

https://dspace.org/

A feasibility study in the development of an off-line PLC based robot control system

Bryson, Craig Weir

1990

http://hdl.handle.net/10539/21777

Bryson, Craig Weir (1990) A feasibility study in the development of an off-line PLC based robot control system, University of Witwatersrand, Johannesburg, http://wiredspace.wits.ac.za /handle/10539/21777>

Downloaded from a DSpace repository. https://dspace.org/

A FEASIBILITY STUDY IN THE DEVELOPMENT OF AN OFF-LINE, PLC BASED, ROBOT CONTROL SYSTEM

ABSTRACT

Robotics are becoming a more prominent force in the industrial environment, and research is being concentrated on control rather than on the robot. The feasibility of a substitute, off-line, plc based control system was investigated. Many advantages are associated with an off-line system, as well as the large financial saving (at the most 50% that of the existing controller).

A PLC with discrete I/O modules and a fast counting module were used. Open loop control was looked at, with optical encoders used for position control. Overshoot of the DC motors consistently occurred, and other external factors ensured the unpredictability and instability of open loop control.

It was concluded that closed loop control was necessary to ensure accurate positioning and speed control. PLC modules were investigated, and an axis control system (not yet commercially available) was found to ideally suit the purpose of servo/encoder control. This system makes use of speed and position feedback signals, essential for accurate terminal control of the robot.

ABSTRACT

Robotics are becoming a more prominent force in the industrial environment, and research is being concentrated on control rather than on the robot. The feasibility of a substitute, off-line, plc based control system was investigated. Many advantages are associated with an off-line system, as well as the large financial saving (at the most 50% that of the existing controller).

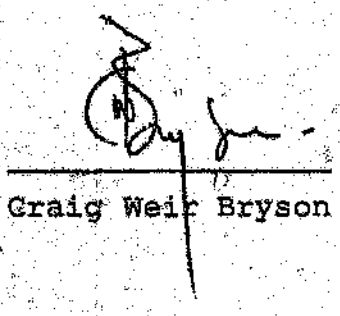
A PLC with discrete I/O modules and a fast counting module were used. Open loop control was looked at, with optical encoders used for position control. Overshoot of the DC motors consistently occurred, and other external factors ensured the unpredictability and instability of open loop control.

It was concluded that closed loop control was necessary to ensure accurate positioning and speed control. PLC modules were investigated, and an axis control system (not yet commercially available) was found to ideally suit the purpose of servo/encoder control. This system makes use of speed and position feedback signals, essential for accura e terminal control of the robot.

DECLARATION

I declare that this is my own, unaided work. Any assistance received during the project has been acknowledged or referenced.

It is being submitted as a requirement for the research project (MECN 560) as partial fulfilment for the degree of Master of Science in Engineering (Industrial), at the University of the Witwatersrand, Johannesburg.



Craig Weir Bryson

20th day of December, 1990

ACKNOWLEDGEMENTS

First and foremost, my thanks go to my supervisor, Mr E. R. Fielding, for his assistance and support, and willingness to try something new.

Thank you to Mr M. Nethersole and Dr J. van Coller for their expert assistance and advice regarding all the electronics that had to be contended with.

Thanks also to the staff of Telemecanique (Victor and Paul) who were always willing to help and give advice regarding the PLC equipment that was used.

Finally, my thanks to my colleague, Chris Barker, whose contribution to this project was invaluable, and to Anton Weavind for his enthusiasm, advice and support throughout the duration of the project.

CONTENTS

	page
ABSTRACT	i
DECLARATION	ii
ACKNOWLEDGEMENTS	iii
CONTENTS	iv
LIST OF FIGURES	viii
1 INTRODUCTION	1
1.1 ORIGINS OF ROBOTICS	1
1.2 DEFINITION OF A ROBOT	2
1.3 WHY USE A ROBOT ?	3
1.3.1 INCREASE PRODUCTIVITY	3
1.3.2 FLEXIBILITY	3
1.3.3 REDUCED LABOUR, MATERIALS AND ENERGY COST	4
1.3.4 IMPROVED QUALITY AND CONSISTENCY	4
1.3.5 HUMANIZATION OF THE WORKPLACE	4
1.3.6 ECONOMIC JUSTIFICATION	4
1.4 ROBOTICS RESEARCH JUSTIFICATION	6
2 LITERATURE SURVEY	8
2.1 THE EVOLUTION OF ROBOTICS	8
2.1.1 DEVELOPMENT OF INDUSTRIAL ROBOTICS	8
2.1.2 ROBOTICS WORLDWIDE	11
2.1.3 CURRENT STATUSES	14
2.2 ROBOT CONFIGURATIONS	15
2.2.1 CYLINDRICAL CONFIGURATION	16
2.2.2 SPHERICAL OR POLAR CONFIGURATION	17
2.2.3 CARTESIAN OR RECTANGULAR CONFIGURATION	17
2.2.4 REVOLUTE OR ARTICULATING CONFIGURATION	18
2.2.5 SCARA CONFIGURATION	18
2.2.6 PARALLEL CONFIGURATION	19
2.3 ROBOT MOTIONS	21
2.3.1 POINT-TO-POINT (PTP)	21
2.3.2 CONTINUOUS-PATH	22
2.4 ROBOT CLASSIFICATIONS	25
2.4.1 FIXED/VARIABLE SEQUENCE ROBOTS	25

2.4.2 SERVO/NON-SERVO ROBOTS 25
2.4.3 POINT-TO-POINT/CONTINUOUS PATH ROBOTS 26
2.4.4 FIRST/SECOND/THIRD GENERATION ROBOTS 27
2.5 COMPUTERISED MOTION CONTROL 28
2.6 ROBOT CONTROLLERS 36
2.7 PLC CONTROL 38
2.7.1 INTRODUCTION TO PLCs 38
2.7.2 CURRENT APPLICATIONS 40
2.7.3 RECENT DEVELOPMENTS 41
2.8 CONTROLLING SOFTWARE 43
2.9 COMMON ROBOT PROGRAMMING TECHNIQUES: ON-LINE 46
2.9.1 MANUAL LEAD-THROUGH PROGRAMMING 46
2.9.2 TEACH PENDANT PROGRAMMING 46
2.9.3 ADVANTAGES OF ON-LINE PROGRAMMING 46
2.9.4 DISADVANTAGES OF ON-LINE PROGRAMMING 47
2.10 COMMMON ROBOT PROGRAMMING TECHNIQUES: OFF-LINE 48
2.10.1 TEXTUAL PROGRAMMING LANGUAGES 48
2.10.2 ROBOT SIMULATION 52
2.10.3 ADVANTAGES OF OFF-LINE PROGRAMMING 61
2.10.4 DISADVANTAGES OF OFF-LINE PROGRAMMING 62
2.11 FUNDAMENTALS IN MOTION CONTROL 63
2.11.1 OPEN LOOP CONTROL 63
2.11.2 CLOSED LOOP CONTROL 64

3 PROJECT OBJECTIVES 67

4 EXPERIMENTAL EQUIPMENT 69
4.1 THE ROBOT 69
4.1.1 GENERAL CONSIDERATIONS 69
4.1.2 STRUCTURAL GEOMETRY 70
4.1.3 ROBOT ARM GEARING 71
4.2 DC SERVO MOTORS 72
4.2.1 EXTERNAL INTERFACING 72
4.2.2 OPERATIONAL CHARACTERISTICS 73
4.3 OPTICAL SHAFT ENCODERS 75
4.3.1 GENERAL OPERATING CHARACTERISTICS 75
4.3.2 EXTERNAL INTERFACING 76
4.4 THE PROGRAMMABLE LOGIC CONTROLLER (PLC) 78
4.4.1 THE TSX 47 PC RACK 78

4.4.2 THE POWER SUPPLY MODULE 78
4.4.3 THE PROCESSOR MODULE 79
4.4.4 DISCRETE I/O MODULES 79
4.4.5 THE FAST COUNTING MODULE 79
4.4.6 THE PROGRAMMING TERMINAL 80
4.4.7 THE SWITCH BANK 80
4.5 PLC/PC INTERFACE SYSTEM 82
4.5.1 SYSTEM ADVANTAGES 82
4.6 FREQUENCY GENERATOR 84
4.7 PULSE WIDTH MODULATOR 85

5 FEASIBILITY STUDY 87
5.1 OPTICAL SHAFT ENCODERS 88
5.1.1 ENCODER REPAIR 88
5.1.2 OPERATING CHARACTERISTICS 88
5.1.3 ENCODER INTERFACING 93
5.1.4 ENCODER RESOLUTION 94
5.2 DC SERVO MOTORS 97
5.2.1 MOTOR CALLIBRATION 97
5.2.2 GEARING DETERMINATION 99
5.3 CONTROL DECISIONS 101
5.3.1 OPEN LOOP CONTROL 101
5.4 I/O CAPABILITIES - COUNTER FUNCTION BLOCK 103
5.5 ACCUMULATION OF AN INCREMENTAL WORD 106
5.6 I/O FACILITIES - FCM (UPCOUNTING) 108
5.7 I/O FACILITIES - FCM (UP AND DOWNCOUNTING) 112
5.8 HIGH SPEED COUNTING 115
5.9 FCM READING CAPABILITIES - FREQUENCY GENERATOR 1 118
5.10 FCM READING CAPABILITIES - FREQUENCY GENERATOR 2 122
5.11 OPEN LOOP CONTROL 123
5.12 EXISTING ROBOT CONTROL 128
5.13 AXIS CONTROL SYSTEM 129
5.13.1 AXIS CONTROL PRESENTATION AND APPLICATION 129
5.13.2 AXIS CONTROL SPECIFICATIONS 133
5.13.3 AXIS CONTROL OPERATION 133
5.13.4 THE CONTROL LOOP 134

6 CONCLUSIONS 136

7 FUTURE WORK 140

8 REFERENCES 142

APPENDIX A 147
PT 300V ROBOT 148
ROBOT CONTROLLER 150
HARMONIC DRIVES 152
ENCODERS 156
PLC COMPONENTS 159
APPENDIX B 165
COUNTER FUNCTION BLOCK 166
FAST COUNTING MODULE 168
FREQUENCY GENERATOR 1 176
FREQUENCY GENERATOR 2 181
AXM SERVO LOOP 183
PULSE WIDTH MODULATOR 189

LIST OF FIGURES

		page
2.1	Estimated 1984 world robot population	12
2.2	The six robot degrees of freedom	15
2.3	Cylindrical coordinate robot	16
2.4	Spherical coordinate robot	17
2.5	Cartesian coordinate robot	17
2.6	Revolute coordinate robot	18
2.7	SCARA type robot	19
2.8	A parallel robot	19
2.9	The coordinate system of a robot end effector	29
2.10	Construction of a homogeneous transformation	30
2.11	Two arm configurations with identical end effector positions	35
2.12	A typical robot and robot controller	37
2.13	Different forms of robotics software	44
2.14	The various levels of abstraction of user software in a robotic system	45
2.15	Classification of robot languages	51
2.16	Example of graphics produced by the GRASP simulation system	53
2.17	An integrated robot welding system	58
2.18	Open loop control	63
2.19	Closed loop control	65
4.1	Line representation of the robot	71
4.2	External interfacing of the DC motors	73
4.3	Pull up of encoder output signals	77
4.4	Emitter follower circuit	77
4.5	PLC, current loop convertor and PC link	82
5.1	Encoder circuit - power input to chip	89
5.2	Square wave triggering	90
5.3	Connection and pin diagram for encoder chip	91
5.4	Encoder circuit - into and out of chip	92
5.5	Wiring diagram to test encoder operation	93
5.6	Timing diagram - B1 high every alternate pulse	95
5.7	Program - encoder resolution determination	95

5.8 Calibration curves for DC servo motors 98
5.9 Gearing determination - horizontal axes 100
5.10 Gearing determination - vertical axis 100
5.11 Component connection and control 104
5.12 Program - simulated counter 106
5.13 Circuit - wiring of FCM 109
5.14 Program - FCM incorporation, upcounting 110
5.15 Circuit - FCM incorporation, up and downcounting 112
5.16 Up and downcounting wave trains 114
5.17 Program - FCM counting simulation 116
5.18 Circuit - frequency generator connection 119
5.19 Timing diagram of pulse train 120
5.20 Program - motor accuracy 124
5.21 Motor overshoot of DC servo motors 126
5.22 Axis control presentation 129
5.23 Axis control module 130
5.24 Internal layout of module 131
5.25 ACM move diagrams 132
5.26 ACM positioning curves 132
5.27 ACM servo loop 134

6.1 Robot controller 151

A3.1 Harmonic drive components 153
A3.2 Harmonic drive operation 155

A4.1 Cut away of a typical encoder 157
A4.2 Optical encoder components 157
A4.3 Encoder pulse train 157
A4.4 Incremental and absolute encoders 158

A5.1 The power supply module 160
A5.2 The processor module 160
A5.3 User memory cartridge 161
A5.4 Discrete I/O module 161
A5.5 Wiring diagram for discrete I/O modules 162
A5.6 TSX AXT 200 counting and positioning module 163
A5.7 The PLC programming terminal 164

B1.1 The function block element 167

B2.1 Channel 0 & 1 inputs and outputs 170
B2.2 Timing diagram of pulse train 171
B2.3 Controller operation 172
B2.4 Maximum counting rate curve 173
B2.5 Counting accuracy curve 174
B2.6 Module wiring configuration 175

B3.1 Achieving quadrature 178
B3.2 Problem encountered with 555 timer chip 178
B3.3 Circuit diagram 179
B3.4 Bread board layout - frequency generator 1 180

B4.1 Bread board layout - frequency generator 2 182

B5.1 Characteristic pulse train output from a PWM 190
B5.2 Voltage dropper 191
B5.3 Voltage adder 191
B5.4 [illegible] chip connection 192
B5.5 The signal driver 192
B5.6 The inverter 193

1 INTRODUCTION

Within a very short period of time, robotics has grown in the public eye from being science fiction to being the panacea for industry. Increasing numbers of people from a variety of backgrounds are being involved, either through choice or through necessity.

1.1 ORIGINS OF ROBOTICS

Man had being toying with the idea of somehow building a mechanical version of himself long before tentative work was started, which eventually led to the successful introduction of the industrial robot in the early sixties.

Throughout recorded history man has had a preoccupation with self-moving machines made, at least partially, in his own image. In the March 1942 issue of 'Astounding Science Fiction' Isaac Asimov wrote a story, 'Runabout,' in which he first propounded the famous three laws of robotics. [1]

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm;
2. A robot must obey the order given to it by a human being, except where those orders would violate the first law;
3. A robot must protect its own existence, except where it would violate the first and second laws.

Unimate [1], the founder of Unimation, and considered by many to be the father of modern industrial robotics, has pointed out that the three laws remain worthy design standards for roboticists to this day.

1.2 DEFINITION OF A ROBOT

The definition used by the Robot Institute of America (RIA) and the British Robot Association (BRA) are largely similar, and state that a robot is a reprogrammable, multifunctional manipulator designed to move materials, parts, tools or specialised devices through variable programmed motions for the performance of a variety of tasks. [1],[2]

The Japanese Industrial Robot Association (JIRA), [1], divides the term robot into six classes: manual handling devices, pick-and-place devices, programmable variable-sequence manipulators, robots taught manually, robots controlled by a programming language, and robots which can react to their environment.

Thus, in Japan, a pick-and-place device would be termed a robot, whereas in Europe and the West, it would be considered to be a form of fixed automation. Excluding the definitions supplied by JIRA, the following would not be classified as robots: prostheses and exoskeletons, [illegible] devices, computer numerical control (CNC) devices, [illegible] vehicles, and the list goes on. [1],[2]

1.3 WHY USE A ROBOT ?

Before one can even think of implementing a robot into a work environment, various preliminary considerations must be taken into account, i.e. if a robot is the right choice, parts orientation and location, the requirements of the system in terms of the robots cycle time, load and reach, the availability of support staff, consulting with labour in connection with automation implementation, good engineering versus expensive robot capability, and the impact on the workforce in terms of relocation [2].

Perhaps one of the major incentives for introducing robots arose because the cost of employing a worker increased particularly steeply throughout the 1960's, especially in the USA and Western Europe. As a result, the cost of buying and running a robot gradually became a more attractive proposition to managers whose firms were struggling to remain competitive. Another reason, in some countries such as Japan, has been a steadily increasing shortage of available manpower [1]. Various reasons are listed below.

1.3.1 INCREASE PRODUCTIVITY

The advantage of using a manufacturing robot is that it is able to sustain a high rate of production with good accuracy over a long period of time, i.e. there is continuous operation (no tea breaks). [2], [3]

The implementation of a robot in a working environment would mean multiple shift operation, possibly unmanned, high up time, and improved equipment utilisation.

1.3.2 FLEXIBILITY

Investing in a robot will allow one to accommodate different products simply by changing the controlling software. The robot can be used for different tasks as

desired, thus providing time and space elasticity. [2]

1.3.3 REDUCED LABOUR, MATERIALS AND ENERGY COSTS

The capital cost of a robot is not affected by inflation, thus a robot gets less expensive while labour costs continue to rise. There is a reduction in work in progress inventory and reduced rejection rates, thus less scrap. In using robots, there is no education or retraining of manpower costs, simply reprogramming costs. An off-line system will further lessen these costs. In this way robots will work towards providing a structured environment. [2], [3]

1.3.4 IMPROVED QUALITY AND CONSISTENCY

If a robot is implemented in a working environment, and its operation is continuous, there will exist a stability of the process involved. Precise and consistent repeatibility may be desired (and usually is), and in this field a robot is excellent. Hopefully this will lead to improved quality control.

1.3.5 HUMANIZATION OF THE WORKPLACE

It has been suggested that a reason for increasing robot introduction may be the growth of legislation and awareness about health and safety in factories. [1]. Robots can take over from workers who would otherwise be exposed to toxic fumes, excessive heat, noise and vibration, or physical danger. Physically demanding jobs can also be passed along to a robot. Similarly, there are demands, in some countries at least, for greater job satisfaction. Robots can quite happily be used to carry out the most tedious of tasks without becoming alienated or bored.

1.3.6 ECONOMIC JUSTIFICATION

There is no point in a firm installing robots if, in the

long term at least, the move does not result in an improved economic position. This is not to say, of course, that removal of humans from dangerous activities should not be considered as the most important justification, but to be realistic, most companies will be interested in how robots may improve profits. It therefore becomes vital to understand some of the economic considerations involved in evaluating potential robotics applications. This is true especially for managers.

There are 'traditional' direct factors that are of prime importance. Various costs include system purchase price, cost of special tooling, installation cost, on time costs, maintenance costs, operating and programming costs, depreciation and cost of capital. Savings include sale of old equipment, labour savings and increased throughput. In addition to the costs and savings already mentioned, there are indirect factors which may be considered in the decision of robot implementation. These 'educated estimates' for indirect savings include raw material, WIP and finished goods inventory, scrap and rework, inspection operator ttaining, floor space, quality, safety and even goodwill.

All the above mentioned considerations are vitally necessary for a comprehensive evaluation, and could economically justify the introduction of the robot in certain industrial environments.

1.4 ROBOTICS RESEARCH JUSTIFICATION

The question of why robots are used has been addressed, and now it is necessary to look at why research is carried out in the field of robotics in South Africa.

The factors already mentioned regarding the justification of robots in the workplace are especially relevant in developing countries, particularly those making the transition from third to first world status and manufacturing strategies. South Africa is a prime example of such a country, and the scope for productivity improvement is extremely vast. The economic situation of this country at the moment warrants further research to avoid added expenses arising from licensing and importing technology.

Recent developments in the field have been concentrated mainly in the field of advancing computing power and its associated control implications. As the mechanics behind robots have been available to us for so long, there is little room for improvement, though this does not mean that any new technologies cannot be integrated into the hardware where they would enhance dexterity, efficiency, or capabilities of the robot. With computers having recently acquired to low cost, user friendly, yet ever powerful states, it makes sense to capitalise on this new development and to incorporate it into the field of robotics. In this light, it is the personal computer (PC) integration into the robot loop that is of prime consideration. Further reasons for the adaption of the control side of the robot loop include numerous advantages associated with off-line programming methods, the improvement of the robot through enhanced motion capabilities, and others such as enhanced problem solving abilities made possible through simulation systems.

Programmable logic controller (PLC) integration into the

robot loop is justified by the neccessity of a powerful I/O handling device. This is necessary if the current robot controller is to be substituted with a PLC/PC system. Numerous advantages associated with such a system are prevelant, and these are listed below.

- A large saving in cost, as a PLC and PC combination will result in a cost saving of anywhere between 30 to 60 % over the cost of an existing controller.
- The PLC is much smaller than the robot controllers, making them very convenient, and increasing the robot mobility.
- The advantages of programming a robot through an off-line facility are numerous, and are listed in section 2.10.
- Advantages of computer integrated manufacture are apparent, and these include improved machine utilisation, batch manufacture elimination, shorter manufacturing lead times, and greater scheduling flexibility.
- Advantages of using a PLC are also prominent (in certain applications), and these include ease of use, they are fast, powerful, flexible, modular, and easy to maintain.

2 LITERATURE SURVEY

2.1 THE EVOLUTION OF ROBOTICS

2.1.1 DEVELOPMENT OF INDUSTRIAL ROBOTICS

It was never inevitable that the industrial applications of robotics should have been its first major market. It is all too easy to think of robotics as being a currently very fashionable subset of industrial automation which, after the initial excitement has died down, will become just yet another tool at the production engineer's disposal. Yet this is only true of that part of robotics which is 'industrial' robotics; there are potentially very many 'non-industrial' applications for robotics which may even one day overtake industrial robotics in importance. If, 30 years ago, by some quirk of history, mankind had been very adept at building computers yet comparatively poor at constructing mechanical devices, we might now have large numbers of 'intelligent' domestic robots and be wondering how long it would take before this robotics technology could be adapted for practical industrial use, in much the same way as the Ancient Greeks used the principles of the steam engine to operate gimmicks in their temples, yet it took the best part of two millennia before the same principles were put to practical use.

Modern industry has grown out of the mechanisation of the industrial revolution which was fired by the development by James Watt of his steam engine in the latter half of the eighteenth century. With mechanised production came the ability to make parts which were almost identical - so much so that in the early nineteenth century Eli Whitney was able to produce 12,000 muskets for the US Government in which all the same parts were interchangeable between guns. Mass production had truly arrived, and by the late nineteenth century the constant demand for improved production rates had led to several refinements to

metal-cutting machines which culminated in automatically controlled machines, such as automatic lathes.

The seemingly insatiable desire for the newly invented motor car created a demand for better machine tools to manufacture them with, and the development by Henry Ford of the production (or transfer) line meant that by 1914 his company was producing over 1 000 000 model T cars a year. However, although such automation (which later in some cases became extremely sophisticated and was called Detroit automation) is suitable for very large volume production, with reduced labour costs and low unit cost, even so it requires long lead times (a major changes in design may require several years to implement) and so is inherently inflexible to evolving technologies and requirements.

For a long time medium and low-volume batch production had to be undertaken using conventionally manually operated machine tools, but in the late 1940s John T Parsons suggested a method for automatically guiding a milling machine by means of coded punch cards in order to machine the complex shapes of the helicopter blades he was working on. In 1949, the US air force commissioned the MIT Servomechanisms Laboratory to develop such a numerical control (NC) machine. A decade later, NC machines were being successfully used in production, and work had begun on developing a universal programming language called APT. Nevertheless, as mentioned earlier, NC or even CNC machines, are not thought of as robotic, because they do not exhibit the flexibility of the type of operation that true robots do.

Although the application of numerical control to specific machines grew in range from automatic fabric knitting to blueprint drafting, the newly available industrial robot of the 1960s, which was in effect a general purpose handling machine, had an uphill struggle to find cost effective applications. Whereas NC machines tended to replace craftsmen operating manual versions of the same machine,

metal-cutting machines which culminated in automatically controlled machines, such as automatic lathes.

The seemingly insatiable desire for the newly invented motor car created a demand for better machine tools to manufacture them with, and the development by Henry Ford of the production (or transfer) line meant that by 1914 his company was producing over 1 000 000 model T cars a year. However, although such automation (which later in some cases became extremely sophisticated and was called Detroit automation) is suitable for very large volume production, with reduced labour costs and low unit cost, even so it requires long lead times (a major changes in design may require several years to implement) and so is inherently inflexible to evolving technologies and requirements.

For a long time medium and low-volume batch production had to be undertaken using conventionally manually operated machine tools, but in the late 1940s John T Parsons suggested a method for automatically guiding a milling machine by means of coded punch cards in order to machine the complex shapes of the helicopter blades he was working on. In 1949, the US air force commissioned the MIT Servomechanisms Laboratory to develop such a numerical control (NC) machine. A decade later, NC machines were being successfully used in production, and work had begun on developing a universal programming language called APT. Nevertheless, as mentioned earlier, NC or even CNC machines, are not thought of as robotic, because they do not exhibit the flexibility of the type of operation that true robots do.

Although the application of numerical control to specific machines grew in range from automatic fabric knitting to blueprint drafting, the newly available industrial robot of the 1960s, which was in effect a general purpose machine, had an uphill struggle to find any application, whereas NC machines began to replace manual versions of the machine

robots were capable only of unskilled labour which was still relatively cheap, especially for low production rates. In addition, it was not at all clear for which jobs a robot was best suited, as there was no single type of machine for which it was an automatic replacement.

As the sophistication of robot controllers improved however, and techniques were developed for structuring robot surroundings in ways which, to an extent, compensated for robot inadequacies, so new applications such as spray painting became cost effective. In the 1970s, industrial robotics came to be considered as one of the aspects of an overall philosophy of computer aided manufacture (CAM), and with ever increasing labour costs and pressure to remove workers from hazardous or unpleasant jobs, together with an insistence from the consumer market for wider diversity of product styles, interest in robotics grew in industry. As the world recession of the late 1970s struck, so academic interest rapidly turned into a desperate fight to increase productivity while trimming manning levels.

Mechanical engineering requirements for industrial robots had never really been particularly demanding, and could more than be catered for by existing technology. The restraining technical factor however was primarily in control, yet even so, the available computing power for cost-effective industrial robots was just about sufficient to allow them to start to integrate with the factory environment, and take over many materials-handling tasks and a few semiskilled jobs such as painting and spot welding. As with robots on the factory floor, such autonomy requires sophisticated intelligence gathering and communication systems, which simply were not available before.

Towards the end of the 1970s, the promise of increased computing power in only a few years, prompted many countries to invest heavily in second-generation robotics research in the hope that the implementation of such

technology would help to pull them out of recession. In addition, plans by the world computing community to develop fifth-generation computers within a decade, stimulated research into artificial intelligence and related topics. So, due to a combination of historical factors, in the early 1980s there was a tremendous boost given to robotics work in general, and to industrial robotics in particular.

Nevertheless, this must of course also be seen in the more general context of the overall technological revolution started in the 1970s, largely based on the integrated circuit. This was producing consumer products such as miniature calculators, personal computers, video games and digital watches. Thus every 'man in the street' was directly affected by the new high technology, and there was consequently a high level of awareness even among the more conservative professions that something important was happening. Information technology (IT) became a catchphrase as computing techniques infiltrated even offices of traditionally 'low-tech' firms in the form of word processors and electronic mail. So, to an extent, robotics was able to take advantage of this new awareness of high technology, and enter markets which might otherwise have required substantially more persuasion about potential benefits.

2.1.2 ROBOTICS WORLDWIDE

Despite the origination of robotics in the West, the country which undoubtedly took most advantage of first-generation robots was Japan. One of the factors involved in this may have been that, in contrast to Western practice, many members of Japanese firms tend to have a 'job for life', so that management may have felt able to make longer term, longer payback, higher risk plans involving the use of robots than their Western counterparts would have felt were 'safe'. Similarly, the introduction of robotic technology held no fear for the Japanese workforce because they were still guaranteed a job somewhere in the factory.

Some Japanese claim that there is a historically based team spirit prevalent in Japan. Whatever the validity of this belief however, there is the undeniable success of far-sighted consistent Government support for robotics, and of the Japanese ability to take maximum advantage of other countries' inventions. The first Japanese company to become heavily involved in robotics was Kawasaki - yet that was as late as 1968, a decade behind the US lead.

Although, owing to the different definitions employed by various countries, it is very difficult to compile accurate worldwide surveys of 'robot populations' in different areas, many attempts have been made, from which an overall picture emerges. Even by 1970 there were only a few hundred robots throughout the whole world, with 200 in the USA and 150 in Japan. By the mid 1970s, however, the world population was closer to 4000, and by 1984 was about 37,000. At the present time, Japan is undoubtedly in the lead with more robots installed than any other country, even when the differences in definitions are taken into account.

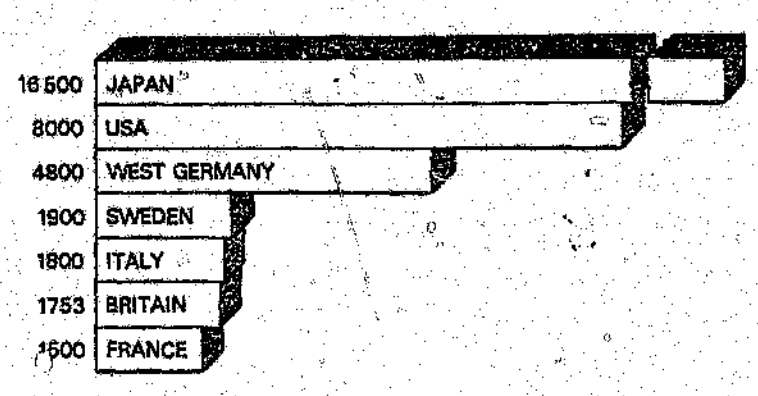


Fig 2.1 Estimated 1984 world robot population [1]

Although Japan undeniably leads the world in actual numbers of installed robots, in robotics research they seem to have little or no superiority. Among the world leaders in research should be included the USA, UK and West Germany. Similarly, there are many large robot manufacturers outside Japan, some of which have specialised in the field for

several years. Recently there has been a proliferation of new robot manufacturers and it is debatable how long the market will sustain such a wide diversity of suppliers. In many countries, just a few firms may supply over 80% of the robotics market, leaving only a 20% share for all the other suppliers to fight over. Many predict a substantial 'shake-out' in the industry, leaving a comparatively small number of large manufacturers (with the required combination of product technology, systems and applications expertise and marketing skills) to serve the whole world market. In addition, there is a growing tendency for large companies to attempt to build up a 'robotic presence' by taking over some existing robot producers and forming licence agreements with others. Such licensing agreements are only likely to be a short term measure. Indeed it is already being predicted that in only a few years many large manufacturers who once joined with technically skilled producers, will once again break away from such overseas partners.

Of all the current worldwide industrial robot applications, until now almost all have employed first-generation technology, and the most common uses have been for surface coating, spot welding, parts handling and machine servicing. Recently, however, it is noticeable that worldwide there has been a dramatic increase in the application of robots for arc welding and assembly, tasks which largely require second generation robots.

Spot welding, on the other hand, looks likely to saturate the automotive market in the near future, so that the number of new such installations is likely to drop substantially towards the early 1990s. Even relatively cheap and simple teaching robots are finding increased application outside the educational environment, as some of the more sophisticated versions become sufficiently advanced, some believe, for limited application in commercial use.

2.1.3 CURRENT STATUSES

The prevalent trends that have dictated robotic development over the past few decades, are still with us today. Computers are still the dictating factor in robot sophistication, and manufacturing trends and requirements are moving still further towards a desirable small batch, high variety, environment, a situation that necessitates computer control in its purest form, Computer Integrated Manufacture (CIM).

The power of computers has now reached a level that allows them to be easily programmed to perform even the most sophisticated control tasks. It remains, merely, to interface this power to the mechanics of the robot, in the cheapest, simplest, and most effective manner, in order to gain this control. It should be noted that, in all previous developments, this link-up has been performed via a dedicated robot controller (an enhanced input/output device with computing power).

This seems to be an undesirable sidetrack into which robotic research has been unwittingly led. In the 1970s, what little computing power there was, was being used to its full advantage in flexible interfacing with the robot hardware, somewhere along the line, however, robot controllers started to become more dedicated and thus their capabilities were restricted to their exact requirements. This step was probably taken in order to minimise the cost of the controllers, previously the prohibitive factor, which has now resulted in a situation in which the integration of new computing technology has been increasingly difficult to implement due to the redesign involved in the controllers. It would thus be desirable to 'steer' control back onto the right path by redesigning controllers from scratch and staying in the path laid out, by other research areas, and dictated by computer development.

2.2 ROBOT CONFIGURATIONS

The versatility of a robot is defined by its degrees of freedom. It may possess any combination of these six which are illustrated in the following diagram.

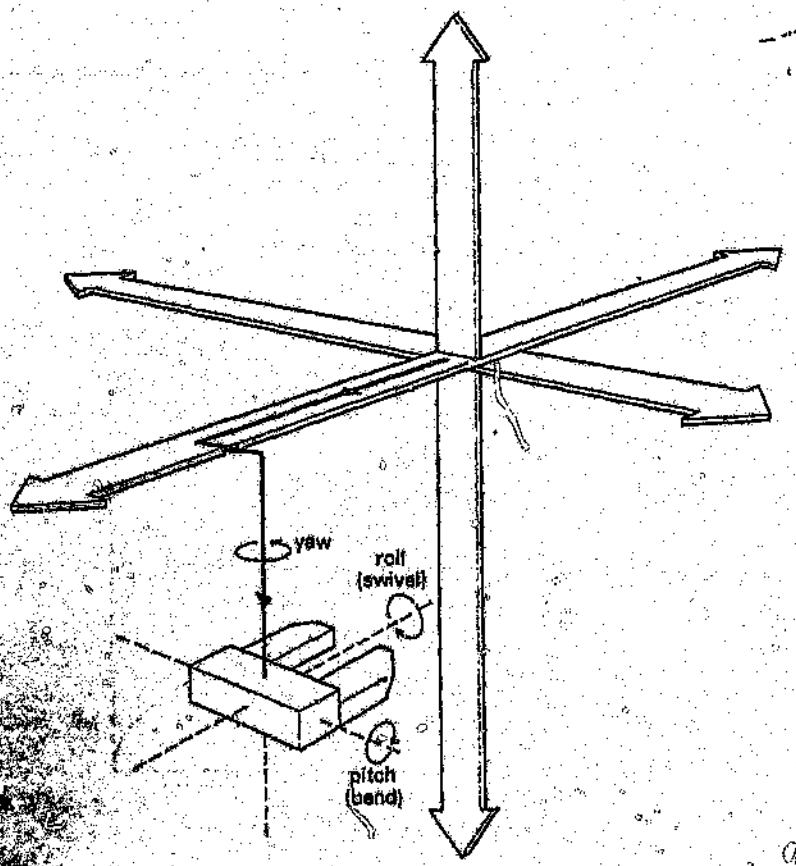


Fig 2.2 The six robot degrees of freedom [1]

The linear degrees are obtained from base, shoulder, and elbow movement, whilst the other three are wrist characteristics. The large wrist movements that robots are capable of, often makes them more flexible than the human arm. Often gripper closure and base linear movement are cited as further degrees of freedom, but strictly speaking they are not.

Robots come in various configurations, characterised by their joint arrangement and availability. With increased

computing power, others are becoming more feasible and are joining this recognised group. Although the configurations suit specific coordinate systems, they may be addressed in any form as the controller is capable of translating inputs into the more convenient system. Each has a characteristic work envelope, as shown in the following figures:

2.2.1 CYLINDRICAL CONFIGURATION

(ROTATION-TRANSLATION-TRANSLATION)

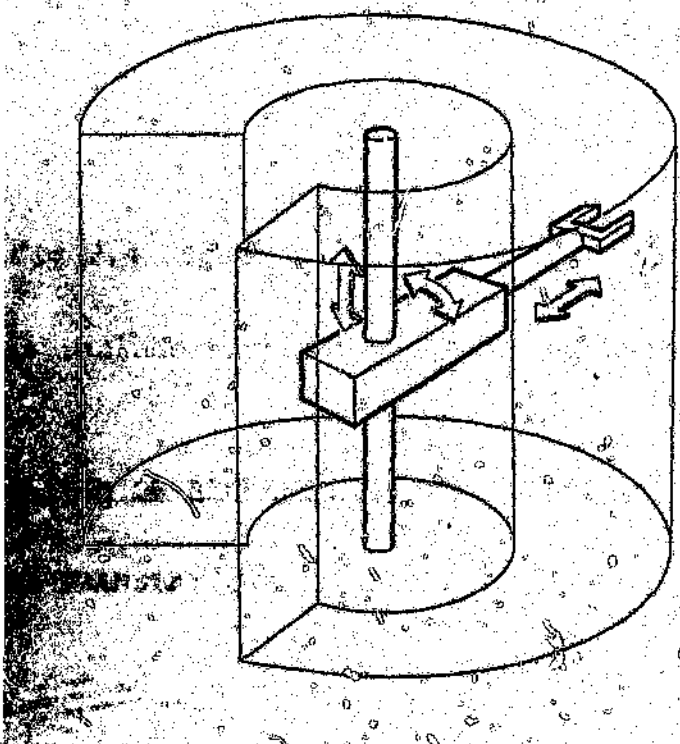


Fig 2.3 Cylindrical coordinate robot [1]

The main axis can be orientated horizontally or vertically allowing for lateral movement or lifting applications.

2.2.2 SPHERICAL OR POLAR CONFIGURATION

(ROTATION-ROTATION-TRANSLATION)

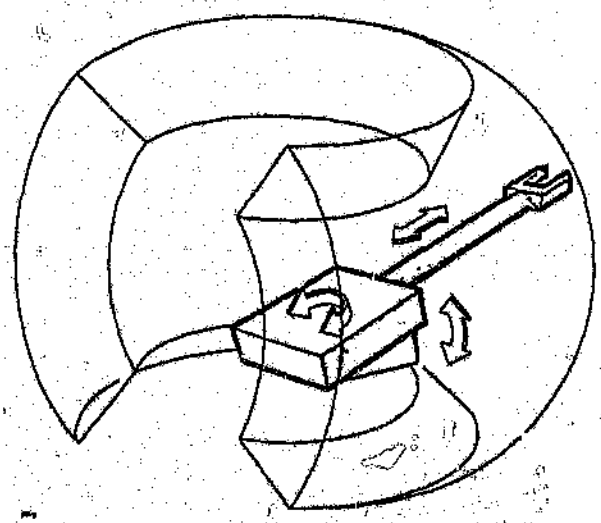


Fig 2.4 Spherical coordinate robot [1]

Applications include extensive use in the automotive industry.

2.2.3 CARTESIAN OR RECTANGULAR CONFIGURATION

(TRANSLATION-TRANSLATION-TRANSLATION)

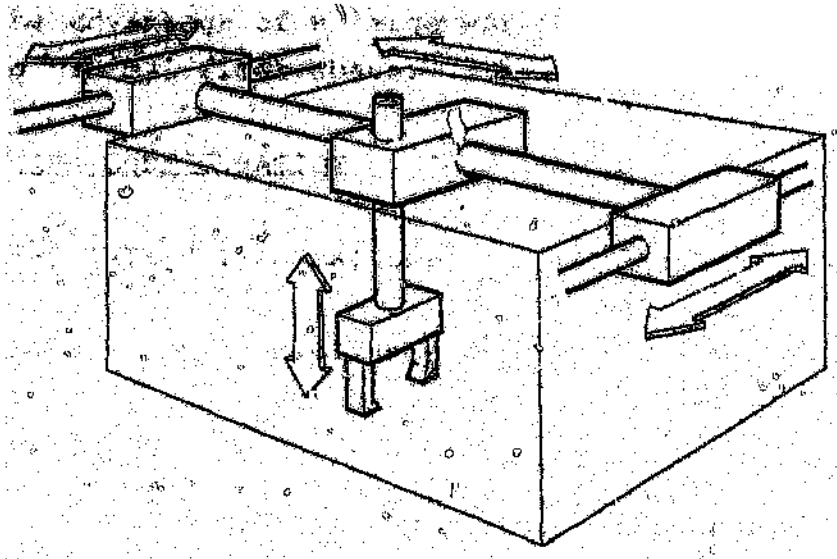


Fig 2.5 Cartesian coordinate robot [1]

The rectangular configuration is most suitable for assembly tasks due to its inherent high mechanism precision. It is also the strongest and most stable configuration lending itself to high load applications.

2.2.4 REVOLUTE OR ARTICULATING CONFIGURATION

(ROTATION-ROTATION-ROTATION)

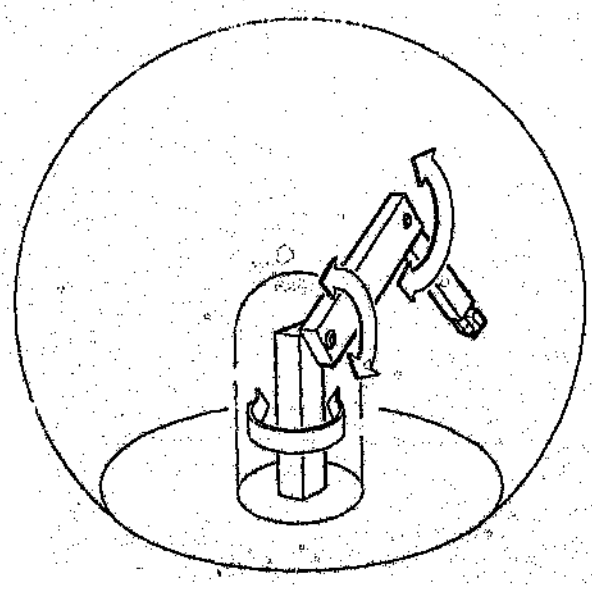


Fig 2.6 Revolute coordinate robot [1]

It has the advantage of a large working envelope whilst utilising minimal floor space.

2.2.5 SCARA CONFIGURATION

(ROTATION-ROTATION-TRANSLATION)

The SCARA (Selective Compliance Assembly Robot Arm), is similar to the articulating configuration. It is a fundamentally new structure and exhibits properties of both revolute and cylindrical coordinate systems.

(See figure 2.7 on page 19)

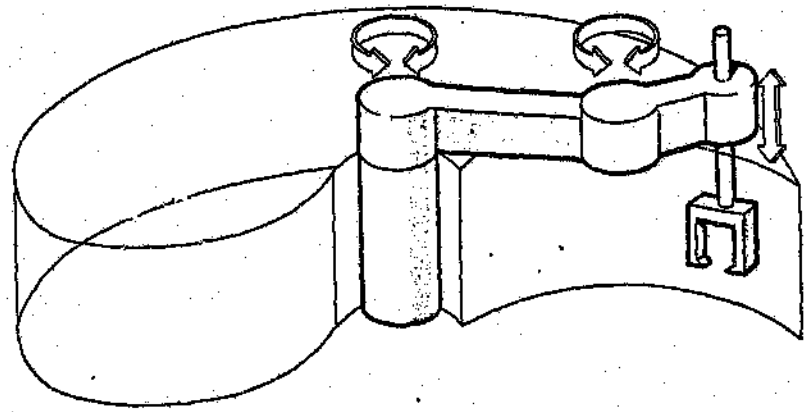


Fig 2.7 SCARA - type robot [1]

2.2.6 PARALLEL CONFIGURATION

(PARALLEL TRANSLATIONS)

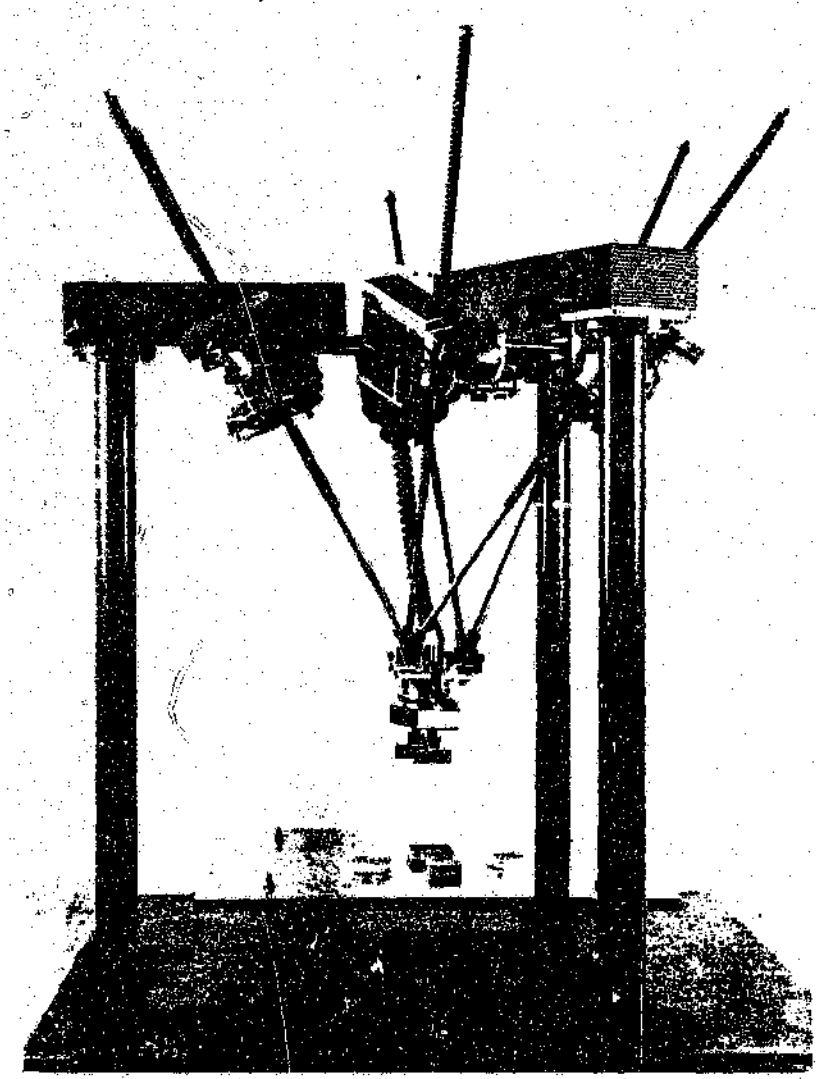


Fig 2.8 A parallel robot

2.3 ROBOT MOTIONS

Robot motion takes on various degrees of sophistication. These degrees can be broken down into two basic categories as follows:

2.3.1 POINT-TO-POINT (PTP)

There are fundamentally three different ways of controlling PTP motion in a robot, but each makes use of independent position servos for each joint, and for each, the actual path followed by the robot arm between taught points is not of concern.. As a result, the robot controller does not need to employ a full kinetic model. The simplest form of control is sequential joint control (robots with this form of control are often termed pick-and-place devices) in which one joint at a time is activated, while all the others remain stationary. The resulting path which the end-effector of the arm follows is likely to zig-zag across the work envelope, and consequently the time taken to move from one taught point to another tends to be far longer than necessary. Nevertheless, sequential control requires the simplest of control structures, and may also suit highly modular systems where individual joints may subsequently be replaced by multijoint manipulators.

With uncoordinated joint control, although all joints move at once, the fact that one joint has completed a given fraction of its travel does not imply that the other joints have covered a similar fraction of theirs. Each joint moves through its required path in its own time, and then waits for all the others to finish. Owing to the inherent lack of coordination between the different axes, it is difficult to predict the path and velocity of the end effector as it travels between the taught points. With terminally-coordinated joint control, however, the individual joint motions are timed so that they all start and stop together, making it the most convenient, although the most expensive, of the three methods.

This design shows a radical departure from configurations. Once again it is a recent development and, to date, has not been exploited extensively in industry. It consists of three pairs of rods with free swiveling bases, as the rod lengths are varied, the end effector moves through all six degrees of freedom. It has a small, spherical, working envelope, but is extremely light, fast, and accurate.

2.3 ROBOT MOTIONS

Robot motion takes on various degrees of sophistication. These degrees can be broken down into two basic categories as follows:

2.3.1 POINT-TO-POINT (PTP)

There are fundamentally three different ways of controlling PTP motion in a robot, but each makes use of independent position servos for each joint, and for each, the actual path followed by the robot arm between taught points is not of concern.. As a result, the robot controller does not need to employ a full kinetic model. The simplest form of control is sequential joint control (robots with this form of control are often termed pick-and-place devices) in which one joint at a time is activated, while all the others remain stationary. The resulting path which the end-effector of the arm follows is likely to zig-zag across the work envelope, and consequently the time taken to move from one taught point to another tends to be far longer than necessary. Nevertheless, sequential control requires the simplest of control structures, and may also suit highly modular systems where individual joints may subsequently be replaced by multijoint manipulators.

With uncoordinated joint control, although all joints move at once, the fact that one joint has completed a given fraction of its travel does not imply that the other joints have covered a similar fraction of theirs. Each joint moves through its required path in its own time, and then waits for all the others to finish. Owing to the inherent lack of coordination between the different axes, it is difficult to predict the path and velocity of the end effector as it travels between the taught points. With terminally-coordinated joint control, however, the individual joint motions are timed so that they all start and stop together, making it the most convenient, although the most expensive, of the three methods.

2.3.2 CONTINUOUS-PATH

When the form of the actual path followed by a robot end-effector between two points is of primary importance, it is necessary for the controller to employ a kinematic model if the intermediate movements are to be interpolated, not all taught. It must then simultaneously move each axis the minimum distance necessary to reach the next intermediate point in space required to keep the end effector on a controlled predictable path. Such sophistication usually requires a 16 bit microcomputer as the controller, a characteristic of the new AT range of computers and something previously restricted to mini computers. In addition to a robot following a path of closely spaced points that it has actually been taught, it will then be possible to request that linear interpolation be performed between only two widely spaced points (or even 'fitted' through more than two points), resulting in an untaught straight line motion. With some controllers, circular interpolation is also possible. For welding robots it is possible to superimpose a weaving pattern over the basic path.

There are various approaches to continuous-path control, each using different amounts of information about the actual path to be followed. The simplest method is the straight forward servo-control approach, which makes no use of any knowledge about where the path goes in the future. Although details of the path may be stored in the robot's memory, all the controller refers to when driving the arm motors is the error signal indicating the difference between where the arm actually is, and where the next intermediate point is that it is heading for. It is this method of continuous-path control which is most common in present industrial robots.

More advanced systems make use of such approaches as preview control (or feed forward control). This method takes into account the way the path changes immediately in

front of the end-effector's current position. A still more advanced approach employs path planning (or trajectory calculation), in which knowledge of the whole path to be followed is incorporated into a mathematical model of the robot arm and the load it is carrying, and a detailed acceleration profile for each joint is computed, together with predictions for the required motor control needed to make the arm follow the desired path. Such a dynamics based approach can incorporate the effects of forces such as gravity, inertia, damping and friction, and it permits highly accurate movements at speed which would otherwise be impossible.

The use of transformations allows translations and rotations to be easily performed, not just on points but on whole programs. However, the complexity of the computations required can sometimes result in intolerable speed reduction or inaccuracy. Continuous-path control with such systems involves applying an interpolating function to the transformed world-coordinate locations, and then, in real time, rapidly transforming them into joint positions which the robot can understand. This must be continuously kept up so that the motor servos have a constant stream of new intermediate points to head for.

A particularly useful application is to provide a full tracking capability that allows the robot to be synchronised with the motion of a conveyor belt (eg. for spraying or welding a passing car chassis). Conventionally the robot would be taught its task with the object moving in front of it, and so long as the synchronisation of the robot and the object can always be maintained in the future, and so long as the conveyor speed never varies, this method will be perfectly adequate. However this is not always the case, especially with technological advances allowing for improved and varying line speeds. Where this constant, uninterrupted conveyor speed cannot be guaranteed, it is possible to teach the robot with the

object stationary, and for the robot then to use coordinate transformation to perform its task relative to the object, and by monitoring the conveyor speed, to be independent of the motion of the transfer line.

object stationary, and for the robot then to use coordinate transformation to perform its task relative to the object, and by monitoring the conveyor speed, to be independent of the motion of the transfer line.

2.4 ROBOT CLASSIFICATIONS

In addition to identifying robots according to their arm configuration, various other terminologies are commonly used.

2.4.1 FIXED/VARIABLE SEQUENCE ROBOTS

Pick and place devices, although strictly speaking not robots at all, nevertheless are often referred to as fixed sequence (or limited sequence) robots. The stroke of each axis of motion is determined by adjusting mechanical end-stops, and sensors come typically in the form of limit switches which can only sense the end points, and none of the points in between. Such devices cannot be reprogrammed to execute a new task, but must instead be reset and adjusted as would a traditional automatic machine.

This is in fundamental contrast to true robots (variable sequence robots), which can immediately execute a different task or sequence merely by running a new program. To some extent however, there are a few pick-and-place type devices which can now switch in various different end-stops under program control, so creating a grey area in the definition. In addition, of course, there is always an incentive for industry to refer to pick-and-place devices as robots, as this sounds far more glamorous and progressive.

2.4.2 SERVO/NON-SERVO ROBOTS

With variable-sequence robots it is necessary to be able to stop a particular joint of the arm at any point along its travel. There are basically two approaches to this problem. With the simplest method the controller merely sends power to the joint for as long as it estimates the arm will take to get to the desired position. Although with certain special electric motors (stepper motors) this approach can sometimes be satisfactory, on the whole such

open-loop control with no feedback of information about the joint's actual position is grossly inaccurate - the arm might have got stuck and not have moved at all. Consequently, all but educational robots (or teaching robots), make use of a second method, which involves placing a servo-mechanism on each joint which effectively checks both the position of the joint and the position in which the controller wants the joint, and then moves the arm until the positions coincide. Robots employing such closed-loop control are termed servo-controlled robots or simply servo-robots.

2.4.3 POINT-TO-POINT/CONTINUOUS PATH ROBOTS

A common distinction is drawn between two different types of controller used in industrial robots. Many of the earlier robots only had sufficient computer memory to store discrete points in space which the arm had to move to, and [illegible] the movements between those points the path of the [illegible] was not defined and often difficult to predict. Such [illegible]-to-point robots are still very common and are [illegible] adequate for such tasks as spot welding. As the [illegible] memory has come down, so the number of points which [illegible] stored has increased, and many manufacturers use the [illegible] multipoint control if a very large number of discrete [illegible] can be stored.

[illegible]

[illegible] some tasks, such as paint spraying and arc welding, it is necessary for the path followed by the robot to be controlled at all times. Such continuous path (CP) robots in reality approximate a continuous line by splitting the path up into a very large number of separate points very close together. The positions of these points are either recorded during programming, or are calculated during the actual movement by filling-in (interpolating) between, for example two points to produce a straight line. These robots can be thought of as a natural extension of point-to-point systems, and there is in fact a 'grey area' in which multipoint control systems can approximate a

continuous path system by not stopping at each discrete point but merely passing through them.

2.4.4 FIRST/SECOND/THIRD GENERATION ROBOTS

The first-generation robots are generally considered to be those 'deaf, dumb, and blind' robots which are currently most common on the factory floor. Second-generation robots have been around for some time in laboratories and are already in a few factories. Such robots, which may often look the same as their first-generation counterparts, can make use of varying degrees of sensory information about their immediate environments to modify their behaviour during a task (corresponding to the most sophisticated of the six classes of the JIRA definition of 'robot' mentioned earlier). Typical sensors include vision systems and tactual systems (which provide a 'sense of touch').

Some people have called second-generation robots 'intelligent robots', but this term should really be reserved for the still more sophisticated third-generation [illegible]. In contrast, current research is really only [illegible] 'common sense' robots. Nevertheless, [illegible] indeed leading to so-called intelligent robots [illegible] be equipped both with senses and the ability to [illegible] and have some understanding of objects in the [illegible] world, and so to an extent, to have the capability [illegible] thing of their own accord.

With all these definitions, there are 'grey areas' - one simple sensor does not make a second generation robot, it is necessary that the sensor(s) significantly affect the robot's operation. On top of this, even the accepted definitions vary: some authorities insist that the first generation was pick-and-place devices. It may well be that eventually only second-generation robots upwards will be considered 'true robots', with the first-generation being thought of as 'programmable devices'.

2.5 COMPUTERISED MOTION CONTROL

Whatever the configuration of a particular robot, if any computation is to be performed involving its given structure, then mathematical techniques must be employed which permit the description of the varying locations of the different robot-arm joints. Such techniques allow the absolute position and orientation of the robot end-effector to be determined from the relative positions of the robot joints, and (still more difficult) the necessary joint locations needed to place the end-effector in a required position. These sorts of computation are necessary for sophisticated control of the robot arm.

Each rigid section of a robot manipulator arm can be thought of as having its own private coordinate system locked around it. In this way, it is only necessary to specify the location and orientation of that coordinate system relative to some base system to in fact specify the location and orientation of that section of the robot arm. Assuming that the coordinate system used is cartesian (although this approach can be modified to accept other systems), then the figure below might represent the location and orientation of the robot-section under consideration (for example the end-effector) with regard to the chosen base system.

The base system might, for example, be chosen so that the origin was at the centre of the base of the robot, with the X-Y plane horizontal, and the Y axis pointing away from the front of the robot base. The equivalent coordinate system around the required section of the robot arm can be thought of as rotating within the base system. To actually describe the position of this second system in the figure, much can be achieved merely by specifying the direction of the vector from the origin Ob, the base system to the origin Or of the other system, however, this will not provide any information about the orientation of the second

system. Mathematically, orientation can in fact be specified by using a matrix consisting of three rows of values. The three values in a given column in effect represent the X, Y and Z orientations of a particular axis of the second system, as if it were a straight line in the base system. By specifying in this way orientation for each of the three axes of the second system (taking three columns of the matrix) the orientation of the whole of the second system can be uniquely specified.

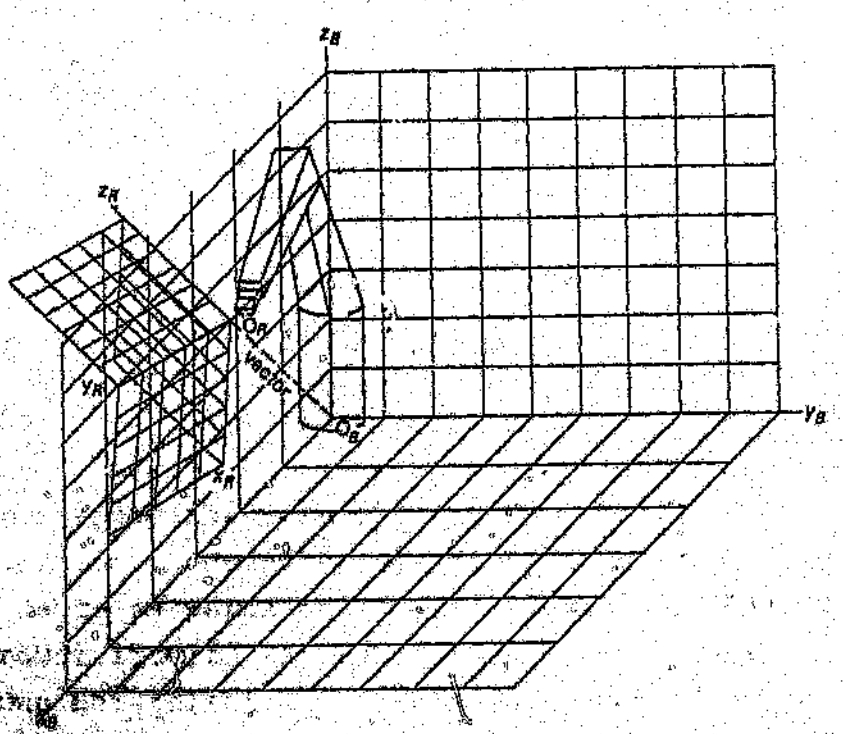


Fig. 2.2. The coordinate system of a robot end effector [1]

By combining the information in both the position vector and the orientation matrix it becomes possible, in principle, to start with the coordinates of a point in the system locked round the section of the point under consideration, and then transform them into the equivalent coordinates of the base-system. In practice this is performed by using a homogeneous transformation which consists of a matrix of four rows of four columns, easily constructed (as shown in the following figure) by writing the orientation matrix next to the position vector, and then writing 0 0 0 1 underneath them.

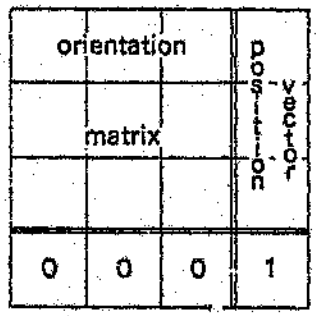


Fig 2.10 Construction of a homogeneous matrix [1]

In this form the transformation can be accomplished merely by multiplying the coordinates of a point in the second-system by the homogeneous transformation matrix using matrix multiplication.

Clearly, in practice it would not always be convenient to specify a set of secondary coordinate systems all in terms of the base system. For a start, robots do not consist of a series of independent sections 'floating' in the base system - the sections are all linked one to another in a serial fashion. Secondly, descriptions of positions of objects (such as parts on a pallet) which are not connected to the robot itself, may remain stationary relative to each other, but not relative to the base system.

To take account of such factors, homogeneous transformations can be employed to describe the positions and orientations of two coordinate systems relative to each other, both of which are different from the base system itself. In this way, the parts on the pallet, for example, could each be described, once and for all, relative to the pallet. As the pallet was subsequently moved, only the transformation relating the pallet to the base system would need to be changed. By employing the relative transformations which described the relation of the objects to the pallet, the absolute coordinates of the objects could be determined comparatively easily wherever the pallet was moved.

By stringing a series of relative transformation calculations together in this way, it is possible to work along a 'chain' of different systems in which only the relative positions and orientations of neighbouring systems are known. Of course, the most important chain is usually that of the robot arm itself. By using relative transformations, it is possible to work outward from the base of the robot (fixed relative to the base of the system) through each joint in turn, until the absolute position (with respect to the base system) of the end-effector itself is determined.

As mentioned under the robot degrees of freedom, in addition to the linear degrees of freedom, there are an additional three rotational degrees. These are commonly referred to as roll, pitch and yaw, terms also used in aviation. When relative orientations are specified in this way, the three rotations are commonly referred to as Euler angles.

Of course, in specifying the three Euler angles, the sequence of rotations is most important. Imagining a glider, a 90^{O} clockwise roll, followed by pulling the nose up by 90^{O} (pitch) will result in a dramatic turn to starboard, leaving the glider heading still in a horizontal plane (although on its side) at right angles to its original course. On the other hand, a 90^{O} pitch followed by a 90^{O} roll will leave the glider heading straight upwards. As a result of this importance of sequencing, by convention, the sequence of Euler angles is taken as being roll-pitch-yaw. Using these angles, it is in fact easy to construct an appropriate 3 x 3 rotation matrix for inclusion in a homogeneous transformation matrix.

Nevertheless, unfortunately, there is in fact no universally accepted convention regarding which angles the term 'Euler angles' refers to. Although we have been using the angles corresponding to the sequence roll-pitch-yaw, another commonly employed approach is to measure the angles

corresponding to the sequence roll-yaw-roll. Both offer the ability to orientate the gripper at any desirable orientation, and both can be employed for constructing homogeneous transformation matrices.

When considering the actual design of a robot wrist, building three articulations mutually at right angles, as would be needed for a roll-pitch-yaw approach is frequently more difficult that employing a rotating wrist onto which is attached a yaw-roll arrangement. However, design consideration must also be given to those occasions on which two joint axes align with each other; a rotation about either axis then results in identical motion of the end-effector. In this situation the robot arm has lost one of its degrees of freedom, and is called degenerate.

When we consider the movements of a robotic manipulator-arm without any reference to force (that is, the kinematics of the arm), we can treat the manipulator as being composed of a series of individual sections, linked together with a particular kind of joint. The joints in such a serial-link manipulator can be either revolute joints or prismatic joints, and the links in effect maintain a fixed relationship between the joints which can then be modelled. A single revolute (R-type) joint can only rotate about one axis, and it is the angle between the joint in question and the next which varies. With a prismatic (P-type) joint, it is the distance between the joint and the next that varies. The sequence of joints and links is known as a kinematic chain.

Using such a model, it is possible to construct a separate coordinate frame around each joint. Naturally, a consistent method of assigning such frames to each joint must be adopted, and a common approach is the Denavit-Hartenberg convention which prescribes a sequence consisting of a rotation, followed by two translations, followed by a further rotation, to bring any one coordinate frame into exact coincidence with the next. From the

values of the rotations and translations needed to do this, one can easily derive the homogeneous transformation which describes the relative position and orientation between the two coordinate frames.

Commonly, the homogeneous transformations which describe the relations between adjacent links on a manipulator are called 'A' matrices. A given 'A matrix', in other words, is simply a description of the change in orientation and position between that link and the next. Thus, matrix A1 refers to the position and orientation of the first link of the manipulator, while A2 refers to the relative transformation between the coordinate frame of the first link and the second link. As explained earlier it is possible to work along the kinematic chain, multiplying all the relative transformations (A matrices) together, to obtain the absolute coordinates (in the base system) of any particular link. In other words, the position and orientation of the third link, in base coordinates, is obtained by multiplying (using matrix multiplication) A1, A2 and A3 together.

Such products of A matrices are commonly called T matrices, and, for a six link (i.e. six degrees of freedom) manipulator, the absolute location of the end-effector (on the sixth link) would be given by the T matrix:

$$T6 = A1.A2.A3.A4.A5.A6$$

So, knowing the individual A matrices for each link at a given time (in other words a given arm configuration) it is possible to calculate the resultant position of the end-effector (which in practice is frequently the only part of the robot for which position and orientation information is of direct concern). Indeed, it is also possible after performing such a series of transformations, to extract from the resultant T matrix the orientation of the end-effector in terms of Euler angles. For practical robot work this can be of particular benefit.

Frequently in robotics the most important kinematic calculation is actually to obtain the individual joint positions of a robot arm, given the absolute position of the end-effector (T6). In other words, we know the current position and orientation (the pose) of the end-effector as well as where we want it to end up. What is required is a list of all the new joint positions necessary to result in the desired new position of the end-effector. The solution can be of vital importance for controlling the arm, yet deriving the form of such a solution for, say, a six-axis arm is a far from trivial task.

There is no algorithm by which the appropriate kinematic equations needed to solve the problem can automatically be derived and, generally, geometric intuition is needed to determine the solution. Nevertheless, once satisfactory kinematic equations have been constructed for a particular arm design, it is then straight forward to compute any required arm configuration by simply substituting into the explicit equations. Naturally, although there is only one end-effector pose, T6, corresponding to a particular set of joint positions, there may be more than one possible arm configuration which results in exactly the same end-effector pose. For instance, with a conventional anthropomorphic jointed-arm robot, the same gripper pose might be obtainable with the 'elbow' either up or down as shown in figure 2.11. Usually, however, it is only in fact necessary for one of these alternatives to be calculated.

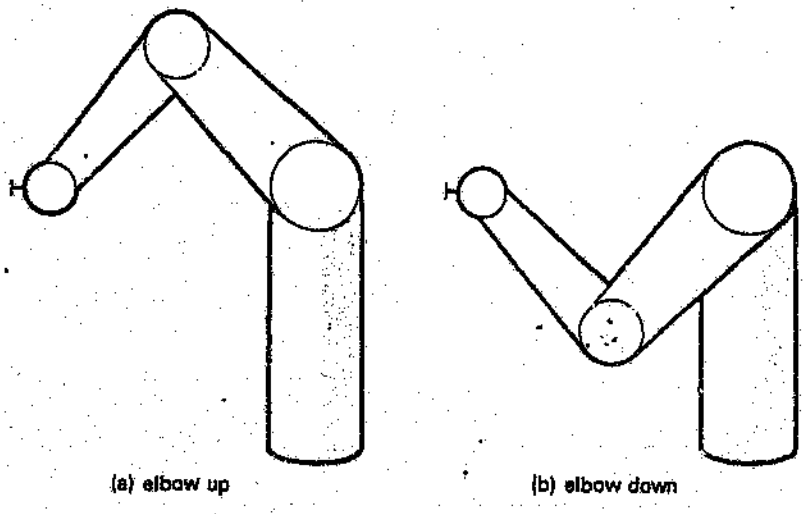


Fig 2.11 Two arm configurations for identical end effector positions [1]

Dynamic load considerations have not been dealt with in this section as they are fundamentally an aspect of design with which we are not concerned. They would, however, be analysed using Lagrangian mechanics which would yield a system of matrices comprising several thousand terms. These can be reduced by identifying the effective inertia relationships (relationships between torque and acceleration in joint) which would then be used in a manner which could be quickly manipulated for advanced control purposes.

2.6 ROBOT CONTROLLERS

Physically a robot consists of three components: the power supply, the mechanical unit, and the controller [4]. The controller has a threefold function; first, to initiate and terminate motions of the manipulator in a desired sequence and at desired points; second, to store position and sequence data in memory; and third, to interface with external devices. The controller is essentially the controlling hardware of the robot. Robot controllers can be step sequencers, pneumatic logic systems, diode matrix boards, electronic sequencers, microprocessors, or minicomputers. The complexity of the control determines the capabilities of the robot.

The robot industry is constantly seeking how it can advance its products to increase the level of value-added it can offer the end user [5]. There is not much more that can be done to the present range of robots apart from making the machines more accurate, faster, lighter and cheaper. But an area receiving considerable attention is controllers. Controller manufacturers will be looking for niches in the market place - trends they can identify and exploit. Noises are being made that to continue to have one controller for each robot in a multi-robot cell could be over-kill, especially if one considers the costs involved. Firms could save a great deal by having a single controller for three or four robots. These controllers could possibly be more expensive than a single controller, but they could offer potential savings in multi-robot installations.

The size and shape of a robot controller in relation to a typical robot is shown in figure 2.12.

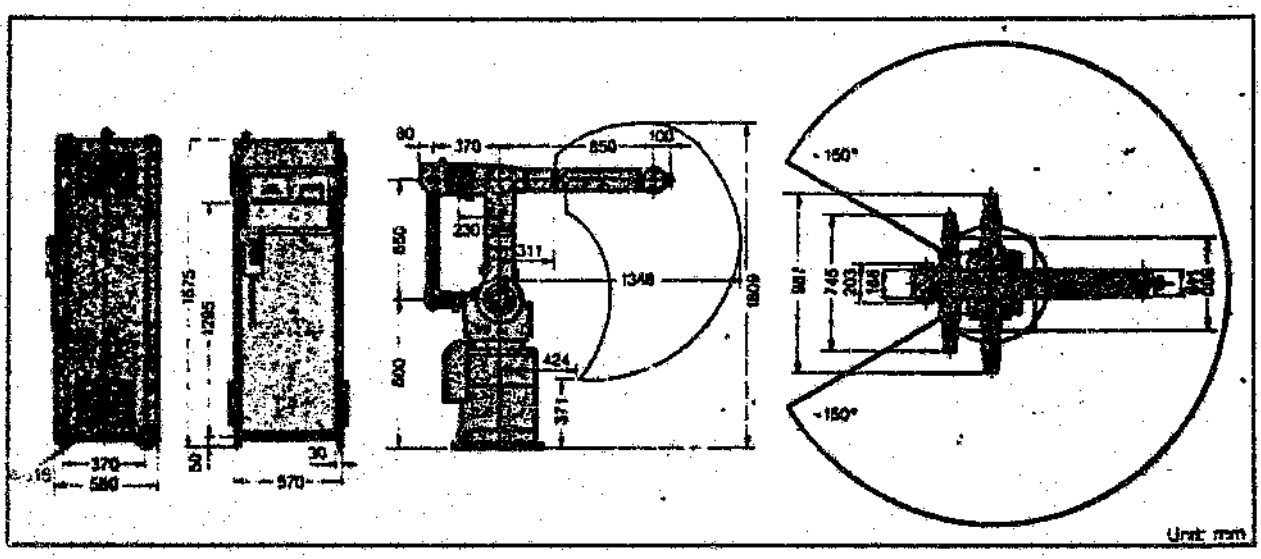


Fig 2.12 A typical robot and robot controller [5]

2.7 PLC CONTROL

2.7.1 INTRODUCTION TO PLCs

The PLC is another form of hardware control, although not in the robotics field, as far as can be ascertained. Many articles in the technical press these days mention programmable logic controllers (PLCs) [6]. If one is familiar with relay control, then PLCs are easy to understand.

Relay control is a collection of contacts, coils, switches and input and output (I/O) signals. This is how plants and machines used to be controlled. If all these relay elements are replaced by one standard unit, which is then connected only to the input and output signals, and easy to control functions are added, then the resulting system is called a PLC.

Firstly, the I/O signals must be divided into digital and analog signals. A controller is therefore a method of taking input signals, combining them according to some sort of logic, and then transferring the result to the output signal. Input signals come from pushbuttons, limit switches, pressure switches and temperature sensors. Output signals are sent to contactors, motors, clutches, valves and even pilot lights.

The PLC is a standard mass produced unit, and smaller in size than the older relay systems. The method of combining the signals is written into the program memory via a keyboard. Instead of wiring up a cabinet full of relays, all that is needed is to type in a few statements. It is even easier to duplicate this process if more than one application is required, simply by copying the program to another PLC.

The heart of the PLC is a microprocessor (MPU), which does the calculating and the controlling of all the signals. To

2.7 PLC CONTROL

2.7.1 INTRODUCTION TO PLCs

The PLC is another form of hardware control, although not in the robotics field, as far as can be ascertained. Many articles in the technical press these days mention programmable logic controllers (PLCs) [6]. If one is familiar with relay control, then PLCs are easy to understand.

Relay control is a collection of contacts, coils, switches and input and output (I/O) signals. This is how plants and machines used to be controlled. If all these relay elements are replaced by one standard unit, which is then connected only to the input and output signals, and easy to control functions are added, then the resulting system is called a PLC.

Firstly, the I/O signals must be divided into digital and analog signals. A controller is therefore a method of taking input signals, combining them according to some sort of logic, and then transferring the result to the output signal. Input signals come from pushbuttons, limit switches, pressure switches and temperature sensors. Output signals are sent to contactors, motors, clutches, valves and even pilot lights.

The PLC is a standard mass produced unit, and smaller in size than the older relay systems. The method of combining the signals is written into the program memory via a keyboard. Instead of wiring up a cabinet full of relays, all that is needed is to type in a few statements. It is even easier to duplicate this process as far as to include application is required, simply by [illegible] and producing another [illegible]ds [8], but only a few applications are [illegible]e.

[illegible]he PLC is a [illegible]

do this it uses a list of statements that have been typed in, which is known as the user program. It also has other statements which are loaded when it is made that handle the management of the whole device, and save the user from doing all the work. These user programs are stored in memory, and consist of a number of single control statements. The capacity of the memory, though, is limited. The statements define exactly how the incoming signals must be combined to achieve certain output values. After processing the last statement the program starts at the beginning, continuing in this cycle indefinitely.

There is another part of the memory that stores any intermediate results. At the start of the program cycle a snapshot of all the input signals is taken and stored in memory. These values are used for the all calculations, and the results are stored in memory too, until they are transferred to the outputs. The values of timers and counters are also stored in this memory.

The application of PLC control covers a very wide field. They are found wherever there is anything that needs to be controlled. The speciality of the PLC is the control of use by Boolean logic functions, and situations where the control logic must be changed often. Typical applications include: - processing machines;
- ventilation plants;
- galvanising plants;
- batching plants;
- textile machines;
- machine tools;
- packing machines;
- car wash plants. [6]

This list is not exhaustive, and goes as far as to include filling products into plastis tubs [7] and producing venetian blinds [8], but only a few applications are mentioned above.

2.7.2 CURRENT APPLICATIONS

A few applications where PLCs have recently been successfully implemented are detailed below.

Control in Foundaries
The Sheerness Steel plant on the Isle of Sheppey was designed to produce high yield, high quality but low cost steel. This plant has been on the forefront of computerised manufacture of steel since in was commissioned in the early 70's [9]. The recent installation of a dual integrator PLC system has enabled the company to achieve better control of its continuous casting process, integrating data between the plant floor PLCs and supervisory level computers.

Utilisation of the specific PLC concept to integrate all process and management functions is part of an overall strategy to eliminate plant floor manual data entry and make the whole system multivendor compatible, irrespective of hardware type.

Automation in Foundaries
Within 10 years the Intalco Aluminium Corporation has changed the complete operation of its plants, with over 70 PLCs controlling existing machines [10]. A manufacturer of a casting machine supplied the machine with a PLC, and once the PLC had proven itself on that piece of machinery it did not take long to find its way throughout the entire plant. It is not just the manufacturing equipment that is controlled by PLCs, but Intalco have extended this control to the plant's air and water pollution control systems too. Intalco insist on reliability, and for this reason all their hard-wired panels have been replaced with PLC systems.

Productivity Improvement
The experts at the Kingsford Products Company wanted to improve quality and the productivity of their charcoal

briquet plant in Burnside, USA [11]. Kingsford utilised the expertise of its own employees, calling upon Burnside's equipment operators and engineers to build a foundation for new automation systems and equipment. Management wanted to know from the operators what they needed to do their job best. The answer they received was that of better process control, including items such as information on feed rates, levels and temperatures. Installing PLC control into the process gave the operators greater control over the manufacturing, which inevitably led to an increase in productivity.

Energy Saving

It is not only in the plant environment that one would expect to see PLC control, but in almost any situation where control is needed. A conventional PLC was installed in engineering offices, and was used for controlling office lighting. This led to an energy saving of between 70 and 80% over conventional office lighting [12]. Typically, this demand is experienced in sales rooms, museums, underground parking and generally in most public or industrial buildings. The high energy savings are achieved by the optimization of light sources combined with a substantial reduction in illumination time.

RECENT DEVELOPMENTS

There has been a tremendous growth in the local demand for process control instrumentation over the past 3 years [13], with developments having to keep pace with this demand. A few recent developments are mentioned.

Combining PLCs, Computers and Workstations

There is a move afoot to make changes in the packing, architecture, and support of mid-to-high-level PLCs [14]. The driving force behind it is the desire for a higher-level industrial control application - a single integrated unit able to perform most, if not all, of the functions that mid-to-high-level control applications demand.

The new units combine the traditional capabilities of a PLC, an I/O interface, an industrial computer, and (when appropriate) an industrial workstation for man-machine interface. The control, communications, information processing, and operator interface functions are integrated into a single control platform.

Networking Capabilities

A major PLC manufacturer has developed a system whereby a PLC can be programmed with an IBM PC, using dedicated software in conjunction with a handheld programmer [15]. The system includes features such as timers, counters and maths functions.

Another PLC manufacturer has also developed a PC programmable PLC system [16]. The advantages of PC programming and menu-guided programming with 'help' functions, the facility for archiving and documentation, fast commissioning and testing, and the retrofitting of existing controls. There is also the advantage of having a teach-in mode and simple stand diagnostics.

Only a few have been mentioned, but manufacturers of PLCs are constantly developing smaller and more sophisticated PLCs that can only be to the benefit of the user. With over 200 PLCs available to the local user [17], there is no difficulty in selecting the PLC that suits his

The new units combine the traditional capabilities of a PLC, an I/O interface, an industrial computer, and (when appropriate) an industrial workstation for man-machine interface. The control, communications, information processing, and operator interface functions are integrated into a single control platform.

Networking Capabilities

A major PLC manufacturer has developed a system whereby a PLC can be programmed with an IBM PC, using dedicated software in conjunction with a handheld programmer [15]. The system includes features such as timers, counters and maths functions.

Another PLC manufacturer has also developed a PC programmable PLC system [16]. The advantages of PC programming are menu-guided programming with 'help' functions, the facility for archiving and documentation, fast commissioning and testing, and the retrofitting of existing controls. There is also the advantage of having a teach-in mode and simple standstill diagnostics.

Only 3 developments are menticned, but manufacturers of PLCs are constantly developing smaller and more sophisticated PLCs, that can only be to the benefit of the user. With over 20 PLCs available to the local user [17], there is no difficulty in selecting the PLC that suits his application.

2.8 CONTROLLING SOFTWARE

Since the introduction of industrial robots, the methodology for programming them has seen a great deal of improvement. In general, two mainstreams of software organisation have been pursued for the control of robots. The first is the 'explicitly programmed' system, which makes the user primarily responsible for every control action and requires explicit instructions. The second type is 'world modelling,' which makes the robot system responsible for knowing specific facts about the objects it works with. In this respect, explicitly programmed systems force the user to decide exactly what sequence of library routines the robot must use to perform a given task. World modelling systems make some of these decisions based on a more goal-orientated description of the task.

Most [illegible] states that software systems in industrial [illegible] explicitly programmed type. In these robot [illegible] different programming techniques are popular:

- manual lead-through programming;
- [illegible] programming;
- [illegible] programming.

Each technique offers unique application advantages over [illegible].

[illegible] mentions that robotics software can be thought of as ranging from 'robot software', which is neccessarily used within robots, through 'robot related software', which tends to expand the sophistication demonstrable by them, to 'robotics related software', which covers those areas relevant to both robot systems and others. As with all such classifications, it is not always clear into which group a given type of robotics software should be placed. One such grouping may look as shown in figure 2.13.

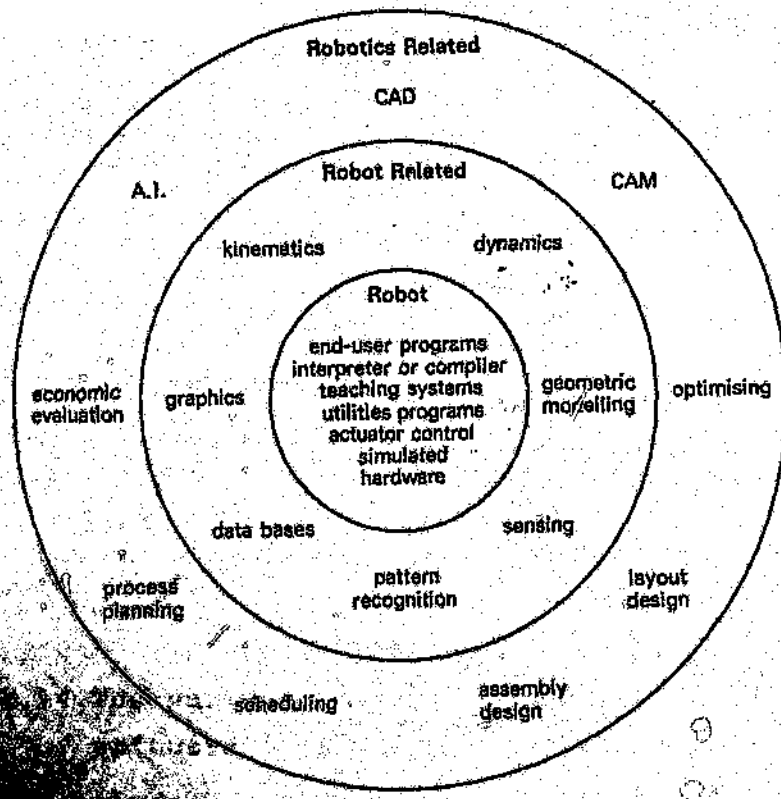


Fig 2.13 - Different forms of robotics software [1]

The software in a generalised robot configuration can be thought of as interfacing with two major parts of the system in addition to the programmer: the 'mechanics' which includes the robot manipulator, gripper and other such components, and the 'sensing devices', which provide the robot with information concerning its immediate environment. This is represented in figure 2.14 (see page

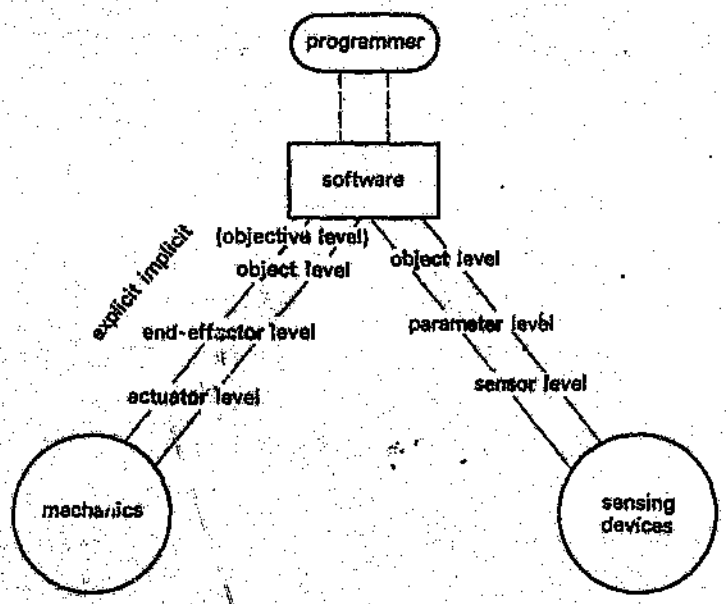


Fig. 3.14 The various levels of abstraction of user - software in a robotic system [1]

2.9 COMMON ROBOT PROGRAMMING TECHNIQUES: ON-LINE

2.9.1 MANUAL LEAD-THROUGH PROGRAMMING

This is a typical form of on-line programming. In addition to the basic components of a robot system, i.e. manipulator, control unit, and power source, this type of programming system requires a lead-through aid. This programming technique is very common in the robots used for spraying applications. The operator must manually lead either the teach arm, or the manipulator (with teach aid) through the motions, switching the spray gun on and off as required. The software organization for this type of robot system basically consists of an interpreter.

2.9.2 TEACH PENDANT PROGRAMMING

This programming system is the most common on industrial robots, and is another form of on-line programming. Besides the three basic components of the robot, this system includes a teach pendant. Using the teach pendant the manipulator is moved and the position is recorded by pressing a record button. Various parameters such as gripper state, time delays, output states, wait for inputs, and travel speed can be set and recorded. Unlike the manual lead-through technique which records the complete path, this teach technique records only the end states of the path.

2.9.3 ADVANTAGES OF ON-LINE PROGRAMMING

- This form of programming is very easy to use, and programming is made simple [2].
- The robot programmer need not be an expert in computer science, but must be skilled in the technical aspects of the task to be carried out, and thus shop floor expertise is utilised [3].
- The final program does not require testing, as the robot

will follow the exact path the programmer has led it through [3].
- The final robot accuracy is the same as the accuracy of the input program [2].
- Good repeatability is achieved with factors such as speed control [4].
- This form of programming is good where skill is required in determining the exact trajectory, i.e. the robot will follow the exact path and speed, e.g. spray painting and welding applications [1].

2.9.4 DISADVANTAGES OF ON-LINE PROGRAMMING

- The robot cannot perform any assigned tasks while being programmed or reprogrammed, and this leads to large downtimes [3].
- Certain applications may be too complex for this type of programming, and if programming is attempted, is slow.
- The running time of small batch sizes may be less than the actual programming time [2].
- In applications requiring accurate positioning, there is large room for error [35].
- On-line programming is not suitable for a large number of machines [2].
- Programs cannot be stored and moved from one machine to another, unless the robot controllers are compatible with each other [5].
- Programming must take place on the shop floor, and cannot be carried out in an environment more conducive to concentration [3].
- The program can not easily be edited, and very often the whole program has to be entered from the beginning [1].
- One cannot use subroutines with this sort of programming, and this generally limits the operator to simple programs with limited branching capabilities [2].
- The robot cannot be well integrated with other equipment, or with computer integrated manufacture (CIM) [18].
- Only simple 3D digitising is possible [33].

2.10 COMMON ROBOT PROGRAMMING TECHNIQUES: OFF-LINE

2.10.1 TEXTUAL PROGRAMMING LANGUAGES

Simple push-button teach-and-repeat schemes are sufficient for industrial robot programming in a wide variety of tasks, especially where a robot will perform a single task for a long run of workpieces [19]. However, reasons for going beyond use of the push-button approach to a textual robot programming language are numerous (see section 2.10.3). This teaching technique provides instructions to move the manipulator, read sensors, send signals to external equipment, set counters, perform logical branching, and many other instructions that make the task of programming the robot simpler. This form of teaching then becomes a program in the classical sense (off-line programming), and is either compiled or interpreted by the robot software system.

Some textual language robot systems in use are HELP, VAL and RAIL [18]. The software features common to all of these systems include the hierarchy of control levels, with the highest monitoring and coordinating the the complete robotic system, and the lowest interacting with the specific hardware.

Only a few textual languages are discussed in the following paragraphs.

AL

AL is a second generation robot programming language produced at the Stanford Artificial Intelligence Laboratory [19], an early leader in robot research. AL is a high level manipulator language with ALGOL-like control, and based on PASCAL [20]. AL provides constructs for control of multiple arms in cooperative motion. Commercial arms have been integrated into the AL system.

AML

The design concept for AML is to provide a powerful base language with simple subsets for use by programmers having a wide range of experience [19]. An interpreter implements the base language and defines the primitive operations, such as the rules for manipulating vectors and other aggregate objects that are required to describe robot behaviour. A design feature of the language is that these rules should be as consistent as possible, with no special case exceptions. It is intended that such a structure shall provide a growth path as programmers and applications become more sophisticated. Excluding task-orientated languages, AML is a good tool for the implementation of complex robotic applications [21],[22].

RAIL

RAIL was designed as a language for control of both vision and manipulation. It is an interpreter, loosely based on PASCAL. Many constructs have been incorporated into RAIL to support inspection and arc-welding systems [19].

SRL

SRL was designed on the basis of experience with high level robot languages (eg AL) and with traditional programming languages (eg PASCAL). Various features are present with SRL, such as sensor interfaces, input-output to digital or analog ports, and several more statements for different kinds of interpolation. The goal of the development of SRL is the design of a language which can easily be learned and adapted to further developments and applications, and also to provide an interface between future planning modules and the 'traditional' programming system [23].

WAVE

Wave, one of the first systems and languages for manipulator control, was developed at the Stanford Artificial Intelligence Laboratory [24]. It provides for the symbolic description of robot manipulator actions with manipulator positions defined in terms of the cartesian

coordinates of the end effector. All motions are coordinated functions of time and programs are expanded at planning time into position dependant, coordinated motions.

The WAVE system is simple to program as the instructions are high level and the on-line editor and macro system eliminate turnabout time. Speed of operation of WAVE, however, is at best one-third to one quarter human speed which, given the cost of the system, is a major drawback to its application.

RAPT

A language for instructing assembly robots was developed. In the language APT, one is able to describe the movement of cutters, which are conceptually cylindrical, with respect to geometric surfaces defined by the programmer. In RAPT one needs to describe assemblies of complex bodies and the actions required to bring them together. A situation is a state of the bodies composing the robot's world, which the programmer has chosen to describe, usually by specifying spatial relationships between features of some of the bodies [25].

FAR

A FORTH based robot control language called FAR was proposed. FORTH's major advantage is that it is flexible, and robotics is a suitable application towards which to mould it [3]. FAR is based on some of the desirable features of a robot language that include real-time capability, modular structure, parallel task extension, interactiveness and expandibility.

Up till now the development and application of various textual languages have been discussed, the choice being based on the language having been in actual use either in industry or in laboratories, the language having a textual form, and manuals or other detailed documentation being available to the author. These languages fall mostly into the structured programming level (see figure 2.15).

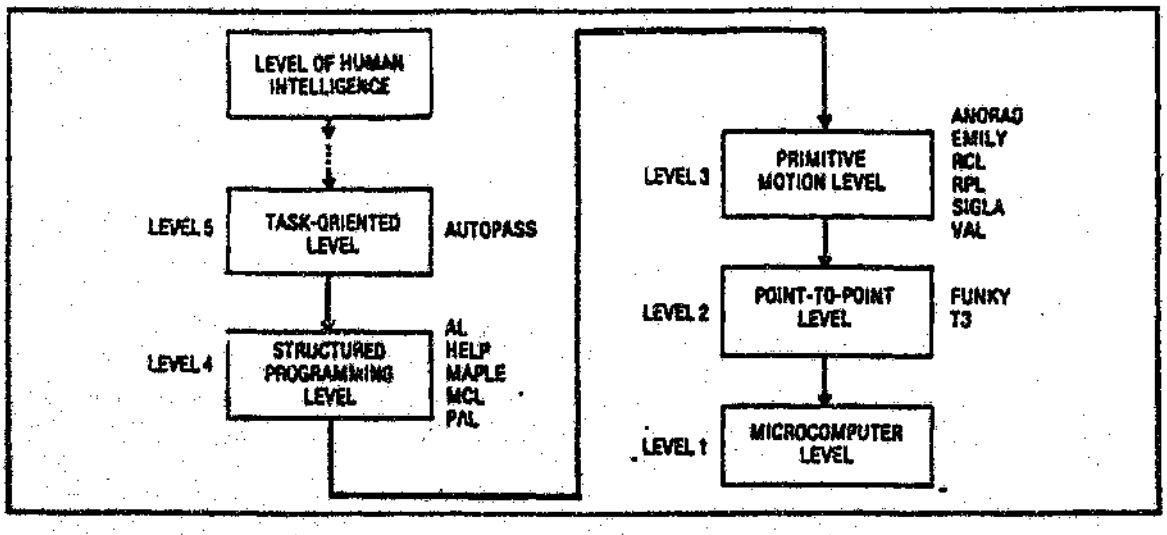


Fig 2.15 Classification of robot languages [2]

A more sophisticated and complex language is a task oriented language (TOL). A TOL is one in which the robot executes high level operations, and in which high level sensory feedback such as vision and range sensing is required.

AUTOPASS

Automated Parts ASsembly System (AUTOPASS) is a very high level programming system for computer controlled mechanical assembly. The AUTOPASS language is oriented towards objects and assembly operations, rather than motions of mechanical assembly machines [26]. It is intended to enable the user to concentrate on the overall assembly sequence and to program with English-like statements using name and terminology that are familiar to him. To relate assembly operations to manipulator motions, the AUTOPASS compiler uses an internal representation of the assembly world. This representation consists of a geometric data base generated prior to compilation and updated during compilation; it thus represents the state of the world at each assembly step. The level of the language has been chosen to provide a high degree of assistance to the user without the system's having to perform artificial intelligence type problem solving operations [2].

The languages that have been discussed are not the only ones that have ben developed. For comparative reasons, in 1982, Motiwalla [18] discussed four main textual languages that were in industrial use. In 1984, a publication [27] stated that over forty languages were in use or had been developed in the last decade. Not all the languages fall within the same level, but this still acts as a fair indicator of the rate of robot language development. A major problem with such development is that each manufacturer has tended to develop its own robot language for its own specific purpose, and this has resulted in little conformity and standardisation, and no universally accepted language.

2.10.2 ROBOT SIMULATION

As was previously discussed (see section 2.9 and [18]), one method of programming a robot is by on-line methods, such as manual lead-through and teach pendant programming. The robot, however, has to be interrupted from its assigned task while being programmed, thus resulting in large down times.

In the early 80's off-line programming was seen to be the solution to problems encountered with on-line methods, but it took almost a decade for this vision to transform into the reality of state of the art off-line programming.

In designing an FMS installation consisting of several cells, it very soon becomes apparent that any attempt to optimise the design manually is likely to fail. There are so many variables involved in the design that it is impossible for a human to balance them all. Somehow the relative benefits of using all the different combinations of various workpieces, tools, conveyors, pallets, vehicles, routes, operations, part types and machine groupings must be evaluated, to try to maximise potential production. For this task, computer simulation is an ideal tool, as systems are able to transfer engineering data to external

management control and information systems, and thus lead to increased productivity. Increasingly sophisticated software packages are now becoming available on the market. An example of such a system is GRASP (Graphical Robot Applications Simulation Package), developed at the University of Nottingham, UK. This software system can assist in workplace layout, position and velocity evaluations, clash detection and coordination. An example of the graphics produced is shown in figure 2.16.

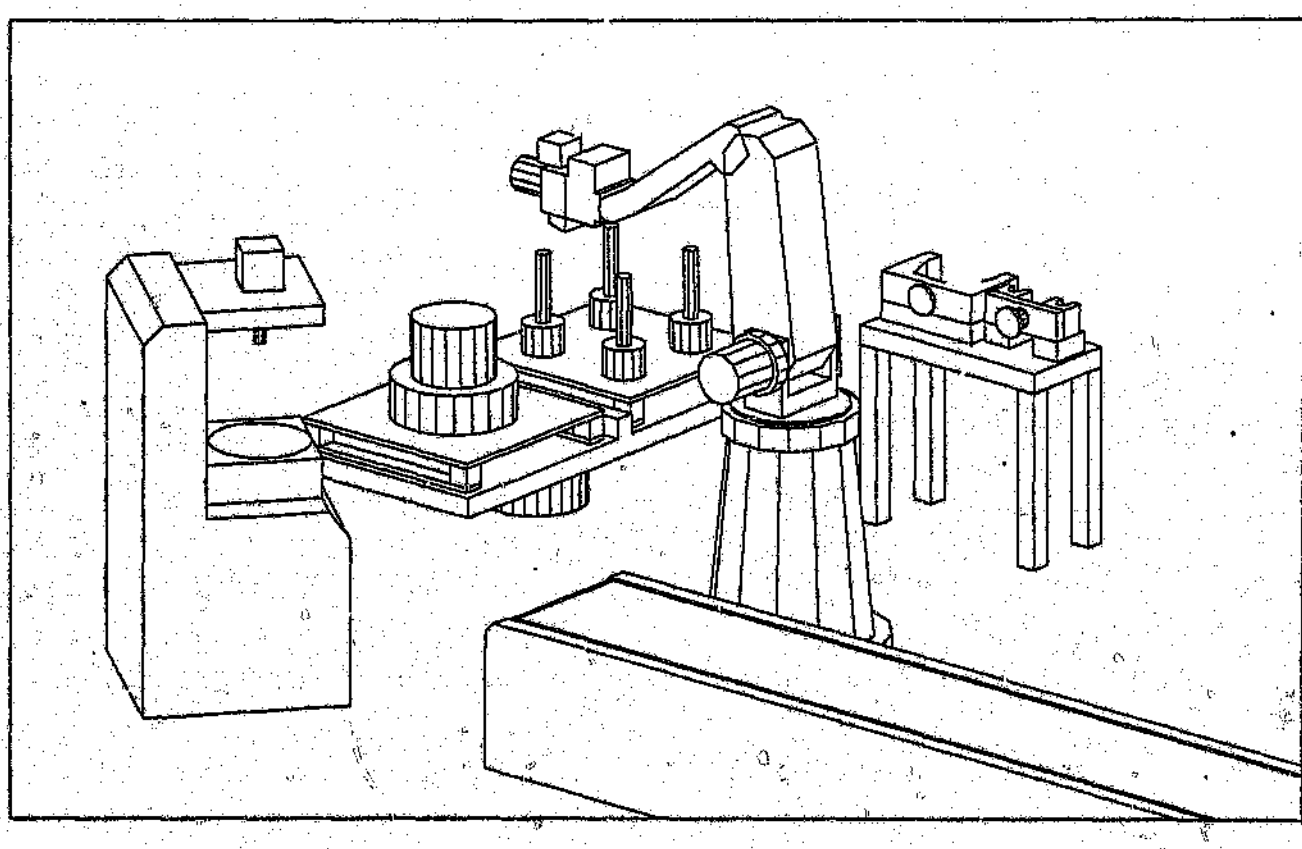


Fig 2.16 Example of graphics produced by the GRASP simulation system [1]

In 1984, Weck and Niehaus [27] addressed the problem of a lack of standardisation in current robot systems, making off-line programming difficult to obtain. In Germany a system has been developed to overcome this problem.

A problem arises with a user who operates various robots from different suppliers. Recognising this problem, all

major German manufacturers came together, aiming to standardise the interface for off-line programming of industrial robots. The system developed greatly enhances the productivity of robot programming, and has benefits such as portable and adaptable software, low priced hardware requirements and CAD integration.
With this standardised interface, controllers are now able to accept off-line created robot programs.

Hartley [28] reports that decentralised control systems for FMS of all types, off-line programming for robots and improved sensors are all part of a package under development at Hitachi's Production Engineering Research Laboratory.
Hitachi's off-line programming system is part of a plan to move to computer control of manufacture. Two basic systems are involved in the project; the FMS controller and the programming unit.
Hitachi demonstrated the system with a line of four robots, but it was set up so that a program could be written or modified only when the line was inoperative, which in a real situation would negate some of the advantages of off-line programming. On problem Hitachi came across was the cost of the hardware, but with the rapid development in this field, this hurdle is now easily overcome.

A robot programming system should bring several important features together in one unified programming environment. This allows important price - performance tradeoffs to be made to suit particular applications. A comprehensive language capability offers substantial flexibility, but is difficult for some users to apply. Systems programmed by teach pendant or menu-style interfaces on the other hand are simpler to use. SILMA developed ROBOCAM [29], which attempts to offer the best of both approaches by allowing the user to program in a high-level language, or by using a mouse to interact with a menu and a graphic display of the workcell. ROBOCAM also supports RISE, a high-level arm-independant interactive language. This language is

specially designed for robot programming and supports many geometric shapes and operations. RISE is similar to PASCAL in syntax, however it is an interpreted language leading to good interactivity.

ROBOCAM has a small CAD sub-system which can be used to create and edit three-dimensional wireframe models of workcell components. This modelling is done in one of two ways: textually using RISE, or through use of special model construction facilities which make use of the mouse - menu interface.

Accurate manipulator kinematic and dynamic models are important in making off-line programming of robots feasible. ROBOCAM supports a family of path generation algorithms and chooses the one that matches that used by the robot manufacturers so that robot paths in simulation match the path taken by the actual device. This not only helps assure accurate path generation, but also makes it possible to generate good timing estimates of workcell cycles. Multiple robot manipulators working together in a single workcell can be simulated with ROBOCAM, with a separate program written for each of the robot controllers.

When used with VISICAM [30], SILMA's 3-D computer vision system may be added to ROBOCAM to form SILCAM.

Rathmill [31] discusses using a computer aided robot simulation in the development of a robotised production line. The most effective way to design and analyse the performance of a workcell is to graphically simulate the motions of the cell components, thereby providing a visualisation of the operation of the actual cell. Recent advances in CAD enables one to build and display mathematical models of physical objects, to give these models the same constraints as robots, and to visualise and manipulate 3-D models far more effectively with a far greater degree of accuracy than is possible using conventional drawing and modelling techniques.

Computer Aided Robot Simulation (CARS) systems [31] allow system integrators to position components in robot workcells and to dynamically simulate the operation of the robot. Any problem identified at this stage can be corrected on the graphics terminal, rather than on the shop floor. Design modifications can be made before commitments to purchase equipment are undertaken.

This type of design, simulation and off-line programming tool is a powerful aid to companies committed to robotised production processes. The types of production gains, and paybacks, that can be achieved using simulation systems can significantly increase the benefits to be realised by implementing a robotised production system. There are many cases where the use of a CARS system could have more than justified the cost of that system on the first project alone [31]. CARS could make the difference between success and failure of those critical manufacturing processes.

One of the major users of such simulation systems is the automotive industry, especially in Europe and the US. The system is also used in the aerospace and electronic industries. The current applications include welding, especially of complex paths and spot welding, assembly, materials handling and palletising. These systems also offer great advantages to consultants and system houses, who can evaluate different cell layouts and designs very quickly and with no risk, either financially of physically.

Carter [32] states that the future of off-line programming systems lies with the graphical system. Alternative approaches, such as the early off-line programming languages (see section 2.2.2), offer no viable method of producing valid robot programs, because of the difficulties in visualisation encountered when trying to program a six degrees-of-freedom robot in three dimensions.

The emphasis in robotic cell design and programming, should also change from being subdivided into compartments, i.e.

CAD, simulation, cell design and off-line programming, to a unified approach, using systems such as MRS, from McDonnell Douglas (see [31]), which offers all the required elements for this task in one package.

Williamson and Ashton [33] insist that the only way to justify robotic implementation, especially in low volume production, is by having the robot cell as flexible as a human, if not more. Their research showed that solid modelling with a CAD system was the solution to flexible manufacture.

It is reported by Daniels [34] that BULA and FILS, a company producing a range of machines for polishing, used off-line simulation to design, test and program robot motion in polishing and buffing operations. Customer [illegible] the quality of the product, the surface and [illegible] all factors contributing to a need to produce [illegible] off-line programs, without a loss of downtime [illegible] and to maintain or improve the product. A [illegible] CADCAM simulation software system was used [illegible] with an ASEA IRB 1000 robot in this [illegible] BULA and FILS concluded from the quality of [illegible] components that off-line programming can, and [illegible] the most demanding tasks. It decreases the [illegible] to a bare minimum and significantly [illegible] the production rates, thus reducing the expense [illegible] robots on-line.

[illegible], Mattis and Gill [35] looked at the problem of choosing the best robot for a task, and how simulation could help the decision process. If simulation is used properly it can eliminate many of the pitfalls. A simulation package ensures that a thorough analysis of the project is carried out because it forces the engineer to question the assumptions about cell design that are usually made at the start.

Austin Rover (ARG) developed the software package for

off-line programming, ROBOGRAPHIX [35], in collaboration with Computervision. The package has helped ARG in the assessment of software solutions and aided the design and evaluation of a workcell. It has also been used for robot modelling and for off-line programming. ROBOGRAPHIX incorporates software modules which allow workcell design, robot simulation, off-line robot programming and the generation of output and definition of desired robot kinematic models. Mattis and Gill conclude that a graphical simulation package is a most useful tool within a manufacturing facility when designing robot workcells, and that it is an important element in any CAD/CAM system.

During an international conference on developments in automated and robotic welding [36], a lot of interest was generated in off-line programming. Middle and Goh [36] of Loughborough University told the conference of WRAPS - Welding Robot Adaptive off-line Programming and expert control System. This is a relatively low-cost system implemented on a micro-computer capable of standing alone as an electronic planning tool, with the capability of serving as the core of an integrated manufacturing system.

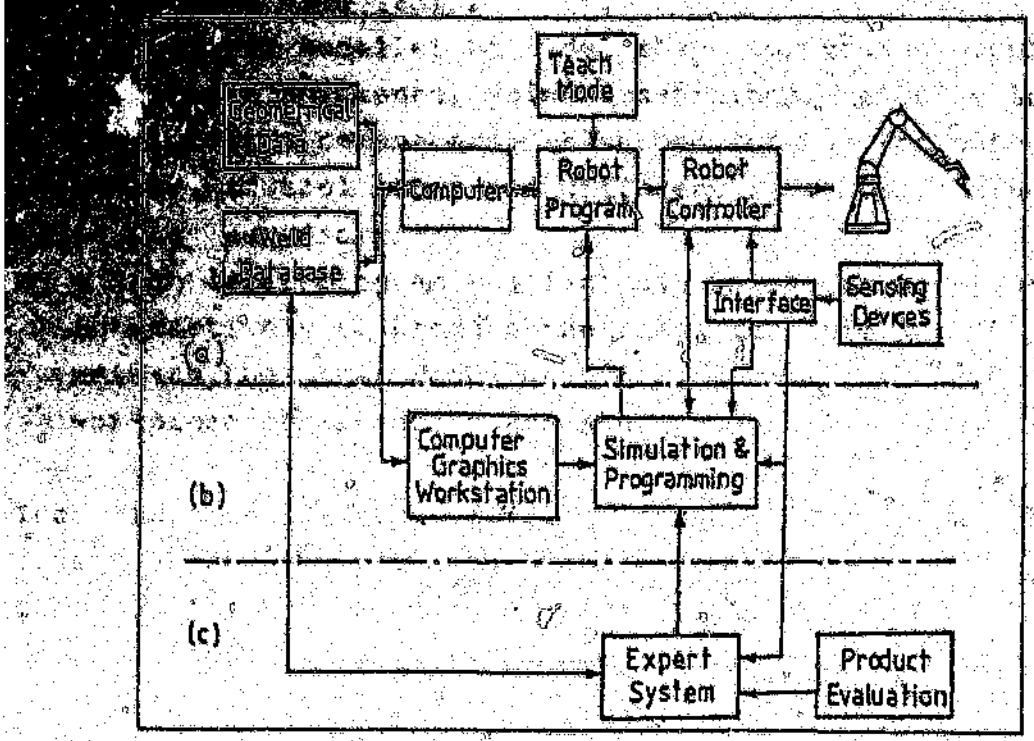


Fig 2.17 An integrated robot welding system [36]

Widfelt [36] of IVF, Stockholm reported on three different methods of off-line programming; alphanumeric, alphanumeric with graphics support, and simulation and off-line programming with a CADCAM system, citing actual projects that have been carried out using currently available systems.

The FIAT Automobile Company of Italy recently ordered a ROBCAD system [37]. ROBCAD is a world leader in design, simulation and off-line programming systems for robotic applications. FIAT now join other leading companies (BMW and Volkswagen) who have implemented ROBCAD as a standard system for simulation and off-line programming.

Sorenti and Bennaton [38] discuss the benefits derived from off-line programming in complex welding tasks, using a [illegible] graphical simulation system. They mention that [illegible] important characteristics which must be [illegible] system to allow it to be used for the off-line [illegible] industrial robots. These characteristics [illegible]:

- [illegible] modeller to create 3-D models of the robot cell [illegible] (see [33]);
- [illegible] modeller to model robot structures complete [illegible] their movements, joint constraints, velocity and [illegible] parameters. A method for the accurate [illegible] of tool paths and different robot configurations [illegible] necessary;
- [illegible] level programming language which can be used [illegible] interactively to produce the robot programs and to [illegible] the other components of the robot cell, such as [illegible] workpiece manipulators and conveyors.

[illegible] interfaces between the off-line programming system and other CAD systems provides the added capacity of transferring the geometry of parts directly to the programming system. The 3-D simulation allows the robot programs to be replayed using computer animation to check for correct operation, to detect collisions which may occur

between objects and to estimate process cycle time.

For off-line programming, estimates of the relative positions of objects in the robot's workspace are no longer sufficient to ensure that the task will be performed successfully. BYG Systems has recognised the importance of relating the real worlds' spatial environment to that of GRASP's internal 3-D simulation model and has developed a method of workspace calibration which uses the robot as a measuring device. This allows the relative position of objects within the GRASP model to be updated to give a better correlation between where the robot is required to go and where it actually goes.

This off-line programming exercise produced very promising results. As a comparison, BYG estimated that the robots would be required to be out of production for between six to eight weeks using existing manual programming techniques. Using GRASP, the project was completed with less than 5 days of lost production time. This included robot calibration time.

The use of 3-D graphical simulation as a technique for off-line programming is now a viable practical proposition. This is backed by developments in a number of areas including interfaces to CAD packages and calibration of the off-line programming systems. The technique is proving beneficial to specific processes such as welding, where programming is a primary factor for economic production. In today's competitive market place, manufacturers who implement this new technology will be able to achieve the flexibility and lower productions costs they seek.

2.10.3 ADVANTAGES OF OFF-LINE PROGR^MMING

- A textual language permits robots to be programmed, allowing the robot to continue to perform useful work during the programming process, and thus reducing the downtime of the robot to a minimum [3].
- Teaching points by use of a teach pendant can be cumbersome for operations such as palletizing, where each new position can be computed rather than taught [32].
- Programming is instantaneous, and is ideal for small batches where programs have previously been stored [32].
- The complexities of the robot task can be aided by the capabilities of the computer [1].
- Programs can be stored, recalled, or transferred between machines, provided that sufficient accuracy is available [1],[31].
- Often programs can be easily edited, and in addition, features such as logical branching, subroutine and debug capabilities and use of data structures provide useful tools to develop large and complex programs [18].
- Off-line programming involves developing robot programs partially or completely without requiring the use of the robot itself [1].
- The actual programming of the robot can take place in an environment more conducive to concentration, i.e. in a clean, quiet office remote from the factory floor [3].
- Off-line programming of robots permits the responsibility of programming to be moved to a higher level [27].
- The operator need only be skilled in computing [23].
- A robot program can be dumped onto any number of machines, whereas with on-line programming it is only possible to control one robot with one program [4].
- Production rates can be significantly increased, reducing the expense of programming the robots and maintaining software for this purpose [34].
- Off-line programming is perfectly suitable for CIM, and CAD/CAM can be integrated with robot programming [18],[33].

- The level of visualization available on simulation packages enables a potentially expensive problem to be identified and rectified early on in the design stage of a project, and not when in production. The safety of the robot system is also improved [35],[38].

2.10.4 DISADVANTAGES OF OFF-LINE PROGRAMMMING

- Off-line programming methods and languages are more difficult to learn and use [1].
- The user or programmer of a robot is restricted to communicate through a terminal [2].
- Simulation packages are difficult to use, and are not inexpensive [3].
- Discrepancies between CAD models and actual components manifest as incorrect positioned sub-parts, and a degree of distortion from the expected shape, and this problem of unknown final accuracy using a program will have to be overcome [35].

2.11 FUNDAMENTALS IN MOTION CONTROL

A control system is essentially an arrangement of physical components connected or related in such a manner so as to command, direct or regulate itself or another system [44]. There are two forms of control, these being open loop control (without feedback) and closed loop control (with feedback).

2.11.1 OPEN LOOP CONTROL

The generalised open loop control system is a control arrangement where no feedback loop is present.

In a control situation, one would have some sort of energy input to a system. This can be in the form of a force, pressure, displacement, voltage, torque, etc. This actuating signal is fed into one or more control elements, which can include transducers, switches, PLCs, microchips, etc. Through a manipulating variable, the system is acted on by means of friction, damping, electrical power, etc. resulting in some sort of desired output. The operation of such a control system is best explained through a flow diagram.



Fig 2.18 Open loop control

Open loop control, to be successful, relies on the perfect modelling of the system. If the behaviour or operating conditions of the system is known, then the input signal must be such that the output is that which is desired, taking into account the characteristics inherent to the system.

2.11 FUNDAMENTALS IN MOTION CONTROL

A control system is essentially an arrangement of physical components connected or related in such a manner so as to command, direct or regulate itself or another system [44]. There are two forms of control, these being open loop control (without feedback) and closed loop control (with feedback).

2.11.1 OPEN LOOP CONTROL

The generalised open loop control system is a control arrangement where no feedback loop is present.

In a control situation, one would have some sort of energy input to a system. This can be in the form of a force, pressure, displacement, voltage, torque, etc. This actuating signal is fed into one or more control elements, which can include transducers, switches, PLCs, microchips, etc. Through a manipulating variable, the system is acted on by means of friction, damping, electrical power, etc. resulting in some sort of desired output. The operation of such a control system is best explained through a flow diagram.

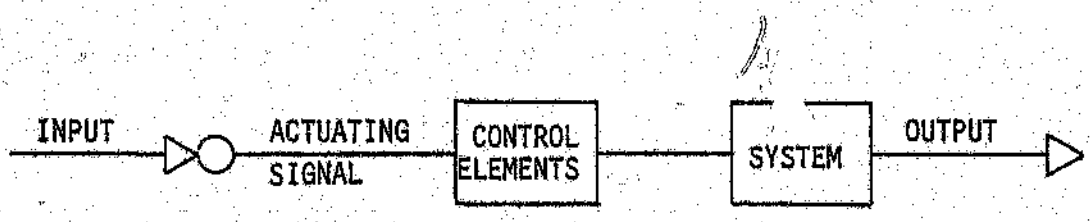


Fig 2.18 Open loop control

Open loop control, to be successful, relies on the perfect modelling of the system. If the behaviour or operating conditions of the system is known, then the input signal must be such that the output is that which is desired, taking into account the characteristics inherent to the system.

A problem with such control is that if the conditions of the system change under operating conditions (e.g. a motor heats up, or the load on a drive changes), it is very unlikely that open loop control will output the desired results. Take the example of a motor driving an arm of a robot, picking up a load. As the arm rises and the centre of gravity of the structure beyond the joint shifts closer to the joint, the torque needed to displace the arm at a constant speed is reduced. If no feedback loop is present, and the input signal does not compensate for this change in the system's operating conditions, the results would be undesirable.

The difficulty of such a control loop is in precisely determining the operating characteristics of a system, factors such as how the age of a motor might affect its operation being extremely difficult to establish. This leads one on to the second form of control, closed loop control.

2.11.2 CLOSED LOOP CONTROL

In a closed loop control system, three components are essential. They are a servo amplifier, an actuator, and feedback devices [44]

In the same way that a human uses a brain, eyes and arms to pick up an object, a control system uses these three components. The brain uses the position of the object as the input signal to the system and the position of the hand, which is detected by the eyes, as the position feedback signal. The objective of the control system is to move the hand to the object and therefore reduce the distance between the hand and the object to zero. This will mean that the brain will continue to send a signal to move to the hand, until there is no difference in position between the hand and the object.

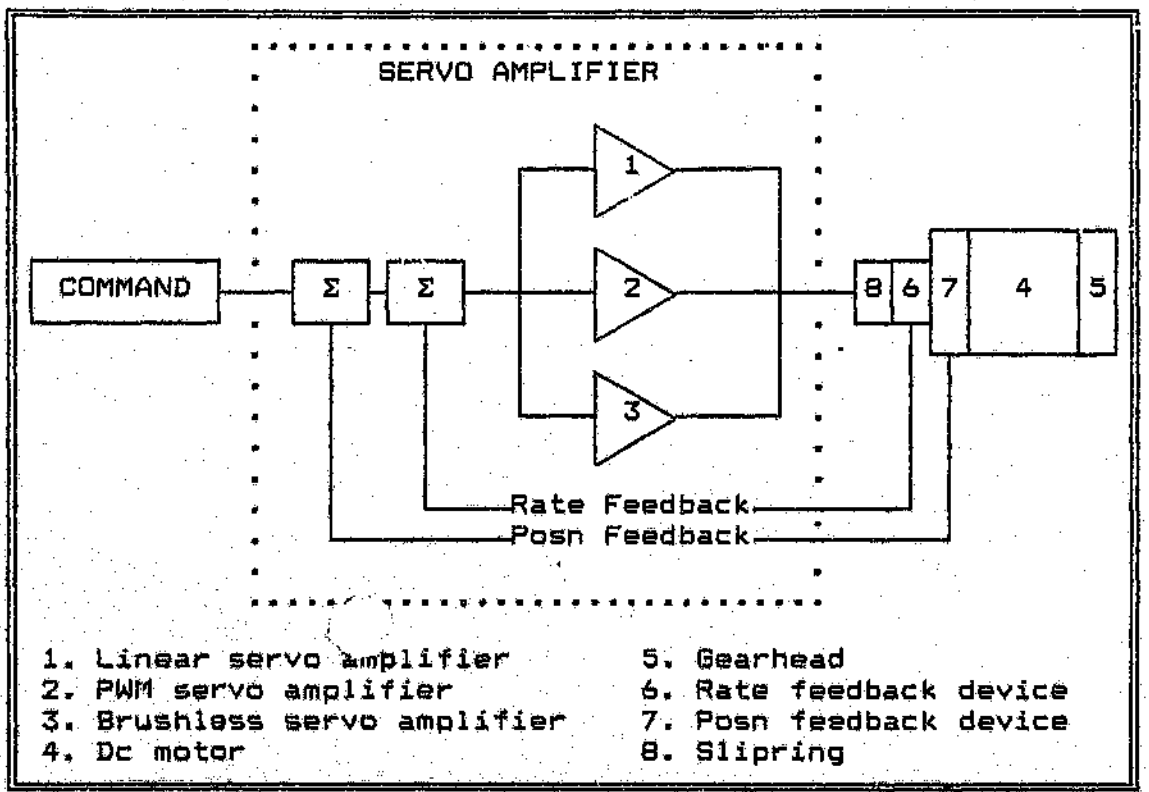


Fig 2.19 Closed loop control

Servo Amplifiers

The servo amplifier can be seen as the brain of the control system, and is the driving force of the actuators. It compares the command signal with the feedback signal and depending on the outcome, control the power delivered to the motor. Servo amplifiers can be divided into three categories, namely linear, pulse width modulator and brushless servo amplifiers.

Feedback Devices

Feedback devices are used to close the loop in control systems by measuring the speed or position of the output. This is done by means of signals that can be related to the speeds or positions. This signal is then fed to the servo amplifier which will compare it with the input signal, and if there is a difference, the servo amplifier will produce an error signal and will then control the power to the motor in order to reduce the error signal to zero.

Position feedback devices include potentiometers, optical encoders, contact encoders, synchros and resolvers, and inductosyns.
Rate feedback devices include AC tachometer, brushed DC tachometer, and brushless DC tachometer.

Actuators
An analogy is drawn between the human hand and the actuator. DC motors are used, the motors falling into three classes: torque motors, servo motors, and brushless motors.

3 PROJECT OBJECTIVES

The project carried out entails a feasibility study into the development of a substitute off-line robot controller system.

At present, all robots in industry are controlled through a robot controller. Programming of the robot is either on-line or off-line, and control is eventually fed through the controller from the programming source to the robot.

The feasibility study deals with bypassing a controller as a means of controlling the robot, and rather concentrates on the use of a programmable logic controller (PLC) and a personal computer (PC) combination to establish control. A standard PLC equipped with a fast counting module for axis control, and a PC with dedicated software are the critical components in the study.

In order to achieve acceptable control of the robot, various secondary objectives will have to be met. These include:

- gaining proficiency in programming the PLC in both Grafcet and Ladder languages;
- determination of the operating characteristics of the robot, including optical encoder operation and resolution, and DC motor operation and accuracy (i.e. voltage range and speeds);
- verification of robot control by controlling first a single DC motor, and later, four motors;
- incorporation of any additional PLC modules necessary to enhance motion control;
- interfacing of the PLC to the PC;
- determination of the PLC programming methods from the PC, i.e. control of the robot directly from the PC;
- determination of the operating characteristics, accuracies, and efficiencies of the robot under this type of control as a final analysis of the system's industrial feasibility.

These objectives are in the direct scope of the work carried out, but are not limiting factors. The objectives must inevitably extend themselves finally to the development of a PC software package that will facilitate the programming of the PLC and robot from a CAD package or other simulation tool.

4 EXPERIMENTAL EQUIPMENT

4.1 THE ROBOT

4.1.1 GENERAL CONSIDERATIONS

The robot that is being used is from the DAROS range (see appendix A1 & [39]), (see plate 1), model Part Time (PT) 300V, and distributed by DAINICHI - SYKES, UK. It is a precision robot specially suited to many applications such as arc welding, machine loading of smaller components, general handling and packaging.

The robot has five axes, has jointed arms, and has a load [illegible] of 5kg. The drive system of the PT [illegible] is fitted with optical shaft [illegible] the robot has a repeatability of ±0.1 mm.

[illegible] is controlled with a robot controller [illegible] it is a high performance controller [illegible] axes (including the manipulator), [illegible] the robot its task.

[illegible] control, each arm is equipped [illegible] two indicate extremes of arm [illegible] the other indicates the home position. The [illegible] can be defined as the point from where each [illegible] initiates its motion and the point from which [illegible] arm references its current position.

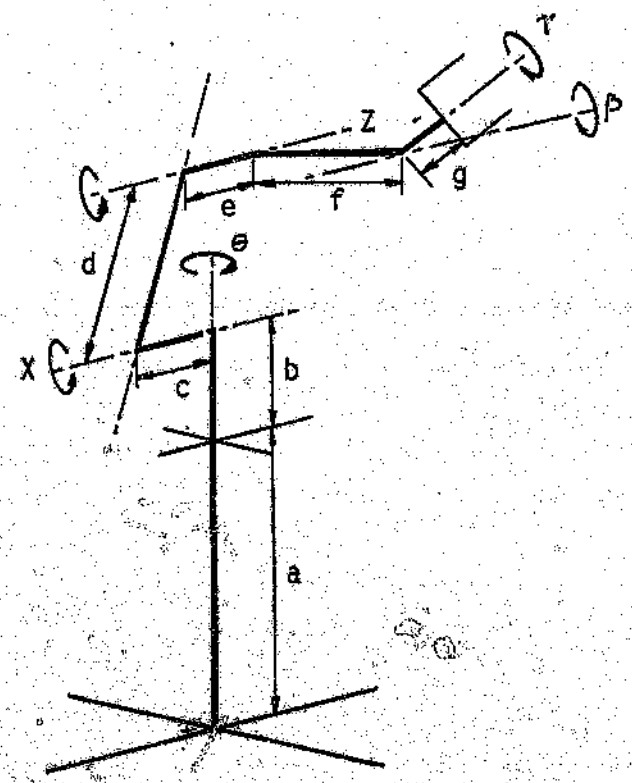


Plate 1 PT300V Robot

3.1.1 KINEMATICAL GEOMETRY

The kinematical geometry is best described with the aid of a line diagram. The axes of rotation are labelled from the base of the manipulator as θ, X, Z, β and γ respectively. θ is the rotation of the base, and β the movement of the wrist (pitch).

The line diagram gives the dimensions of the robot structure, from the centre line of rotation of one axis to the next. These dimensions are necessary when computerised robot motion is performed, where exact dimensions are needed for the geometric manipulation matrices. The line diagram given follows the position and configuration of the robot in plate 1.



a = 640
b = 145
c = 170
d = 350
e = 170
f = 350
g = 115

Fig 4.1 Line representation of the robot

4.1.2 ROBOT ARM GEARING

Each arm of the robot revolves around an axis. A DC servo motor is coupled to a harmonic drive, this being the final drive to move the arm, except for the β axis, where the drive is from the harmonic drive, through a belt to the axis.

Each joint of the robot structure is rotated with a drive, and the gearing ratios related to the harmonic drives were determined. Table 1.5 (see section 5.2.2) lists the number of pulses of the optical encoder per degree rotation of each arm, as well as the gearing ratios of each drive.

4.2 DC SERVO MOTOR

On the robot are mounted four DC servomotors. The motors comprise of three basic subassemblies, a motor, a brake, and an optical shaft encoder (see appendix A4 & plate 2).

The DC servomotor are high performance DC motors, with direct excitation. There are various advantages of using DC servomotors, which include high starting torque, the ability to operate at zero speed, high torque at low speed, high power conversion efficiency, excellent speed regulation and the fact that they are ideally suited to control applications.

4.2.1 EXTERNAL INTERFACING

Each of the DC motors has a four pin MIL-SPEC plug through which the supply and brake is accessed. Each pin has a letter code, pins A & B being the motor power supply, and pins C & D the electromagnetic brake release. The DC supply to the motor has a range of -24 V to +24 V. The brake disengages when supplied with a minimum voltage of +12 V, and engages when there is a break from power.

The motor is coupled to the robot through a harmonic drive (see appendix A3 for a detailed explanation of harmonic drive operation). Harmonic drives enable high torque, high efficiency speed reducers to obtain large single stage reduction ratios without the traditional disadvantages of excessive backlash and mechanical complexities [40]. Other advantages of using harmonic drives include compact size, giving a small total envelope size, and high reliability.

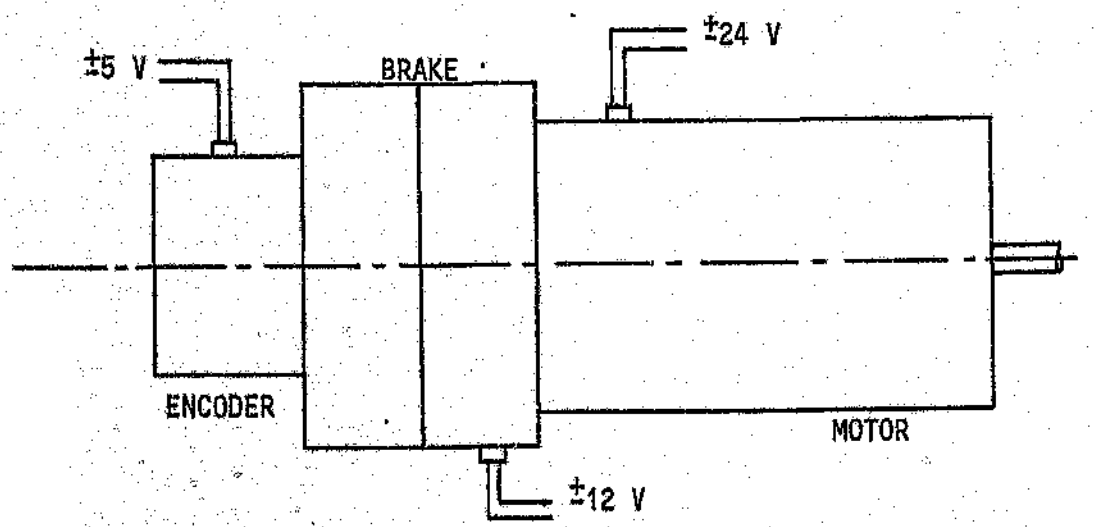


Fig 4.2 External interfacing of the DC motors

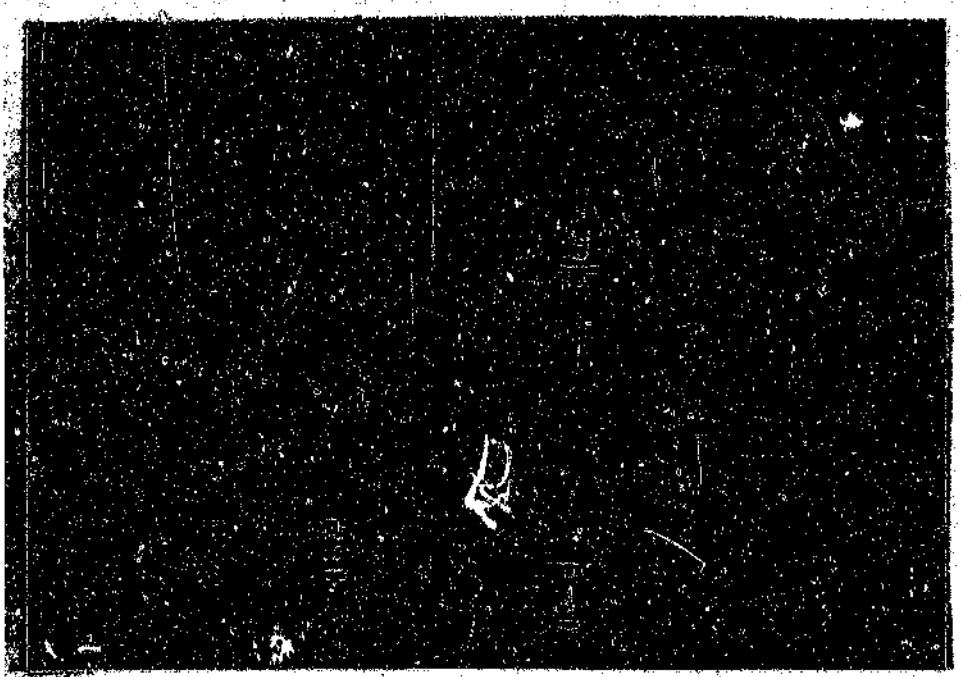


Plate 2 DC Servomotor

1. DC servomotor.
2. Magnetic brake.
3. Optical shaft encoder.
4. Harmonic drive connector.

4.2.2 OPERATIONAL CHARACTERISTICS

Calibration of the motors revealed that the two size groups of the four larger motors proved to be identical in this respect. Curves, relating input voltage and associated current, to speed, for these motors can be found in section 5.2.1.

The calibrations were made with the motor's end bearing being held stationary in order to simulat. the grip of the harmonic drive. The exact effect of loading from the drive has not been calibrated, but it has been determined that loading causes an increased current to be drawn from the motor power supply, and that the above trends shift to give lower rotational speeds whilst their linearity is maintained.

Over current protection is not necessary, but over voltage protection may be desired. Powering the motor while the brake is engaged is one way of burning these motors out, and this is something that should be avoided.

4.3 OPTICAL SHAFT ENCODER

4.3.1 GENERAL OPERATING CHARACTERISTICS

The encoder, connected directly to the end of the motor shaft, is an electro-mechanical device, converting angular position of its shaft into a digital electric signal. When connected to suitable electronic circuits and through proper mechanical links, the encoder is able to measure angular displacements, linear and circular movements, and also rotational speed and acceleration. The incremental encoder supplied with the robot makes use of the photo-electric method of control. This is the most versatile and reliable method. A collimated light beam is aimed against two radial reticles; a static and a moving (disk) one. Light (infra-red) that can pass through both reticles drops on a group of phototransistors placed immediately beyond the static reticle. By using several lines (instead of one) on both reticles, the resulting signal is strong, and is actually the average of lines of the rotating disk (see appendix A4).

encoder mounted on the DC servomotors is powered by a . In place of slots in a disk, our encoder is a disk marked with a series of uniform lines in a track around the perimeter. As the lines interrupt light beam, 'increments' of information are produced in form of a square wave pulse train output signal.

Analysis of the encoder operation yielded a resolution of 500 lines or pulses per revolution, and is thus capable of position feedback accuracy of 0.733^{o}. The frequency of the pulse relates to the number of lines on the disk and the disk speed. By using two scanning heads, one obtains two wave trains 90^{o} out of phase, and thus direction of rotation is sensed, as well as up or down counting of angular or linear displacements. A single line on the disk is present to provide a marker pulse for reference purposes, i.e. one pulse every rotation of the motor shaft.

4.3.2 EXTERNAL INTERFACING

Before any of the encoders can be used, there external interfacing must be known. The encoders, mounted on the back of each motor, are accessed by a 10 pin MILSPEC plug. Each pin is assigned a letter, and the wiring is referenced through these letters, shown below.

PIN	DESCRIPTION
A	Lead true (+)
B	Lead ground (-)
C	Lag true (+)
D	Lag ground (-)
E	Reference true (+)
F	Reference ground (-)
G	CT +5 V DC
H	CT ground
I	Shield ground
J	Unused

[illegible] make up the quadrature sine wave train, and are [illegible] output signals. Pins E & F form the outputs for [illegible] pulse. Pins G & H are the only input pins, [illegible] DC power supply is needed.

[illegible] necessary to boost the output signals from the [illegible] to a level that it capable of being read by the PLC [illegible]. This is simply achieved by pulling up the signal [illegible] the aid of a suitable voltage source and resistor. [illegible] principle is illustrated in figure 4.3.

A resistor value of 100Ω connected to a 5 V power source allows the lead and lag signals to be pulled up to the required level required by the PLC module. However, due to the voltage required by the PLC module of the reference signal, a simple pull up resistor is not sufficient, and an emitter follower circuit is needed.

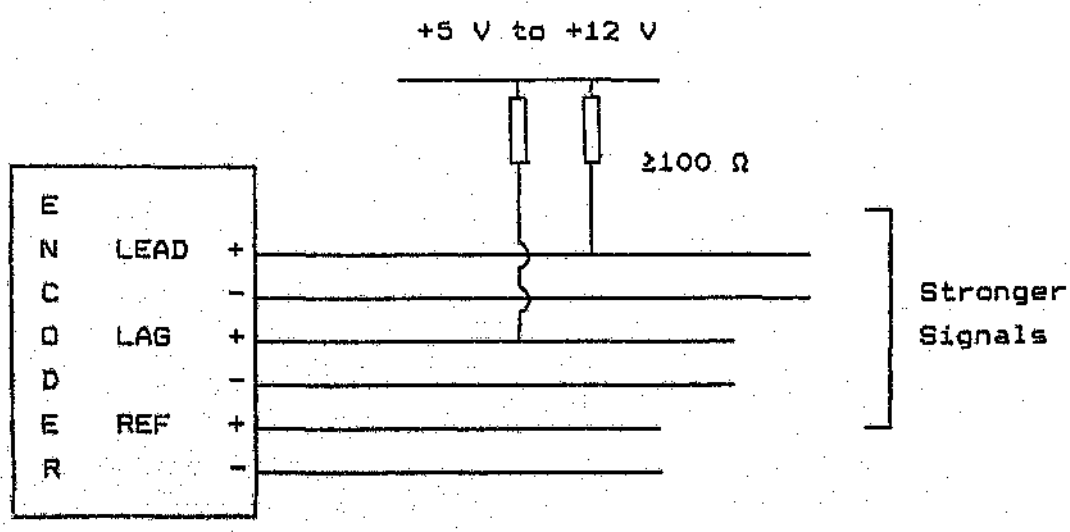


Fig 4.3 Pull up of encoder output signal

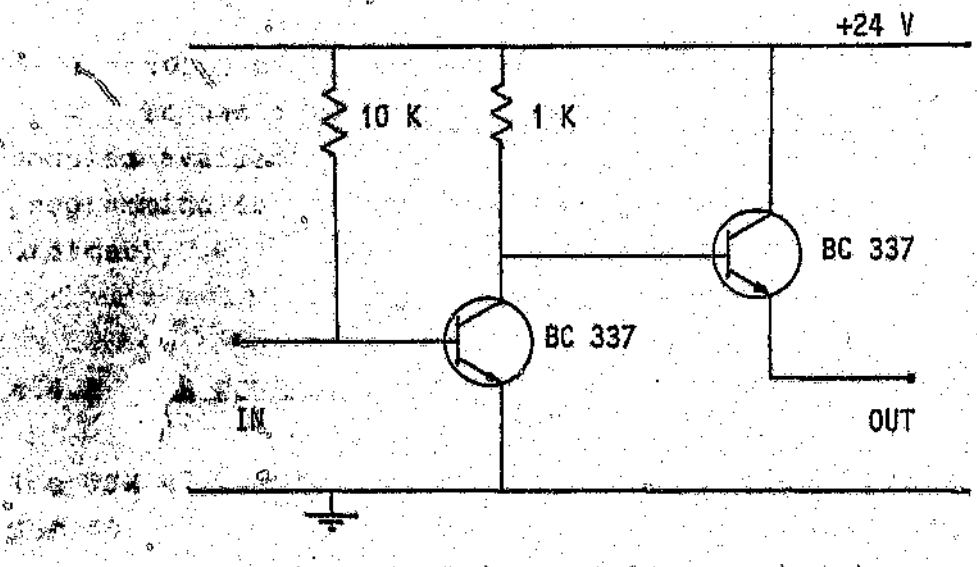


Fig 4.4 Emitter follower circuit

Because the encoders hold the reference signal high, dropping low once every revolution, two transistors are used in the emitter follower circuit to invert the signal, achieving a constant low signal, going high once every revolution. This is needed by the PLC module for certain program executions.

4.4 THE PROGRAMMABLE LOGIC CONTROLLER (PLC)

A PLC is defined as a digitally operating apparatus which uses a programmable memory for the internal storage of instructions for implementing specific functions such as logic, sequencing, timing, counting and arithmetic to control through digital to analogue input or output modules, various types of machines or processes [3].

The various components (see plate 3) that make up the complete PLC system (see appendix A5 for technical specifications [41]) are:

4.4.1 THE TSX 47 PROGRAMMABLE CONTROLLER RACK

Discrete I/O modules can slot into the TSX 47 10 controller rack. It has a modular design, with a wide choice of I/O [illegible] available. There is complete I/O bus security, and [illegible] is through familiar languages (Ladder and [illegible]

[illegible] POWER SUPPLY MODULE

[illegible]47 10 is supplied with a power supply module, TSX [illegible] The power supply is a double-size module mounted [illegible] left hand side of the basic rack. The panel [illegible]ises:

- an OK light (green) - internal voltage present and correct;
- an ON light (orange) - mains power present and correct;
- a power supply terminal block;
- an alarm relay contact;
- a mains power failure detector ON/OFF switch.

4.4.3 THE PROCESSOR MODULE

A processor module, TSX P47 12, is also present. This comprises:
- the central processing unit (CPU);
- the serial data port for the terminals;
- the I/O bus interface;
- the monitoring and display functions.

Slotted into the processor module is a user memory cartridge. This cartridge is a volatile random access memory that is used during program creation and debugging.

4.4.4 DISCRETE I/O MODULES

The input module, TSX DET 16 12, receives data signals from sensors and transmits them to the processor after performing necessary isolating, filtering and matching functions.
The output module, TSX DST 16 35, receives commands from the processor and controls the corresponding connected circuits through isolating and amplifying circuits.
Each I/O module is in the form of a rectangular box which encloses and protects the components. An I/O status light indicates the status of each input and output point.

Several different methods of wiring the discrete I/O modules can be used. (See appendix A5 for wiring diagram used for the study.)

4.4.5 THE FAST COUNTING MODULE

The TSX AXT 200 is a simple counting and positioning module that is used in conjunction with the TSX 47 programmable controller [42]. It has two independent input channels, each of which can count the 5V DC or 24V DC pulse signals received from discrete input devices such as incremental encoders and limit switches. The fast counting module comprises of:
- an up/downcounting discriminator switch for each channel;
- a fault light for each channel which comes

on when the counting capacity is exceeded;
- two LEDs per channel indicating the logic states of the two counting inputs (IN0 & IN1);
- one LED per channel indicating the logic state of the reset input (RST);
- one LED per channel indicating the logic state of the inhibit input (INH);
- one socket connector for each channel.

A detailed summary other than the FCM manual is to be found in appendix B2.

4.4.6 THE PROGRAMMING TERMINAL

The TSX 7407 0 programming terminal is the universal programming and debugging tool for the TSX 47 programmable controllers [43]. The terminal has a personal computer type setup, comprising the following:

- [illegible] keys;
- [illegible] keypad;
- [illegible] keyboard;
- [illegible] "soft" keys displayed on screen;
- [illegible] consisting of six lines of 24 characters (LCD);
- [illegible] on and power supply cable;
- PC [illegible] ion socket: 20mA serial data link;
- [illegible] eral connection socket (printer, tape, recorder);
- [illegible] age cartridge;

The [illegible] board is used to enter the program and data into [illegible], and to dialogue with the PC. It is continuously [illegible] tored to prevent incorrect entries and syntax errors, [illegible] ich are indicated by an audio warning.

4.4.7 THE SWITCH BANK

A bank of four toggle switches was available for use with the programmable controller. These switches can be used for practical demonstrations, and even to simulate the outputs expected from an incremental encoder, i.e. up/downcounting (except very much slower). They can be

used to indicate the active state of a switch in a ladder program, and functions such as reset and inhibit are easily demonstrated.

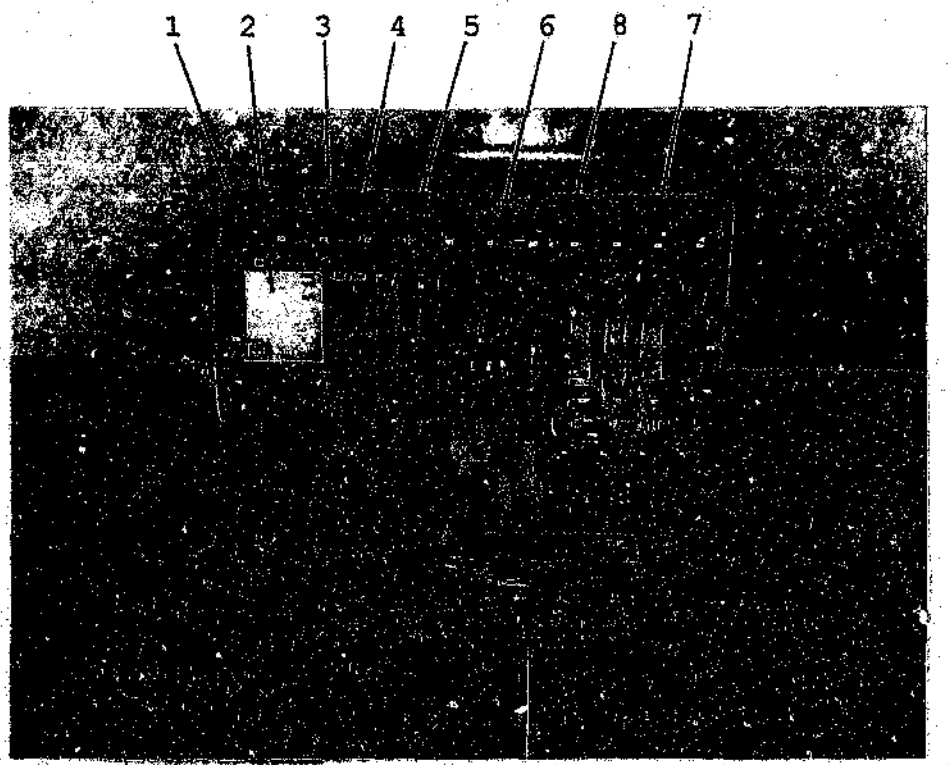


Plate 3 The complete PLC system

1. TSX programmable controller rack.
2. The power supply module.
3. The processor module.
4. The user memory cartridge.
5. Discrete I/O modules.
6. The fast counting module.
7. The programming terminal.
8. The switch bank.

4.5 PLC/PC INTERFACE SYSTEM

The PL7-2 software package supplied by Telemecanique allows a standard IBM PC 286, AT or PS (with at least EGA display) to operate as a programming terminal. It effectively substitutes the TSX T407 programming terminal by simulating the more advanced TSX T607 terminal.

The system comprises of:
- the PC, or a PS, with VGA display;
- a software key inserted in the PC printer port;
- an active (as opposed to passive) current loop convertor;
- the PLC.

The system is connected up as seen in the following diagram, the current loop convertor (CLC) being the essential link between the PC and the PLC.

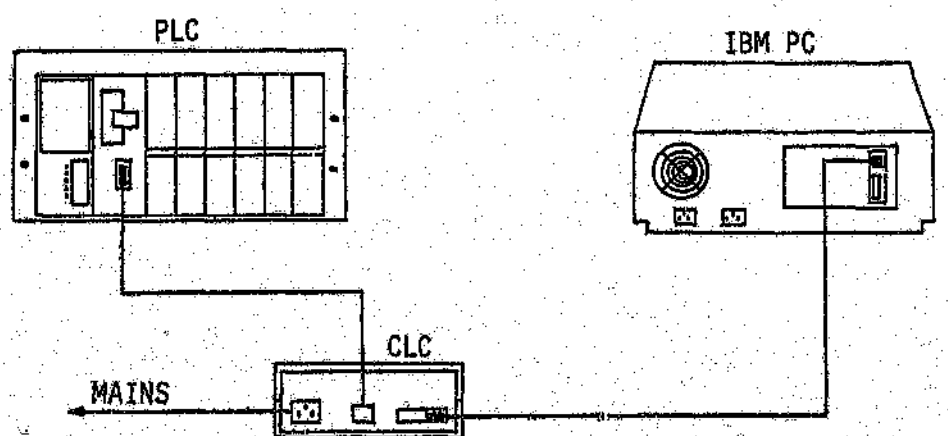


Fig 4.5 PLC, current loop convertor and PC link

4.5.1 SYSTEM ADVANTAGES

The first and foremost advantage of a PLC/PC system is that it forms the first step to an off-line facility. The PC does not have to be connected to the PLC when developing programs, and so the user can program in any environment desired, only transferring programs when and where needed. Other advantages include:
- PL7-2 program transfer between the PC and the PLC;
- Ladder and Grafcet program storage and retrieval;
- extensive programming and debugging enhanced through informative displays;

- application documenting and printing;
- on-line parameter adjustment facilitated by full screen real-time displays;
- simpler menu accessing due to larger screen displays.

For installation and running purposes, the software requires at least 1 MB of free PC disk space.
Operation of the system requires extensive learning time, although previous experience on the smaller TSX T407, as well as the large screen with informative displays, would speed up the learning process.

Whenever a new program is written, the software has to be configured to suit the system parameters, i.e. the PLC and the various modules in use. However, configurations are saved with each program, and are not lost at each reset. The PLC used is a TSX 47-20, the memory module size being 8 K, and the pre-processing language is Ladder. As a single unit, the PLC is addressed both physically and logically, in network 0 and at station 1.

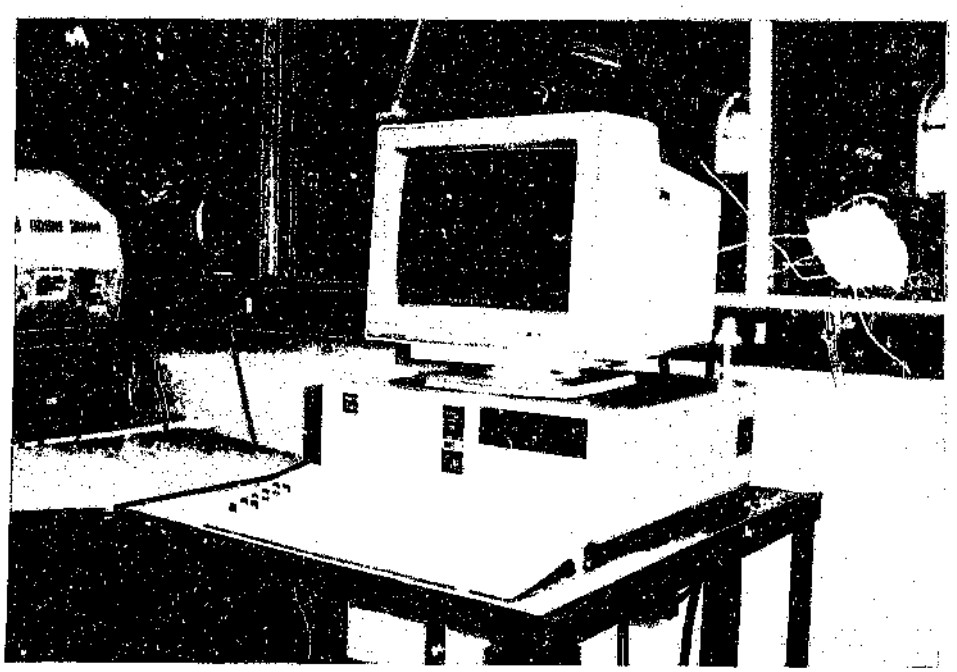


Plate 4 IBM PC with current loop converter

4.6 FREQUENCY GENERATOR

The TSX AXT 200 fast counting and positioning module has two independent input channels, each channel having two counting inputs: IN0 and IN1. To perform both up and downcounting, both inputs must be used. Two square wave inputs are generated, with either lead or lag characteristics. If IN0 leads IN1 then upcounting takes place, and if IN0 lags IN1 then downcounting takes place. To simulate a dual logic frequency input to the fast counting module, a frequency generator was built. An explanation of the circuit, how it works, and the wiring diagrams are included in appendix B3&4, and sections 5.9&10. The outputs from the frequency generator are connected to the fast counting module to simulate the output characteristics of an incremental encoder.
The complete system (wired up), including the frequency generator and the fast counting module is seen in plate 5.

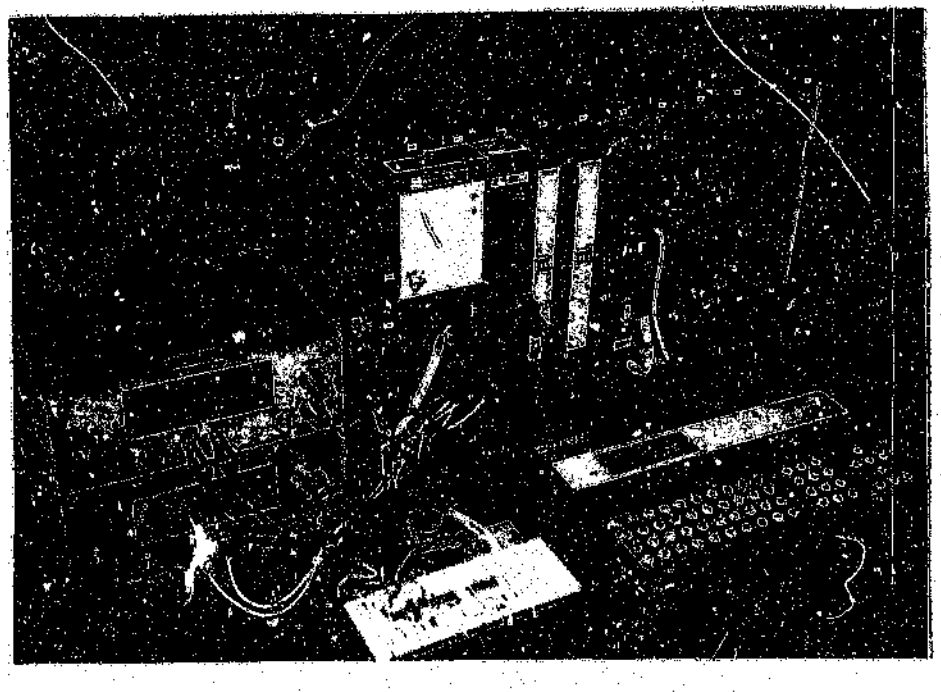


Plate 5 / The complete system connected

4.7 PULSE WIDTH MODULATOR

A fundamental piece of equipment in a control system is an amplifier. The pulse width modulator (PWM) is a circuit designed to regulate the switching of a power supply in order to obtain an amplified output voltage proportional to an input signal.

The principle behind the operation of the amplifier, the various circuits and an explanation of each module can be found in appendix B6. The construction of the device is broken down into five modules in order to facilitate simple analysis and testing of the circuit during its construction phases. The circuitry that was built relates to an amplifier designed to accept an input signal of -10 V at 0.8 A and to generate a proportional voltage in the range of -24 V to +24 V at 10 A.

The complete PWM consists of the following five modules:
- the voltage dropper,
- the adding amplifier;
- the pulse width modulating regulator;
- the driver, and;
- the inverter.

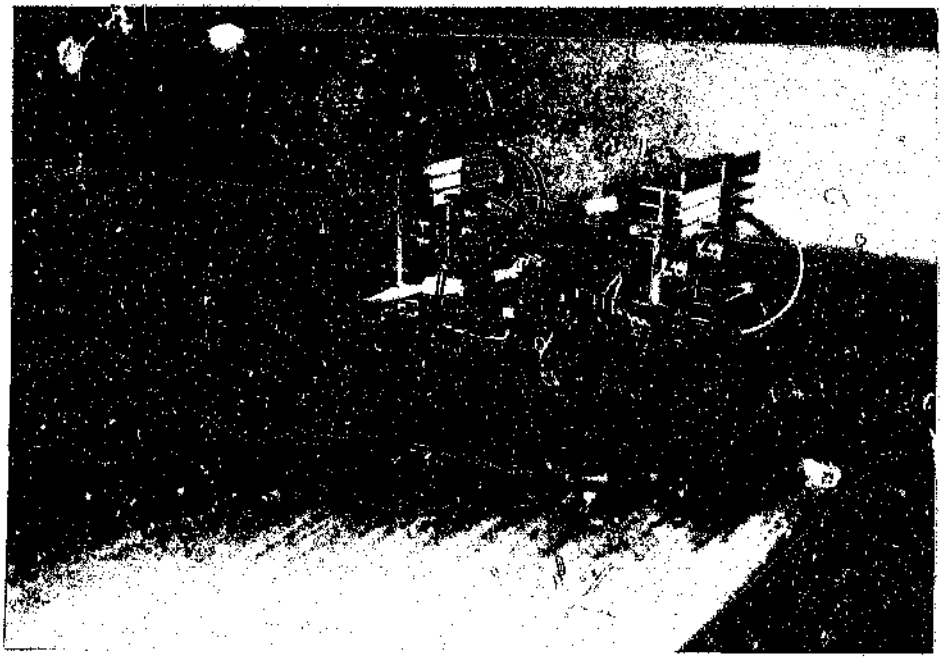


Plate 6 PWM mounted on vero board

The entire circuit can be mounted on a vero board (see plate 6), with loose wiring being used for the inputs and outputs for each module to enable singular operation. For final operation, these wires can readily be joined together to form the complete PWM.

The PWM was built in anticipation of the procurement of an axis control module (see section 5.13). A requirement of such an axis control system is an amplifier, and the amplification characteristics of the PWM are according to those as specified by the manufactures of such control modules.

5 FEASIBILITY STUDY

The feasibility study carried out on the establishment of control of the robot with the aid of a PLC is described in the section that follows. The study involves the discovery process regarding the robot motions, encoder and servo motor operations, and feasibility of the control decided on.

5.1 OPTICAL SHAFT ENCODERS

The encoders that are used on the robot are incremental encoders, and are mounted on the back end of the DC servo motors. Each axis of rotation on the robot is controlled by a motor, and each axis employs encoder feedback for position control. The control of the robot hinges on the encoders, this being seen as the most important piece of equipment in the control study, and is looked at first.

5.1.1 ENCODER REPAIR

Due to the fact that none of the four motor-encoder combinations would yield distinguishable signals, certain circuit analyses and repairs had to be carried out. Without fully functional encoders, the establishment of control would be impossible.

No circuit diagrams were available on the optical encoders, and thus a thorough idea of how the circuits operated had to be arrived at from comparisons between the four encoders, as well as by tracing out the circuits.

Various problems arose and became evident as the analysis of the circuits was carried out. These included burnt out [illegible] LEDs, a blown chip, burnt out resistors and [illegible] and a few loose wires. No serious hardware problems arose, and it was possible to repair the encoders without any expert assistance.

5.1.2 OPERATING CHARACTERISTICS

A detailed understanding of the encoder circuitry was obtained during the repair process, and acted as an aid in the control process. With a thorough knowledge of the encoder operation, control would be simplified.

The layout of the circuit was only established in detail at its output end, the reason being that with incorrect

operation, this tended to be the end that received the most damage. The schematic that follows gives an indication of the layout of the circuit from the power input to the chip (the chip used is essentially a triggering chip, used to trigger a square wave from a sine wave).

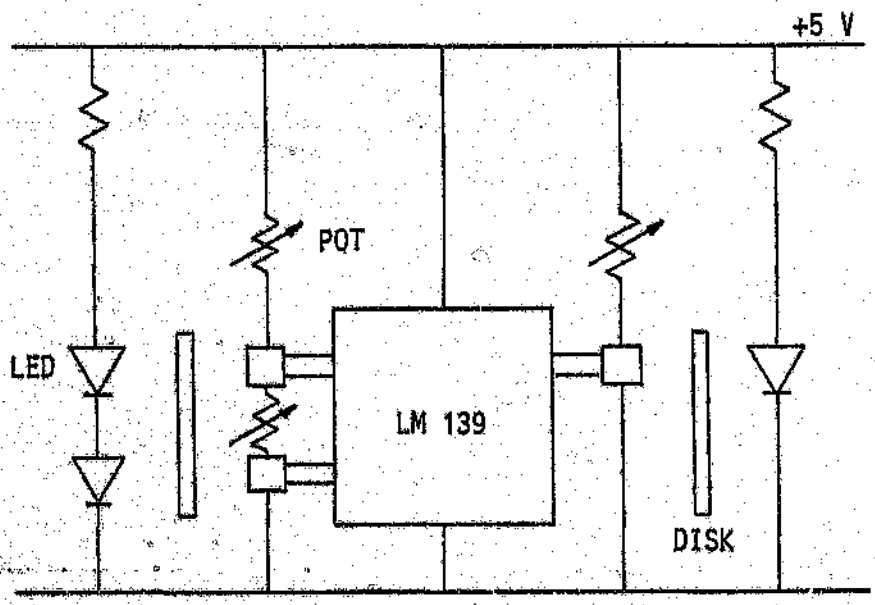


Fig 8.1 Encoder circuit - power input to chip

In the first half of the circuit, the two pulse train LEDs are powered in series, and hence damage to one will result in the loss of the other. A third photo electric cell and LED combination is situated on the opposite side of the circuit board, at a smaller radius than the other two LEDs. This cell is responsible for detecting the reference slot on the glass disk of the encoder.
One LED was replaced, and for any future replacement of these LEDs, a suitable LED would be an FPE 500. However, the potentiometers have to be adjusted due to the lower power rating of these LEDs.

The infra red beam emitted by the LEDs are picked up by photocells positioned on the opposite side of the glass disk to the LEDs. Each photocell is powered through a potentiometer. Adjustment of this potentiometer adjusts the trigger level and thus the balance of mark time to space time on the final square wave output from the chip.

This principle is shown in the figure 5.2, the ideal balance being at 0^{O} on the quasi-sine curve, resulting in a 90^{O} balance in the square wave. It should be noted that the principle of passing a light source past a slot does not result in a straight edged light intensity signal being received by the photocell. This is due to a fraction of light showing at first, proportionally increasing as the two slots align, before fad'ng again to zero visibility. This results in a quasi-sine wave, and this wave has to be triggered to produce a square wave output.

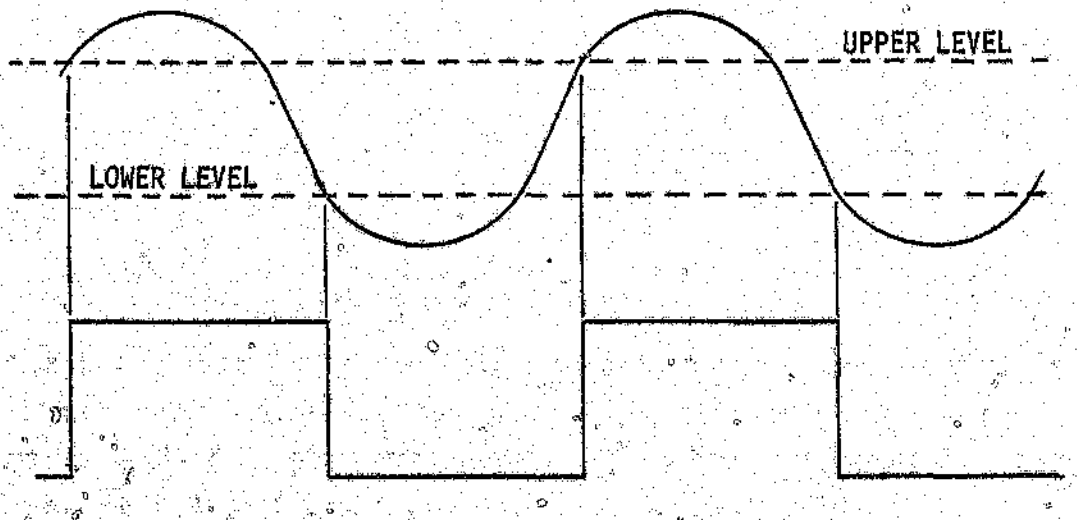


Fig 5.2 Square wave triggering

The operation of the chip is imperative to any analysis and repair to the circuit. The main function of the chip is to perform the triggering on the sine input in order to obtain a square train. As this trigger level is set according to a comparative op-amp value, it is the strength of the input wave that will vary the triggering level, hence the effect of the potentiometer, in front of the chip.

The chip (LM 2901) consists of four op-amps, laid out as in figure 5.3.

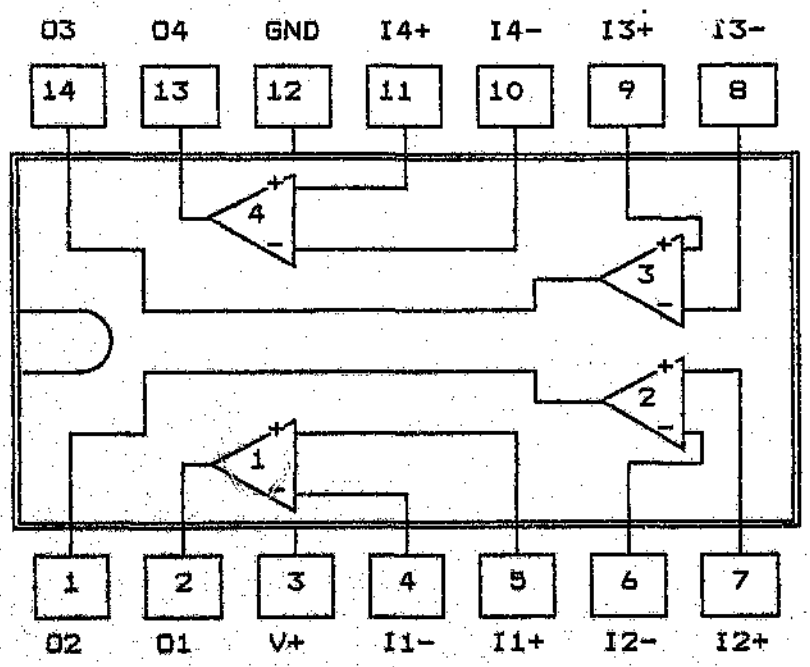


Fig 5.3 Connection and pin diagram for encoder chip

The photocells are connected directly to the chip input points, the quadrature cells being connected to inputs 1 and 2, and the reference being connected to op-amp 3. Thus, each input can be observed on an oscilloscope, and should appear as a quasi-sine wave, and each output should be a square wave. For the reference signal, these forms only occur once every revolution.

From the chip, the signal is sent to a cable driving transistor. By having the output connected to the collector of the transistor, it is possible to pull these signals up to desired voltages allowing for greater flexibility. The output signal from the encoder can be increased with a larger pull-up. This will, however, require the use of larger resistors to limit the back current entering the cable driving transistors.

The transistors are the first components to fail when incorrect voltages are supplied to the encoder, or the pull-up circuit is strapped to a voltage that is too high with a resistor that is too low.

The wiring diagram for each of the signals, shown entering and leaving the chip, is illustrated below.

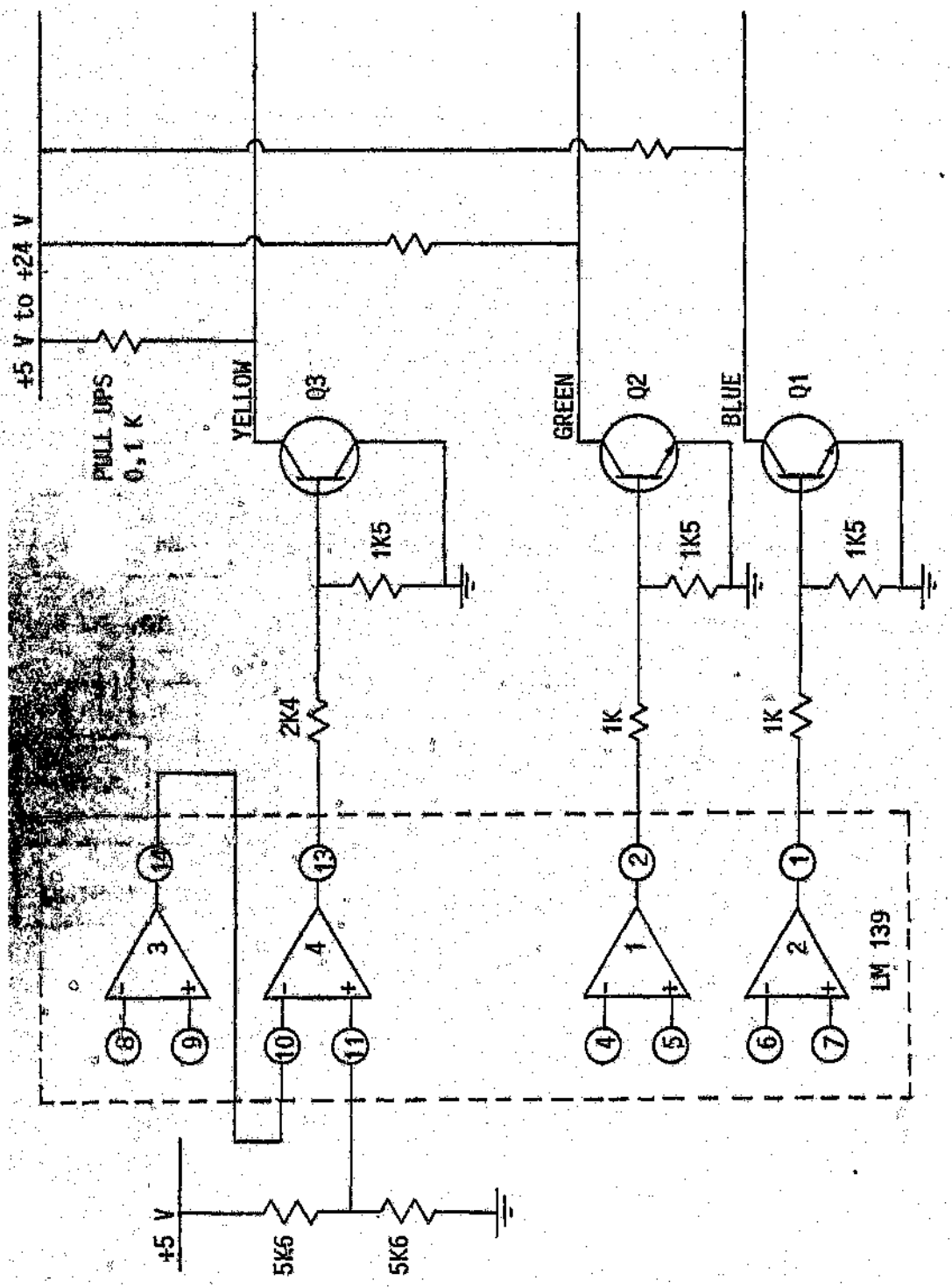


Fig 5.4 Encoder circuit - into and out of chip

5.1.3 ENCODER INTERFACING

To test the operation of the encoders, it is necessary to connect these to the PLC.
Three power supplies are used, these being used to power the encoder, the motor, and the brake. The connections are summarised in the wiring schematic shown in figure 5.5.

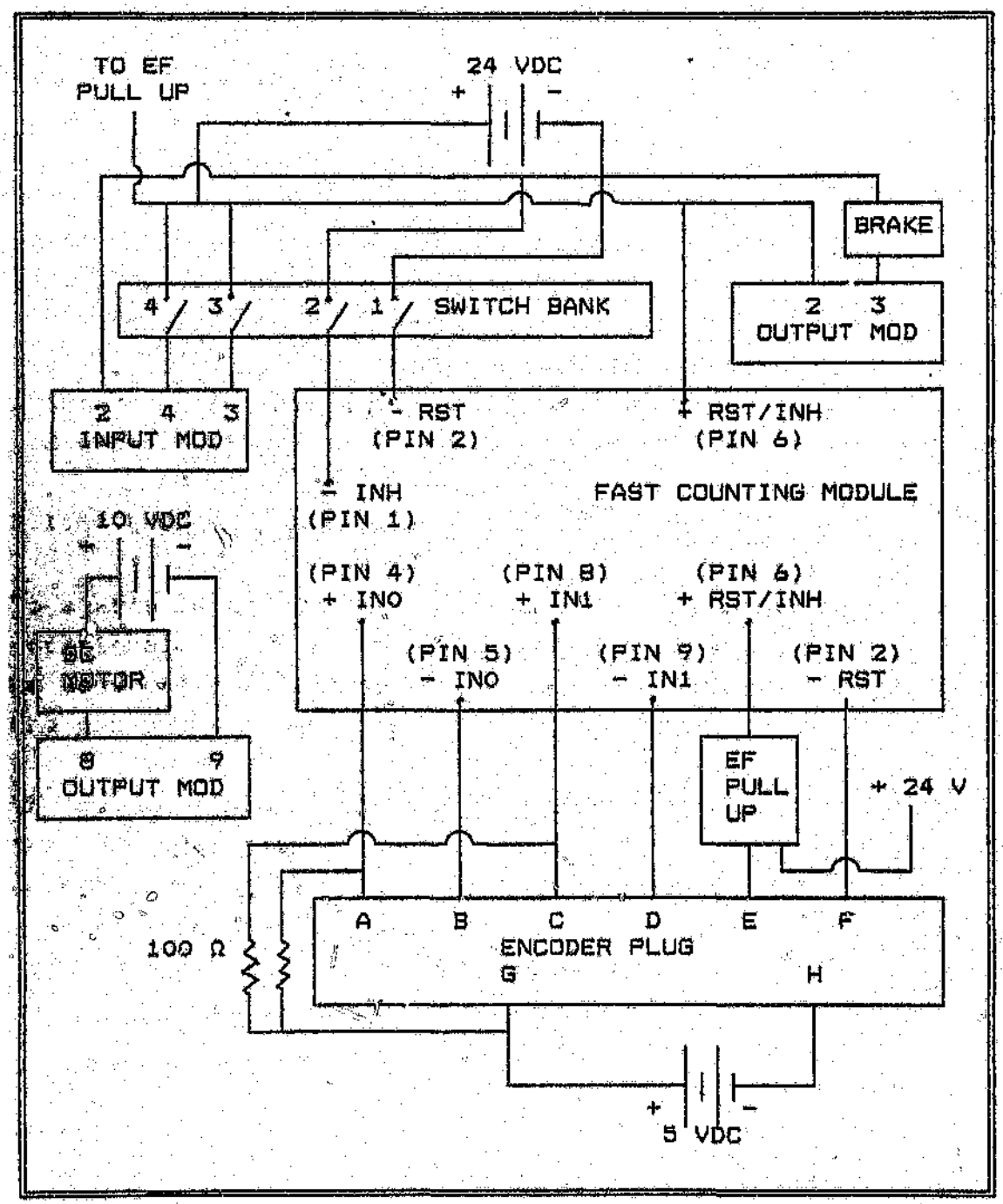


Fig 5.5 Wiring diagram to test encoder operation

The one 24 V circuit powers the reset through the switch bank, and the other utilises the emitter-follower (EF) circuit to reset the FCM on the encoder reference output. The emitter-follower is only needed when the reference output is needed to measure the resolution of the encoder. The inconvenience of this circuit, as well as the necessity of an additional 24 V power source, rendered it undesirable in the final control circuit. It was decided that the reference pulse would be used only for encoder resolution determination, and not for final position control of the motor.

With the present circuit layout, the brake is powered by output O1,0 and the motor is switched through output O1,4. The first two switches on the switch bank set the reset (RST) and inhibit (INH) inputs to the FCM, and the second two set inputs I0,0 and I0,1 on the input module. The programs making use of this circuitry make use of these addresses.

5.1.4 ENCODER RESOLUTION

To enable full control of the motors with respect to their rotational position, the resolution of the encoders have to be accurately determined, i.e. the number of grid lines on the glass plate per revolution of the motor shaft.

The grid lines were too fine to count physically, and thus the reference pulse had to be made use of. By starting to count pulses received as soon as a reference pulse was detected, and then holding the count constant when the next reference pulse was detected enabled the pulses per revolution of the encoders to be determined.

A Ladder logic subroutine was developed that, when utilised in a counting program, would perform the desired counting operation. This subroutine toggles B1 on the rising edge of input I1. As shown in the following timing diagram (Fig 5.6), B1 will be high for every alternate pair of I1 pulses.

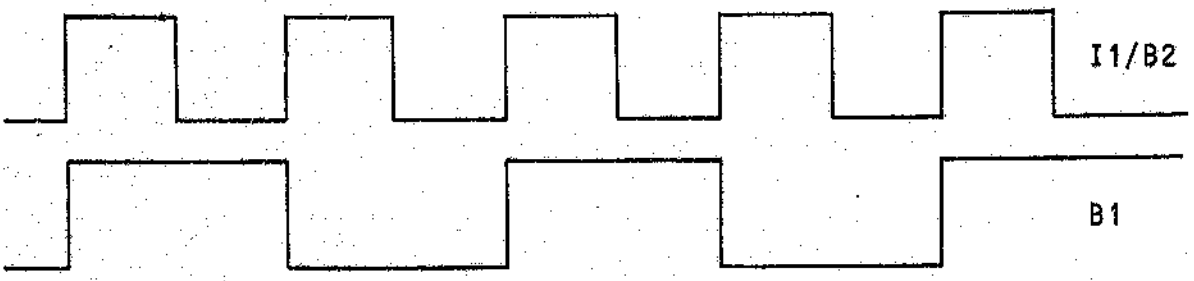


Fig 5.6 Timing diagram - B1 high every alternate pulse

The program that follows incorporates the subroutine, counting the pulses from the encoder for one revolution, holding the count for the next, and resetting the count before the process is repeated.

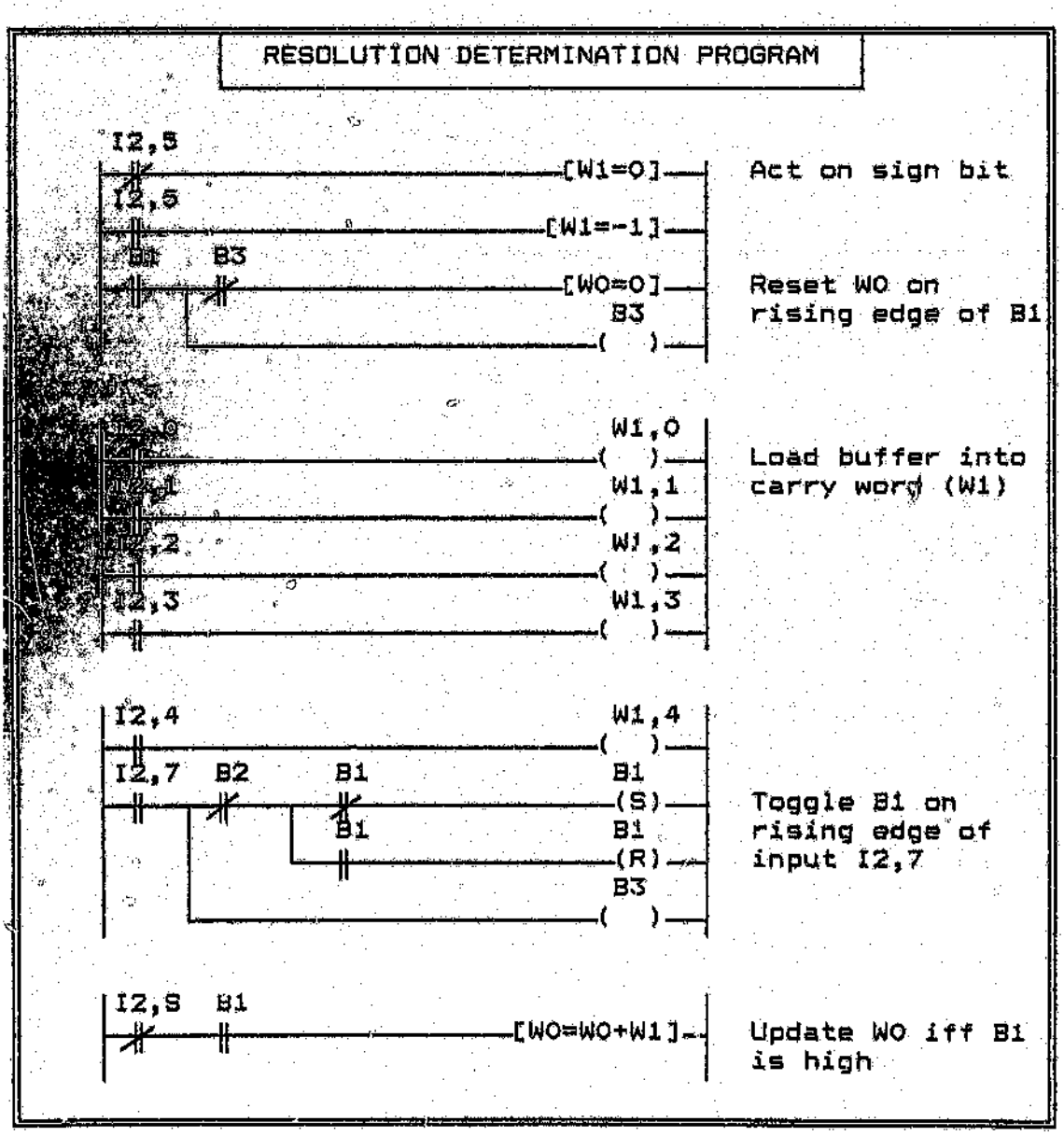
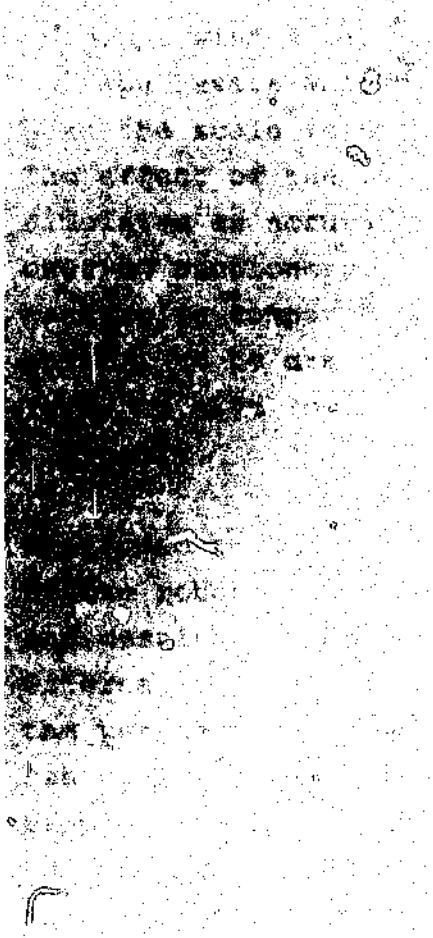


Fig 5.7 Program - encoder resolution determination

The program above yielded a resolution of 500 pulses per revolution for each encoder. This is in agreement with the calculated resolution of the encoders found by monitoring the frequency of the reference pulse and the quadrature pulses when connected to an oscilloscope. The reference pulse had a frequency of 3.42 Hz, and the quadrature pulses had a frequency of 6.84 Hz. This resulted in a ratio of 500.3 : 1.

A point of note brought out by this experiment was that the limiting factor in high speed counting was found to be the lead or lag displacement between the two quadrature pulse trains.



5.2 DC SERVO MOTORS

Rotation of each axis is controlled by a DC servo motor, and thus the understanding of their operation is vital to the establishment of final control of the robot.

5.2.1 MOTOR CALIBRATION

On the robot are four DC servo motors, two sets having the same dimensions. Operational curves were obtained for these motors, relating input voltage and associated current to rotational speed. The curves obtained follow.

The data from the motors was obtained by taking six readings with a tachometer at discreet voltage levels. The voltage levels were varied from 0 V to +12 V, moving up and down the scale in order to detect any discrepancies. The effect of the torque from the harmonic drive was simulated as accurately as possible by holding the end bearing stationery. The effect of a load on the end bearing, as compared to a free standing motor causes more current to be drawn by the motor, and the speed curves [illegible] give lower rotational speeds while retaining their linearity.

The speed relationships are important when terminal motion of the robot is desired, such that each axis reaches its end destination at precisely the right time. However, external sources of torque may be present and can effect the performance of the motor. A varying load on the motor can cause it to act unpredictably, and is present in the simple example of the robot arm picking up a weight. As the arm is raised, the centre of gravity of the arm moves closer to the joint, causing a reduction in the torque applied to the motor. This must be kept in mind when input voltages are used to manipulate the robot arms at various speeds.

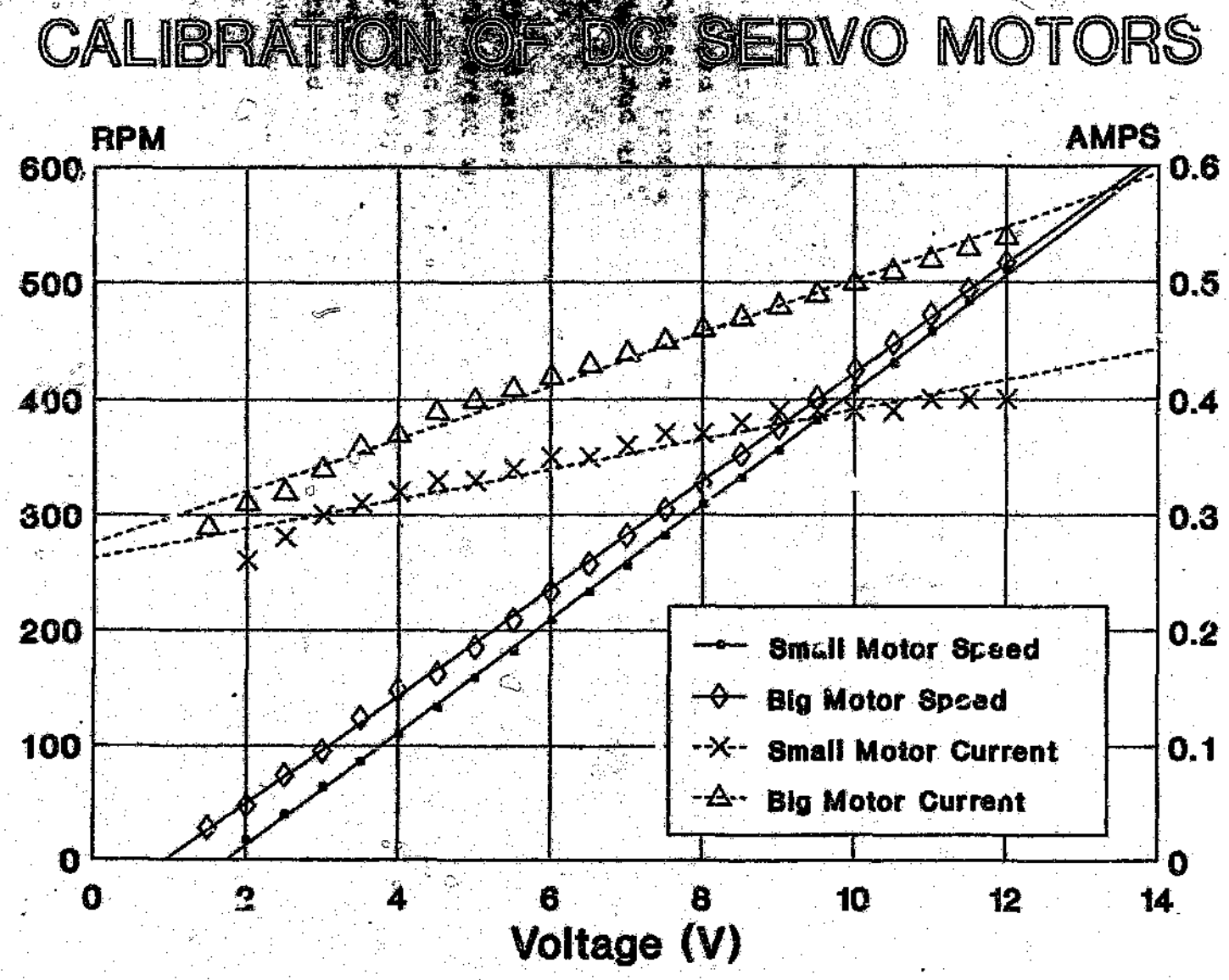


Fig 5.8 Calibration curves for DC servo motors

5.2.2 GEARING DETERMINATION

Once the speed of the motors have been related to varying input voltages, it is important to determine the way the harmonic drives are geared.

A practical approach to this problem was to measure the rotational displacement (in degrees) of the robot arm, and to relate it to the number of pulses the encoder has moved through. In this manner, the number of pulses per degree rotation of the arm was determined.

The encoder output of each motor was accessed through the PLC, and the motor was powered directly from a power supply. For all horizontal axes, where rotation of the arm is in a vertical plane, a protractor was placed vertically on the respective arm. An angular movement of the arm resulted in a similar movement of the protractor, and a thread with a weight on the end, suspended from the centre of the protractor resulted in an angular reading being obtained with an accuracy to within 0.1^{o} (see figure 5.1).

[illegible] rotating in a vertical plane required a [illegible] approach, and the following was used. A single [illegible] fixed to a horizontal plate anywhere on the [illegible]. A second thread was placed over the first, but [illegible] at each end to the walls of the room. Thus one [illegible] moved through the same angle as did the robot, while [illegible] thread remined fixed in one position (see figure [illegible]). Again the encoder output gave a pulse reading that [illegible] related to the change in angle between the two [illegible] eads.

This exercise was carried out 15 times for each axis, giving a realistic aggregate. Table 5.1 gives the pulses per degree and the gearing ratio (motor : arm) of each axis.

AXIS	PULSES/DEGREE	GEARING (MOTOR:ARM)
Z	224	161.3 : 1
X	220	158.4 : 1
β	134	96.5 : 1
θ	218	157.0 : 1

Table 5.1 Axis gearing ratios

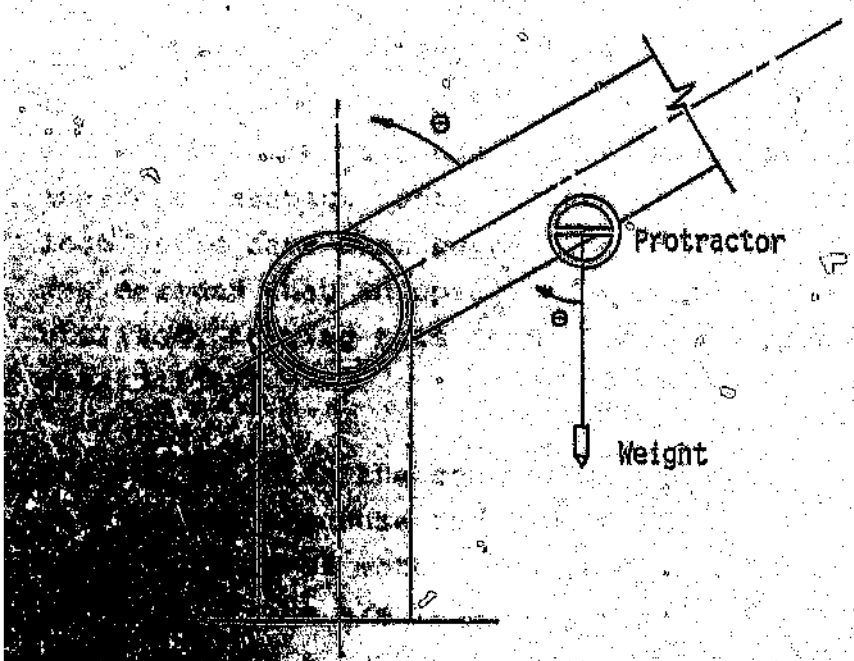


Fig 5.9 Gearing determination - horizontal axes

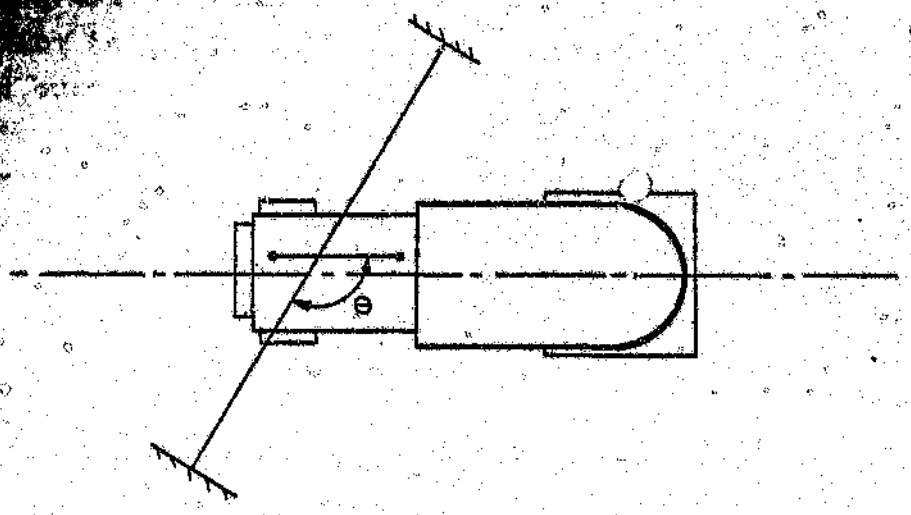


Fig 5.10 Gearing determination - vertical axis

5.3 CONTROL DECISIONS

Once it had been established that all the components in the robot system (i.e. motors, encoders, brakes, etc.) were in working order, a decision had to be made regarding the form the control would take.

The encoders were seen to be the key to control, as these could relate how many rotations the motor had rotated through, as well as the speed with which it had rotated.

5.3.1 OPEN LOOP CONTROL

Due to the fact that none of the existing modules had inherant feedback control loops, and the design of such a loop would have been beyond the scope of the project, it was decided that an open loop control system was to be [illegible] this form of control for its industrial

[illegible] the control of the robot was envisioned is [illegible] the resolution of the encoder and the [illegible] motor : robot arm is known, a desired [illegible] arm is simply effected by rotating the [illegible] a certain number of encoder pulses. When the [illegible] initiates the motion the brake is disengaged, and [illegible] desired position of the arm has been reached [illegible] to the encoder reading), the brake is then [illegible] once more.

[illegible] the control hierarchy would follow a certain pattern. [illegible] user of the robot would decide on and specify the [illegible] sition of the end effector. The controlling software (a controlling algorithm) would interpret and act upon this input.

In an open loop control, the operating characteristics of the various components must be known, thus allowing a known input to give one a predictable output. The DC motors

voltage - speed characteristics have been calibrated, and thus these speed characteristics of each motor can be utilised to allow terminal motion control. The desired speed the motor must rotate at to give the user terminal control is decided through a controlling speed algorithm that relates a desired speed to a certain voltage. The desired position of the robot arm is known, and this is related to the number of pulses the encoder has to rotate through. The controlling algorithm calculates the [illegible] displacement of the arm, making use of the [illegible] and the motor brake to initiate and [illegible] motion. As the encoder is incremental and not [illegible] the motor movement will have to be referenced [illegible] position.

[illegible] has to be taken into account, as the [illegible] arm will effect the position of the end [illegible] transition matrices have to be manipulated [illegible] algorithm to ensure the correct end

[illegible] has to take the desired end effector [illegible] into a movement for the controlled [illegible] movement into a rotational [illegible] and interpret this as a specific [illegible] pulses.

[illegible] study starts by looking at the inherent PLC [illegible] control, extends itself to incorporate a [illegible] counting, assessment of this system, and [illegible] desired system that will give adequate

5.4 I/O CAPABILITIES - COUNTER FUNCTION BLOCK

It was realised that some form of counting was needed in the control process, whereby the pulses of the encoder could be read, counted and stored, to be acted on when necessary. Starting from the simplest form of control, it was decided to experiment with the input/output (I/O) handling capabilities of the PLC, as well as to establish the basis for future counting control. This also presented an opportunity for the author to gain a practical understanding of the PLC.

Initially, Ladder logic programming was used, even though Graphcet options were available. It was anticipated that Graphcet would have to be used at a later stage for simultaneous motor control.

It was decided that a counting function block (CFB) would be programmed into the PLC, and input pulses generated by an external voltage source switched through a bank of toggle switches. The operation of a counting function block can be found in Appendix B1. Understanding of this section is essential for an understanding of the following paragraphs.

It was intended that this program and circuit should best model the control system, that is, an output signal driving a motor, and an input fed back from this motor (generated by an encoder) that would dictate the state of the motor, either running or not running. The simulation was laid out as shown below, with the corresponding program being used to simulate control.

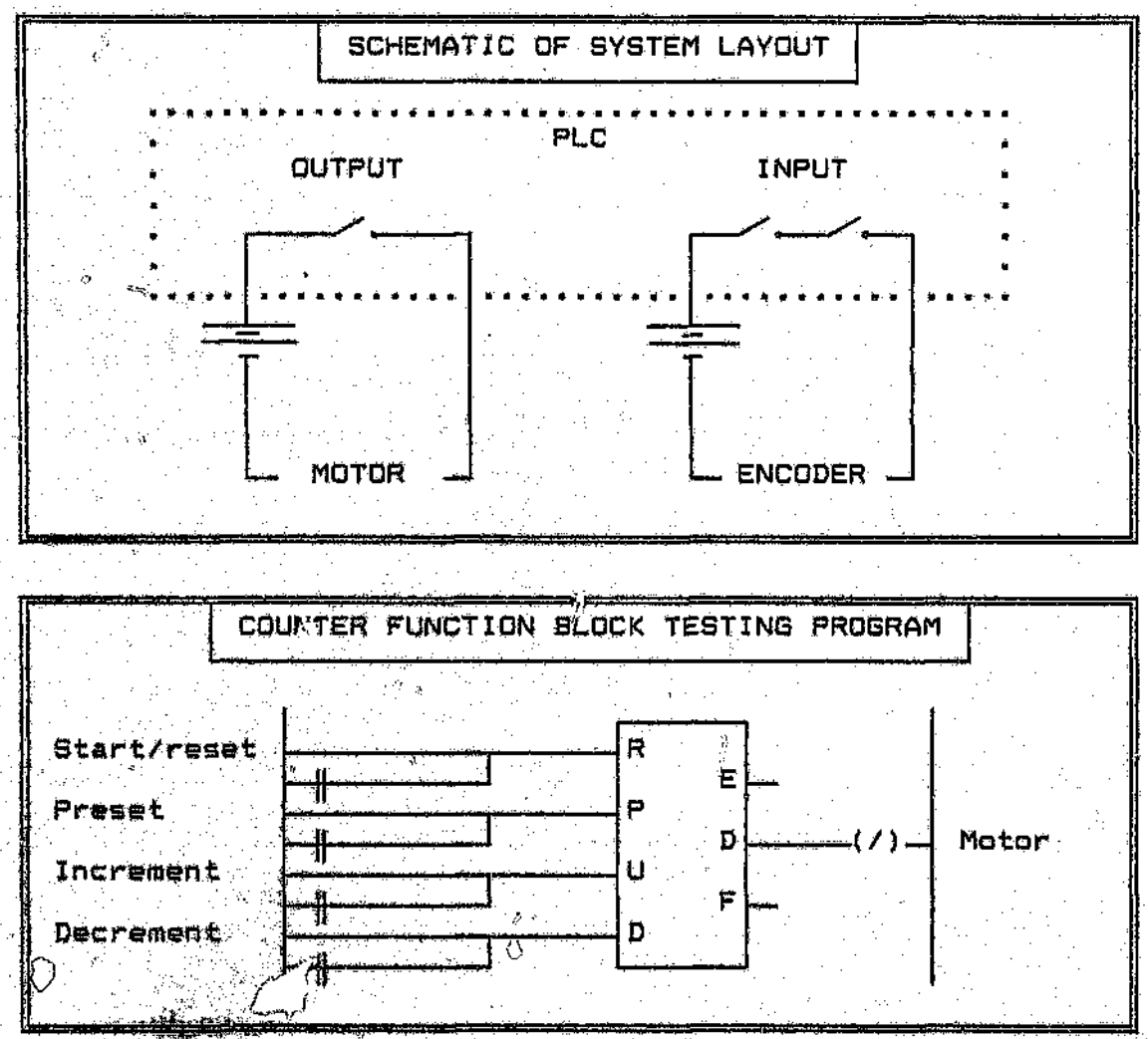


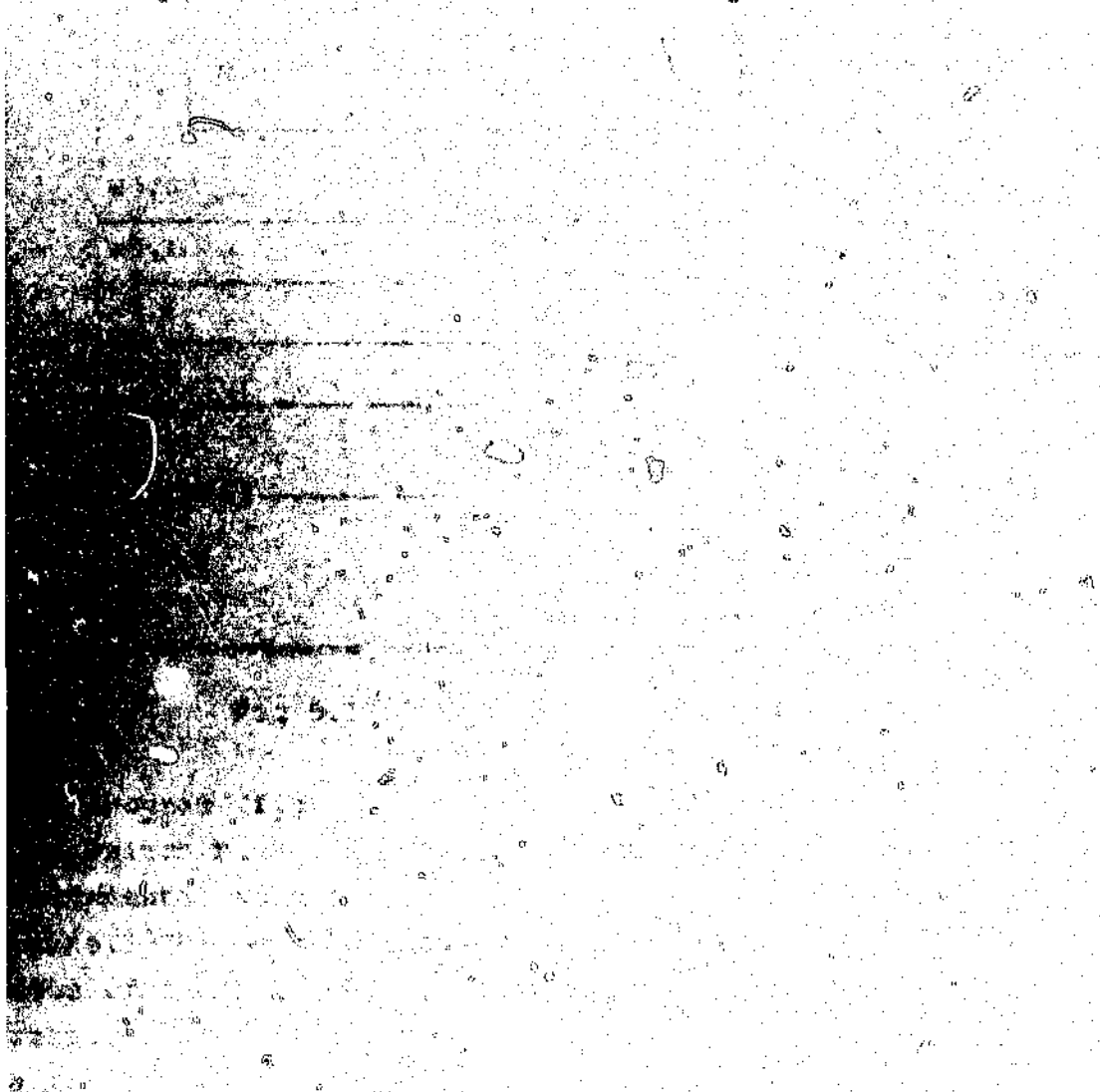
Fig 5.11 Component connection and control

The output signals from the encoder were modeled through the switch bank, i.e. the input read by the counter, indicating an incremental signal, was generated by the toggling of one of the switches on the switch bank.

The results from this rough model served to illustrate the futility of the standard I/O modules in fast input control. It was found that, in principle, the circuit worked perfectly with the motor running until a certain number of increments had been read. The problem was however, that these inputs were only being read at a speed of about 3 Hz. That is, that toggling the counting switch at a rate faster than 3 time per second was futile in that only the

occasional signal was acknowledged by the counter. This rate was far below the 43 Kz capabilities advertised for the PLC. Further reading indicated that the software function block itself was at fault in that it was incapable of running at this speed, and that the higher speed expected was only available on standard ladder logic programs.

Thus the counter function block had to be eliminated, as counting the pulses from the encoder required a counting rate very much more than the CFB was capable of.



5.5 ACCUMULATION OF AN INCREMENTAL WORD

The design of a program operating without the use of this restrictive function block was a natural progression, accumulating an incremental signal, and storing it in a word which could be compared to a set value.

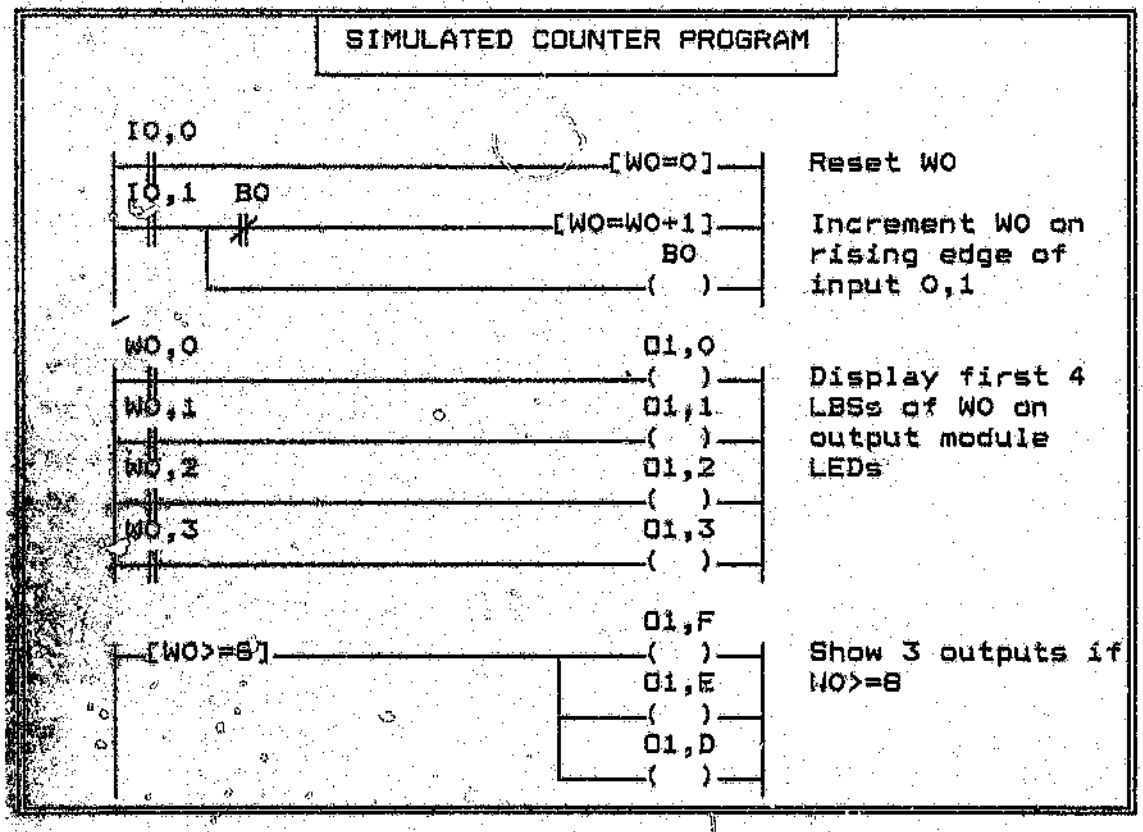


Fig 5.12 Program - simulated counter

The program (fig 5.12) allows for two inputs (both through the switch bank), the first a 'reset' that sets the incremental word to zero, and the second the incremental pulse. The program increments a word (W0) on the rising edge of input 2 (I0,1). This word is then displayed on the output module, for easy reference, before being compared to a preset value. The rising edge branch is necessitated by the fact that without this restriction the PLC will increment the word on every scan of the program when the switch is on. This means a 10 ms rate which cannot be accurately monitored in this simulation situation. If the

counting word proved to be higher than the preset value, three outputs (LEDs on the output module) were set.

It should be noted, at this stage, that the setting of an output on the PLC results in the closing of the output switch and the simultaneous illumination of an indicator, on the PLC panel, for that output. These indicators were used extensively throughout the following experiments in order to give immediate feedback on the program's state, as they eliminate the need for any attached, powered, indicators such as lamps or motors.

This program effectively simulates the counting block, with the same reset, comparative and incremental facilities being available. It could easily be enhanced to provide decremental and inhibiting/enabling inputs for completion, but for experimental purposes this was not necessary. The results showed that this program was indeed far faster than the previous one utilising the CFB.

5.6 I/O FACILITIES - FCM (UP COUNTING)

It was realised that a fast counting facility was needed to accomodate the speed of the pulse train from the encoder. With a resolution of 500 pulses per revolution, with the motor rotating at only 240 rpm, a counting capability of 2 kHz was necessary. Fortunately, this was a result that had been anticipated, and a compatible fast counting module (FCM) had been procured from Telemecanique. An explanation of this module, and its operation, can be found in Appendix B2. Once again, an understanding of this section is imperative for an understanding of the following paragraphs.

The investigation of the operation of the FCM formed an integral part of the feasibility study. The encoders emit a dual logic signal in the form of two square waves, with a phase displacement of 90^{0}, i.e. in quadrature. These waves lead or lag each other depending on the direction of rotation. The FCM which had been procured was ideally suited for handling this type of input. Its discriminatory capabilities also eliminated the need for separate incremental and decremental inputs, these now being distinguished from the difference in the two wave forms.

The first step in the feasibility investigation was repeated with the FCM and a modified program designed to handle the generated inputs to the module. The module was wired through the switch bank as shown in figure 5.13, and the accompanying program (figure 5.14) used to test its operation.

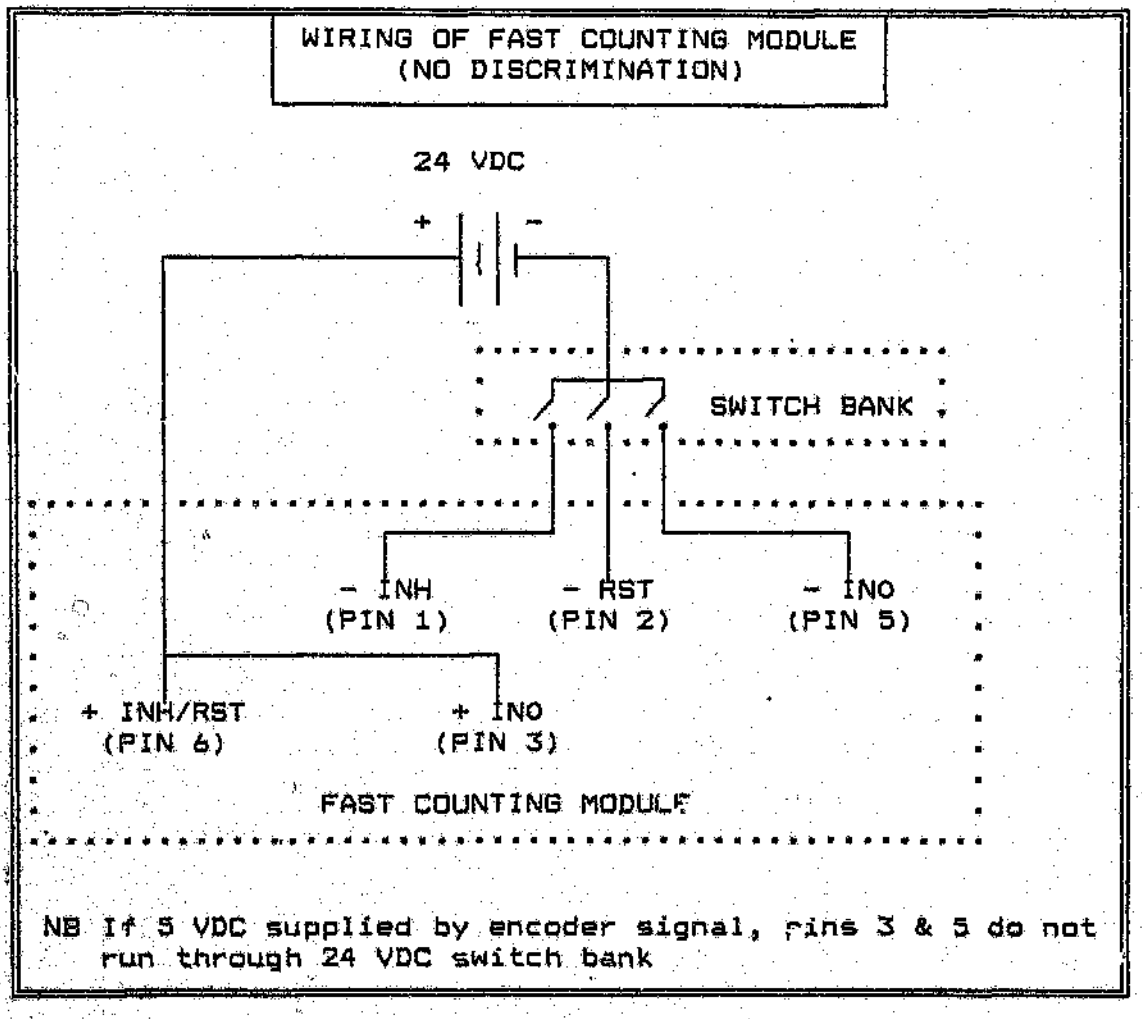


Fig 5.13 Circuit - wiring of FCM

Initially, the FCM was not set on discriminatory input, but was only capable of reading an incremental signal. All switch bank inputs were wired into the 5 V FCM input points, thus the program only accepted the addresses generated by this module i.e. I2,0 to 7 as explained in appendix B2. The program initially examines the state of the generated sign bit (I2,5) and sets the buffer word (W1) accordingly. The five buffer inputs (I2,0 to 4) are then down loaded into W1. As long as the sign bit is not negative (this requires careful handling in binary and can create confusion if allowed to wrap into the negative region), this buffer word is then added onto an accumulative counting word (W0). In the event of the reset

bit (I2,7) being high (due to an input from the switch bank), or if the sign bit goes negative, the accumulative word is set to zero. The last part of the program displays the state of the 5 least significant bits of W0 on the output module LEDs.

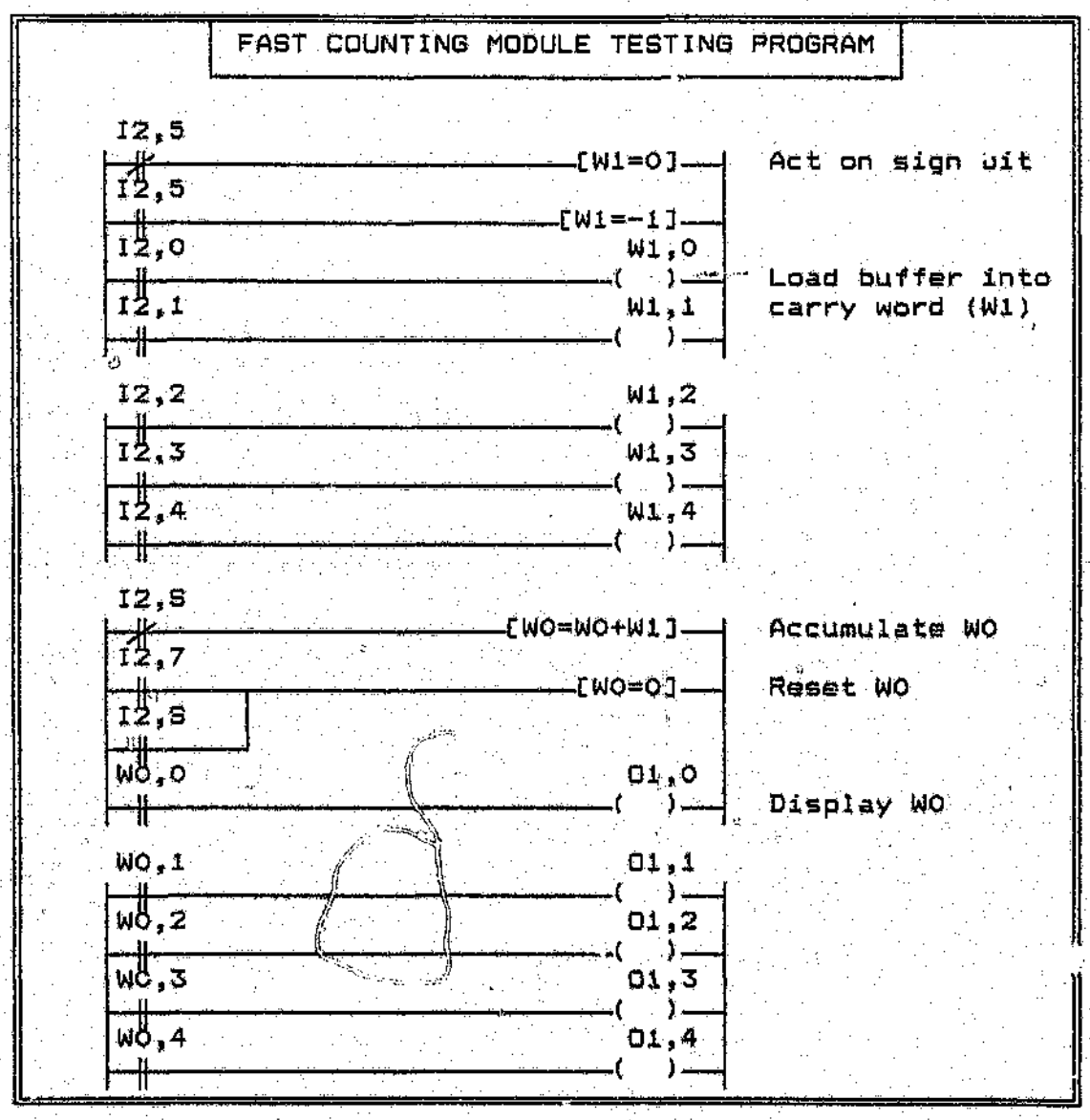


Fig 5.14 Program - FCM incorporation, upcounting

On a macroscopic scale, the FCM generates a number of inputs according to the three hard inputs it receives from the switch bank, these new inputs include the number of

incremental pulses received since the last program scan (10 ms ago) which are loaded into a carrying word which transfers them to the counting word. The process is then repeated. The reader will note that the carrying word is the same as the input buffer for each scan, and is thus a reflection of the counting rate. Thus by observing the constantly updated state of this word in debug mode, the input rate can be accurately monitored.

The results indicated that the module was working exactly as predicted. A problem that arose was that with each flick of the toggle switch, W0 was increasing by up to three units. This was due to switch bounce which was now detectable with this high input detection resolution. When W1 was monitored in debug mode, it was found that extremely fast toggling of the input pulse switch could only just cause the word to reflect any state. This indicates that the buffer is being down-loaded and reset very much faster than the input speed of the counting pulse, which was as predicted.

5.7 I/O CAPABILITIES - FCM (UP & DOWNCOUNTING)

It now remained to test the FCM in its discriminatory mode. Once again, only a single input channel was used, the other only needed for simultaneous control. The following circuit (figure 5.15) was set up, this being a modification of the previous model to allow for an additional input designed to simulate the second pulse train.

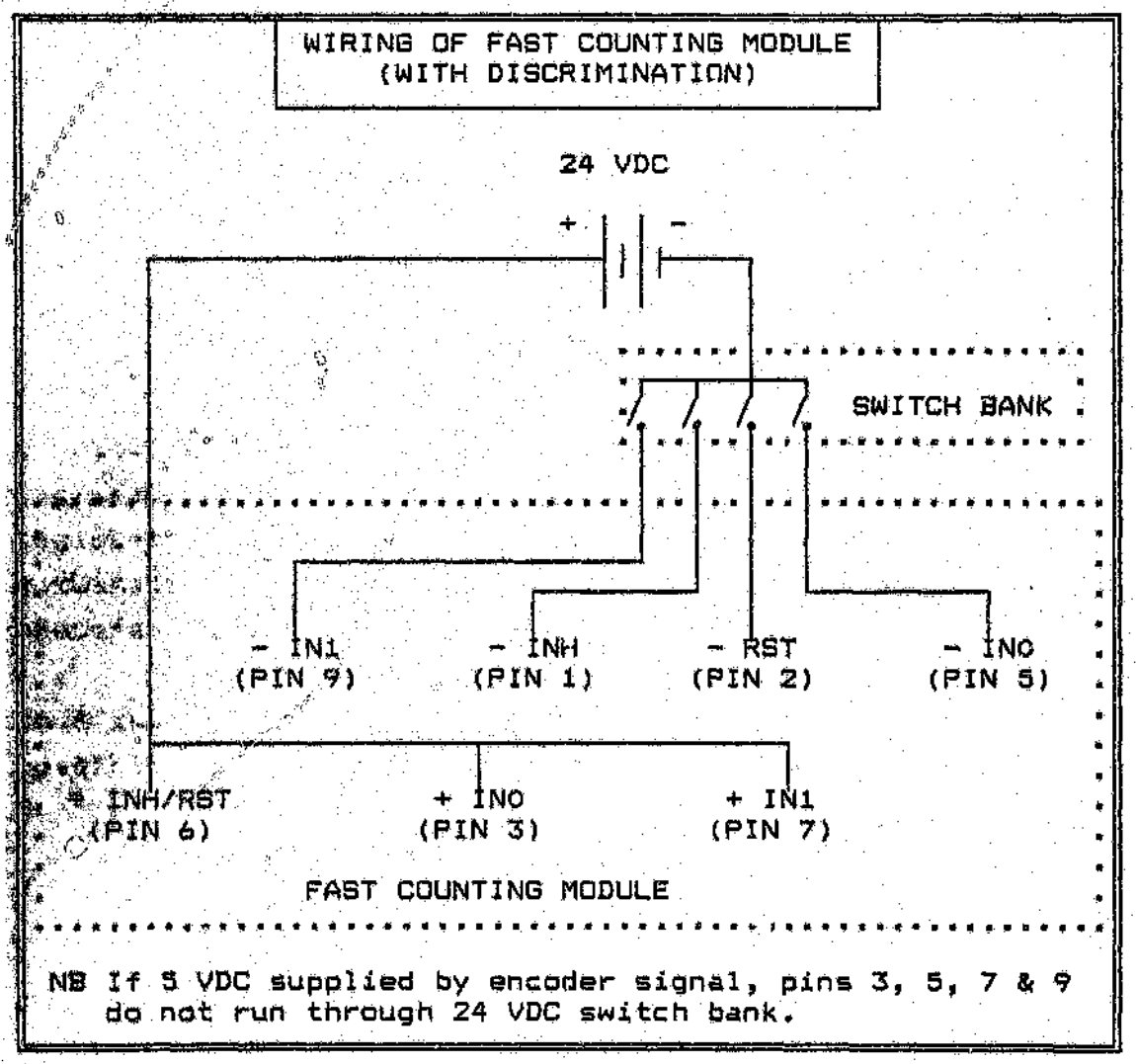


Fig 5.15 Circuit - FCM incorporation, up and downcounting

No further programming is required to handle dual logic inputs, all this is taken care of by the FCM with results being stored in the internal register.

By toggling the two input switches in succession (1 on, 2 on, 1 off, 2 off, etc), a very slow square wave train (with lead on input 1) could be simulated. This resulted in up-counting. By reversing the order of switching, down-counting could also be achieved. Switch bounce was not a problem in this case, the reason being that after the first rising edge, the discriminating module would look immediately to the other input for a rising edge and did not, therefore, 'see' the multiple edges from the first input.

The advantage of using this form of input control, was that it became possible to detect a transition phase between up and down-counting. This transition requires control in order to prevent the loss of a count between lead and lag transitions, which would result in a loss of resolution.

Experimental results yielded certain relationships:
If a lead/lag set has occurred, then the next lead/lag combination will result in the respective count, however, a lag/lead set will effect no change in the FCM's internal register. The next effect is dependant upon the last input combination being its base indicator as in an iterative process.

This relationship is illustrated below, covering all possible combinations of consecutive lead/lags:

INPUT CHANGEOVER	INTERNAL REGISTER RESULT
Lead - Lead	Increment
Lag - Lag	Decrement
Lead - Lag	Constant
Lag - Lead	Constant

In addition to the above result, it was noted that incrementing or decrementing of the internal register took place at a specific position in the duty cycle combination of the two wave forms. This was dictated by input IN0 on the module, with incrementing on the rising edge, and

decrementing on the falling edge (see figure 5.16).



Channels	Ch0	Ch1							
Counting bits	Ixy,4	Ixy,C	0	0	0	0	0	1	1	0
	Ixy,3	Ixy,B	0	0	0	0	0	1	1	0
	Ixy,2	Ixy,A	0	0	0	0	0	1	1	0
	Ixy,1	Ixy,9	0	0	1	0	0	1	1	0
	Ixy,0	Ixy,8	0	1	0	1	0	1	0	0

The maximum values that can be coded on the five counting bits are + 31 and -32.

Figure 5.16 Up and downcounting wave trains

5.8 HIGH SPEED COUNTING

The next stage in control feasibility development required the establishing of a program that could handle counting in the large regions that this control required. So far, counting control had only been able to handle up to 32767 pulses (2^{15}) in the 16 bit-word (NB the 16^{th} bit is a sign bit). With the encoders being used, this only allows for abo ± 65 revolutions of the motor, not nearly sufficient for the robot's range of movement. A program had to be established that would augment this range by coupling two words to give a 30 bit word (2 sign bits not included). This gives a range of 32768^2 (or 2^{30}), or over a million revolutions of control.

Another option for high revolution counting would be the use of the encoders reference output. As described earlier, the encoder outputs an additional pulse for every revolution. This could be used to break counting down into revolutions and fractions of a revolution (from the normal quadrature inputs). The 30 bit word option allows for far greater scope in fast counting.

A program was written (figure 5.17) that increments word W0 from the FCM inputs and sets two output indicators if the word is smaller than a constant word (CW1). This is how final comparison will take place, with CW1 set from an external source, either from a PC or a PLC.

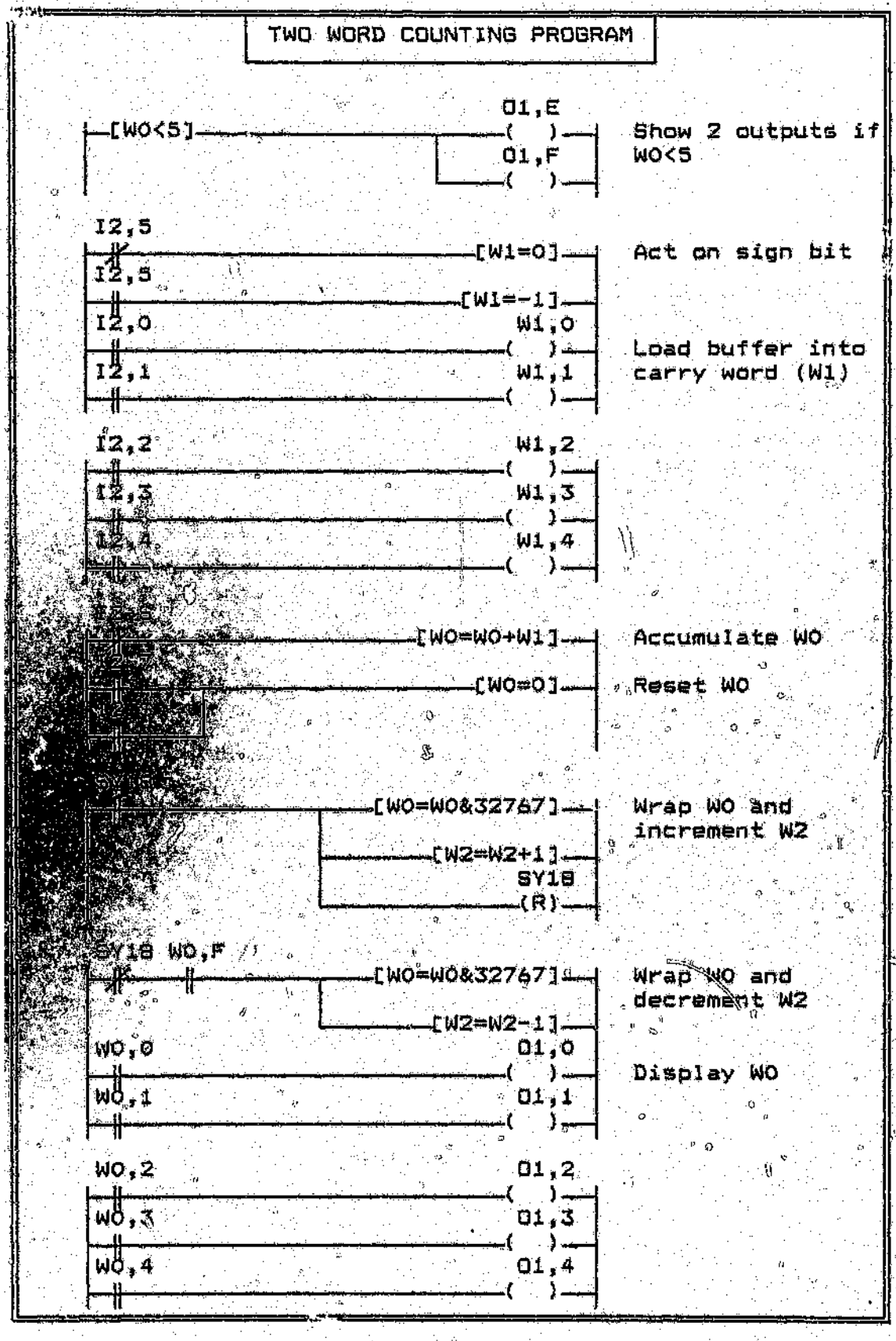


Fig 5.17 Program - FCM counting simulation

The program uses word W1 as a carrier for W0, and W2 is the augmented word used for enhanced scope as mentioned previously. By using the state of the system bit (SY18), an indication of word overflow error, W0 is made to wrap between 32767 and 0, with W1 being incremented or decremented for each wrap, depending on the direction of wrap.

The comparison block can be doubled up to cater for the two word number by placing two blocks in series, the one checking the more significant of the two words, and the other the lesser.



5.9 FCM READING CAPABILITIES - FREQUENCY GENERATOR 1

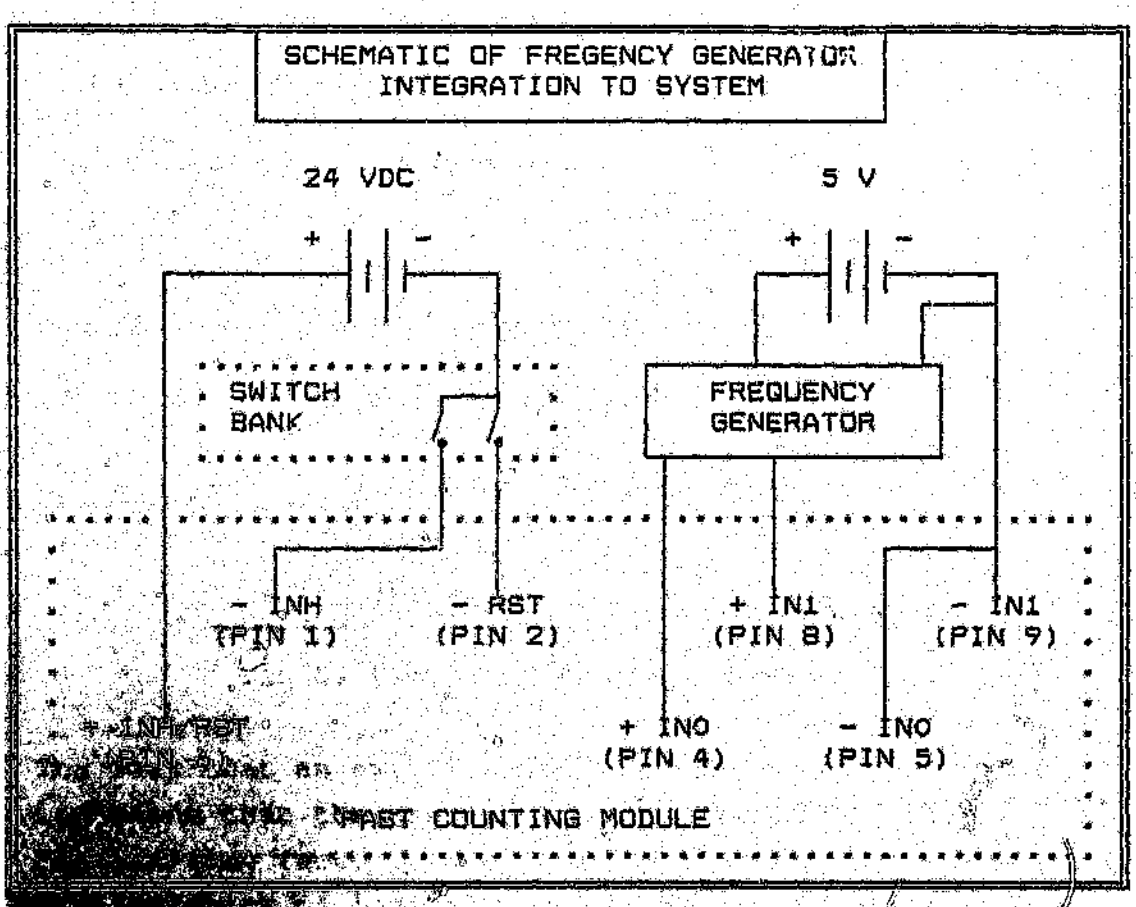
Now that the exact operation of the FCM, in both modes, had been determined and its operation in monitored counting established, the next step was to test its reading capabilities. It was decided that this should be done using a frequency generator with output frequencies that could be varied and determined. Note that all the previous steps, as well as the inclusion of a frequency generator into the feasibility study were necessary, as the encoders had, up to this stage, not been repaired, and thus their outputs had to be simulated.

A [illegible] square wave generator could not be [illegible] purpose, so one had to be built.

It was intended that this generator be ideally suited for [illegible] made feasible by the introduction of an enable switch (simulating motor start-up), variable frequency control (to simulate motor speed), and with a lead/lag option (to simulate bi-directional motor movement).

The frequency generator, as described earlier under [illegible] and explained in detail in appendix B3, was [illegible] and wired into the PLC as shown in the following diagram.

The program used to test the generator was exactly the same as that used in the previous testing phase. However, the [illegible] of the first four bits of W0 was removed.



Circuit - Frequency generator connection

By observing the state of W0 (in debug mode) it was possible to observe the correct up/down-counting. However, the more important, and relevant, indication of the operation of the FCM could be obtained by observing the state of W1, the carrying word. As mentioned earlier, this word gives an indication of the rate of counting, in either direction, and its state would be the first sign of the FCM having reached one of its limiting factors.

The counting speed was increased by varying the frequency range of the generator. This had to be done slowly as the sampling interval of the debug update meant that immediate effects in the system might not be reflected in W1 as fast. At a particular speed, the size of W1 dropped off sharply indicating that the module was not detecting every single

input pulse set. A few Hertz further on, the counter stopped detecting any input pulses whatsoever. By connecting the circuit up to an oscilloscope, the unstable point was found to be 2.72 kHz, with a wave period of 370μs, and a phase difference of 60μs (or 58^{O}). (See figure 5.19.)

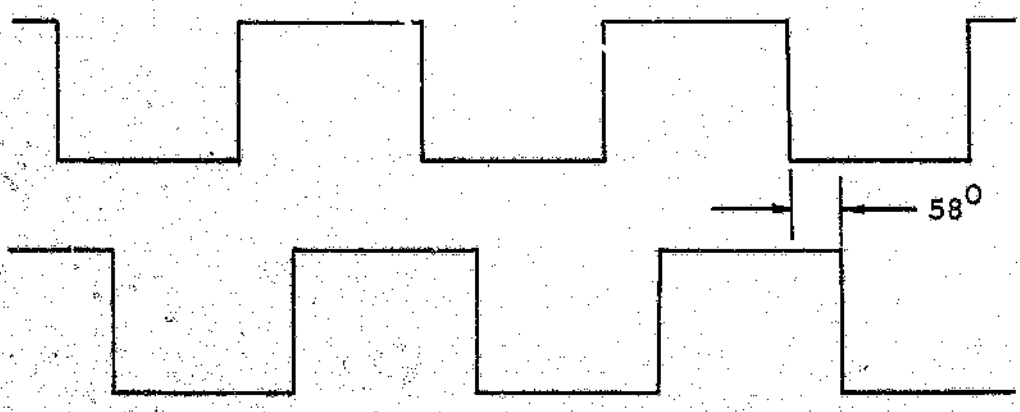
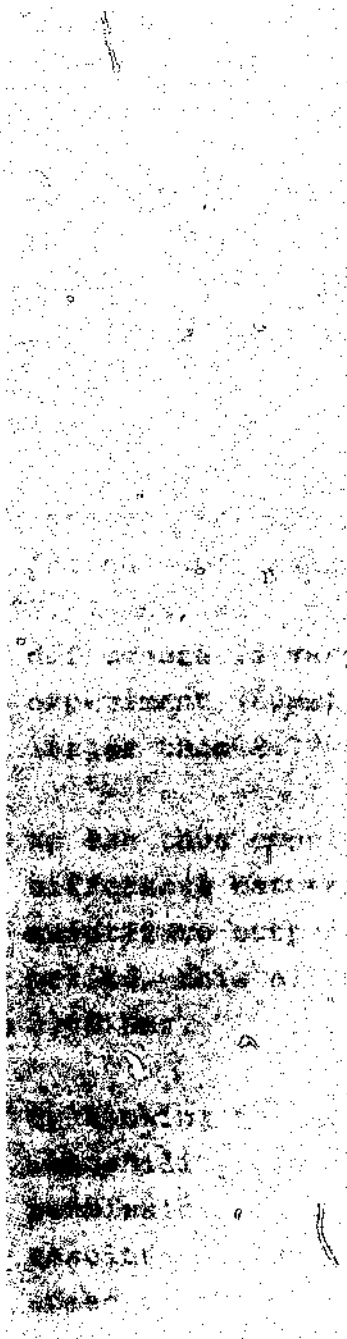


Fig 5.19 Timing diagram of pulse train

The fact that an overflow had not occurred at this speed indicated that the scan rate versus the buffer size was not the limiting factor. W1 was peaking at about 18 (averaging at 16), which is indicative of a number of things. Firstly, the limitation associated with the FCM's bucket accumulation effect can be exactly monitored. At this input frequency, the FCM accumulates the counting word in steps of 18 units, and thus counting resolution is reduced 18 times. Secondly, if input frequency is not the limiting factor, it would seem possible to increase it by a factor of 1.7 (31/18) before the buffer size became the dictating limitation in counting, which it will do when W1 starts to peak at 31.

The only two remaining explanations for the inability of the FCM to detect the impulse trains at this high frequency, are that the wave period is too short for the module to distinguish, else the phase difference is too small to detect. The oscilloscope also revealed that at this frequency the 555 timer limitation of uneven duty cycles was indeed presenting a problem. As this would have a serious effect upon the results, it was decided to modify

the circuit with the addition of a more accurate timer chip. This comprised the next step of the feasibility research development.



5.10 FCM READING CAPABILITIES - FREQUENCY GENERATOR 2

Now that the feasibility and usefulness of a generator had been ascertained, it was decided to build a final model that would conform with all the aforementioned requirements of such a generator. Thus, the circuit which followed was capable of true quadrature output, frequency variability, and switchable states. The layout and operation of this second circuit is mentioned in appendix B4.

By increasing the frequency input to the FCM, and monitoring W1 as in the previous case, it was possible to observe the limiting frequency at which correct operation ceased. The results were very similar to the previous case with counting ceasing due to the FCM's inability to detect the inputs.

Analysing the results, it was seen that the lead/lag interval was the limiting factor. The results of the second experiment yielded a maximum input frequency of 3.67 kHz, with a phase difference of 69μs. This phase difference is very close to that obtained in the first experiment (60μs), whilst the frequency is substantially larger than 2.72 kHz.

We can thus conclude that the minimum permissible phase difference between the two wave forms is 65μs. In true quadrature output, where this difference is 1/4 of the wave period, this allows for a maximum input frequency of 3.85 kHz.

By knowing the limitation of the encoder output readability, it becomes possible to define the maximum permissible motor speed as related to the encoder resolution. By obtaining curves for each motor relating speed to input voltage, it becomes possible to place upper and lower limits on the voltage supply to these motors, this being vital for accurate open loop control.

5.11 OPEN LOOP CONTROL ACCURACY

As mentioned previously, the choice of open loop control was initially chosen as none of the existing PLC modules had inherent feedback loops, and the existing modules were expected to perform the task without any detrimental effect to the control.

Determining the efficiency of the control loop is essential, as this will determine if the proposed system is feasible from an industrial point of view. The proposed method of control was through controlling each motor individually, from a power supply and a PLC module. By controlling one motor successfully, the control could ultimately be extended to all four motors on the robot. Thus the accuracy of a single motor was assessed first, before the control loop was extended to the entire robot system. This is desirable, for, if the control loop proves infeasible, time and money would not have been wasted on control development that was worthless.

The loop study required a single DC motor to be connected to the FCM of the PLC. By running the program that follows (figure 5.20), it was possible to activate the motor (deactivate the brake and supply power), and by reading the signals received from the encoder, stop it by applying the brake when the desired rotational position had been reached.
Initially, the program sets the outputs to both the motor and the brake when the counting word (W0) is smaller than 5, i.e. it has just been reset. The reset input on I2,7 resets W0, and initiates the counting. This input is reset with a switch on the switch bank, and not through the reference pulse of the encoder. When the motor starts rotation, counting of the pulses is initiated. When word W0 reaches 3000 pulses, the motor power is cut, and the brake applied simultaneously. Thus, theoretically, the motor should have turned through 6 revolutions.

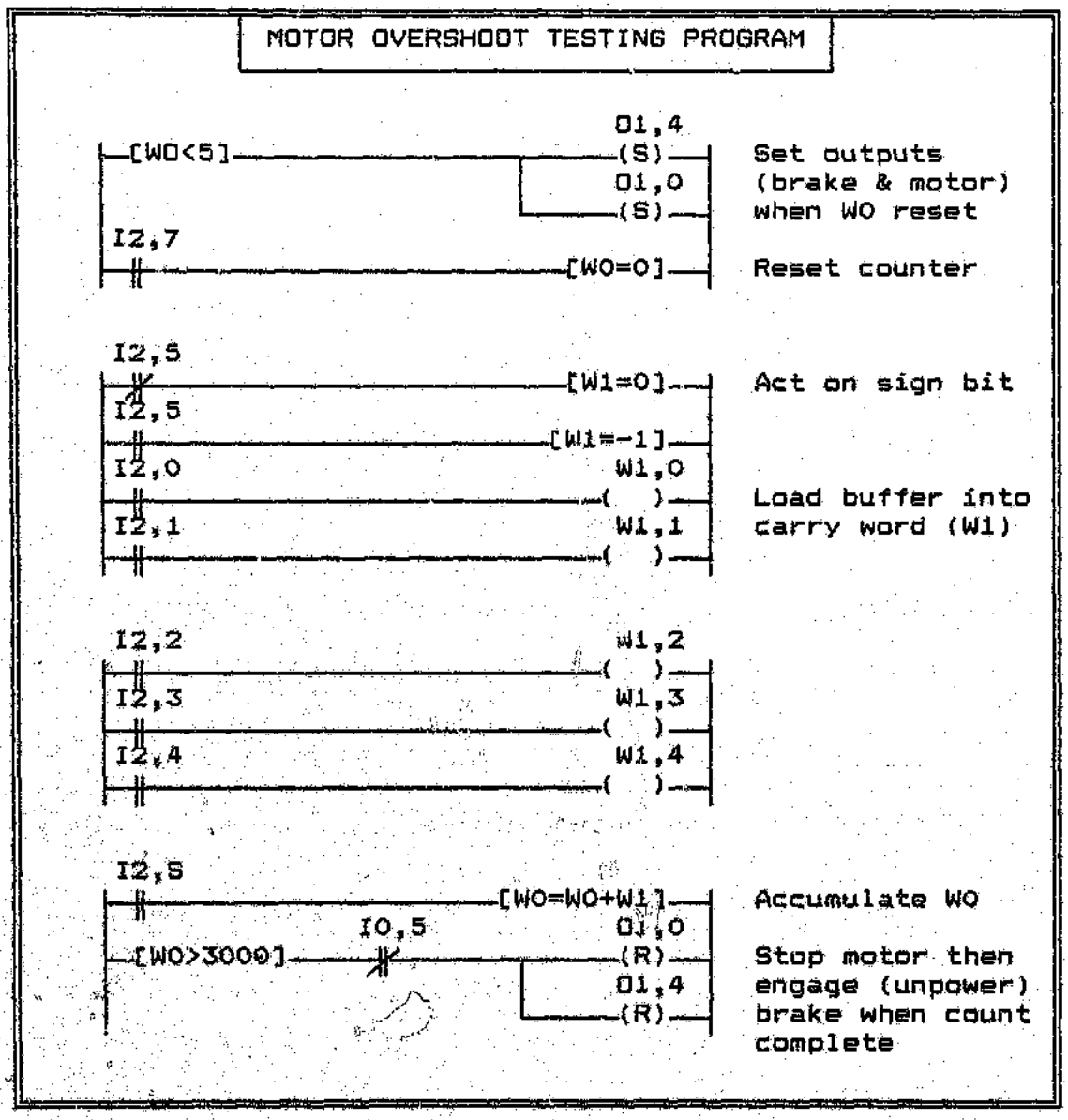


Fig 5.20 Program - motor control accuracy

By observing the counting word W0 in debug mode, the error, if any exists, can be compared to the required 3000 pulses. An error was observed, this always being greater than the desired 3000 pulses, indicating that the motor was overshooting the stopping position. At low speeds of the motor the overshoot was low, increasing as the rotational speed increased. The reason for the overshoot can be attributed to the momentum of the motor carrying it beyond the point of stopping, as well as a delay in the reaction of the activation of the electromechanical brake.

The curves (figure 5.21) of the overshoot for the two sizes of motors follow. The curves also give an indication of the bucket counting effect error at [illegible]ase voltages, the difference in these values being the minimum error that can be attributed entirely to the momentum of the motor.

A problem envisaged with this overshoot is that this error can behave in an unpredictable manner. That is, if the robot arm is lifting a weight up, the torque applied to the motor by the robot arm acts opposite to the torque applied by the motor to the arm, and thus a smaller overshoot error can be expected. When the arm is putting a weight down, the torque applied to the motor by the arm is in the same direction as the torque applied by the motor to the arm, and the overshoot error can be expected to be larger. This error is complicated by the many positions the robot geometry can assume in carrying out a task.

If a task to be performed by the robot is allowed to run with this overshoot, a cumulative error for each motor will result, giving an undesirable final position of the end effector. A form of feedback compensation of this overshoot can be developed, i.e. the overshoot can be read, and the motor can be instructed to move back the number of pulses the motor overshot the desired mark. By performing this task a number of times, the overshoot error will eventually be reduced to zero, or very close to zero. If the overshoot can be anticipated from the graphs, the brake can be commanded to activate that much before the motor has reached its final position. However, the loading on the robot arm complicates this issue, making the overshoot estimation a mere guess, and the corrective process described above would have to be repeated to ensure accuracy. Regardless, terminal motion of the robot arm will take much longer using this 'correction process' than if the arm moved to the correct position the first time.

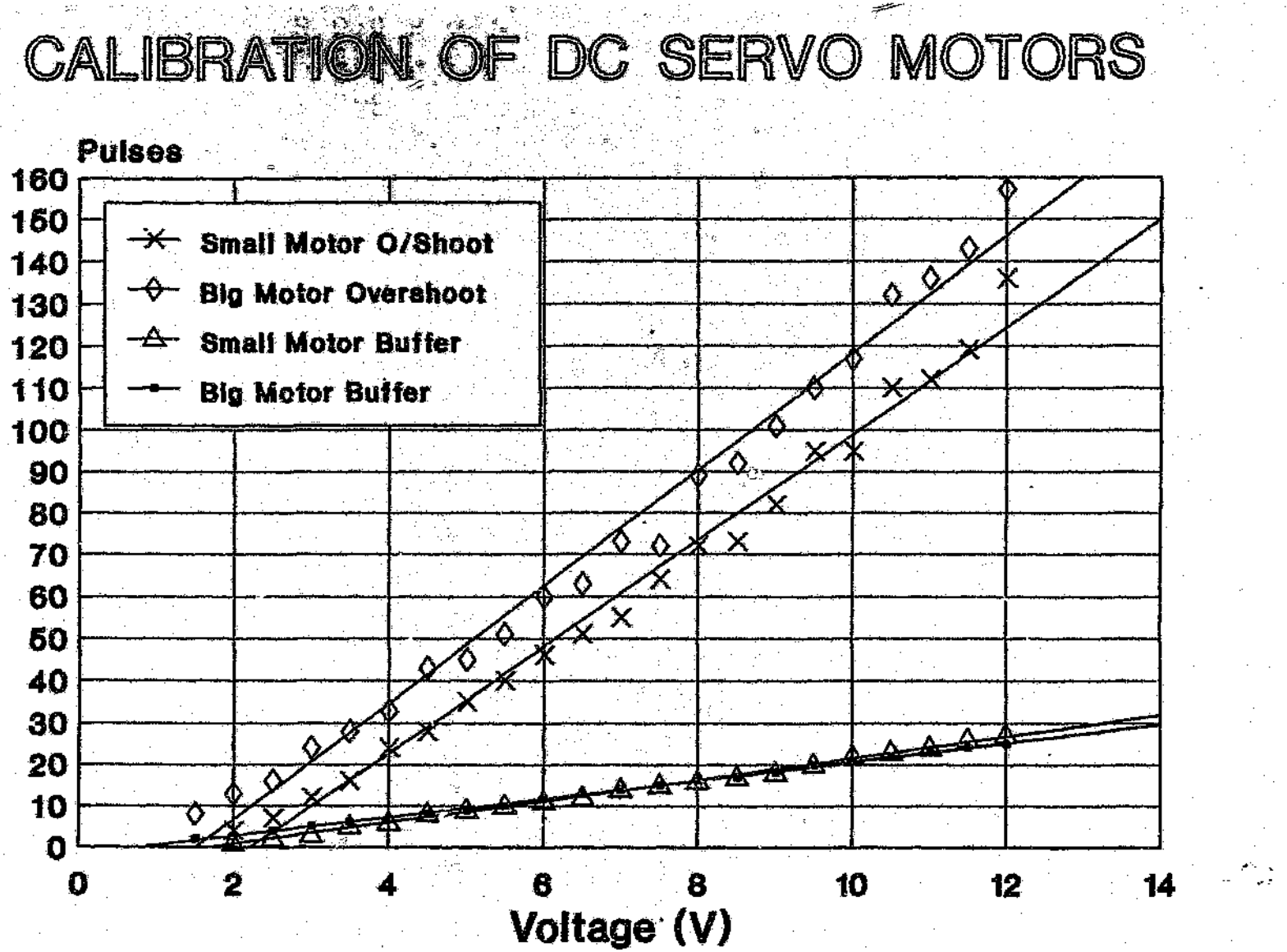


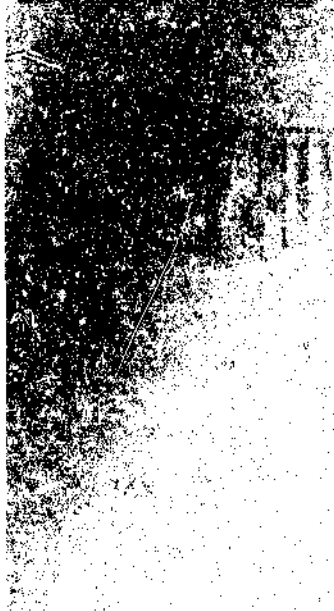
Fig 5.21 Motor overshoot of DC servo motors

Another feature that is difficult to incorporate into an open loop system is speed control. Speed can be determined in a number of ways. The time interval between two successive pulses from the encoder can be measured. Another method would entail the counting of pulses within a certain time period. A third method, as described in section 2.11.2 regarding closed loop control makes use of a rotational tachometer, but this is not feasible as the motors on the robot are not equipped with such devices. A closed loop control system would be able to determine the speed the motor is rotating at, and adjust it according to some predefined controlling algorithm. In an open loop system, a predetermined voltage, relating to a certain motor speed will have to be specified, without any dynamic adjustment to the speed.

For open loop control to operate accurately, the response of the system to some input signal must be known one hundred percent, so that the output is that which is expected. In the case of a robot whose motions are controlled by DC motors, this gives rise to certain problems. Data relating to the operation of the motors is unreliable. The effect of external loading on the motor makes its operation unpredictable, and indeterminable factors such as the effect of temperature, age and humidity on the motors operation makes the idea of an open loop control system undesirable. Some form of closed loop control looks more attractive the more is learned about the open loop control envisaged for the robot.

5.12 EXISTING ROBOT CONTROL

The open loop control discussed thus far does not include any form of load or error compensation. The robot controller used at present employs closed loop control. When the robot arm has reached its desired position, the arm is held in place by an opposing torque applied to the motor. In this way the arm is held fixed, and in the case of an uncalculated load on the arm, the closed loop control system senses this load and applies an opposing current to the motor, causing the arm to remain fixed. Incorporating load compensation into an open loop system (without making use of the brake) is impossible, and is even extremely complex in a closed loop system. However, this will have to be looked at if successful control is to be achieved. The brake is only engaged when the power to the robot is interrupted, and is not used, as was envisaged in the open loop control system, for stopping the motor when it has reached its required position. This suggests that the method of braking the motor when it is to be stopped is not as was designed by the robot designers, and excessive braking may even cause damage.



5.13 AXIS CONTROL SYSTEM

From the disadvantages concluded by studying a form of open loop control for the robot, and due to the fact that the existing robot controller uses a closed loop control system, it was decided to investigate the possibility of additional modules for the PLC that had some form of inherent feedback, utilising closed loop control.

The existence of a module supplied by Telemecanique, although not yet commercially available, became known of during the course of the feasibility study, an axis control system. This system, at first, seemed as if it could solve many, if not all of the problems encountered with open loop control, and thus was investigated in more detail.

5.13.1 AXIS CONTROL PRESENTATION AND APPLICATION

The TSX AXM 172 module is designed to operate as an individual module in the PLC rack, and interfaces with peripheral devices as depicted in figure 5.22.

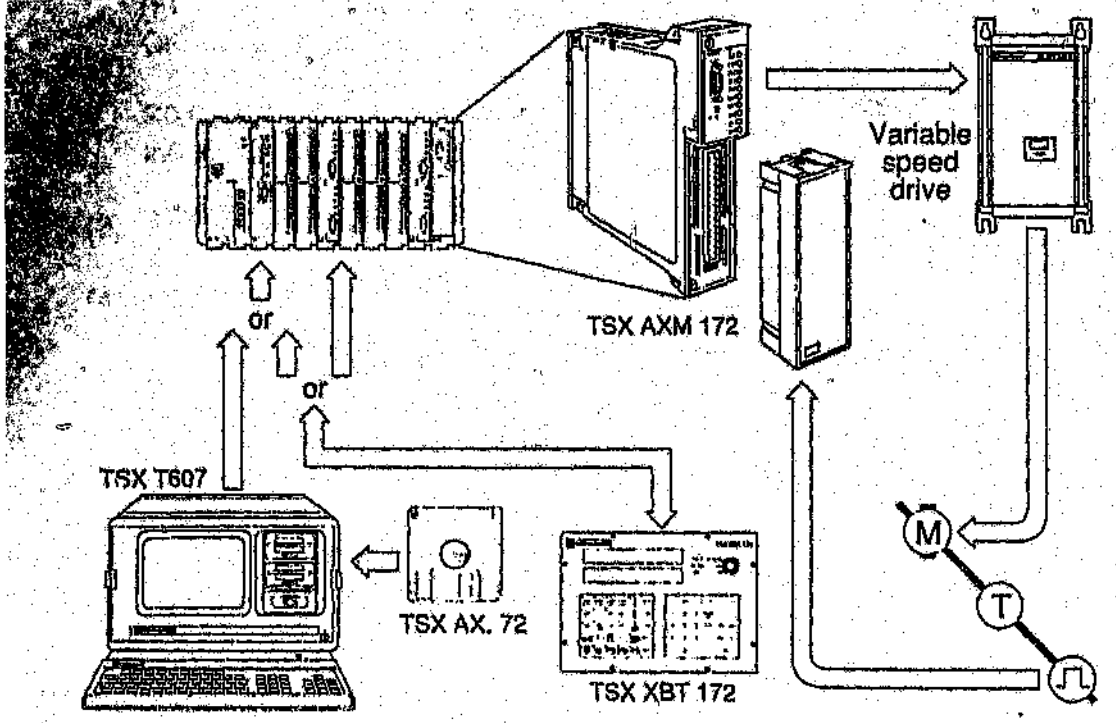


Fig 5.22 Axis control presentation

The TSX AXM 172 module is for use with TSX 47-30, TSX 67 and TSX 87 PLCs running software version V3 or higher. This programmable axis control module is designed to control a variable speed drive that determines the motion of a moving part along a linear axis with servo loop position control, according to the instructions given in the user program. This concept is referred to as axis control, and can be adapted to a rotational system through the translation of linear coordinates into circular coordinates. [45]

The TSX AX 72 software is used to program and debug the module, operated through a TSX T607 terminal.
A TSX XBT 172 operator terminal can be connected to allow operating and adjustment functions.

The TSX AXM 172 is presented as shown. It is a single height module, comprising a number of parts.

(1) a protective enclosure,

(2) 10 indicator LEDs:

(F)	module failure,
OK	module power on and operating correctly,
R0 to R2	relay outputs active,
In0 to In4	inputs at state 1,

(3) a connector for the analog output,

(4) a connector for all encoder inputs and the 4 relay outputs,

(5) a removable TSX BLK4 terminal block equipped with 32 screw terminals,

(6) a 9-pin subminiature D-type connector TSX CAC 04,

(7) an adjustment potentiometer for the analog output offset level.

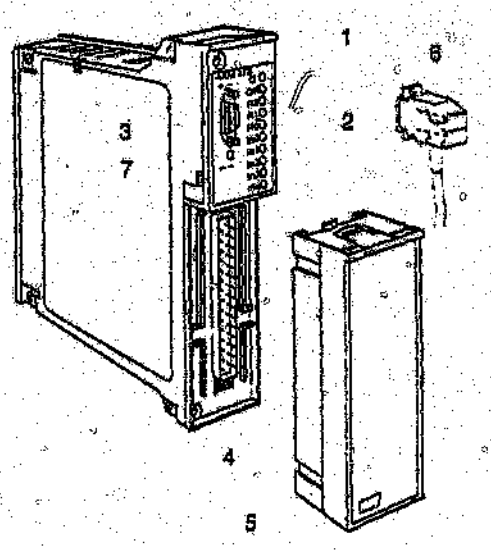


Fig 5.23 Axis control module

The internal layout of the module and the way in which it is to be connected is shown below.

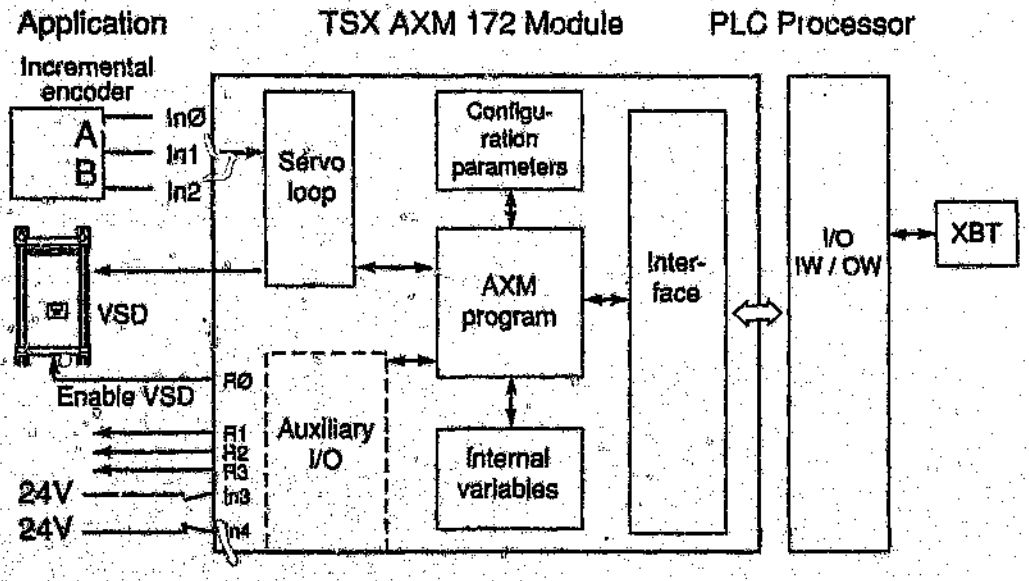


Fig 5.24 Internal layout of module

The module incorporates five main functions:

- protection of the application programmed by the user;
- generation of motions;
- moving part position servo control;
- interface with the application, and;
- interface between the PLC and an XBT programming terminal.

It is intended that the PC on-line interface will replace the XBT terminal. Although programs are intended to cover a complete repetitive cycle (see figures 5.25 & 5.26), it is hoped that through a PC link continuously updating commands, sequential, non-repetitive control will result.

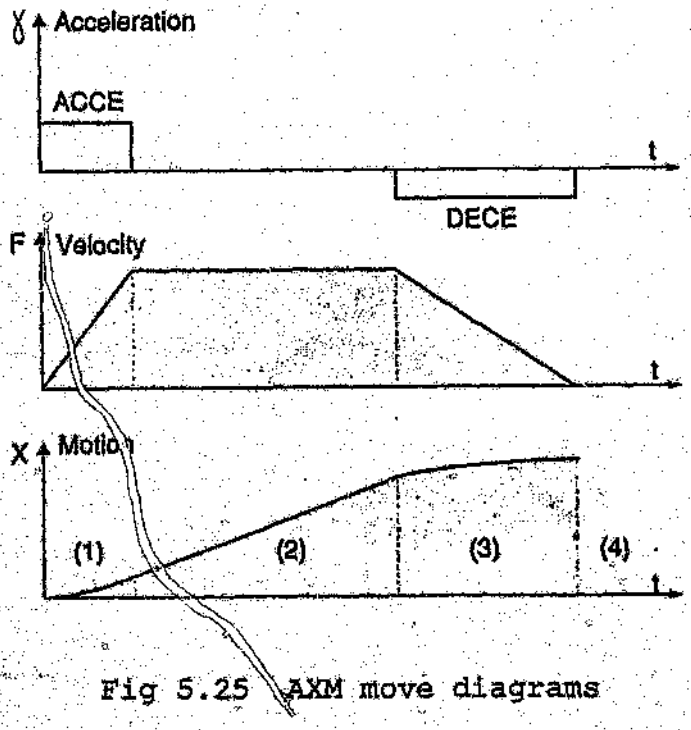


Fig 5.25 AXM move diagrams

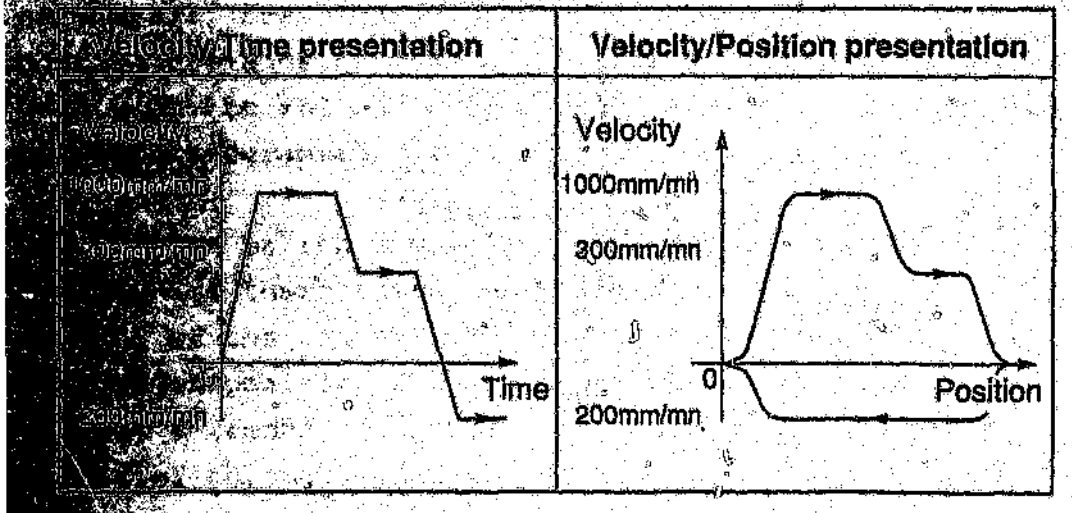


Fig 5.26 AXM positioning curves

The module facilitates setting up of the hardware through a pre-programmed reference set up mode in which the system is instructed to move until a home position has been reached. The robot's limit switches can easily be used for this purpose.

5.13.2 AXIS CONTROL SPECIFICATIONS

The module caters for a counting rate of 36 kHz (compare with the fast counting module's advertised counting rate of 2 kHz). The module advertises a proportional feedback positional compensation servo loop, requiring only three inputs from an encoder, and allowing for an additional two inputs catering for event detection (i.e. cam, home position, and points, etc.) and a variable speed drive safety interlock signal. It offers a single analog output, i.e. can only control one axis per module, and four auxiliary relay outputs with one being assigned to the variable speed drive safety interlock input.

For a complete specification list, see reference 45.

5.13.3 AXIS CONTROL OPERATION

Programming the TSX AXM 172 module is through simple user [illegible] these being converted into sophisticated motion control commands through the software package. As is the case with the PLC and programming terminal, once the module has been programmed, it can operate independently.

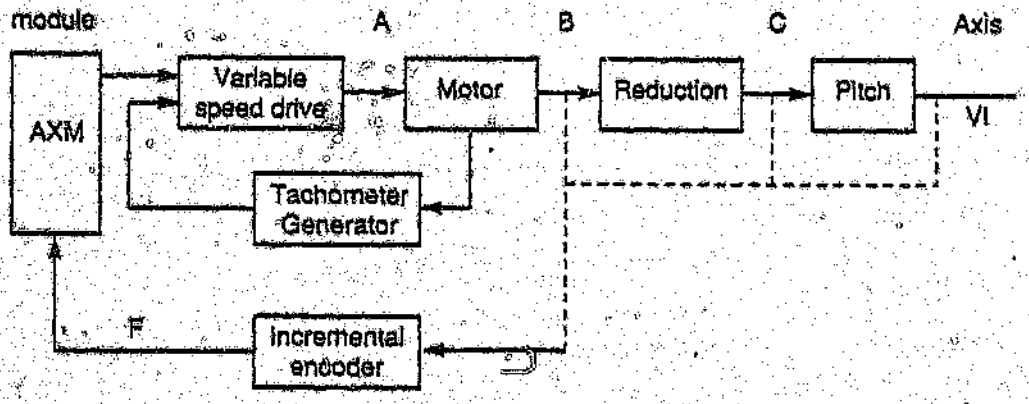
[illegible] can be run simultaneously, and instructed to synchronise their operations (done to facilitate terminal control of the robot). Requiring only an encoder input, from which position and speed are determined, they can produce a feedback controlled DC output used to feed an amplifier through which the motor is driven.

From controlling parameters (input by the user), velocity curves are generated from which output voltages are generated. This allows for accurate positioning, minimum overshoot (deviation of the control system from its required setpoint before the control oscillations settle down to the acceptable limits), and minimum dynamic loading on the system caused by instantaneous start up and stopping. Through the closed loop control, a voltage is

generated corresponding to a required motor speed, ensuring the speed is maintained between an upper and lower limit.

5.13.4 THE CONTROL LOOP

The main function of the servo loop is to link the position of a moving part to an instruction value. As in all servo controlled equipment, static and dynamic components are dependent on the complete motion system (including both the algorithm and mechanical parts) and on the variation of the instruction over time. The machine layout of the control loop is shown in figure 5.27.



Pitch = distance covered in a single shaft revolution.

Fig 5.27 AXM servo loop

The incremental encoder used in the feedback loop can either be linear or rotary. In both cases it provides a signal with a frequency proportional to the velocity and a number of pulses that is proportional to the position. Hence, this control module solves the problem of both accurate position and speed control.

The various servo loop parameters and their effects on the system's responsiveness are included in Appendix B5.

Equipped with extensive error checking facilities, numerous safety devices, and the capability of closed loop control, the axis control system investigated tends to overcome many of the previously envisaged hurdles, making it an attractive proposition in the development of robot control.

As yet, it is not commercially available to industry, and sole agency has been given to Microdyne. This company should be approached regarding the procurement of such a module, as extensive and valuable work investigating the full operating characteristics of the module can be carried out.

6 CONCLUSIONS

The development of the industrial robot is seen by some as the panacea of the industrial environment, and an increasing number of people are getting involved, either through choice or through necessity.

The use of a robot has to be justified, and no longer is the financial consideration the only one. Implementation of a robot has many advantages, including:

- increased productivity;
- increased flexibility;
- reduced labour, materials and energy costs;
- improved quality and consistency;.
- humanization of the workplace;
- other economic considerations including direct and indirect costs.

There is not much more that can be done to present day robots, other than making them faster, lighter and more accurate. However, an area receiving growing interest is the field of controllers, with manufacturers concentrating on more advanced and sophisticated controllers. This is essentially the area of research that was pursued in this project.

Robots can be programmed in two ways, on-line and off-line. On-line methods include lead through and teach pendant programming, while off-line programming deals with some form of textual language providing instructions to the manipulator, where the program is interpreted by the robot software system. Off-line programming is gaining popularity amongst users, not least because of the many advantages associated with it. There is a move afoot to CAD systems, whereby the robot and its environment are simulated on a computer. In this manner, money and time is saved by identifying problems in the conceptual stage of robot implementation.

The project dealt with a feasibility study into PLC control of a robot, essentially bypassing the existing robot controller, and creating an off-line programming facility with a PLC/PC interface. Other than the advantages associated with off-line programming, a large cost saving is realised, as the price of a PLC/PC system is at the most 50% that of a robot controller. Further, the flexibility of PLCs arising from their modularity make them extremely adaptable to any control situation.

The feasibility study could not successfully carried out until it was determined how all the components in the robot system operated, and were fully operational. Thus the repair of encoders, and the determination of resolution and gearing had to be carried out.
All optical shaft encoders were repaired, and their interfacing and operating characteristics were determined. The resolution of each encoder was found to be 500 pulses per revolution.
The interfacing of the DC servo motors and the electromechanical brake was determined.
Their operating characteristics were determined, and curves relating input voltage and output rotational speed were drawn up.
The complete robot geometry was determined and documented for future work entailing computerised robot motion.
The gearing of each robot joint relating motor rotation to the movement of the arm was determined.

Considering the existing equipment, and that none of the available PLC modules were capable of inherent feedback control, an open loop system was decided on as the form of control.

The form the feasibility study took was in determing the suitability of a fast counting module for motor/encoder use. The encoder was linked up to the FCM, and acceptable signals were obtained. By controlling a single motor using

Ladder language, the control could be extended to four motors simultaneously using Grafcet language. By knowing the resolution of the encoders, as well as the number of pulses required to bring about a certain angular displacement of each robot arm, position control would be possible. By powering the motor and disengaging the brake, and by recording the pulses of the encoder, the brake could be applied and power disengaged when the motor has turned through the required number of pulses.
By carefully calibrating the speed of the various motors over a range of voltages, speed control would be possible.

Various observations were made and conclusions regarding the open loop control system were drawn.
The momentum of the motor carried it beyond the stopping point desired, and this was related to the delay in activating the brake. At higher speeds the overshoot was greater than at lower speeds, and this overshoot was calibrated for the various motors at various speeds. The bucket effect associated with this overshoot was also calibrated. With these errors calibrated, they could be incorporated into the control loop. This is essentially what open loop control hinges around. The behaviour of the system must be absolutely predictable, and thus the output achieved is what is expected. It is possible to compensate for the overshoot and bucket effect, but it is impossible to predict how accurately the system will behave with different robot positions and geometries. As the encoders are incremental, this error would be cumulative, resulting in an undesirable system with regard accuracy.

Another observation made regards dynamic loading of the robot arm. The load on the motor from the arm and subsequent weight (that the robot picks up) changes as the arm rotates about its hinge, moving its centre of gravity closer to or further from the joint. The speed of the motor does not remain constant as the toque applied to the motor changes, making an open loop speed control system unpredictable and undesirable.

Further, algorithms necessary to predict and compensate for the external effects experienced by an open loop system are too complex to be worked out and communicated through the PLC/PC network at any satisfactory rate.

Once the anticipated open loop system was found to be unsatisfactory, a closed loop control system had to be investigated that could be incorporated into the existing PLC system. This closed loop control would have to compensate for dynamic loading of the motors, variable speed output, and correct positional accuracy.

An axis control system, supplied by Telemecanique but not yet commercially available, was investigated. This module caters for rate feedback, as well as position feedback, agreeing with the requirements of closed loop control. Position feedback is effected through the encoder signal, the error algorithm working to provide no error due to zero overshoot.
Speed feedback is derived from the encoder pulses, provided the necessary fast timing equipment is available to measure time between pulses. This has to be done as no tachometer is supplied with the motor/encoder pairs.
Contained within the pulse width modulator is a segregated current supply control feedback loop, necessary to ensure that a constant torque, proportional to the desired acceleration, is supplied to the motor.

Open loop control was deemed undesirable for robot control purposes, and thus a closed loop control system was analysed in terms of an axis control module, which seemed to cater for all the problems encountered with open loop control. The procurement of such an axis control module (when it is available) is highly recommended for any further work in this field.

6 FUTURE WORK

As the project is an ongoing one, the future work proposed will form a good base for any work carried out in this field.

The only motor that was not accessed was the motor controlling the roll or swivel of the wrist, i.e. the T axis. The operation of this motor must be fully established for complete control.
Another aspect of the robot itself that needs interfacing with are the limit switches associated with each axis. Because the level of control achieved was not as desired, this interfacing was not carried out, but is essential for safety reasons when full control is realised.

With the PLC/PC interface, investigate how the PLC can be addressed from the PC. It is desirable that the user only enter, with the PC, end point coordinates, and through the controlling modules and algorithms associated with the PLC are acted on, resulting in the movement of the robot to the desired position. Instead of the user writing a new program for each new robot motion, these simple user inputs are required.

The procurement of an axis control system is highly recommended, as the development of a closed loop form of control with the existing equipment is very difficult, if not impossible.

The axis control system must be incorporated into the existing PLC unit. The existing PLC system must be upgraded to a TSX 47 30 system (by replacing the processor module and memory cartridge) to enable axis control module use. Ascertain if this system is capable of being programmed and controlled with the installed software.

The control loop characteristics must be investigated, first with a single motor. The speed and positional accuracy of the loop are the important factors that need consideration. The module should be capable of stopping and holding a motor in a required position, under load (even dynamic load), without the use of the brake. The module's speed control capabilities should also be investigated, as this is important if terminal motion is to be realised.

Once the control of a single motor has been accomplished, the procurement of a second module is necessary, as each module can only control one axis. The control of two motors must be investigated, looking at discrepancies in the communication rate, if any, of simultaneous control.

Once the suitability of the axis control system has been established, computerised robot motion must be considered. That is, relationships between the movements of the end manipulator and successive motor rotations must be established. Thus, a user input will result in a desired displacement of the end effector.

As a final stage in this project, a simulation package needs to be looked at, whereby the robot and its working environment are simulated on a CAD system. Defining robot motion in the simulation will result in robot motion on the shop floor. Thus a CAM or CIM environment will be possible.

8 REFERENCES

1. Scott P.; "The robotics revolution;" Basil Blackwell Publishers Ltd.; England; 1984.

2. Rodd M., Bloch G.; "Robotics;" The University of the Witwatersrand; Department of Electrical Engineering; 1986.

3. Weavind A.; "The development of an off-line interface for a manufacturing robot;" Final Year Research Project; University of the Witwatersrand; 1989.

4. Ardayfio D., Pootinger H.; "Computer control of robotic manipulators;" Mechanical Engineering; August 1982.

5. Mortimer J.; "Robot controllers - where do we go from here;" The Industrial Robot; September 1984.

6. "An introduction to PLCs for the uninitiated;" Computech; Vol. 6; No. 7; July 1990.

7. "At the end of a pneumatic packing line for tubs is an automatic PLC - controlled palletiser;" Computech; Vol. 6; No. 8; August 1990.

8. "PLC and pneumatics combined in automatic machine for producing venetian blinds;" Computech; Vol. 6; No. 8; August 1990.

9. "Pyramid integrator improves control of continuous steel casting process;" Computech; Vol. 6; No. 7; July 1990.

10. "PLCs automate aluminium producer;" Computech; Vol. 6; No. 7; July 1990.

11. "Charcoal briquet maker improves productivity with PLC - based interactive automation;" Computech; Vol. 6; No. 7; July 1990.

12. "PLC - based system cuts office lighting energy costs by 70%;" Computech; Vol. 6; No. 7; July 1990.

13. "The South African market for process control instrumentation: 1990;" Computech; Vol. 6; No. 7; July 1990.

14. "Bundling PLCs, computers and workstations;" Computech; Vol. 6; No. 7; July 1990.

15. "PLC offers networking capability;" Computech; Vol. 6; No. 7.; July 1990.

16. "PC programming of PLCs;" Computech; Vol. 6; No. 7; July 1990.

17. "Locally available PLCs;" Computech; Vol. 6; No. 7; July 1990.

18. Motiwalla S.; "Development of a software system for industrial robots;" Mechanical Engineering; August 1982.

19. Gruver W., Soroka B., Craig J., Turner T.; "Industrial robot programming languages: a comparative evaluation;" IEEE Transaction on systems, man, and cybernetics; Vol. SMC-14; No. 4; August 1984.

20. Mujtabu M.; "Current status of the AL manipulator programming system;" Standford Artificial Intelligence Laboratory; Memo AIM-477; Standford University; August 1982.

21. Taylor R., Summers P., Meyer J.; "AML: a manufacturing language;" The International Journal of Robotics Research; Vol. 1; No. 3.; August 1982.

22. Buckley S.; "AML: a structured robot programming language;" Massachusetts Institute of Technology - Artificial Intelligence Laboratory; August 1983.

23. Blume C., Jakob W.; "Design of the structured robot language (SRL);" Advanced Software in Robotics; Elsevier Science Publishers; North Holland; 1984.

24. Purdue R.; "WAVE: a model based language for manipulator control;" The Industrial Robot; Vol 4.; No. 1.; March 1977.

25. Popplestone R., Ambler A., Bellos I.; "RAPT: a language for describing assemblies;" The Industrial Robot; Vol. 5; No. 3; September 1978.

[illegible] L., Wesley M.; "AUTOPASS: an automatic programming system for computer controlled mechanical assembly;" IBM Research Centre; July 1977.

[illegible] M., Niehaus T.; "Off-line programming via standardized interfaces;" The Industrial Robot; Vol. 11; No. 3; September 1984.

[illegible] Hartley J.; "Hitachi leads the move off-line;" The Industrial Robot; Vol. 12; No. 1; March 1985.

29. Craig J.; "Anatomy of an off-line programming system;" Robotics Today; SILMA Inc.; February 1985.

30. "SILCAM: adding vision to ROBOCAM;" Robotics Today; SILMA Inc.; February 1985.

31. Haffenden A.; "Robotics simulation in practice;" The Industrial Robot; Vol. 12; No. 4; December 1984.

32. Carter S.; "Off-line robot programming: the state-of-the-art;" The Industrial Robot; Vol. 14; No. 4; December 1987.

33. Williamson T., Ashton M.; "Using CAD in the real world;" The Industrial Robot; Vol. 13; No. 3; September 1986.

34. Daniels G.; "Letting the industrial robot take the grind;" The Industrial Robot; Vol. 15; No. 1; March 1988.

35. Mattis P., Gill K.; "The best robot for the job - simulation can help decide;" The Industrial Robot; Vol. 15; No. 1; March 1988.

36. "Solutions to weld problems are well on the way;" The Second International Conference on Developments in Automated and Robotic Welding; The Industrial Robot; Vol. 15; No. 1; March 1988.

37. "FIAT buys ROBCAD;" Technomatix; The Industrial Robot; Vol. 16; No. 4; December 1989.

38. Sorenti P., Bennaton J.; "Off-line programming moves into practice;" The Industrial Robot; Vol. 16; No. 4; December 1989.

39. DAINICHI-SYKES: specialists in industrial robot systems; DAROS robot range information brochure; Preston; England.

40. DAINICHI-SYKES Robotics; Operators Training Manual; 300 Controller; HAMPO Systems; South Africa.

41. TSX 47 Programmable Controller Installation Manual; Manual 09.1987; Telemecanique; South Africa.

42. TSX AXT 200 Counting and Positioning Module; Users Manual; Manual 06.1987; Telemecanique; South Africa.

43. TSX T407 Programming Terminal Users Manual; Manual 12.1986; Telemecanique; South Africa.

44. "The First Step in Understanding Motion Control;" Computech; Vol. 6; No. 10; October 1990.

45. TSX AXM 172 / XBT 172 / AXB 72 Axis Control System User's Manual; Manual TSX D23 000E; Telemecanique; South Africa.



APPENDIX A

EQUIPMENT

A1 DAROS PT 300V ROBOT
A2 Robot controller
A3 Harmonic drives
A4 Optical shaft encoder
A5 PLC Components
- Power supply module
- Processor module
- User memory cartridge
- I/O modules
- Fast counting module
- Programming terminal



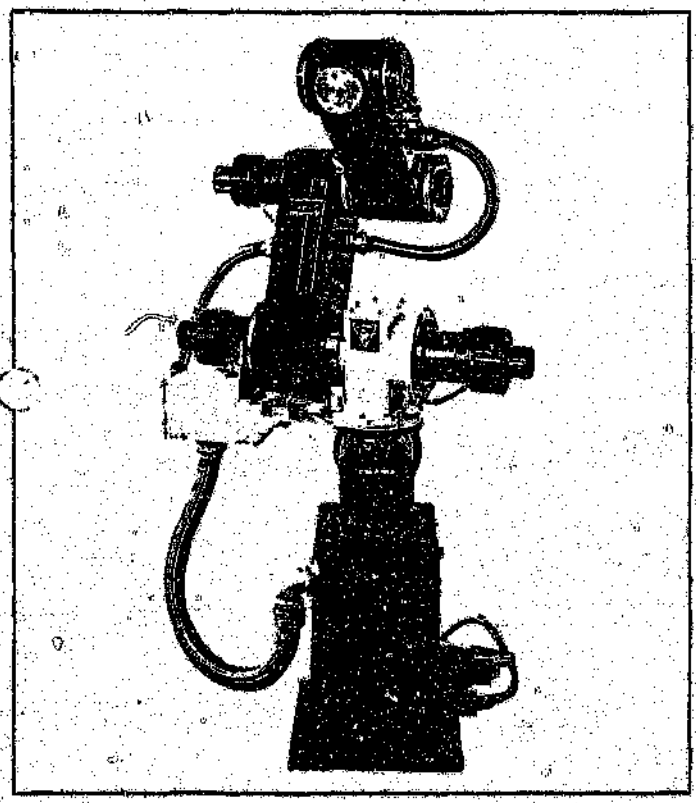
APPENDIX A1
PT 300V ROBOT

DAINICHI-SYKES

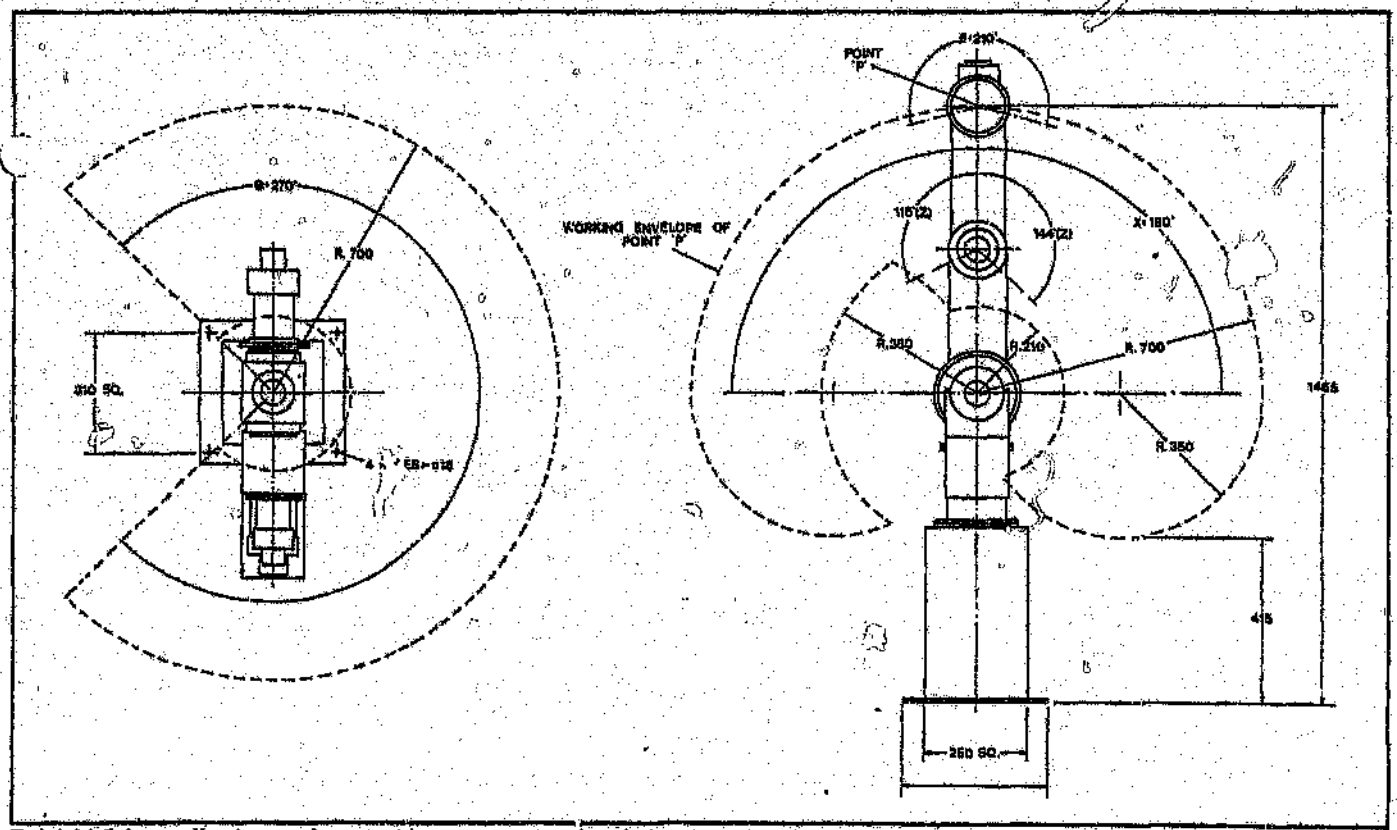
Daros PT300V

Advanced Microcomputer Technology
Rugged Construction
Easily Programmed

A precision robot specially suited to many applications such as arc welding, machine loading of smaller components, general handling and packaging.



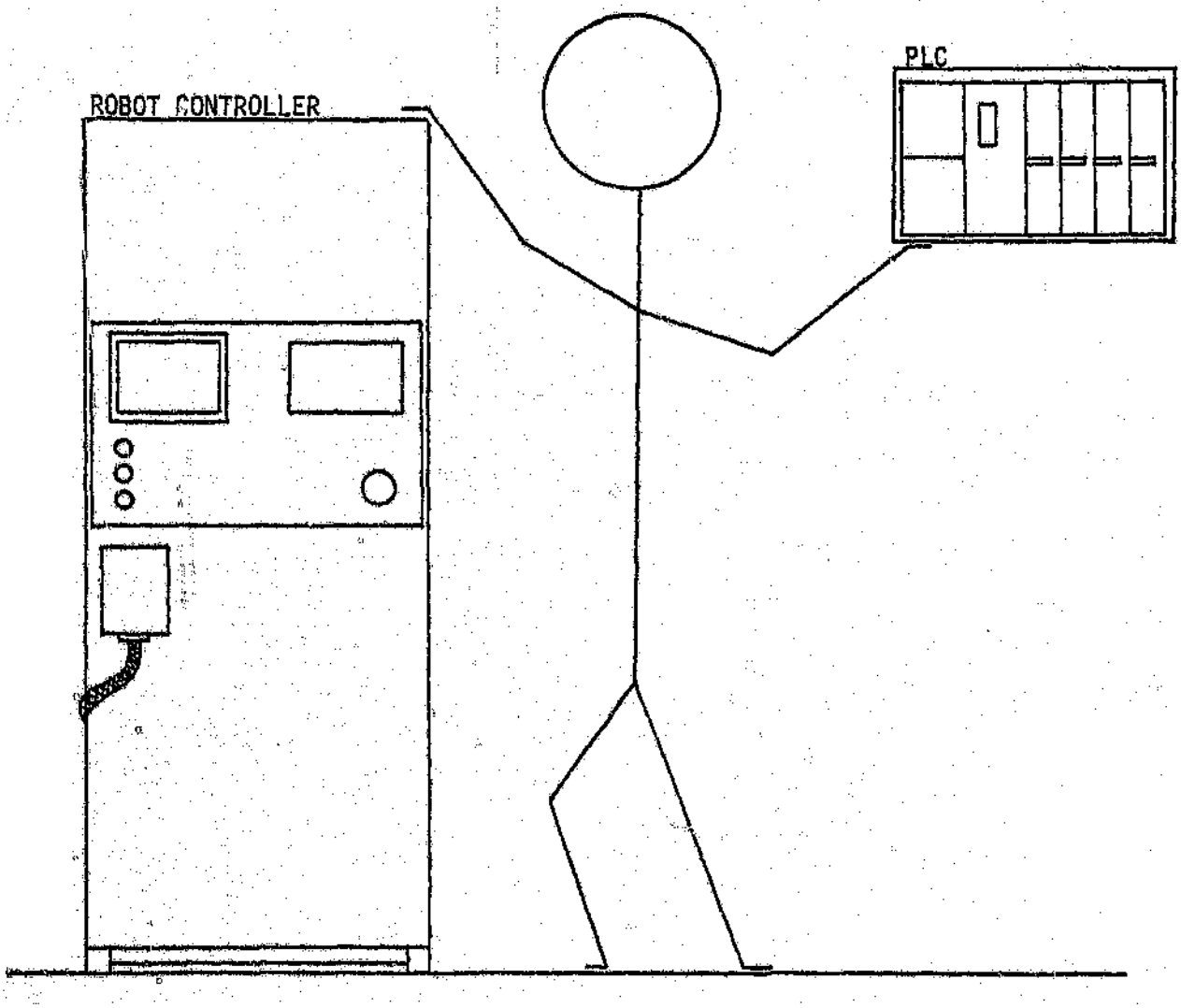
Model			PT300V	
Type			Jointed Arm	
Load carrying capability			3kg	
Number of axes			5	
	Axis		**Momentary max speed**	**Operating range**
Main axes	Shoulder arm	(X)	90°/s	±90°
	Elbow arm	(Z)	90°/s	+116° −144°
	Rotation	(θ)	90°/s	±135°
Wrist axis	Vertical	(β)	90°/s	±105°
	Rotation	(γ)	90°/s	+100° −170°
Drive system			DC Servo Motor	
Axis positioning repeatability			±0.1mm	
Balancing mechanism			—	
Power requirements	Electric		3 phase AC380/415V, 50/60Hz, 1.0kVA	
	Pneumatic		6.5kg/cm²G, to suit gripper	
Weight			188kg	



Dainichi-Sykes policy is one of constant improvement and updating of all the company's products, and specifications are thus liable to change at short notice.

APPENDIX A2
ROBOT CONTROLLER

COMPARISON BETWEEN CONTROLLER AND PLC



APPENDIX A3
HARMONIC DRIVES

A3 THE OPERATION OF HARMONIC DRIVES

Harmonic drives are a major advance in transmission design. They provide extremely large speed reduction in a single stage, coupled with high output torques. This is obtained through multiple tooth engagement, as much as 15 to 20% of total tooth complement. They are lightweight, more reliable, and more efficient. Input and output shafts are coaxial and concentric, allowing machine designers to dramatically reduce the space needed to accommodate high ratio, high torque drives.

All harmonic drive units have tight backlash, an output loss motion of better than 2 minutes of arc being quite common. Torque transfer is by a single, low friction ballrace, eliminating the compound losses associated with multi-stage gears, shafts and bearings. Tests have yielded efficiencies of up to 90%.

Harmonic drives offer high, single stage reductions, with ratios of up to 320:1. A drive of this capacity would occupy the same envelope size as a standard 80:1 drive unit. Further, harmonic drives can be coupled together to achieve excellent reductions.

A harmonic drive consists of three basic components, i.e. a wave generator, flexispline, and circular spline, shown below in figure A3.1.

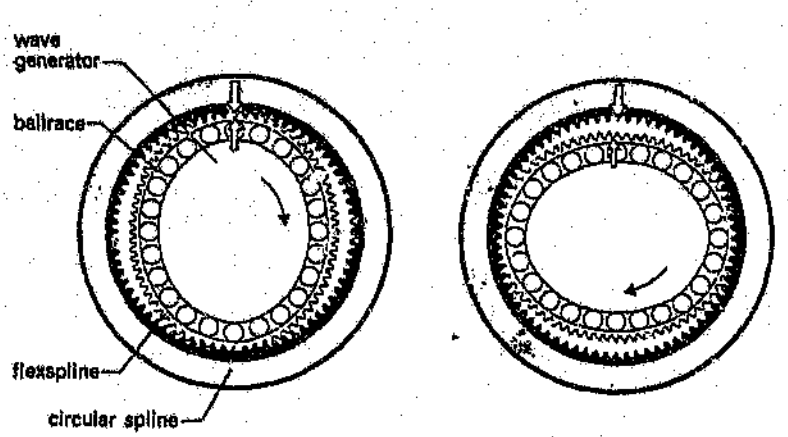


Fig A3.1 Harmonic drive components

The flexispline is a flexible toothed steel ring which can be deformed. It is forced into an elliptical shape by the elliptical wave generator which is situated inside it, acting as a former. This wave generator is specially constructed so that, even if the flexispline is held fixed, the generator can still slide round on the inner surface of the spline, and so rotate. This is accomplished by fitting a ballrace onto the surface of the generator so that there is very little friction between it and the flexispline. As the generator is non circular, as it rotates inside the flexispline (fixed), the flexispline is deformed (elliptically) by the generator. For a complete harmonic drive, the wave generator and flexispline are fitted into the circular spline which comprises an internally toothed solid steel ring. The size of the teeth on both splines are the same, however, there are a few less teeth on the flexispline than on the circular spline. As a result, the teeth on the elliptical flexispline only mesh with those on the circular spline in two areas. If the circular spline is held stationary, as the wave generator is rotated, it is these areas of contact (not the flexispline) that rotate around with the wave generator.

Because there are a few less teeth on the flexispline than on the circular spline, once the flexispline has been pushed against the whole surface of the circular spline (wave generator has completed one whole revolution), the flexispline will end up those few teeth away from its original starting point. That is, for each rotation of the wave generator, the flexispline moves a few teeth in the opposite direction. Thus, an input shaft on the wave generator will result in a high gearing reduction from the flexispline. The operation is illustrated in the following diagram (see figure A3.2).

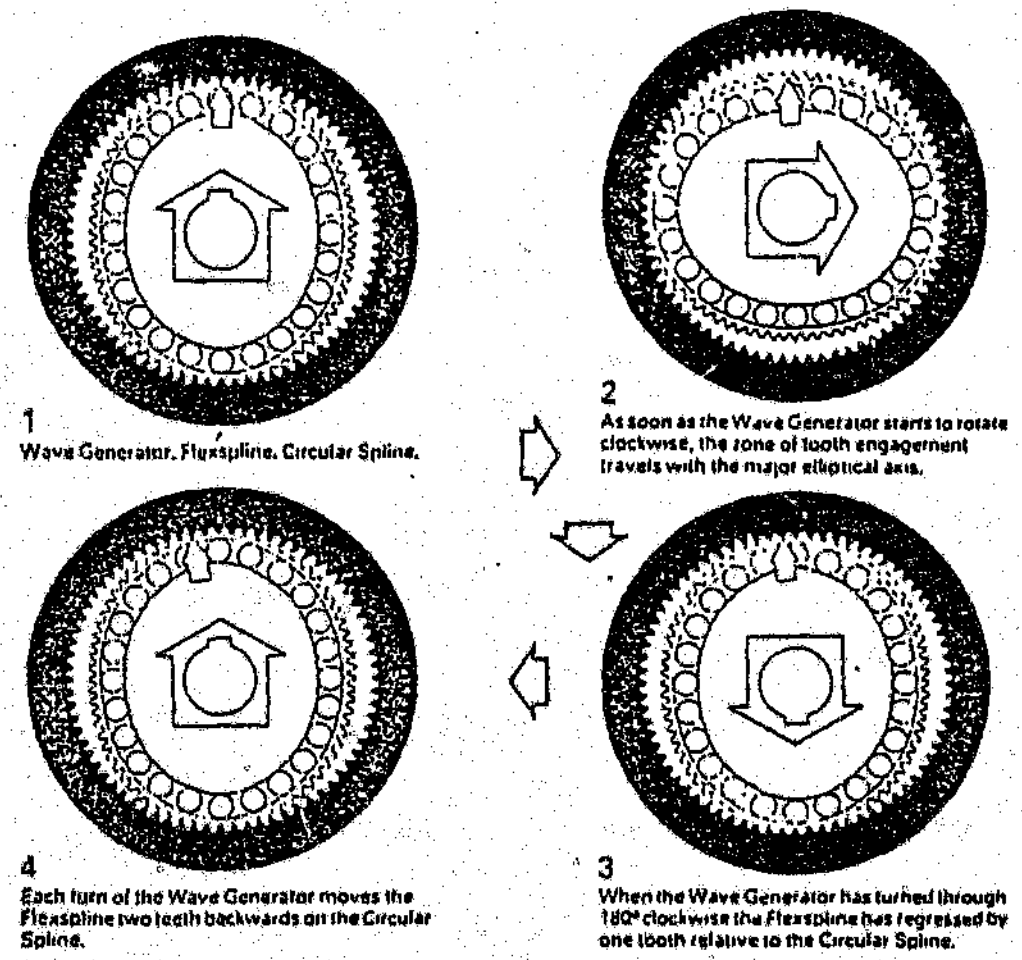


Fig A3.2 Harmonic drive operation

APPENDIX A4
OPTICAL SHAFT ENCODER



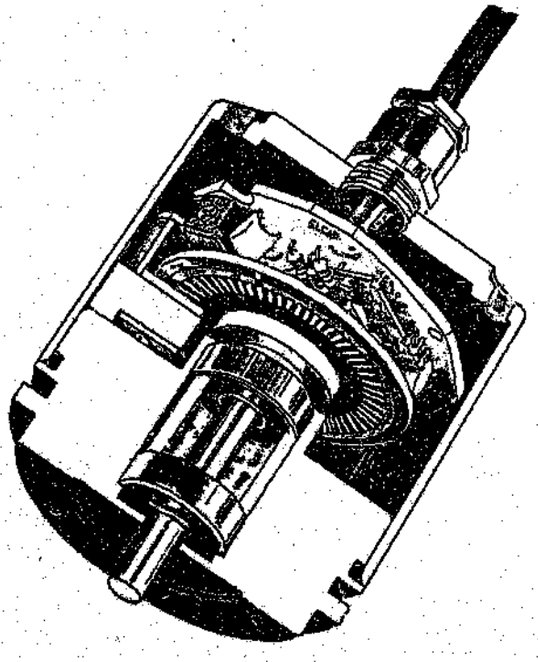
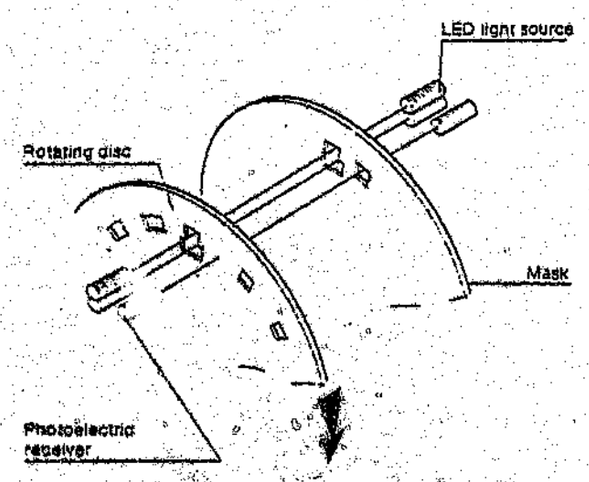


Fig A4.1 Cut away of a typical encoder



The main components responsable for the operatior of a rotary encoder are:

- An LED light source
- A slotted, synchronising mask
- A rotating, slotted disc
- A series of photoelectric receivers.

The light, transmitted from the LED source, passes through the slots in the mask and is thus aligned onto the photoelectric receivers.
As the disc rotates the light beams are interrupted by the slots to give the appropriate output signal, via the receivers.

Fig A4.2 Optical encoder components

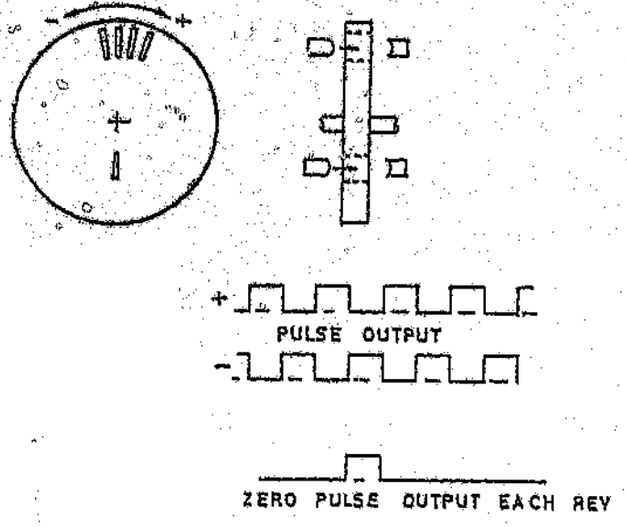
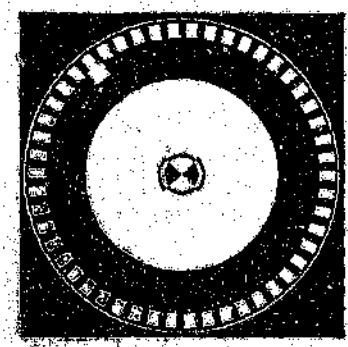
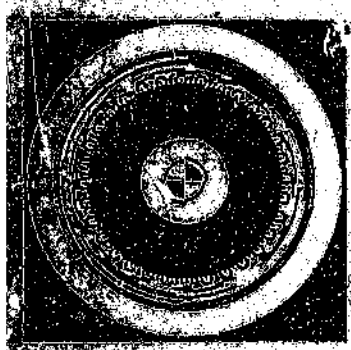


Fig A4.3 Encoder pulse train



"Incremental output": To measure displacement
This encoder will output a "pulse-train" signal, such that each pulse will represent a precise angle of shaft rotation.
Machine positioning may be determined by a system processing these output pulses. Two offset output signals are provided so that direction of rotation may be determined.
Principal areas of application:
Counting, speed control, feedback signalling, linear and angular positioning.

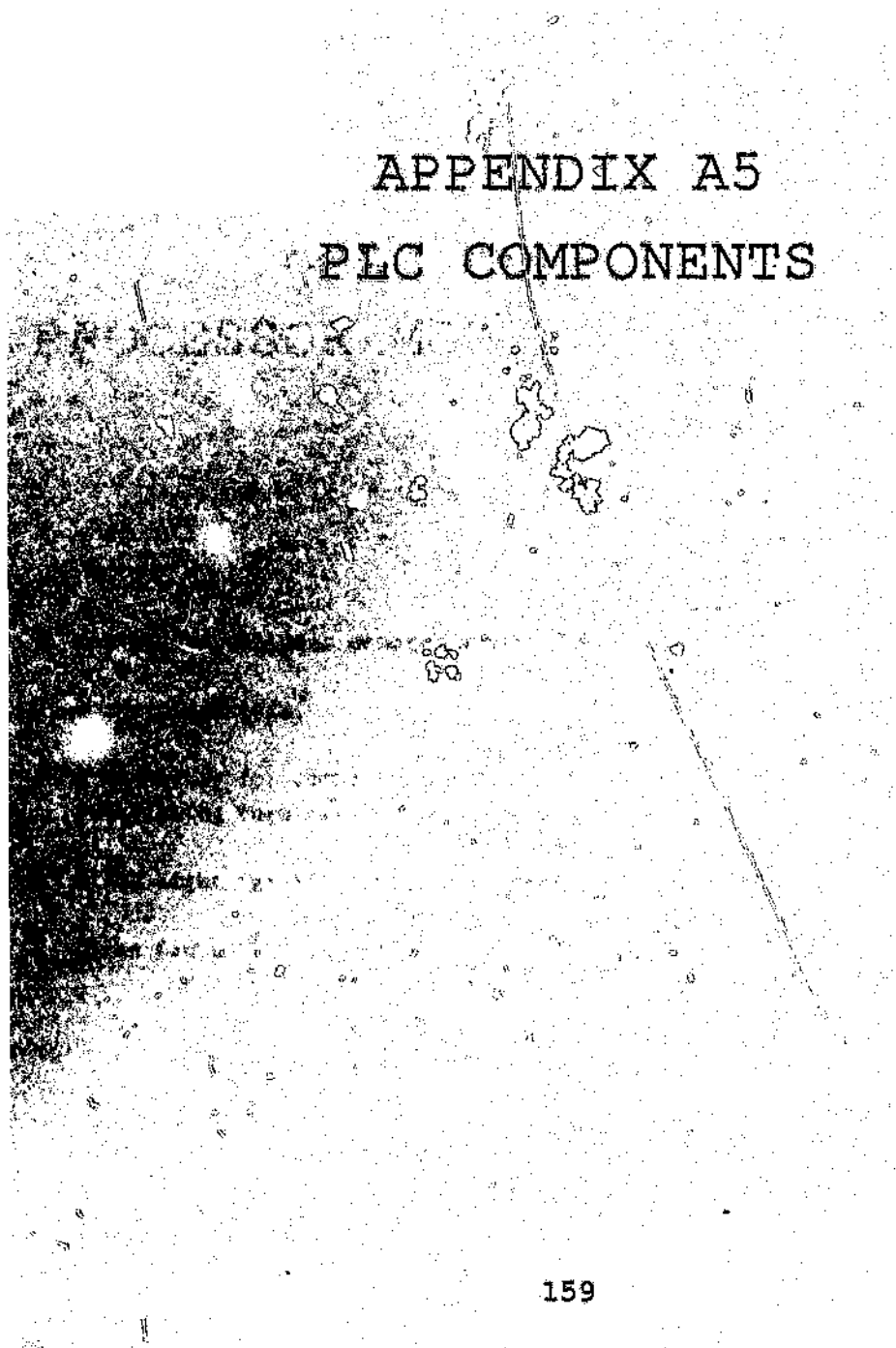


"Absolute output": To give actual position
The output from this type of encoder constitutes a binary code for each angular position of the shaft. The resolution is equal to 2^n where n = number of bits.
Multiturn encoders with "absolute output" are available, giving a resolution of $m \times 2^n$.
Typical applications:
Angular and linear positioning, measurement of distances, levels, etc.

Fig A4.4 Incremental and absolute encoders

APPENDIX A5

PLC COMPONENTS



POWER SUPPLY

. 1 An OK light (green) - internal voltages present and correct,

. 2 An ON light (orange) - mains power present and correct,

. 3 A power supply terminal block,

. 4 An alarm relay contact,

. 5 A mains power failure detector ON/OFF switch.

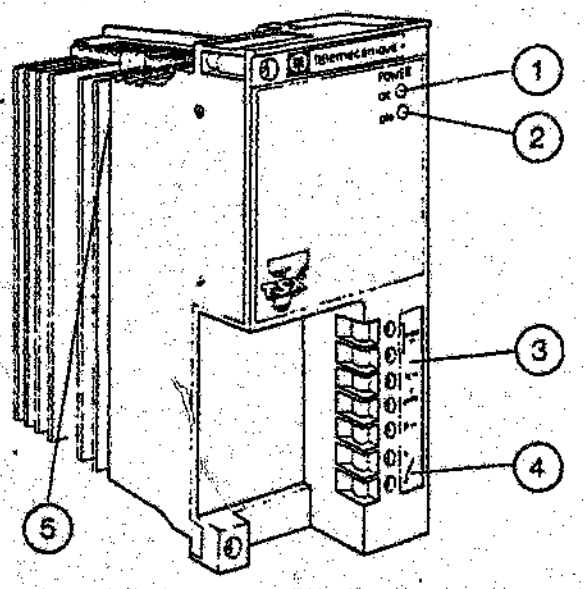


Fig A5.1 The power supply module

PROCESSOR MODULE

memory cartridge,

communication with the
nal and peripherals,

green): program running

ault lights (red):
CPU processor fault
MEM user memory fault
I/O I/O fault

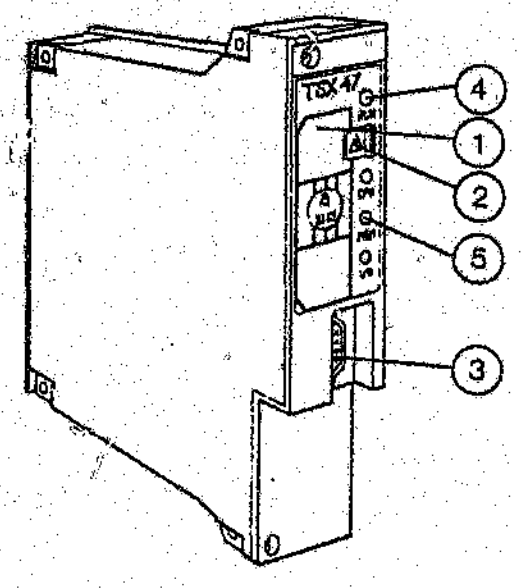


Fig A5.2 The processor module

POWER SUPPLY

. 1 An OK light (green) - internal voltages present and correct,

. 2 An ON light (orange) - mains power present and correct,

. 3 A power supply terminal block,

. 4 An alarm relay contact,

. 5 A mains power failure detector ON/OFF switch.

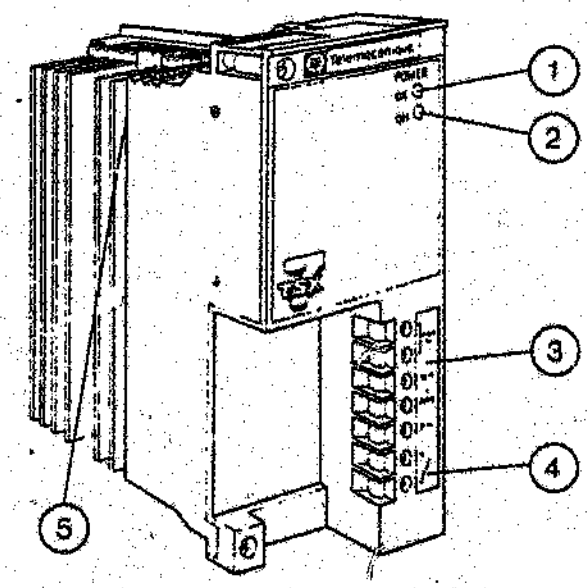


Fig A5.1 The power supply module

PROCESSOR MODULE

1 A slot for the user memory cartridge,

2 A cartridge lock,

3 A connector for communication with the programming terminal and peripherals,

4 A RUN light (green): program running

5 Three fault lights (red):
CPU processor fault
MEM user memory fault
I/O I/O fault.

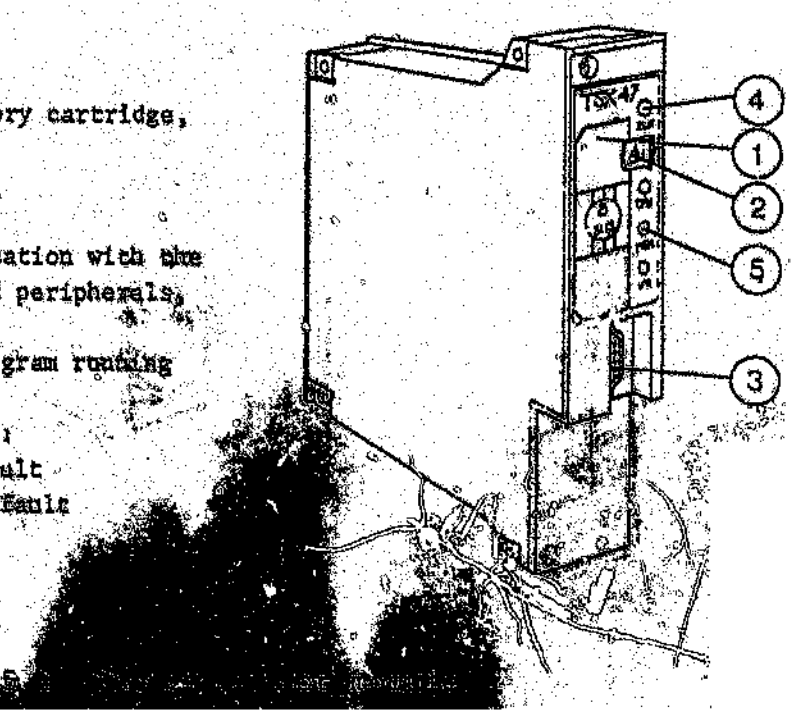


Fig A5

MEMORY CARTRIDGE

Each cartridge comprises :

1 A colour-coded extractor:
Red : RAM memory
Black: EPROM memory

2 A sliding panel for erasing and protecting EPROMs

3 A locating device

4 A label permitting identification of the program

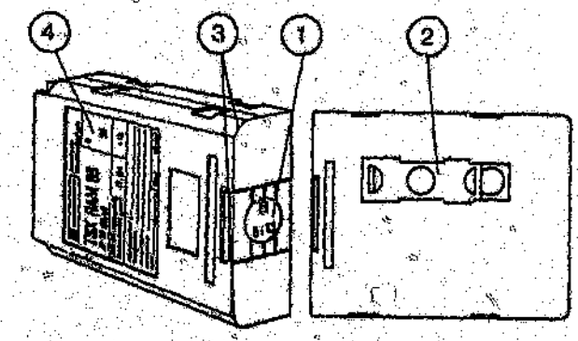


Fig A5.3 User memory cartridge

DISCRETE I/O MODULES

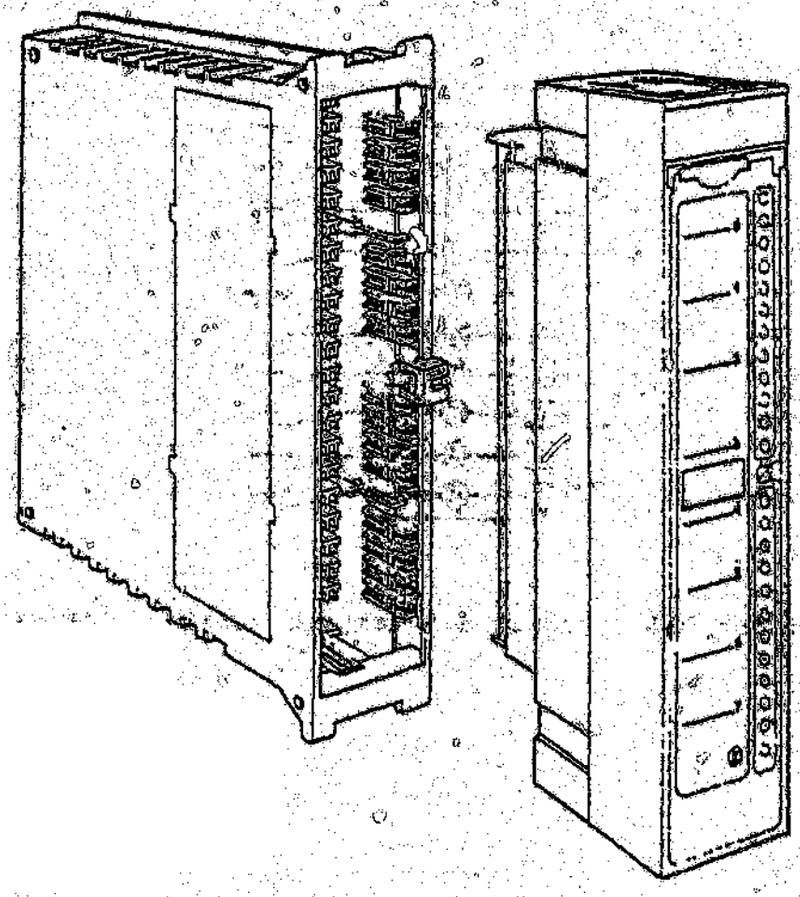


Fig A5.4 Discrete I/O module

WIRING DIAGRAM

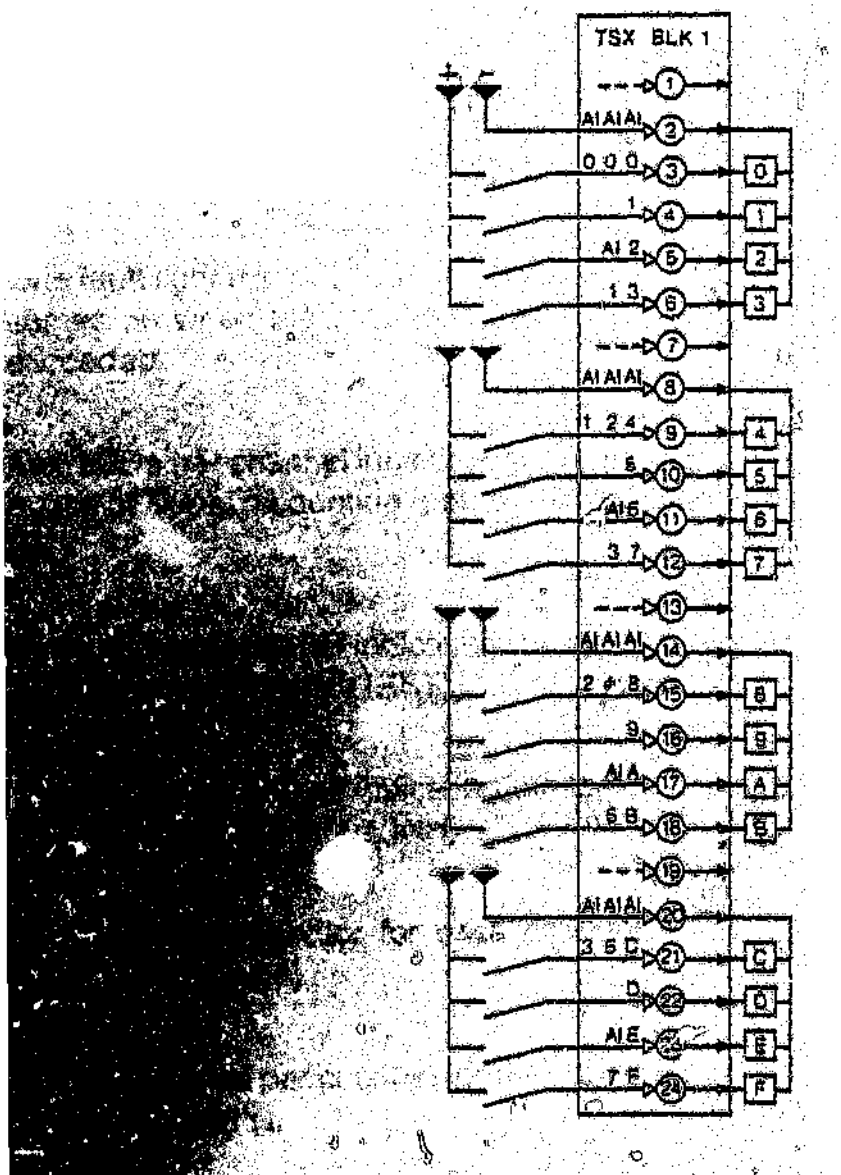


Fig. A5.5 Wiring diagram for discrete I/O modules

FAST COUNTING MODULE

(1) A metal case which protects the components physically and against radiated electrical noise.

(2) An opening on the left side which gives access to the upcounting/downcounting discriminator switches for each channel.

(3) One fault light (F) for each channel, which comes on when the counting capacity is exceeded.

(4) Two LEDs per channel indicating the logic states of the two counting inputs (IN0 and IN1).

(5) One LED per channel indicating the logic state of the associated output (S0).

(6) One LED per channel indicating the logic state of the counting input (IN0).

(7) One socket connector for each channel (9-pin Sub-D type).

(8) One blank label per channel for the user's identifying marks.

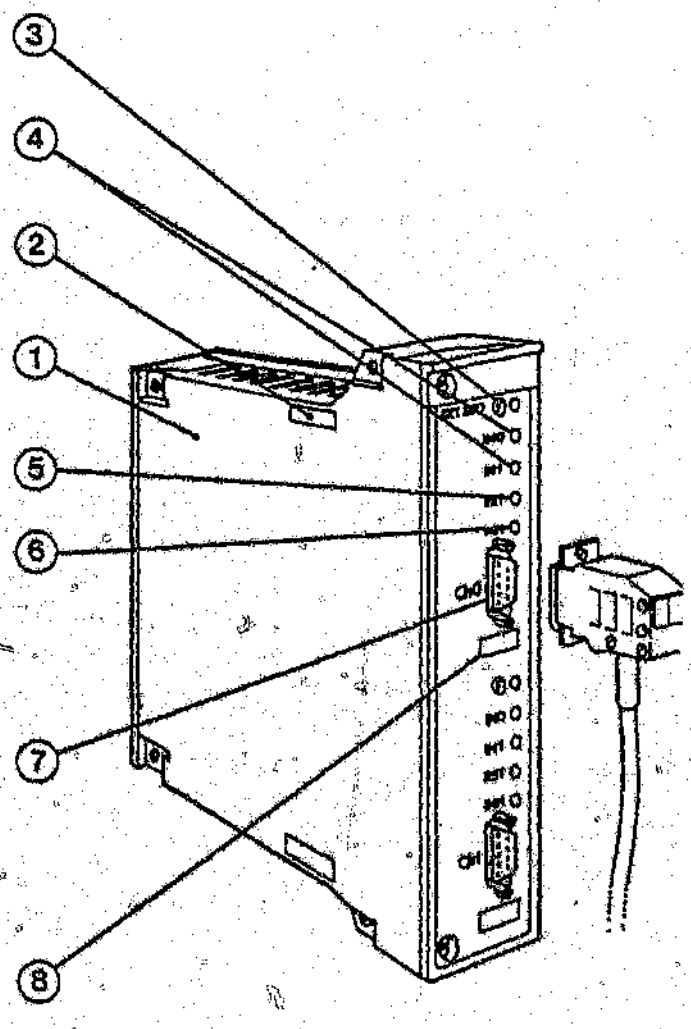


Fig A5.6 TSX AXT 200 counting and positioning module

PROGRAMMING TERMINAL

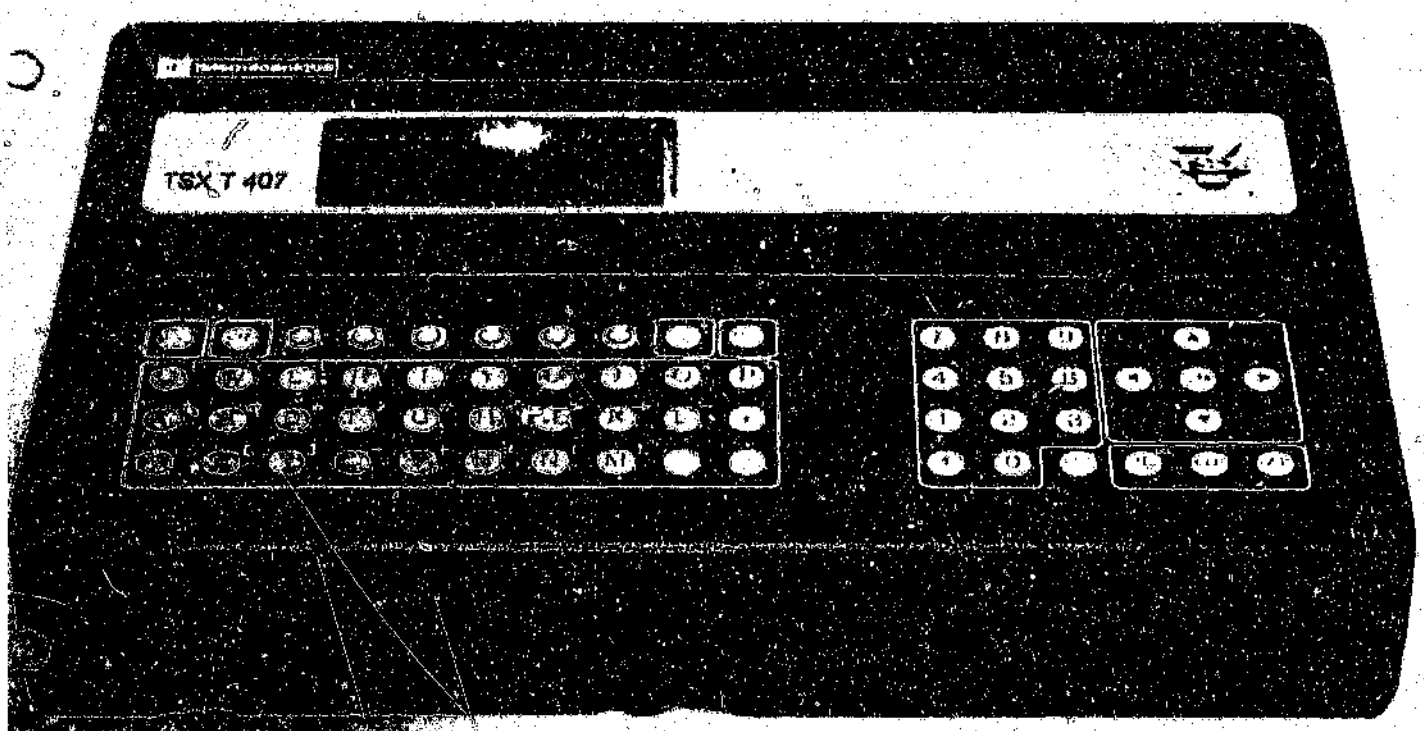


Fig A5.7 The PLC programming terminal

APPENDIX B

Counter function block
Fast counting module
Frequency generator one
Frequency generator two
Axis control system
Pulse width modulator

APPENDIX B1
COUNTER FUNCTION BLOCK

COUNTER FUNCTION BLOCK

The counter function block accepts rising edge inputs to increment or decrement an internal register 'value'. Outputs are set depending on how this value compares to a redefinable internal word 'preset'.

The block has four inputs and three outputs which are summarised below in terms that make their interpretation far simpler than the manual indicates. Each block is labeled with an arbitrary user number (i), and 'preset' can be set from the adjust mode (if modify is allowed), else from within the normal programming mode or within the program by addressing bit Ci,P.

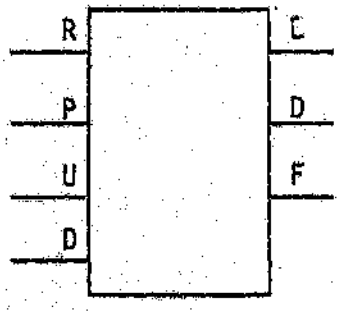


Fig B1.1 The function block element

Input R : The reset input sets 'value' to zero when it is high (not necessarily a rising edge).

Input P : The preset input sets 'value' to 'preset' when it is high (not necessarily a rising edge).

Input U : The up-counting input increments 'value' on its rising edge only.

Input D : The down-counting input decrements 'value' on its rising edge only.

Output E: The empty output is set high if 'value' equals -1 (set low at all other times).

Output D: The done output is set high if 'value' equals 'preset' (set low at all other times)

Output F: The full output is set high if 'value' has been incremented to zero (will not be set if 'value' reached zero from +1).

APPENDIX B2
FAST COUNTING MODULE

THE FAST COUNTING MODULE (FCM)

The principle of this module is that it accepts a single or double input pulse train, and generates a number of readable 'inputs' that can then be used for different applications.

The primary (hard) inputs to the module comprise 24 V reset and inhibit switches, and 5 or 24 V dual logic pulse inputs (preferably square waves). The module reads the inputs much faster than ..e PLC's standard scan rate (time to run through a single iteration of a program), hence its fast counting capa' 'ties. These inputs are interpreted (depending on dual or single logic input) and their results stored in an internal register. It is this storage poin' that comprises the generated inputs that are readable for program use.

The internal register is made up of 16 bits. Bits 0 to 4 and 8 to C are ' 'd as 5 bit buffers for the counting storage of the , input channels 0 and 1 respectively. Bits 5 and D are the respective sign bits associated with these buffers, and bits 6, E, 7, and F are set high according to the counting inhibit and buffer reset hard inputs for the two channels respectively. Bit S (not the Hex code), is a bit common to both channels and indicates when either of the 5 bit buffers is full. This is accompanied by a fault light on the panel, corresponding to the exceeded channel. Lights do appear on the module to indicate the state of the hard inputs (reset, inhibit, and the two counting channels). The counting channel lights will flash at the frequency of input, but will look steady for high frequencies (see fig B2.1 that follows).

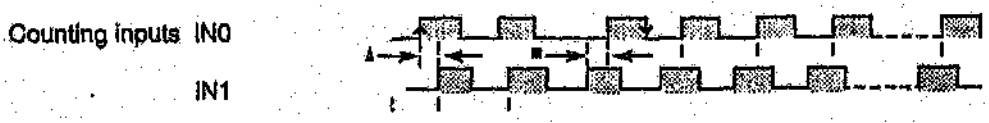
THE FAST COUNTING MODULE (FCM)

The principle of this module is that it accepts a single or double input pulse train, and generates a number of readable 'inputs' that can then be used for different applications.

The primary (hard) inputs to the module comprise 24 V reset and inhibit switches, and 5 or 24 V dual logic pulse inputs (preferably square waves). The module reads the inputs much faster than the PLC's standard scan rate (time to run through a single iteration of a program), hence its fast counting capabilities. These inputs are interpreted (depending on dual or single logic input) and their results stored in an internal register. It is this storage point that comprises the generated inputs that are readable for program use.

The internal register is made up of 16 bits. Bits 0 to 4 and 8 to C are used as 5 bit buffers for the counting storage of the input channels 0 and 1 respectively. Bits 5 and D are the respective sign bits associated with these buffers, and bits 6, E, 7, and F are set high according to the counting inhibit and buffer reset hard inputs for the two channels respectively. Bit S (not the Hex code), is a bit common to both channels and indicates when either of the 5 bit buffers is full. This is accompanied by a fault light on the panel, corresponding to the exceeded channel. Lights do appear on the module to indicate the state of the hard inputs (reset, inhibit, and the two counting channels). The counting channel lights will flash at the frequency of input, but will look steady for high frequencies (see fig B2.1 that follows).

incremented for every pulse input. However, for dual logic input (two quadrature pulse train inputs, up/down-counting, and discrimination set), the buffer is dec/incremented depending on whether the one pulse train leads or lags the other. This buffer is updated constantly, irrespective of what is going on in the main program's cycle.



Count value			0	1	2	1	0	-1	-2	-32
Channels	Ch0	Ch1								
Counting bits	Ixy,4	Ixy,C	0	0	0	0	0	1	1	0
	Ixy,3	Ixy,B	0	0	0	0	0	1	1	0
	Ixy,2	Ixy,A	0	0	0	0	0	1	1	0
	Ixy,1	Ixy,9	0	0	1	0	0	1	1	0
	Ixy,0	Ixy,8	0	1	0	1	0	1	0	0

The maximum values that can be coded on the five counting bits are +31 and -32.

▲ phase shift of +90° or $\pi/2$ ■ phase shift of −90° or $-\pi/2$

Fig B2.2 Timing diagram of pulse train

With the sign bit setting (bit 5), the counting range actually reaches 65. This bit can be used as an extra bit, although the protection provided by a negative range warning is then lost in the wrap between 64 and 0, and is only useful for up-counting situations. To do this, the sign bit is added as the most significant bit of the buffer. This gives a 6 bit buffer (2^6=64), which is interpreted in straight binary. Using the sign bit as designed, when set at 0, the buffer bits (0 to 4) can be interpreted in straight binary. When this bit is high, however, the correct negative value is read by obtaining the ones complement of the buffer, adding 1 to this to get the true complement, and converting this value to straight binary and inserting a minus sign in front of it. The table overleaf gives a list of values encompassing the whole range of the 5 bit buffer.

This module is used as a counter, in a program, by down-loading its buffer into an accumulating counting word

on every scan. Simply put, this means that as the program is running through its steps, the counter is accepting pulse inputs and accumulating them in its buffer. As the program comes round to reading this buffer, all the pulses accumulated since the last scan are down-loaded and the buffer is reset.

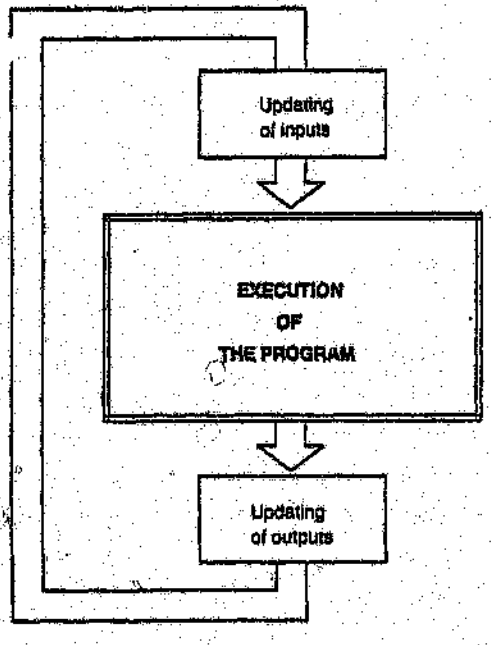


Fig B2.3 Controller operation

This means, that on each PLC scan, anything from 0 to 31 (2^5) pulses will be added to the accumulating word in the program. This is where the modules limitations become apparent.

Firstly, the module must scanned frequently enough for its buffer not to overflow (indicated by the setting of bit S and the illumination of the respective fault light). That is, the buffer must be down-loaded and reset before is can accumulate more than 31 pulses. As the standard scan rate of the PLC is 10 ms, the input train pulse period must not exceed 0.32 ms. For up-counting, using the sign bit as an additional bit on the buffer will allow this value to reach 64 before overflow, thus reducing this problem to well

within the limitations of the following ones. It should be noted, however, that in this case after each scan, the buffer must be set to -31 (or 0 over the 6 bit range) to avoid it sending a fault signal half way along its counting range.

Secondly, the down-loading of the buffer gives a bucket effect in that the word is incremented in blocks which may range from 1 to 31 counts in size. Thus the resolution of the counting system may be reduced by up to 3100 %. Curves giving this resolution for varying scan rates and input frequencies are given below, but this can be observed on program execution by monitoring (in debug mode), the size of the word being used to transfer this buffer to the accumulating word.

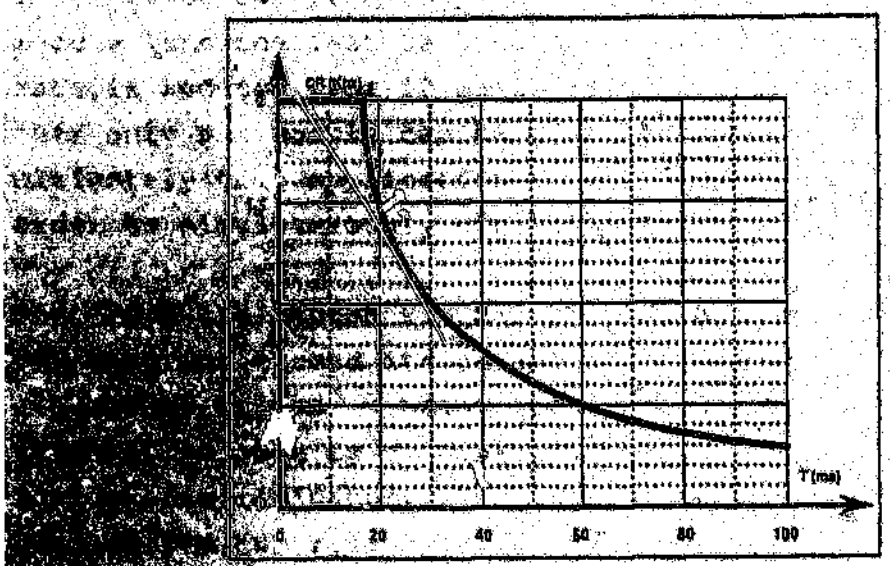


Fig B2.4 Maximum counting rate curve

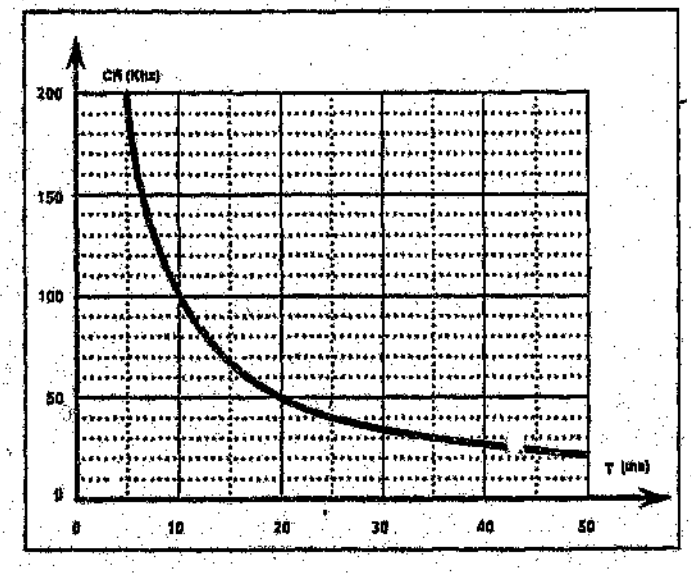


Fig B2.5 Counting accuracy curve

These first two problems can be reduced by programming the module into the fast task of the PLC, in which the scan rate is reduced from 10 ms to 5 ms. The problem however is that only one module can be configured into this task, and ultimately this application is going to require two in order to simultaneously control four motors.

The third limitation is that the input pulse rate to the PCM must not exceed its capabilities (advertised as being 2 KHz, but tested as being closer to 3 KHz), as it would then not be able to distinguish the different pulses. Finally, discriminatory counting requires a difference between the two rise times (lead/lag). Naturally large frequencies of the input trains would result in small lead/lag times which may eventually become incoherent. This is something that must be monitored.

The enormous advantage of this module, is the fact that it handles discriminatory counting without the extensive programming that would normally be required to do this. Naturally, in this application, this facility is imperative as bi-directional control of the motors is required and the direction of movement is an essential part of the feedback

loop. Naturally this reduced programming will increase the capabilities of the PLC in that it will not be required to perform the time consuming calculations that would otherwise be necessary.

For use of this module, it must be emphasised that the 'inputs' are generated by the module, but should be treated as direct inputs for programming, and are addressed through the respective module location e.g. I2,C if the module is in slot 2 of the PLC rack. The reset input zeros the internal register, it use within the program also allows for zeroing of the accumulating word. It should be noted that the only way to reset an overflow fault is to power the reset input.
The inhibit input, not usually used within the program, prevents the dec/incrementing of the internal buffer when it is powered high.

Wiring into the module is done through a nine pin plug. Entering into the female socket (on the module), else into the back of the male adaptor, the configuration as shown as

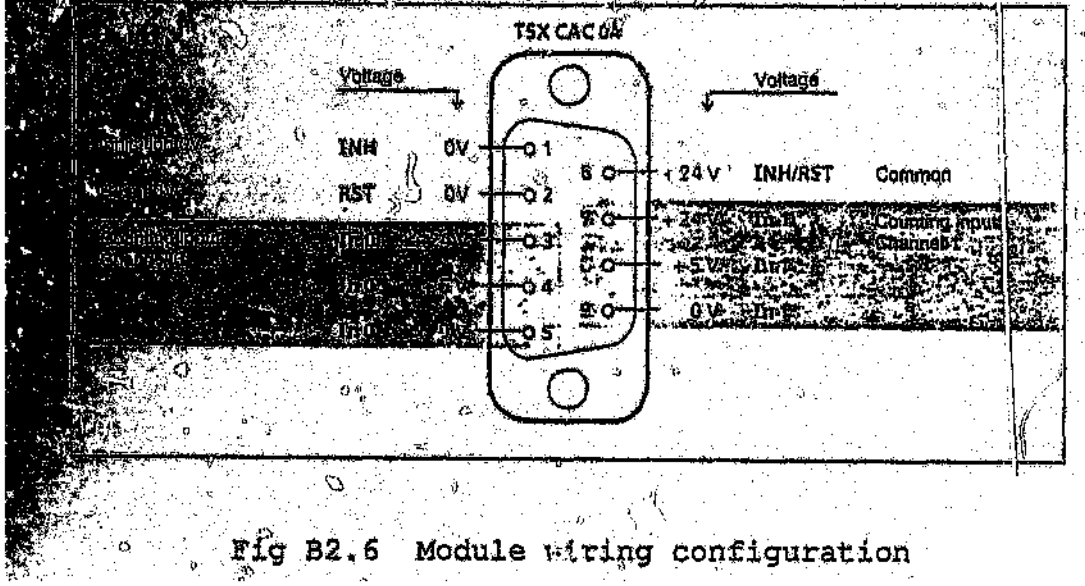


Fig B2.6 Module wiring configuration

APPENDIX B3
FREQUENCY GENERATOR 1

FREQUENCY GENERATOR 1: DESCRIPTION & OPERATION

The generator had to meet a certain number of specifications. Firstly, it must be able to generate a twin 5 V square wave output, secondly these outputs must be in quadrature (90^{O} out of phase), and lastly it had to span a frequency range from 1 Hz to over 3 KHz.

The circuit makes use of a 555 timer chip, characterised by its ability to generate a square wave output with a variable frequency. The principle of operation behind the generation of a quadrature output involves duplicating this square wave, inverting one of the signals, and generating two further square waves exactly half the frequency of the first two, each related to one of the first. The idea is that the first two waves will be 180^{O} out of phase (inverted), and thus the second two, of half the frequency, will be 90^{O} out of phase (quadrature). This principle is shown in figure B3.1.

The circuit diagram follows later (see figs B3.3 & B3.4), with all the necessary circuit protection, and with each chip being marked for segregated identification.

The frequency of the 555 timer output is controlled through a potentiometer connected across pins 2, 6 & 7. A set of three switchable capacitors has been added across these pins to give a frequency range switching facility giving a total scope, in combination with the potentiometer, as follows:-

Capacitor 1 (1μF) : 5.8 Hz - 56 Hz
Capacitor 2 (.1μF) : 61 Hz - 582 Hz
Capacitor 3 (.01μF): 634 Hz - 5400 Hz

The output from the 555 timer is channelled into two NOR gates on a 7402 Quad 2 Input NOR chip, consisting of 4 NOR gates. One gate is used to invert the signal and send it

straight out for further use, and another two are strung in series to 'beef up' the signal (otherwise unchanged) so that the following chip can read it easily. This gives the two inverted signals required for quadrature development. These two signals are then fed into a 74LS107 chip comprising two rising edge triggered JK flip flops. By outputting a square wave with its edges triggered by a rising edge input, we get a square wave exactly half the frequency of the input form. Using the inverted 555 timer outputs as inputs to these flip flops, we get two signals exactly in quadrature.

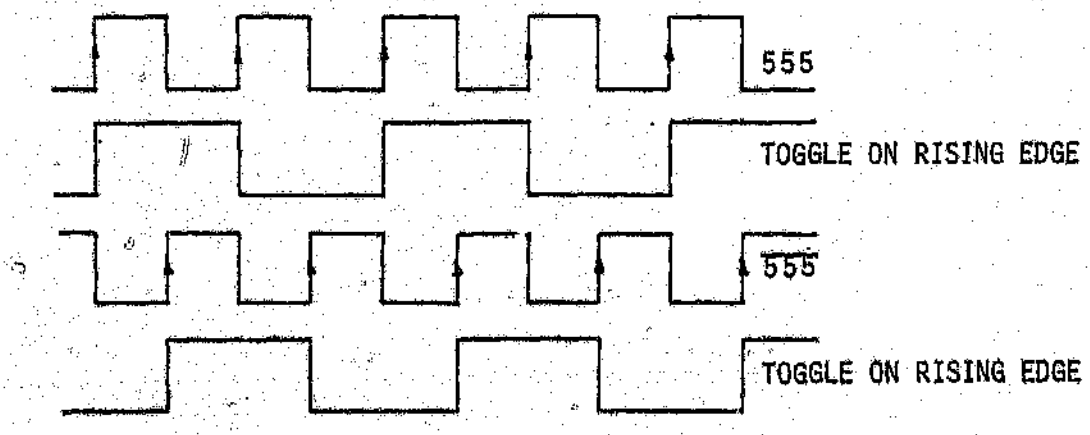


Fig B3.1 Achieving quadrature

The problem with this circuit was one that had been anticipated, i.e. the 555 timer does not give a constant duty cycle (ratio of mark to space time) over all frequencies. The disadvantage of this inconsistency is that the final wave forms will not be in quadrature, i.e. the rising edge of the second pulse will not fall exactly half way across the mark space of the first, as is illustrated in the following timing diagram.

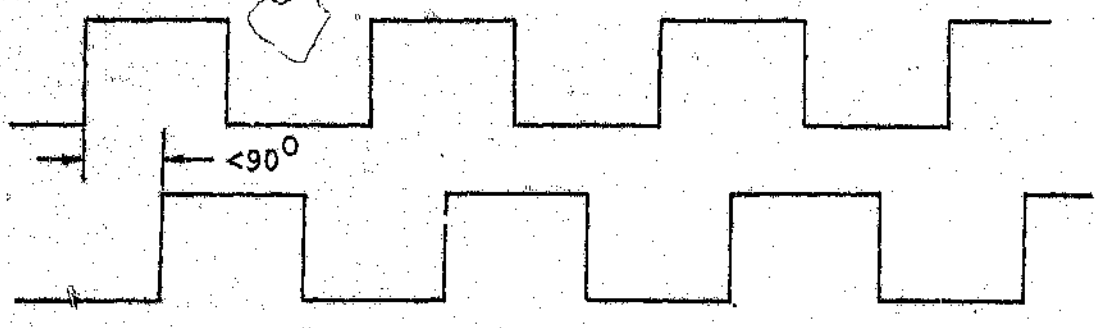


Fig B3.2 Problem encountered with 555 timer chip

As this situation worsens, the lead/lag between the two pulse trains becomes indistinguishable. It was hoped that the frequency rage required of this circuit would not be high enough to allow this problem to be too severe, this risk being taken on the grounds that a more accurate chip would take time to identify and acquire.

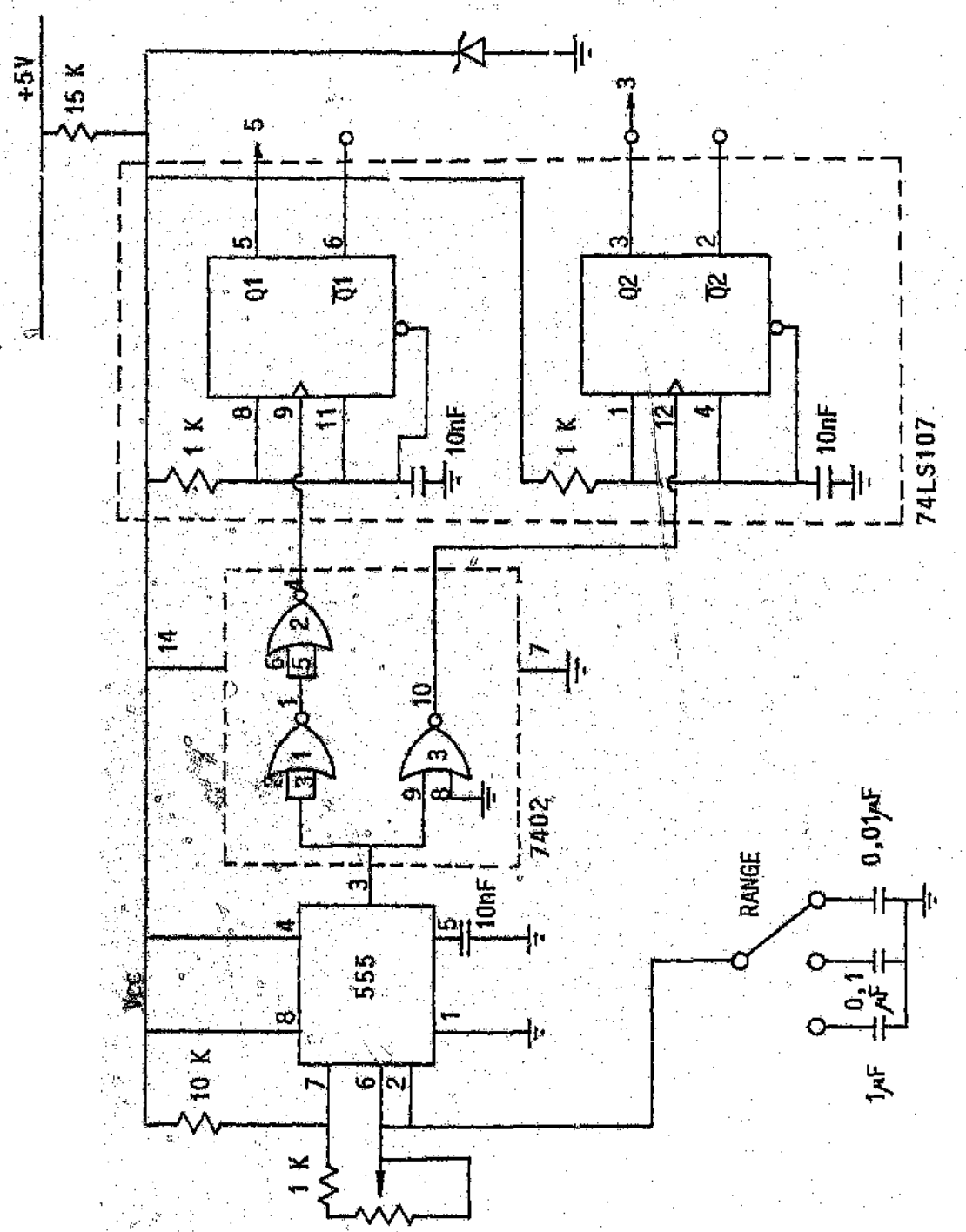


Fig E' 3 Circuit diagram - frequency generator 1

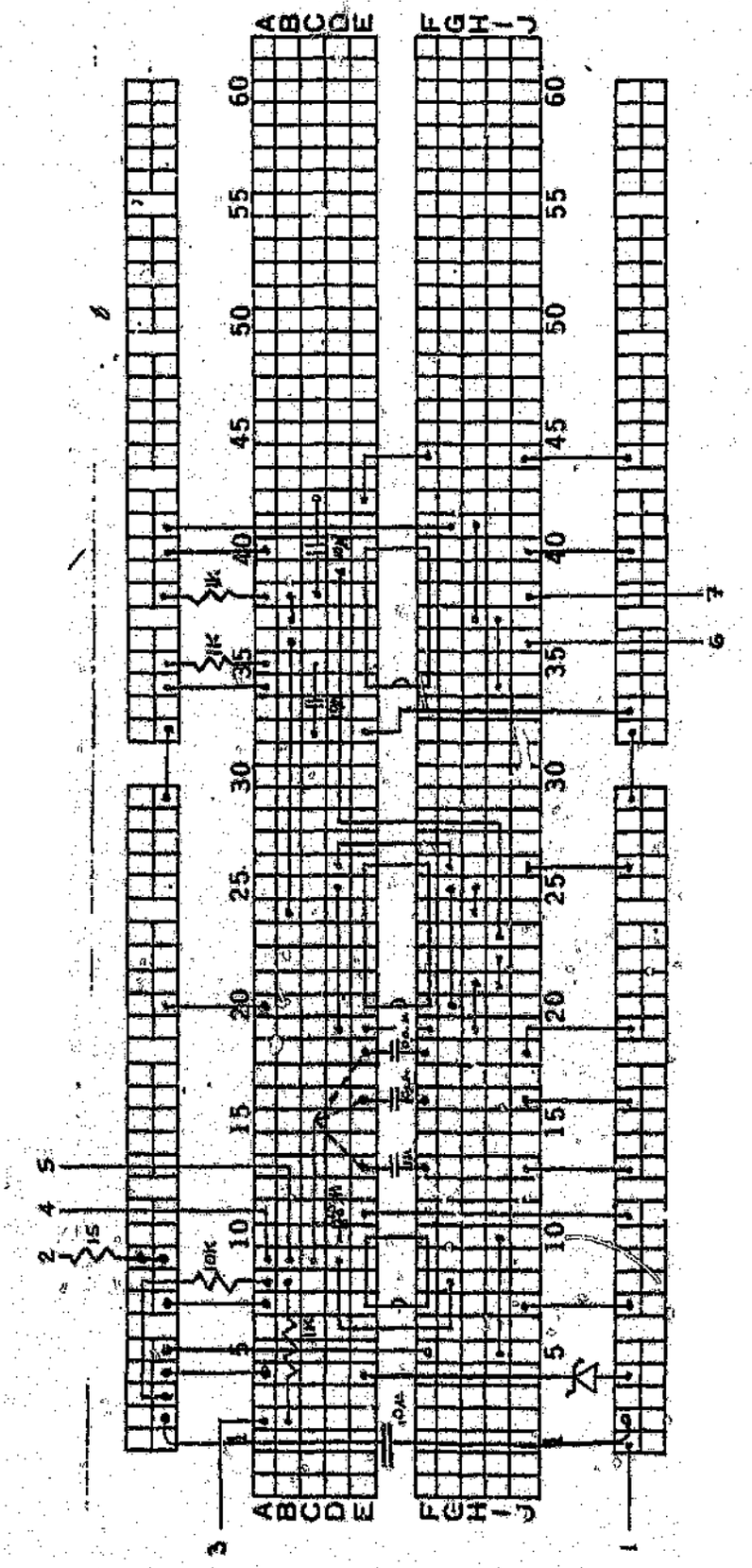


Fig B3.4 Bread board layout - frequency generator 1

APPENDIX B4
FREQUENCY GENERATOR 2

FREQUENCY GENERATOR 2: DESCRIPTION & OPERATION

The replacement of the 555 timer chip shown in the first frequency generator circuit did away with the need for the NOR gates inverting chip. The reason for this was that this new chip had a number of functions that the 555 timer did not, namely an enable switch and two outputs, one of them inverted. The new circuit is laid out as in figure B4.1.

The operation of the circuit is exactly the same as the first generator, with inverted signals being used to generate a quadrature output from a twin flip flop chip. Naturally, the frequency ranging available form this circuit was different from that obtained from the first. In addition, a few more capacitors were necessary to achieve a fluid range. These ranges were as follows:

Capacitor 1 (100μF) : 4.6 Hz - 52 Hz
Capacitor 2 (80μF) : 6.2 Hz - 62 Hz
Capacitor 3 (10μF) : 96 Hz - 943 Hz
Capacitor 4 (1μF) : 202 Hz - 1980 Hz
Capacitor 5 (0.1μF) : 440 Hz - 4360 Hz

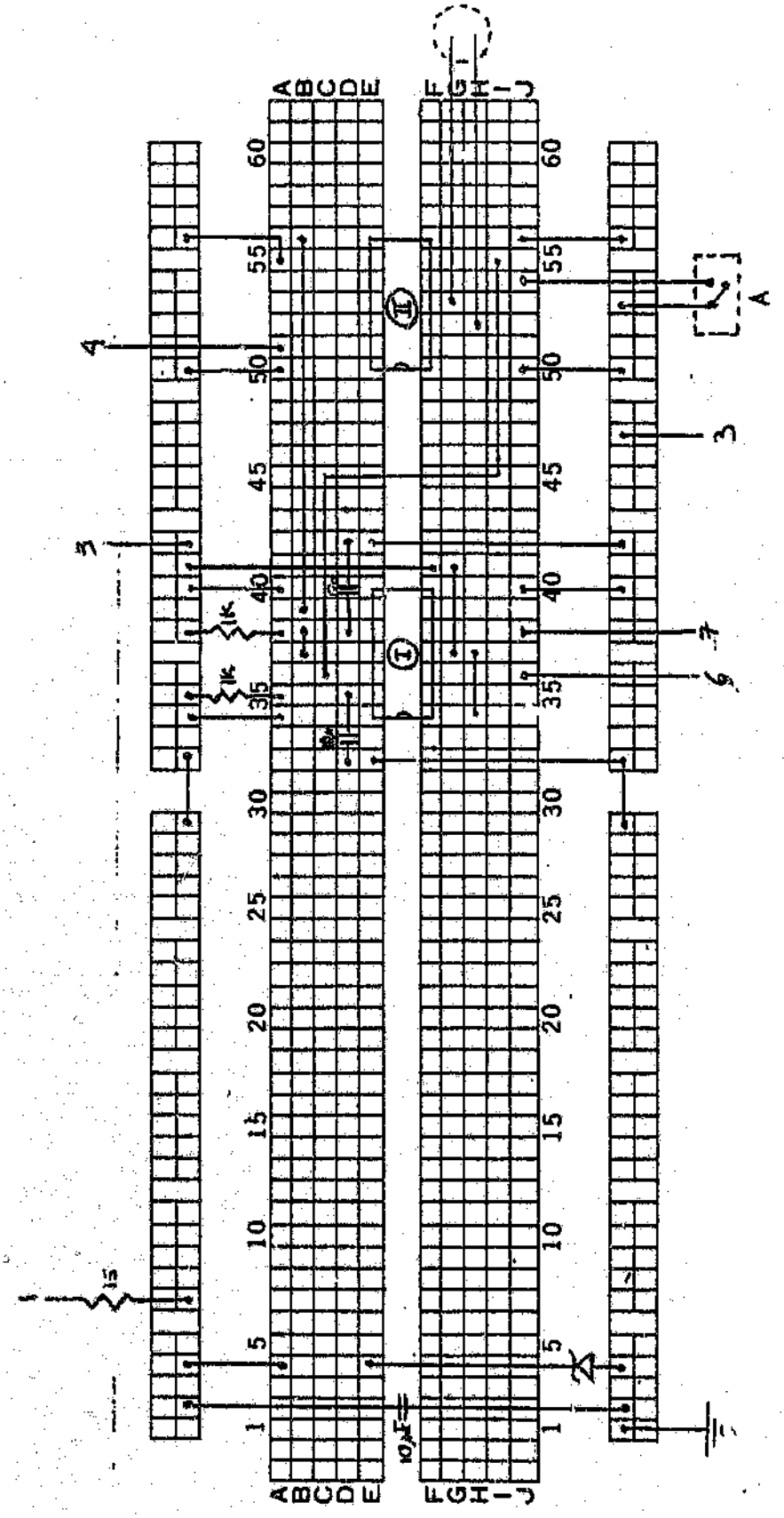


Fig B4.1 Bread board layout - frequency generator 2

APPENDIX B5
AXM SERVO LOOP

1.2 Servo Loop Parameters

Resolution (R)

This is the distance that the moving part must travel to increment the sensor signal.

This gives:
N = number of pulses per revolution (rotary) or over the length of the sensor (linear).
L = useable length of the linear sensor.

In the case of a linear sensor, the calculation of R is immediate:

$$R = L/N$$

In the case of a rotary encoder, the position of the reduction gear is critical.

$$R = ne \cdot Pitch/N$$

ne = equivalent reduction ratio resulting from the reduction ratios located between the encoder and the leadscrew pitch (sensor position B with ne = 1).

In all cases:

$$Vl = F.R \qquad \text{and} \qquad e = I.R$$

Vl : linear velocity
e : distance travelled
F : frequency
I : number of increments for a given motion.

Position Gain

The performance of the motion control system is usually expressed as a function of the position gain KPOS.

$$KPOS = V/de$$

V : velocity
de : position error

With velocity and position measured at the same point on the machine system.

KPOS : represents the static gain of the transfer function when operating as an open loop.
If the velocity is constant, KPOS represents the inverse value of the time required to correct the position. It is expressed as 1/second.

1.3 Precision / Stability

Increasing the position gain KPOS improves precision, reducing KPOS improves stability (the precision-stability trade-off).

When the variable speed drive is correctly adjusted the performance of the entire system is summarized by the following formula:

$$4\,m^2.KPOS.T = 1$$

T = time constant for the variable speed drive/motor/mechanical drive assembly,
m = damping coefficient.

The overrun value as a function of m is :

Overrun	50%	4%	0%
m (damping)	0.2	0.707	1

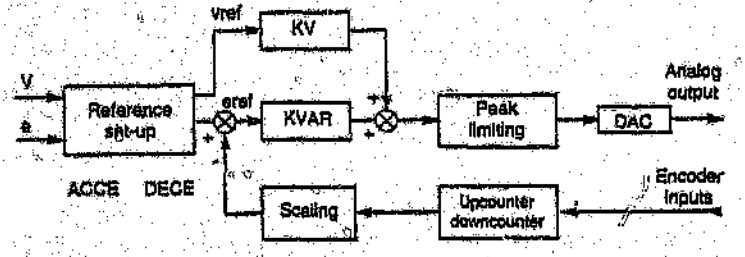
Example :

If T = 30ms and the user requires a maximum overrun of 4% or m = 0.707, the result is a KPOS value of :

$KPOS = 1/4m^2.T = 1/(4.(0.707)^2.30.10^{-3}) = 16\ s^{-1}$

2.1 Presentation

Block Diagram



Setting-up the References

This function is not accessible by the user. It is used to enter the instruction values for eref = f(t) and vref = f(t) as a function of the overrun and velocity values specified by the user along with the acceleration (ACCE) and deceleration values set in the configuration and the velocity modulation coefficient.

Scaling

Depending on the maximum velocity, distance and resolution, the module calculates the value of the scaling factor KR (that is also referred to as the machine characteristic factor).

The user can access a fine adjustment of this factor (function F61 on the XBT terminal) to compensate for any lack of precision in the machine parameters entered in the configuration (refer to Part 2, Sub-section 3.2).

Upcounter/Downcounter

The sum of the sensor increments gives the position of the moving object and allows monitoring of its motion.

KVAR

Proportional gain factor:

$$KVAR = C \cdot KPOS \cdot UMAX$$

C : constant
UMAX : the value of the variable speed drive instruction corresponding to VMAX velocity (UMAX < 9V)

The user enters the required KPOS value in the configuration and the module calculates the corresponding KVAR value.

Peak Limiting

The LIMV factor entered in the configuration allows the user to adjust the authorized velocity overrun value.

$$5\% < LIMV < 20\% \text{ of VMAX}$$

KV

The feed forward factor or overrun compensation factor is expressed as a percentage. 100% corresponds to the value that would completely remove any position error at constant velocity when using a variable speed drive without continuous error.

When KV increases, the position error decreases, however the risk of an overrun including the risk of stop point overrun is increased.
Therefore a compromise between the two must be found.

Note : In some cases, the position error passes through a minimum level and may change sign when KV increases.

Digital Analog Converter (DAC)

The function of this circuit is to convert the digital output value (10 bits + sign) into an analog voltage of between +10V and -10V.

Specific Servo Loop Features

In addition to normal proportional action on (KVAR) there is an added action proportional to the velocity (KV).

Once the two separate adjustments (KPOS and KV) have been made separately, the user will have the best possible compromise for the application between precision, stability and speed.

Assuming that the variable speed drive is correctly adjusted, the transfer function is :

$$\frac{e(p)}{eref(p)} = \frac{(1+ KV.p/100.KPOS)}{T.p^2/KPOS + p/KPOS + 1}$$

e(p) = a Laplace transform of the representation of true machine motion as a function of time.

eref(p) = a Laplace transform of the required representation of machine motion as a function of time, (servo loop reference).

2.3 Performance Levels

Error Transfer Function

$$\frac{de(p)}{eref(p)} = \frac{p}{KPOS} \cdot (Tp + 1 - KV/100) \cdot \frac{1}{T.p^2/KPOS + p/KPOS + 1}$$

This function will give the theoretical average error values after stabilisation, as a function of the input signal. It is assumed that there is no continuous variable speed drive error.

Note : all of the formulas are given for reference only, they assume that the entire system is perfectly adjusted.

Response to a step : $eref(t) = Constant$

$de(t) = \pm R + dev$
$t \rightarrow$ infinite

R : resolution
dev : returned error due to the variable speed drive(*).

Response to a slope : $eref(t) = V.t$ (V = velocity)

$$de(t)_{t \rightarrow infinite} = \frac{2}{KPOS} \cdot (1 - KV/100)\, V \pm R + dev$$

or

$$de(t)_{t \rightarrow infinate} = R \cdot (\pm 1 + \frac{72000}{KPOS} \cdot k \cdot (1 - KV/100)) + de$$

with $k = \frac{\text{working velocity}}{\text{maximum velocity}}$

Response to a 2nd degree function : $eref(t) = 1/2\ A.t^2$
(A = acceleration)

- with KV = / 100 : $de(t)_{t \rightarrow infinate} = infinate$
- with KV = 100 : $de(t)_{t \rightarrow infinate} = \frac{2}{KPOS} T.A + R + dev$

If variable Ta is introduced: Ta = VMAX/A

$$de(t)_{t \rightarrow infinate} = R\,(\pm 1 + \frac{72000}{KPOS} .VMAX . \frac{T}{Ta}) + dev$$

(*) Note : $dev(t) = dy(t) \cdot \frac{ne \cdot pitch}{KPOS.\ H}$

with 1/H = static gain of the motor and variable speed drive assembly
dy(t) = error in volts on the input to the variable speed drive

5.1 General Procedure

Complete machine system performance is described by the following approximation formula :

$$4m^2.KPOS.T = 1$$

KPOS = position gain,
T = time constant for the variable speed drive, motor, mechanical parts assembly in seconds,
m = stability factor.

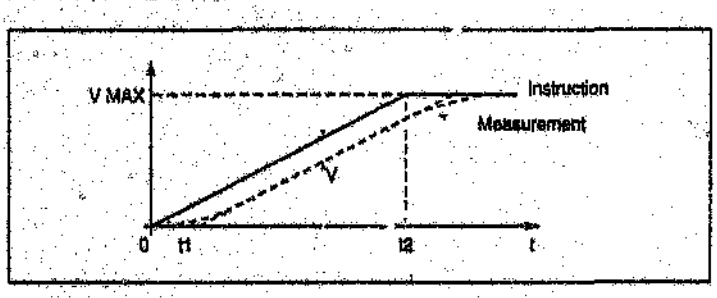
Identification comprises finding a numerical value for T so that the user can predict the performance of the machine system and check that the adjustments are optimized for the application.

Procedure

- Send an instruction value with linear evolution over time, the slope should be such that the variable speed drive does not have to limit the intensity,
- Read the true velocity instruction (tachometer-generator),
- Under these conditions, T is given by one of the two formulas shown below :

$$T = t1$$

$$T = V/P$$



$P = VMAX/t^2$

APPENDIX B6
PULSE WIDTH MODULATOR

PULSE WIDTH MODULATOR

A pulse width modulator (PWM) is a commonly used device for the control of servo motors. The device operates by switching a power supply at a rate proportional to the voltage of the input signal. Thus, a small analog signal can be given to the PWM, and a source of any size can be switched to produce the desired output. The main advantage of this type of amplifier is that it can cater for any input signal, as it does not use this as a source for its output. The principle behind its operation can best be explained through a timing diagram.



Fig B6.1 Characteristic pulse train output from a PWM

Although the pulse periods are kept constant, their mark time is the proportional factor. At high frequencies, the average output voltage is proportional to the mark time. Thus, the output voltage can be regulated between the limits of the above pulse train, where one of these limits can be negative and the other positive. As these limits are entirely dependant upon the regulated power supply, the output range is limited only by the available source.

The PWM circuit consists of five smaller circuits, each explained below.

THE VOLTAGE DROPPER

This circuit takes a 48 V DC load and reduces it to a 24 V DC output. The 48 V input must consist of a -24 V and +24 V pole. The result is a ±12 V output. (See figure B6.2).

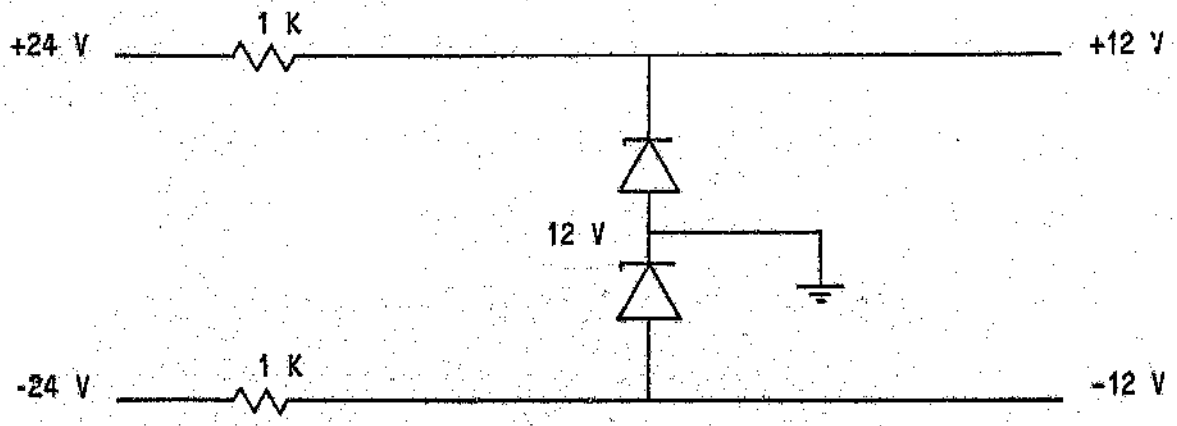


Fig B6.2 Voltage dropper

THE ADDING AMPLIFIER

The chip used in the oscillatory modulating circuit required an i put signal in the range of 0 to +24 V. As the input signal ranges from -10 V to +10 V, it is necessary to add a constant value to this signal in order to bring it into the acceptible range.

An adding op-amp (LM 741) is used to add 12 V to this range and thus produce an output ranging from +2 V to +24 V.

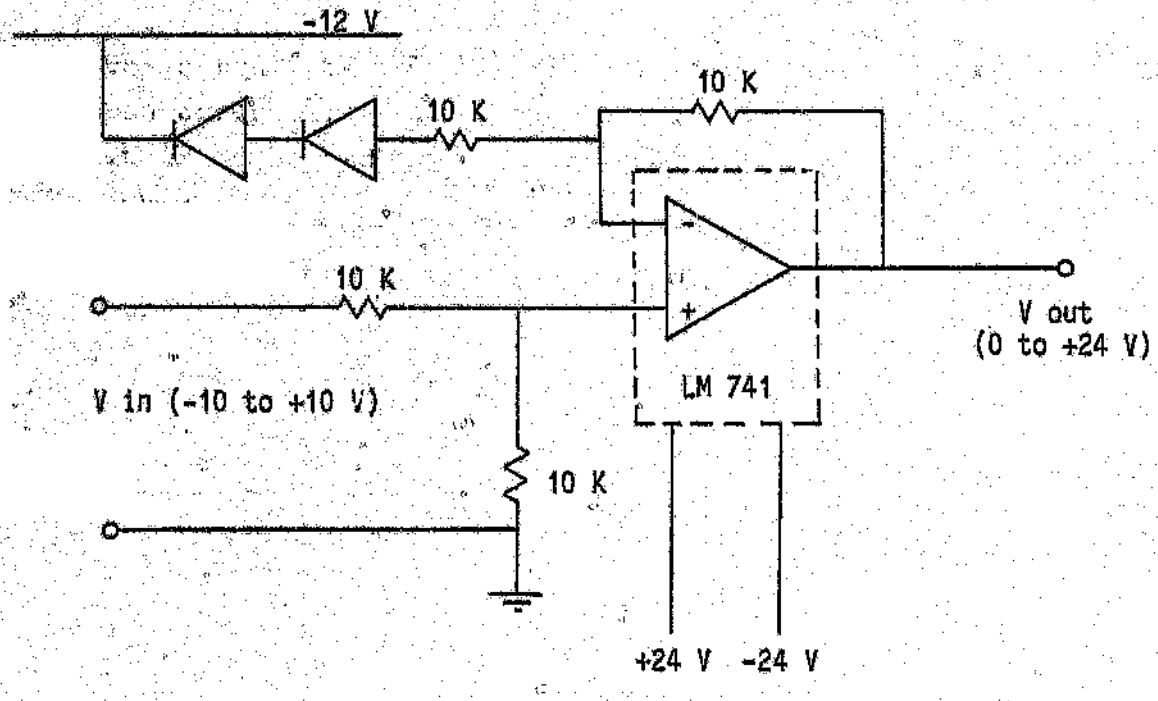


Fig B6.3 Voltage adder

THE PWM REGULATOR

An LM 3524 chip is used to control PWM regulation. The chip is wired as below.

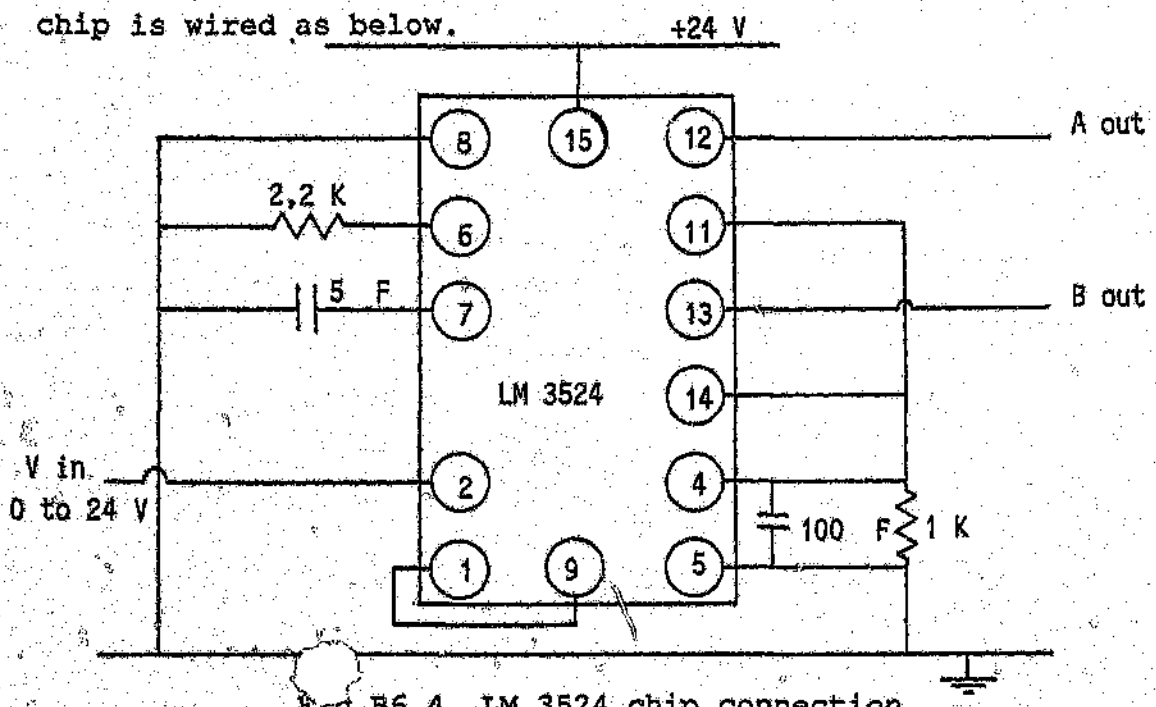


Fig B6.4 LM 3524 chip connection

THE DRIVER

A driver circuit is required to pull signals from the chip up to an acceptable level, for the inverting circuit. This circuit is connected to a +24 V power rail.

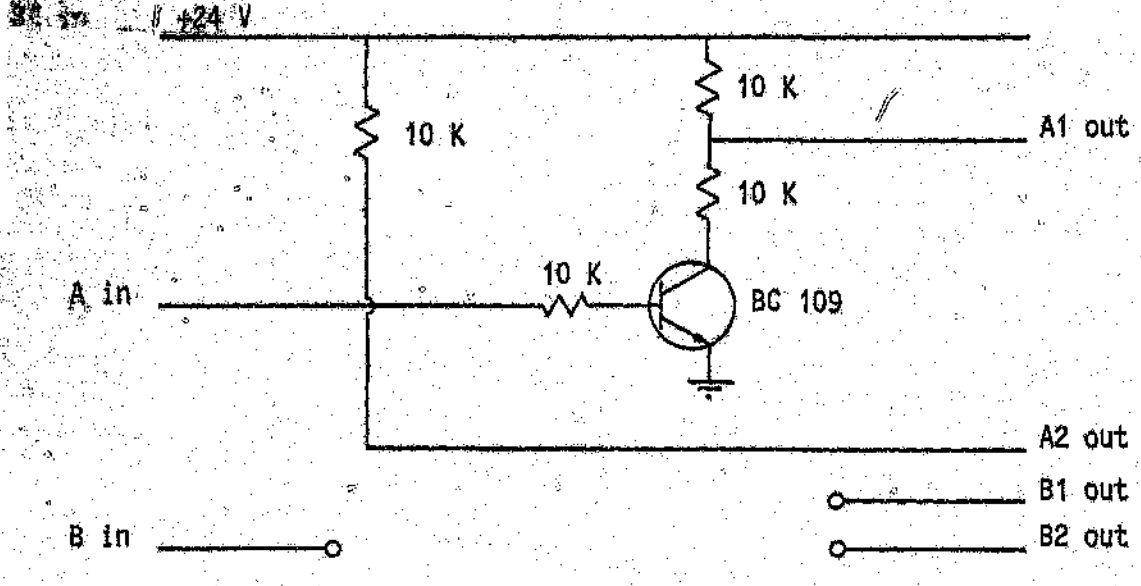


Fig B6.5 The signal driver

THE INVERTOR

The invertor takes the regulated, modulating signal produced by the chip, bolstered through the driver, and uses them to switch the output power supply between its low and high states. It is the ratio of mark time spent in each of these states that dictates the average output voltage. The 15 A diodes, as well as the 2N3055 transistors are to be mounted on heat sinks.

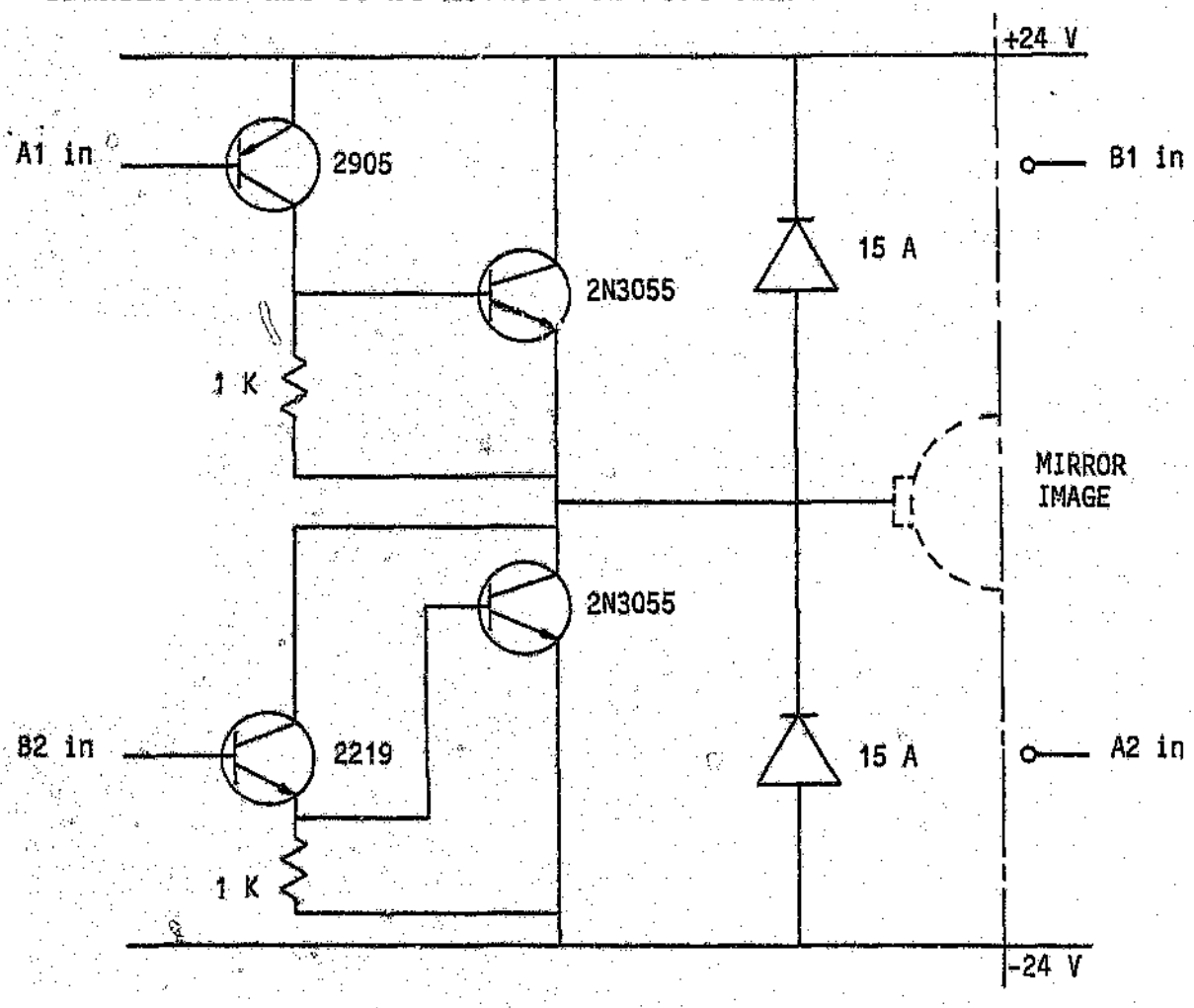


Fig B6.6 The invertor

A person constructing such a PWM must take note of the fact that there is a chip available that performs the task as well, especially designed for servo/encoder loop control.

The user of the PWM will be able to achieve the desired amplification from ±10 V at 0.8 A to ±24 V at 10 A.

EXTRACTS FROM COMPUTECH

An introduction to PLCs for the uninitiated

Almost every article in the technical press these days seems to mention programmable logic controllers (PLCs), and they all take it for granted that everyone knows exactly how they work. However, if you happen to be in a plant where PLCs are not used, or are in management and do not always get a chance to keep up to-date with the latest technology, this article is for you.

What is programmable control anyway? You are familiar with relay control. Relay control is a collection of contacts, coils, switches and input and output signals. This is how plants or machines used to be controlled, and in many cases still are. The control function is decided by the method of wiring up these elements, and this system can be thought of as wire-programmed, or wire-logic control (WLC).

What if all these relay elements are replaced by one standard unit, which is then connected only to the input and output signals, and easy to change control functions are included? Well, that is exactly what has been done, and it is called a programmable logic controller.

Great, let us start using it!

Firstly, those input and output signals should be divided into digital and analog signals. Digital are the on/off, high/low signals, which can only have two values, for example, the state of a pump. Analog signals can have one of a range of values, for example, the temperature of a liquid. In most applications, both types of signals are found.

A controller is therefore a method of taking input signals, combining them according to some sort of logic, and then transferring the result to the output signals. Input signals come from pushbuttons, limit switches, pressure switches and temperature sensors. Output signals are sent to contactors, motors, clutches, valves or even pilot lamps.

Wire-programmed logic was designed from a wiring or logic diagram, and each application usually had a different cabinet and internal elements. The PLC however, is a standard mass-produced unit, and smaller in size. The method of combining the signals is written into the program memory via a keyboard. So, instead of wiring up a cabinet full of relays, all that is needed is to type in a few statements. It is even easier to duplicate this process if more than one application is required, simply by copying the program to the other PLCs.

The AEC A020 PLC is typical of the smaller standalone PLCs. It is compact, maintenance free and can be programmed very quickly. It even checks itself and will indicate if there is a fault.

OK, but how does it work?

THe heart of the PLC is a microprocessor (MPU), not unlike those found in the IMB PCs that are commonplace today. The MPU does the calculating and the controlling of all the signals. To do this it uses a list of statements that have been typed in, which is known as the user program. It also has other statements which are loaded when it is made that handle the management of the whole device, and saves the user from doing all the work.

These user programs are stored in memory, and consist of a number of single control statements. The capacity of the memory is limited, and is available in amounts of 1024 statements, known as 1K of memory. The statements define exactly how the incoming signals must be combined to achieve certain output values. After processing the last statement the program starts at the beginning, continuing in this cycle indefinitely.

There is also another part of memory that stores any intermediate results. At the start of the program cycle a snapshot of all the input signals is taken and stored in memory. These values are used for all the calculations, and the results are stored in memory too, until they are transferred to the outputs. The value of timers and counters is also stored in this memory. The A020 has 16 timers and 16 counters.

Where can PLCs be used?

The applications cover a very wide field. They are found wherever there is something that needs to be controlled. The specialty of the PLC is the control by use of Boolean logic functions, and situations where the control logic must be changed often.

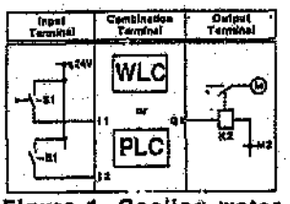


Figure 1. Cooling water temperature controller

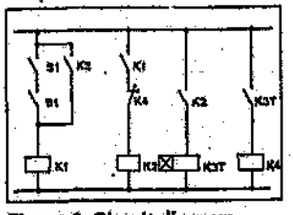


Figure 2. Circuit diagram

1. AND switch S1
2. AND sensor B1
3. OR contactor K2
4. = relay K1
5. AND relay K1
6. AND NOT relay K4
7. = contactor K2
8. AND contactor K2
9. = timer K3T
10. AND timer K3T (time reached)
11. = relay K4

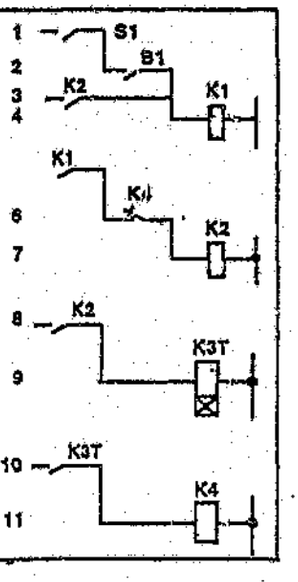


Figure 3. Logic statements

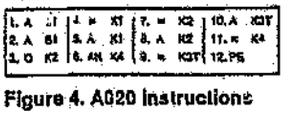


Figure 4. A020 instructions

Typical applications include:

- car washing plants;
- processing machines;
- ventilation plants;
- transporting plants;
- galvanising plants;
- batching plants;
- textile machines;
- packing machines;
- washing and drying plants; and
- tool machines.

Can we have a simple but practical example?

Assume you have to control a fan where the following situation exists.

The cooling water of an engine is checked by a temperature sensor (B1) (Figure 1). When the temperature limit is reached the sensor unit switches on a fan motor (M1) provided a key operated switch (S1) is switched on as well. A timer unit switches off the fan after 10 min, but only if the sensor indicates a temperature less than the limit points.

This can, of course, be done by conventional relay logic, but this is only a simple example. Consider the WLC solution first. On the signal input terminal of the controller there is the key operated switch (S1) (Figure 2) and the temperature sensor (B1). On the output terminal of the controller is the contactor (K2) of the fan motor. For the signal combination, the logic, this control requires two auxiliary relays and a timer. The n/o

and n/c contacts are wired up either in parallel or in series. The program is made by wiring up according to the wiring diagram.

The realisation of this control is only possible if, one after the other:

- ☐ the wiring diagram is drawn;
- ☐ the amount of elements (relays, contactors, timers) is clear;
- ☐ the wiring-tables and construction plans are ready;
- ☐ the cabinet for the housing of the elements and terminals is available.

Changes in the functions of the controller have always meant change in the wiring diagram and therefore in the wiring, and quite often also in the type of switching elements. The programming of such a control is done by screwdriver and wire.

The PLC solution is based on the AEC A020. The wiring diagram can be the same. If the logic of the single current paths are transferred into control statements, the results are as shown in Figure 3. Now, this can be condensed into PLC notation to give 12 short PLC control instructions (Figure 4).

These 12 control instructions are the bases for the user program, which will be written into the program memory. The program language is standardised (DIN 19239). According to that a control instruction consists of an operator and an operand. The operator part of an instruction tells the PLC how to combine the signals.

Operators are, for example:

- ☐ A (AND);
- ☐ O (OR);
- ☐ AN (AND NOT);
- ☐ ON (OR NOT);
- ☐ = (relates to, that means switch on or off).

The operand part of an instruction tells the PLC which signals will be combined. An operand consists of an index and a parameter, and the A020 format is shown in Figure 5.

To allow the controller to handle the 12 instructions, the operator and the operand must be written in a way the A020 can understand. At first all operands will be related to terminals, or to intermediate storage registers (Figure 6).

To write the prepared program into the memory of the A020 in the form of an instruction list, a programming unit is needed. To do so any standard data terminal or printer with keyboard will do, provided they have a standard interface. The interface is required to connect the programming unit to the A020. It is only important that the required interface data of such a device fulfills the requirements of the A020.

And this really works?

Yes, and an important advantage of the A020 is that the complete programming and handling intelligence is part of the hardware.

Under programming intelligence is understood that the controller already knows all operators and operands with which to do logic control, such as: timers, counters, jumps and

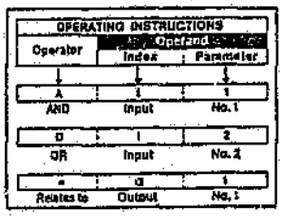


Figure 5. A020 instruction format

Key-operated switch	S1	= input	I1
Temperature sensor	B1	= input	I2
Relay	K1	= int points	M1
Relay	K4	= int points	M4
Timer	K3T	= timer	T1
Contactor	K2	= output	Q1

Figure 6. Input and output signals

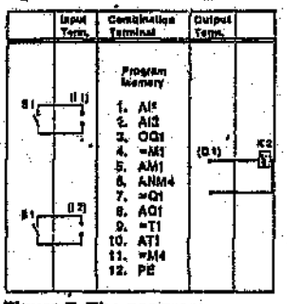


Figure 7. The program

logic commands.

Handling intelligence means that the necessary functions for the dialogue between operator and controller can be given via the keyboard of the programming unit in the form of characters.

This handling system guides the user by dialogue between the programming unit and controller. An important part of this section of the firmware is the checking of the right functional procedure. That means checking the input sequence and the proper input of operation and operand. Handling errors will be indicated on the programming unit.

But, back to the example. You have written the program for the fan control and also related the operands to the terminals of the A020. That was part of the design work, and is now finished. All that is left is the writing of that program, the 12 instructions, into the controller (Figure 7). At this stage you should be thinking to yourself that PLCs are not that complex. And you are right! What you should now consider is a proper training course, available at most PLC suppliers.

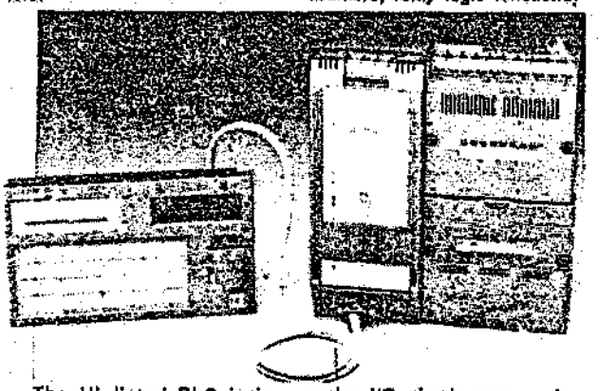
607-16-02 service coupon

PLC offers networking capability

Square D's SY/MAX Model 50 can be programmed with an IBM PC using SY/MATE software in addition to its own handheld programmer. The unit is also compatible with the SY/NET LAN for communication with other SY/MAX controllers.

"EEPROM or 4K EPROM are used. This allows users to send preprogrammed memory packs to a job site to accomplish program changes.

Other features are: 4-function maths; synchronous shift register commands; timers; counters; relay logic functions;



The UL-listed PLC is intended for applications requiring 32 or more I/O and can handle up to 256 I/O. Plug-in memory packs with 1K or 4K RAM, 4K analog I/O; simple progam development in Boolean-format.

The Automation Foundation

PC programming of PLCs

It is now possible to program both versions of electronic Festo sequence controls (the Step Controller and the FPC 101 19" plug-in card), not only with a plug-in programmer, but also with a PC, using the simple matrix language. Festo can supply the necessary software from its overall FST (Festo Software Tools) package.

Inputs and outputs are programmed with a X, 0 or 1. Each step can be assigned a time and a monitoring time (watchdog), as well as additional functions such as 'jump' and 'loop', and the output of texts via the serial interface to displays or printers. The 'step magnifier', a window system, allows each step to be displayed at increased size and provided with statements and comments.

The advantages of PC programming are menu-guided programming with 'help' functions, the facility for the programming of EPROMs, archiving and documentation, fast commissioning and testing, and the retrofitting of existing controls.

The method of carrying out the simple programming of a sequence using the plug-in programmer nonetheless remains attractive, offering the advantage of the teach-in mode and simple standstill diagnosis.

The controls can be supplied preassembled in housing, including power supply units and terminal strips for inputs and outputs. Preassembled solutions can also incorporate an interface card. This card carries integrated solenoid valves, which do not require tubing connections, to act as an interface to the pneumatic system.

Festo SA

607-21-05 service coupon

Bundling PLCs, computers and workstations

There is a move afoot to make changes in the packaging, architecture, and support of mid-to-high-level programmable controllers to achieve a higher-level industrial control appliance.

The lower end of the programmable controller (PLC) market has become increasingly dominated by compact, low-cost units that embody the concept of an industrial control appliance – a single device dedicated to performing all the ladder logic control functions required for a specific application.

Medium to large PLCs, on the other hand, are typically combined with industrial computers, industrial workstations, and local-area networks (LANs), as well as third-party and custom software packages. Each is optimised to perform a particular function. But when these devices must share capabilities – like database access or control and communications – the result is often a less-than-ideal solution from the standpoint of both cost and performance.

An extreme, though not unrealistic, example is an application requiring a PLC performing ladder logic, linked to an ASCII/Basic module, linked to an industrial computer performing nonladder logic control and other functions (MIS, data logging, etc), linked to an industrial workstation or other man-machine interface device – all connected to a local area network (LAN). In this case, the user must link as many as four different hardware products – each with its own channel of support. In addition to all of this, the user must deal with the multiple sources of software and software support that may be required.

Bundling PLC, computer, and workstation

There is a move afoot to make changes in the packaging, architecture, and support of mid-to-high-level PLCs. The driving force behind it is the desire for a higher-level industrial control appliance – a single, integrated unit able to perform most, if not all, of the functions that mid-to-high-level control applications demand.

The new units combine the traditional capabilities of a programmable controller, an I/O interface, an industrial computer, and (when appropriate) an industrial workstation for man-machine interface. They fully integrate the control, communications, information processing, and operator interface functions into a single control platform. Examples include GE Fanuc Automation's Cimplicity/CIMstar DX, Allen-Bradley's Pyramid Integrator, Vivid Technology's LD4000 Integrated Logic Controller, and Nematron's Ultra CIM. These can directly communicate with traditional PLC remote I/O and/or emulate a PLC, and can support displays and networks – typically DECNET.

The units are designed for applications requiring both a PLC and an industrial computer, and provide maximum efficiency when a display or operator interface device (ie an industrial workstation) is also required.

Making the marriage work

With scanner cards, an industrial computer can communicate directly with PLC remote I/O without the need for a PLC CPU. Allen-Bradley, Modicon, Texas Instruments, and General Electric (via GEnius I/O) currently offer these cards commercially; other suppliers may provide them as well. As an alternative, remote I/O processors, such as Transition Technology's TransPort I/O, Digitronics Sixnet, or Action Instrument's I/O PAK, and Opto 22's Optomux, can be linked in multiple combinations directly to the computer's serial port to provide coprocessing functions and I/O signal isolation.

SUPERVISORY SYSTEMS

BATEMAN INSTRUMENTATION

BATEMAN PROCESS INSTRUMENTATION LIMITED

AFE/ ELCON/ ONSPEC

607-13-01 service coupon

When they are used with appropriate software packages, such as the Flo-Pro Flowchart Programming System (Universal Automation), 86-Ladders (Wizdom Systems), or Relay Ladder Logic (Burr-Brown), industrial computers can emulate the functions of a PLC while providing generally higher performance.

As indicated before, simply emulating or replicating the PLC CPU's functions is not enough. The key is to be able to link the PLC I/O database directly and efficiently with program running on the industrial computer performing non-traditional PLC tasks – as well as ladder or flow logic. By sharing a common bus and memory, this can now be effectively and economically achieved. By locating all computational elements together both economy and performance can be improved.

Advantages of this approach include:

- ☐ reduced number of devices required;
- ☐ a universal, integrated programming panel;
- ☐ built-in operator interface;
- ☐ industrial-grade mass storage devices directly accessible;
- ☐ open architecture supporting third party hardware and software;
- ☐ compliance with de facto hardware and software standards;
- ☐ replacement of many hard-wired components such as buttons, lights, gauges, and loop controllers;
- ☐ compatibility with installed I/O;
- ☐ greater choice of application-specific hardware and software.

This architecture fills a need, and will probably become more popular with time. It integrates several items that work much better together and apart, and has the prospect of driving down the costs for users.

Nematron is represented in South Africa by Genwest.

607-13-02 service coupon

Industrial workstation links to most PLCs and becomes fully operational in less than an hour

Built into the unit are communications drivers for PLCs manufactured by Allen-Bradley, General Electric, Modicon, Square D, Texas Instruments, and Westinghouse.

Nematron provides the Series 100 Workstation with menu-driven configuration software, which runs on an IBM PC or compatible, and eliminates the need for user programming. The software's fill-in-the-blank prompts enable a user to select the PLC model and memory size; setup communications ports; edit descriptions of registers, timers, counters, contacts, and coils; save, retrieve, list, and erase files on the PC's disc drive; transmit information to printers connected to the PC; and download all entered data to the workstation's memory.

... the Series 100 workstation's four-line, 20-character-per-line display and 20-position keypad make it easy for plant-floor operators to view and change PLC setpoints. The unit can also be set up for alarm annunication, recipe entry, report printing, and other operator functions.

Powering the Series 100 workstation is an 8032-based microprocessor running at 11 MHz. The unit provides 8 k-bytes of built-in non-volatile memory for storing up to 100 screen files. Memory can be expanded to 24 k-bytes for storing up to 300 screens. The expanded-memory model also features a realtime clock.

Users can connect the Series 100 workstation to PLCs via its RS232 serial port. A self-powered RS422 converter is available as part of the workstation's connecting cable. Also featured are RS232 and parallel (Centronics-compatible) printer ports.

The Series 100 workstation's display area measures 75 mm wide by 25 mm inch high. Its sealed-membrane keys provide audible and visible feedback. Also on the front panel are LED indicators for 'Run/Fault', 'Alarm', 'Communications Active', and 'Keyboard'.

Cast in aluminium, the Series 100 workstation is sealed to meet NEMA 4/12 standards for dust-tight, watertight operation in harsh environments. It can mount in a NEMA 4/12 enclosure with a cut-out and four screws. The unit measures 260 mm wide, 150 mm high, and 50 mm thick. It weighs 1,6 kg.

Nematron pioneered the development of rugged, low-cost industrial operator interface terminals, which it calls Industrial Workstations. Nematron's products range from relatively simple display terminals to IBM PC/AT-compatible models with high-resolution colour graphics capabilities. Over 20 000 units are installed on plant floors worldwide. Customers include some of the largest automotive, petroleum, chemical, steel, textile, and food-processing companies, among others.

607-13-03 service coupon

PLC-based system cuts office lighting energy costs by 70%

A new lighting system, installed in electrical engineering offices, and controlled by a *Sprecher + Schuh* Sestep 390 programmable logic controller, is providing an energy saving of 70 to 80% over conventional office lighting. The 'movement-dependent illumination system' won the coveted energy-conservation Prix 'Eta', awarded to the originators, Herzog + Krull.

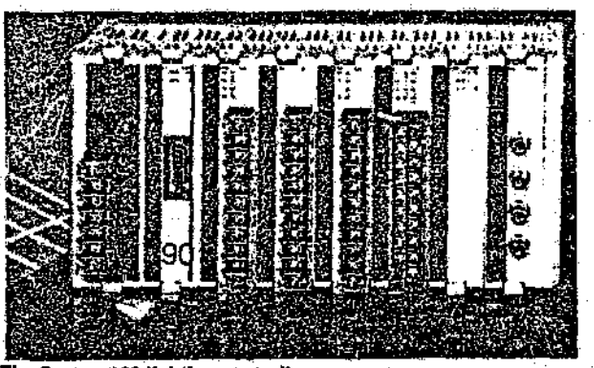
The new system has met with astonishing success in applications where large interiors must be illuminated to meet widely varying usage patterns. Typically this demand is experienced in sales rooms, exhibition halls, museums, warehouses, underground garages, passage areas in hotels and generally in most public or industrial buildings.

The high energy savings are achieved by the optimisation of light sources combined with a substantial reduction in illumination time.

Light sources are optimised by the combination of base lighting and accentuation lighting. Base lighting is fluorescent, with a high illumination yield, and infinitely regulated by high-frequency ballasts. The complementary accentuation lighting consists of low-voltage halogen spot lamps. In exhibition and sales areas in particular, this creates a pleasant, individually variable atmosphere. *Sprecher + Schuh* engineers installed a ribbon cabling system, in the ceiling, as a regular grid, so that lights could be arranged, and re-arranged, in any desired pattern.

The fundamental innovation in the system is its dependence on movement. Infrared sensors scan all areas and supply information on visitor frequency. On the detection, by the sensors, of movement, the PLC turns on the operating system and the illumination is raised to a pre-programmed level. If no further movement is detected within a defined time, the lighting in that sector is turned off. In addition to energy saving, this system can be used as a security surveillance system.

The *Sprecher + Schuh* Sestep 390 processes information from the infrared movement



The Sestep 390 lighting controller

sensors and controls the lighting of numerous, differently structured sectors, accounting for the difference between large areas and passage zones, the incidence of daylight, pause or night operation, and if it is being used in a security application.

The completely upward and downward-compatible Sestep family of PLCs plays an increasingly important role in the solution of sophisticated industrial automatic control tasks. Depending on the performance requirements, different controller sizes are available, from the modular compact Sestep 290 to the high-performance Sestep 690 which can control complex processes.

Sprecher + Schuh
(011) 493 5022
607-20-03 service coupon

PLCs automate aluminium producer

Nowhere in North America is more aluminium produced than at the Intalco Aluminium Corporation. Intalco's sprawling facility churns out 293 000 t of aluminium annually.

Focal point of the 23-year-old plant is a series of six long, narrow buildings resembling oversized chicken coops. These buildings contain the potlines consisting of 720 electrolytic cells each producing more than a ton of molten aluminium every day of the year.

A visitor to the plant in 1980 who returned today would not notice immediately the overwhelming changes in the plant's operations. But throughout the plant, older equipment is operating under the control of more than 70 programmable controller-based systems.

Ironically, though, the facility's first programmable controller was introduced into the plant as part of a major piece of equipment purchased from an outside supplier.

An order was placed for a casting machine. The manufacturer produced the machine with a programmable controller, rather than the usual relay board. Proving itself on that first piece of machinery, the programmable controller has quickly found its way into virtually all of the plant's key operations.

Those operations begin with creating carbon electrodes used to transmit the intense electric current (140 000 A) into the pots. With one of each pot's 18 electrodes requiring changing daily, Intalco's electrode manufacturing process is not nearly as large and complicated as the smelting process itself.



Aluminium is smelted in 720 electrolytic cells housed in six buildings that form the heart of Intalco's facility. Each pot requires a constant electric current of 140 000 A to liquify the metal

In the potlines, each of the 48 cranes that service the pots has its own controller. Those cranes are used to add alumina (pulverised and chemically extracted from bauxite) to the pots, break the crust that forms on top of the melted metal, change electrodes and other general service functions. In the casting house, where the liquid aluminium is formed into various shapes and sizes of ingots, many casting pits and saws are controller operated. And finally, the plant's extensive air and water pollution control systems are mated to programmable controllers.

The 50-person staff splits the time spent working with programmable controllers between installation of new systems and replacement of the plant's existing controllers with Square D units. The existing controllers were made redundant by supplier, and their new model would not run the old programs.

The Square D controller was chosen for two reasons, the commitment to compatibility throughout the product line and the service support.

The staff have all attended 2-day training courses and do all the PLC programming in-house.

While the company intends to continue expanding the plant's programmable controller systems, their approach emphasises reliability. For example, in the carbon electrode production area, the 23-year-old hard-wired panels are replaced with 12 programmable controller systems. Although functions handled by the panels could be managed by fewer programmable controllers, Intalco disperses tasks across more controllers to provide greater overall dependability.

This task-specific philosophy ensures a 24 h a day, 365 days a year safeguard.

The Automation Foundation
(011) 493-6260/1/2
607-19-03 service coupon

Pyramid Integrator improves control of continuous steel casting process

The major industry growth sectors adopting automated manufacture as a solution to ensure future survival are forecast to be automotive, electronics and machinery. However, the first UK company to utilise Allen-Bradley's Pyramid Integrator is not in one of these areas, but at Sheerness Steel, which perhaps indicates how process-industry management has a better understanding of the benefits of automation. In the process industries, companies cannot survive without automation as their managers have understood for many years.

Sheerness Steel is a 750 000 t/a producer of quality profiled steel, in bar, rod and billet form, using recycled scrap from car crushing plants as its feedstock.

The Sheerness Steel plant on the Isle of Sheppey was designed and equipped to produce high yield, quality assured but low cost steel to meet the demands of today's international markets.

Commissioned in the early 1970s, the plant has been at the forefront of the application of the computerised manufacture of steel.

The recent installation of a dual A-B Pyramid Integrator (PI) system there has enabled the company to achieve better control of its continuous casting process, integrating data between the plant floor PLCs and supervisory level computers.

Utilisation of the PI concept to integrate all process and management functions is part of an overall strategy to eliminate plant floor manual data entry and make the whole system multivendor compatible, irrespective of hardware type (IBM, PDP or VAX). An electronic data interface for customer order entry is also planned.

The scrap steel is melted and refined in two electric arc furnaces, rated at 80 MVA and 46 MVA respectively, and a 95 t ladle furnace. Two 4-strand casters are used for the continuous casting process, each line having a Pyramid Integrator interfacing with the five Allen-Bradley PLC-5 processors used to govern casting on that line. One PLC is allocated to each strand, controlling all operations specific to the casting process, such as mould oscillation, strand speed, cooling, straightening and flame cutting. The fifth processor on each line is employed for general purposes common to all four strands, such as temperature control.



A PLC-2/30 is used in the ladle furnace where liquid steel undergoes secondary treatment prior to casting

Additionally, a PLC-2/30 is used in the ladle furnace where liquid steel undergoes secondary treatment prior to casting. Molten steel from the primary melt is further treated in the secondary furnace to achieve the required steel quality and composition before being passed to the casting moulds.

All the PLCs sit on data highways and interface with the two Pyramid Integrators which serve primarily as comprehensive gateways to the Ethernet LANs. In this way, total communication across the entire 'concast' process network is provided, the lines being managed by a VAX 3300 processor.

Information processor

Data is transmitted between the foundry and main administration block (from the Allen-Bradley MicroVax information processor within the Pyramid Integrator to the VAX 3300) using fibre optics, connecting at each end to Ethernet cable.

The unique flexibility of the Pyramid Integrator, which typically contains the MicroVax information processor, a PLC-5/250 programmable controller and a configurable vision input module (although only the MicroVax information processor is used at Sheerness, to improve throughput and response times) in a single chassis, makes it ideal for the 'concast' process application.

The MicroVax information processor module provides software utilities for close coupling between hardware and software, from the plant floor to control information processing. It is a multitasking, multiprogramming computer processor capable of performing both complex calculations and supervisory functions.

Programmable controller data is translated by the information processor module to the higher level VMS format used by the DEC MicroVax computer and back again through a shared memory interface. Features include 8 Mbytes of RAM, one DECnet/Ethernet port and a four-channel RS232 port.

At Sheerness Steel, the PLC I/O table is inputted to the Integrator every half second with a response within less than 100 ms. If required the speed of the data throughput could be increased even further.

Management information

Project-managed by Scomagg, one of the independent engineering companies participating in the Allen-Bradley Pyramid Solutions Programme, installation of the system was completed within two weeks and without the loss of any production.

A wide range of information is now available to Sheerness Steel as a result of integrating the overall system, giving management much tighter control of the business. For instance, caster performance, material tracking, batch identification stock is further processed by hot rolling to form the reinforcing bars, coiled rod, merchant rounds, angles, flats and other sections produced by the plant.

In the bar mill, control and management of billet re-heating to more than 1000 °C, delivery of billets to the mill and the speed of continuous rolling is facilitated by the use of PLC-2/20 and PLC-5 processors. Instructions are input by the furnace operator at the DEC Vax level and transferred by serial link to an Allen-Bradley PLC. Extensive refurbishment of the rod mill currently taking place will enable similar use of PLCs to achieve the required level of rod manufacture, control and management.

For more information on the Pyramid Integrator concept contact the local agent, Flexible Automation Systems, at telephone (011) 887 0620/1/2 via the service coupon.

607-19-02 service coupon

Charcoal briquet maker improves productivity with PLC-based interactive automation

Information from Square D

The experts at the Kingsford Products Company recently joined in the design of automation improvements to their charcoal manufacturing facility. Kingsford wanted to improve quality and increase the productivity of its Kingsford Charcoal briquet plant in Burnside, KY, USA.

Kingsford utilised the expertise of its own employees, calling upon Burnside's equipment operators and engineers to build a foundation for new automation systems and equipment.

Their work came to a climax late last year, when during a 10-week shutdown more than $5 million in improvements were installed in the facility. Included in the improvements was a control system based on the SY/MAX programmable controller family and ScreenWare2 colour graphics software. These improvements gave operators greater control over the briquet manufacturing process while increasing productivity.

A better idea

The origin of modern charcoal, like modern automation, traces its roots back to Henry Ford. In the years prior to 1920, Ford operated a sawmill in the woodlands around Iron Mountain, MI, to make wooden parts for the Model T automobile. Ford saw the growing piles of wood scraps, and felt there had to be a way to use them.

In 1921, after refining a method for turning wood chips into charcoal briquets, Ford opened his first charcoal manufacturing plant. E.G. Kingsford, who owned one of Ford's earliest automobile dealerships and was distantly related to Ford by marriage, was later named Manager of the briquet operation.

A series of acquisitions and equipment transfers evolved into the present-day Kingsford Products Company. Today, the Kingsford Products Company is recognised as the leading charcoal briquet manufacturer in the USA. A Fortune magazine survey recently identified Kingsford's briquets as one of the '100 products that America makes best'.

That kind of tradition for quality translates to consumer demand, and in Kingsford's case the need to increase the productivity of the Burnside plant. With the plant operating 24 h per day, the purchase of additional equipment was a logical starting-point.

Paul Cundiff, a plant engineer, notes: "We recognised that to achieve our goals, we had to integrate the various stages of our production process. The whole plant, people and equipment, needed to operate as a unit."

Before anything was on the drawing-board, Kingsford operators were asked: "What do you need to know from your equipment to do your job best?" The answers comprised a process control wish list – feed rates, speed rates, temperatures and pressures, trends, levels, the status of material handling equipment and the status of interdependent equipment up and down the line.

Interactive production

Assisted by equipment purchased from a Square-D distributor, Kingsford took what operators designed and, 1650 rungs of ladder logic and 3300 I/Os later, gave it to them in a highly interactive SY/MAX control system.

Three SY/MAX model 700 programmable controllers, a SY/Net local area network, ScreenWare2 software, three SY/View workstations, and several IBM PC/AT-compatible computers, provide operators and engineers with a hand in the manufacturing process they did not have before. The SY/MAX systems, along with new wood drying, briquet drying, conveyor and packaging equipment, completes the set of tools Kingsford needs to improve productivity.

Trucked into the plant from area lumber mills, wood by-products continue to be the main raw material in Kingsford's charcoal briquets. The manufacturing process begins with the loading of wood by-products on a conveyor that leads into the retort area. In the retort area the wood by-products are dried through heating in a new rotary dryer, then turned into char, or carbonised, in the controlled environment of a retort.

The startup and shutdown sequences of the rotary dryer and retort feed system are especially important. The control system is set to respond to 35 different alarm points.

In the briqueting area, ingredients that include char and a starch binder are blended and pressed into the familiar briquet. The SY/MAX system regulates the mixing of the binder for the charcoal blending process.

Once the briquets are formed, they are moved by conveyors into a dryer. The pride of the Burnside plant is a new, more efficient briquet dryer. Here briquets are dried at a controlled rate to ensure they perform to consumer satisfaction.

Explains Larry Decker, Briqueting Managing: "We must have control to ensure that all our briquets are precisely and uniformly dried, and so that our customers are completely satisfied with their performance."

A SY/View workstation displays screens that detail events leading up to, through and past the briquet dryer. Equipment diagrams on the screens change colour to tell operators their status – green for operating and yellow if not able to operate as intended.

The control of temperature in the zones through thermocouple inputs is of critical importance. When the temperature in a dryer zone reaches a danger point, the zone turns red on the SY/View monitor, alerting operators to take action. The motor currents of fans are displayed and the operation of dampers that bring heat to and from the briquet dryer are also controlled and displayed.

The dry briquets are transported by material handling equipment to storage silos before packaging. Feeders and conveyors link the silos to small hoppers above the packaging scales. Packaging control room operators use a SY/View workstation to monitor and control the packaging system, selecting silos, the route for the conveyor system and feed rates into the packaging scales and other equipment. Packaged briquets move down the line to balers and then onto automatic palletisers where they are placed on pallets.

Tons of achievements

Regarding the installation and startup of the new equipment, Paul Cundiff says operator involvement played a critical role. "We started up all areas in less time than most other companies would use in starting up a single area. Most of the control logic was tested and approved by the operators before completing the setting and tie-in of the mechanical equipment."

Three months after the new equipment and control systems were in place, Kingsford's operators and engineers pointed proudly to a major achievement. During the first month after startup, they had produced almost double the first month's scheduled production.

Kingsford's operators continue to look for ways to use all their capabilities and those of their equipment. Larry Decker notes: "The automation of our production represents a shift in cultures. We wondered how our operators would take to automation. That is no longer a question. Now they are pushing us for even more."

Square-D products are marketed and supported in South Africa by The Automation Foundation CC.

607-23-03 service coupon

The first step in understanding motion control

Control systems are enjoying a tremendous growth and many of these applications are coming from engineers new to this field. The purpose of this document is to describe in simple terms: a control system, the working thereof and the various components that can be used in such a system and when to use it.

A control system is an arrangement of physical components connected or related in such a manner as to command, direct, or regulate itself or another system.

One of the most simplest tasks for the human being to perform, to control motion, is a difficult subject in the mechanical/electronical engineering field. A child of six months can already move his hand to an object he wants to take hold of. In this basic example of a closed loop control system there are three basic elements identified: the brain, the arm and the eyes. The brain must give a signal to the arm to move closer to the object, and the eyes must tell the brain when the hand has reached the object.

The comparison between the above-mentioned system and a closed loop control system is quite evident and must be seen as follows:
brain = servo amplifier
eyes = feedback devices
arms = actuators.

The brain uses the position of the object as the input signal to the system and the position of the hand, which is detected by the eyes, as the position feedback signal. The objective of the control system is to move the hand to the object and therefore reduce the distance between the hand and the object to zero. This means that the brain will continue to send a signal to move to the hand, until there is no difference in position between the hand and the object.

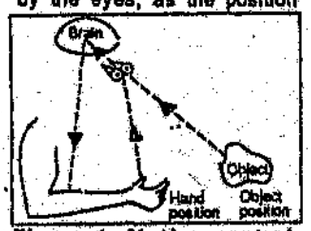


Figure 1. Motion control principle

In a control system the difference between the input or command signal, and the feedback signal is called the error signal.

A control system can be made up out of three basic elements as mentioned in the paragraphs above (see Figure 2).

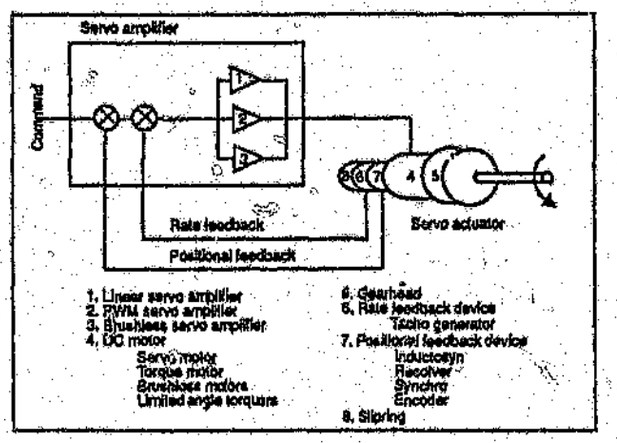


Figure 2. The components of a servo system

Servo amplifiers

The servo amplifier can be seen as the brain of the control system and is the driving force for the actuators. It compares the command signal with the feedback signal and depending on the outcome, control the power delivered to the motor. Servo amplifiers can be divided into three categories namely linear, pulse width modulator and brushless servo amplifiers.

☐ *Linear servo amplifiers*
This type of amplifier varies the voltage applied across the armature and dumps the unnecessary power into a power dump which makes it inefficient.

☐ *Pulse width modulator servo amplifier (PWM)*
A constant frequency variable duty cycle pulse train is generated. This pulse train is then used to switch the voltage over the armature.

☐ *Brushless servo amplifier*
A brushless servo amplifier needs additional circuitry to electronically commutate amplifier power to the motor windings for rotation to occur. A brushless servo amplifier provides electronic commutation, feedback and command signal comparison, and the power needed for brushless system control.

Due to the switching nature of the PWM amplifier EMI levels are high which is a disadvantage. But then there are many advantages in a PWM amplifier which can be summed up as good power dissipation, smaller size and high efficiency.

Feedback devices

As explained in the human model, feedback devices are used to close the loop in control systems by measuring the speed or position of the output. This is done by means of signals that can be related to the speeds or positions. This signal is then fed back to the servo amplifier which will compare it with the input signal, and if there is a difference the servo amplifier will produce an error signal and will then control the power to the motor in order to reduce the error signal to zero.

Position feedback devices

☐ *Potentiometers*
This is the most basic position feedback device and gives an analog voltage proportional to the position of the shaft. Typical accuracies and linearities of a potentiometer are not high but it is a simple device to use. It is an absolute feedback device which means that it eliminates wake-up problems.

☐ *Optical encoders*
By making use of a transparent wheel with lines on it, a wheel with splits, and and light cell a pulse high voltages and low voltages can be generated when turning the wheel. By knowing the number of lines on the wheel and by counting the voltage peaks it is possible to determine the angle through which the wheel has turned. By counting the number of pulses in a discrete time, velocity signal can be obtained.

☐ *Contact encoders*
This encoder consists of a disc with a pattern of conductive and non-conductive areas. These areas are energised and as the disc rotates, a number of brushes are arranged in a meaningful manner a digital word can be produced which is a direct indication of the position of the shaft.

☐ *Synchros and resolvers*
Both these types of position feedback devices use the same principal of operation, which can basically be described as a revolving transformer. This means that it consists of a winding, capable of revolving in a fixed stator. When an AC reference voltage is applied to the rotor winding, the voltages on the stator windings will be proportional to the trigonometric relationship between the angle of the fixed windings, and the field set up by the rotor winding.

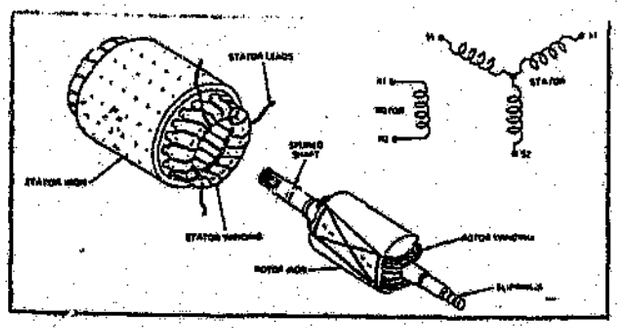


Figure 3. Internal structure of a synchro control transmitter and its electrical representation

As the input shaft rotates, the field rotates and the output voltages vary proportionally. The output can then easily be converted into an absolute representation of the shaft position. The basic difference between a synchro and a resolver is that in a resolver the windings between the rotor and the stator are displaced mechanically at 90° and 120° in a synchro.

☐ *Inductosyns*
An inductosyn is an electrical geared resolver and consists of two discs with printed circuits on it. The printed circuit is arranged in such a manner that the same interaction as in a resolver is produced between rotor and stator. Because of the repetition of the pattern the sine and cosine waves will repeat themselves the amount of poles there are on the inductosyn, resulting in a much more accurate position sensor.

Before a designer can decide which feedback device to use, he must first find out whether he needs absolute feedback and what accuracies he needs. Table 1 gives an indication of the typical accuracies.

Rate feedback devices

The tachometer generator generates a voltage which is linear to the velocity of the rotor. It can be divided into three types, the AC tachometer, the brushed DC tachometer, and the brushless DC tachometer.

Velocity feedback can also be supplied by differentiating a position feedback signal. In order to decide when to use which type of tacho it is necessary to look at the accuracies as to what extent the velocities can be controlled.

AC tachometer: 1%
Brushed DC tachometer: 0,1%
Brushless DC tachometer: 1%
Incremental encoder: 1–10%.

Typical applications are:
☐ AC tachometer: instrument servo mechanisms,
☐ Brushed DC tachometer: robots and machine tools,
☐ Brushless DC tachometer: high speed and severe environments,
☐ Position feedback: low cost.

Actuators

The DC motor uses direct current and permanent magnets to convert electric energy into torque. This torque can be used for speed or positions control and can be seen as the workhorse of control systems.

DC motors can be divided into the following categories:

☐ *Torque motors*
A torque motor normally has a large diameter and narrow width, with a large number of poles. Due to the large number of poles this type of configuration produces high torques in a small space. The large number of poles unfortunately causes more commutation arcing at high speeds. Therefore a torque motor is typically used for positioning systems and low speed applications.

☐ *Servo motors*
A servo motor is the opposite of a torque motor and is longer with a smaller diameter. By increasing the length of the motor the torque available is also increased. But because of the smaller diameter of the motor the inertia of the rotor is kept low. This means increased motor response and therefore a servo motor is suited in high speed application.

☐ *Brushless motors*
A brushless motor consists of four basic components, the armature or stator, the field or rotor, the position feedback or hall devices, and the electronic commutation amplifier.

This type of motor makes use of electronic commutation which is accomplished with a switching amplifier which energises the various motor poles depending on the position of the rotor. The position of the rotor is detected by the position feedback device.

Advantages are that the machines can operate over a broad speed range and in special environments, they have lower rotor inertia, therefore higher acceleration, and have better cooling of the armature with no maintenance.



Figure 4. Operating principles of the (inductive) inductosyn

Table 1.

Type of transducer	Accuracy
Rotary inductosyns	1,5 arc secs
Multipole synchros/resolvers	7 arc secs
High accuracy (10 s) synchros/resolvers	20 arc secs
Absolute optical encoders	23 arc secs
Selected (2 arc min) synchros/resolvers	3 arc mins
Standard synchros/resolvers	7 arc mins
Potentiometers*	7 arc mins
Incremental optical encoders	11 arc mins
Contact encoders	26 arc mins

* Assuming 14-bit resolution
Note: In the above table, mechanically geared coarse fine synchro/resolver systems have not been included.



Figure 5. Typical brushed DC tachometer characteristic

☐ *Gearheads*
Whenever it is necessary to reduce the motor speed or increase the torque a reduction ratio can be used. Although the modern tendency is to go direct drive because of the zero backlash and higher efficiencies, it is not always possible. If it is necessary to use a gearhead the following typical specifications are available. The unit has zero backlash, very high torsional stiffness, high efficiency, compact design, high single stage reduction, long life, and high position and repetitive accuracy.

☐ *Sliprings*
Sliprings are not as such a part of the actuator, but form an integral part of control systems and are used in synchros, resolvers, inductosyns, gyros, rotating antennas and all types of rotating devices requiring the transmission of electrical energy where the use of wires or cables is not practical.

Conclusion

Before the designing of a control system can start, it is essential that the engineer must choose the right components (this for example means what type of motor and not which motor) and therefore the following questions need to be answered first:
☐ position or rate servo?
☐ torque requirements?
☐ speed requirements?
☐ what type of motor?
☐ what type of servo amplifier?
☐ what type of feedback device?
☐ is a reduction ratio necessary?
☐ is backlash critical?
☐ is size critical? and
☐ is it linear or rotational motion?

Directions for the unexperienced designer, to find answers for the above-mentioned questions, were given in this paper. It also explains a control system, the working thereof and the various components used in such a system. There are many excellent papers available, which in detail describe all the aforementioned components and can be made available on request. There will also, in the future, be follow-up papers which will lighten the subject of control system future.

Tio-Lec
(011) 51 8395

610-31-01 service coupon

Author: Bryson Craig Weir.
Name of thesis: A Feasibility Study In The Development Of An Offline, Pic Based, Robot Control System.

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2015

LEGALNOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page)for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.