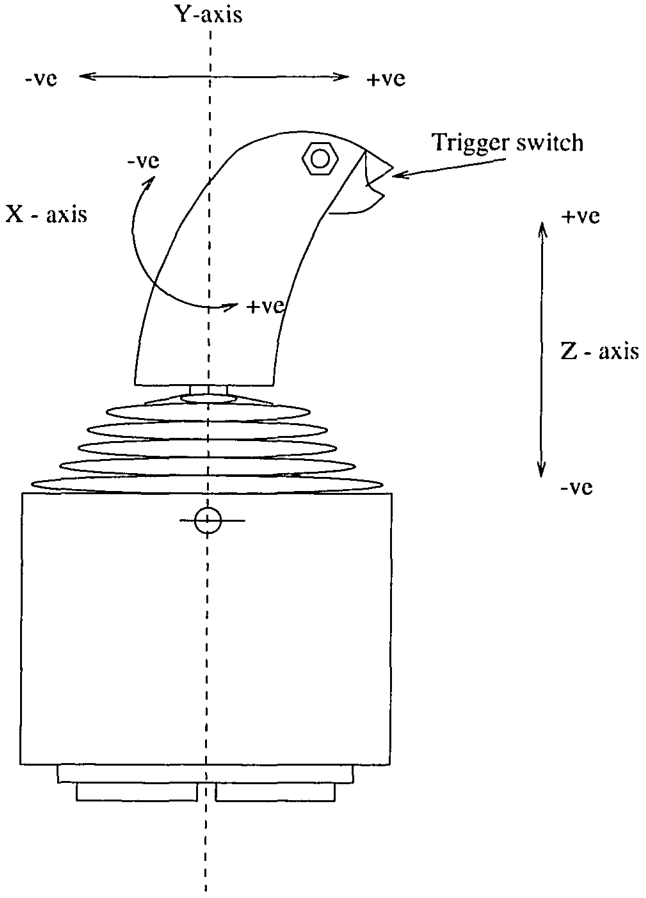


Figure 5.4: A typical joystick with axis direction indicated

The voltage input transmitted by the joystick is proportional to its position from the neutral axis; the larger the displacement, the higher the voltage passed such that the maximum voltage of +6 or -6 volts is transmitted when the joystick is at the farthest

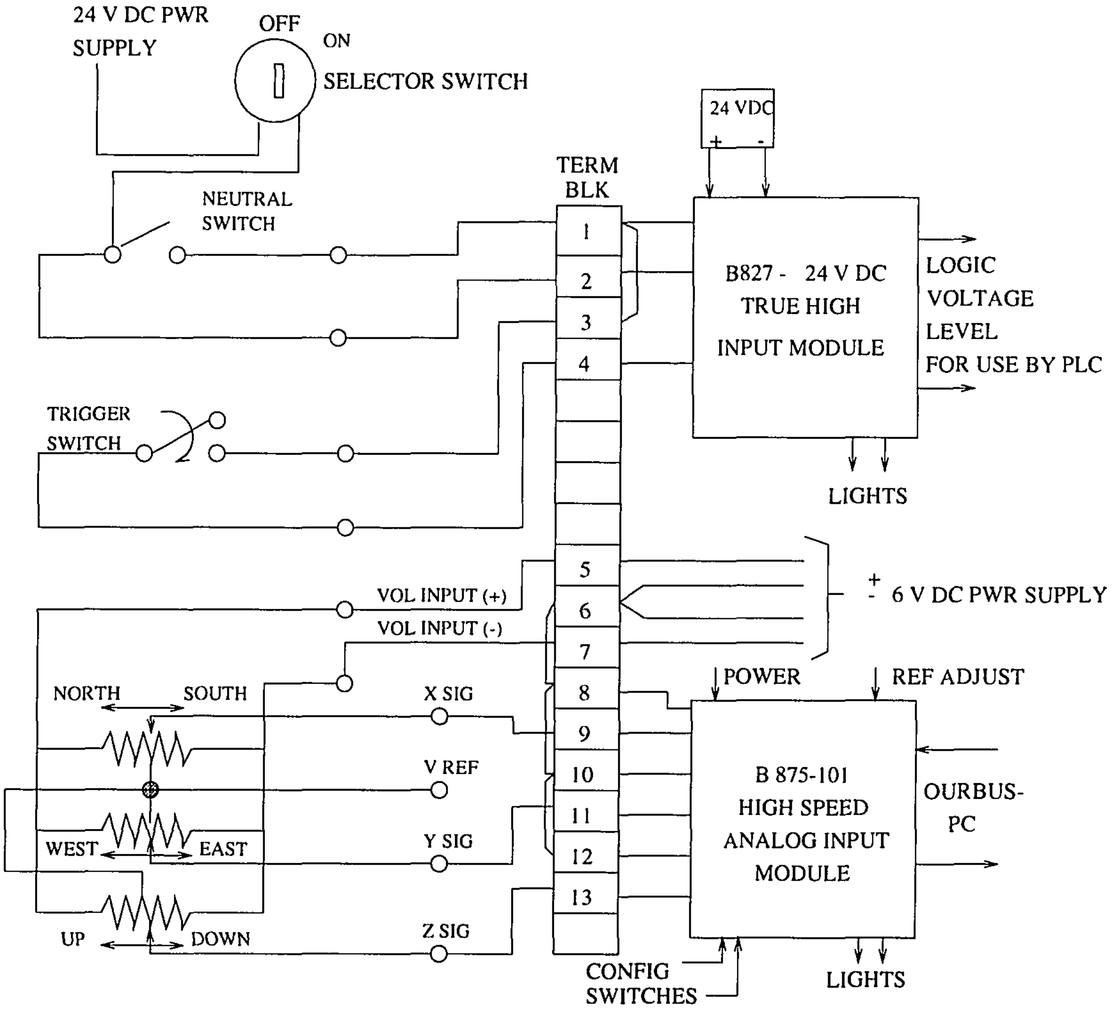


Figure 5.5: Schematic of electrical connection of joystick

position from the neutral axis on either direction respectively. The analog input module digitizes the input signals with a 12 bit resolution proportional to the input voltage signal, with the maximum voltage corresponding to 4096. The numerical value calculated is sent to the Programmable controller which is manifested in the designated input registers. The logic used for joystick control is briefly explained below.

The numerical value corresponding to the neutral position of the joystick was found to have values in the range of 2050-2170 in the X, Y and Z respectively. As the joystick is a sensitive device, these values generally fluctuate in that range. A **SUB** block [23] was used which performs an absolute subtraction of value 1-value 2 held in the top and middle nodes respectively and stores the result in a designated holding register in the bottom node. As the value in the input register corresponding to an axis is greater than 2170, the top output passes power and the speed command is the value contained in the top node (register), for the positive direction. Similarly, when the value in the input register is less than 1950, the difference that is stored in the designated holding register in the bottom node is the speed command to the controller for the negative axis movement.

This chapter has detailed how teleoperation is achieved from the supervisory control computer using a graphic interface and with the presence of human in the control loop which is very essential in teleoperation, especially in hazardous environments. Manual control in case of emergencies or machine malfunctioning, through joystick, has also been explained at length. Chapter 6 discusses the importance of collision avoidance, its complexities in teleoperation and an implementable algorithm for collision avoidance of gantry robots and concludes this research.

Chapter 6

Collision-Free Motion planning

This chapter discusses the collision problem of the two Robotic bridge transporters sharing a common workspace and proposes a practically implementable collision avoidance scheme. The collision-free algorithm is discussed at length and the results of the same when tried for two cases are provided. A brief note on the implementation methodology has also been made towards the end of the chapter.

6.1 Introduction

A coordinated multiple robot system is one in which two or more robots share a common workspace. If the robots only share a common workspace but execute independent tasks, such coordinations, termed loose coordinations as discussed in Sheu [4], allow for more freedom in motion planning. However, when two robots share a common workspace and co-operate on executing a common task, the problem becomes more complex and imposes more stringent kinematic and dynamic constraints which need to be addressed when solving the motion planning problem. The collision problem involving the two bridge type transporters is discussed before presenting an implementable solution.

6.2 The Collision problem

Multiple robots are used on a cooperating task to handle complex tasks in an efficient manner. When such robots operate in a common workspace they become moving obstacles to each other. The two robotic bridge transporters whose configuration were discussed in detail in Chapter 2, share a common workspace. This collision problem applies only to the bridge transporters and do not extend to their end effectors. There are four possible cases of collision that could occur while the robots move in their workspace as listed below.

1. Collision between the left end of the trolley frame of RBT-II and telescoping tube of RBT-I.
2. Collision between the right end of the trolley frame of RBT-II and telescoping tube of RBT-I.
3. Collision between the front end of the trolley frame of RBT-II and telescoping tube of RBT-I.
4. Collision between the front end of RBT-II trolley motor and extended frame of RBT-I bridge assembly.

In case 1, the left end of the trolley frame of RBT-II may collide with the telescoping tube of RBT-I as shown in the top view of the two robots in figure 6.1. Case 2 is similar to case 1 but the other end of the trolley frame can come into collision with the tube assembly of RBT-I. This case is shown in figure 6.2.

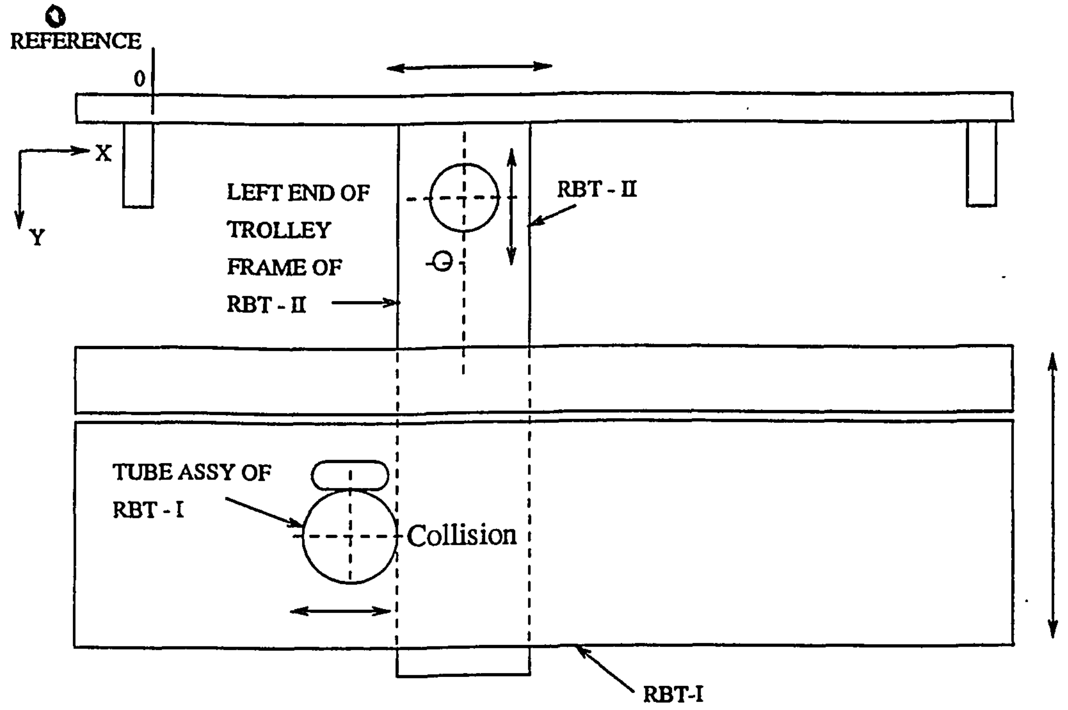


Figure 6.1: Case 1: Top view of collision between left end of RBT-II trolley frame and telescoping tube assembly of RBT-I

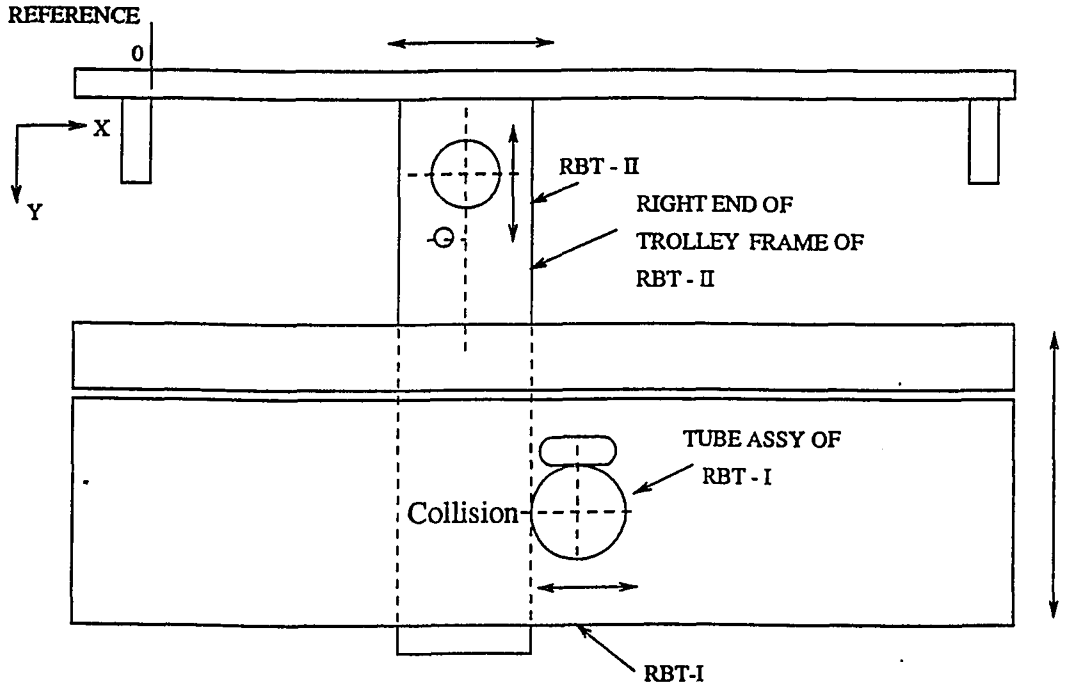


Figure 6.2: Case 2: Top view of collision between right end of RBT-II trolley frame and telescoping tube assembly of RBT-I

In case 3, there is a possible collision between the front end of the trolley frame of RBT-II and the tube assembly of RBT-I as shown in figure 6.3. The trolley frame can squeeze through the cable assembly located in front of the tube assembly of RBT-I to collide with the tube assembly. Hence for computational ease, the cable is considered as part of the tube assembly and the diameter of the tube is increased accordingly. Figure 6.4 shows the last possible case of collision wherein, the extended frame of bridge of RBT-I may collide with the trolley motor assembly of RBT-II. Here again, for computational ease, the motor

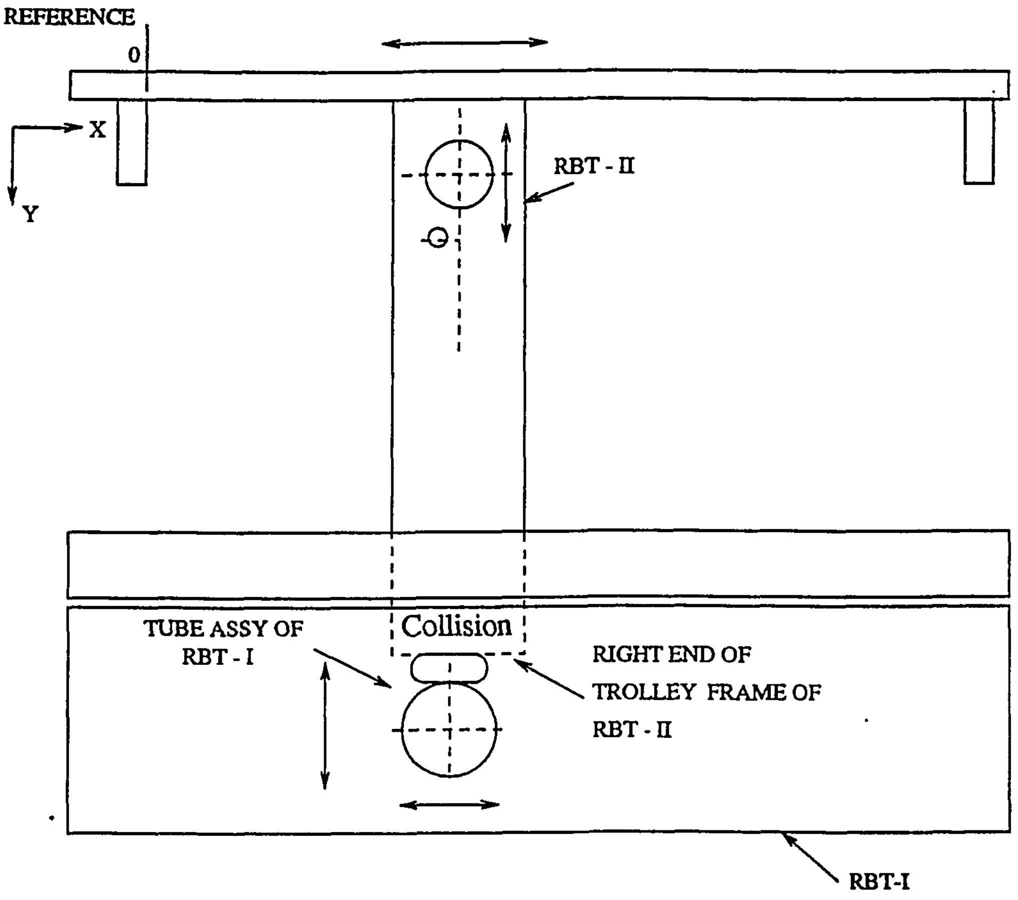


Figure 6.3: Case 3: Top view of collision between front end of RBT-II trolley frame and telescoping tube assembly of RBT-I

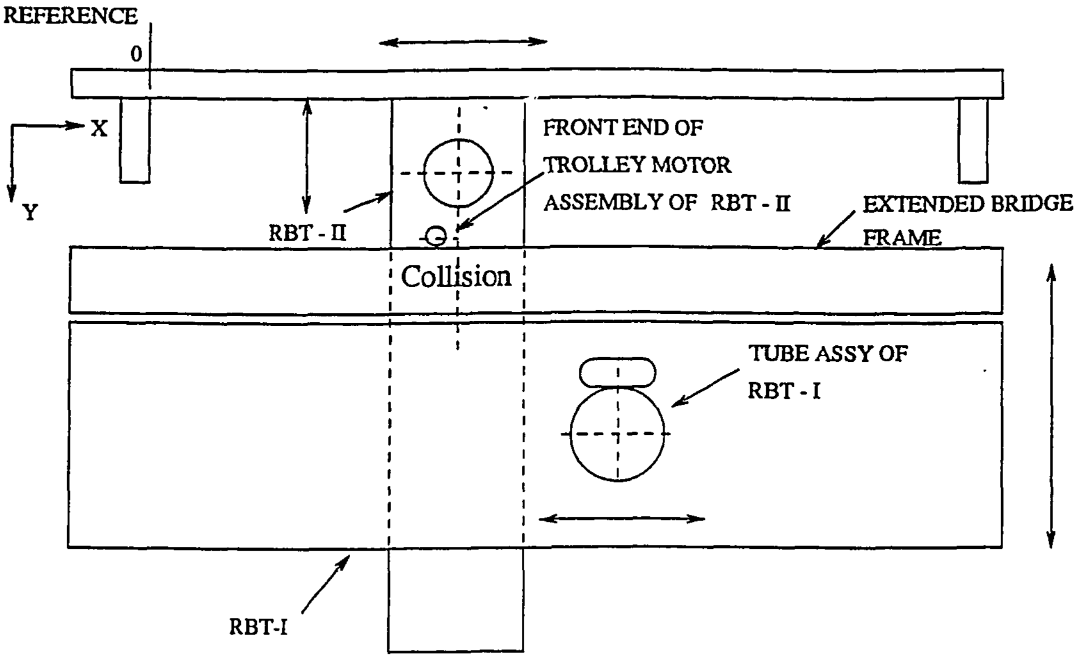


Figure 6.4: Case 4: Top view of collision between front end of RBT-II trolley motor and extended frame of RBT-I bridge assembly

assembly is considered as part of the telescoping tube assembly of RBT-II as it is mounted to a bracket on the trolley frame and moves along with the tube as an integral part. The diameter of the tube assembly for RBT-II is increased accordingly to accomodate this case.

6.3 Solution Methodology

6.3.1 Basis for solution

As mentioned in Chapter 1, the solution proposed in this study has its roots on the technique proposed by Jihong Lee [12]. The objective of this method which introduces a delay time between robot path segments, is to minimize the delay time involved to avoid collision. This technique has been suitably modified and extended to apply to the two gantry type robots.

6.3.2 Assumptions

The following are some of the assumptions and conditions considered before proposing the solution.

1. This method applies only to the bridge transporters for a given path and velocity and does not extend to their end effectors.
2. The paths of the two robots are composed of one or more segments both geometrically and in time. The robots move to a task point, stop to do something by end-effector, move to the next task point and repeat such sequence till reaching the destination point.
3. Trapezoidal Velocity profile is considered as it is controller dependent, although this method can in general be applied to any velocity profile.
4. Although this method can apply for any type of path segment, this study discusses only straightline path segments.
5. The robots can stop only between path segments and will not stop while executing a segment.

6.3.3 Defining Coordination space and Collision region

As discussed in [13], let s^1 and s^2 denote the travelling lengths along the paths of the two robots, RBT-I and RBT-II respectively and let s^1_{max} and s^2_{max} denote the entire length of the paths of the two robots respectively. Then, coordination space is defined as a collection of ordered pairs (s^1, s^2) such that $0 \leq s^1 \leq s^1_{max}$ and $0 \leq s^1 \leq s^2_{max}$.

Any continuous curve in Coordination space which starts from point (0,0) and reaches final point (s^1_{max}, s^2_{max}) is termed as the coordination curve. A collection of all points in Coordination space (CS) corresponding to the two robots in collision is called Collision region (CR). The coordination curve which does not pass through the collision region is called collision-free coordination curve (CFCC). Figure 6.5 illustrates the above definitions better.

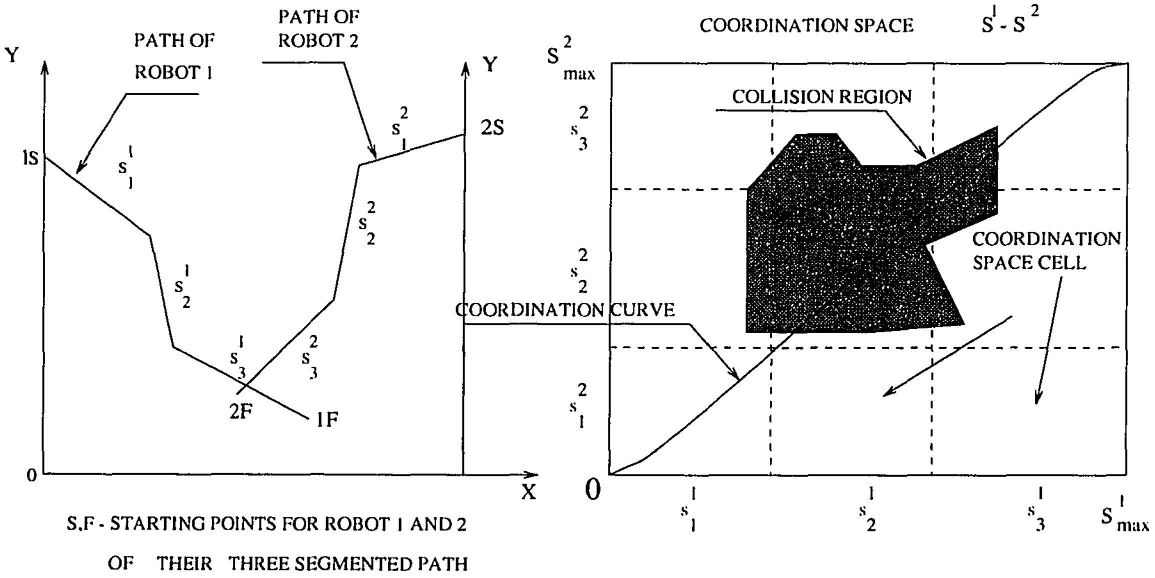


Figure 6.5: Concept of Coordination Space and Coordination curve

6.3.4 Collision criteria for two Gantry Robots

The telescoping tube of both the RBTs can be approximated to circles of specified diameter. The ends of the trolley frame of RBT-II as well as the extended frame of the bridge of RBT-I can be treated as straight lines with known end points. Thus the four possibilities of collision can be treated as collision between circle and straight line both moving along specified straight line path segments.

The following general parametric equations apply for cases 1, 2 and 3. These cases represent collision between the trolley frame of RBT-II and the telescoping tube of RBT-I. Parametric equation for the circle center moving along a straight line path is given by

$$\begin{aligned} x_b &= x_{mo} + u\,(x_{mf} - x_{mo}) \\ & \qquad\qquad 0 \leq u \leq 1 \\ y_b &= y_{mo} + u\,(y_{mf} - y_{mo}) \end{aligned} \tag{6.1}$$

where x_b and y_b represent the coordinates of the circle center, (x_{mo}, y_{mo}) and (x_{mf}, y_{mf}) represent the coordinates of the starting point and end points of the straight line path segment respectively and 'u' is the parameter. Parametric equation for a point on the circle circumference can hence be written as,

$$\begin{aligned} x &= x_b + r_b\,(cos(2\,\pi\,t)) \\ & \qquad\qquad 0 \leq t \leq 1 \\ y &= y_b + r_b\,(sin(2\,\pi\,t)) \end{aligned} \tag{6.2}$$

which yields

$$
\begin{aligned}
x &= x_{mo} + u\,(x_{mf} - x_{mo}) + r_b\,(cos(2\,\pi\,t)) &\quad (6.3)\\
y &= y_{mo} + u\,(y_{mf} - y_{mo}) + r_b\,(sin(2\,\pi\,t)) &\quad (6.4)
\end{aligned}
$$

Similarly, parametric equation for the straight line is given by,

$$
\begin{aligned}
x &= x_{ro} + s\,(\Delta x_r) \\
& \qquad\qquad 0 \leq s \leq 1 \\
y &= y_{ro} + s\,(\Delta y_r) \qquad (6.5)
\end{aligned}
$$

where (x_{ro}, y_{ro}) and (x_{rf}, y_{rf}) are the coordinates of the starting and end points of the line respectively and $\Delta x_r = x_{rf} - x_{ro}$ and $\Delta y_r = y_{rf} - y_{ro}$. Since this line itself moves along a straight line path, which is effectively the path of the robot, we have,

$$
\begin{aligned}
x_{ro} &= x_{po} + v\,(x_{pf} - x_{po}) \\
& \qquad\qquad 0 \leq v \leq 1 \\
y_{ro} &= y_{po} + v\,(y_{pf} - y_{po}) \qquad (6.6)
\end{aligned}
$$

where (x_{po}, y_{po}) and (x_{pf}, y_{pf}) represent the starting and end points of the straight line path segment respectively. Substituting, eqn (6.6) in (6.5) we have,

$$
\begin{aligned}
x &= x_{po} + v\,(x_{pf} - x_{po}) + s\,(\Delta x_r) &\quad (6.7)\\
y &= y_{po} + v\,(y_{pf} - y_{po}) + s\,(\Delta y_r) &\quad (6.8)
\end{aligned}
$$

as the parametric equation for the straight line moving along a straight line. Combining equations (6.3) and (6.4) and (6.7) and (6.8), we have,

$$x_{mo} + u\,(x_{mf} - x_{mo}) + r_b\,(cos(2\,\pi\,t)) \;=\; x_{po} + v\,(x_{pf} - x_{po}) + s\,(\Delta x_r) \qquad (6.9)$$

$$y_{mo} + u\,(y_{mf} - y_{mo}) + r_b\,(sin(2\,\pi\,t)) \;=\; y_{po} + v\,(y_{pf} - y_{po}) + s\,(\Delta y_r) \qquad (6.10)$$

Rearranging squaring and adding to eliminate "t", the above equations can be expressed as

$$\begin{aligned}(\Delta x_r^2 + \Delta y_r^2)\,s^2 + (-2\Delta x_r \Delta x_m\;u + 2\Delta xp\Delta x_r\;v + 2\Delta x_o\Delta x_r + 2\Delta yp\Delta y_r\;v\\ +2\Delta y_o\Delta y_r - 2\Delta y_r\Delta ym\;u)\,s + (\Delta x_o^2 + 2\Delta x_o\Delta xp\;v - 2\Delta xp\Delta xm\;u\;v - 2\Delta x_o\Delta xm\;u\\ \Delta xp^2\;v^2 + \Delta xm^2\;u^2 + \Delta y_o^2 + 2\Delta y_o\Delta yp\;v - 2\Delta yp\Delta ym\;u\;v - 2\Delta y_o\Delta ym\;u\\ +\Delta y_m^2\;u^2 - r_b^2) =\\ 0 \quad (6.11)\end{aligned}$$

where

$\Delta xp = (x_{pf} - x_{po})$, $\quad \Delta yp = (y_{pf} - y_{po})$,

$\Delta xm = (x_{mf} - x_{mo})$, $\quad \Delta ym = (y_{mf} - y_{mo})$

for computational ease.

Equation 6.11 is a quadratic equation in 's' and hence has two solutions for 's' namely

$$s = \frac{-B \pm \sqrt{(B^2 - 4AC)}}{2A} \qquad (6.12)$$

where

$$\begin{aligned} A &= \Delta y_r^2 + \Delta x_r^2 \\ B &= (-2\Delta x_r \Delta xm - 2\Delta y_r \Delta ym)\, u + (2\Delta yp \Delta y_r + 2\Delta xp \Delta x_r)\, v \\ &\quad + 2\Delta x_o \Delta x_r + 2\Delta y_o \Delta y_r \end{aligned}$$

and

$$\begin{aligned} C &= (\Delta xm^2 + \Delta ym^2)\, u^2 + ((-2\Delta xp \Delta xm - 2\Delta yp \Delta ym)\, v \\ &\quad - 2\Delta y_o \Delta ym - 2\Delta x_o \Delta x_m))\, u + (\Delta yp^2 + \Delta xp^2)\, v^2 \\ &\quad + (2\Delta xp \Delta x_o + 2\Delta yp \Delta y_o)\, v + \Delta x_o^2 + \Delta y_o^2 - r_b^2 \end{aligned}$$

respectively.

Equation (6.12) is solved for values of "u" and "v" between 0 and 1 and the value of s checked. If "s" lies between 0 and 1 there is collision and not otherwise.

The same procedure applies to collision case 4 also except that in the latter, the telescoping tube of RBT - II is the circle colliding with the extended frame assembly of bridge of RBT-I, a straight line, both moving along straight line paths.

6.4 Defining collision region boundary

As stated earlier, the collision region is a collection of all points in coordination space (CS) corresponding to the two robots in collision. However, it is desirable to define the collision region boundary in order to propose a solution of minimum time delay. This section discusses how the boundary of the collision region can be defined in coordination space.

Figure 6.6 shows the intersection that the line segment can encounter with the circle while moving along a straight line path.

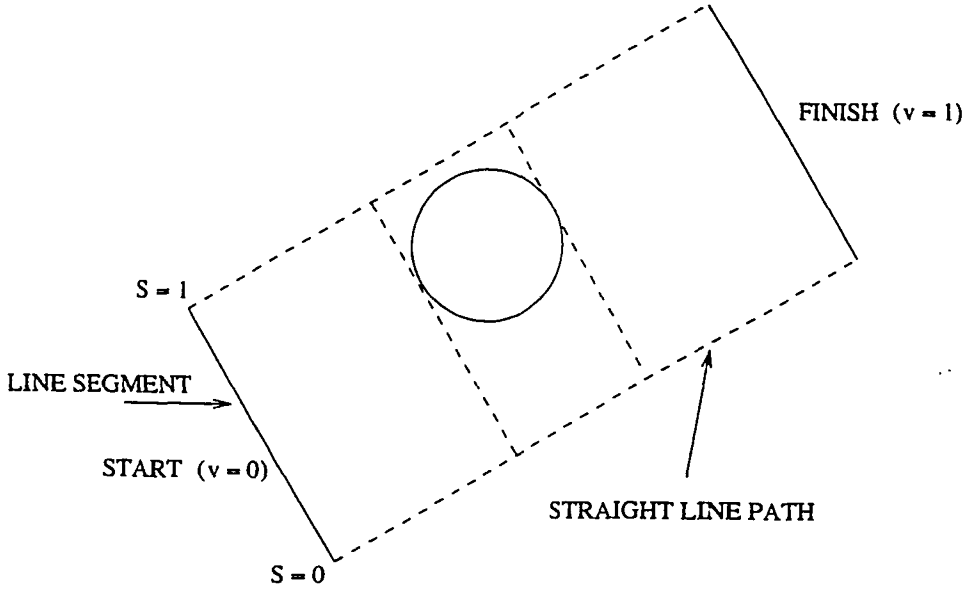
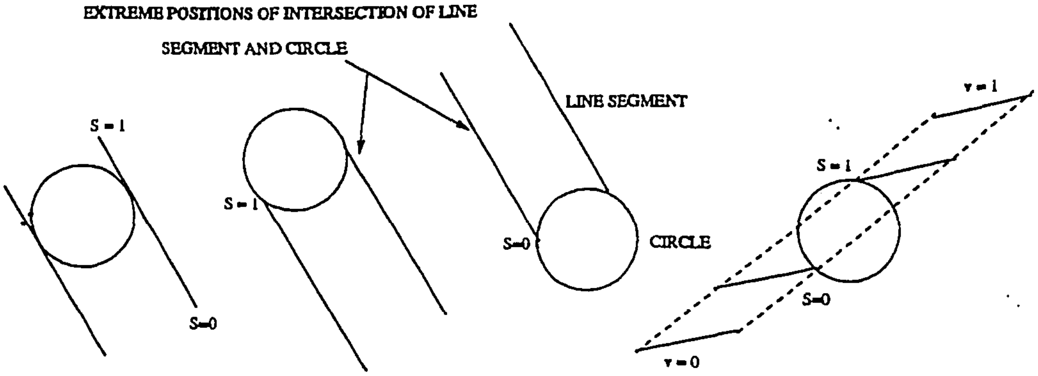


Figure 6.6: Intersection between the line segment moving along a straight line path and the circle

The four possible types of intersection that the circle and line segment can have are shown in figure 6.7.



CASE A: TANGENCY CONDITION

CASE B : NO TANGENCY LINE BELOW CIRCLE

CASE C: NO TANGENCY LINE ABOVE CIRCLE

CASE D: NO TANGENCY S = 0 (&/OR) S = 1 OCCUR WHEN $0 < v < 1$.

Figure 6.7: Possible cases of intersection between circle and line segment

The procedure of defining the collision boundary can be easily explained by means a simple flow chart as shown in figure 6.8.

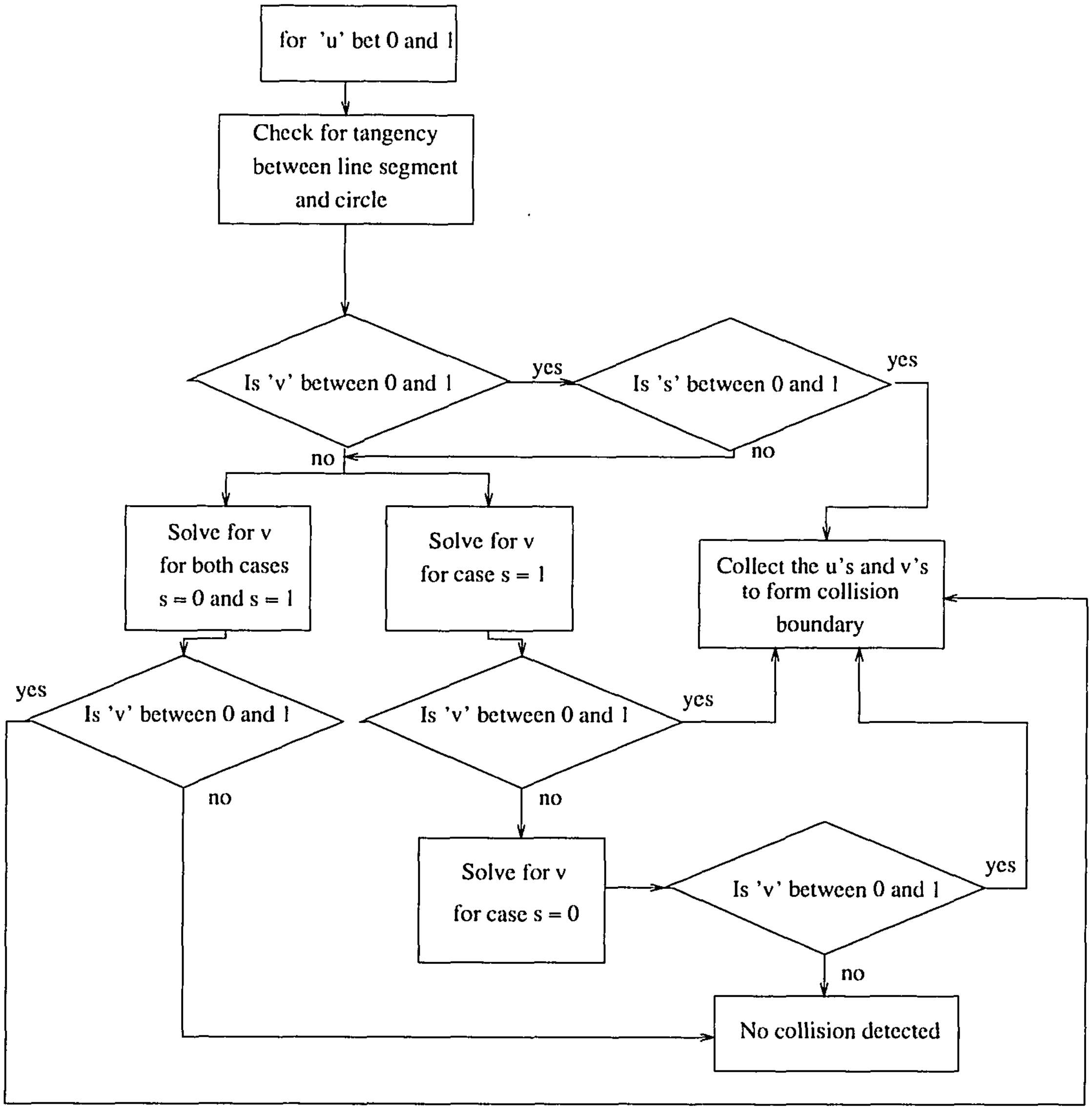


Figure 6.8: Flow chart to determine the collision boundary

As the first step towards determining collision boundary for each of the above four cases, the line segment is checked for tangency case (a) with the circle for various positions along its path. In equation (6.12), (B^2 - 4AC) = 0 is the tangency condition. If the tangency condition fails, there could be two possible types of intersection. In case (b) as shown in figure 6.7, where the line segment lies below the circle, the collision boundary can be formed by joining all points of intersection of the top edge of the line segment with the circle (s=1). Similarly, in case (c) the boundary can be formed by joining those points of intersection of the bottom edge of the line segment with the circle (s=0). Case (d) is similar to cases (b) and (c) and occurs for collision case(3). For case (b) and (d), when s= 1, from equation (6.12),

$$\begin{aligned} -B \pm \sqrt{(B^2 - 4AC)} &= 2A \\ \text{i.e} \qquad B^2 - 4AC &= (2A + B)^2 \\ \text{i.e} \qquad A + B + C &= 0 \end{aligned} \tag{6.13}$$

where A, B and C are the same as explained earlier. Similarly for case (c) and (d), when s=0, we have,

$$\begin{aligned} -B \pm \sqrt{(B^2 - 4AC)} &= 0 \\ \text{i.e} \qquad B^2 - 4AC &= B^2 \\ \text{i.e} \qquad AC &= 0 \\ \text{i.e} \qquad C &= 0 \end{aligned} \tag{6.14}$$

as A being a constant, A= 0 results in a trivial solution. If all the four cases fail to provide a solution, then no collision has been detected.

6.4.1 Constructing the Coordination curve

Since the velocity of the two robots are known, the coordination curve in the coordination space can be constructed by collecting the point pairs representing the run-lengths of the two robots corresponding to the same time. As mentioned earlier, the robots can stop only at points between segments and do not stop otherwise. The collision regions determined for the various cases for a given path and the coordination curve for the two robots are then represented on the same graph which would indicate the possibility of collision for the given path. The coordination curve resulting in an intersection with any of the collision boundaries , represents collision.

6.4.2 Collision mapping in t-s space

The proposed collision avoidance scheme employs a delay time technique in which one robot waits before running next path segment till the other robot completely clears the collision boundary. Hence it is desirable to project the collision information and path in space composed of time of the first robot and space of the second robot (or from $s^1 \times s^2$ to $t^1 \times s^2$ space) and plan the motion of the second robot without violating the velocity constraints such that it does not enter the collision region. Since the velocities of both the robots and the time of constant acceleration and deceleration are known, the time at which the robots can cover a given distance can be computed for the different velocity regions of their trapezoidal velocity profile as below. For the constant acceleration region,

$$tm = \sqrt{\frac{2\,xm\,t1_n}{V_n}} \tag{6.15}$$

where xm is the distance travelled. Appendix B may be referred for information on trapezoidal velocity profile and computation of distance in each velocity region. For the constant velocity region the time required to traverse a given distance is given by,

$$tm = \frac{xm - V_n\, t1_n/2}{V_n} + t1_n \tag{6.16}$$

and for the constant deceleration region, it is given by,

$$tm = -\frac{(t3_n - t2_n)}{V_n}\left[-\frac{V_n\, t3_n}{(t3_n - t2_n)} + \frac{\sqrt{-V_n(V_n\, t2_n - 2xm - V_n t1_n + V_n t3_n)}}{\sqrt{-t3_n + t2_n}}\right] \tag{6.17}$$

Therefore, each of the collision region boundary is independently checked for the velocity region of the first robot it lies in. Then the time in which the first robot would have traversed the same at its velocity, computed. Thus the whole collision region is transformed in $t^1 \times s^2$ space before suggesting which robot to delay. The trajectory of the second robot is computed in the first robot's time and depicted in the same figure along with the transformed collision region. Figure 6.9 shows an example of coordination curve and collision region in $s^1 - s^2$ space and figure 6.10 shows the same collision region transformed in the $t^1 - s^2$ space along with the trajectory of the second robot computed in the time of the first robot. The trajectory of the second robot resulting in intersection with any of the collision boundaries represents collision.

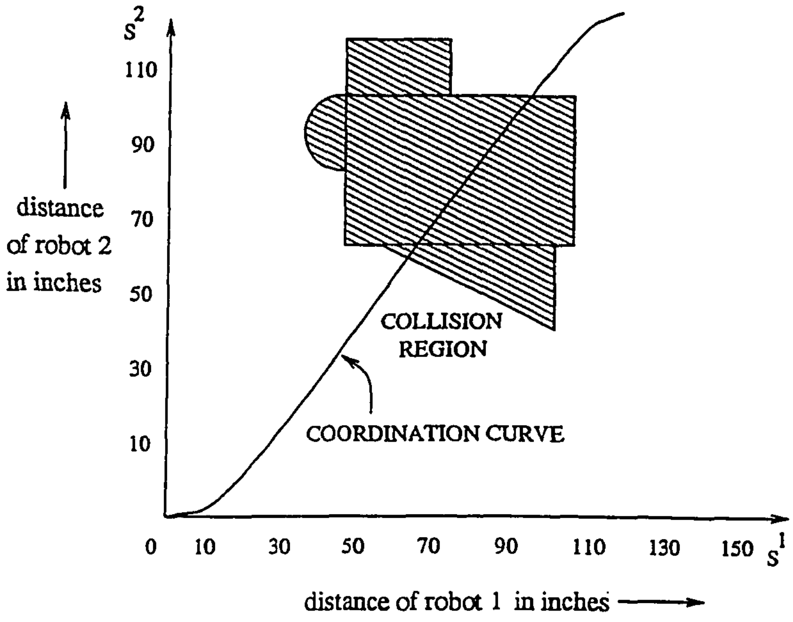


Figure 6.9: Collision region and coordination curve in $s^1 - s^2$ space

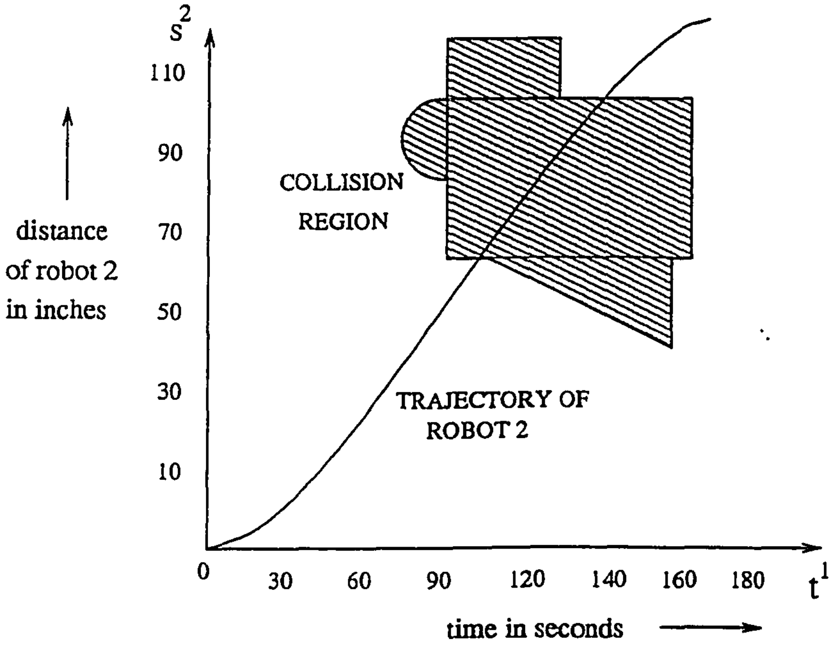


Figure 6.10: Transformed collision region in $t^1 - s^2$ space and trajectory of robot 2

6.5 Collision avoidance

Having transformed the collision region in t-s space, the means of avoiding collision can now be discussed. As can be seen from figure 6.10, there could be two possibilities of deviating the robot trajectory such that it clears the collision region. In general, any collision region can be approximated to a polygon. The robot path is planned in such a way that it either passes through one of the vertices of the polygon or tangent to its sides, if the side happens to be an arc such that it clears the collision region. The resulting delay or advance as the case may be, will hence be the minimum required time for the robot to have a collision free motion. Figure 6.11 shows the two possibilities of either delaying or advancing robot 2 such that it clears the collision region.

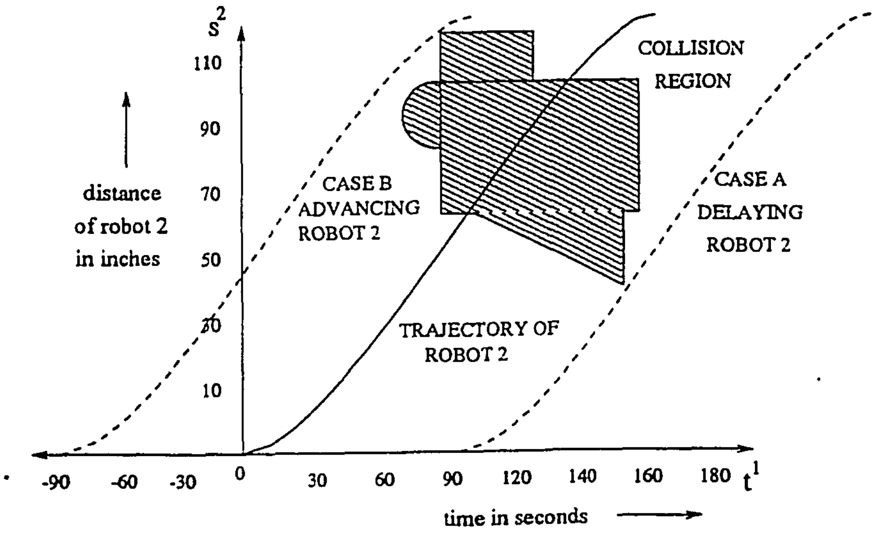
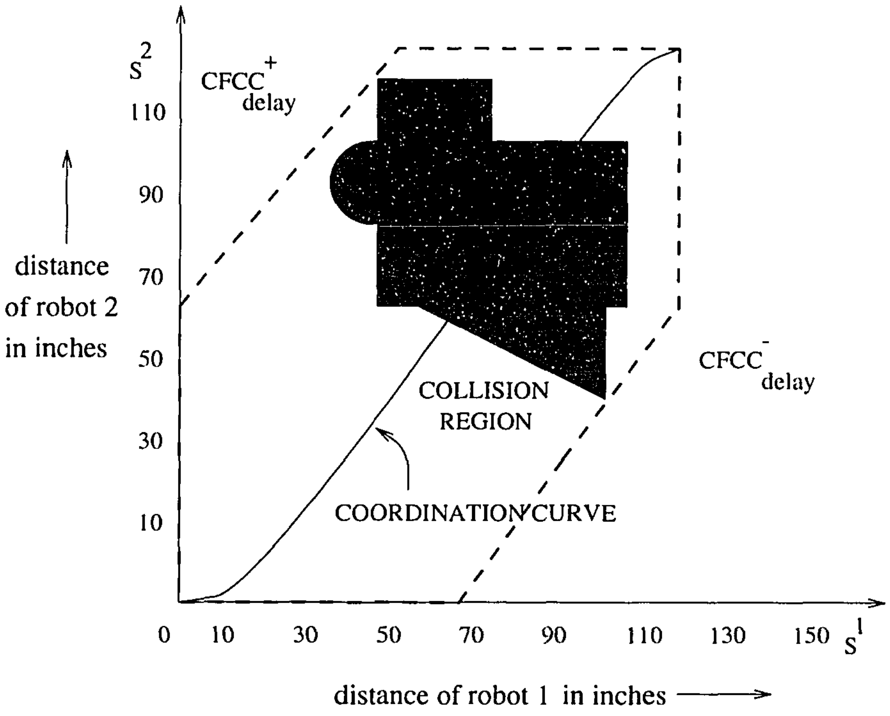


Figure 6.11: Possibilities of delaying or advancing robot 2

As can be seen from the figure, in case (a) when robot 2 is delayed the trajectory shifts below the collision region and in case (b) the trajectory shifts above the collision region when it is advanced. Also, it may be noted that advancing robot 2 is the same as delaying robot 1. The minimum of the two delay times is considered as the best solution for



TWO TYPES OF COLLISION FREE COORDINATION CURVE

Figure 6.12: Two types of collision free coordination curve

collision avoidance. The solution will then shift the coordination curve out of the collision region. This resultant coordination curve is called **Collision free coordination curve - CFCC** as shown in figure 6.12.

When the coordination curve shifts to the right of the collision region it is called $CFCC^{-}_{delay}$ and when it shifts to the left it is it is called $CFCC^{+}_{delay}$.

However, more often it may not be possible that both solutions are feasible. As a special case for instance, when the collision boundary has the X-axis as one of its sides, it will not be possible to delay robot 2 and the solution reduces to advancing it and determining the minimum time of advancement. On the otherhand, if the collision region has the Y-axis as one of its sides, then the solution reduces to delaying robot 2. These cases are shown in figure 6.13.

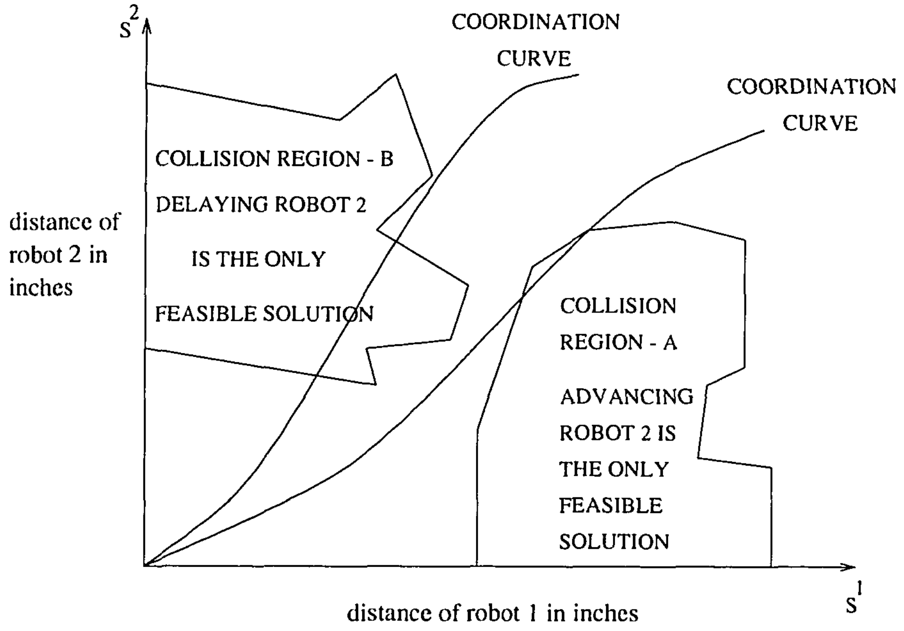


Figure 6.13: Special cases of Collision region

6.6 Summary of algorithm

The algorithm presented can be summarised into the following steps.

1. Firstly, the coordination curve is constructed and the collision region marked in $s^1 - s^2$ space.
2. If the coordination curve passes through the collision region, the region is transformed in $t^1 - s^2$ space and depicted along with the trajectory of robot 2.
3. The collision region is then checked for special cases as described above which will directly provide the solution of whether to delay or advance the robot.
4. The vertices of the collision boundary polygon are investigated in turn. These points have time and distance as their x and y coordinates respectively.

5. For the trajectory of the second robot to pass through a given vertix, the additional time needed to cover the required distance is computed. This time "t_y", is called the delay time.
6. A positive value of t_y indicates advancing robot 2 which effectively is delaying robot 1 and a negative value indicates delaying robot 2.
7. The new path of robot 2 is now computed incorporating this delay time which is the minimum time needed to avoid collision.
8. If the collision region does not fall under the special categories, then both solutions are possible and the minimum of the two is taken as the desired solution. Thus the objective of minimizing the total traverse time for the two robots for a given path segment is achieved whilst avoiding collision.

6.7 Application example

The algorithm was tested for two cases for the two bridge transporters RBT - I and RBT-II. In the first example, the robot paths were made up of one segment each whereas in the second case, the proposed path of the two robots were made up of two segments each. For example 1, both the robots were made to travel at the same velocity of 0.5 inches per second. The total travel time for RBT-I was 258 seconds and that of RBT-II was 294 seconds. The proposed path of the two robots are as shown in figure 6.14.

The individual collision regions corresponding to the four cases are shown in figure 6.15 - 6.18.

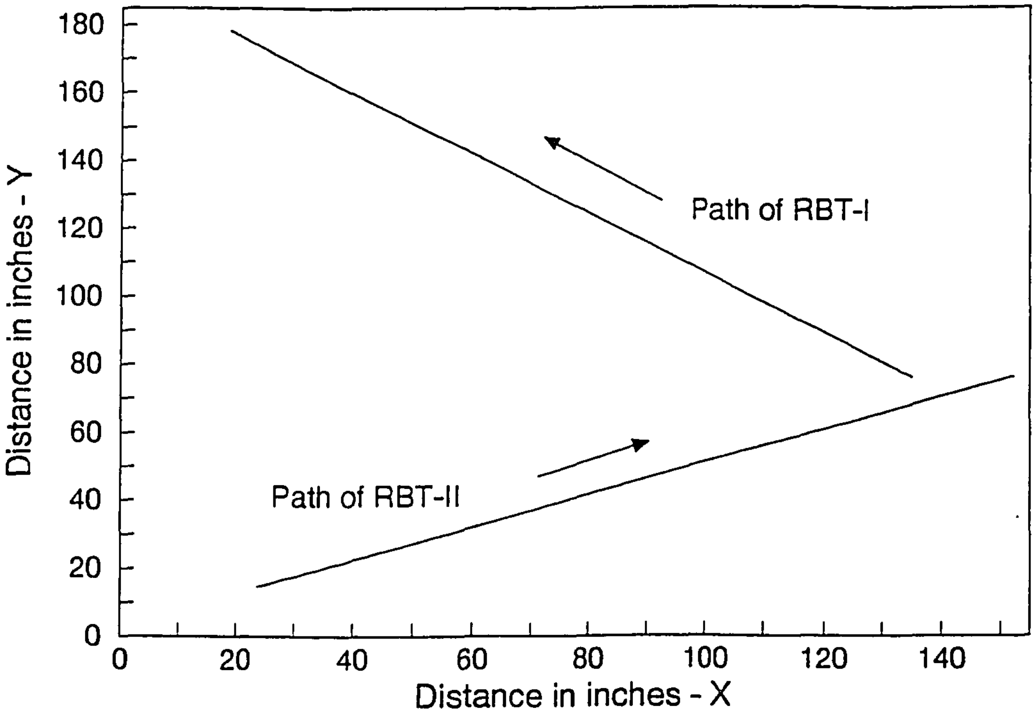


Figure 6.14: Example:1 Paths of RBT-I and RBT-II

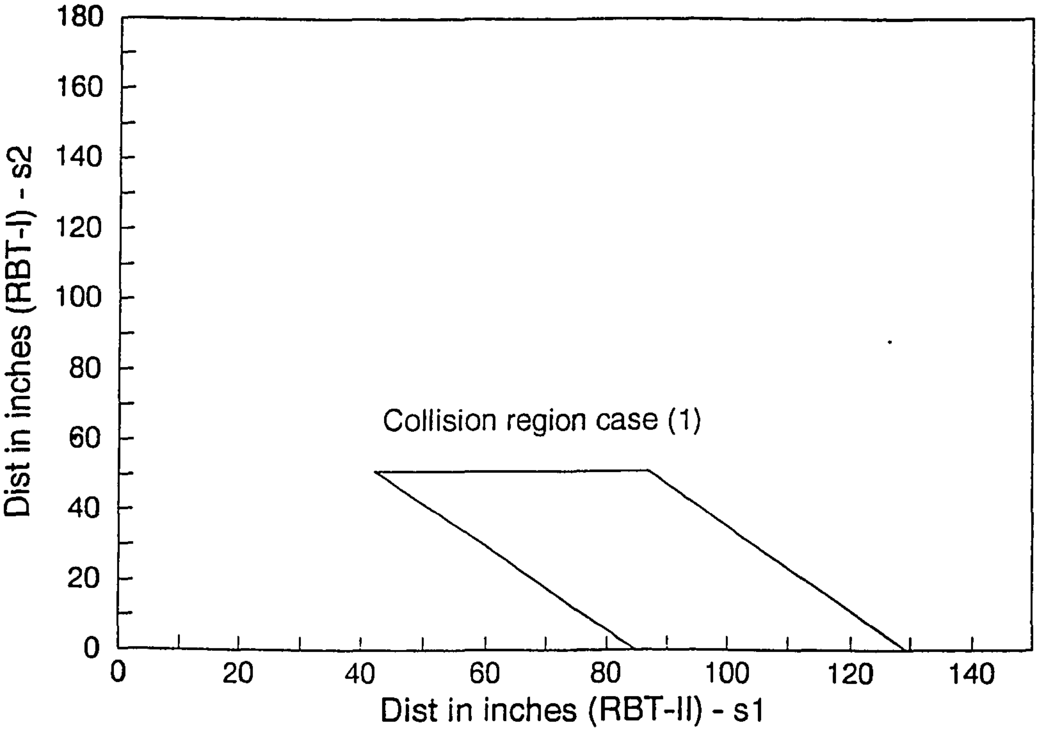


Figure 6.15: Example:1 Individual collision region in $s^1 - s^2$ space - case 1

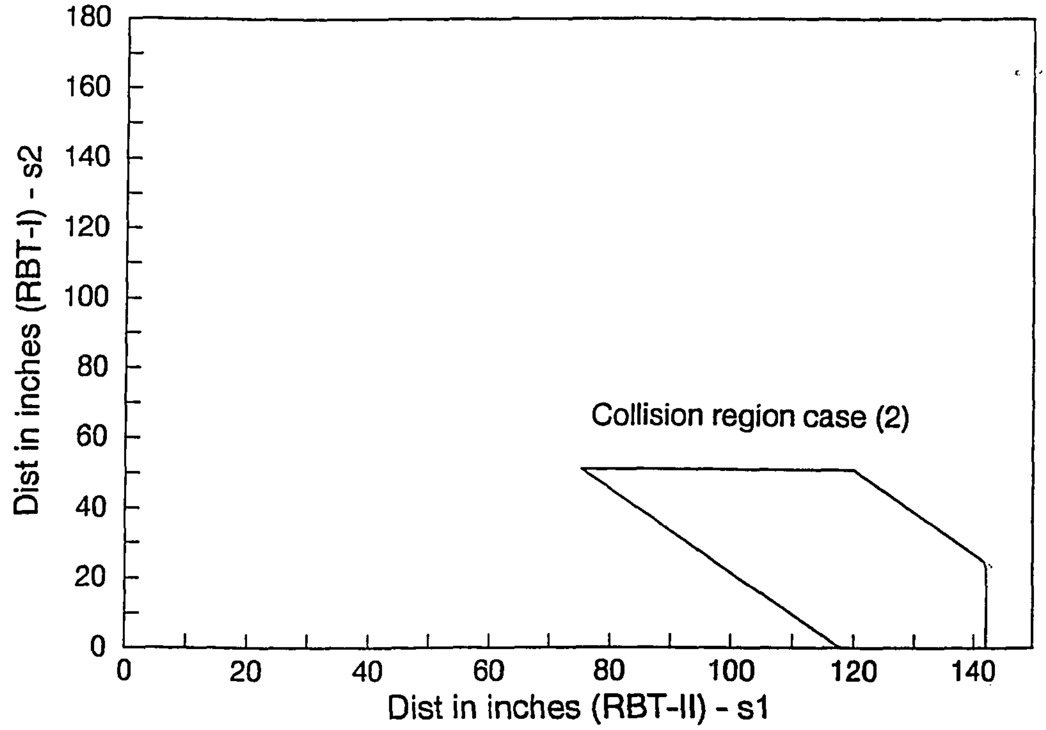


Figure 6.16: Example:1 Individual collision region in $s^1 - s^2$ space - case 2

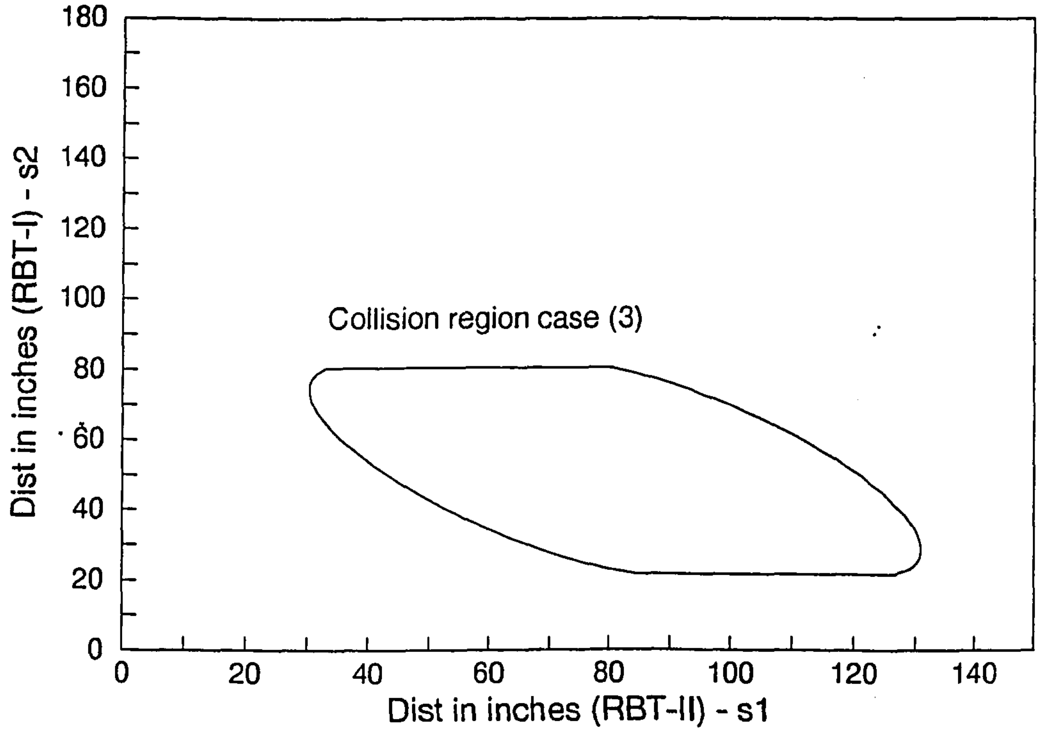


Figure 6.17: Example:1 Individual collision region in $s^1 - s^2$ space - case 3

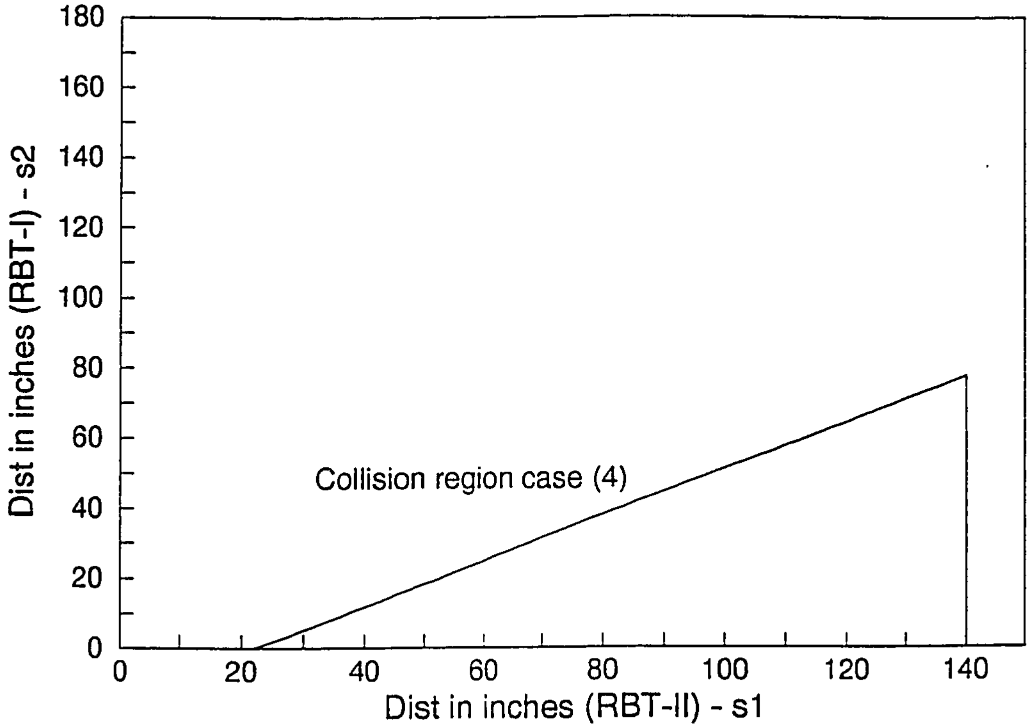


Figure 6.18: Example:1 Individual collision region in $s^1 - s^2$ space - case 4

The collective collision region is formed in $s^1 - s^2$ space as shown in figure 6.19 and the coordination curve constructed.

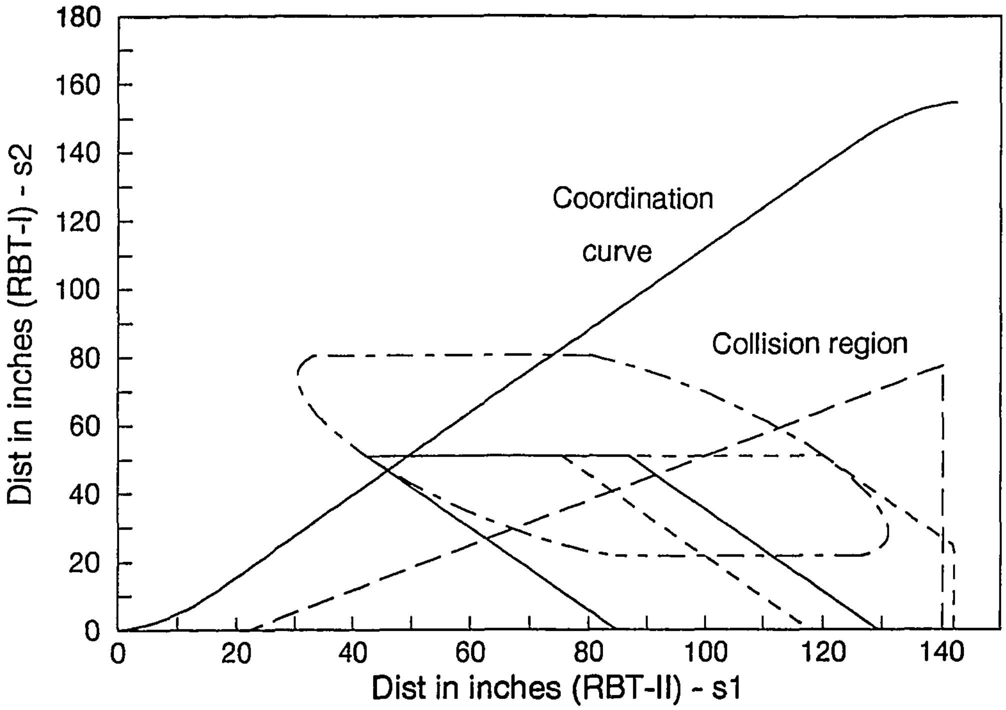


Figure 6.19: Example:1 Coordination curve and collective collision region in $s^1 - s^2$ space

The collision region is then transformed in $t^1 - s^2$ space and the path of RBT-I constructed as shown in figure 6.20. The algorithm found that the solution is to advance RBT-I. In other words, RBT-II was needed to be delayed by 59.01 seconds inorder to avoid collision. Figure 6.21 shows that RBT-I has to start ahead of RBT-II by the said time to avoid collision.

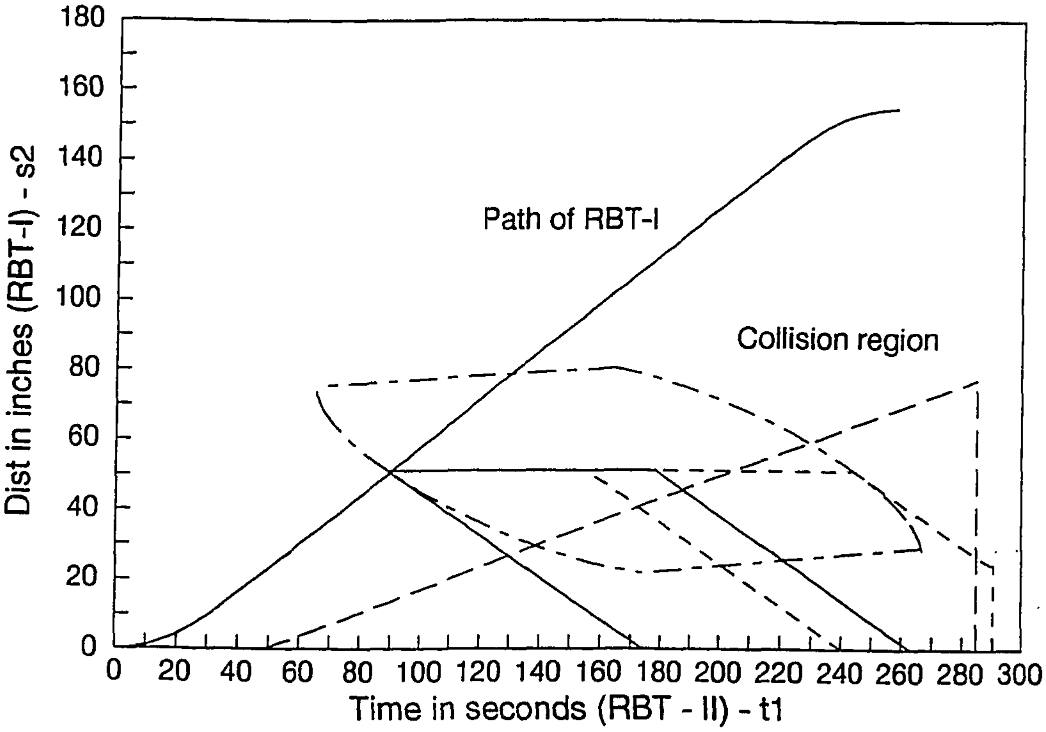


Figure 6.20: Example:1 Collision region and path of RBT-I in $t^1 - s^2$ space

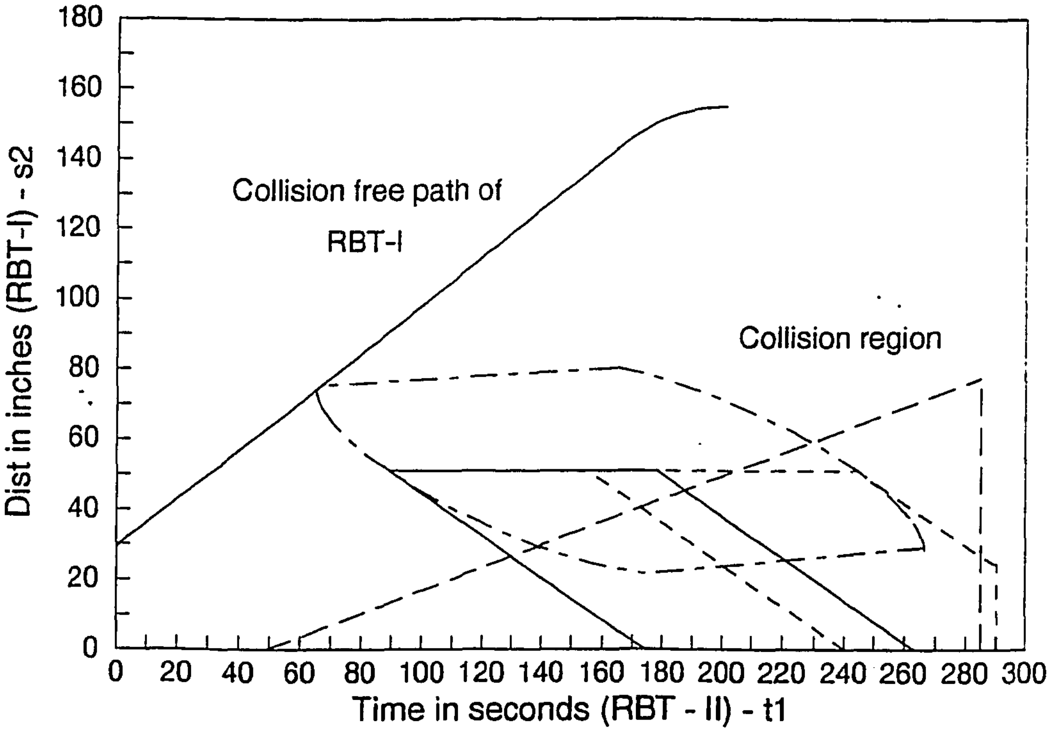


Figure 6.21: Example:1 Collision free path of RBT-I in $t^1 - s^2$ space

In example 2, the path of the two robots were made up of two segments. While RBT-I travelled at the same velocity in both the segments, the velocity of RBT-II was 0.7 inches/second in the second segment of motion. The paths of the two robots are as shown in figure 6.22. The total time taken by RBT-I was 156 seconds for path segment 1 while it traversed path segment 2 in 78 seconds. RBT-II had a travel time of 222 seconds and 111 seconds for its first and second path segments respectively. The individual collision regions corresponding to the four cases are shown in figure 6.23 - 6.26.

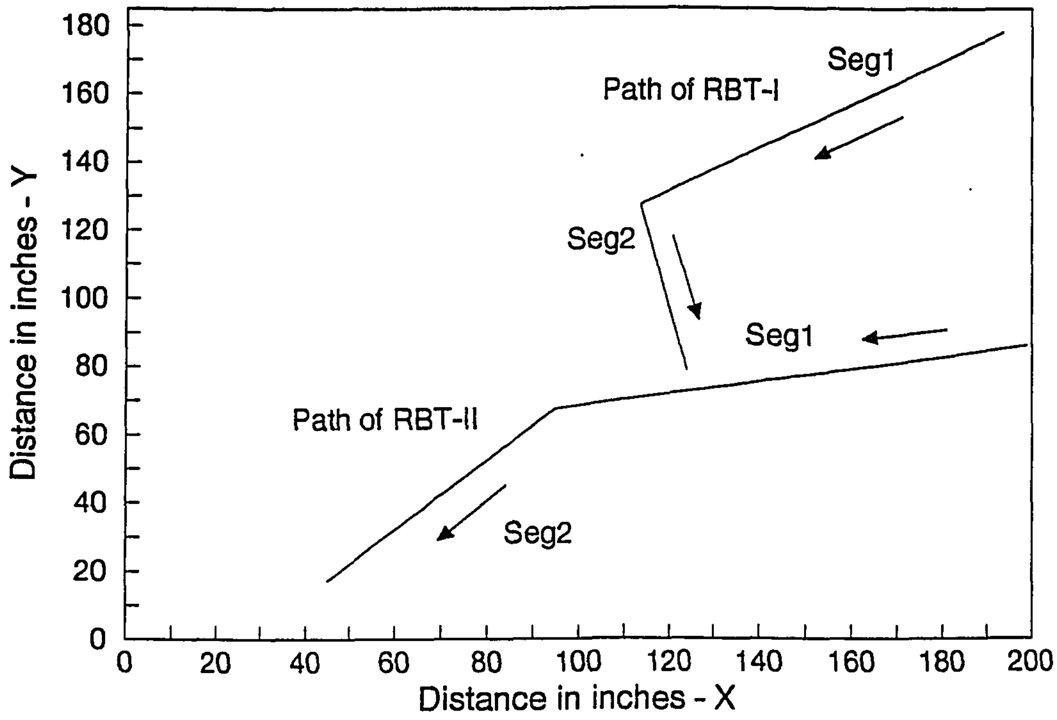


Figure 6.22: Example:2 Paths of RBT-I and RBT-II

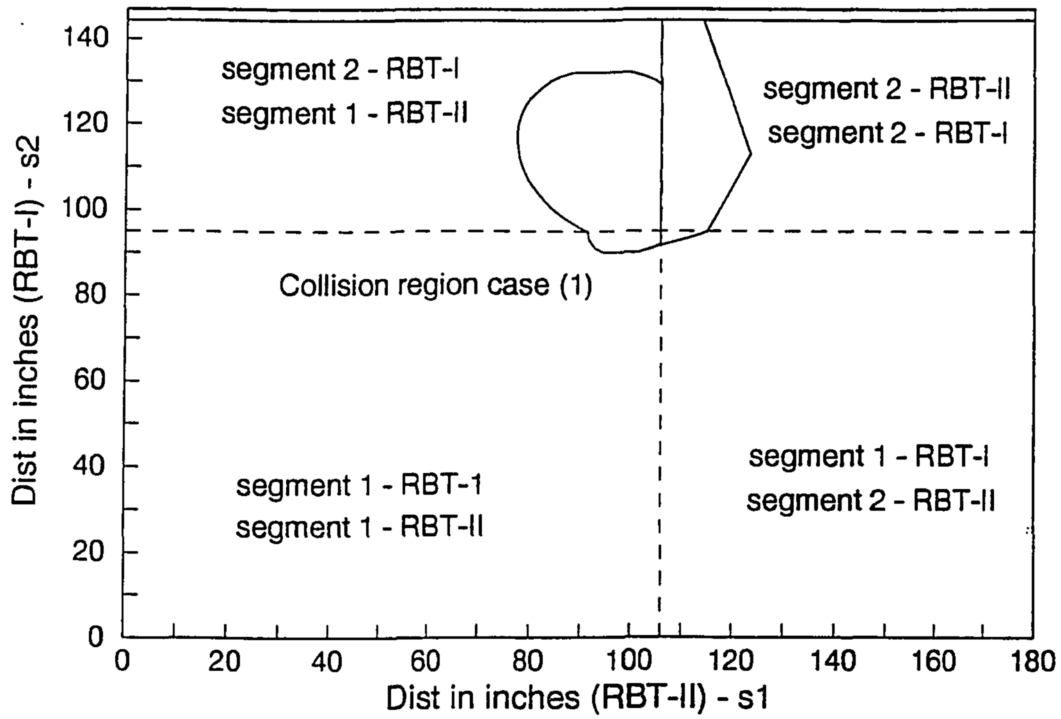


Figure 6.23: Example:2 Individual collision region in $s^1 - s^2$ space - case 1

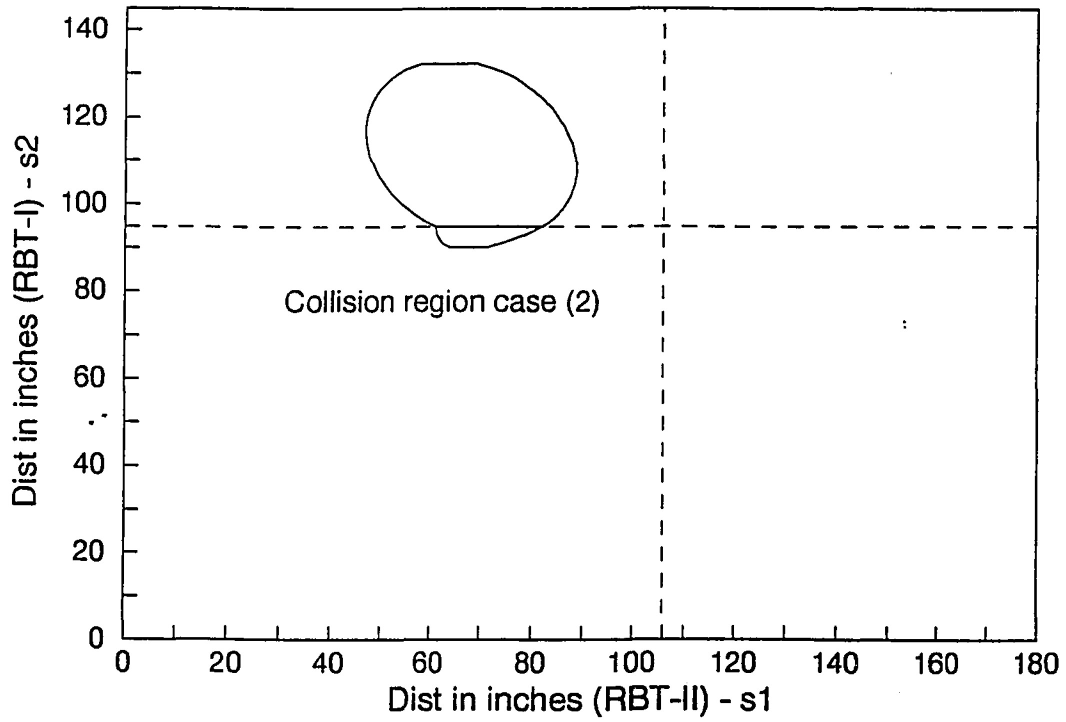


Figure 6.24: Example:2 Individual collision region in $s^1 - s^2$ space - case 2

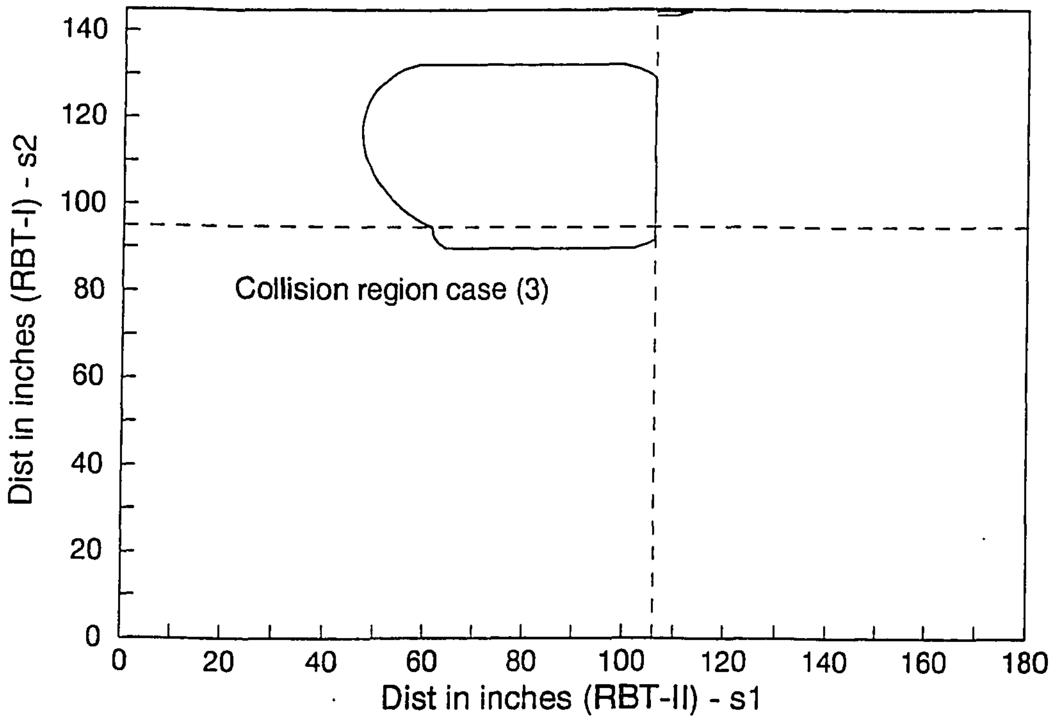


Figure 6.25: Example:2 Individual collision region in $s^1 - s^2$ space - case 3

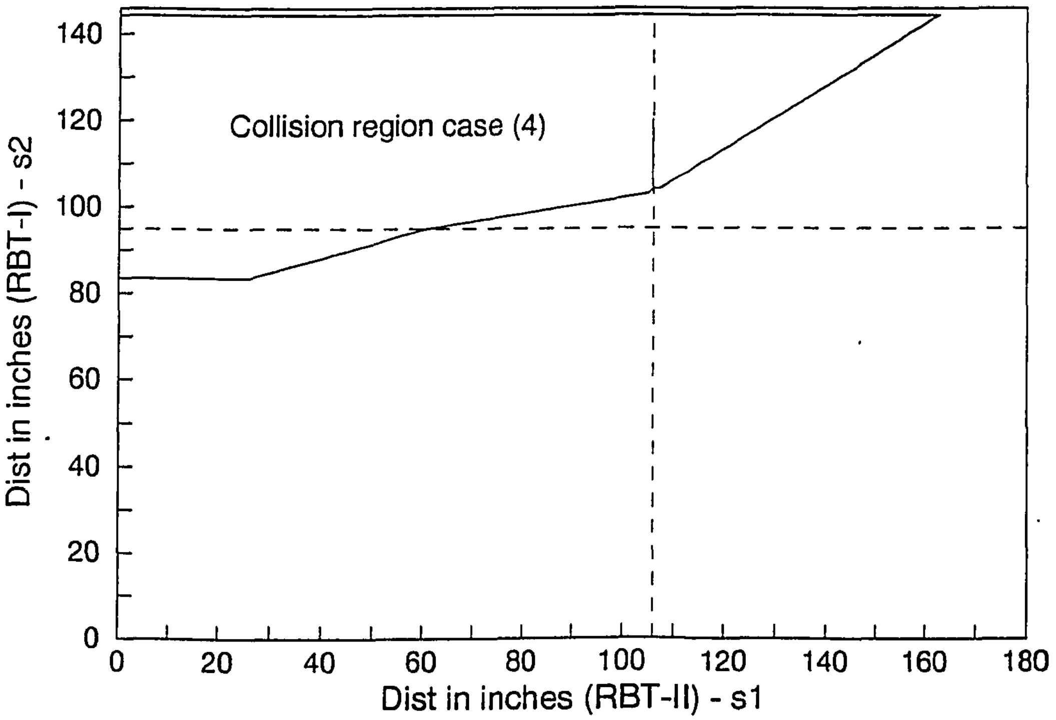


Figure 6.26: Example:2 Individual collision region in $s^1 - s^2$ space - case 4

Figure 6.27 shows the collective collision region and coordination curve in $s^1 - s^2$ space for this example. It can be observed from the figure that there is collision towards the end of the first path segment and in the better part of the second segment of the robots. The collision region was transformed into $t^1 - s^2$ space as before and plotted along with the path of RBT-I as shown in figure 6.28. The solution to this case was a delay of RBT-I by 86.78 seconds in its first path segment. The new path constructed with this delay time did not intersect the collision region any further and hence did not call for further delay in the second segment. The resulting collision free path is as shown in figure 6.29.

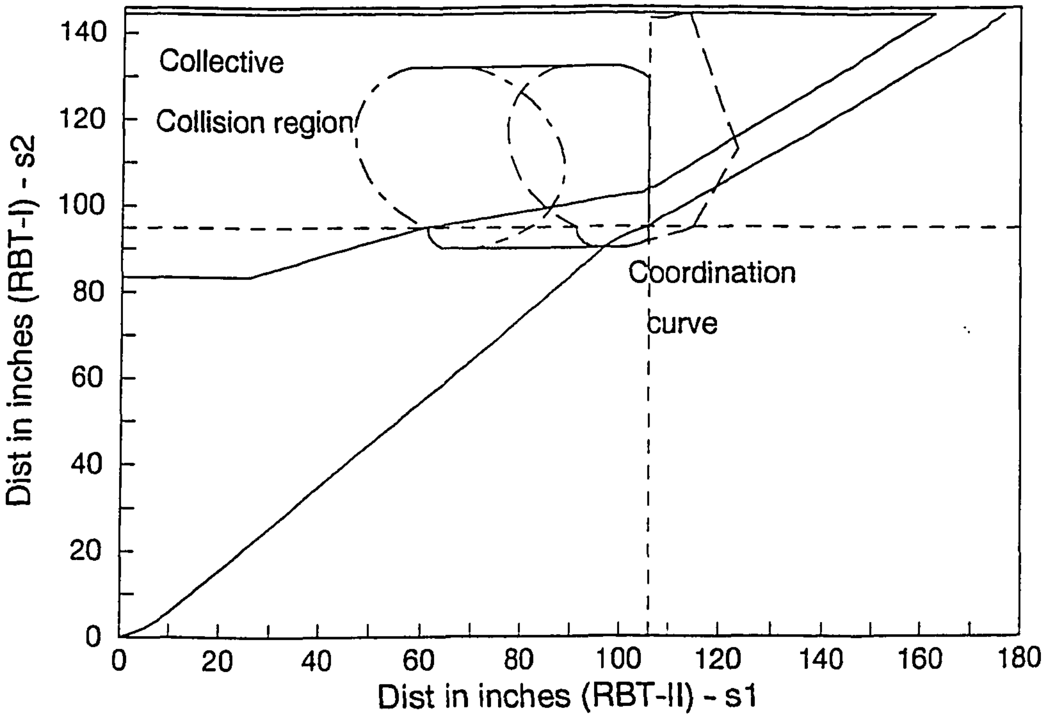


Figure 6.27: Example:2 Coordination curve and collective collision region in $s^1 - s^2$ space

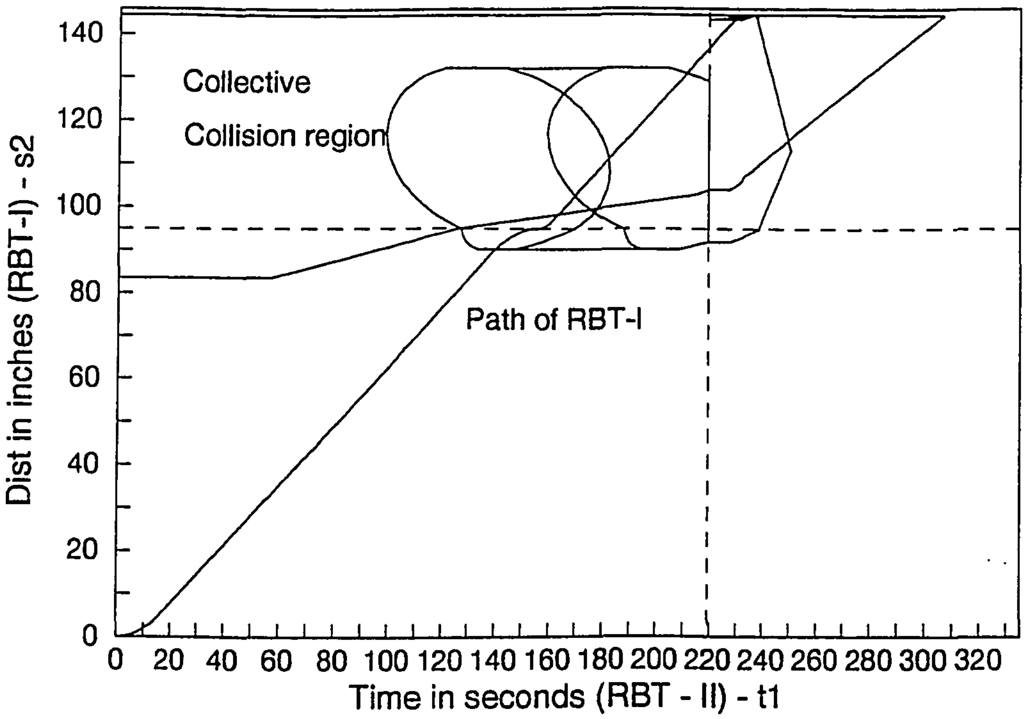


Figure 6.28: Example:2 Collision region and path of RBT-I in $t^1 - s^2$ space

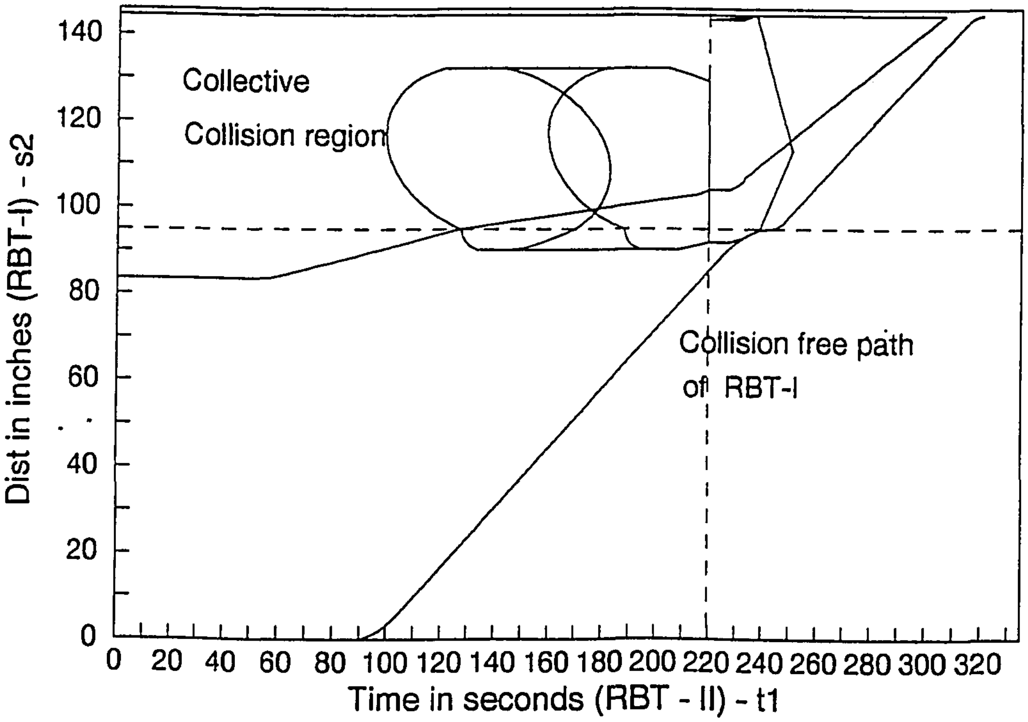


Figure 6.29: Example:2 Collision free path of RBT-I in $t^1 - s^2$ space

6.8 Implementation methodology

The proposed collision avoidance algorithm can be generalised for any given path and implemented in real time. The user provides input information about the proposed paths and velocity and acceleration time constants to Intouch, the human machine interface software. This then computes the total time required for the motion for the two robots. This information along with the user input are then passed on to another client application Microsoft Excel by using the DDE communication protocol. A constant DDE link needs to be established between Intouch and Excel for this purpose. This information is accepted by Matlab, a high performance numeric computation and visualisation software, from Excel, the latter acting as a server. Matlab simulates the result and returns the solution which is essentially the delay time, back to Excel. The communication channel between Excel and Matlab is closed once this data is received by Excel. This information is relayed back to Intouch, which is recognised by Intouch as there is a simultaneous change in the value of its DDE tagname linked to this variable in Excel. The user can then pass on this information to the programmable controller which then commands the respective automation controllers accordingly.

The collision problem of two gantry type robots sharing a common workspace was investigated. A practically implementable algorithm for collision avoidance of the two robots for a given path was presented. The solution was based on delaying one robot in between path segments till the other robot clears the collision region completely. This methodology can be extended to several number of segments as may be required and also extended for different types of paths. Chapter 7 concludes the research with a summary of the whole work, suggesting possible improvisations that can be effected and future research direction.

Chapter 7

Conclusion

This chapter summarizes the information provided in chapters 2, 3, 4, 5 and 6. It also highlights some of the draw-backs of the set-up, experiences during the course of research and suggests improvizations or recommends alternatives. A brief note on future course of research is made.

7.1 Conclusions

This experimental nature of research involved a part of mechanical installation and complete electrical installation of the experimental set-up and also making the system operational. Subsequently, a practically implementable collision avoidance scheme was proposed for the two robotic bridge transporters sharing a common workspace.

Chapter 1 started with a brief introduction to telerobotics and provided an idea of the hardware/software configuration of the control system. In chapter 2, the experimental set-up was explained in detail. Firstly, this set-up having been designed for industrial process operation called for a huge working area. As this could not be provided, the control cabinets, which otherwise would have been located in a separate control room had to be placed near the working area of the robotic bridge transporters thus providing only a small workspace for performing any operation. Manouvering the highly thick insulated cables, all designed for nuclear radiation environments, in such a small area was hence really difficult. It would have been a better idea to have routed the cables through a cable trench. The same problem applies to placing the supervisory control station near the working space of the robots. The concept of **Teleoperation** would have been better visualised had it been placed in a remote location or had the robots been in a different location. These problems need to be addressed in future.

Chapter 3 discusses the principle of operation of brushless servomotors and Modicon 4 axis automation controllers. The system makes use of brushless servo motors in line with modern technological development in this field. These motors have a greater capacity, are noise free and work in less favorable environments. They are easy to integrate with the main equipment. Modicon uses a high performance method called direct numerical processing technology to control the servo motors. The servomotor was modelled and simulated for various software settable parameters and the effect of them on velocity and position responses were observed.

The FA 3240 automation controllers are designed for controlling only four axes at a time. It would have been ideal to control both the RBTs using one controller for better coordinated motion which now is a limitation. Also the firmware for these controllers, C/ROS does not have a capability to accept target point locations as input on run-time. Hence it may not be an ideal software for teleoperation. Durrcon, a software package which can handle this problem has other limitations that C/ROS has overcome. Hence it calls for further investigation and update of the executive program module for the automation controllers for better control.

Chapter 4 discussed the theory of operation of the programmable logic controller used in this system. Although the 984 controller is capable of handling all major issues related to teleoperation, the ladder logic programming software ProWorx Plus has limitations in that it cannot handle complex function blocks and subroutines that can be done in other improvised ladder logic programming languages. This feature could have made the logic more concise and hence lead to faster operation.

Chapter 5 discussed about Intouch, the human machine interface software used for teleoperation and explained how tele-control is effected from the supervisory control station. Position information for RBT -I is obtained from both motor resolvers and absolute resolvers whereas for RBT-II, only the motor resolvers provide the data. This position information can be read off from the TR-130 operator terminal for the respective axes. However, it is desired that this information is available at the supervisory control station computer for effective tele-operation of the system. Hence a logic was provided to transfer this data to designated PLC registers from C/ROS registers and to read this information from PLC registers in Intouch. However, while performing other I/O operations when the equipment is in motion it calls for momentary turning off of the position reading sequence which demands operator alertness. This problem could be avoided by reading the position information from absolute position resolvers through an interface module which directly transfers the information to configured input registers of the PLC and hence does not require any logic for the same. However, only RBT-I has been provided with absolute position resolvers and the above problem is overcome. It is desired that RBT-II also is provided with absolute position resolvers to take care of this issue. Also, although the resolution of the equipment is up to thousandth of an inch, the resolution of position information obtained at the supervisory station is only up to tenth of an inch primarily because of the resolution limitation of the absolute interface module used for this purpose. It would be better if they could be replaced with higher resolution interface modules.

When the equipment moves at a higher speed, the value of the DDE tagname defined in Intouch database and linked to the resolver output changes so rapidly that the time required for Intouch to repaint the window viewer environment to accomodate this change is longer than it receives the next change in data from the resolver. Hence this results in the object moving by more number of pixels for a given time and projects that the real motion is in a jumpy, erratic fashion while it is actually not. This could be a limitation of Intouch being used for this sort of application.

Chapter 6 discussed the collision problem encountered when the two robotic bridge transporters share a common workspace and operate on a common task. A practically implementable collision avoidance scheme was proposed and tried for two different paths. The method to implement the same in real time was discussed.

7.2 Future research

This study focussed on control of the bridge transporters in general and did not discuss about control of the end effectors. Research is in progress to use a 50 Mflop digital signal processing technology based cotroller for controlling the 14 axes slave arm being used as the end effector. The teleoperatic control of the bridge transporters established in this research will hence be a part of the total control of the end effector. The collision avoidance algorithm proposed can still be used while moving the end effector between specified points although the collision problem between the end effectors themselves will then have to be considered for a comprehensive solution.

Appendix A

A. 1 Setting of Resolution

Resolution is defined as the distance moved in inches per motor revolution. The example below shows the computation of resolution for the following data.

1. Reducer ratio = 225:1
2. Circumference of pinion gear = $3\pi = 9.42$ inches

Distance moved for one revolution of gear = 3π inches

Hence for 1 inch movement the motor will have done $\frac{225}{3\pi}$ revolutions.

In other words, for $\frac{225}{3\pi}$ revolutions of motor, the equipment will have moved by 1 inch

Hence distance moved for one revolution of motor = $\frac{3\pi}{225} = 0.04188$ inches

It was found that C/ROS has a scale factor of 10. Hence actual resolution was set to 0.419 inches per motor revolution. Based on the above calculation the resolution has been computed for all other controller axes.

The following resolutions have been set for the different controllers.

- Controller 1 - RBT-II Bridge (x-axis) : 0.785

- - RBT-II Trolley (y-axis) : 0.419
- - RBT-II Telescoping tube (z-axis) : 0.940
- Controller 2 /3 RBT-I Trolley (x-axis) : 0.785
- RBT-I Bridge (y-axis) : 0.942
- Controller 4 - RBT-I Telescoping tube (z-axis) : 3.00

A. 2 Axis Parameters

The following is the setting of other important axis parameters common to all controllers.

- Acceleration time constant (ATC) : 64
- RIG Frequency Break : 128
- RPG Servo gain : 32
- Home offset : 0
- Following Error : 9
- Average Torque limit : 120
- Peak Torque limit : 127
- Homing direction : 0
- Number of poles : 6
- Linear Acceleration Time Constant: 1000

All other axis parameters listed in the menu have been set to their default values.

Appendix B

B. 1 Trapezoidal velocity Profile

For a trapezoidal velocity profile shown in figure,

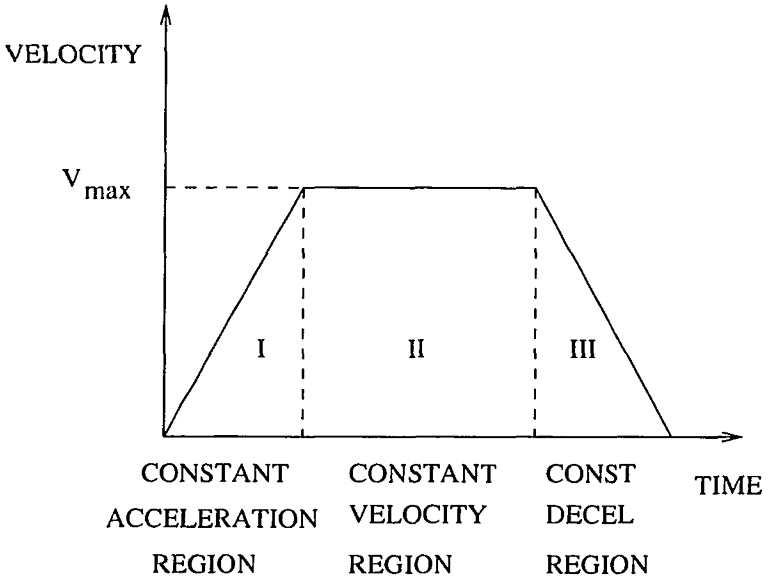


Figure B. 1: Trapezoidal velocity profile

the distance travelled in a given time "t" in the constant acceleration region is given by,

$$x1_n = \frac{V_n}{2\,t1_n}(t_n^2) \tag{B. 1}$$

where V_n is the velocity in the nth segment and $t1_n$ is the linear acceleration time constant for the nth segment in seconds. The distance travelled in constant velocity region is given by,

$$x2_n = V_n(t_n - t1_n) + \frac{V_n}{2}\, t1_n \tag{B. 2}$$

and similarly for the constant deceleration region, the distance travelled is given by,

$$x3_n = \frac{t_n - t2_n}{2}\, V_n \left[\frac{2t3_n - t_n - t2_n}{t3_n - t2_n}\right] + V_n\,(t2_n - t1_n) + \frac{V_n\, t1_n}{2} \tag{B. 3}$$

where $t2_n$ is the time at the end of the constant velocity region and $t3_n$ is the total travel time for the path segment.

Bibliography

[1] Thomas. B. Sheridan, Telerobotics, Automation and Human Supervisory Control, The MIT Press, 1992.

[2] Mohammad Jamshidi and Patrick J. Eicker, Robotics and Remote Systems for Hazardous Environments, Prentice Hall Inc., 1993.

[3] Understanding DNP servo Parameters, Application note, Modicon, Oct. 1985.

[4] Phillip C-Y. Sheu and Q.Xue, Intelligent Robotic Planning Systems, World Scientific Publishing Co., 1993.

[5] E. Freund and H. Hoyer, "Real-time Pathfinding in Multi-robot Systems Including Obstacle Avoidance," Int. J. Robotics Research, vol. 7, no. 1, pp 42-70, Feb. 1988.

[6] J. Lee and Z. Bien, "Collision-free trajectory control for multiple robots based on neural optimization network," Robotica, vol. 8

[7] C. Chang, M. J. Chung, and Z. Bien, "Collision-free motion planning for two articulated robot arms using minimum distance functions," Robotica, pp 137-144, 1990.

[8] B. H. Lee and C. S. G. Lee, "Collison-free motion planning of two robots," IEEE Trans. Syst. Man, Cybern., vol. SMC-17, no 1, pp 21-32, 1987.

[9] Y.Shin and Z.Bien, "Collision-free trajectory planning for two robot arms,' Robotica, vol. 7, pp. 205-212, 1989.

[10] Zeugnam Bien and Jihong Lee, "A Minimum-Time Trajectory Planning Method for Two Robots," IEEE Trans. on Robotics and Automation, vol. 8, no. 3, pp 414-418, 1992.

[11] Z. Shiller and H. H. Lu, "Robust Computation of path constrained time optimal motions," Proc. IEEE Int. Conf. Robotics and Automation, pp 144-149, 1990.

[12] Jihong Lee , Heon Seong Nam and Joon Lyou, "A Practical Collision-Free Trajectory Planning for Two Robot Systems," IEEE Int. Conf. Robotics and Automation, pp 2439-2444, 1995.

[13] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The Path-velocity decomposition," Int. J. Robotics Research, vol.5, no.3, pp72-89, 1986.

[14] Prototypical Rod Consolidation Demonstration Project, Phase III Final Report, Design description, vol. 2/4, Book 1/2.

[15] Yasuhiko Dote and Sakan Kinoshita, Brushless Servomotors - Fundamentals and Applications, Oxford Science Publications, 1990.

[16] C/ROS for 3240 Flexible Automation Controllers, Modicon, March 1992.

[17] 3240 C/ROS Remote I/O Processor, Modicon, 1990.

[18] Thomas. A. Hughes, Programmable Controllers, Resources for Measurement and Control Series, 1989.

[19] 984 Catalog and Specifier's Guide, Modicon, 1993.

[20] 984 System Planning and Installation Guide, Modicon.

[21] Pro**Worx** Plus - User's Manual, Version 1.30, Taylor Industrial Software, 1994.

[22] 984 Programmable Controller P190 Programming Guide, Modicon.

[23] Ladder Logic Block Library User Guide, Modicon, 1994.

[24] Edwin. G. Johnsen and William. R. Corliss, Human Factors in Teleoperator Design and Operation, John Wiley and sons Inc., 1971.

[25] Intouch - User's Guide, Wonderware Corporation, 1994.