THE AUTONOMOUS NAVIGATION OF AN ELECTRIC VEHICLE

ALISON BAKER

Thesis submitted for a

DOCTORATE OF PHILOSOPHY

at

THE UNIVERSITY OF ASTON IN BIRMINGHAM

SEPTEMBER 1983

The University of Aston in Birmingham.

THE AUTONOMOUS NAVIGATION OF AN ELECTRIC VEHICLE.

Summary.

A navigation and positioning system for an electric automatic guided vehicle has been designed and implemented on an industrial pallet truck. The system includes an optical sensor mounted on the vehicle, capable of recognizing special markers at a distance of 0.3m. Software implemented in a Z-80 microprocessor controls the sensor, performs all data processing and contains the decision making processes necessary for the vehicle to navigate its way to its task location. A second microprocessor is used to control the vehicle's drive motors under instruction from the navigation unit, to accurately position the vehicle at its destination. The sensor reliably recognises markers at vehicle speeds up to $1ms^{-1}$, and the system has been integrated into a multiprocessor controlled wire-guidance system and applied to a prototype vehicle.

Keywords: NAVIGATION, AUTOMATIC, VEHICLE, CONTROL, POSITIONING.

Author: Alison Baker.

Submission for the degree of Doctor of Philosophy, 1983.

CONTENTS.

Page

LIST OF ILLUSTRATIONS.

LIST OF TABLES.

LIST OF PLATES.

Chapter 1

INTRODUCTION

This course of study was undertaken as part of the research activities of the sponsoring organisation Cableform Limited, under supervision of the Interdisciplinary Higher Degrees (IHD) Scheme at the University of Aston in Birmingham. The scheme aims to bring together academic research and industrial needs for mutual benefit and to equip the student with problem solving techinques that can be applied across the spectrum of industrial enterprise.

Cableform Limited specialises in the design, development and manufacture of control systems for battery electric vehicles, supplying equipment to a wide range of truck manufacturers in the U.K. and abroad. They envisaged large changes in mechanical handling and storage methods with the onset of the microprocessor 'revolution' and saw this as an opportunity to widen their product field. In the short term, microprocessor based versions of present products were to be designed, but in the long term, automatic control of electric vehicles was seen as a natural extension of Cableform's involvement in their existing markets. A research department was therefore set up to investigate the automatic vehicle field and develop a series of prototypes that would enable the company to capture a market share. The team assigned to this project at the outset consisted of one full-time engineer, two post-graduate I.H.D. students and one technician.

An initial survey of companies producing automatic control equipment indicated that the major interest was in rail and wire-guided vehicles and the future development of these systems was thought to be a free-roving mobile robot with automatic load handling capabilities. Such machines have long been heralded as the answer to the automated factory dream in both industrial and academic spheres, and yet commercial automatic vehicles of today are still tied to special routes and have limited loading facilities. Automatic positioning and navigation systems are primitive in practice, although much research has been carried out in the fields of 'Artificial Intelligence' and 'Computer Vision'. Systems that do employ the fruits of this work are either built into highly specialised demonstration vehicles or rely heavily on large computers to process complex algorithms. There is also a trend towards 'Software Simulation' whereby theories are implemented and tested in computer models without the expense and inflexibility of building prototype hardware. This can lead to a heavy dependence on computing power which is difficult to implement on a vehicle. From an industrialist's point of view, a suite of programs on a computer is not as attractive as a working truck, and there is a need for practical demonstration of advances in vehicle automation to be made in the industrial setting in which it is used.

Research policy at the sponsoring company affected the project in two ways: firstly, the technology employed was to be microprocessor-based digital systems, and secondly, the scope was limited to developing individual control systems and not a custom vehicle. A Z-80 software development system was already available at the company, and this microprocessor was adopted throughout the project. The second constraint stemmed from the company's present activities and its desire to expand into present markets rather than to embrace new ones. Those markets are truck manufacturers and not truck users, so products are designed to fit a variety of vehicles of different makes as part of the truck manufacturers' own customization processes. This has led to a modular concept of basic building blocks from which many systems can be built by combination, and the concept was naturally applied to the automatic vehicle project.

The author was involved to a varying extent in all aspects of the project except load handling. The first task was to develop the concept and proposal for the automation system, which involved all members of the research team. Subsequently, the author assumed specific responsibility for the navigation and positioning functions.

By the end of the period covered by this thesis, various microprocessor based systems had been added to the chassis of an industrial pallet truck to produce a prototype Automatic Guided Vehicle (AGV). The vehicle followed an inductive wire loop laid on the floor and travelled at speeds up to $0.7ms^{-1}$. Navigation was achieved by using an optical sensor to read markers placed alongside the route to identify key locations, and a special set of markers was used in conjunction with this sensor to aid positioning of the vehicle at workstations. Loads could be transferred to and from the vehicle by means of an on-board conveyor. The navigation and positioning system was not specific to the prototype AGV and could be used on any guided vehicle regardless of the method of route indication (rail, wire, white line or other).

The following chapters describe the design and development of the navigation and positioning system and its integration into the prototype vehicle. First, the scope of vehicle automation in the mechanical handling environment is introduced with a review of the technology available for further development. Then the concept of an AGV is discussed and the proposal for the Cableform research programme is presented. This is followed by the design detail of the various system components and how they interact to navigate the vehicle. Finally the work is evaluated in a discussion of field tests and results.

Chapter 2

THE AGV WORLD

2.1 Commercial AGV's.

AGV's have a long history of industrial use, dating back to 1953 when a towing truck was modified by Barrett Electronics to follow an overhead guide wire (1). Today they fall into three basic categories: rail-guided (either on the ground or overhead or both), in-floor towline, and route guided (for instance a current carrying conductor buried in the floor).

2.1.1 Rail-guided Vehicles.

These are found specifically in narrow-aisle high-rise warehouse applications and on assembly lines. Control is effected by a remote operator or computer utilizing the A.C. power feed along the rail as a communications channel. Semi-automatic versions exist which take care of horizontal and coarse vertical positioning, leaving the driver free to concentrate on fine positioning and loading. This is a common feature of order-pickers where small quantities of goods are required to be collected from a pallet load or storage bin.

2.1.2 In-floor Towlines (2,3).

Towline systems are used as a cost effective alternative to conveyor belts or manually driven pallet trucks where there is a high throughput of goods over large distances - between short-term stores on assembly lines and in parts depots for instance. Operation of these systems requires a slot to be cut in the floor to carry the towing chain. Tugs are then hooked onto this chain to be moved along the route, and automatically decoupled at work-stations and stores. In complex systems with several vehicles, the route is divided into zones monitored by stationary controllers. These are distributed around the route and can stop the tow-line in their particular zones to prevent congestion and collisions. The major drawback of this form of automatic load transport is the disruption of the floor which prohibits the use of manual trucks in the same area, and renders alterations in layout impractical.

2.1.3 Route-guided Vehicles (1).

These are the most popular and widespread AGV's and find applications ranging from multi-truck installations in automated warehouses (4) down to a single tug shuttling between a parts store and a machine shop in a small company.

In the majority of systems, the guide is a current carrying conductor producing a local magnetic signal which is received by sensors on the vehicle. This signal gives position information relative to the wire which is used to control vehicle steering. Some environments prohibit the use of a wire guide, and other methods such as white tape and optical sensors have been employed, but the same control systems and basic operating features can be realized in all cases.

Methods of control and load handling capabilities vary widely according to application. Appendix A contains a review of current commercial wire-guidance systems and associated companies.

2.2 Current Mobile Robot Projects - A Sample.

The industrial world has become used to wire-guided vehicles and is happily operating with them, although drawbacks and limitations have been expressed in many quarters. Several projects exist to develop additional sensors and increase flexibility, but to date, it has been left to the academic world to advance the AGV, release it from its guide-route and aim for a truly mobile robot. Contemporary work includes three major projects - two based in France and one in England - which are here described in turn and summarised for commercial applicability.

2.2.1 The HILARE Mobile Robot (5,6,7).

This project began in September 1977 with the object of using multiple sensors to help a mobile machine navigate through its environment. Each type of sensor was used for a specific control function and tasks were expressed in terms of these functions such that the sensors were used efficiently: cameras were not used for following a wall at a fixed distance when ultrasonic sensors could be used with much simpler data processing.

The vehicle was purpose-built to carry the requisite equipment and consisted of a triangular chassis with two rear drive wheels and a front castor. Two microprocessor controlled stepping motors provided traction control. Sensors comprised a video camera and laser range-finder to give a 3-D image of the environment for navigation, ultrasonic sensors for close range navigation and safety, and infra-red beacon triangulation for absolute position determination. On-board intelligence consisted of two microprocessors - one for the drive system and one for processing of sensor data - and a minicomputer for complex data processing, low-level navigation decisions and external communications. In addition a link was provided to an IBM 370/168 for use during learning phases and for complex decisions.

The camera image was contrast thresholded to pick out homogeneous areas such as doors, walls and cupboards. No attempt was made to identify these features, rather they represented possible obstacles or passages. Their position within the image was determined and a set of points within each area was decided upon for range measurement to give the navigation function sufficient information to decide on the next move. A laser range-finder was mounted with its optic axis coincident with that of the camera and its direction varied by stepping motor driven mirrors under control of the image processor. The laser was directed to each of the specified image points in turn, so building up a simple 3-D model of the field of view. A few points would suffice to detect an obstacle, or many points used to measure the width of a door.

Ultrasonic data was derived from ten emitter/receivers under microprocessor control mounted around the periphery of the vehicle giving a range of a few centimetres to 1.5m and resolution of 0.02cm. This data had two uses: collision avoidance by stopping the robot as soon as any obstacle was detected within some threshold distance; and close range navigation for tracking at a specified distance from a wall or surface.

Finally, the robot could determine its co-ordinate position within its operating environment by triangulation. Two infra-red emitter/receivers were mounted on a revolving base and scanned round. Three retroreflectors with special patterns for identification were placed round the boundaries of the test room and the robot measured the receiver angles at which reflected beams were detected, from which it could calculate its position.

In early 1981, a progress report (7) stated that simulation of the data processing techniques for segmentation of the video image was complete and the next step was to implement it in high-speed logic. The strategy for using the range-finder was under investigation, the ultrasonic sensors were working reliably and wall tracking had been demonstrated successfully.

2.2.2 Institute Nationale des Science Appliquees Project (8,9,10).

This robot was another purpose-built machine for investigating control and navigation of a mobile vehicle. The object was to build a minimum system operating under microprocessor control via compact, fast software, with automation of industrial conveyor trucks in mind.

The robot consisted of a circular base, 0.6m in diameter, mounted on two independently driven wheels and two castors, and carrying batteries, all the microprocessor and interfacing electronics, and the sensors. A simple robot arm was mounted on top, at a height of 1.1m above ground. Control was achieved with four microprocessors: a central processor and three dedicated units for motor control, arm control and ultrasonic sensor operation. This enabled the vehicle to perform two basic functions of motion and arm manipulation.

Vehicle motion was broken down into three distinct phases of Speed, Rotation and Translation, and any desired movement achieved by successive combination of these phases. In Speed mode, the vehicle moved in a straight line at constant speed by implementing identical speed control loops with each of the two drive motors, using the same speed reference input for each loop. Rotation involved turning the vehicle on its axis through a set angle by driving the two wheels in opposite directions for a set distance under position control loops. Finally, Translation used the position loops with identical reference inputs to move the vehicle in a straight line for a given distance. In both cases, position was measured by odometry using a shaft encoder.

The robot arm could only be moved when the vehicle was stationary and was used as a tactile sensor to test for obstacles ahead of the vehicle. If the arm collided with an object, the arm extension was recorded, and the arm retracted. If there was no collision, the arm moved to its fully extended position before retracting. For collision avoidance, the vehicle could be moved with the arm extended forward to give advanced information of obstacles and allow the vehicle to stop or take evasive action. Obviously, the arm had a limited area of view, so a fender was provided around the vehicle base to give a stop alarm in the event of a collision.

A single ultrasonic sensor was provided at a later date, again aimed at the simplest system both in terms of hardware and software. However, it did include a self adapting system for dealing with the large variations in range encountered, and a self-correcting system to cope with variation in errors over this range. Problems with multidependence of the reflected beam on surface texture and angle of reflection, and blind spots in the image due to the transmitter/receiver manufacture, led the research team to consider using additional sensors rather than design more complex software to interpret these signals.

System control, implemented through the central processor, was a simple interrupt structure, with each sensor having a priority ranking. Communications to other processors was via set commands and could only occur when the vehicle was stationary and therefore not receiving any interrupts from its sensors. This proviso meant that commands had to be sequential and the current task completed before the next command could be given.

Future investigations appear to be absolute position determination via odometry or optical triangulation from beacons.

2.2.3 Warwick University Project (11,12).

A forklift truck manufacturer has sponsored this project with the aim of developing an automated forklift truck. Work to date has resulted in a prototype free-roving vehicle and it is hoped to apply this experience directly to a forklift truck in the next two years.

The commercial links are apparent in that the prototype is a three-wheeled industrial electric tow-truck which has been converted from manually driven to automatic operation. The major objective was to produce a navigation system that was independent of the environment so the vehicle could be truly free-roving, and this has resulted in considerable on-board intelligence being employed in the form of a minicomputer complete with floppy disc for storage. Odometry and sonar are currently the sole inputs to the navigation system and are used in conjunction with a software map stored on the vehicle, to give position at any instant. From the map and knowledge of its position, the navigator works out a strategy to reach its goal location, and implements it through computer control of steering (via a motor attached to the steering column) and the drive motor.

Position information is mainly derived from logging the number of drive wheel revolutions and the steering inputs, but this is subject to accumulative errors due to wheel slip and measurement errors. Eight ultrasonic transmitter/receivers are therefore distributed around the vehicle periphery to build up an image of the surrounding obstacles and measure their distance. This information coupled with the rough odometric position measure and the map, gives an accurate position fix which is then used to correct the odometer and update the map. Unexpected obstacles can also be identified and evasive action can be taken in the event of a near-collision. If a collision cannot be avoided, compressible tactile bumpers instigate an emergency stop before the main body of the truck comes into contact with the obstacle.

Problems with sonar guidance, such as interference from stray echoes has led to the addition of a camera to monitor objects moving across the field of view as the vehicle proceeds along its route. This is in its early stages, and processing is being simulated on a mainframe computer, although it is hoped to implement it on-board in another minicomputer. External communications is also at a primary level with a V.D.U. and keyboard currently placed on the truck.

To bring the prototype to a production stage, it is proposed to implement all the software in dedicated microprocessors controlling specific vehicle functions.

2.2.4 Conclusions.

In their present states, the two French projects bear little resemblance to a practical industrial truck, and even though Warwick's prototype is based on a commercial vehicle, its function as a load transporter is limited to its towing capability since its own pallet platform is fully loaded with computer and peripherals. (There is also no indication that the navigation system could cope with towing trailers). From these three projects, there is therefore no directly usable system for the Cableform goal.

The concepts are, however, applicable and it is instructive to see how features of the final designs have developed as a direct consequence of those concepts. The first two projects discussed here had a common interest in simplicity in control and data processing which has led to the use of combinations of sensors or the dedication of a particular sensor to a particular control function. It has also resulted however, in very specialized design of the vehicles such that their only function is self navigation and considerable re-design would be required to develop them into an industrial vehicle. Warwick's approach was fired by the desire for a free-roving, environment independent vehicle and resulted in very few sensors and very complex data processing requiring massive on-board intelligence. This has again led to a vehicle which is capable of finding its own route, but at present has no other abilities, and is also very costly.

These two approaches suggest a middle road based on an industrial vehicle with the capabilities of the final objective, and employing the minimum system approach, but with constant consideration of the goal to avoid the introduction of specialities which must later be re-designed as the project advances.

2.3 Sensors.

The purpose of sensors on an AGV is to gather requisite information to enable the various control systems to function correctly. This information may be about the vehicle itself, for instance motor current and speed, or about the environment - ambient temperature, distance and shape of a specified object, relative position of a light source. Since this work is concerned with positioning an AGV, a general review of sensors presently used on robots (both mobile and static) follows, together with a more detailed look at the transducers and data processing techniques which could be usefully employed in a positioning system.

2.3.1 A General Look at Robot Senses.

Robot sensors are basically employed to identify objects and to measure quantities - usually distance. This information is generally used passively, for example in quality control monitoring, or actively such as for safety, positioning or navigation. Some sensors can be used for more than one function and often only part of the sensor capability is utilized, therefore there is usually a choice between sensors for any application. Various classifications exist (13) and these are summarised in table 2.1.

Commonly used sensors can be divided into optical, acoustic, tactile and magnetic. The optical category encompasses the greatest range - both in terms of variety of transducer and function - and includes passive photocells, infra-red emitters and detectors, self-scanned photodiode arrays, vidicon cameras and lasers. Typical applications are: detection of light beacons for positioning with photocells; optical triangulation and position sensing from markers using I-R transceivers and retro-reflectors; high resolution range-finding or bar-code reading with lasers. Solid-state and video cameras are the most versatile, and also present the most complex data processing problems, but can be used for target recognition, measurement of object dimensions and determination of position and orientation (usually of small objects nearby).

Acoustic sensors are usually ultrasonic transmitter/receivers, used for range-finding or sonar imaging and for communications links. Tactile implies contact and any such sensors employed for measurement are reserved for robot arms to 'feel' their way towards a workpiece or measure its dimensions and orientation once picked up. Proximity sensing is an important function of tactile transducers and these are found on all industrial AGV's in the form of compressible bumpers which stop the

CRITERION	EXAMPLE
Physical property exploited	optical, magnetic, sonic
Contact / non-contact	obstacle detection by safety fender / ultrasonics
Passive / active	vision with ambient light / controlled illumination
Temporal / spatial	laser time of flight rangefinder / photodiode array
Dimensions of sensed targets	nuts and bolts for assembly versus car bodies for paint sprayers
Dimensions of image representation	linescan versus matrix cameras

Table 2.1 Classification Of Robot Sensors.

vehicle when contact is made with any object, thus preventing catastrophic collisions. Finally, magnetic sensors are extensively used on commercial AGV's where a current-carrying guide-wire is laid down from which the vehicle derives position information by measuring the strength of the sensed magnetic field (14,15). Stopping points have also been identified magnetically with solenoids or fixed magnetic studs (16) or by using separate guide-conductors for each location. In the latter case, all the conductors are energised by the same frequency current signal, but each has a pre-determined phase relationship with neighbouring conductors, so producing a 'homing' path (17).

Ultrasonic and magnetic sensors have been used in many applications and their success and shortcomings are well documented: the former are plagued by multiple reflections, effects of surface texture on reflected beam strength, and poor resolution due to the large angle of acceptance of the receivers (7,10,11). The major factor against using magnetic sensors for position information is the necessity to alter the environment to provide the sensed signal, and the problem of ensuring that all routes are free from extraneous conductors such as metal plates. The brightest field for providing accurate position data with little or no change to the environment, is optical sensors and 'computer vision'.

2.3.2 Visual Sensors and Image Processing.

Video and solid-state cameras offer very fast aquisition of data with control over resolution and field of view simply through choice of lenses. Processing of the vast quantities of data in a camera image is an enormous topic, but much of the established work is concerned with reducing image degradation during sampling, and image enhancement to aid photointerpreters (18). Image processing for robots - specifically mobile machines - is still in its infancy as the problems here are not how to exactly reconstruct an image from the video signal, but how to extract the required data (usually a very small proportion of that available) in a realistic time interval, using minimum data storage and processing power. The minimum system arises from constraints of cost and physical size of the sensor. 'Realistic' times depend on the application, but for visual servoing of robot arms, a maximum of 0.1 sec. has been suggested (19). (Vehicles will be expected to move at faster speeds and will therefore make greater demands on the system).

Most of the published work on vision systems for automation is concerned with providing sight for assembly robots, especially for checking or picking parts from a conveyor. Here, the general approach has been to severely constrain the environment to simplify processing. The most common constraint is to use strong illumination or back-lighting to give high contrast so that simple thresholding will yield a binary image (only two colours - black and white) (20,21). Others include limiting the image content to just the part in question on a plain background and ensuring that all parts are correctly orientated before inspection by the robot (21). The National Bureau of Standards claim to be the first to use a 'structured' light source to highlight range information (19). A plane of light is reflected as a bright line by the object viewed and the height of this line in the camera image is proportional to the object's distance from the robot. Surface projections on the object cause the line to be segmented in a pre-determined way, so the shape can also be detected. A similar method has recently been used at Oxford University to aid welding robots(22).

The above applications deal with small objects at close range. For mobile vehicles, the environment is more varied and more costly to constrain. Sensors must therefore be more complex, and as intelligence increases, costs rise and processing times lengthen. Fourier transforms have been used to characterize target patterns as a data compression technique, with limited success (23), and another unusual approach advocates basing the design of optical transducers on the human eye which exhibits some advantageous pre-processing properties (24).

Early interest in pattern recognition techniques for mobile robots (21,25,26) has waned due to slow speeds and inflexibility - the sensor can only recognize objects that it holds in its image store - and current research is tending towards simply detecting the presence of objects (6) and tracking their motion relative to the machine (11,27). There is also a move away from pure software processing necessitating minicomputers and mainframes, towards composite systems using fast parallel pre-processing by hardware interfaced between transducer and computer (28,29), or hardware implementation of the algorithms by custom logic arrays.

A report of a recent symposium on industrial robots (30) notes the lack of progress in commercial applications of cameras for robots (specifically static ones) and suggests that solid state cameras are too expensive and delicate for factory work. (This may be exemplified on mobile industrial vehicles). The alternative was seen as simpler vision systems based on discrete photodetectors and/or structured illumination. Mitsubishi Electric's welding aid (30) and Skinko Engineering's parts inspection unit (31) are examples of this approach.

Although computer vision has been heralded as a cure-all for current inadequacies in automation, it too has had its problems and AGV's remain blind. If visual sensors are to be used as a positioning aid, several basic problems must be overcome: efficient filtering is required to eradicate unneccesary data (uneven illumination must be avoided or catered for as it is a source of noise in edge detection algorithms and areas of shadow may be interpreted as surfaces or obstacles). Ambient lighting is a problem even with a controlled illumination source since stray light can saturate the photodetectors. On a commercial note, the major limitations are cost and speed of operation.

2.4 Control Strategies For Mobile Robots.

A wide variety of mobile robots exist, ranging from remote controlled inspection vehicles operating in hazardous environments, to the famous Stanford Research Institute robot (26) which was completely free-roving in an unknown (but limited) environment. Although control system complexities vary enormously, similarities exist in implementation both in task delegation and co-ordination within sub-systems, and in realization in computer software and hardware.

Larcombe (27) has devised a useful classification of mobile robot control and this can also be applied to the constituent sub-systems. Table 2.2 lists the classification and figure 2.1 describes a hypothetical automatic vehicle control system in these terms.

Once the system structure has been decided, processor hardware can be assigned. Crenshaw (32) gives three computer implementations of multiple systems: Federated systems comprise several computers, each dedicated to a specific task and communicating with each other through

CLASSIFICATION	DESCRIPTION
Remote Control	Operator's controls in 1:1 correspondence with sub-system actions. Feedback via operator's sensors.
Servo Control	External guidance and task instruction. Automatic task execution by sub-system. Feedback: sensor to servo.
Semi-autonomous Computer Control	Operator divides task into set of sub-tasks. Sub-system executes sub-tasks as instructed. Feedback: sensor to computer.
Autonomous Computer Control	Operator provides high-level task. Sub-system has a computer and forms its own itinerary of sub-tasks. Feedback: sensor to sub-system computer.

Table 2.2 Classification Of Control Systems.

CLASSIFICATION	DESCRIPTION
Remote Control	Operator's controls in 1:1 correspondence with sub-system actions. Feedback via operator's sensors.
Servo Control	External guidance and task instruction. Automatic task execution by sub-system. Feedback: sensor to servo.
Semi-autonomous Computer Control	Operator divides task into set of sub-tasks. Sub-system executes sub-tasks as instructed. Feedback: sensor to computer.
Autonomous Computer Control	Operator provides high-level task. Sub-system has a computer and forms its own itinerary of sub-tasks. Feedback: sensor to sub-system computer.

Table 2.2 Classification Of Control Systems.

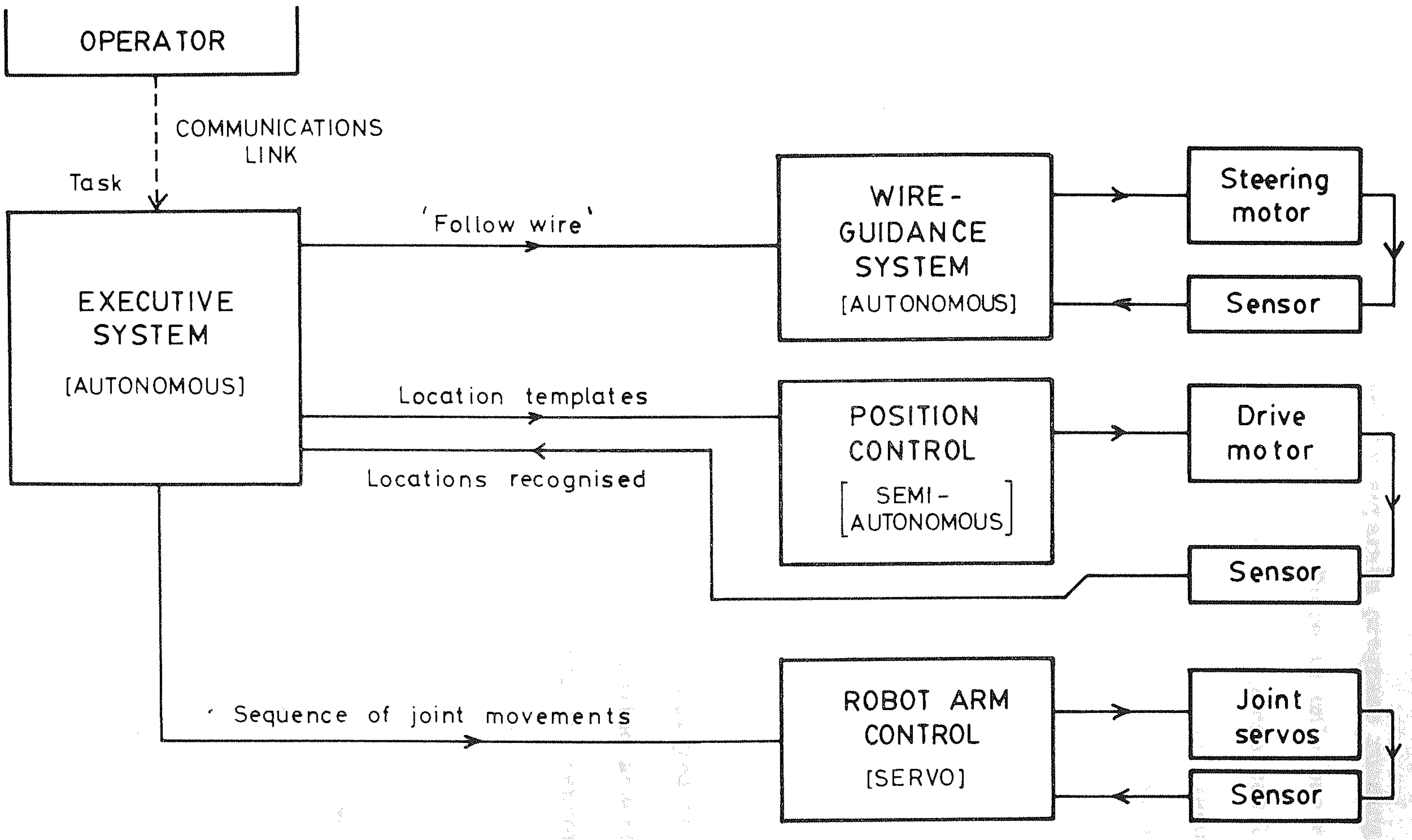


Figure 2.1 A Mobile Robot Control System.

input/output devices. This allows redundant systems to be included as a back-up in the event of processor failure. Each computer has its own executive program and job queue. An Integrated system is one in which unrelated tasks are performed by separate programs running on one or more processors sharing common memory. There is only one executive program and one job queue. Finally, several programs can be run simultaneously on one computer by interleaving their execution. This is called a Multiprogrammed System and is implemented by timed or externally controlled interrupts. The choice of system depends upon the particular problem and requirements.

In general, developers of mobile automatic vehicles have used federated systems (5,8) based on microprocessors for compactness, flexibility and cost considerations. The capabilities of such systems can match that of a minicomputer as illustrated at the Swiss Federal Institute of Technology where ten Z-80 microprocessors are employed to control the ten articulations of two robot arms in an automated assembly system. An eleventh processor is used as a task supervisor, sensor controller, and to communicate with the operator's console (33).

2.5 Vehicle Positioning Systems.

Sections 2.3 and 2.4 have illustrated the diversity of methods available for implementing a position control system on a vehicle. This section presents a resumé of systems designed to aid vehicle navigation and positioning specifically for industrial AGV's.

Commercial automatic vehicles are generally either wire-guided or run on rails on the floor and/or overhead. In the latter case, there is no navigation function and in the former, this is achieved via magnetic sensors as previously described in section 2.3.1. Fine positioning is achieved in a number of ways: physical buffers which the vehicle senses with its safety fenders; magnetic studs fixed at the workstation (16); retroreflectors fixed to stopping points and detected by a light source and photodetector on the vehicle (34); ultrasonic transmitter/receivers detecting the presence or absence of loads on pallet stacks (35). These are all simple markers which do not convey any information about absolute position. A coarse measure of distance travelled along the guide-route is generally made by counting wheel revolutions and this data is corrected as the markers are sensed. Absolute position detection has been employed by EMI through a system of unique guide conductors (17) and by Dexion Ltd. who exhibited a location marker system at a recent exhibition(36). This was for an automatic rail guided narrow-aisle side-loading vehicle: the markers consisted of combinations of retro-reflective strips on a non-reflective background arranged to form binary numbers when read vertically by an infra-red transmitter/receiver array on the vehicle. They were mounted on the uprights of the racking and were used to mark every horizontal location at ground level. Sensor to marker distance was a few centimetres.

Projects in the literature include a laser-based triangulation system for a wire-guided vehicle with raisable platform to enable it to leave the wire and guide itself to loads placed on special stands (37). Previously, spurs of the guide-wire had been used to guide the vehicle up to the stand, which required the loads to be accurately positioned to facilitate correct removal. The system developed entailed three reflective strips attached to each load which were scanned by laser beam and detected by a scanning photodetector. The angles at which the three reflections were received were then used to calculate the range and direction of the load. This gave acceptable guidance over distances up to eight metres.

Another method using markers has been employed by Nakamura and Fukui (38) to supplement odometric measurements for navigation. The markers consisted of illuminated slits arranged in doublets and triplets to form a code word, and were viewed by a camera on the vehicle. The special marker format facilitated fast image processing and recognition, and triangulation methods gave positional accuracies of ±5mm, but required many markers. Problems were encountered with focussing when trying to cope with range variations over eight metres.

The Warwick project covered in section 2.2.3 is attempting to overcome the need for markers and guides by using ultrasonics and vision combined with an on-board map for navigation and position. Norton-Wayne of City University also recognizes the need for trackless systems and has developed an algorithm for detecting road boundaries using camera images (39). The chosen application is automatic navigation for cars in streets and the process has been simulated on a PDP 11-10 computer by using photographs as the sensor input. This is not therefore, real-time processing and not directly applicable to mobile robots, but the results are promising and illustrate the ultimate approach - tailor the processing techniques to cater for the environment and not vice versa.

Chapter 3

THE AGV RESEARCH PROGRAMME

3.1 Introduction.

A position system for an automatic vehicle cannot be designed and implemented in total isolation from the other control systems on the vehicle. It is necessary to know the capabilities of the vehicle to determine what form the control should take; resolution and accuracy of sensors; type of sensors, and to specify an interface with other operations.

At the outset of this investigation, Cableform Ltd. had no practical knowledge of automatic vehicles or control systems, so there was a seemingly free choice for the implementation of each system. However since they all had to be brought together to operate the one vehicle, great care had to be taken to ensure that a decision taken on one system was viewed in the light of its effect on all the others. To this end the author and the other members of the Research team investigated the concept of an AGV and broke it down into a set of basic control functions and their interfaces. These functions were then delegated to individual members of the team for development and a programme of prototypes was devised to lay down the course of research for each function and bring them together at each major stage to operate a demonstration vehicle system.

This chapter describes this investigation and sets out the development programme. It then goes on to outline the areas of the programme that were the responsibility of the author and the concern of this thesis.

3.2 The goal - the intellingent AGV.

The company's aim was to produce a machine which could be given high level fetch and carry instructions and be left to its task without further external inputs. An example of such an instruction would be "load A at B and take to C", and the desired response would be for the vehicle to find a quick, safe route to location B; stop to pick up load A: then continue to location C where it would stop, signal the successful completion of the task and await further instructions.

3.3 Fundamental Functions of an AGV.

"Load A at B" implies motion of the vehicle from its present location to point B and hence finding a suitable route between the two. The vehicle must therefore know where it is and the relative position of B. However, unless the operating environment is an empty space, there will be areas which the vehicle may not or cannot physically occupy, so it does not have a free choice of route. It therefore requires information from the environment to enable it to keep within the route areas and identify task locations, but does not necessarily require the absolute position of every point it passes through.

Once the vehicle reaches B, it must stop to pick up load A. The load manoeuvre is dependent on the vehicle being stationary at B, so not only must it recognize the location, it must have advanced warning of approaching B so that it can slow down and stop as required. Once there, the load mechanism must be aware of the relative position of A so that the load transfer can be accomplished.

The second part of the instruction: "take to C" implies that the vehicle move off once loaded, hence it must be aware that the loading operation has been completed. This is an internal position measurement of the load on the vehicle. Stopping at C implies a similar set of operations as for finding B, and then the vehicle must be capable of communicating completion of the task to its external control system - be it man or a computer - and receiving instructions for the next task.

It must also remain where it is until the next instruction is given.

From the above discussion of a typical instruction, several requisite features begin to appear. For example, the vehicle must be able to control its own motion and the operation of its loading mechanism. Information must be collected from the environment to monitor progress in a task and to determine the actions to be taken. The vehicle must be aware of what areas it may enter and be able to identify them.

A more comprehensive breakdown of the sample task will highlight these features more clearly and also give an idea of the degree of interpretation required of the vehicle to translate the high level instruction into low level control signals to the various actuators available to it (such as the drive motor).

Assuming 1 is the highest level of instruction, this breakdown might be as in figure 3.1: the second level is an itinerary of tasks in chronological order and level three divides the tasks into different actions such as moving the vehicle and sensing the environment.

From this treatment, six fundamental functions can be identified:

Traction - control of speed and acceleration of the vehicle.

Route Following - knowledge of available routes; detection of their position on the ground, and control of the vehicle steering.

1st LEVEL	2nd LEVEL	3rd LEVEL
Load A at B and take to C	a) Go to B	a) i Move forward along route at cruise speed ii Look for advanced warning of B
	b) Stop at B	b) i When B recognized, slow down ii Look for stopping point iii Stop at target
	c) Check for A	c) Look for presence of load A
	d) Load A	d) i Start loading mechanism ii Watch for load in correct position iii Stop loading mechanism
	e) Go to C	e) Repeat a) but looking for C
	f) Stop at C	f) Repeat b) but looking for C
	g) Check C is empty	g) Check C can receive a load
	h) Unload A	h) i Start loading mechanism ii Wait for load correctly positioned at C iii Stop loading mechanism iv Signal task completed

Figure 3.1 Command Interpretation For An AGV.

Position Detection - recognition of vicinity of task locations and measurement of vehicle distance from desired stopping points.

Load Handling - detection of position and nature of load, and control of speed and acceleration of load mechanism.

Communications - ability to receive task instructions from an external source and to transmit status information and requests for further instructions.

Executive - interpretation of external commands into sub-tasks allocated to specific functions and the co-ordination of these functions to achieve the desired action.

The above sub-divisions formed the basis of the allocation of projects to members of the research team involved in AGV development. The author was given sole responsibility for Position Detection and Traction.

3.4 The Development Programme.

The development programme in figure 3.2 was devised to provide a series of stepping stones from the fundamental principles outlined in section 3.3 to the end goal. Each stage was designed to develop from the previous one either by additional hardware as control problems were solved, or improving the vehicle's capabilities by consolidating data processing knowledge and adding more complex sensors.

	PROTOTYPE	TRACTION	ROUTE FOLLOWING	POSITION DETECTION	LOAD HANDLING
1	SINGLE AGV EXTERNAL COMMANDS AUTOMATIC LOAD/UNLOAD PALLET TRUCK	SEPARATELY EXCITED D.C. MOTOR CONTROL	WIRE GUIDANCE WITH SOME SOFTWARE RESIDENT MANOEUVRES	LOCATION MARKERS OPTICAL SENSOR DISCRETE LED ARRAY	POWERED ROLLER CONVEYORS SIMPLE MOTOR CONTROL
2	MULTIPLE AGV's CENTRAL COMPUTER AUTOMATIC LOAD/UNLOAD MOVING PLATFORM: UP/DOWN	OPTIMISED D.C. MOTOR CONTROL	OPTICAL LINE FOLLOWING	LOCATION MARKERS COMMERCIAL LED ARRAY-BASED CAMERA	I-D LOAD POSITION SENSOR RAISE/LOWER MECHANISM FOR PLATFORM
3	MUTIPLE AGV SYSTEM CENTRAL CONTROL AUTOMATIC LOAD/UNLOAD PLATFORM MOVEMENT IN THREE PLANES	A.C. MOTOR CONTROL	DISCONTINUOUS ROUTE MARKINGS COMPLEX SOFTWARE RESIDENT MANOEURVRES	CAMERA AND SIGNPOSTS WITH TARGET RECOGNITION	3-D HIGH RESOLUTION POSITION SENSOR
4	POSITION DETERMINATION USING A LEARNING SYSTEM	OPTIMISED A.C. MOTOR CONTROL	CAMERA PROVIDING ALL VEHICLE POSITIONAL INFORMATION SOFTWARE CONTROLLED GUIDANCE WITH SOME ENVIRONMENTAL SIGNPOSTS		VISUAL AND TACTILE EDGE SENSING
5	ROBOT ARM ON VEHICLE DEALING WITH FIXED LOAD SHAPES	"	VIDEO PATTERN RECOGNITION TO REPLACE ENVIRONMENTAL SIGNPOSTS		3-D SENSOR FOR ROBOTIC ARM COMPLEX CONTROL OF ARM
6	ROBOT ARM DEALING WITH VARIABLE LOAD SHAPES-ORDER PICKER	"	"		VIDEO PATTERN RECOGNITION FOR ORDER PICKING

Figure 3.2 Cableform AGV Development Programme. (November 1981).

Also, the company required a prototype system to illustrate the feasibility of the project and for use in demonstrations to attract funding and potential custom. This meant that each stage had to have product potential and show a significant increase in sophistication from the previous one. The complete programme had a ten year time-scale, and the first three years covering the design of the programme and development up to prototype 1 are encompassed by this thesis.

3.5 The First Prototype System.

3.5.1 The Vehicle.

In April 1980, the company purchased a prototype Electruk tow-tractor from A.W.D. (figure 3.3) to act as a testing ground for the initial stages of the various systems being developed. The company's original idea was to produce a package which would convert a manually operated vehicle into an automatic one, so a typical commercial truck was chosen for adaptation rather than a specially designed chassis, to gain experience in the problems of "real" vehicles and also the connotations of retro-fitting.

The tug had a nominal payload of 1,000kg and a carry and tow capacity of 2,000kg on the level. Two CAV 0.41kW motors with reduction gearboxes and chains to the rear wheels formed the drive-train and operated from a 24V battery supply. An Ackermann steering arrangement was fitted to the front wheels.

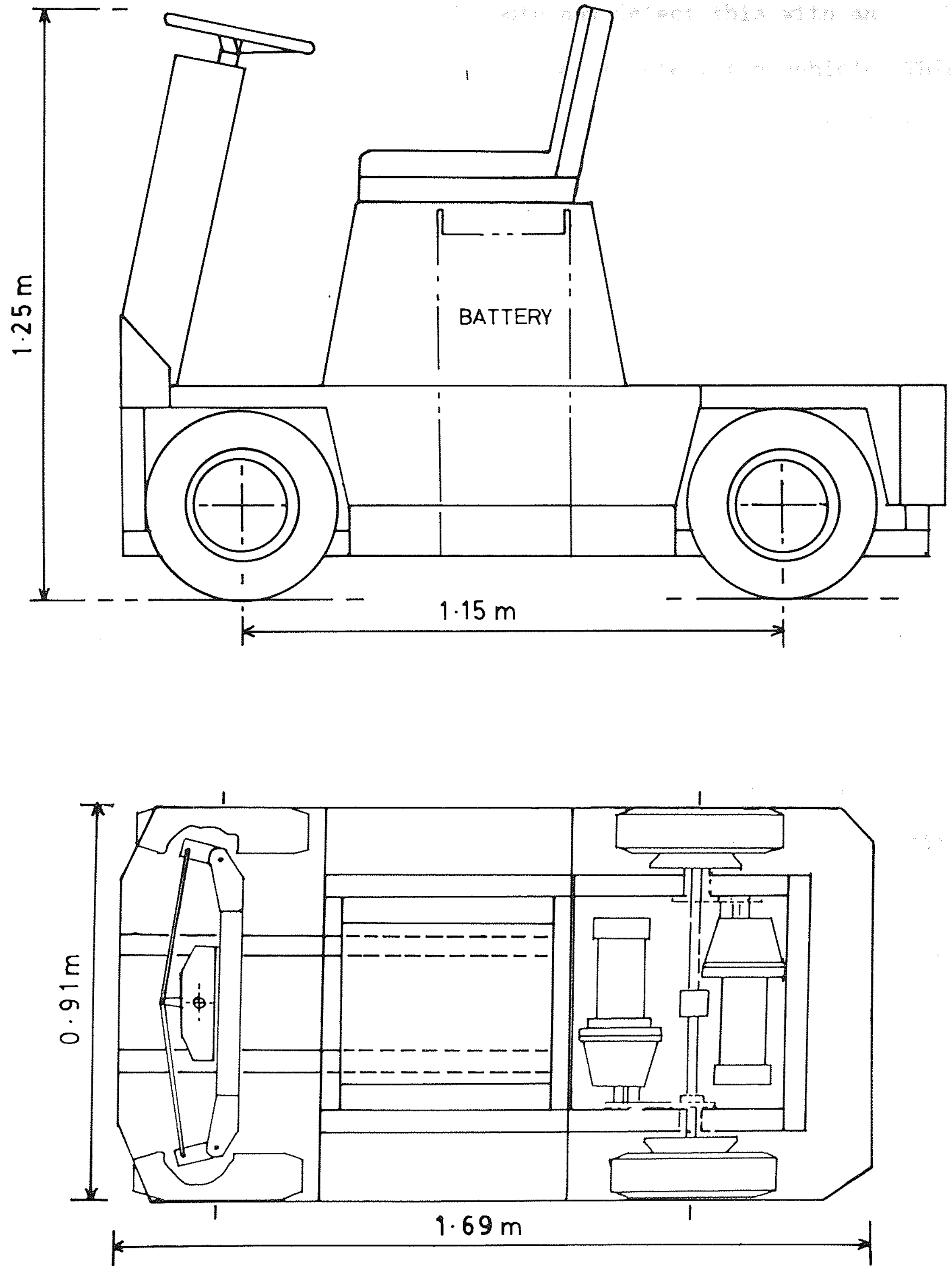


Figure 3.3 A.W.D. Electruk.

3.5.2 Functions.

The review of work in the field of autonomous mobile machines and commercial AGVs suggested that the quickest way to achieve a working system was to lay down a physical route and detect this with an on-board sensor whose output would be used to steer the vehicle. This immediately provided localized position measurement and control in a direction transverse to the guide-path, and since this was dealt with by the Route Following function, the Position function was left with the much reduced problem of providing position measurement along the route only.

A current carrying conductor was chosen for the guide-route as commonly found in commercial AGVs, with steering control acting purely on the Ackermann system. Independent drive of the rear wheels was not required, so the two traction motors were driven in parallel from one control system.

The problems of load handling control were envisaged as being fundamentally similar to vehicle control and would therefore benefit from being tackled after the initial vehicle problems had been solved to save duplicating work. For the first prototype, loading was therefore included primarily for demonstration purposes and consisted of a short roller conveyor on the vehicle which could be activated on arrival at a location and transfer unit loads between fixed conveyors at work-stations. This involved some structural alteration to the tug, but left the basic chassis unchanged (figure 3.4).

3.5.3 Implementation.

Figure 3.5 shows a block diagram of the first prototype with the information flow between the various functions. Six microprocessors

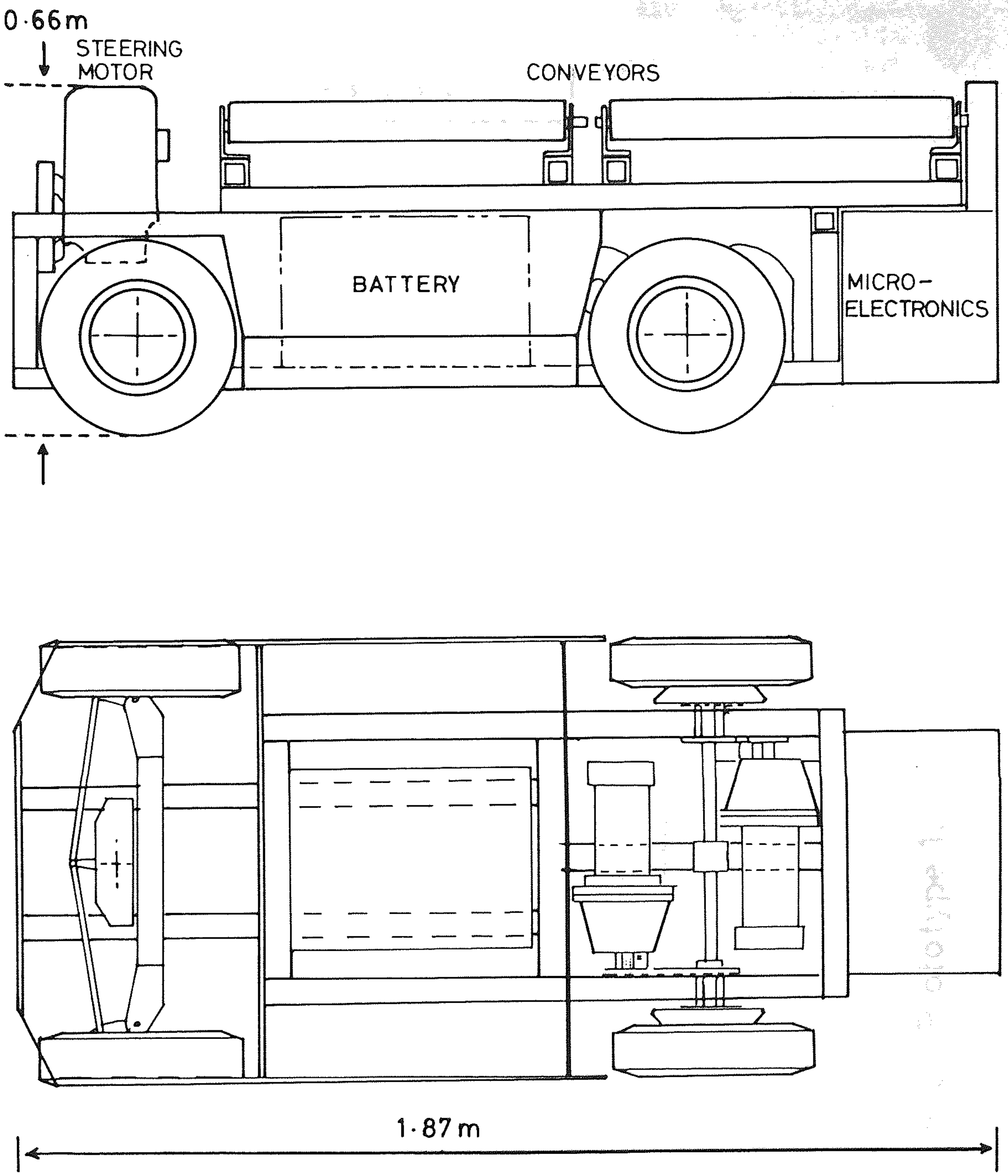


Figure 3.4 The First Prototype.

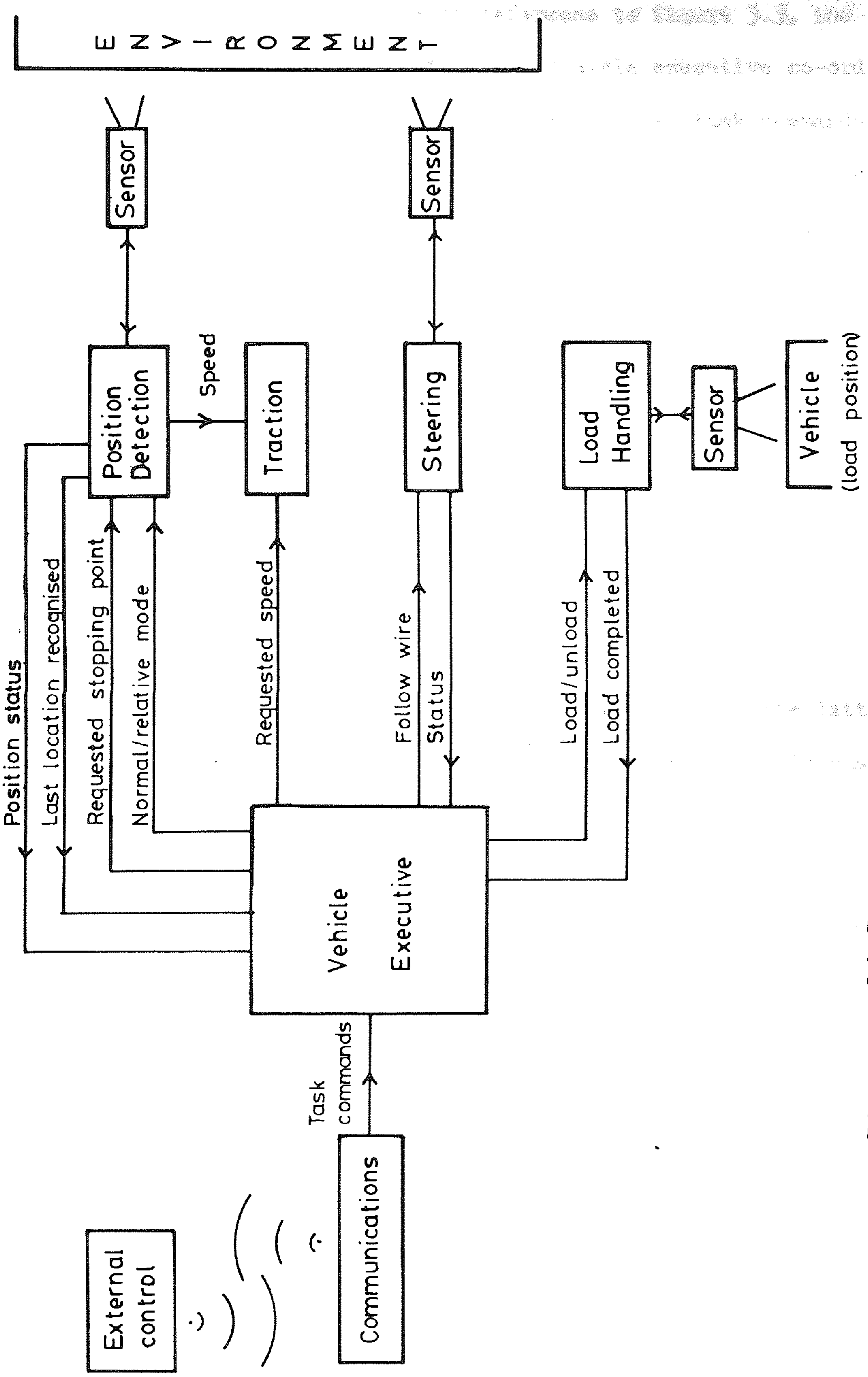


Figure 3.5 Block Diagram Of Prototype 1.

were used, one assigned to each function, so that as far as possible, members of the research team could develop their respective systems without hindrance from others. With reference to figure 3.5, the duties of each processor are as follows: Vehicle executive co-ordinates the sub-system processors and translates high-level task commands into low-level function specific commands. These tasks were to be input externally, but were initially programmed in PROM for tests whilst the communications hardware was being developed. Inter-processor communications were only allowed from sub-system to executive and vice-versa (except from Position to Traction during fine positioning of the vehicle at a task location).

Steering implements a closed-loop control system and operates autonomously, simply keeping Vehicle Executive informed of its current status. In the event of a status change - for example losing the signal from the guide-wire - Steering informs the Executive and the latter instructs the sub-systems as to the required action - in this case to stop the vehicle.

The Traction processor is responsible for controlling vehicle speed as demanded by Executive in normal operation or by Position during a stop manoeuvre. There is no return flow of information.

Position deals with both target recognition and relative position measurement as instructed by the Executive. In the former case, the desired destination is given by Executive so that Position can monitor for this location and enter the relative mode at the required time without further instruction. Information given to the Executive consists of the last location or landmark recognized and the current mode of operation. In relative mode Position takes external control of the

Traction processor so that it can implement a position control system in the region of a stopping point.

Load handling in this initial stage is basically an on/off plus direction control of the conveyor drive motor. These instructions are handed over from Executive once the vehicle is positioned. Return data is simply a task completed signal.

The Communications processor deals with the transmission and reception of external commands and transfers these commands directly from and to Vehicle Executive without any further interpretation.

3.6 The Position Sub-Project.

As stated in section 3.3, the author's work involved the design of the Position and Traction functions, to facilitate vehicle navigation and positioning. To recap, the Position Detection function is responsible for recognizing task locations as they are approached and measuring vehicle distance from stopping points within a very local area. The specification for this project was therefore to develop a method of uniquely identifying task locations and stopping points together with a means of sensing and recognizing such identifiers. In addition, high resolution measurement of distance was required in certain specified regions. A traction control system was required with a means of deriving input data from the position detection unit to enable the vehicle to carry out its tasks.

The decision to equip the first prototype with a wire guidance system reduced the position control part of the function to one dimension only - along the wire, and this implementation will now be described in more detail together with its interface with the other projects of the AGV programme.

3.7 The Longitudinal Vehicle Position System.

3.7.1 Sensor.

The original aim was to derive information directly from the working environment without any introduced landmarks or guide-posts. In the goal system, this information would be used both for guidance and fine-positioning of the vehicle. However, work in this field is still at a very fundamental level and the associated processing times are already too lengthy to be viable on a commercial vehicle. The twofold nature of the project (measurement and control) and the need for a working AGV in the short-term meant that the sensor could not be developed in isolation from the control problems, so the basic concepts of the goal vehicle's sensors were outlined and a simplified version designed and implemented on the first prototype.

Finding a route through a store or assembly line and avoiding obstructions at speeds of $1ms^{-1}$ or greater, requires a vast amount of data from the area round the vehicle in the direction of travel and for some distance ahead. For sheer quantity, data rate and resolution, an optical sensor based on a television camera is the obvious choice with perhaps some ultrasonic or laser range-finding device to recover a 3-D image. With this in mind, it was decided that the longitudinal position sensor should be optical and should incorporate the rudimentary features of sampling and processing to be found in a camera. It was also appreciated that a major problem with computer vision is the very quantity of data in one image so the system was designed to operate at minimum data rates and involve the minimum of signal processing required to achieve a reliable and adequate output for controlling the vehicle.

A system of markers was used to identify task locations as this considerably accelerated the realization of a working prototype. The sensor was then used to recognize these markers and read the information encoded on them. Control of the transducer was by a microprocessor which also performed all the signal processing and interfacing. A diagram of the sensor architecture is shown in figure 3.6.

3.7.2 Control.

For demonstrations, a one-way system was operated, so the navigation control consisted of taking the decision at each location marker of whether to continue along the wire, change guide-wires, or stop. This decision was based on information from the position sensor and the task commands, and was taken in the Executive, which then took action through Traction, Position and Route Following. Vehicle position was controlled as follows: the Traction processor received a numerical input which it converted into a pulse-width modulated signal for controlling current flow through the motors via transistor switches. Vehicle Executive normally provided the input according to the task, but during fine positioning at a stopping point, Position used its sensor information to produce the Traction demand. The initial system used simple speed control with set speeds specified for different tasks. Figure 3.7 shows the author's area of responsibility and the interface with other vehicle systems.

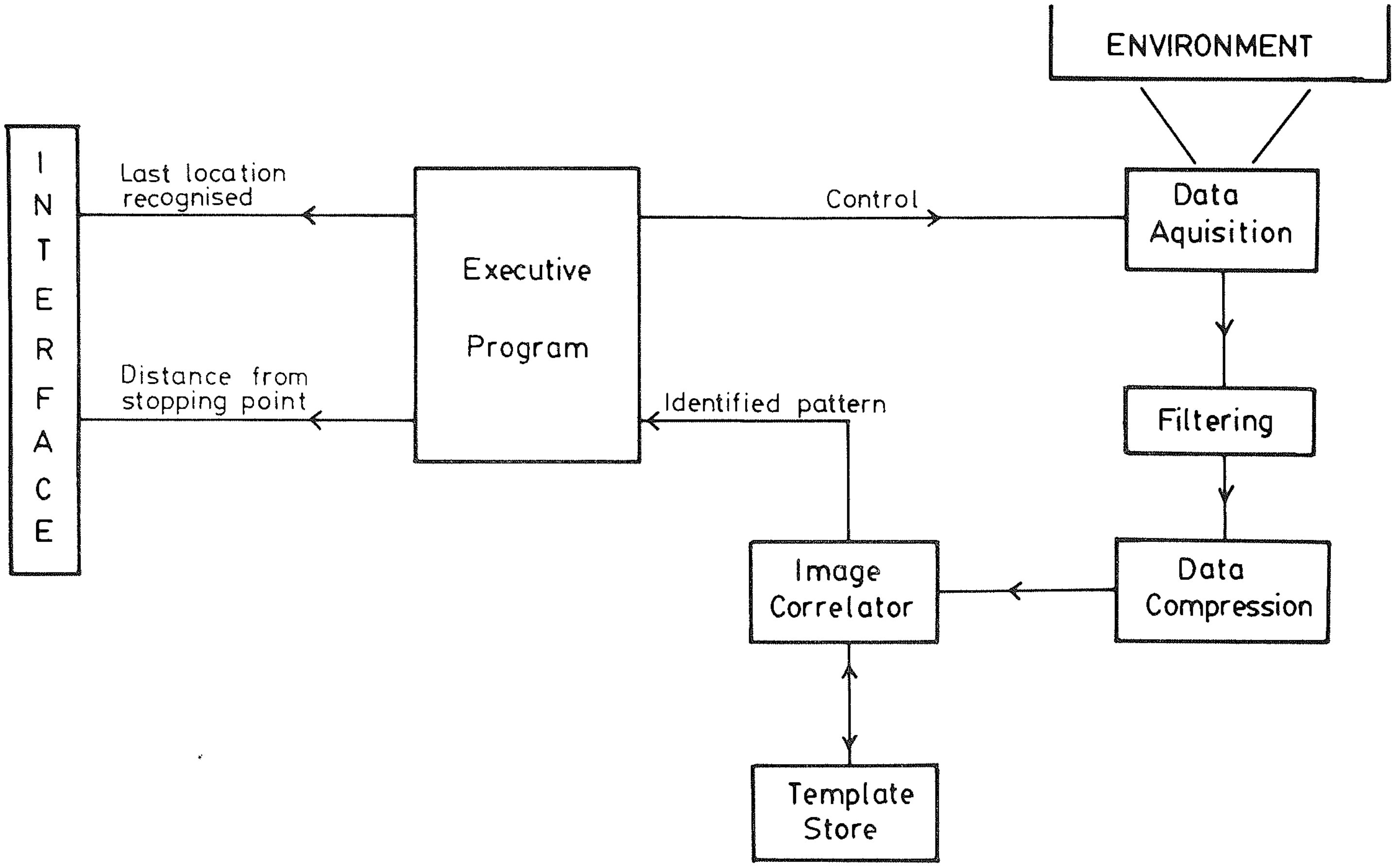


Figure 3.6 Position Sensor Architecture.

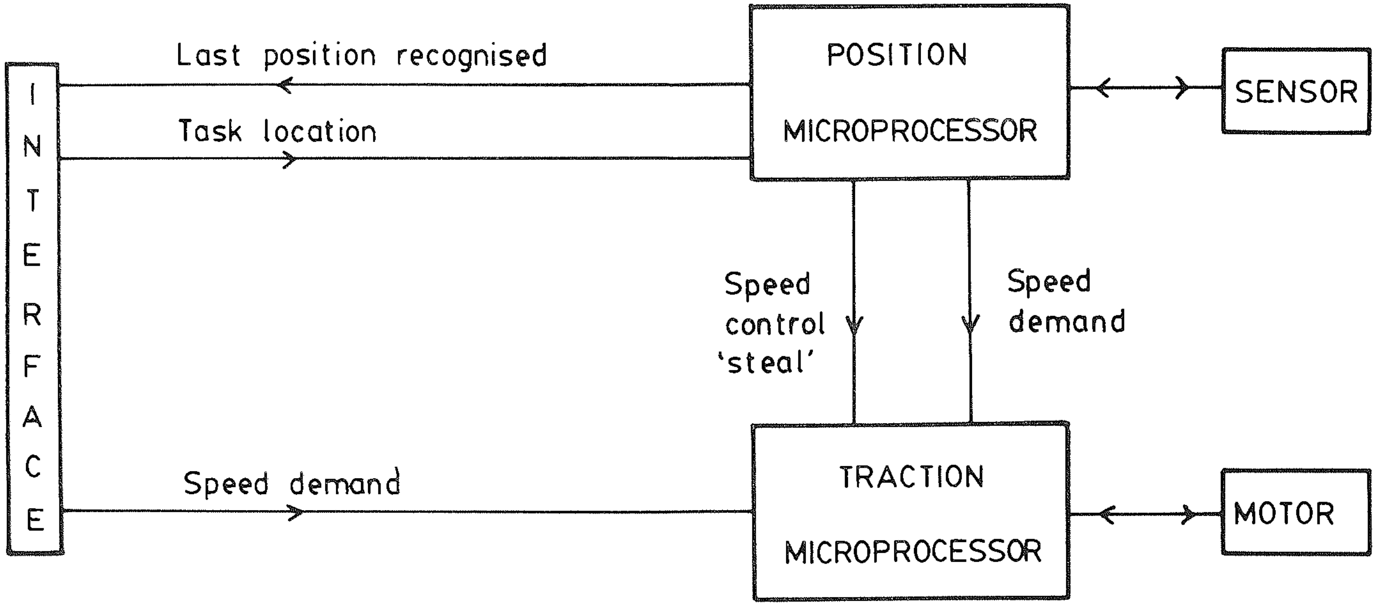


Figure 3.7 Author's Responsibility.

3.8 Summary.

The philosophy of approach to AGV research at Cableform Ltd. has been evolved from consideration of the final goal and expressed in terms of a Development Programme. The first prototype has been outlined in general and against this background, the author's areas of prime research have been described. The design and development of this particular sub-project is considered in detail in the following chapters.

Chapter 4

THE MARKER SYSTEM

The role of the position sensor was to recognize task locations and stopping points. In the latter case, the sensor also had to provide a measurement of vehicle distance from the desired stopping point. The Development Programme specified the first prototype to have an optical sensor with a camera-like image format, which was to gather its information from markers placed in the environment. This chapter describes how sensor constraints and the required information content influenced marker design and how the marker system was used in the execution of vehicle tasks.

4.1 Sensor Considerations.

The sensor consisted of a two-dimensional array of photodetectors viewing an area in a plane parallel to one side of the vehicle, which was illuminated with infra-red radiation from an on-board source. Processing of the sensor image was to be simplified by contrast thresholding to produce a binary picture, hence the markers were fabricated from elements that were either infra-red absorbing (hereafter termed black for convenience), or infra-red scattering (white) to give high contrast for the thresholding process.

4.2 Information Content.

The demonstration environment for the prototype consisted of a single wire loop and two workstations, each having a unique number for identification. As discussed in section 3.3, this number marker had to be read by the vehicle before reaching the desired stopping point to allow time to slow down. A second marker was therefore used to identify the position reference as in figure 4.1. For fine positioning of the vehicle, this marker incorporated a scale of distance either side of the reference which was to be used as the input to the position control system.

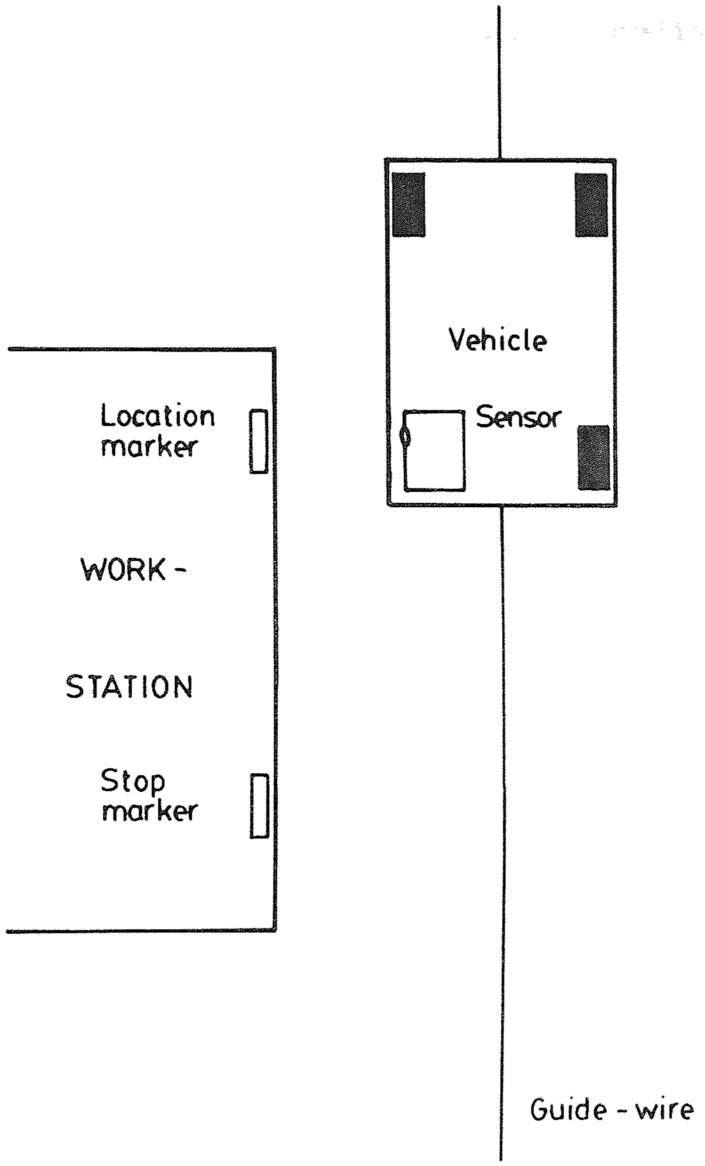


Figure 4.1 The Two Marker Types In Use.

Two distinct types of marker were therefore used: one identified the location (location marker), requiring a unique marker for each workstation, and the other marked the point at which the vehicle must stop (stop marker). The latter carried the same information regardless of location, so the same stop marker was used at each workstation. The problem was how to encode location data and distance scales in the same format so that both types of marker could be read efficiently by one sensor.

4.3 Data Presentation.

The first consideration was the necessity for the sensor to be able to read the stop marker when the vehicle was nearly stationary. Mechanical scanning of the sensor was avoided due to cost and the harsh environment associated with industrial trucks. This meant that each detector in the sensor should view an area no larger than one marker element.

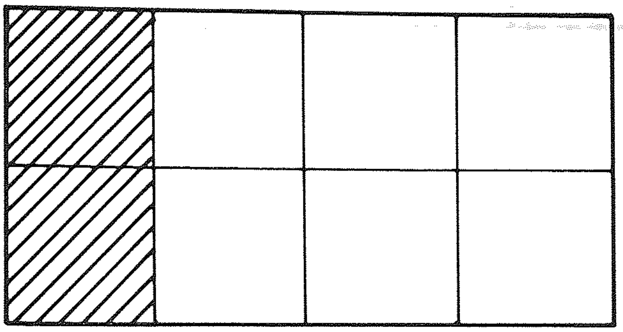
To exploit the fastest instructions of the Z-80 microprocessor, an eight-bit representation of the marker image was desirable. Location markers could be easily accommodated within this format since the Prototype system only needed a few locations for demonstration purposes. At first sight however, the limitation appeared too severe for encoding distance scales. Image processing techniques suggested using the edge between a black and a white area within the marker to represent the position at which the vehicle must stop. Once the sensor had a stop marker in view, the position of this edge within its image could be measured relative to say, the middle of the picture, so giving a measure of vehicle position error. This approach required a two-dimensional image, and a 2 x 4 element format was investigated which gave the widest marker and hence greatest measure of position offset using eight bits.

Stop and location markers had to be easily distinguishable since they required different data processing algorithms to retrieve the data. This was achieved by reserving two elements at either end of the marker for a special identifying pattern, which also helped to break up the image and distinguish it from natural features. Examples of patterns that could be expected to occur in the environment are given in figure 4.2, and figure 4.3 shows the two marker formats chosen. The last constraint was that each partial view of the stop marker (as seen when the vehicle was approaching its final position) was unique and could not be misinterpreted as a location marker. This was achieved by extending the stop marker as in figure 4.4, which also shows how the various views are related to vehicle position error, thus providing the distance scales.

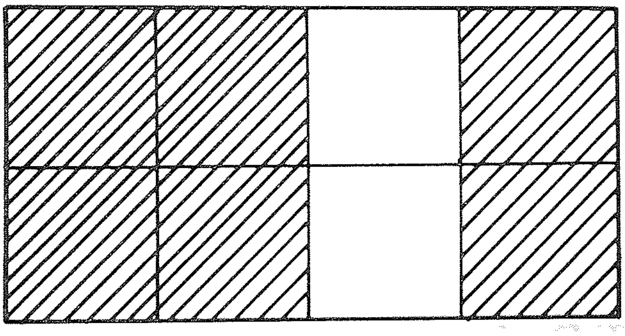
4.4 Implementation.

The most readily available material with the required infra-red scattering properties proved to be matt black card and coarse-grained white paper. Each element was 20mm wide by 40mm high, making the markers 80mm square in total. The vertical elongation was designed to allow for slight variations in sensor height relative to the markers due to vehicle loading: the two rows of photodetectors in the sensor are separated by a gap and view two horizontal strips of the marker. Figure 4.5 shows the imaged area for correct height alignment and illustrates how small variations in height do not affect the image content.

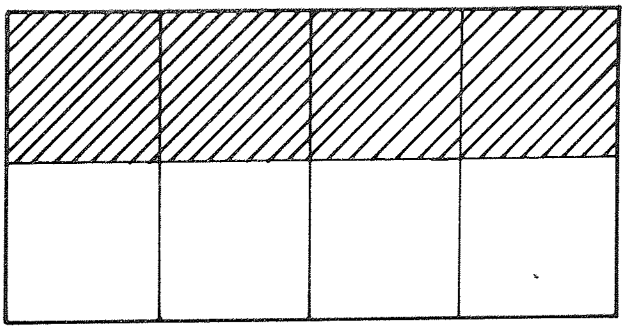
In the test environment, location markers were used for workstations and also as landmarks along the guide route. Stop markers were placed in conjunction with location markers at workstations and used by the



a) Corner of a wall

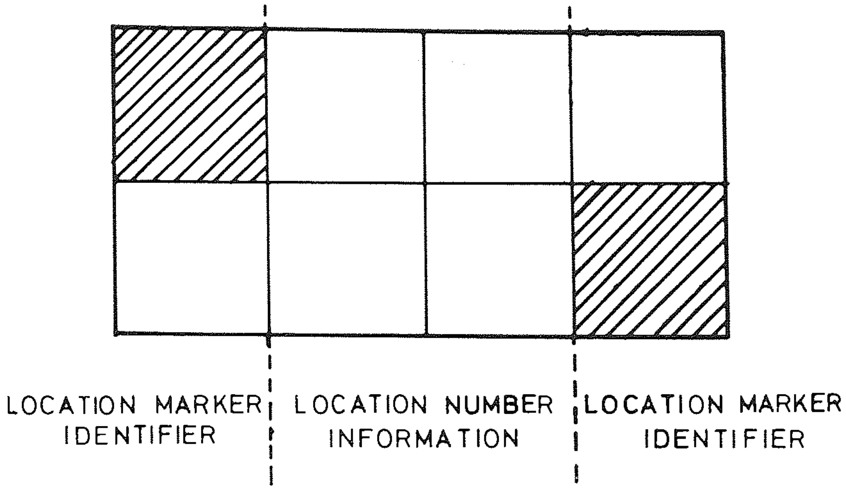


b) Upright of racking

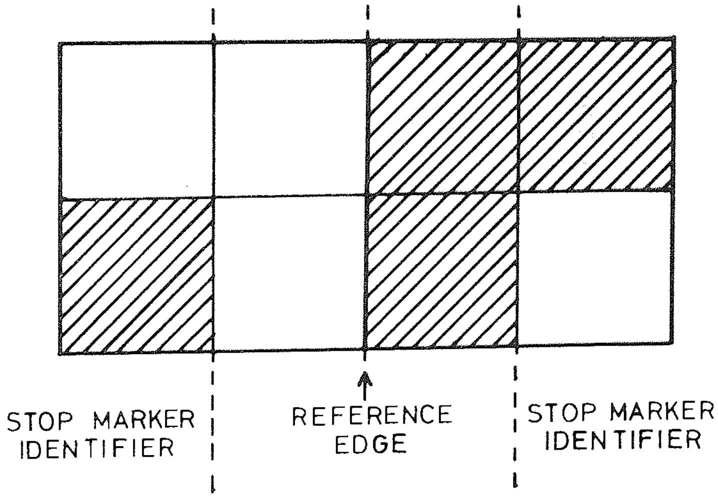


c) Top of pallet

Figure 4.2 Examples Of Naturally Occuring Patterns.

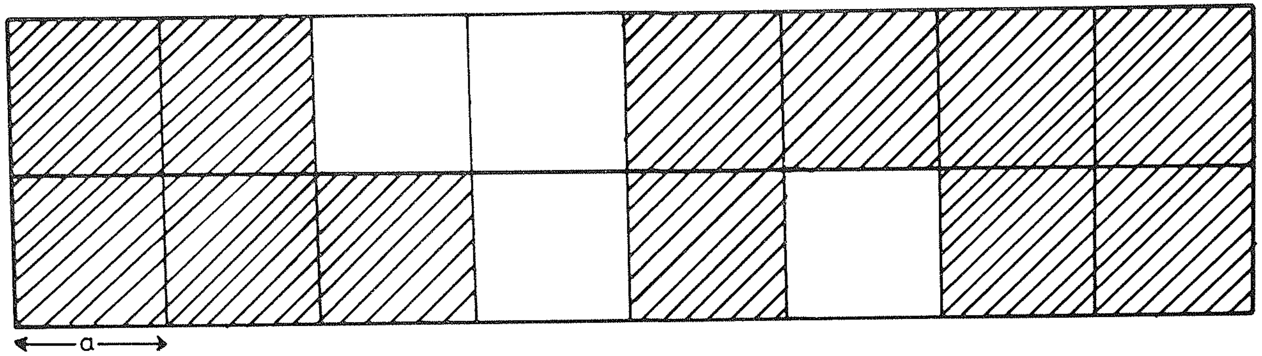


a) Location Marker

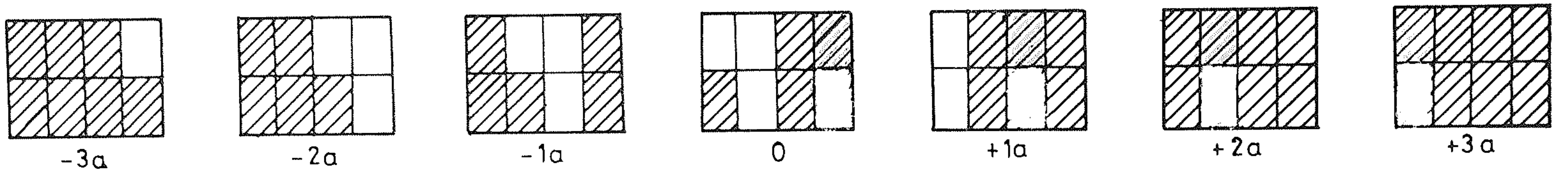


b) Stop Marker

Figure 4.3 Data Presention In The Two Marker Types.

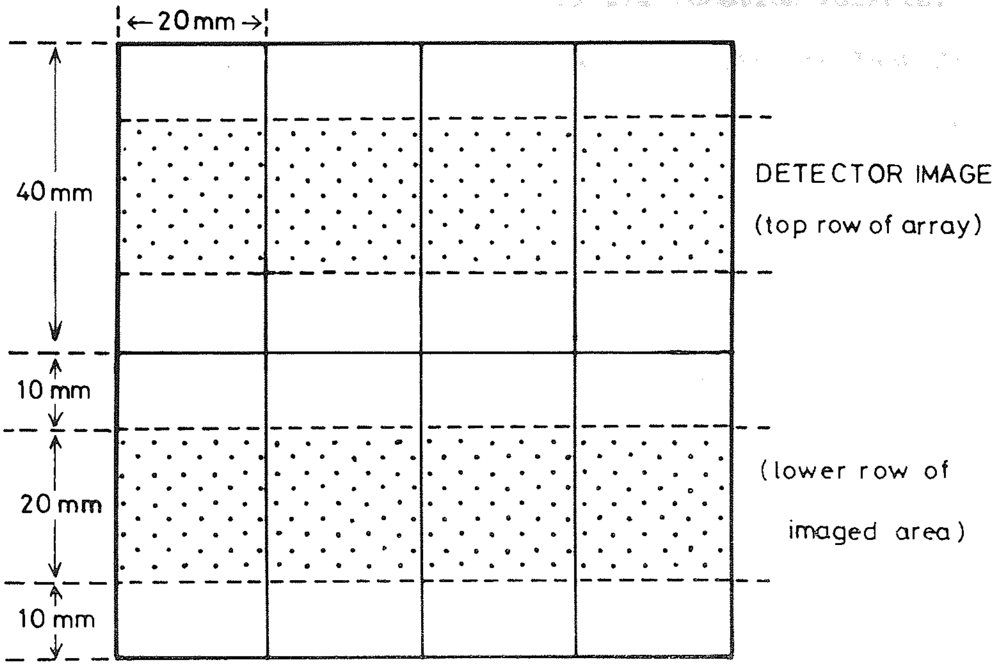


a) Stop Marker

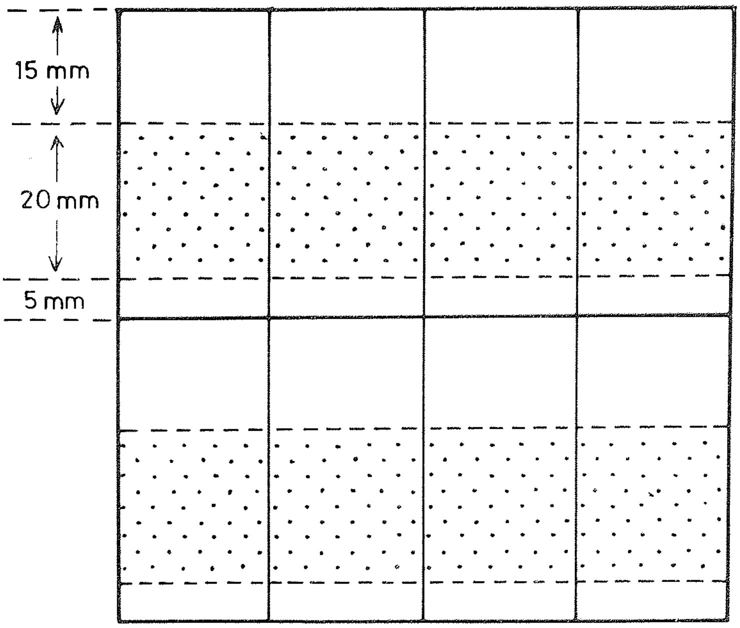


b) Vehicle Position Error For Each View

Figure 4.4 Sensor Images As Vehicle Passes A Stop Marker.



a) Sensor and marker correctly aligned

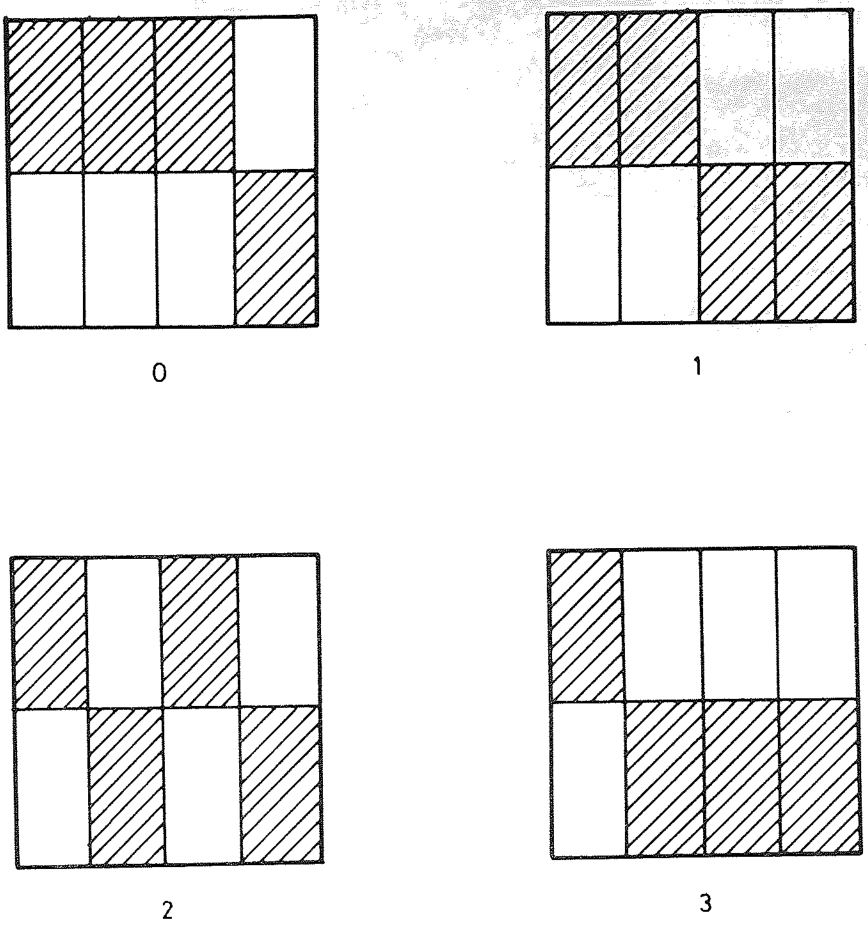


b) Sensor vertically displaced by 5mm

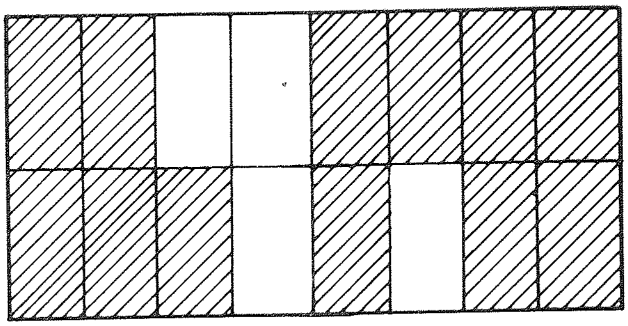
Figure 4.5 Imaged Area Of Marker.

vehicle in executing tasks as follows: on reception of a command to stop at location 3, the vehicle proceeded along the guide-wire (traffic was only permitted in one direction), looking for location markers. When location 3 was recognized, the sensor was instructed to look for a stop marker and the vehicle slowed to creep speed. As the stop marker came into view, further slowing was instigated and the vehicle was brought to a halt when the positioning edge was central in the sensor image.

Only four location markers were required, thus two of the possible four bits of location information sufficed. Data verification was achieved by repeating the pattern of the top row in inverted form on the lower row. This was the simplest method in this case since parity bits or checksums would entail more lengthy processing. The complete set of markers as shown in figure 4.6 fulfilled all requirements outlined in section 4.3 and enabled an image representation of only eights bits to be employed in the sensor. Plate I illustrates the markers in use.



a) Location markers.



b) Stop marker

Figure 4.6 The Marker System.

PlateI Vehicle Positioning Using A Marker.

Chapter 5

THE POSITION SENSOR

The fundamental features of the sensor are illustrated in figure 5.1. An array of photodiodes is mounted behind a lens and receives a focussed image of an area 80 mm square, in a plane 300 mm in front of the lens, this area being illuminated by a light source situated next to the detector lens. In operation, markers appear to move across the image plane in focus, as the vehicle travels past them. The photodiode outputs are monitored continually for allowed patterns and in the event of recognition, the corresponding marker identity is output and maintained until another marker is recognized.

Initial sensor design fell into two areas: the transducer - involving method of detection, optical arrangement and illumination - and data processing - involving signal filtering and pattern recognition. The concepts leading to this design are discussed in sections 5.1 and 5.2, followed by a detailed description of the final implementation.

5.1 The Transducer.

5.1.1 Detection.

For an optical sensor giving a two-dimensional picture there is a basic choice between discrete photodiodes and either vidicon or solid-state cameras. Vidicons were considered too fragile for industrial truck applications and both types of camera provided far higher resolution than required, at very fast data rates. Solid-state cameras use photodiode arrays whose elements are typically 0.5 mm wide on 1 mm centres, thus low magnification lenses can be used. However, one is bound by the standard geometries and array sizes and two-dimensional arrays are particularly costly. Discrete devices provide flexibility in choice of resolution and image shape, but require care in building up arrays to ensure correct alignment.

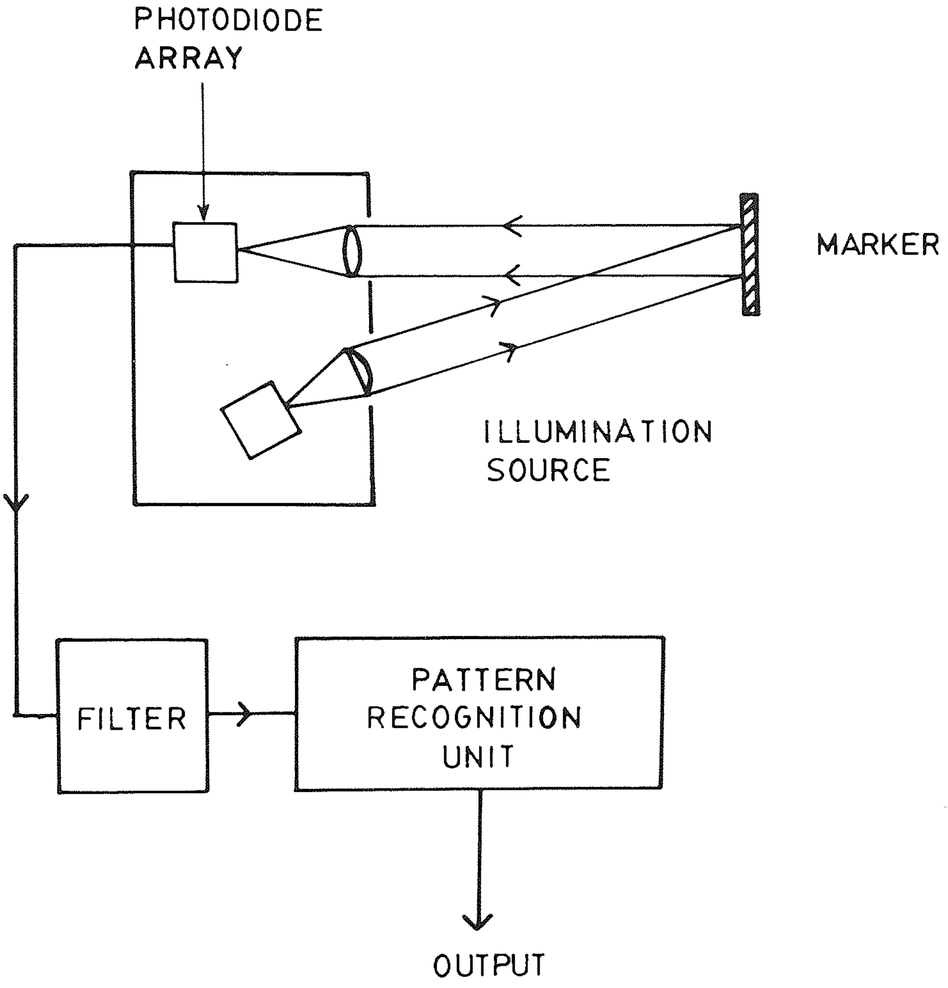


Figure 5.1 The Position Sensor.

The aim of the sensor was to find the minimum data processing requirements for the task, and since a marker system had been developed using only eight bits of information, discrete devices were chosen and built into the 2 x 4 array specified by the marker design.

5.1.2 Optical Arrangement.

Operating distance between sensor and markers was based on the prototype vehicle and its test-room layout and specified at 300mm, which required a lens system to focus light from the marker onto the photodiodes. For a thin lens, the relationship between magnification M and object distance u, is given by equation 5.1, where v is the image distance and x_i is the image width for a corresponding object width x_o, as illustrated in figure 5.2.

$$M = \frac{x_i}{x_o} = \frac{v}{u} \qquad \text{Eq. 5.1}$$

Now, u is specified and x_i will depend on the photodiode dimensions, so for a given device, the width of a marker element (x_e) is inversely proportional to v:

$$x_e = \frac{k}{v} \qquad \text{Eq. 5.2}$$

where k is a constant. x_e determines the resolution of the position error measurement at stop markers (x_e = a in figure 4.4), and v is the major contributor to the physical size of the sensor. Once these were chosen, the focal length f, of the detector lens was calculated from

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \qquad \text{Eq 5.3}$$

5.1.3 Illumination.

During operation, the vehicle was expected to encounter different lighting conditions, so the sensor was designed to be insensitive to

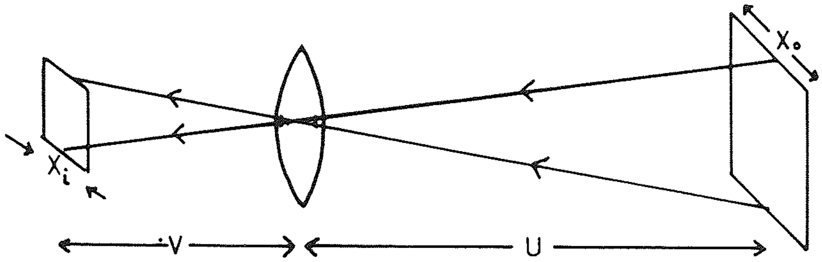


Figure 5.2 Thin Lens Magnification

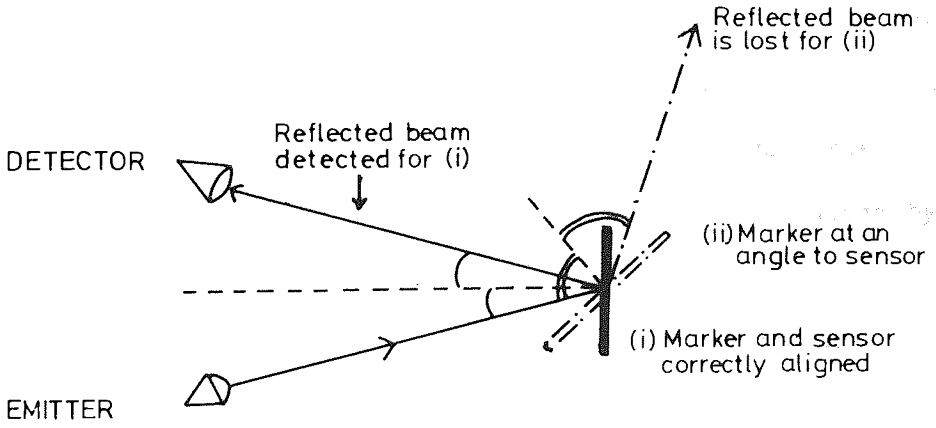


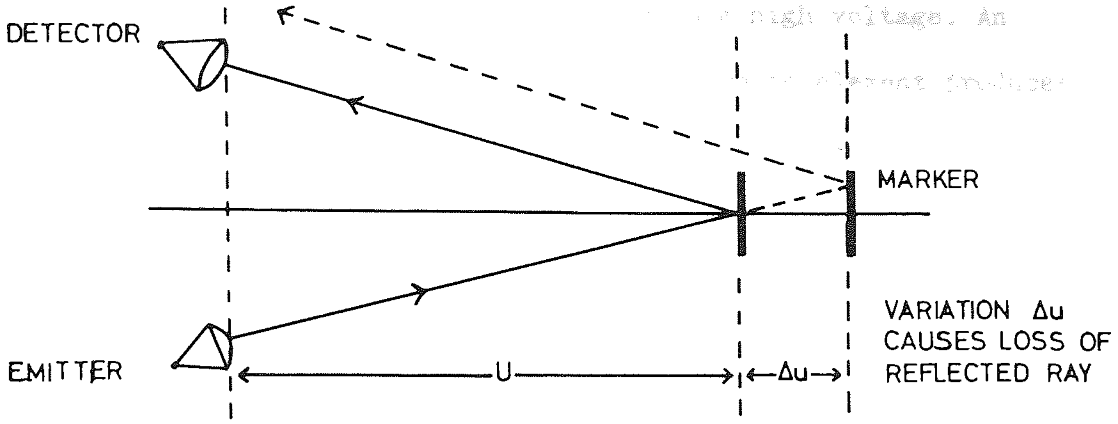
Figure 5.3 Sensitivity To Sensor Attitude For Spectral Reflection Method.

ambient light variations by providing its own pulsed illumination. The transducer received scattered light from markers when correctly aligned, and the photodiode outputs were monitored for amplitude variations at the same frequency as the illumination. Ambient light appeared as a superimposed d.c. level which was then filtered out. (Spectral reflections were avoided since the direction of the reflected beam would vary greatly with sensor distance and attitude to the marker as illustrated in figures 5.3 and 5.4)

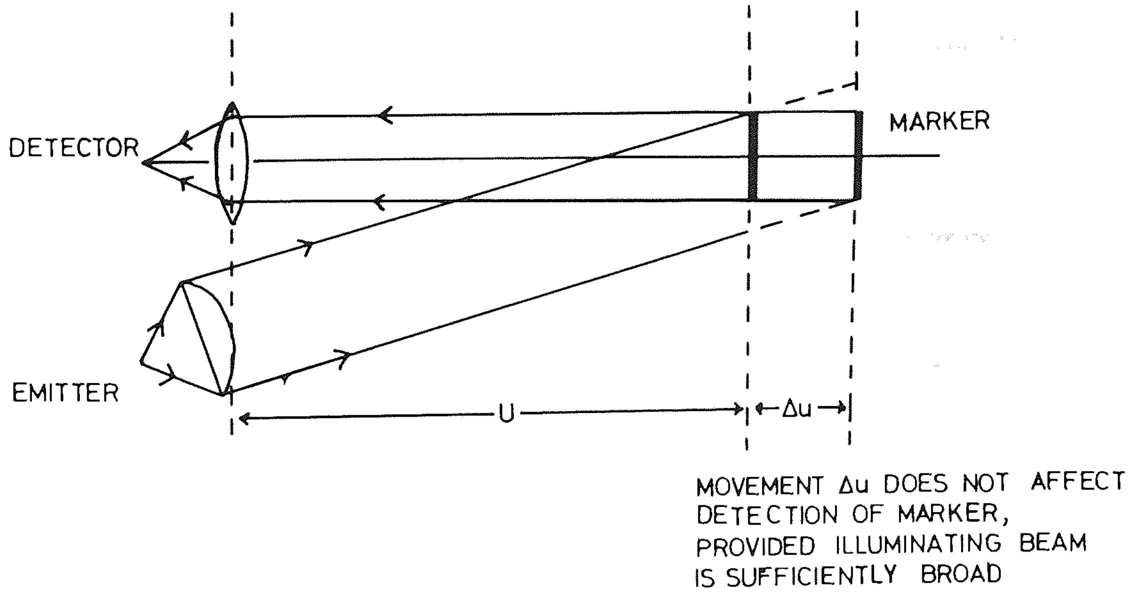
High intensity constant illumination, such as a floodlight, could not be used because the mechanical chopping needed would be susceptible to vibrations from the truck. Other considerations included size of mechanism and additional maintenance required. Xenon flashtubes are too fragile and would require a high voltage supply on the vehicle. Infra-Red Emitting Diodes (IREDs) provide an easily pulsed illuminating source and are commonly matched in wavelength to silicon photodiodes which have a large amplitude response in the near infra-red. The major drawback is that only low power devices are generally available (typically a few milliwatts), but some improvement can be gained by using high forward currents at low duty cycles, which produces high amplitude pulses whilst maintaining low average power dissipation. Consideration of signal processing requirements gave additional factors in favour of IREDs (section 5.2.2.1) and the problem of radiant power was overcome by using an array of devices mounted behind a large aperture lens to focus as much infra-red as possible onto the target area.

5.1.4 Transducer Response.

The output voltage from each photodiode and its associated amplifiers, is proportional to the total irradiance of infra-red incident on the



a) Spectral reflection



b) Scattering

Figure 5.4 Effect Of Changing Object Distance.

diode's active area. In effect, the photodiode samples its area of the image. If that area corresponds to a black marker element, the output voltage is low, and a white element results in a high voltage. An image containing part of a black and part of a white element produces a mid-range voltage proportional to the percentage of white area.

When the vehicle is moving past a marker, each photodiode scans that marker, and the output of a single device can be represented as in figure 5.5 by the convolution of the detector width with the spatial variation of white and black across its image area. In the case of a marker, the two functions convolved are rectangular and the result is a rounded rectangular function whose width is the sum of the two constituent function widths. Hence the degree of rounding, or smoothing, of the output depends on the relative widths of detector and marker. A very narrow detector would faithfully reproduce a broad marker, giving an alternating 'on' - 'off' signal corresponding to 'white' - 'black' as required.

However, as previously mentioned, the IREDs did not provide very strong illumination and only covered a small area in the object plane. To maximize incident illumination, the width of a marker element in the image was made to correspond to a photodiode width. From equations 5.2 and 5.3

$$X_e = X_p\left(\frac{u}{f} - 1\right) \qquad \text{Eq. } 5.4$$

where X_p is the photodiode width, hence the chosen device specifies marker size and focal length.

In summary, each photodiode/amplifier unit provides a voltage signal proportional to the incident infra-red which, in the case of a marker,

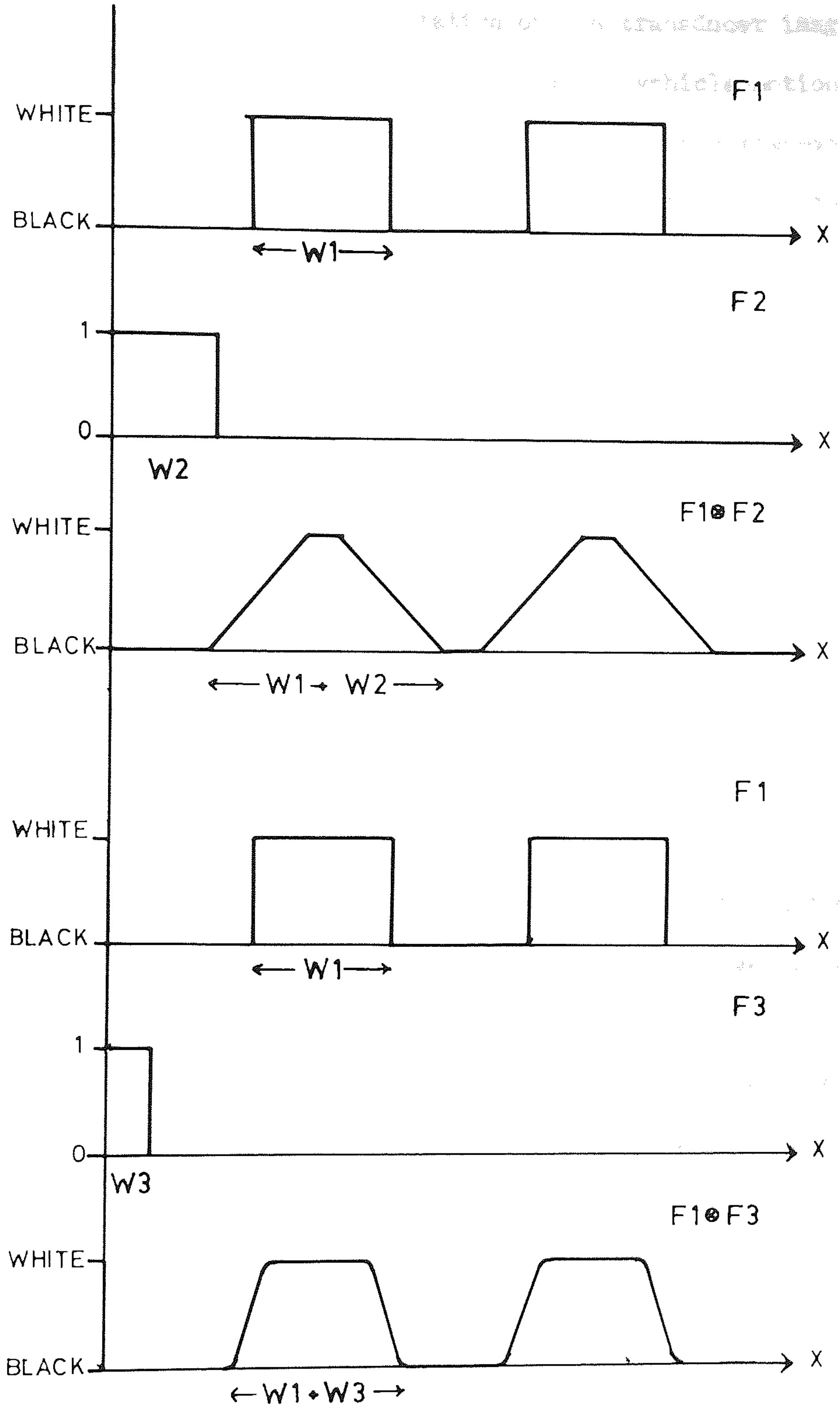


Figure 5.5 The Convolution Process For Different Detector Widths.

is proportional to the area of white viewed. The eight signals read in parallel provide a sampled representation of the transducer image in terms of spatial 'colour' variations. In addition vehicle motion causes each photodetector to scan a marker and produce a time-varying voltage at its output. Spatial frequencies in the transducer image (periodic occurrences of a certain pattern) are transformed into temporal frequencies related to vehicle velocity by

$$f_T = v f_S \qquad \text{Eq. 5.5}$$

where f_T is the output signal frequency in Hz, f_S is the spatial frequency in the object in m^{-1} and v is the vehicle velocity in ms^{-1}. Figure 5.6 shows the output of a single photodetector for various markers and how this varies with speed.

5.2 Data Processing.

With reference to figure 5.6, a simple threshold would recover the binary marker information. At any time instant, the eight photodetector outputs provide a sampled snapshot of the transducer image, which will contain all the required data for the case of a marker. Hence such snapshots could be compared with templates of all the markers to determine whether one of them was in view and if so which one. Filtering would be needed to remove higher frequencies than those given by passing markers (the frequency of the pulsed illumination source must not lie in the range of marker derived signals for example). Since vehicle speeds vary from zero to a maximum value, this range will cover all frequencies from d.c. to some maximum.

From the above discussion, data processing divides into the processes of filtering, thresholding and finally correlation or decoding of the recovered data.

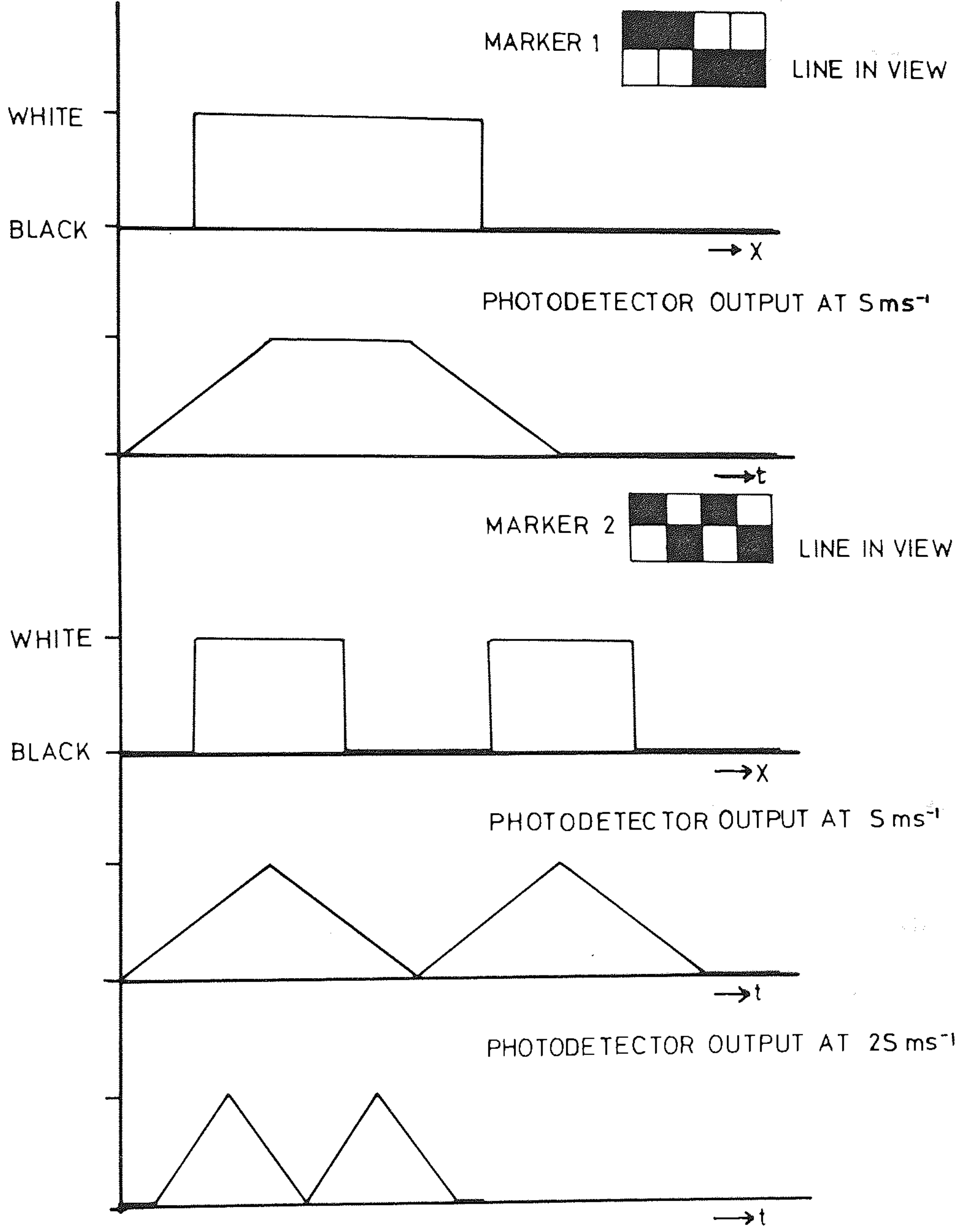


Figure 5.6 Photodetector Outputs.

5.2.1 Filtering - an analogue method.

The major problem with the transducer design was how to derive a measurable voltage from the photodetectors given the low power illumination available from IREDs. The emitters and detectors used were matched in wavelength for best response, but the highest radiant power devices generally available were rated at only 12mW. Photodiode response was typically 15μA light current under incident radiation of 200 Wm^{-2}(40) - equivalent to a separation distance of 200mm between emitter and detector. In the transducer, radiated infra-red must travel 600mm between emitter and detector and large losses occur in scattering. Instantaneous radiant power from the IREDs was increased as previously mentioned, and several diodes used in the source, but photodiode currents remained in the nanoampère range.

For this reason, an initial design was based on tone decoders which can be tuned to detect signals of a particular frequency. These give a logical output which is high when a matching frequency signal of greater than 40mV amplitude is present at the input. The photodiode outputs were amplified such that the 40mV condition acted as a threshold level and the tone decoders were tuned to the IRED pulse frequency.

However, the reference frequency had to be provided by an R-C network which in practice, necessitated variable resistors to fine tune each decoder. Variations in photodiode characteristics also required each amplifier to be individually adjusted to achieve suitable thresholding, and the design was discarded at this stage due to the high component sensitivity.

5.2.2 Digital Processing.

In addition to the above problems, the analogue design required a large number of components because each decoder circuit and any subsequent filtering had to be duplicated eight times. A microprocessor can achieve the same effect by simply multiplexing the eight signals through one block of software. For this reason, an investigation was made into the ability of a microprocessor to perform all data processing.

5.2.2.1 Sampling.

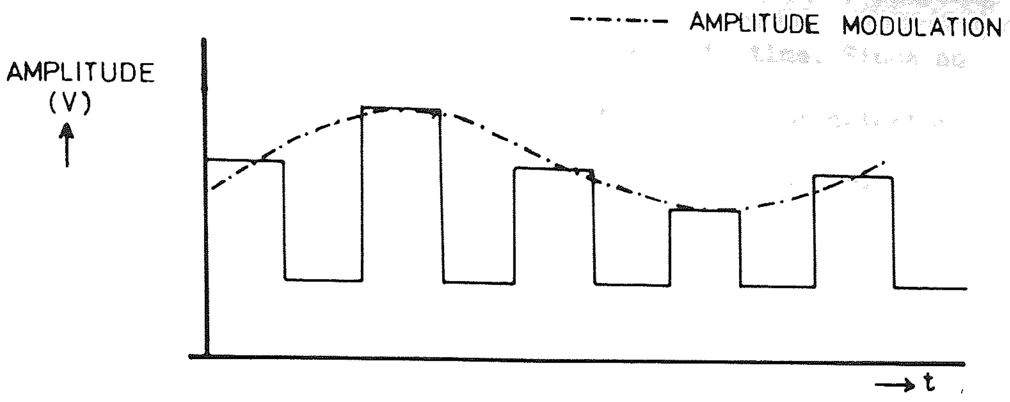
The first consideration for a microprocessor system is the conversion of input data into a form the processor can operate upon. For the photodetectors, this entails conversion of the output voltage into a binary input via an analogue to digital converter (ADC). Since the sensor output is to be used to control a moving vehicle, 'real-time' processing is required. This means that any operations on one data sample must be complete before the next sample is taken, and the time between sample and output must be sufficiently small for the control system to be able to take any necessary action. For example, when the vehicle is required to stop at a marker, the sensor must recognize each offset distance before the next one is passed, or the stopping point will be missed.

Sampling practice suggests that a signal must be sampled at ten times the highest component frequency present to retain all the original information content. The constraints of real-time processing set an upper limit to this sampling frequency and therefore to signals that can be processed. Applying this to the transducer, the pulsed illumination acts as a sampling function since no usable information is received by the photodiodes when the source is off. The pulse frequency must therefore be ten times the highest frequency signal likely to be given by a marker. Now, the highest spatial frequency occurring in the marker system is 40mm^{-1} corresponding to alternating black and white elements (each element is 20mm wide). Equation 5.5 therefore yields a temporal frequency of 25 Hz at the maximum specified vehicle velocity of 1ms^{-1}, and the pulse frequency must be at least 250 Hz.

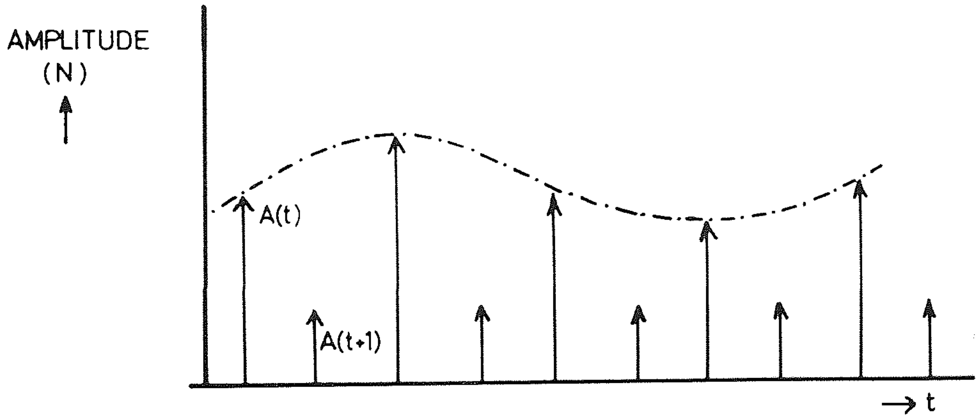
Using the ten times rule, sampling of the photodetector output signals must occur at 2500 Hz, giving 0.4ms for each sample and processing operation. This is beyond the capabilities of the Z-80 microprocessor employed. However, only the amplitude modulation is needed, so the carrier frequency can be discarded. The ten times criterion is based on retrieving both amplitude and phase information from an unknown signal, but a priori knowledge of some of these quantities can reduce the specified sampling rate. There was no appreciable phase difference between the illumination control and photodetector output signals, so a synchronous detection method was employed as follows: the microprocessor provided a square-wave carrier and sampled the photodetectors at the same points along each waveform as shown in figure 5.7b, eliminating the carrier whilst retaining the modulating frequency.

5.2.2.2 Digital Filtering.

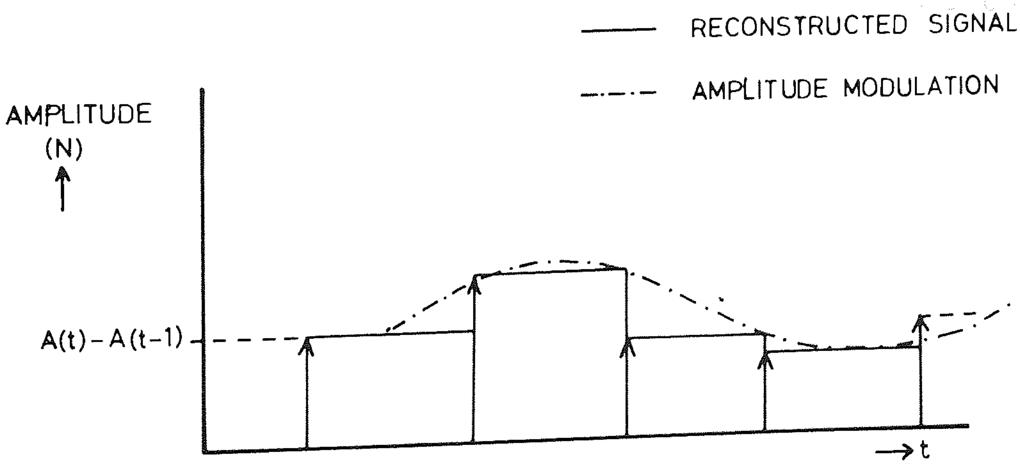
Once the signal had been sampled, filtering was required to remove frequencies higher than 25 Hz and any d.c. offset. Standard digital filtering implementations exist, but they generally involve multiplication operations which need long processing times or dedicated devices. The operating speed of the sensor required fast processing algorithms and the design aimed to avoid additional devices on cost grounds.



a) Detector signal



b) Synchronously sampled signal



c) D.C. filtering

Figure 5.7 Synchronous Sampling.

The combination of a square-wave carrier and synchronous detection provided a simple solution to the d.c. offset problem: the sampling rate was doubled to include a sample during every 'off' time. Since no illumination was provided during this instant, the photodetector output was solely due to ambient conditions. Adjacent samples were then differenced as in figure 5.7c to remove the offset.

Low-pass filtering was achieved using a running mean method (41). This is the simplest digital filter to implement because it only requires additions and subtractions. The algorithm is given by equation 5.6, where n is the number of samples in the mean and k and r are integers.

$$y_k = \frac{1}{n}\sum_{r=0}^{n-1} x_{k-r} \qquad \text{Eq. 5.6}$$

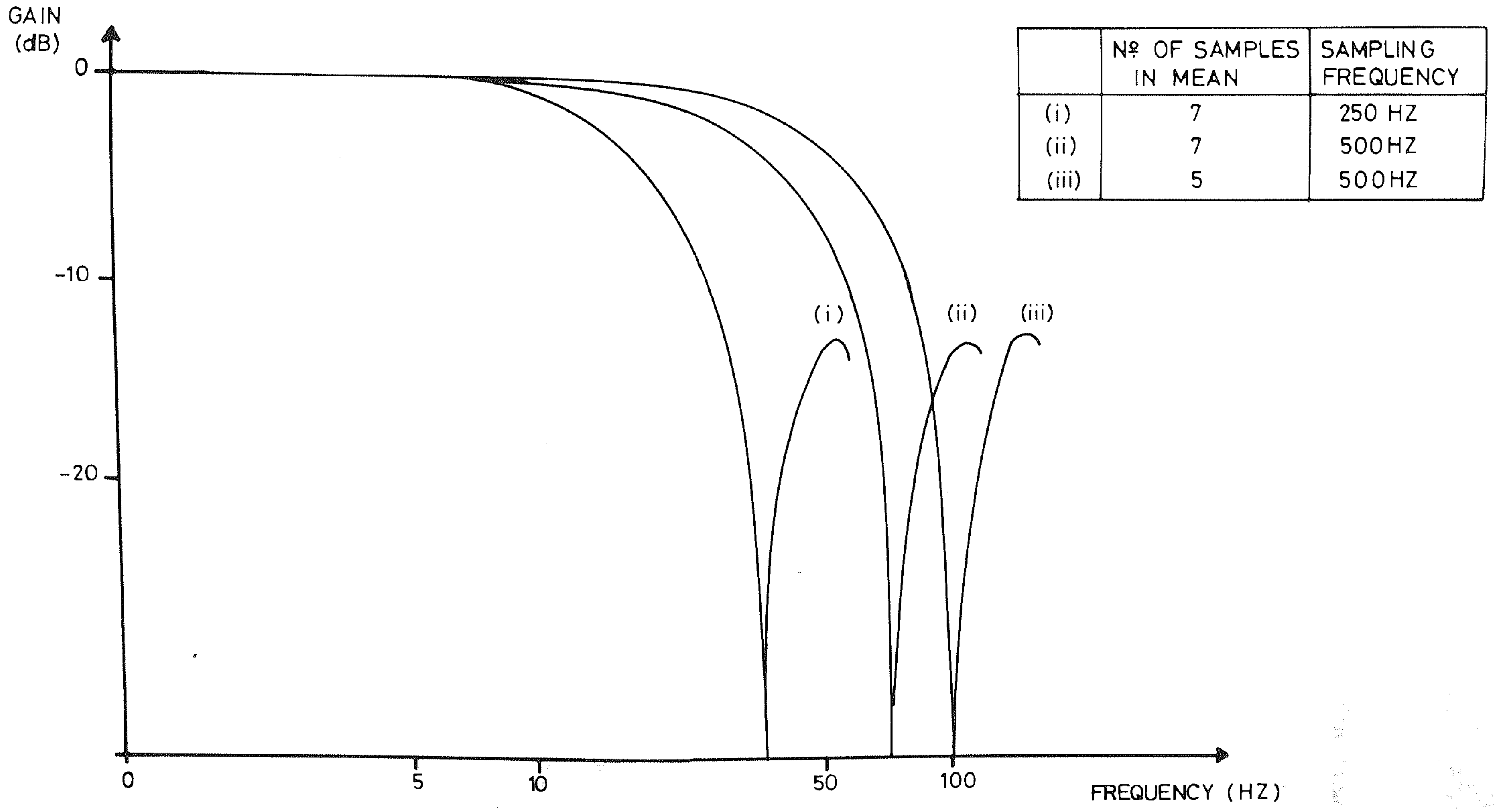
The output at any instant (y_k) is equal to the mean of the last n input samples (x_{k-r}), and can be re-written in the form

$$y_k = y_{k-1} + \frac{1}{n}(x_k - x_{k-n}) \qquad \text{Eq. 5.7}$$

In practice, the division by n was avoided by multiplying the thresholds used by the same factor. Figure 5.8 shows the frequency response for this filter and illustrates its dependence upon both the number of samples in the mean and the sampling frequency.

5.2.3 Thresholding.

Following the filter, the signal was thresholded to reduce the data to a binary input for the recognition process. A band was used to alleviate oscillations about the decision level due to noise. This also effectively removed signals due to higher spatial frequencies than those employed in the markers, since the convolution of a relatively wide photodiode aperture with narrow black and white stripes averages to a 'grey' level within the inadmissible region defined by the



	Nº OF SAMPLES IN MEAN	SAMPLING FREQUENCY
(i)	7	250 HZ
(ii)	7	500HZ
(iii)	5	500HZ

Figure 5.8 Frequency Responses Of Running Sum Filters.

threshold band.

5.2.4 Pattern Recognition.

Figure 5.9 summarises data processing in the sensor. Sampling, filtering and thresholding take place on each photodiode output individually, but pattern recognition operates on the eight signals in parallel for maximum speed. The trade-off between parallel and serial processing depends on the mumber of bits required to characterize an image and the total number of markers used. As these two quantities increase, direct point to point correlation between image and marker templates becomes time consuming and the storage of templates occupies large areas of memory.

However, in this case, the thresholded signals were combined to form a single eight-bit word, achieving an eightfold data reduction and allowing the fast eight-bit comparison instruction of the Z-80 to be used. Four location markers were used with the prototype and the stop marker employed seven recognizable views, thus the template table only required eleven bytes of storage and point to point correlation was the most efficient method to employ.

Each photodiode output was represented by a specific bit in the image word. Each bit was set according to the results of the thresholding operation on that signal (set for white, reset for black) and the entire word was rendered invalid for that sample instant if any signal fell within the threshold band. Only valid image words were compared against templates, and in the event of a true compare, the associated identifier was accessed from the table and placed on the sensor output.

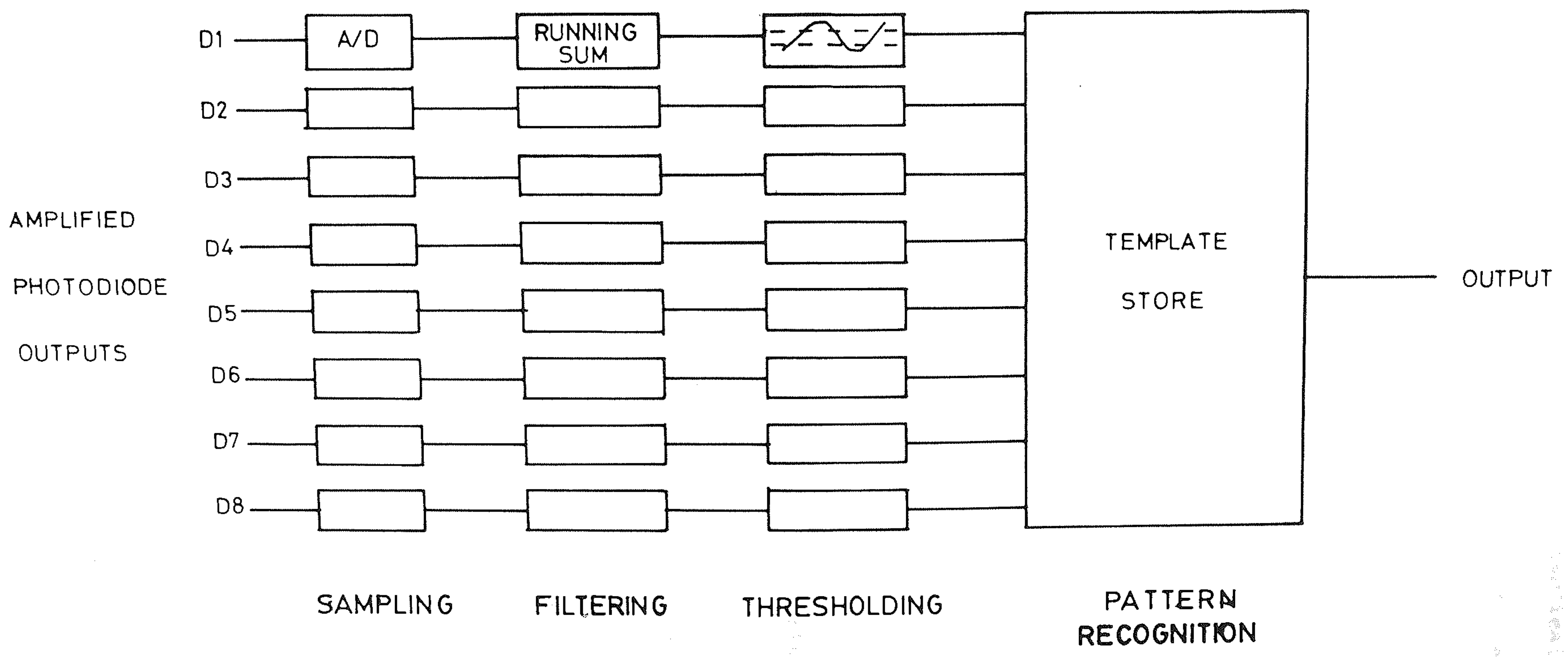


Figure 5.9 Data Processing In The Sensor.

5.3 Implementation.

The above sections have outlined all aspects of the sensor and its operation in theory, with reference to broad practical considerations. Physical realization of a prototype required a detailed look at available components and the interfacing between transducer and microprocessor. Then data processing and sensor control software had to be written and adjusted to obtain the correct sampling frequency. This section investigates the problems encountered in building the sensor, and describes the final design used on the first prototype vehicle.

5.3.1 Construction and Optics.

Figure 5.10 shows the sensor construction and optical arrangement. Detectors and emitters were aligned and rigidly fixed relative to their respective lenses by the bakelite structure, which incorporated the required angle between lens axes. This angle (θ in fig. 5.10) was made as small as possible to prevent large amplitude variations in the illumination across the marker plane. The vehicle's route following system was specified to track the guide-wire to a tolerance of ± 1 cm, so these two extremes were used to find angles which prevented spectral reflections entering the detector lens. In practice, the minimum value of θ was dictated by the physical size of the lenses.

Specification of the optical components proved to be a trade-off between size of sensor and size of marker (section 5.1.2). Commercial lenses were the cheapest and most readily available and limited the choice to standard focal lengths. A flat-field enlarger lens was used for the detectors to reduce radial image degradation due to spherical aberrations. (The flat-field arrangement comprises a convex lens which would focus the image onto a spherical surface, followed by a concave lens to transform this sphere into a plane and maintain image quality

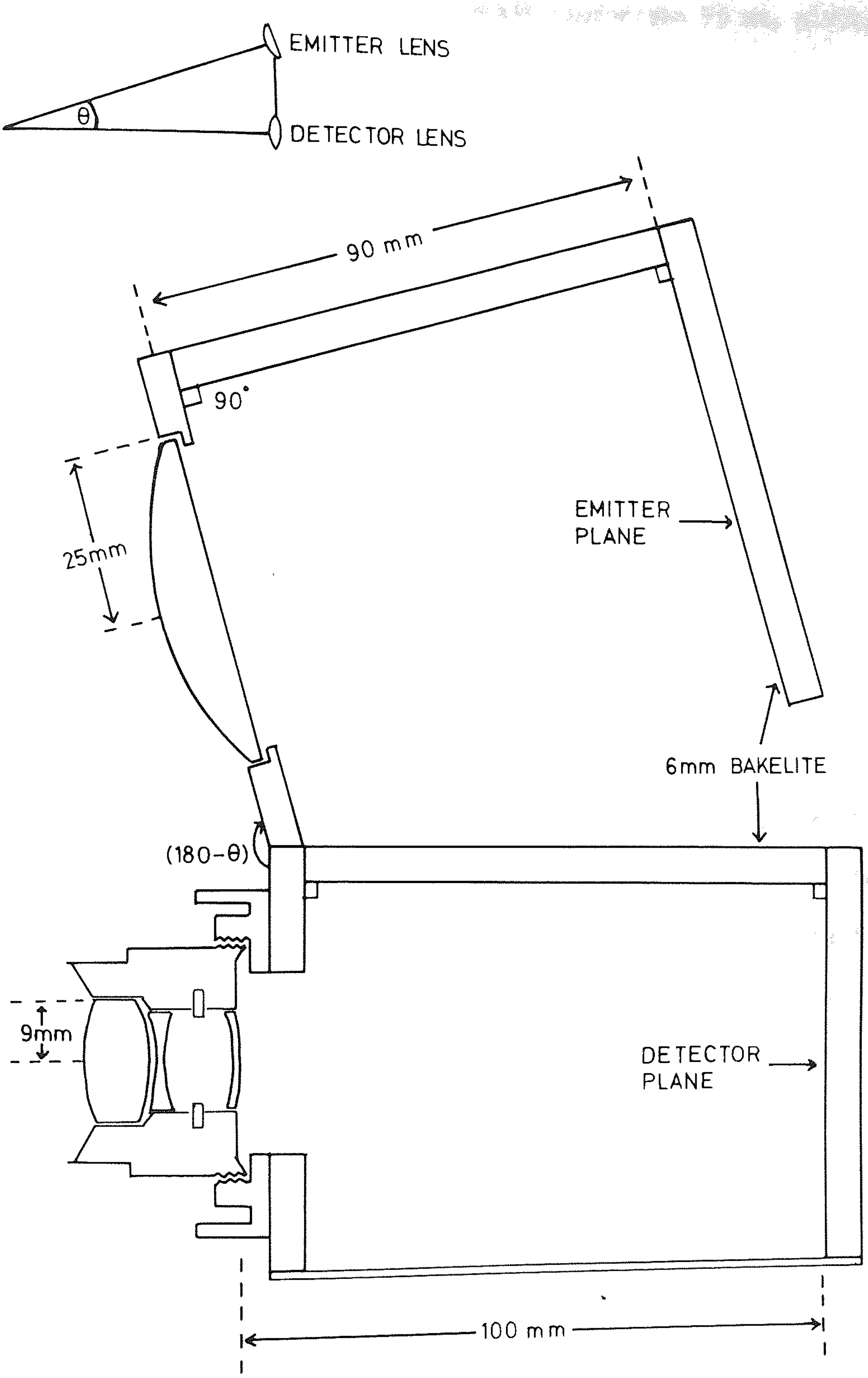


Figure 5.10 Position Sensor Construction.

at the edges). The widest aperture setting was used to allow the maximum light into the lens. The focal length chosen was 75 mm, giving a magnification of $\frac{1}{3}$.

5.3.2 Illumination.

Ideally, the illumination should have been evenly distributed across the object plane of the detector lens, covering the area of a marker. However, the IREDs employed exhibited non-uniform irradiance in a plane normal to the beam as shown in figure 5.11. In addition, there was considerable variation in output between individual devices, and a very small decrease in amplitude across the object due to the angled beam.

Strength of illumination was a premium, as already mentioned, so an array of IREDs was used with a wide aperture lens to focus a magnified image of the array onto the marker plane. The number and position of diodes in the array was determined empirically, using a photodiode to map the irradiance patterns in the marker plane as shown in figure 5.12. The results for various emitter positions were correlated to predict an arrangement where the individual patterns overlapped to produce even illumination. This prediction was used in the first sensor and is shown in figure 5.13 with its measured distribution.

At a later stage, the question of illumination was reconsidered in an effort to reduce the very high gain in the photodiode amplifiers, and the number of emitters was doubled. Due to the lens geometry, the emitters in the mark 2 sensor had to be arranged such that each emitter illuminated the area viewed by one detector. Plate II shows a sheet of white paper placed in the marker plane and illuminated solely by the infra-red source in the sensor. The photograph was taken on infra-red

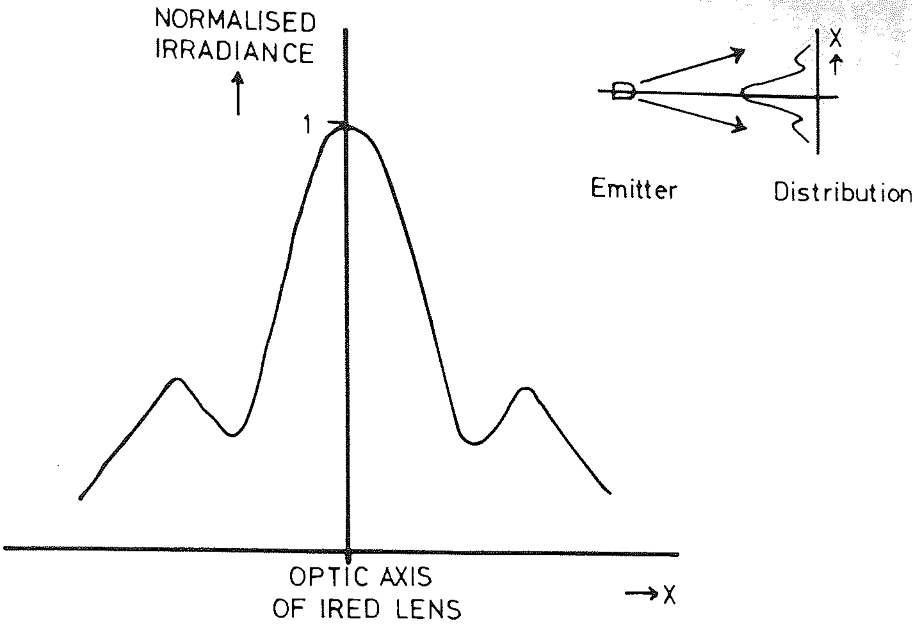


Figure 5.11 Irradiance Distribution Of RS 308-512 Emitter.

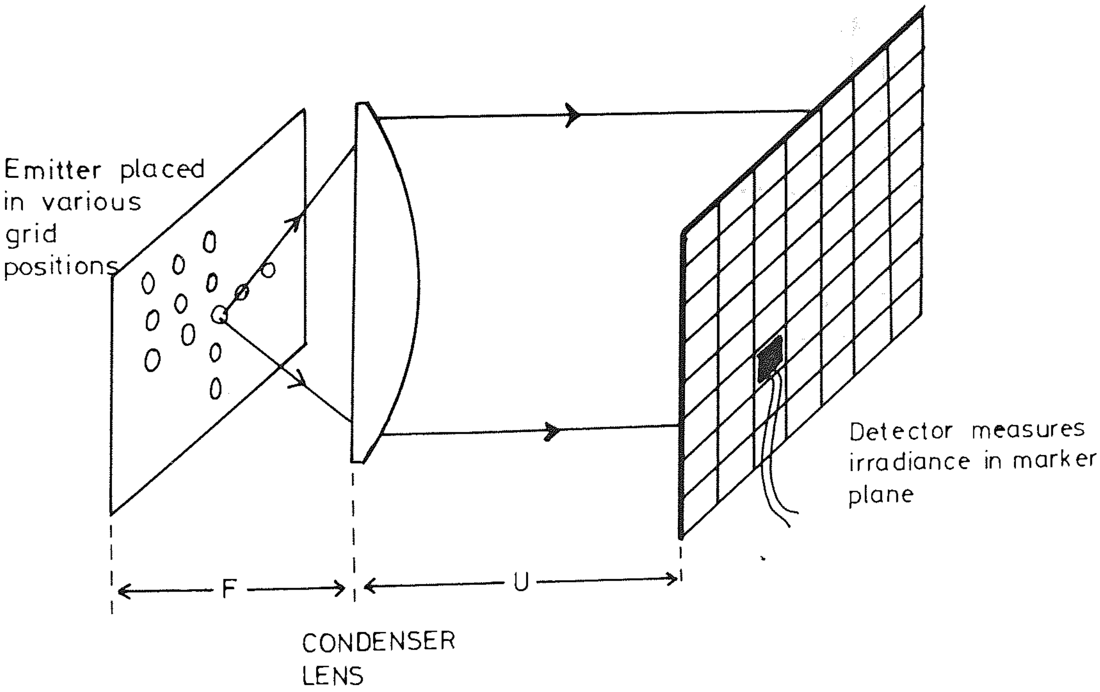
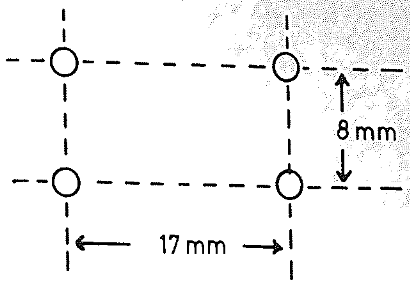
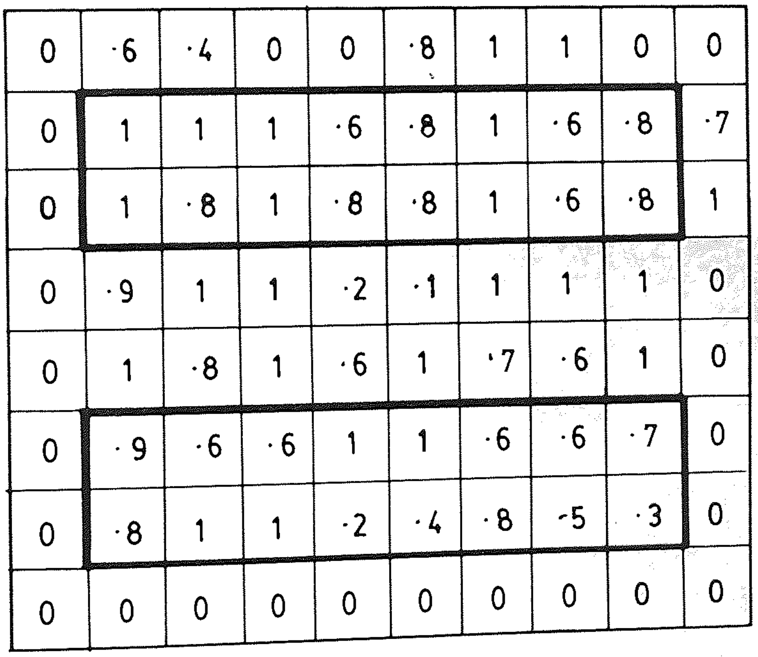


Figure 5.12 Investigation Of Marker Illumination.



a) Emitter array dimensions for minimum device system



0	·6	·4	0	0	·8	1	1	0	0
0	1	1	1	·6	·8	1	·6	·8	·7
0	1	·8	1	·8	·8	1	·6	·8	1
0	·9	1	1	·2	·1	1	1	1	0
0	1	·8	1	·6	1	·7	·6	1	0
0	·9	·6	·6	1	1	·6	·6	·7	0
0	·8	1	1	·2	·4	·8	·5	·3	0
0	0	0	0	0	0	0	0	0	0

□ AREA SAMPLED BY DETECTORS

b) Distribution of illumination using array in a)

Figure 5.13 Marker Illumination For Sensor 1.

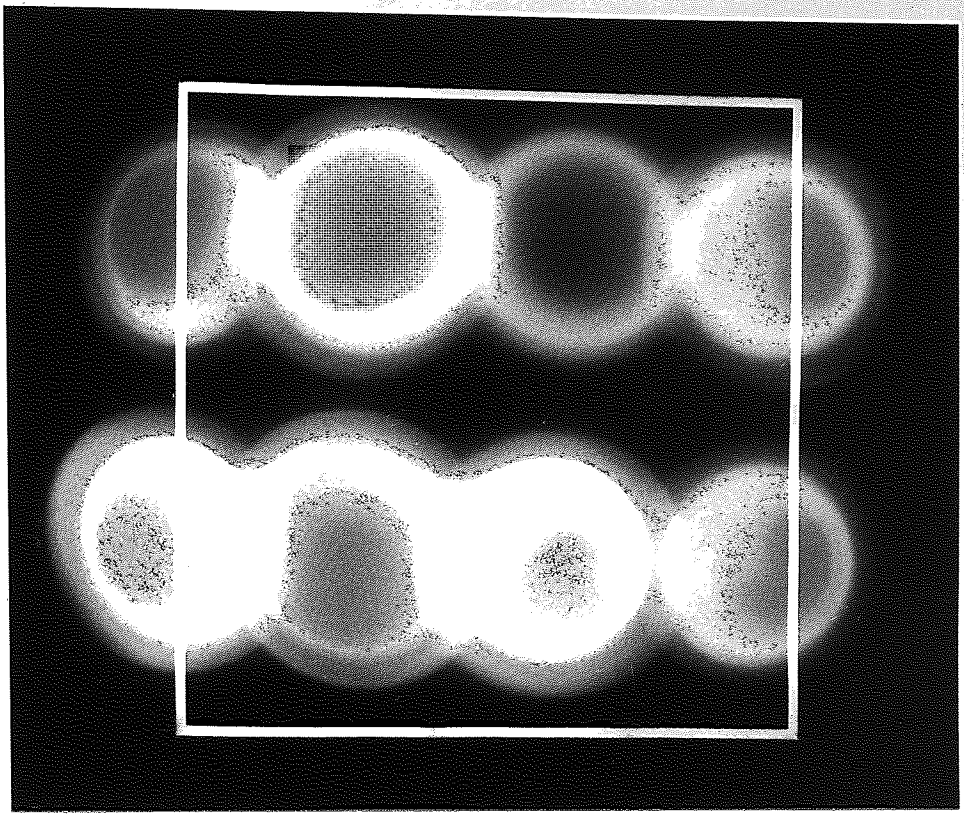


Plate II Marker Illumination Using Eight Emitters.

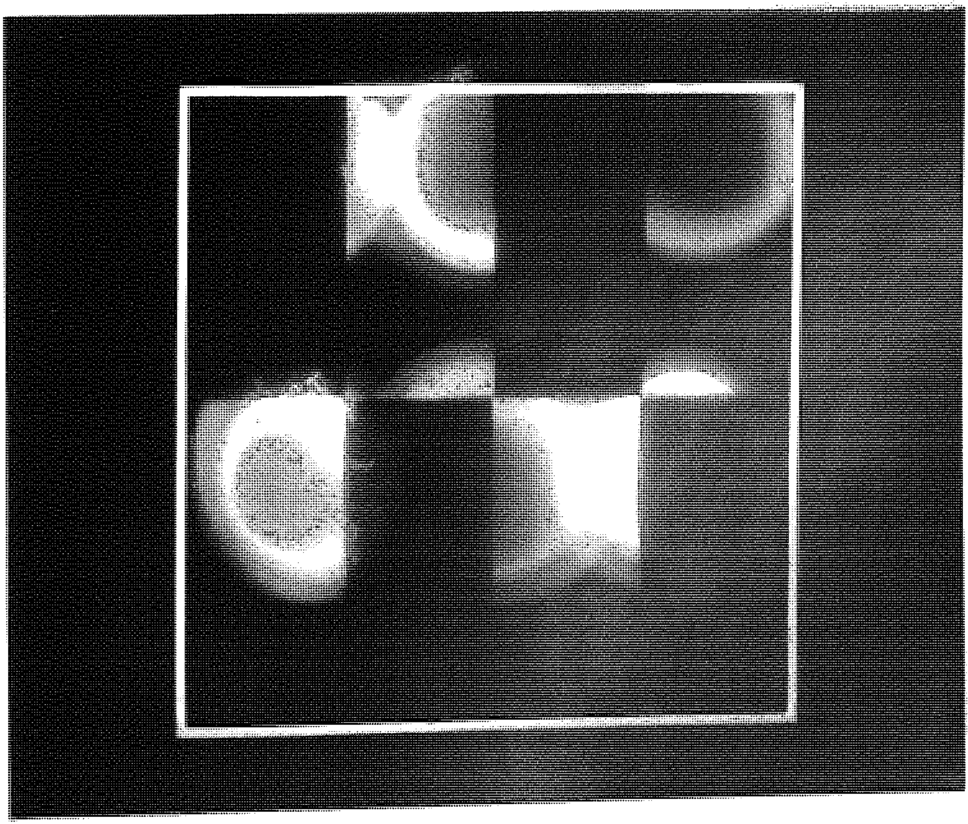


Plate III Marker Under Illumination.

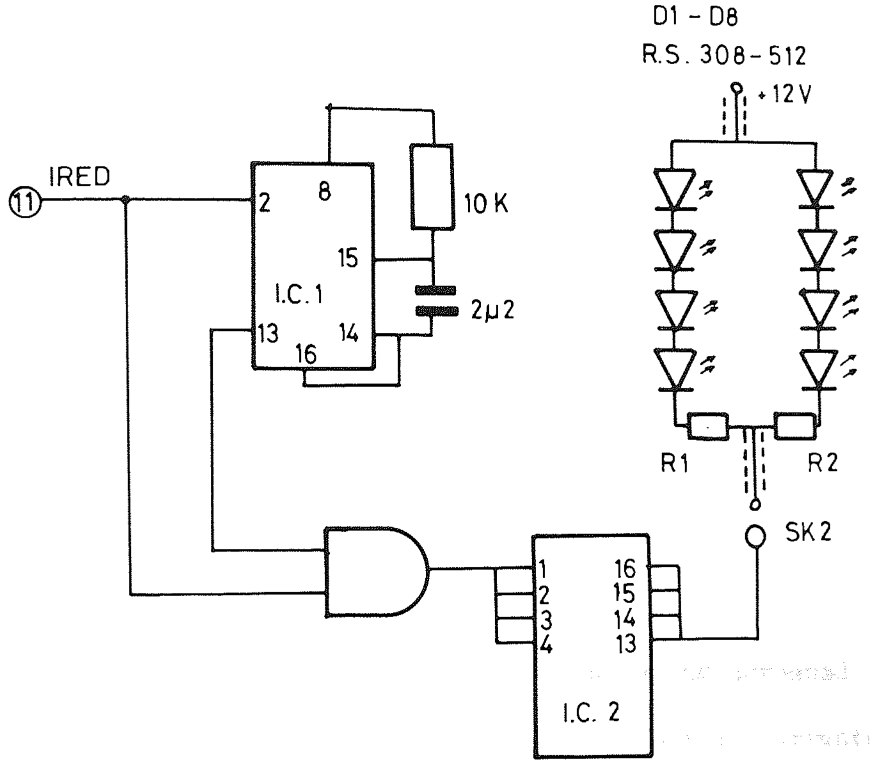
sensitive film (42) and clearly shows the characteristic irradiance distribution of individual IREDs and demonstrates the brightness variation between emitters. In Plate III, a marker has been placed in the sensor field of view, showing the contrast between irradiance levels from black and white elements. (The marker is slightly misaligned). The performance of the two sensors is compared later in Chapter 7.

High power RS 308-512 infra-red emitting diodes were driven by the circuit in figure 5.14 to produce the IRED source. Under microprocessor control, the IREDs were pulsed by a 1A peak current at 250 Hz with an 'on' time equal to the ADC conversion time (typically 100μs). To prevent the possibility of sustained high current damaging the emitters because of microprocessor failure, a safety trip was provided by the monostable IC 1: the time constant was set to 10ms by R1 and C1 so that in normal operation, the monostable was continually retriggered by the control signal. If however, there was a delay of more than 10ms between pulses, the monostable output went low, disabling the AND gate output to the Darlington drivers and hence the emitters. This condition held until the next rising edge in the control signal when normal operation recommenced.

5.3.3 Detection.

RS 308-506 photodiodes were employed in the detection unit as they had sufficiently fast response times to faithfully detect the 250 Hz illumination pulses and were matched to the emitters used. They also had integral optical filters to reduce sensitivity to visible light.

The photodiode outputs were interfaced to the microprocessor through Analogue to Digital Converters (ADCs) whose input signal range was



R1,R2	4R	0·5W
I.C.1	74LS123	RETRIGGERABLE MONOSTABLE
I.C.2	307-109	DARLINGTON ARRAY

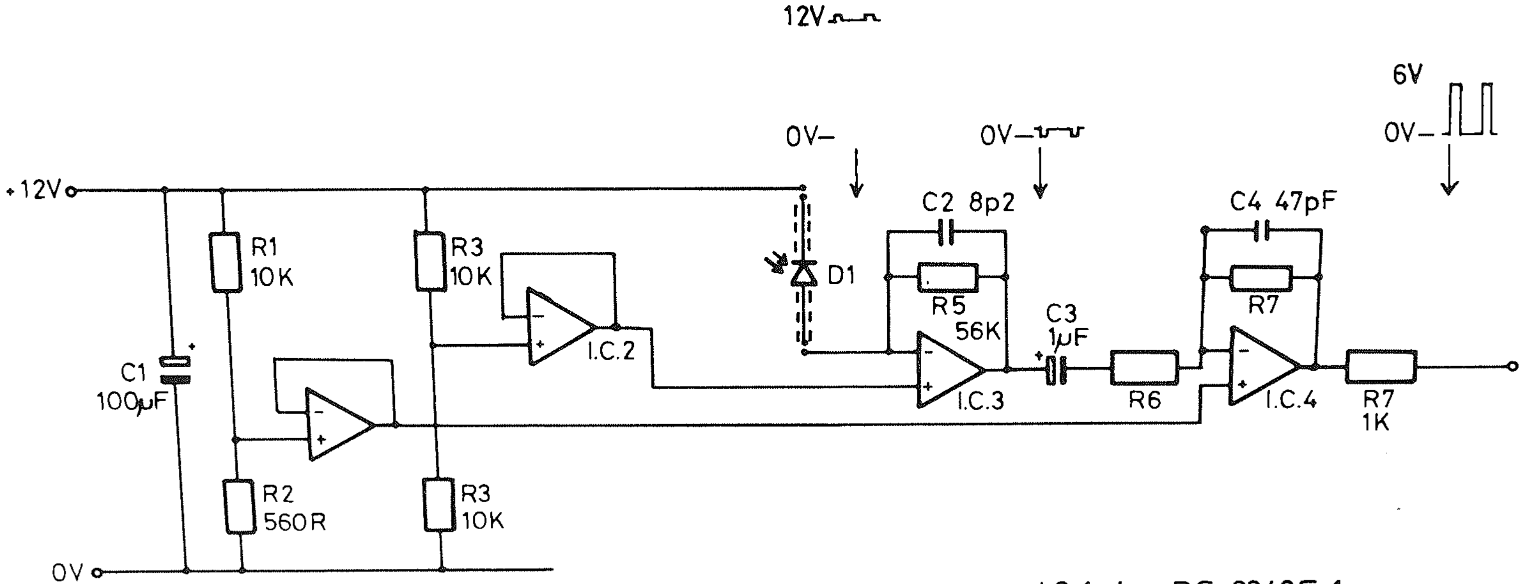
Figure 5.14 Emitter Driver Circuit.

0 to +5V, thus requiring considerable amplification. This was provided by the circuit of figure 5.15, consisting of a current amplifier followed by a second stage voltage amplifier. It was found that the diodes required a minimum bias voltage of 5.5V to operate, hence the 12V supply used. The capacitor-resistor combination at the output stage of the first amplifier was designed to remove the d.c. signal, whilst the voltage reference and closed loop gain of the second amplifier were chosen such that the output was in the range 0-5V. In practice, resistor R7 had to be individually selected for each detector element. The value ranged from 100kΩ to 220kΩ in the mark 2 sensor (eight emitters) and was of the order of 330kΩ in the mark 1 sensor. Signal levels at various nodes in the circuit are shown on the diagram to illustrate operation.

Due to space limitations in the sensor unit, the photodiode array was separate from the amplifier array, necessitating screened flying leads to each device. In view of the very low photodiode currents involved (nanoampères), lead lengths were kept to a minimum.

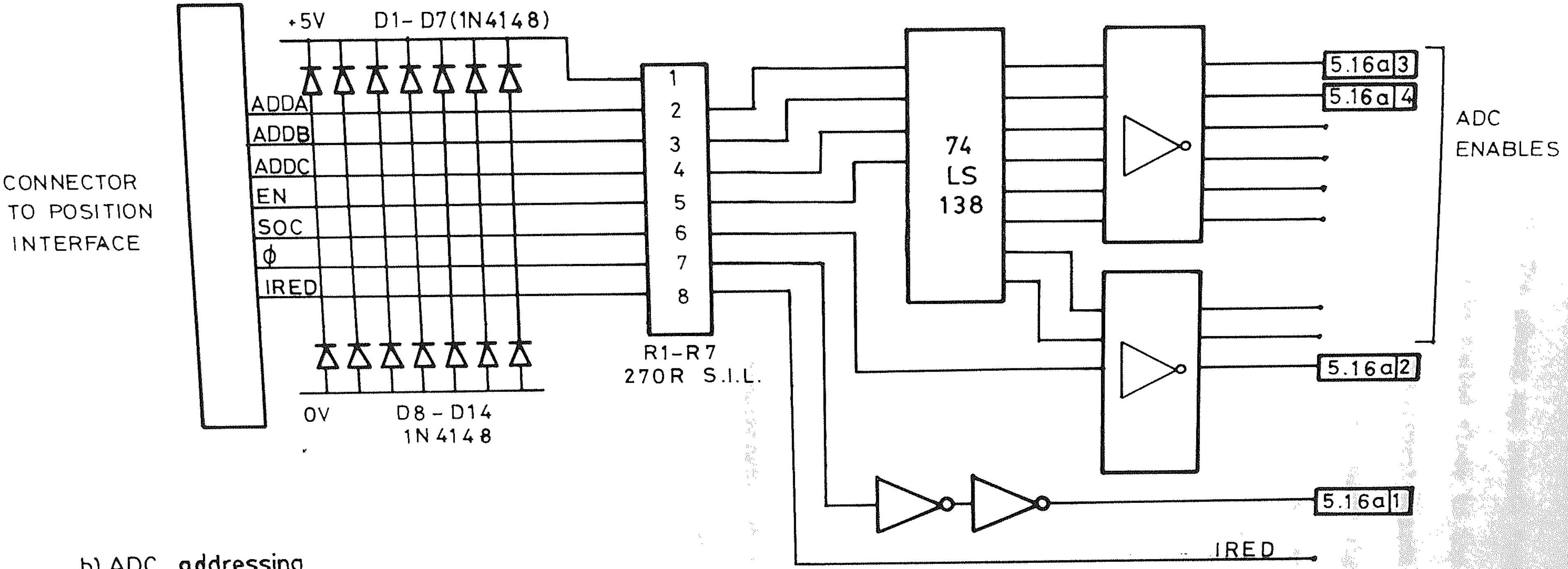
5.3.4 Processing Hardware.

Sampling of the photodetectors was performed in parallel, using an array of eight-bit ADCs as in figure 5.16a. National Semiconductor ADC 0809 devices were used, having a conversion time of 100 μs (with a 1MHz clock input) and latched tri-state outputs. This latter feature allowed the microprocessor to poll each one in turn and read its output from the common data bus. The system software controlled data collection through a three-bit address field which was decoded to one of the eight ADC output enables as in figure 5.16b. A start conversion signal was fanned out to each device and the end of conversion output of just one ADC was used to trigger the polling process.



I.C.1-4 : RS 3240E-1
D1 : RS 308-506

Figure 5.15 Photodiode Amplifier Circuit



b) ADC **addressing**

Figure 5.16 Sensor Interface.

When the system was installed on the vehicle, all microprocessor units were placed in a centralized rack. The sensor was remote from this point and the connecting cable had to be routed close to the battery and traction motors. For this reason, the constituent boards were arranged such that this connection carried low impedance signals, which were passed through line drivers as they left a board and were pulled-up by a resistor to the positive supply on entry. Catching diodes were also placed on all board inputs for device protection against noise. A block diagram of the sensor hardware is given in figure 5.17, showing the division of circuitry between boards and the connection between sensor unit and controlling microprocessor.

The control and data buses for the sensor were interfaced to the microprocessor through a Z-80 Parallel Input-Output controller (PIO) housed on an interface board in the central vehicle rack. This device has sixteen bi-directional lines individually programmable as inputs or outputs and divided into two eight-bit ports. The Z-80 Central Processing Unit (CPU) reads or writes to one port at a time and can operate on the data either as a complete word or on a bit by bit basis. Hence the ADC data bus was assigned to one port for processing as a word, and the control bus was assigned to the other port.

The company had standardized on a circuit for the basic needs of microprocessor operation (board 5 in figure 5.17), which was to be employed on each processor sub-system on the vehicle. In essence, this board contained a Z-80 CPU,a PIO and up to six kilobytes of Erasable Programmable Read Only Memory (EPROM) for operating software. One kilobyte of Random Access Memory (RAM) was provided for 'scratchpad' memory and storing variables during program operation. Other features included a reset circuit and a clock generator. The re-set function initialized programme execution from the first instruction in the EPROM, and was automatically activated on power-up. A push-to-make switch was also provided for manual operation. The clock generator provided a buffered 4 MHz clock signal for the CPU and its support devices. Finally, the CPU address and data buses and all control lines were buffered and brought out to an edge connector to facilitate expansion.

Power supplies for all units within the vehicle control system were derived from the traction battery. A centralized dc-dc converter was used to generate a +5V supply for TT logic and the ADCs for all boards, but the 12-volt detector supply was unique and therefore derived within the sensor unit from the smoothed battery supply of 24 volts.

5.3.5 Software.

The sensor programme had two tasks: to control the illumination and analogue to digital conversion to effect correct sampling; and to process the data for recognized patterns. For programming simplicity, these were kept separate - no calculations were performed during ADC conversion time.

5.3.5.1 Sensor Control.

This comprises a sequence of events which are repeated continuously. With reference to figure 5.18, the infra-red emitters are turned on, and after allowing a suitable period for the detectors to reach full output, a start of conversion (SOC) pulse is sent to the ADCs. This causes the end of conversion (EOC) input to switch to its 'high' level after a short delay. To allow for the rise time of EOC, the multiplex address of the first ADC is output and enabled (EN), before polling for EOC low begins. Once conversion is complete, the IREDs are turned off and data can be read in. After the data from one ADC has

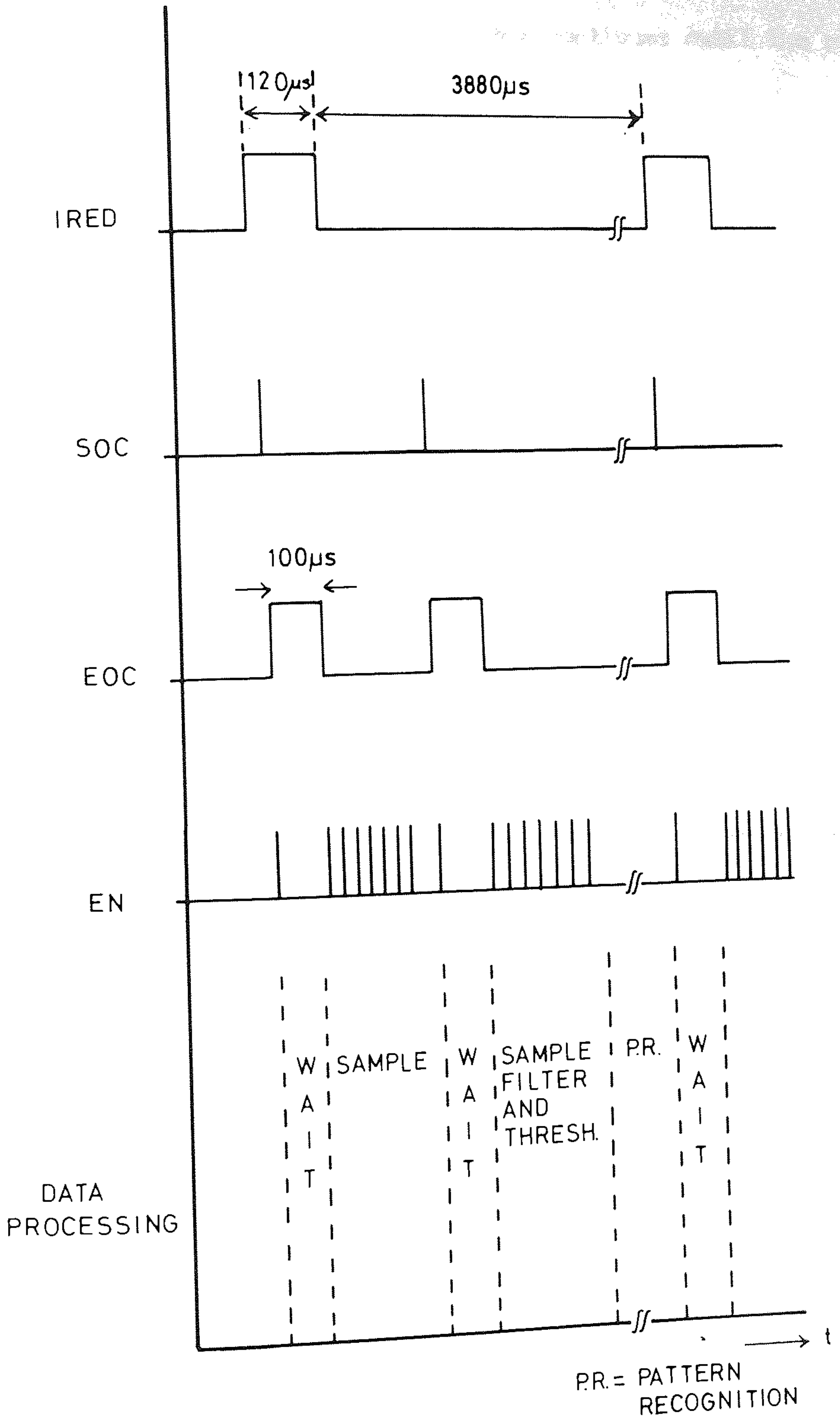


Figure 5.18 Timing Diagram For Sensor Software.

been read, the next is addressed and its output enabled onto the bus. When all data processing is complete on the current reading, the bus is read and the next ADC enabled, and this continues until the eight photodetectors have been sampled.

Once the on-time readings have been taken, the above process is repeated except for the IRED line which is kept low. When the last 'off-time' sample has been taken and the pattern recognition process is complete, the processor waits for the end of the current sample period before repeating the complete sequence.

5.3.5.2 Data Processing.

The operations of filtering, thresholding and pattern recognition are distributed within the control sequence as illustrated in figure 5.18. Real-time operation at this sampling rate meant that the software had to be written in Z-80 assembly language where timings of operations can be controlled. The algorithm is given in figure 5.19.

Running sum filters were computer modelled using various run-lengths at the sampling frequency, and the required frequency response chosen. The filter was implemented by reserving n bytes of storage for each detector, where n was equal to the run length, and each byte held one sampled number. This store was run on a first in, first out basis where the newest reading replaced the oldest. In addition, an array of sixteen bytes was reserved for the running total for each detector at any instant (two bytes per detector). The two tables and filter operation are presented diagrammatically in figure 5.20. At any instant, pointer OLDEST refers to the oldest sample in store for the detector being processed, and TOTAL points to the current running total for that detector. On receipt of a new reading, the total is updated by

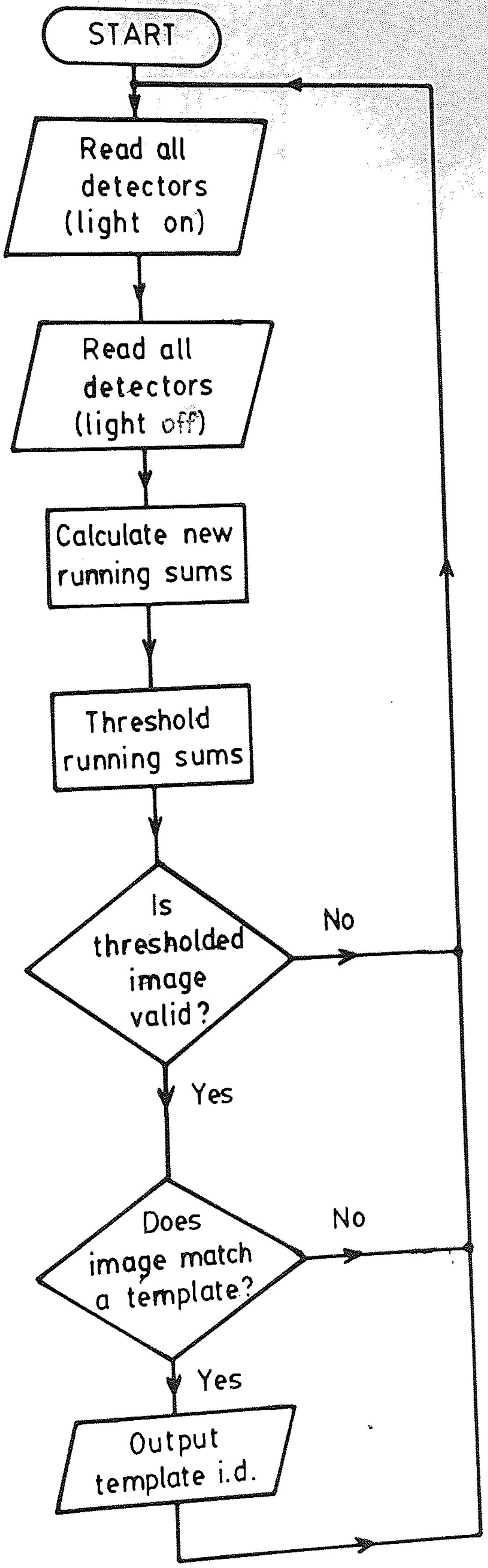
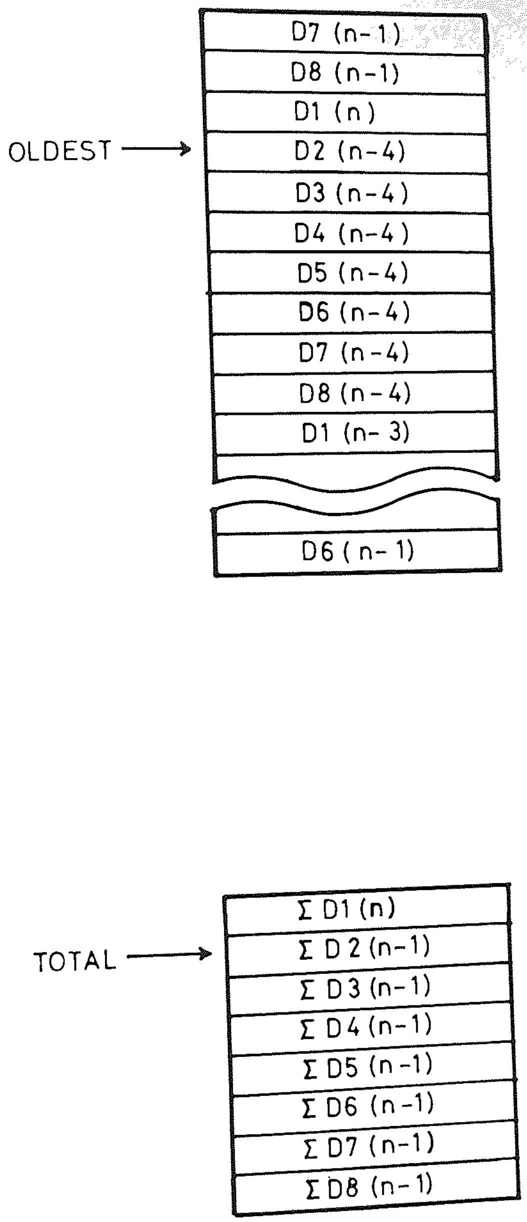
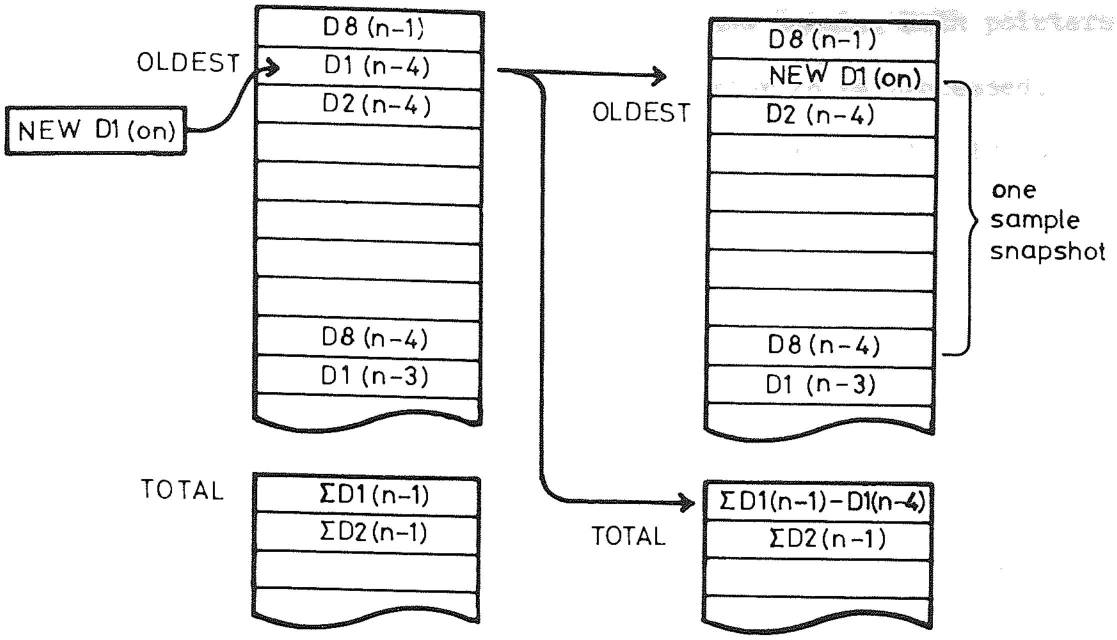


Figure 5.19 Data Processing Algorithm.

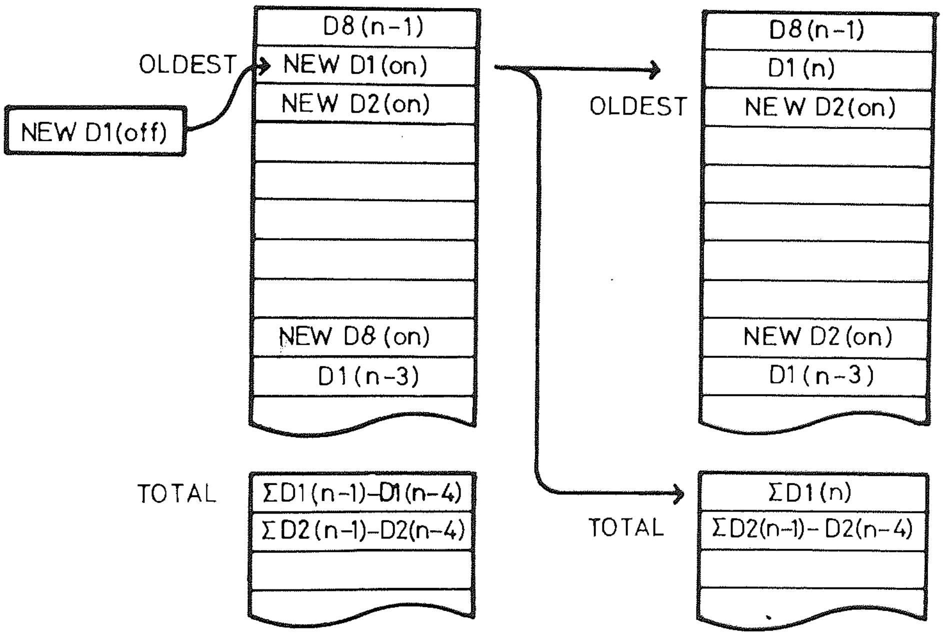


a) Data storage for running sum operation

Figure 5.20 Implementation Of The Running Sum Filter.



b) Processing for 'on' reading.



c) Filtering complete for detector 1.

Figure 5.20 Implementation Of The Running Sum Filter.

adding the new one and subtracting the oldest. This latter value is then overwritten in the sample table by the new sample. Both pointers are then incremented ready for the next detector to be processed. Because of the d.c. filter, the running sum operation actually takes place in two parts: first the oldest sample is subtracted from the total and the newest on-time reading is placed in the sample table as before. After all the on-time readings have been processed, both pointers are placed back at the top of the present 'snapshot' (see diagram) and off-time samples are taken. Each newest reading in the sample table is 'filtered' by subtracting the relevant 'off' sample and finally, the total is updated by adding on this amended new reading. When all eight filter operations have been completed for one sensor snapshot, TOTAL is re-set to the top of its table and OLDEST is simply incremented once more ready for the next sample. When OLDEST reaches the end of its table, it must be reset to the top.

As each new detector total is computed, the next stage of processing can take place: the total is compared with two threshold levels to determine whether this sample represents a black or a white image or some grey-level which is declared invalid. The result of this function is one for white and zero for black and is stored in the relevant bit of the one byte array CODE. Bit 0 represents detector 1 and Bit 7 represents detector 8. If the sample is invalid (the total falls within the threshold band), CODE [n] is unchanged and a flag is set. When the array has been completed, this VALID flag is tested to determine whether pattern recognition should proceed. Although only one invalid sample will render further processing unecessary, all samples are thresholded to keep the programme execution, and hence sample time, constant.

Every valid CODE is compared with each stored marker template in turn. If a true compare exists, the storage location of that template serves as a pointer to the marker identifier which then becomes the sensor output. The sensor microprocessor is also used to run the position control program, so this output actually remains within the processor as a variable which is accessed by the position program.

Data processing is now complete, so the sensor program returns to the start to send out the next light pulse and take the next snapshot. In practice, a sampling frequency of 340 Hz was achieved when the sensor software was interleaved with the position program. This was acceptable since it was higher than the theoretical minimum of 250 Hz. Running sum filter characteristics were computed for a 340 Hz sampling frequency and a run-length of four chosen to give the frquency response of figure 5.21. Threshold values were derived empirically as described in Chapter 7, together with an analysis of sensor performance.

5.4 Summary.

The position sensor has been described from initial design to practical implementation. It can be thought of as a very low resolution 'camera' with a limited field of view, looking at the environment under infra-red illumination. It is programmed to recognise a number of unique markers and output the identity of every marker that comes into focus. Operation is under microprocessor control and is designed for vehicle speeds up to $1ms^{-1}$. Sensor construction is shown in Plate IV and its appearance in operation is illustrated by Plate V. An assembled listing of the sensor software as it appears within the complete position control program is included in Appendix B. The following chapter descibes the Navigation System and the role of the sensor within it.

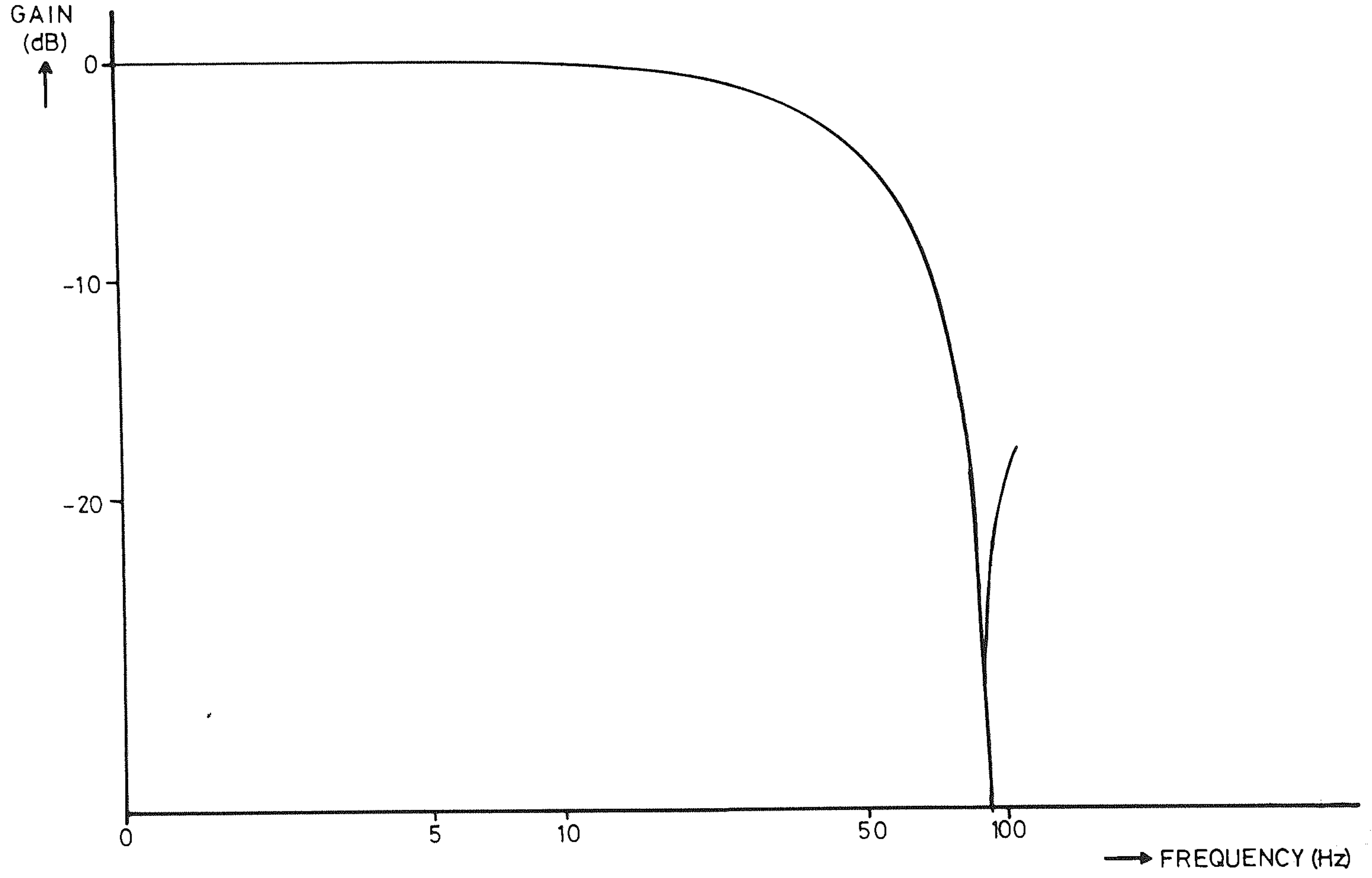


Figure 5.21 Frequency Response Of Running Sum Filter Employed In Sensor.

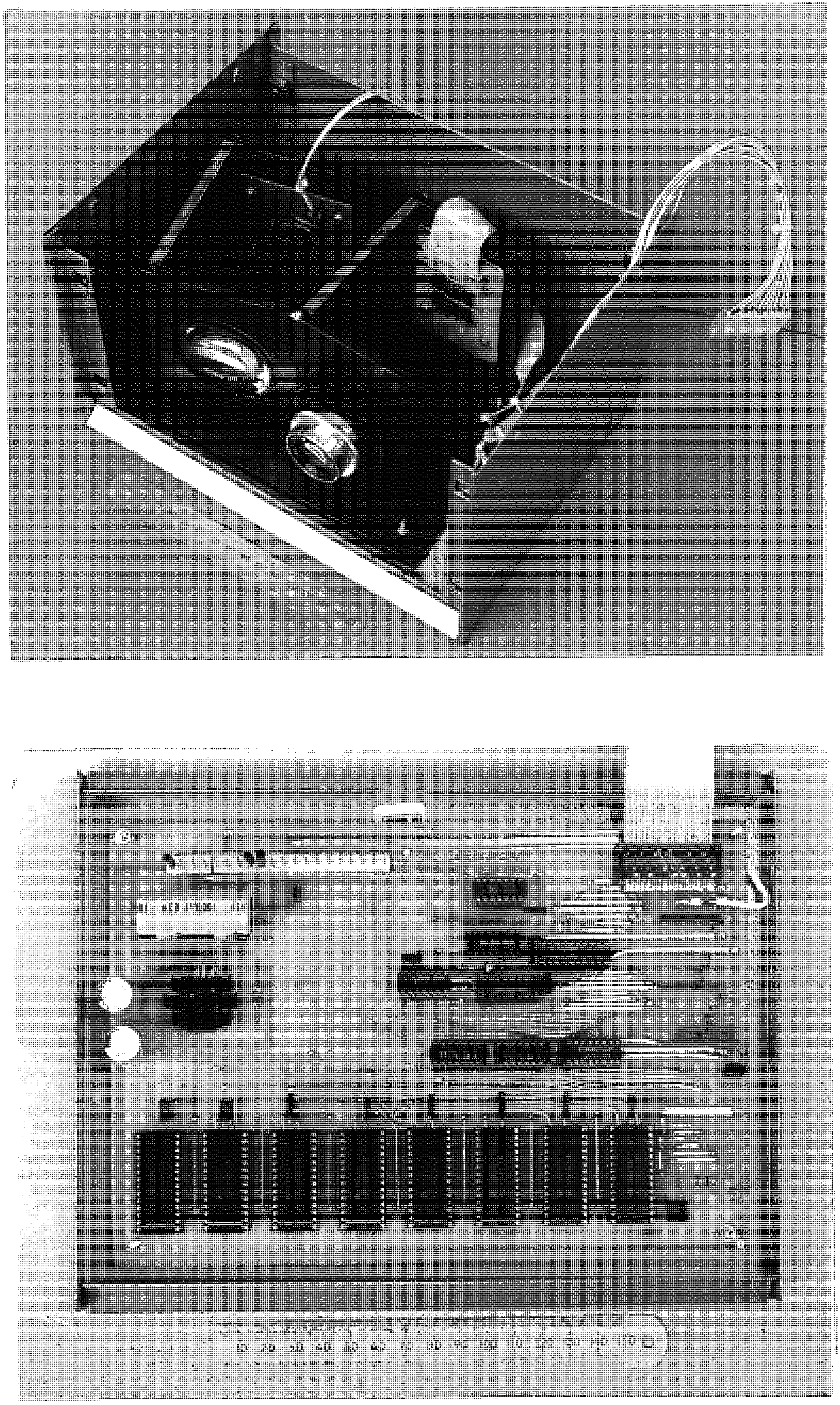


Plate IV The Position Sensor.

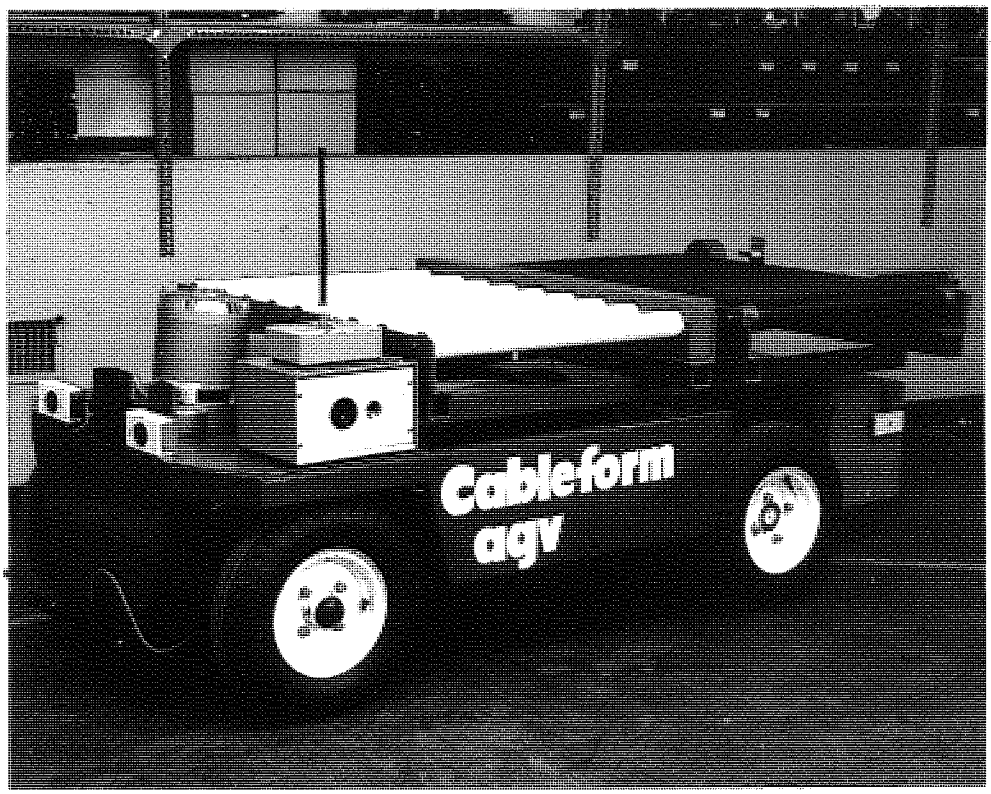


Plate V Prototype Vehicle With Position Sensor.

Chapter 6

THE NAVIGATION SYSTEM

The process of vehicle navigation employed is represented by the flow diagram of figure 6.1, illustrating the combination of position sensing, traction control and task breakdown required. (It is assumed here that whilst the vehicle is in motion, it is following its route). "Task Locations" correspond to location marker identities and the tasks themselves could include load/unload operations or entering a battery charging bay (both requiring the vehicle to stop) or changing between route guides. Implementation of this algorithm within the system architecture outlined in section 3.5 involved three microprocessors: Position, Traction and Vehicle Executive. This chapter describes how the programme of fig. 6.1 was distributed between processors, and also the communications protocols that were used.

6.1 Interprocessor Communications.[1]

Vehicle Executive is responsible for translating vehicle tasks into specific commands for each sub-system processor and co-ordinating execution of these commands to complete the task. It must therefore be able to communicate with each processor to send commands and receive status information, without impairing the operation of any sub-system. On the prototype, inter-processor communications were organized on an interrupt basis whereby program execution in the receiving device is suspended until the data has been transferred, after which the program is continued from the next instruction. If any part of the program must be processed without interruption, the processor's interrupt mechanism can be disabled during this portion. (When it is re-enabled, any pending interrupt is serviced immediately).

Each microprocessor sub-system consisted of a standard processor board

---

[1] Designed by Mr. R.H.Tilbury, Engineer, Cableform Ltd., 1982.

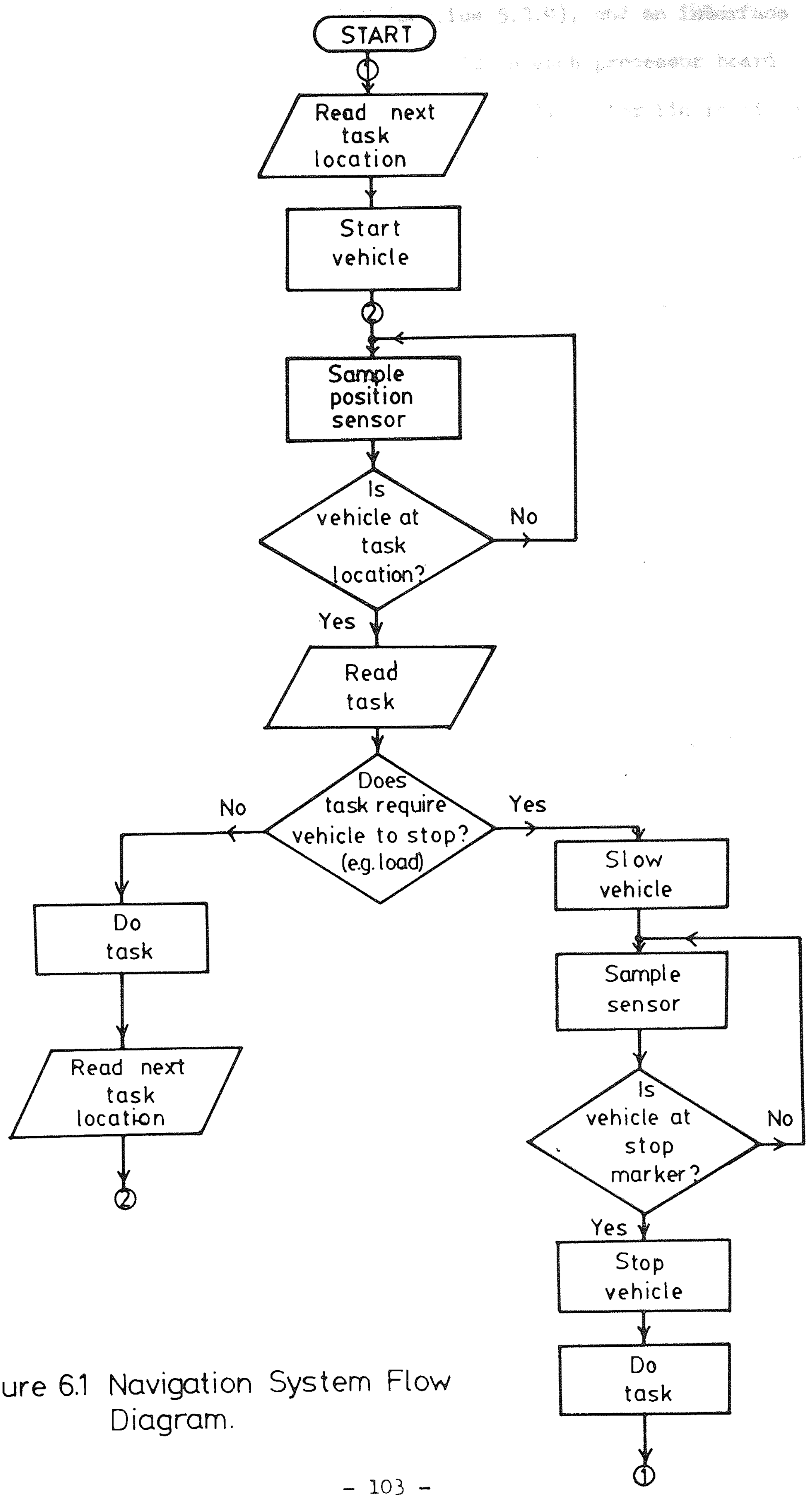


Figure 6.1 Navigation System Flow Diagram.

containing CPU, PIO, RAM AND EPROM (section 5.3.4), and an interface board for specialised input/output. The PIO on each processor board was dedicated to communications giving sixteen lines for bidirectional data transfer, organized into two eight-bit ports. Port A was reserved for operating the interrupt procedure, priority ranking and handshaking, and Port B was used for data. The amount of data to be transferred between any two processors varied, so a standard routine was devised whereby sixteen bytes were sent during each communication. Information was coded by position within this string as illustrated by the protocol for Position in figure 6.2. To avoid contention, all communications had to proceed via Vehicle Executive as in figure 6.3.

6.2 System Operation.

A block diagram of the position control system is shown in figure 6.4. The computation represented by ! in the diagram is shared between the Vehicle Executive and Position processors and operates in two distinct modes.

6.2.1 Navigating.

In this mode, Vehicle Executive instructs Position to monitor for location marker X, and Traction to proceed at velocity V. The vehicle operates at two set speeds: 1ms^{-1} (maximum) when following the guide-wire and moving between destinations, 0.3ms^{-1} when changing between guide wires. Position continuously samples the sensor and updates Executive with the current location each time a location marker is recognised. When location X is reached, Position stops sampling, tells Executive and awaits further instructions. At this point, Executive consults the task and decides whether or not the vehicle must stop. If not, it performs the task - for example instructs Steering and Traction to change wires at reduced speed - and informs Position of the next

INFORMATION FROM EXECUTIVE:

BYTE 1	(LOW ORDER)	TASK LOCATION
BYTE 2	(HIGH ORDER)	
BYTES 3 TO 16	SPARE (00H)	

INFORMATION TO EXECUTIVE:

BYTE 1

BIT 0 = AT REQUESTED LOCATION/$\overline{\text{NOT AT LOCATION}}$

BIT 1 = $\overline{\text{FAST}}$/SLOW

BIT 2 = FORWARD/$\overline{\text{REVERSE}}$

BIT 3 = $\overline{\text{IN MOTION}}$/LOST

BITS 4-7 = SPARE

BYTES 2-3 LAST LOCATION MARKER RECOGNIZED

(LOW BYTE FIRST)

0FFH = NO MARKER ENCOUNTERED SO FAR

BYTES 4-16 SPARE (00H)

Figure 6.2 Position Processor Communications Protocol.

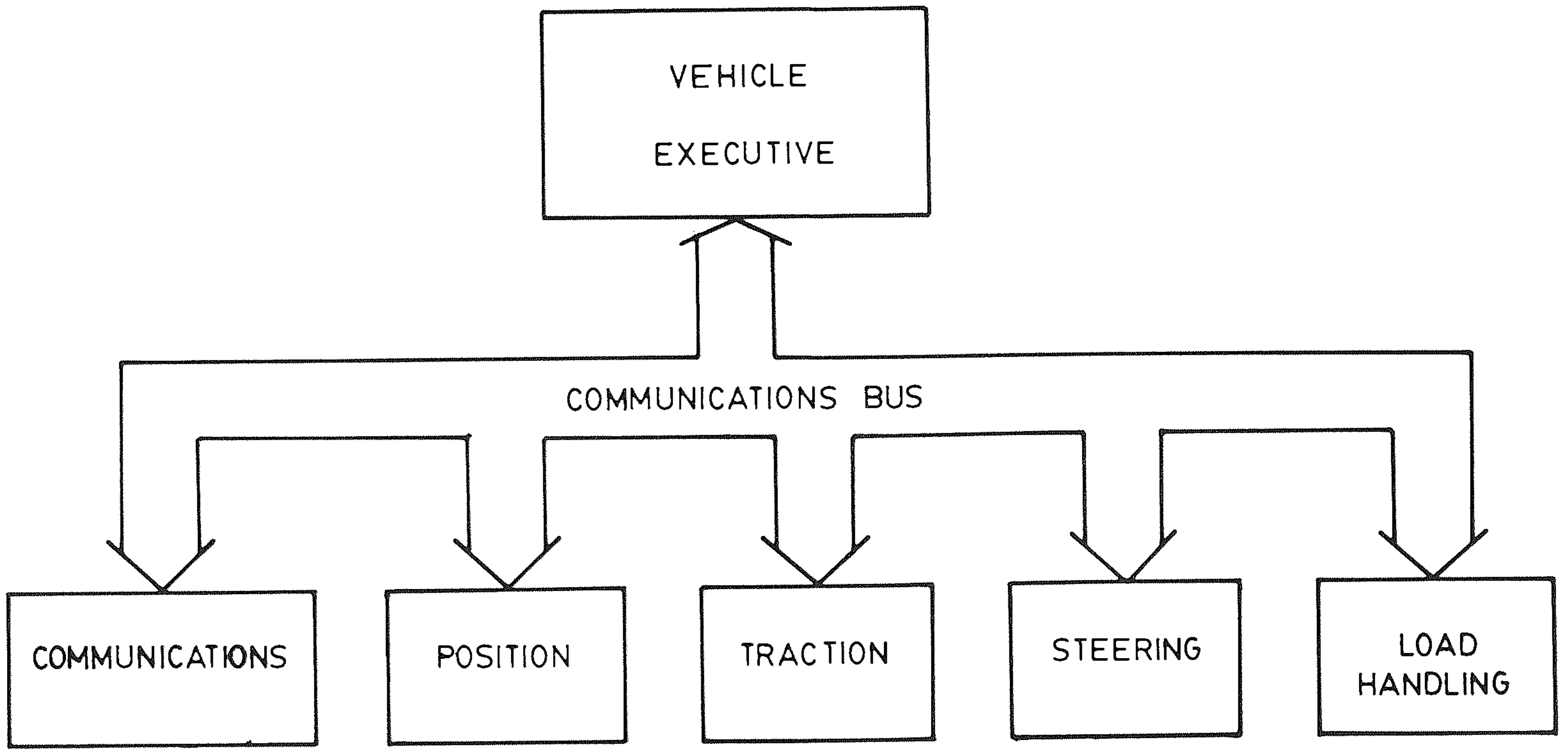


Figure 6.3 Interprocessor Communications Architecture.

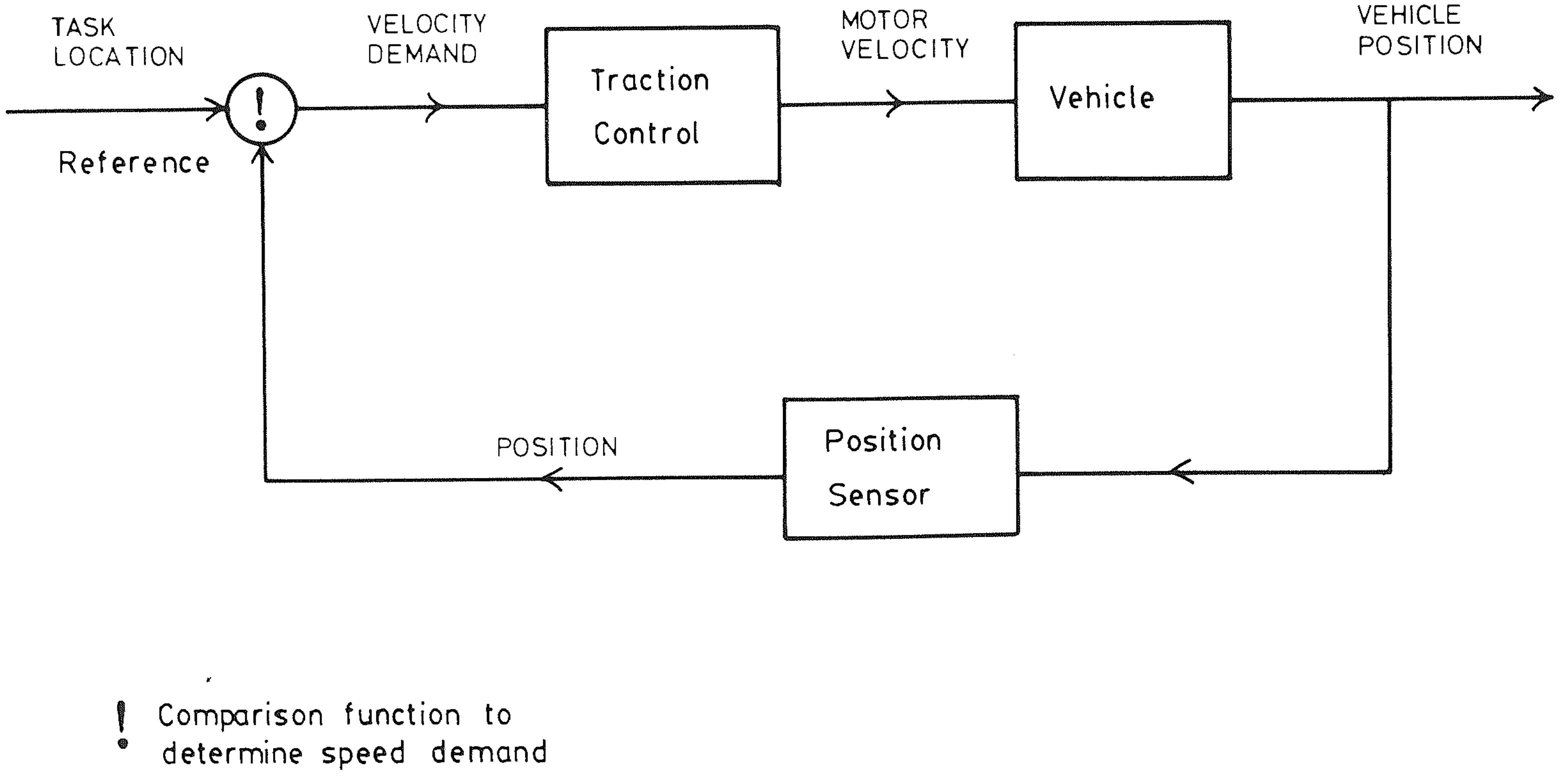


Figure 6.4 Position Control Block Diagram.

task location. The process is then repeated.

6.2.2 Positioning.

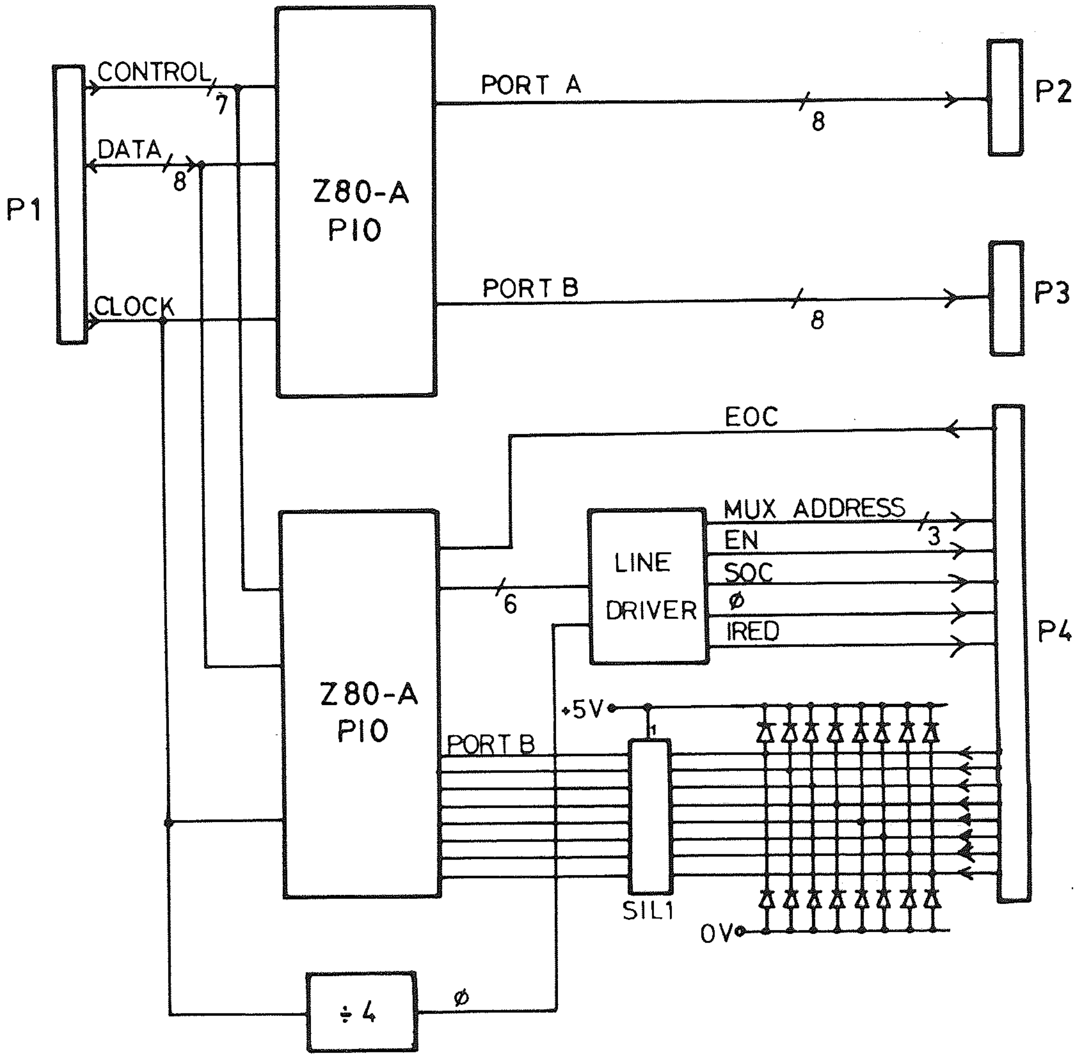
If the vehicle is required to stop - to unload at a workstation for example - positioning mode is entered. Vehicle Executive commands Position to stop and the latter takes control of the speed input to Traction. First, the speed demand is reduced to 0.2 ms^{-1} (minimum allowed), then sampling of the sensor is recommenced - this time looking for a stop marker. When this comes into view, the speed demand is set to zero and the vehicle is brought to a halt. Position informs Executive and retains speed control until given a new task location, when navigating mode is re-entered.

6.2.3 Interfacing.

Direct control of Traction by Position developed because the Executive was not present when initial testing of the sensor commenced. A link therefore had to be made between the two sub-systems, necessitating a further PIO on the Position interface board. One port was used with bit 0 for directing control to Position or Executive, and bits 1 to 7 for the speed demand. Port B was used to drive a single digit L.E.D. display as a debugging and commissioning aid: sensor output and error codes were displayed as a visual guide to performance and also to help when positioning markers. The unit could be unplugged and removed from the vehicle when trials were complete. The Position Interface circuit diagram is shown in figure 6.5.

6.2.4 Software.

The program for the Position microprocessor consists of two loops: one for navigating and one for positioning, and incorporates all the sensor software described in section 5.3.5. On power-up, the



P1 To CPU board

P2 To Traction interface board

P3 To LED display

P4 To Sensor

SIL1: R1-R8 270Ω

Figure 6.5 Position Processor Interface Board.

communications interrupt structure is organized and the PIOs programmed and set to initial conditions, with Position in control of the speed demand and the latter set to zero. The processor then interrupts Executive and waits for instructions. On receipt of the return interrupt from Vehicle Executive, program execution restarts, task instructions are stored and the navigating routine is entered.

All arrays used by the sensor software are cleared and the pointers initialized. The sensor program takes one sample from the transducer and tries to match the image with one of the marker templates held in store. If a true match is found, a flag is set and a pointer indicates the template. If there is no match, the flag is reset. This is common to both the navigating and positioning routines, but different template tables are used in each case, to reduce processing time to a minimum. For navigating, it is not necessary to recognize each view of a stop marker, so in this case, sensor initialization provides a table of all location markers and the central stop marker view with the associated marker identities. Once initialization is complete, the control line to Traction is set and Vehicle Executive informed that speed control has been relinquished. The program then enters a loop during which the sensor is sampled and the match flag tested. In the event of a true match, the loop is left and the marker identity is compared with the task location. If these two match, the program jumps to the positioning routine, otherwise Executive is informed of the latest position and sensor sampling recommences. Meanwhile, Vehicle Executive may have interrupted at any time during the above sequence, so every time the program loops round to re-sample, the current task instructions in store are compared with the contents of the 16-byte communications 'pigeon-hole' and updated as required. Figure 6.6a illustrates the navigating routine.

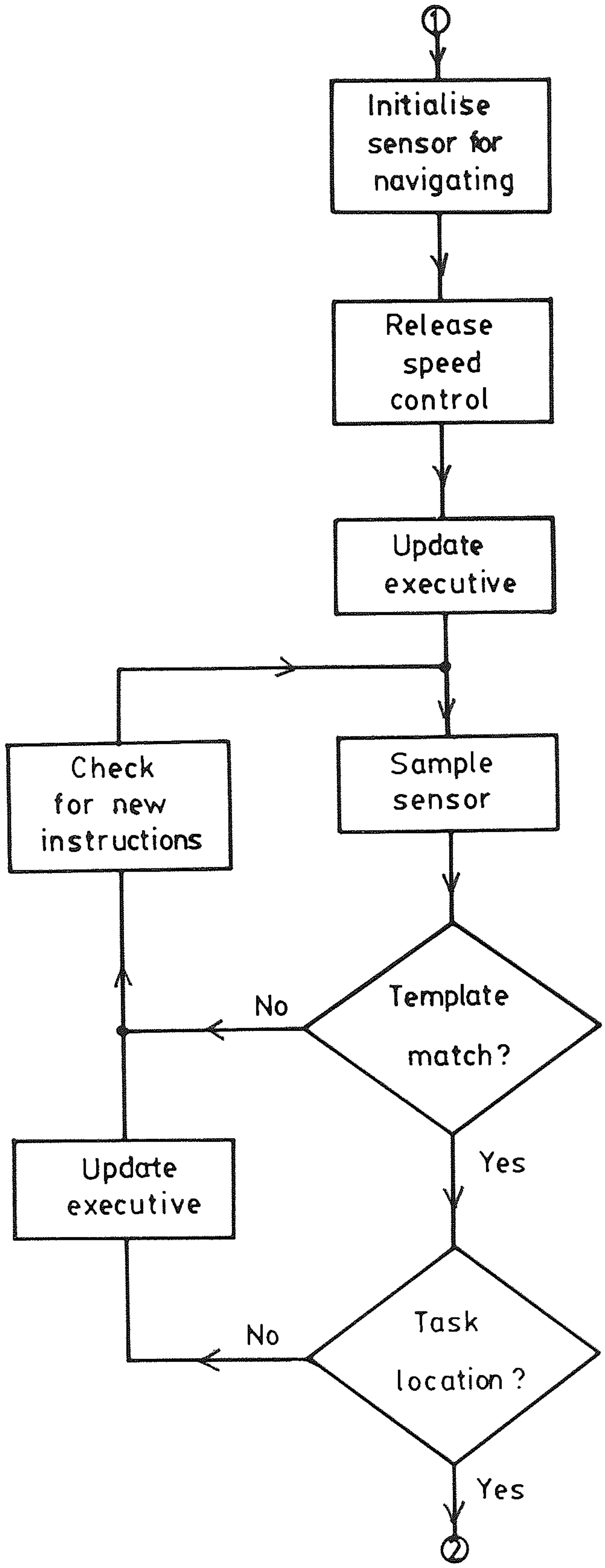


Figure 6.6a Navigating Routine.

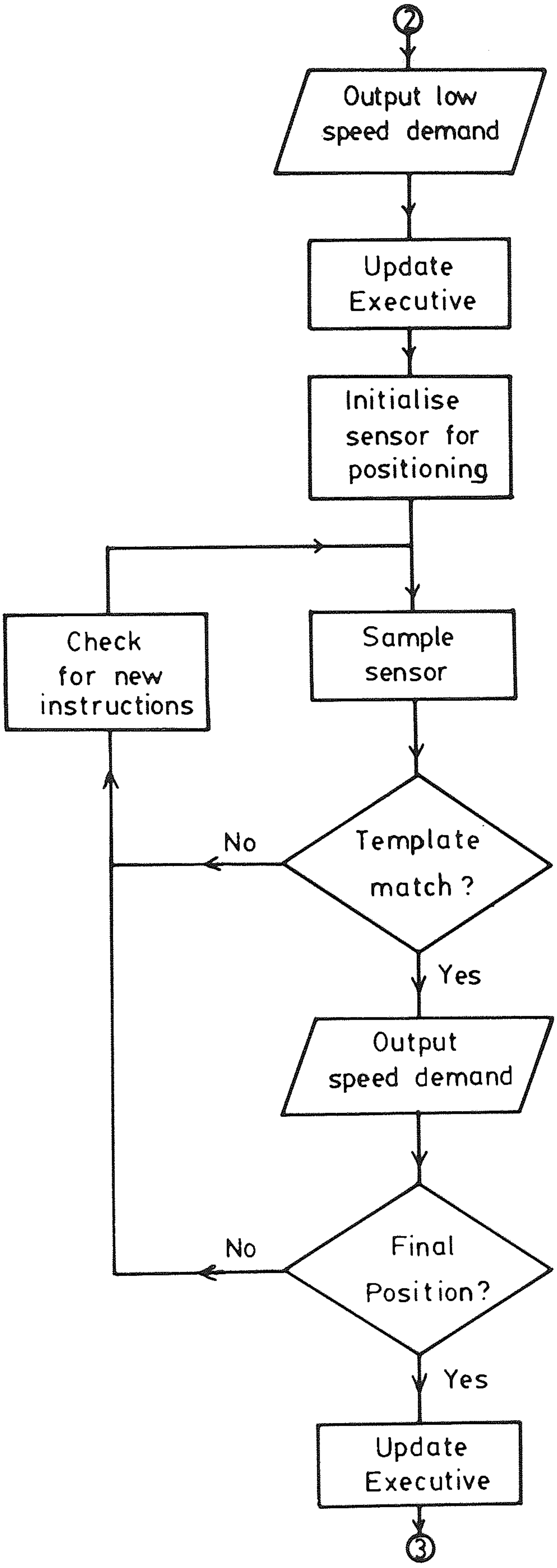
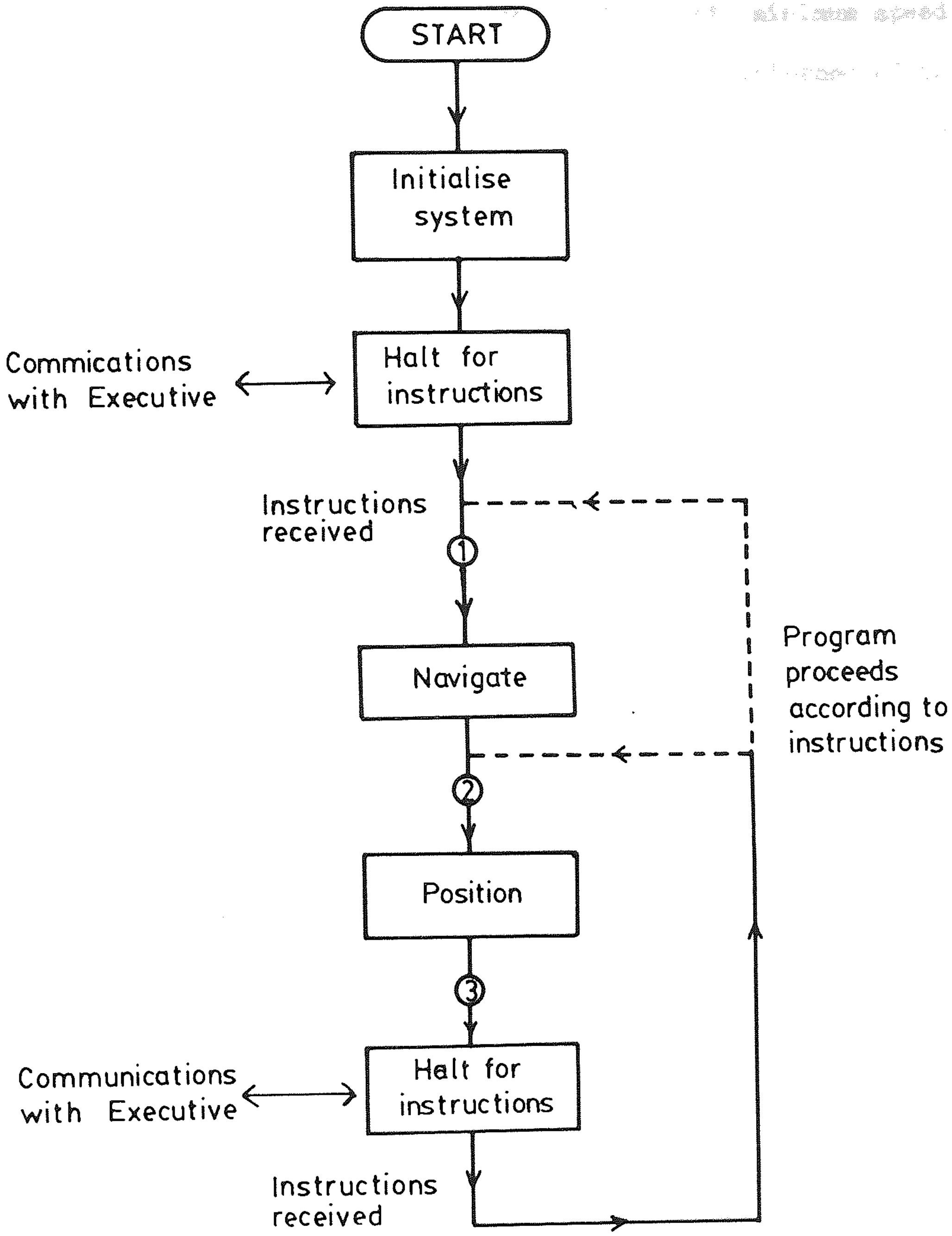


Figure 6.6 b Positioning Routine.



c) Controlling program

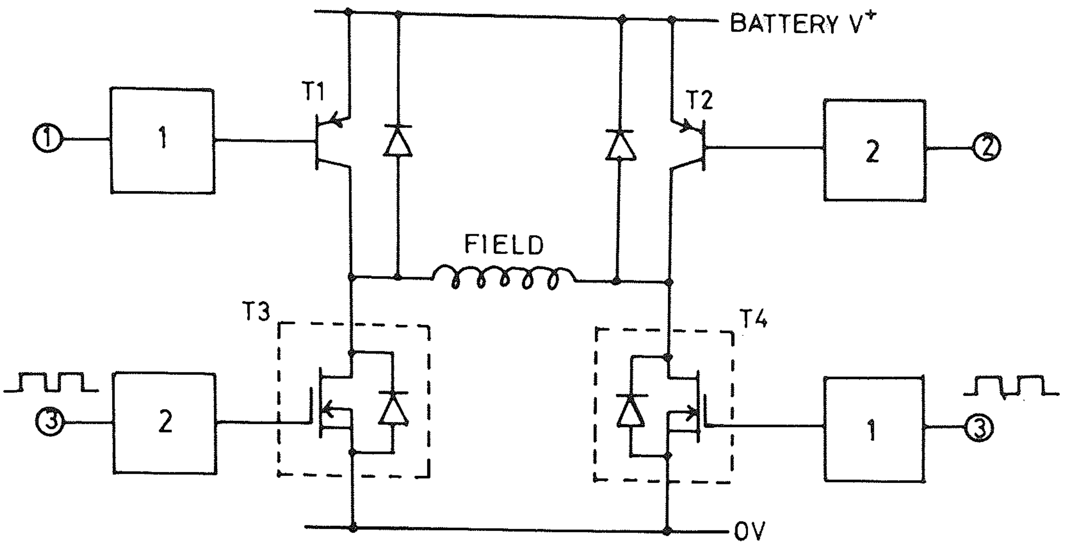
Figure 6.6 Position Processor Software.

The positioning routine is similar in operation and is shown in figure 6.6b. On entry, the control line to Traction is reset, minimum speed is requested via the data lines and Vehicle Executive informed of the current situation. The sensor software is then initialized as before, except that the template table consists of the seven stop marker views only. In place of marker identities, each template has an associated speed request, with zero speed for the central view. The sensor is sampled as before with the periodic check for new instructions from Executive. When a true match is found, the relevant speed request is output to Traction and sampling continued until the final position has been reached, when Vehicle Executive is informed of task completion and the processor halts until new instructions are received. The two routines are combined as in figure 6.6c to form the Position Control program, which occupied 1.2 kbytes of program store.

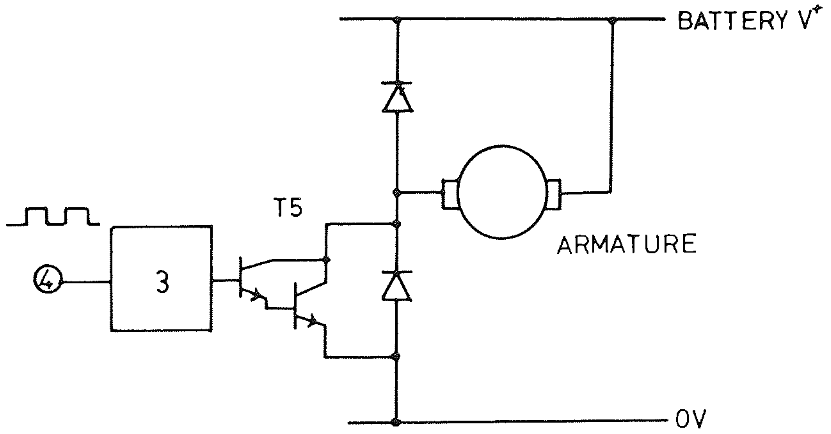
6.3 Traction.

The company already had experience of microprocessor based motor control and an existing design was fitted to the vehicle. This featured separately excited field and armature, and is shown in figure 6.7. Signals from the microprocessor are amplified by the three driver circuits to switch power transistors T_1 to T_5 . Motor direction is determined by the direction of current in the field and controlled by T_1 and T_2 . Motor current is pulse width modulated via T_3 and T_4 for the field and T_5 for the armature.

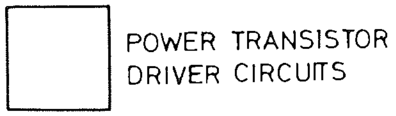
In operation, a constant current duty cycle was applied to the field, and armature control used to achieve the simple speed control system of figure 6.8 (43). Both field and armature were pulsed at a constant frequency of 500 Hz with 90% duty cycle on the field and 0 to 80% (variable in 1% steps) on the armature. Driver circuits 1 and 2 are



a) Field circuit



b) Armature circuit



MICROPROCESSOR SIGNALS:

① FORWARD ENABLE
② REVERSE ENABLE
③ FIELD DUTY CYCLE
④ ARMATURE DUTY CYCLE

Figure 6.7 Traction Motor Controller[1].

---

[1] Designed by Mr. C.E. Watson, engineer, Cableform Ltd., 1980.

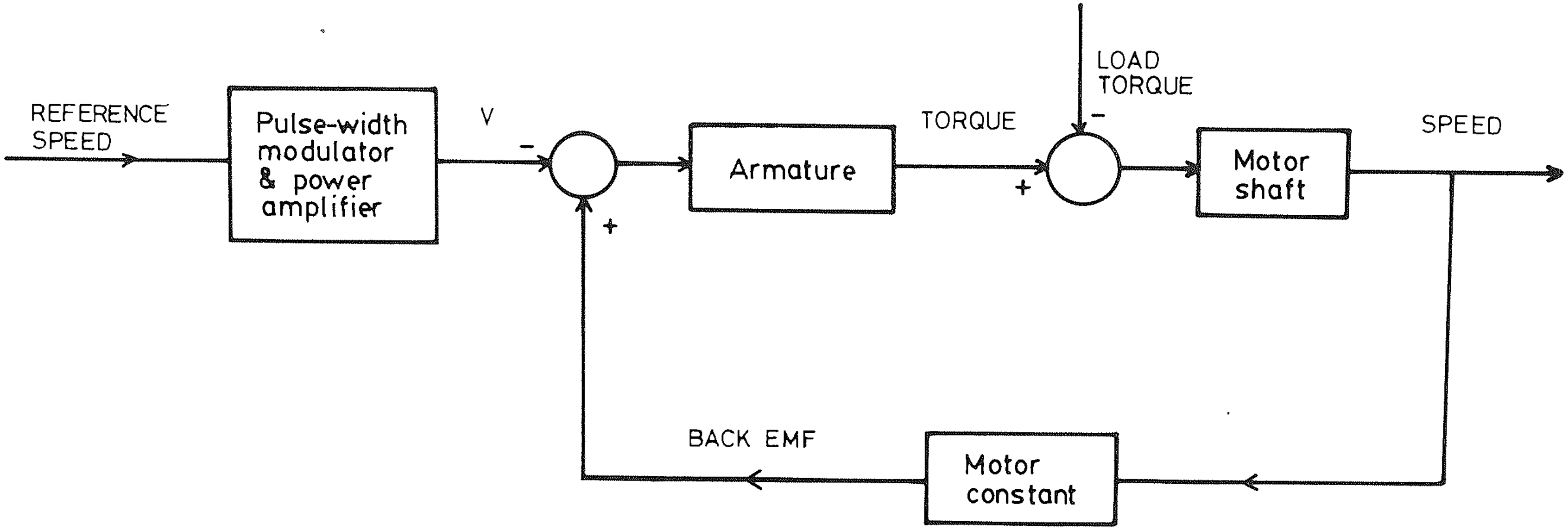


Figure 6.8 Traction Motor Speed Control.

interlocked such that

$$(T_1 \text{ is on AND } T_4 \text{ is pulsed}) \text{XOR } (T_2 \text{ is on AND } T_3 \text{ is pulsed}) \quad \text{Eq.6.1}$$

according to the state of the Forward/Reverse signal from the microprocessor.

Pulse width modulation was controlled by the processor through the Z-80 Counter-Timer peripheral in the circuit of figure 6.9. This device can be programmed to continually count pulses of an external trigger signal and give a single high pulse every time the desired count is reached. Counting then restarts from the next trigger pulse. There are three channels which can be individually programmed and triggered. The pulse width modulator uses one channel and a dual decade counter to divide the system clock and produce the constant frequency duty cycle envelope. The other two channels count pulses that occur 100 times faster than this frequency so that the counter value programmed corresponds to the percentage duty cycle required.

Forward and Reverse enable signals were derived from a single bit of a PIO port and armature and field disable lines provided as in figure 6.10. Radio controlled stop/start override was provided for safety during tests and this was monitored via the PIO. The normally open trip relay was used to open the main battery line contactor to the motors in the event of microprocessor or program failure.

6.3.1 Software.

Figure 6.11 shows a flow diagram for the speed control programme. After the PIO and counter/timer controller have been programmed and initialized, the main program loop proceeds as follows: the speed demand is read and compared with the current speed being output.

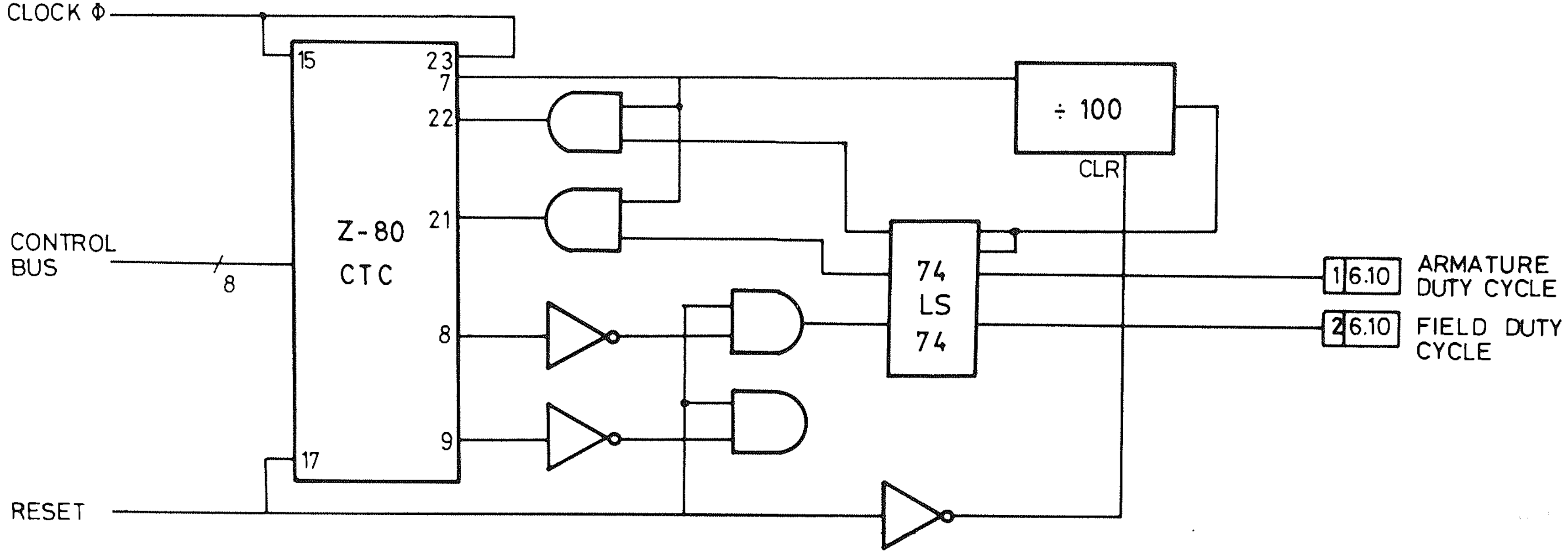


Figure 6.9 Pulse Width Modulator Circuit[1]

[1] Designed by Mr. R.H.Tilbury, engineer, Cableform Ltd. 1982.

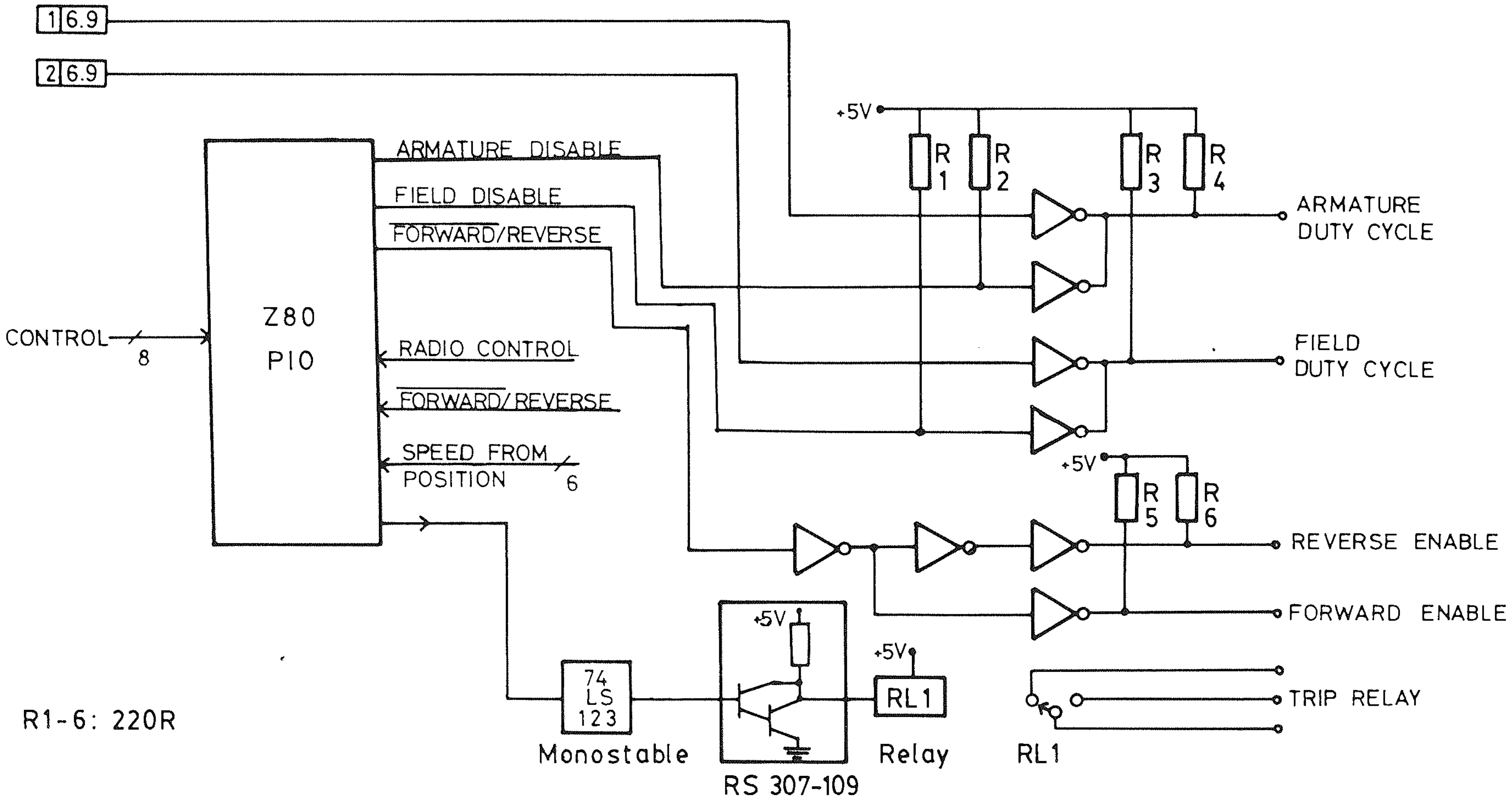


Figure 6.10 Traction Interface[1]

[1] Designed by Mr. R.H. Tilbury, engineer, Cableform Ltd. 1982.

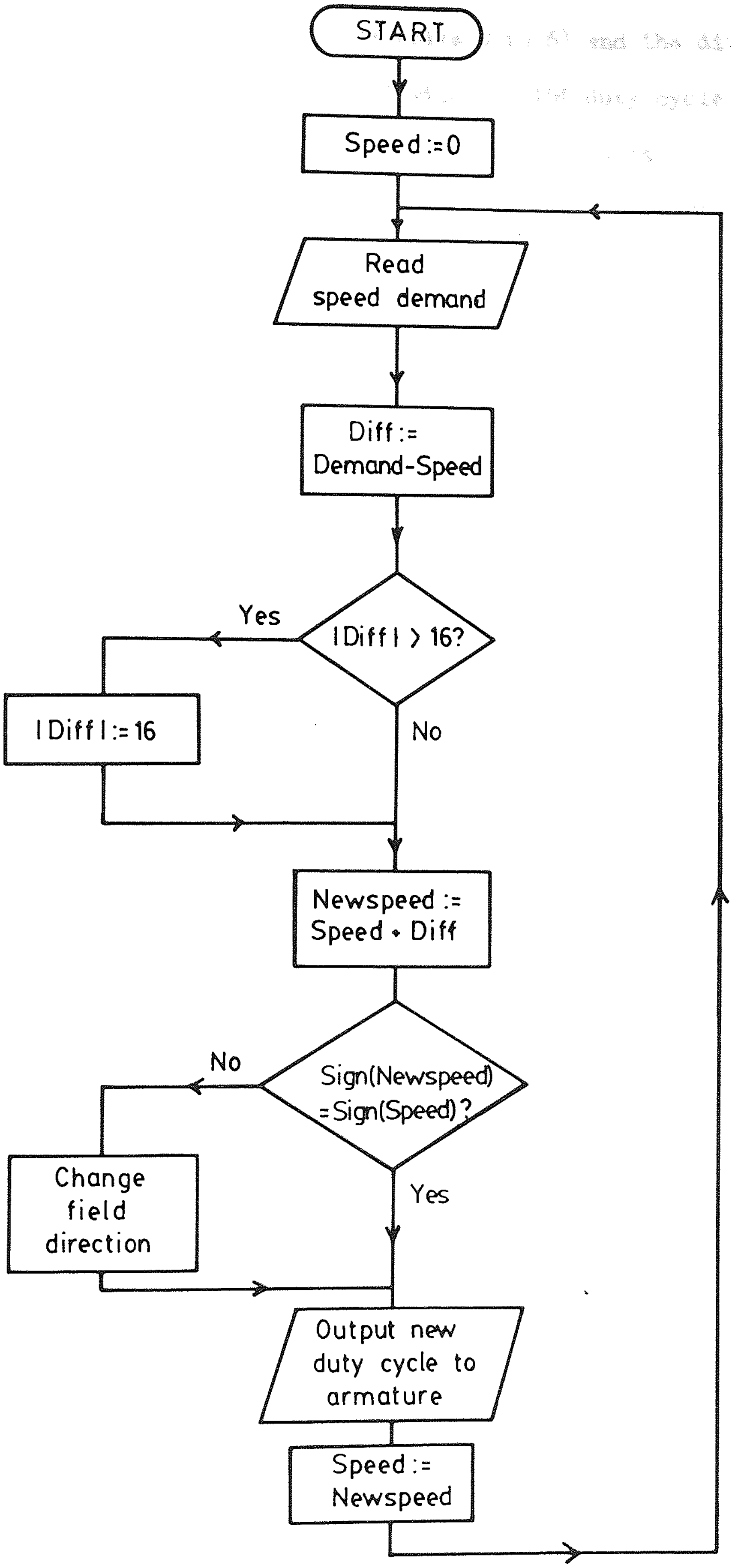


Figure 6.11 Traction Processor Software.

This speed measure is a two's complement number formed from the armature control signal duty cycle (bits 0 to 6) and the direction enable (bit 7). Speed changes are limited to 16% duty cycle steps to produce smooth acceleration, and the amended demand is compared with present speed to see whether a direction change is required. If so, armature and field outputs are disabled and a 100 ms delay is called to allow the field current to decay. Then the direction enable is complemented and output on the PIO and the field duty cycle re-enabled. A further 100 ms delay is given to allow the field current to build up, after which the programme returns to its original course where bit 7 of the speed demand is reset and this word loaded into the armature channel of the counter/timer to form the new duty cycle. The armature output is then enabled, the variable holding the current speed value is updated and the programme loops back to read the reference again. If the speed change had been limited, a 200 ms delay is invoked before re-starting the loop.

As already mentioned, the reference has two sources. In addition, a manual over-ride facility was provided whereby vehicle speed and steering could be controlled via a joystick. Information from Vehicle Executive consists of two bytes as follows:

Byte 1:	bit 7	$\overline{\text{Forward}}$/Reverse
	bits 1-6	Speed if in manual
	bit 0	Manual/$\overline{\text{Auto operation}}$
Byte 2:	2's complement speed demand if in auto.	

If byte 2 is zero, Traction must perform an emergency stop and wait for further instructions. When in manual operation, the speed demand variable is simply loaded with byte 1. Information from Position is received via port B of the PIO:

bit 6	$\overline{\text{Priority to Position}}$/Priority to Executive
bit 5	$\overline{\text{Forward}}$/Reverse
bits 0-4	Speed if priority to Position

Position only takes control during slow speed manoeuvring so in this case, the speed demand is formed as follows:

bit 7	$\overline{\text{Forward}}$/Reverse from Position
bit 6	Zero
bit 5	Zero
bits 0-4	Speed from Position

During the routine to find the speed demand, the radio control signal is read and the vehicle stopped if this line is high. The program then loops round this routine until the r/c input goes low, when execution continues as before.

The trip relay is operated via a monostable which is continually retriggered by pulses from the processor sent at frequent intervals during the program. If the program halts for any reason, these pulses will stop, the monostable times out and the relay opens.

Traction occupies 925 bytes of program store.

6.3.2 Operation.

In automatic mode, the vehicle was driven at fixed speeds according to its task as illustrated in the typical velocity profile of a loading operation shown in figure 6.12. For demonstration purposes, positioning

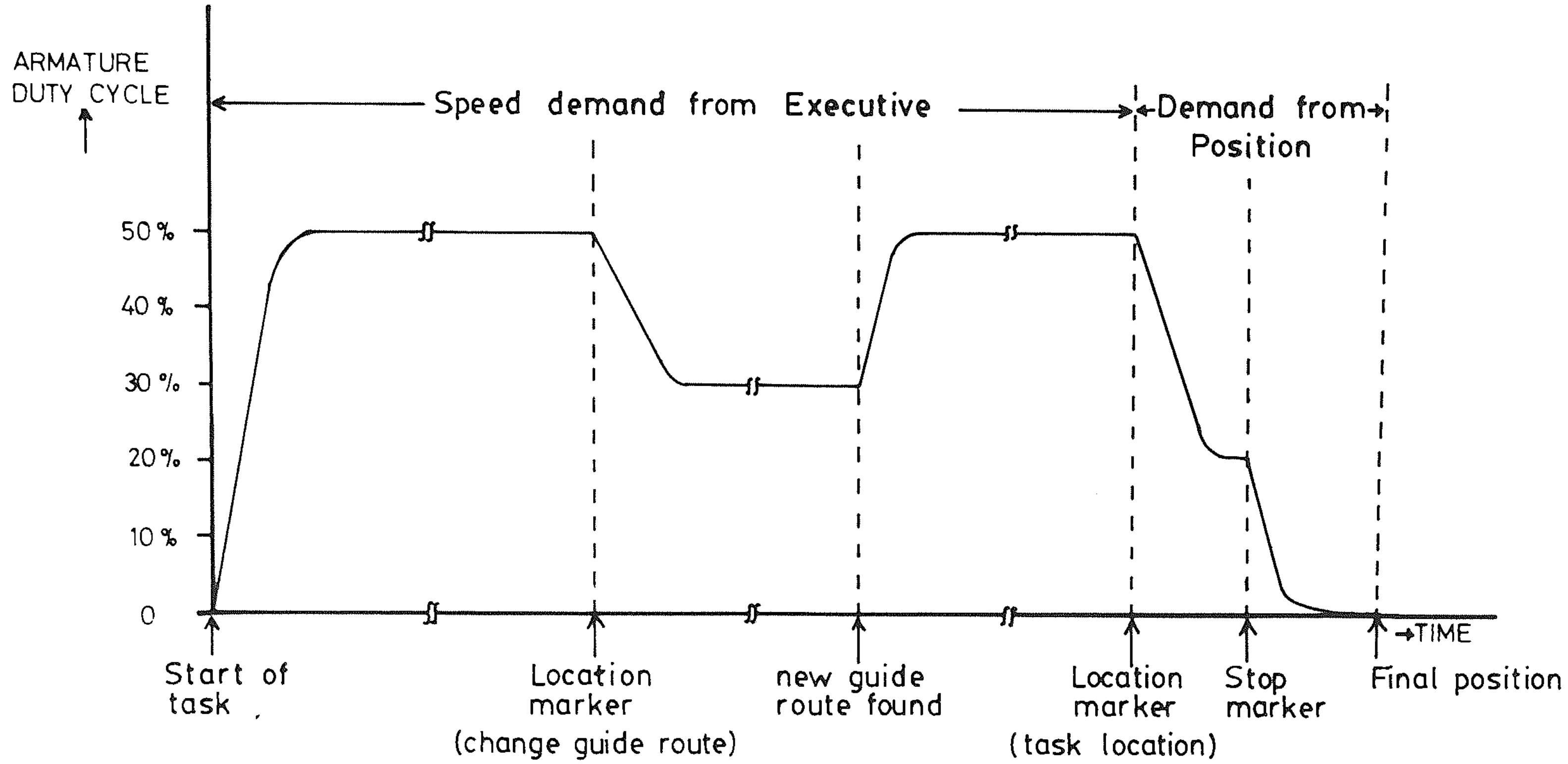


Figure 6.12 Speed Control For A Typical Task.

was achieved by assigning speed demands of 20% duty cycle (minimum speed) to each partial view of a stop marker and zero for the central view. Once the central view had been recognized, sensor sampling was stopped. This meant that the vehicle came to rest a short distance in front of the stop marker, but the positioning was sufficiently repeatable with light loads to allow successful demonstration of automatic loading at stationary conveyors. These workstations were operated by the vehicle via an ultrasonic link which was only activated when the vehicle was correctly aligned.

The position sensor was mounted on one side of the vehicle in line with the steering pivot so that deviations from the guide-wire would have the minimum effect. Markers were mounted on vertical stands along the route and used to identify workstations and points where the vehicle had to change between guide-wires. Location and stop markers were placed 1 m apart (minimum) to allow sufficient distance for the vehicle to slow down. The vehicle was equipped with two conveyors and both could be used at one workstation by using two stop markers. Position was instructed to stop at either the first or second stop marker encountered after the location marker, or even to stop at the first, transfer a load and then proceed at low speed to the second.

Chapter 7

EVALUATION

Testing of the navigation system took place in two phases: investigation of sensor performance and demonstration of the prototype vehicle.

7.1 Sensor Evaluation.

It soon became apparent that the vehicle could not be used as a test-bed for the sensor, since there were too many variables in the system. The main problem was positioning markers relative to the sensor since it was not clear whether recognition failures when the vehicle was in motion were due to the sensor or to the vehicle wandering about the guide-wire. In addition, the prototype Route Following system had a maximum operational speed of $0.7ms^{-1}$, so Position could not be tested to its full specification in this way. A test-rig was therefore devised where speed and relative position could be controlled independently, and the sensor operated in isolation of other vehicle sub-systems.

7.1.1 Test Equipment.

Figure 7.1 shows the experimental arrangement for sensor tests. From the sensor's frame of reference, there is no difference between the transducer moving past stationary markers and a fixed transducer viewing moving markers. From an experimental point of view, it is far simpler to move the markers. The latter were placed on a belt passing over two spools, one of which was driven by an electric motor. Belt speed could be varied and was measured by a sensor and shaft encoder on the motor, outputting via an l.e.d. display. The motor speed control unit was adapted from some existing equipment and had to be calibrated by timing the belt over several revolutions at different speeds. Available speed range was 0 to $3ms^{-1}$ and the display resolution was $0.02ms^{-1}$.

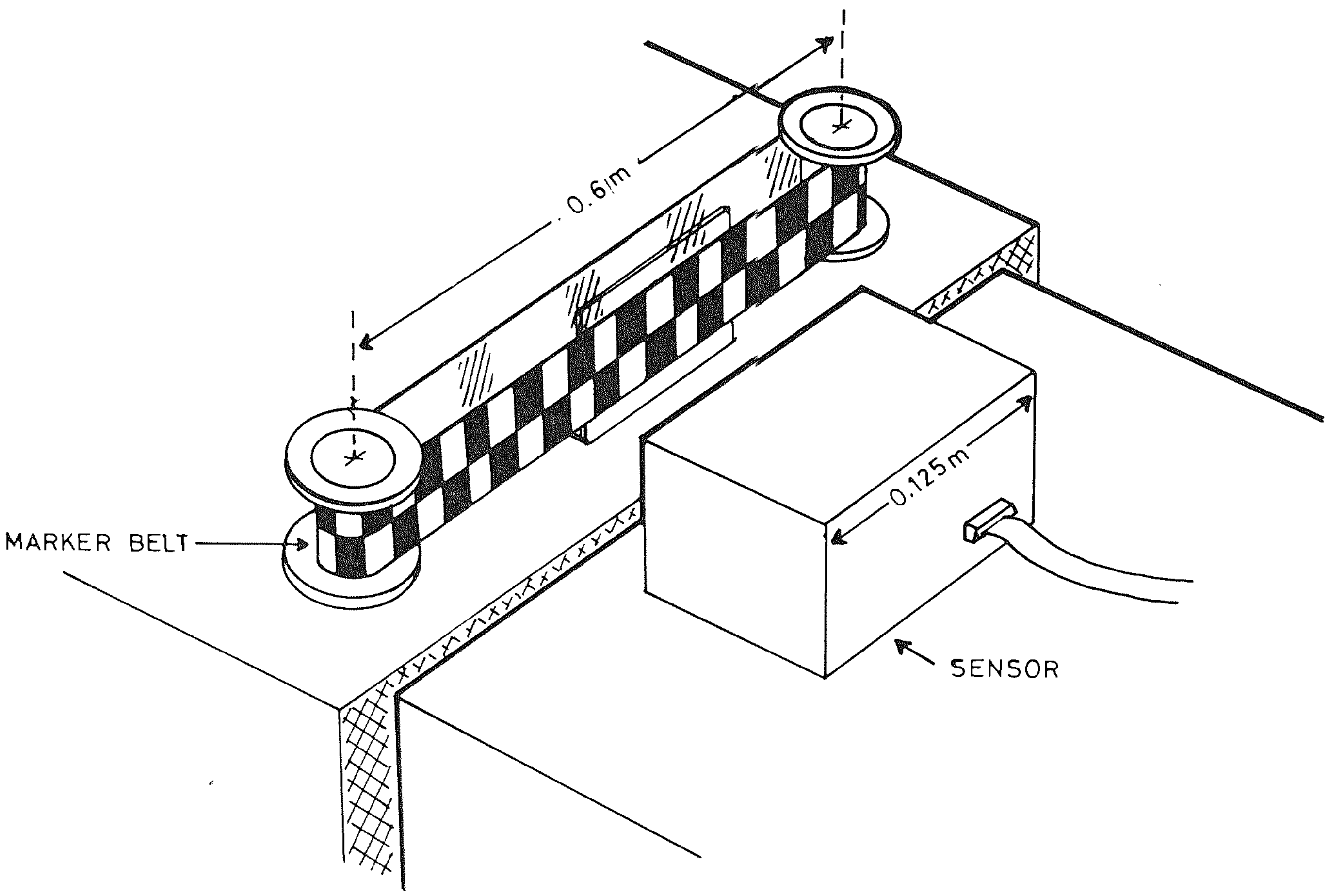


Figure 7.1 Experimental Arrangement For Sensor Evaluation.

One aim of the tests was to investigate suitable threshold levels, so the output of the running sum filter had to be recorded in some way. Under normal operation, this output would be temporarily stored in on-board RAM for further processing and replaced by each new sample. It was therefore decided to implement the program in the Nascom 2 development microcomputer and store this information in computer RAM so that it could be processed after completion of each test. Figure 7.2 illustrates the link between computer and microprocessor system and the division of programming tasks between the two. The PIO which normally interfaced Position to Traction was used as a parallel link to the computer, with port A for data transfer and port B for handshaking. The program held in sub-system PROM provided all the control signals to the transducer and read the photodiode amplifier outputs. Under control of the Nascom 2, this data was transferred to the computer where the rest of the data processing software resided. Data could thus be stored at any stage of processing for a very large number of samples. Great care was taken to ensure that the data transfer procedure and any storage routines did not significantly alter the sampling frequency of the sensor.

In operation, the sensor was placed in front of the marker belt, on a separate support to isolate it from vibrations due to the motor drive. Vertical height relative to the drive belt could be varied as could distance from and incident angle to the belt.

7.1.2 Experiments.

These fell into two categories: measuring the filtered detector signals under various conditions, and logging the number of times a true match occurred as markers passed the sensor. The former set of tests were designed to investigate sensor response and formulation of suitable

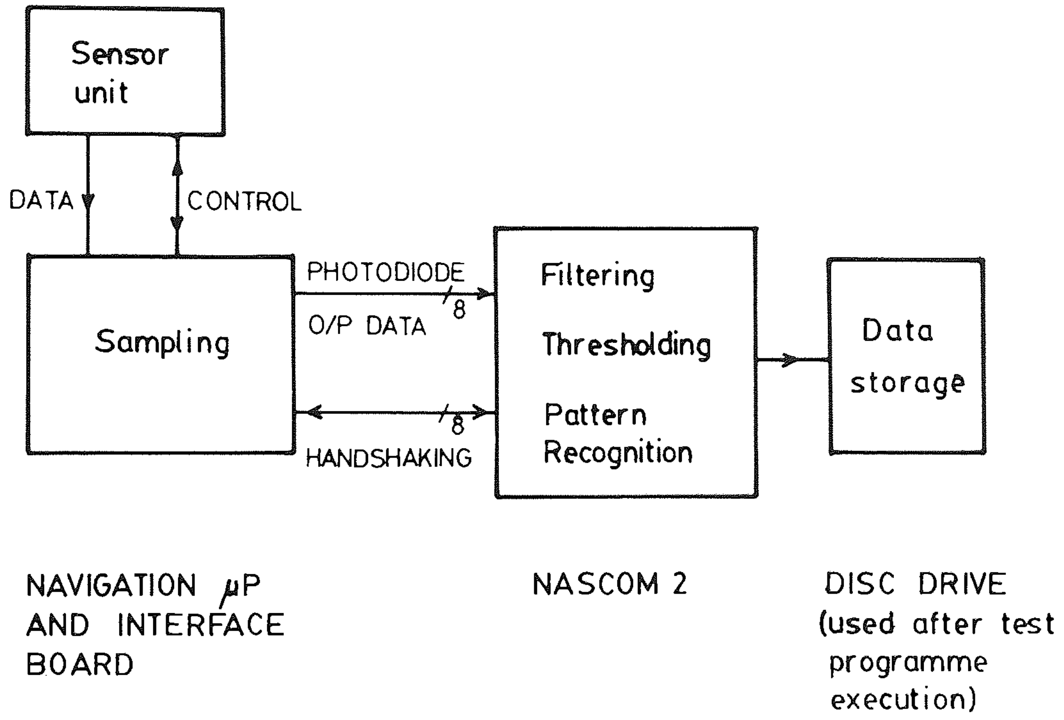
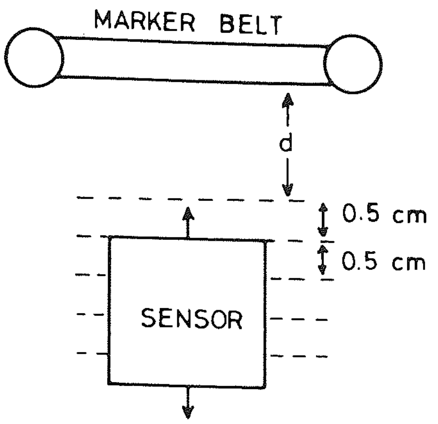


Figure 7.2 Implementation Of Sensor Test Software.

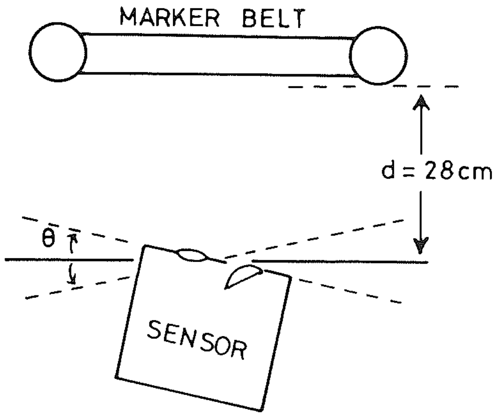
thresholds, and the latter set were for ensuring the sensor met its specification.

The highest spatial frequency that had to be reliably sensed corresponded to alternating black and white elements as in a number 2 location marker. A belt was made up with this pattern along its entire length as in figure 7.1 and driven at the maximum speed of $1ms^{-1}$. Data for each detector was logged at the output of the running sum filter for 256 samples during each test run. The sensor was placed in front of the belt such that the 'marker' was in focus and correctly aligned, and data collected for distances between ±2cm from this point at 0.5cm intervals. The sensor was then returned to its 'correct' position, and its angle to the belt varied. Finally, tests were carried out at various relative heights (figure 7.3). Each set of tests was repeated several times to allow for errors in sensor positioning.

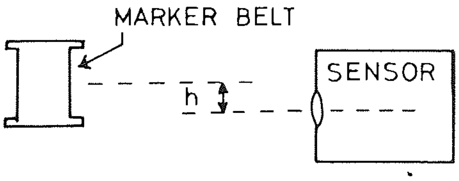
When the above data had been analyzed and thresholds chosen, the belt was replaced with one having individual patterns as in figure 7.4. Six patterns were used corresponding to the four location markers, a stop marker and an all-white marker. This latter pattern was included as a control since its low spatial frequency ensures recognition over greater ranges of speed, angle and distance than the other markers. The intervening areas of belt were matt black. For these tests the sensor program was given a template table containing the six markers and the identity of every true match was logged for 1,024 matches. Distance and angle tests were performed as before, to ensure the sensor met its specification, then the sensor was placed in its 'correct' position and recognitions logged at various speeds.



a) Distance tests



b) Angle tests



c) Height tests

Figure 7.3 Sensor Evaluation.

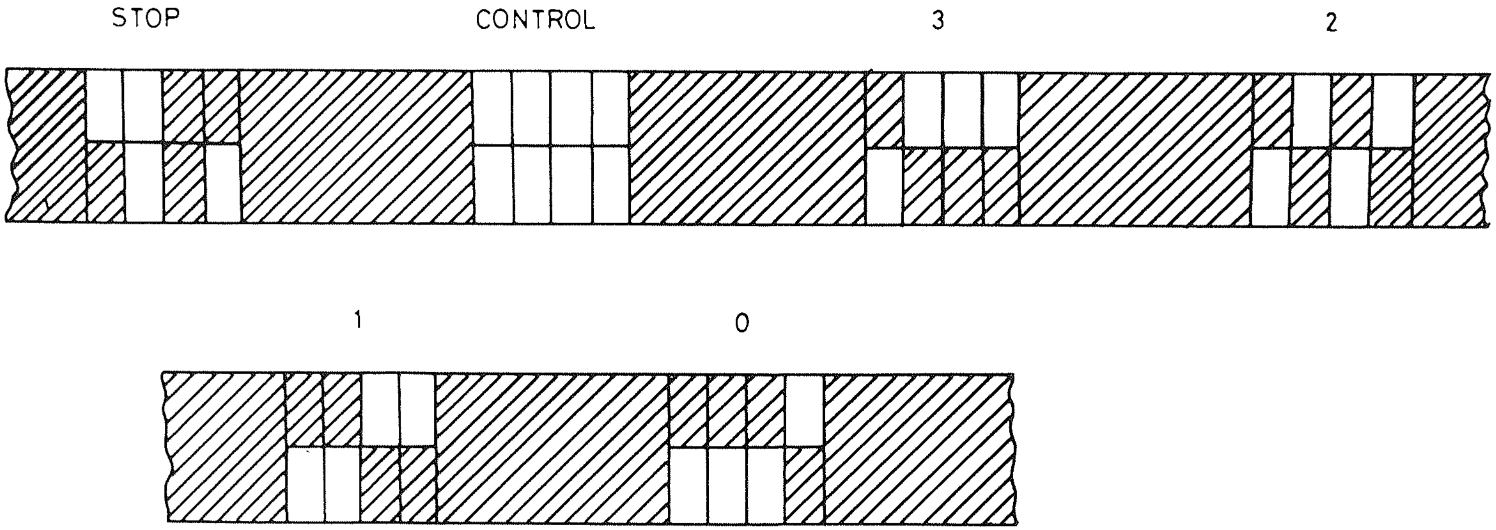


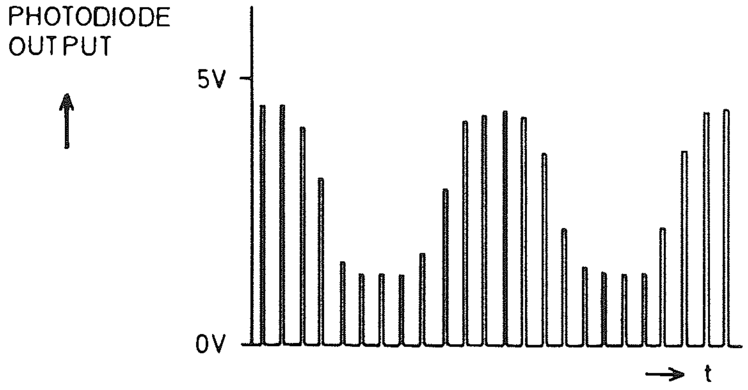
Figure 7.4 Marker Recognition Test Belt.

7.1.3 Thresholding.

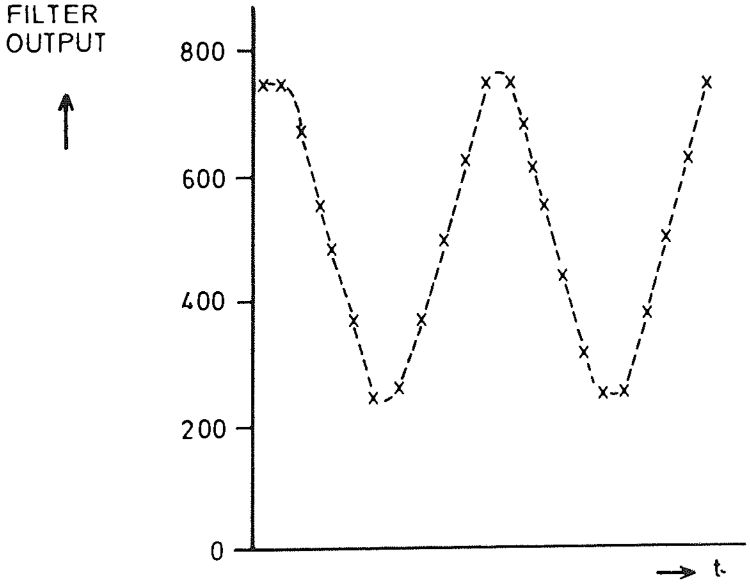
The continous pattern belt gave a photodiode amplifier output of the form shown in figure 7.5a, with the corresponding filtered output of figure 7.5b. The peaks correspond to white elements filling the field of view, and the troughs to black elements. Figure 7.6 shows how the averages of the peaks varies with distance for each detector in the mark 1 sensor, where each detector's distribution has been normalised about its focussed reading at 28cm (this distance was measured from the front plane of the sensor unit and corresponds to 30cm from the detector lens focal plane). Under plane illumination, the curves for all detectors would be expected to be similar with decreasing intensity at larger distances, and perhaps some fall-off in intensity towards the left of the image (detectors 4 and 8) due to the angled source. Given that the mark 1 sensor was only an approximation to plane illumination, deviations from the above prediction were accepted, but vertically adjacent detectors (1 and 5 or 3 and 7 for example) were expected to exhibit similar distributions. That they did not serves to illustrate the great variation in device performance. Since each photodiode amplifier gain had to be individually adjusted to prevent saturation, differences in response across the image due to the illumination could not be isolated. Figure 7.7 shows the normalised mean minima for the same sensor, which mainly follow similar distributions to the maxima.

The mark 2 sensor with its source of eight focussed emitters, generally exhibited more pronounced variations across the depth of field (figures 7.8 and 7.9). The response of detector 7 proved to be a limiting factor in the ability of this sensor to meet the specification.

As a result of the wide variation in response across the specified operating range of 28 ±1cm, together with amplitude differences



a) Photodiode amplifier outputs
(taken from an oscilloscope trace).



b) Running sum filter output
(summed over 4 samples).

Figure 7.5 Continuos Pattern Responses.

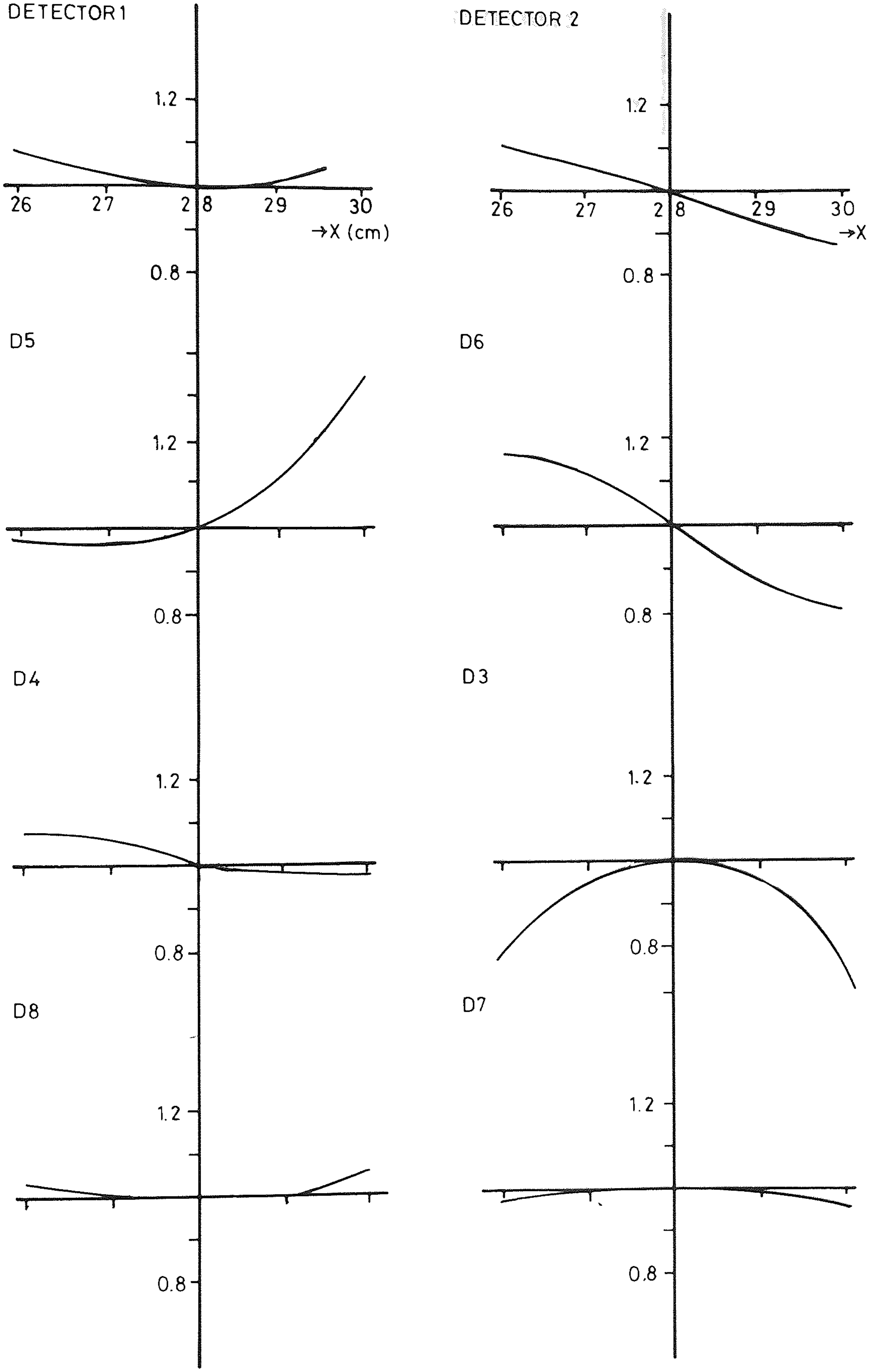


Figure 7.6 Normalised Mean Maxima For Sensor 1 Distance Tests.

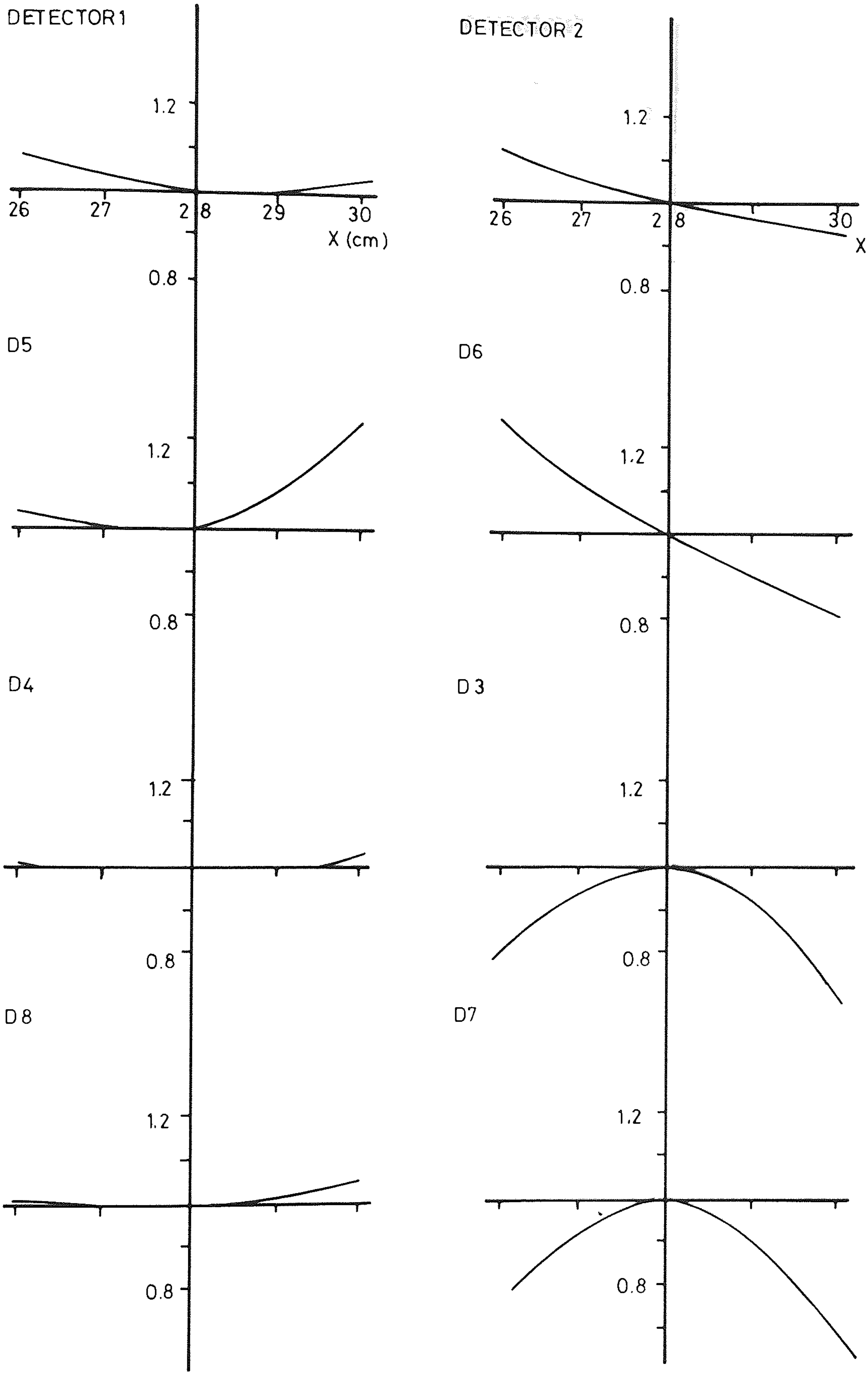


Figure 7.7 Normalised Mean Minima For Sensor 1 Distance Tests.

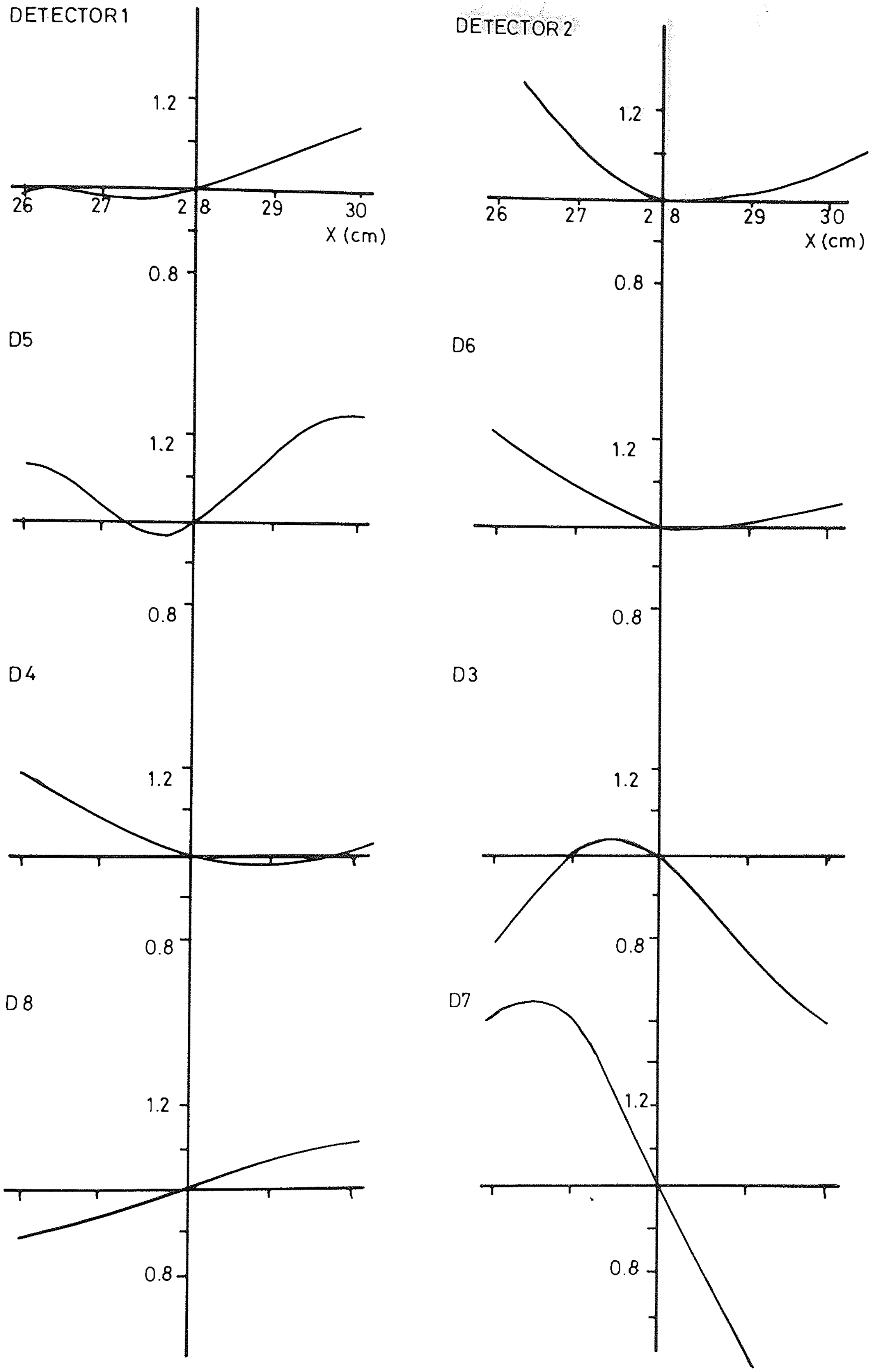


Figure 7.8 Normalised Mean Maxima For Sensor 2 Distance Tests.

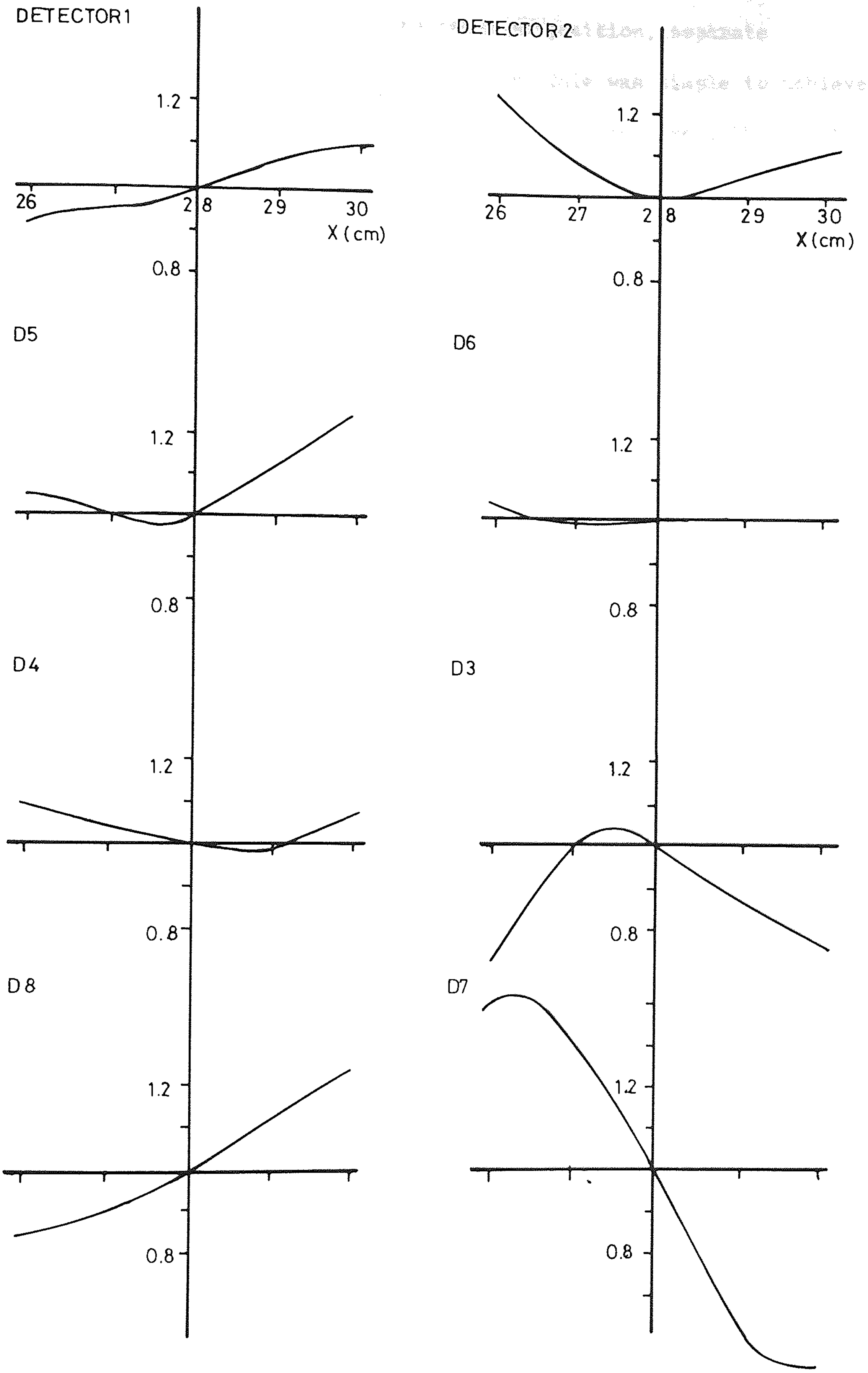
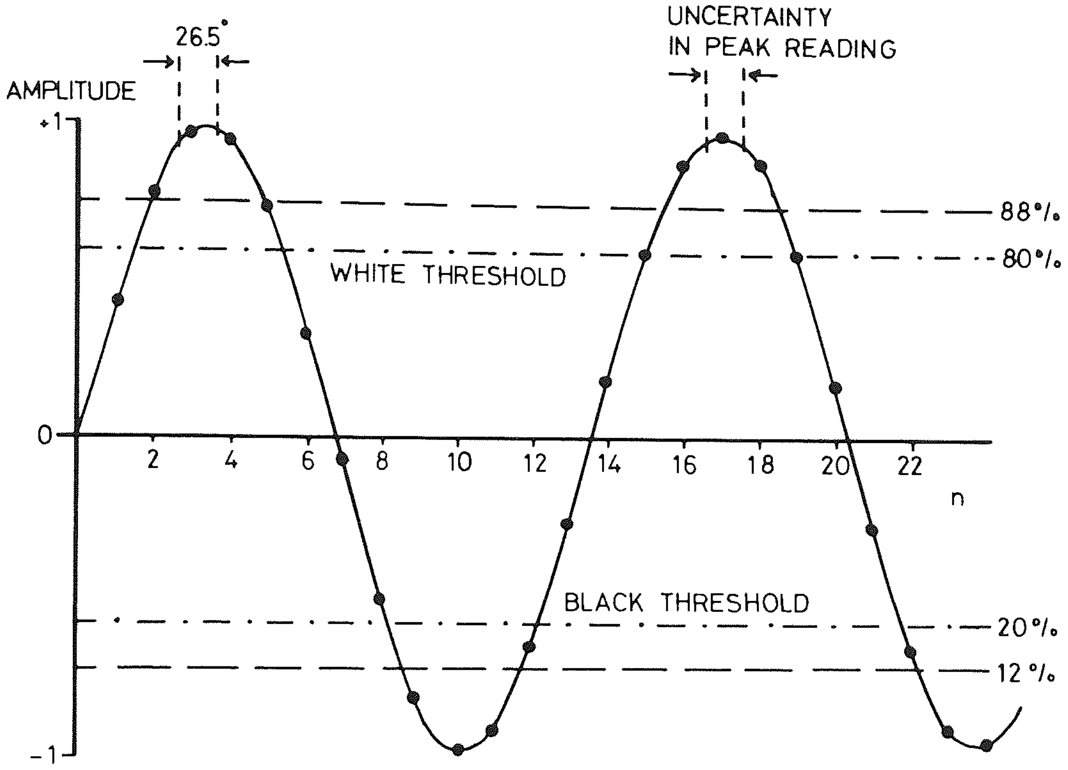


Figure 7.9 Normalised Mean Minima For Sensor 2 Distance Tests.

between individual detectors at the focussed position, separate thresholds had to be used for each detector. This was simple to achieve in software and did not significantly alter sampling times. Choice of thresholds took place as follows: the sampling frequency was such that there were approximately 13 readings per wavelength of the output of figure 7.5b, so for any wavelength, the nearest sample could occur within ± 13.25 degrees of the true peak as in figure 7.10. In addition, comparison of outputs showed a phase difference of ± 1 reading between some detectors, probably due to misalignment of the photodiodes. Therefore when viewing a marker, one detector may reach its peak white value whilst its neighbour samples a point 26.5° below its peak. The thresholds must allow for this to produce a match at this point. Combining this with the positional uncertainty of a sample reading within a particular wavelength, a criteria was formulated for calculating a threshold value from the mean peak value. Approximating the continuous pattern output to a sine wave, a point 40° (26.5 + 13.25 degrees) away from the peak has a magnitude of 88% peak amplitude. Taking this sine wave approximation and the distribution of experimental results into consideration, it was decided to use 80% mean amplitude as the white threshold. By a similar argument, the black threshold was set at 20% mean amplitude as in figure 7.10.

The mean amplitude used for these calculations had to take into account variations with distance. In practice, the range specification was increased to 28 ± 2cm to accommodate the route following system during its development stages, so for each detector, the lowest mean peak and highest mean trough values occurring within this range were used to calculate the worst case amplitude. This was then used to calculate the thresholds shown in table 7.1 for each sensor.



$A = \sin(2n\pi fT)$

$f = 25$ Hz

$T = 1/f_s$

Sampling frequency $f_s = 340$Hz

Figure 7.10 Formulation Of Thresholds.

DETECTOR	WHITE	BLACK
1	290	136
2	418	199
3	473	242
4	434	156
5	308	160
6	456	262
7	707	344
8	406	130

a) SENSOR 1

DETECTOR	WHITE	BLACK
1	564	262
2	538	267
3	326	157
4	564	237
5	378	212
6	366	167
7	144	122
8	384	180

b) SENSOR 2

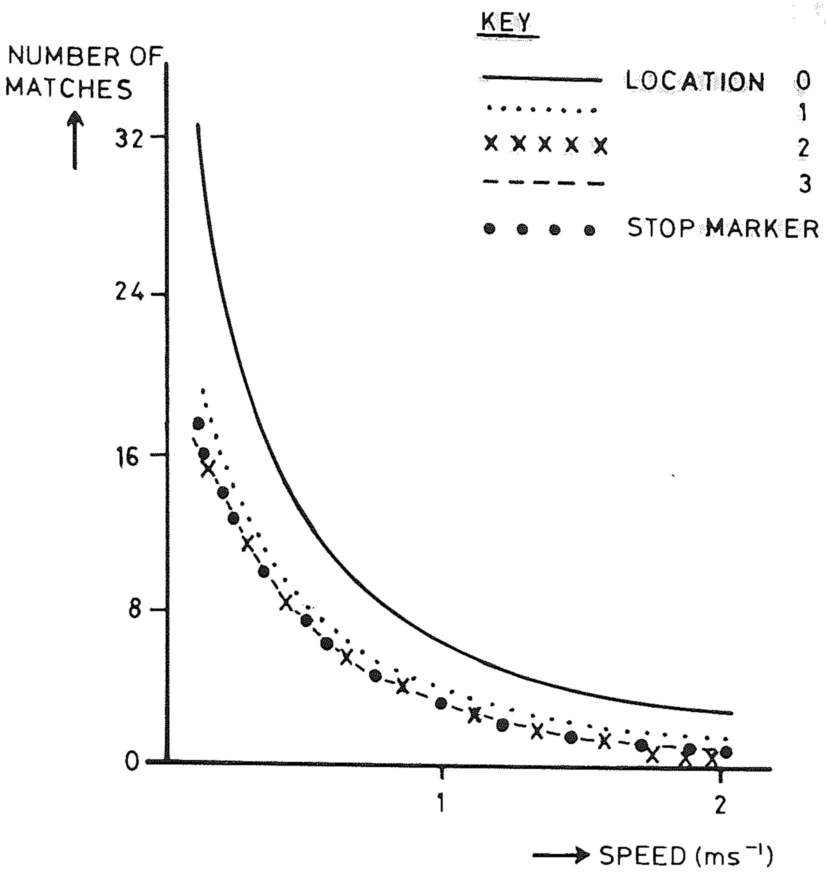
Table 7.1 Threshold Levels For Sensors 1 And 2.

Height tests confirmed a tolerance of ± 10mm in relative height between sensor and marker with no significant change of response.

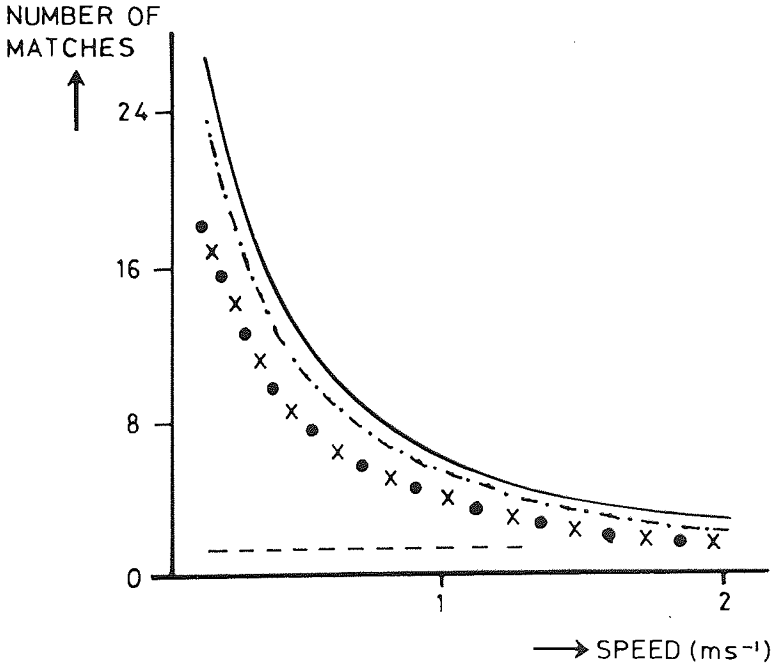
7.1.4 Performance.

Once threshold levels had been chosen, both sensors underwent the recognition tests to ensure they met with the specification and to see how many true matches occurred at a marker. Figure 7.11 shows how the latter varied with speed. For the mark 1 sensor, location marker zero had double the recognition rate of the others, which were all very similar (excluding the all white marker). This was not expected since marker zero is the inverse of marker three and should therefore have had the same rate. The difference may have been due to illumination effects. Sensor 2 showed a wider spread of responses with the stop marker and location 2 having significantly fewer matches as expected due to their higher spatial frequency (40mm^{-1} in both cases). Location zero again showed the highest recognition rate although by a much smaller margin. At speeds up to 1ms^{-1} , a mis-read occurred where a partial view of marker two was 'matched' to template three. This was probably due to the problem of photodetector alignment already mentioned combined with the uneven illumination.

Figures 7.12 and 7.13 show the results of distance and angle tests, from which it can be seen that sensor 1 has a much more even response in each case. Mis-reads again occurred with sensor 2, but mainly involved only one match (except at 26cm where the occurrence of a false location three rose to three matches). The problem was overcome by only registering a match after at least two consecutive occurrences, but due to low visibility of the stop marker, this sensor was not reliable at the closest range specified. Its response to changes in incident angle was also poorer, having a range of ± 10 degrees after

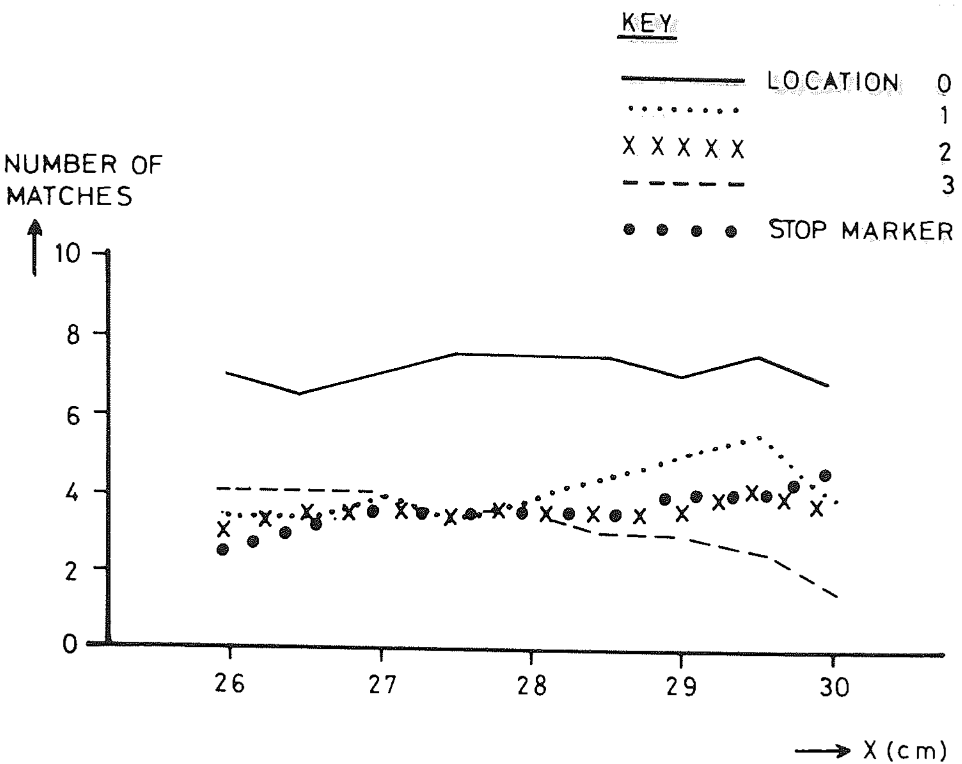


a) Sensor 1

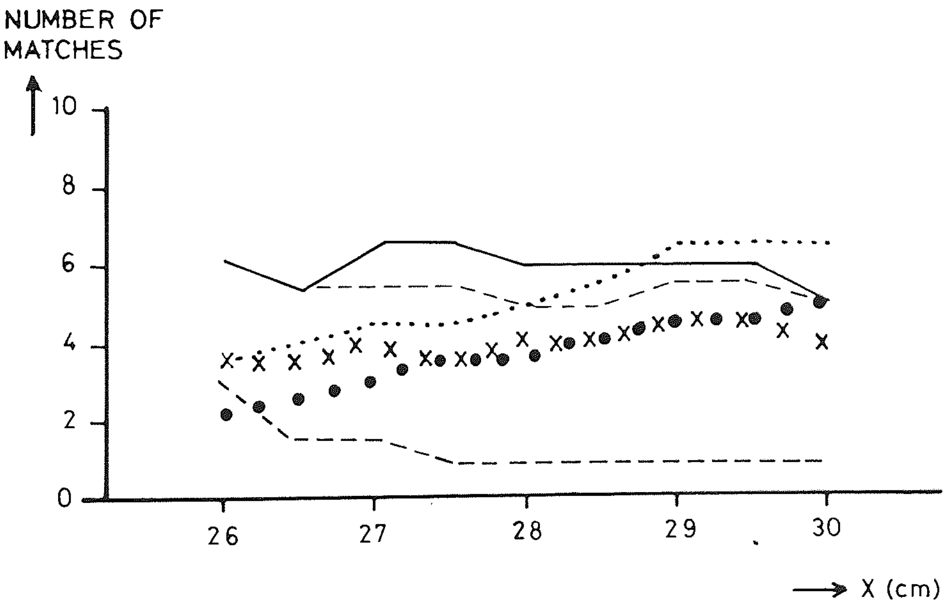


b) Sensor 2

Figure 7.11 Marker Recognition Versus Speed.

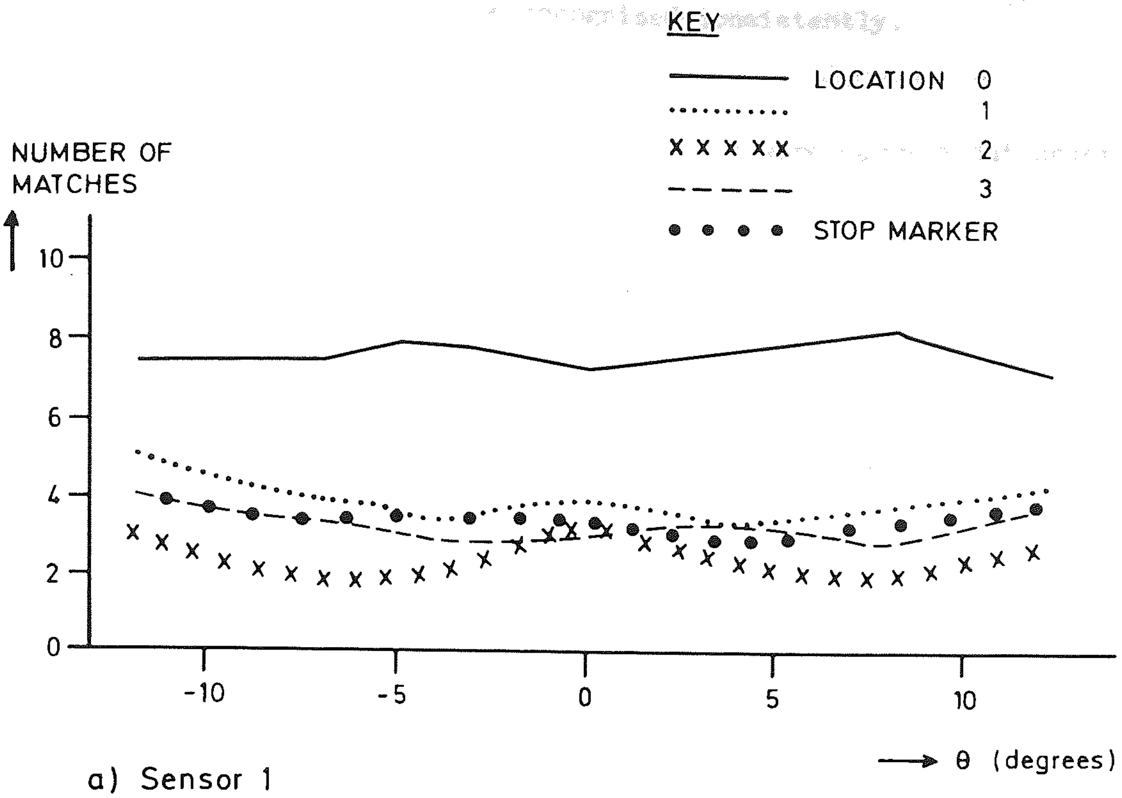


a) Sensor 1

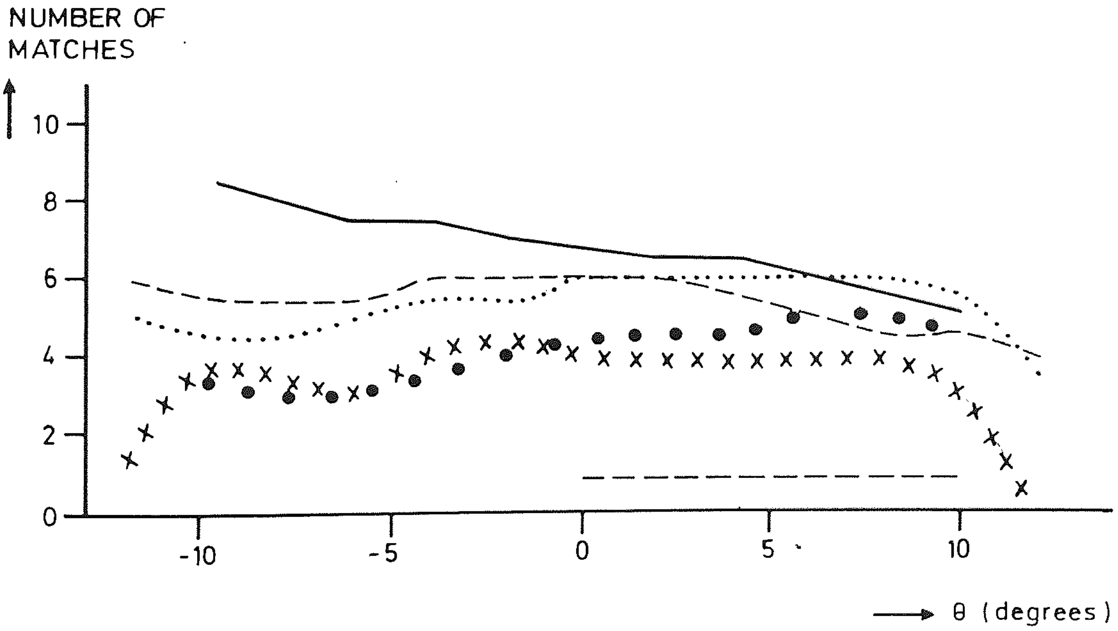


b) Sensor 2

Figure 7.12 Marker Recognition Versus Distance.



a) Sensor 1



b) Sensor 2

Figure 7.13 Marker Recognition Versus Angle.

which two markers ceased to be recognized consistently.

During the course of these experiments, tests were carried out under various lighting conditions such as fluorescent light, daylight and both. In all cases no systematic difference was found and all results fell within the experimental error evaluated during several tests executed under constant conditions.

7.2 Vehicle Trials.

Once all the systems were operational, the vehicle was made to perform a set of tasks designed to fully exploit its capabilities. These were later combined to form a demonstration programme of manoeuvres. The tasks were as follows:

1. Proceed to location n, stop, load, then continue along the guide-wire.
2. Stop at location n, unload and continue.
3. Stop at location n and wait for a prescribed period before moving off.
4. Stop at first stop marker at location n.
5. Stop at second stop marker at location n.
6. Stop at first stop marker of location n, then move on to stop at second stop marker.
7. Leave wire at location n and proceed at a pre-programmed steering angle until the next guide-wire is encountered, when route following recommences (Software Resident Manoeuvre - SRM).

Figure 7.14 illustrates the test environment layout which consisted of two guide-wire sections, four location markers and two workstations. The dotted lines represent the vehicle path during SRMs.

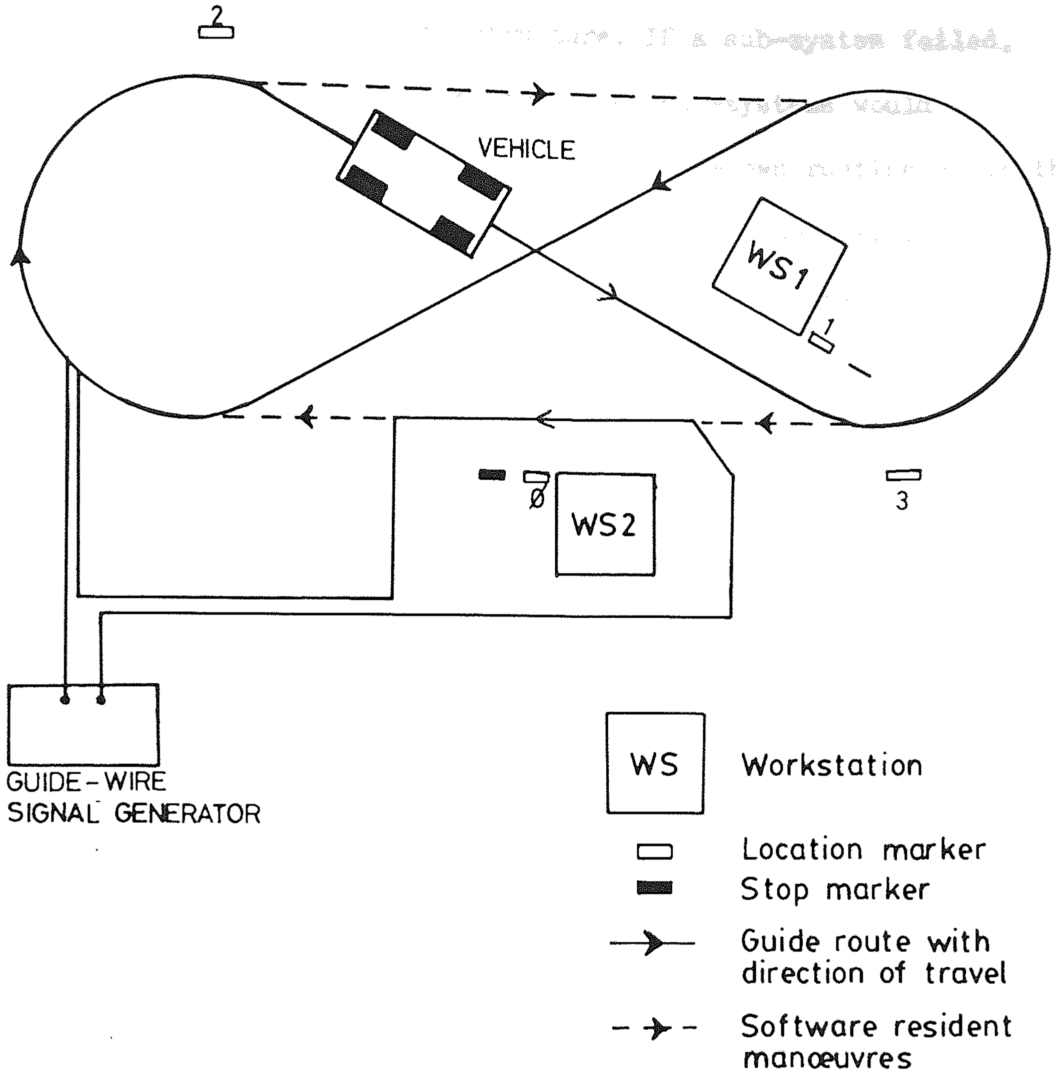


Figure 7.14 Demonstration Layout.

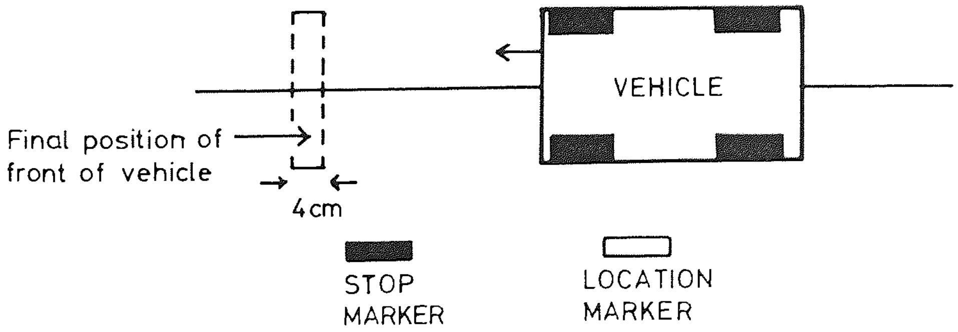


Figure 7.15 Stopping Distances.

The major problems encountered with the complete system involved the inter-processor communications structure. If a sub-system failed, causing communications to cease, the other sub-systems would automatically reset or enter an emergency shut-down routine as in the case of Traction. This had the desired effect of immobilising the vehicle if any mal-function occurred, but in some instances made it impossible to see which sub-system was faulty. This was particularly true in the early stages of debugging the communications routine itself. Since every processor was linked together via a PIO, the system was also susceptible to catastrophic faults in which a noise problem on one board, causing PIO failure, could destroy some or all of the communications PIOs. This happened on two occasions.

Due to problems with the route following system, maximum vehicle speed in practice was $0.7ms^{-1}$. At this speed, it was found that the location and stop markers needed to be 1m apart for the vehicle to slow down sufficiently before encountering the stop marker. As previously mentioned, the vehicle came to rest a short distance in front of the marker and over repeated stopping tasks, the deviation in final vehicle positions was ± 4cm from the mean. When the vehicle stopped at location 0, there was less variation since it was travelling at reduced speed when the location marker was passed, due to the SRM. In this case, the vehicle tended to stop within a 4cm long box as illustrated in figure 7.15. Demonstration loads consisted of cardboard boxes, so no appreciable difference in performance was experienced between loaded and unloaded conditions. Under heavy loads, the vehicle would be expected to have a larger overshoot, but it was not considered worthwhile measuring this as it was a known disadvantage of the simple speed control system used. The final design, using all the possible stop marker views, requires controlled braking, whereas the present system relies on friction to slow the vehicle. Time, and the company's interest precluded the development of acceleration control, but the speed control system was sufficiently accurate for load transfers under the demonstration conditions.

Chapter 8

CONCLUSIONS AND FURTHER WORK

8.1 Project Summary.

This study set out to design a navigation and positioning system for an automatic vehicle within a research project aimed at providing a company with a new product. It has involved an investigation into commercial AGVs from which it was found that the major disadvantages of current systems stem from the need for a fixed guide route with the associated problems of installation and inflexibility of operation. A review of robotic sensors and research has shown that attempts to produce a free-roving vehicle using ultrasonic and tactile methods are as yet commercially unviable and that the more promising computer vision techniques are mainly confined to mainframes and simulations. The author decided to pursue a camera-type sensor and produce a minimum system which could be implemented via an on-board microprocessor. To this end, sampled data systems and image processing have been investigated and the minimum theoretical requirements identified. An optical sensor has been designed to recognise markers using image-template correlation and the markers themselves designed such that template storage was kept to a minimum. Within the sensor, a pulsed illumination source was used to uniquely define the relevant information within the sensed signal, and data input further reduced by arranging light source and detector such that only objects occurring within a specified range of distances were illuminated. This sensor was then incorporated into a control system which enabled a vehicle to execute fetch and carry tasks. The complete system has been demonstrated on a prototype industrial truck with automatic load transference to and from conveyor belts.

8.2 Problems.

8.2.1 Sensor.

Sensor illumination has constituted a major problem throughout the project. The low output power of readily available infra-red emitters precluded use of diffusers to give a plane source. Instead, the radiation was focussed by a lens to maximize the irradiance reaching the marker, which led to highly non-uniform intensity distributions across the plane of view. This was emphasised by the wide variation of device performance necessitating different gain factors for each photodiode amplifier and the use of individual thresholds. Such a high degree of component selectivity both in hardware and software is undesirable since each sensor produced would have to undergo lengthy setting up procedures. Further to this, it was evident in the Mk. 2 sensor that poor illumination was preventing the unit fulfilling its specification. Use of discrete devices in both the emitter and detector, created problems in mechanical alignment which resulted in phase differences between detector outputs. Finally, the high sensitivity required from the detectors made the unit susceptible to blinding by external light sources, although the collimating effect of the lens system limited this to sources directed into the lens itself.

If a much simpler light source were used, such as a tungsten-halogen bulb and diffuser, desirable properties of bright, even illumination would be gained at the expense of selectivity of input data, as it would not be possible to divorce the marker signal from the effects of ambient light by a simple filter. Mis-interpretation of shadows as solid objects is one of the major problems in automated scene analysis today and the two main approaches are to avoid shadows by designing special light sources or to use high spatial sampling frequencies and complex processing algorithms. In the latter method, the scene is usually divided into a large number of pixels, and contrast gradients identified. These are then thresholded either by a pre-set level, or by a strategy based on other information within the scene. Obviously, this requires more processing time and faster sampling rates than the method employed here. However, a photodiode array would simplify construction of the sensor and provide precision diode alignment, and the high contrast between marker elements already used would enable a simple contrast threshold to be employed.

8.2.2 Multiprocessor Operation.

The other area where a change of approach is desirable is the inter-processor connections and communications. With the present arrangement, data is transferred between RAM on the Executive CPU board and RAM on a sub-system CPU board via an interrupt driven procedure. In the case of sub-system initialized communications, two interrupts are involved: sub-system to Executive and a return from Executive to sub-system during which data transfer takes place. The sub-systems are priority-ranked and queue to talk to Executive. In some instances, this queueing time became substantial due to the large number of data transfers occurring. Additionally, one sub-system had to disable interrupts for part of its programme execution, so if Executive required communication at this stage, it was left in a program loop until the sub-system re-enabled interrupts.

In both cases time limits were imposed on communications requests after which a delay was called before retrying. In a more complex system, this could cause considerable delays in transmitting emergency commands, and it is therefore desirable to limit the number of interrupts occurring. An alternative method would be to have an area of RAM directly accessible to all sub-systems and to Executive. This block of memory would be divided into 'pigeon-holes' assigned to each system. Information held here would be updated and inspected within a subroutine call and interrupts reserved for emergency procedures. The direct link between Position and Traction should be removed and communications directed through the standard protocol to simplify fault-finding.

Whatever system is adopted for connecting the processors, some diagnostic fault-finding equipment is required. At present, a fault in one sub-system's operation usually results in the vehicle being stopped via the action of safety trips controlling the main battery line contactor. An exception is the case of recognition failure in the sensor which can be detected by failure of the vehicle to stop at its task location. If the vehicle is stopped, the faulty system can be identified by the state of its trip relay, except in communications failures when one or more systems may re-set and trip. Again, the system which appears to be malfunctioning may only be doing so because of faulty instructions from Executive. Thus most fault-finding investigations have involved a trial and error strategy where key devices are systematically replaced with known working ones and the system re-tested until the faulty component or board is found. This process would not have been necessary in the majority of cases if the current status of each system had been known.

Vehicle Executive should be given some form of user input/output capabilities for the display of status information. This could be as simple as on l.c.d. display and hex keypad whereby the operator could key in the number of the subsystem and receive the 32 bytes defined in the communications protocol. More complex software could be developed whereby the Executive analyses all status records, identifies the sub-system and the fault and displays an error message. In a product situation, the display unit could be a plug-in item solely for maintenance, or used as a permanent accessory displaying the current task in normal operation.

8.3 Commercial Viability.

Each prototype stage of the Development Programme was designed to form the basis of a product. The first prototype obviously operates at a rudimentary level, but nevertheless has some attractive features when compared with commercially available AGVs. Flexibility is the key word in factory automation today, thus one important drawback of wire-guided vehicles is the amount of copper which must be buried in the floor and the difficulty of re-arranging the route. Because of the optical navigation and positioning, the Cableform system only requires a single wire. Minor alterations such as changing the distance between workstations or the introduction of a new station on a line can be simply accomodated by moving the markers and changing the task instructions. Changes requiring a different route only involve relaying or extending one wire.

The most common method of automatic load transfer on existing systems requires precise positioning of the load on special supports so that the vehicle can pass underneath and pick up the load via an on-board elevating table. This in turn requires fixed size or palletised loads. The prototype interfaces with conveyor belts and can thus handle a variety of load shapes and sizes without special constraints on position. In addition, the vehicle automatically activates the work-station conveyor only when it is correctly positioned. Elevating tables could of course be fitted without altering the vehicle positioning system.

In terms of navigation and position, the system developed is not confined to wire-guided vehicles and would be equally applicable to white-line following or tracked vehicles. In its present state the sensor has a capacity of seven location markers. Using the existing processing algorithms, this could be extended with additional microprocessors to operate a larger detector array and hence higher number of marker elements. The limiting factor would be the illumination system, but an increase to 32 locations (five bits of location number information and a parity bit) is feasible with the present IRED source.

For the vehicle as a whole, the first prototype stage appeals to customers who want to ferry items between short-term stores and various finishing processes or to move workpieces through a machine shop or assembly area. In these cases, the attractions are the ease of layout alterations, and speed of installation. It is also a low cost method of introducing automation with the prospect of expansion if it proves successful but low risk if not.

8.4 Achievements Of The Navigation Project.

In order to clarify the boundaries of this study within the larger topic of the automatic guided vehicle, the scope of the Navigation Project is defined here. It covers the design and development of all the sensor hardware and software, including the markers; all the navigation and traction software; interfacing with the rest of the vehicle; and the principles of traction control, except for the power electronics. The interfacing included not only hardware, but also the sub-system software for the communications routine.

The result is a self-contained unit which is capable of retrieving information from markers at a range of 300 ±20mm under both artificial and day light, with a vertical tolerance to marker misalignment of ±10mm. This information is presented as digital numbers at an eight-bit parallel port with a defined handshaking protocol for transmission to another computer. The information is also used internally to produce signals suitable for controlling position via a separately excited drive motor. The unit receives instructions concerning the action to be taken at any marker via the eight-bit port already mentioned, and can be given speed information for the drive motor through a second port. Markers can be read at sensor speeds of up to $1ms^{-1}$ and motor speed and braking during positioning are decided by the navigation routine.

The major achievement of the project has been the production of a viable and attractive alternative to current automatic guided vehicle navigation and positioning methods, which has been demonstrated on an industrial vehicle performing typical tasks. Viability has been enhanced by employing a microprocessor for sensor data processing in place of duplicated units of discrete hardware (one for each photodiode). In this way, the material cost of the system, excluding traction hardware, was kept as low as £80 (one off). In order to cope with the limited capacity of a microprocessor, several techniques have been employed to keep the data processing to a minimum. These techniques include the design of the markers with their highly ordered data presentation and the use of a custom light source to define the sensor field of view. Within the processor, sampling was performed at the minimum reliable frequency for reconstruction of the image, and data was compressed through thresholding and encoding. Such was the success of this methodology that the decision processes of the navigation system were interleaved with the sensor software to run in the same microprocessor and yet the specification of a 250 Hz sampling period was comfortably achieved.

8.5 The Next Steps.

The position sensor was an exercise in minimizing the processing power required to perform a specified set of tasks based on commercial automatic vehicle requirements. This has produced a viable system implemented in a Z-80A microprocessor which has been used on a practical vehicle. The next step is to trade off the associated problems described above against the cost and connotations of larger and more powerful processing systems.

Manufacturing could be simplified by using a solid-state camera and a d.c. visible light source as already discussed. This would be a justifiable course of action since the severe constraints on field of view and image content must be eased to increase system versatility. More detailed images demand higher resolution in detection and more varied image contents require more bits to describe them. In addition to widening the scope of the sensor, the capabilities of the vehicle system must be extended to taking decisions based on past events rather than relying solely on recognition processes. In the case of a large number of locations, the present system would require X unique markers, and X unique templates for X locations. If the system built up a map of its environment, locations could be addressed much like houses on a street using fewer numbers, combined with road identifiers. Complexity is therefore transferred from the sensor - which has fewer markers to recognise - to the control process which has to store and assimilate more data.

To exploit the capabilities of the present sensor/marker system, traction control must be redesigned so that braking and acceleration can be controlled. On the prototype, braking is either purely frictional, or can be augmented for emergency stops by reversing the field current. The position system gathers information on vehicle distance relative to the stopping point with a view to combining this with the current speed and load for calculating the required deceleration for accurate positioning. The Route Following system should also be made capable of operating in reverse so that position adjustments can be made if the vehicle overshoots.

Before a free-roving vehicle can be considered, the longitudinal and transverse positioning functions must be combined. At present, vehicle position is specified by two parameters: the guide-wire indicates where the route sensor will be and the stop marker gives the final location of the position sensor. Vehicle attitude is uncertain, especially if required to stop on a bend in the guide-route. The concept of a free-roving vehicle is that it will optimize its route through a cluttered environment to its target and align itself correctly for its task. This implies accurate positioning of a number of points on the vehicle relative to one or more points in the environment.

When considering the dream of an automatic vehicle, totally independent of its surroundings the same approach should be taken as in the design of the position sensor. There, it was possible to implement real-time pattern recognition in an eight-bit microprocessor when the problem was reduced to its minimum requirements. For the next breed of automatic industrial vehicles, the emphasis should be on defining the capabilities required rather than the ideals desired.

APPENDICES

Appendix A

COMMERCIAL WIRE-GUIDANCE SYSTEMS

The most common wire-guidance vehicle is the simple pallet truck, followed closely by tugs or tow-tractors. In recent years, more ambitious designs have encompassed forklift and narrow aisle trucks and the humble pallet carrier has been given a variety of load handling mechanisms including lifting tables and roller conveyors. Most manufacturers offer a complete automated load handling package consisting of one or more trucks, specialised load/unload stations and a central computer. The latter can usually receive instructions manually from a keyboard or direct from another computer in charge of production or stock control. One company, Logisticon, sells a wire-guidance kit to original equipment manufacturers, which has been applied to narrow aisle trucks as a driver aid by Cleco and Barlow Handling.

Table A1 lists the major suppliers of wire-guided vehicles. Jungheinrich and Malthouse Hunter took over servicing of Robotug systems when EMI left the market in the mid-seventies after twenty or so years of involvement. Malthouse claim to be the only European company specialising solely in driverless vehicle systems. They market tractors and pallet carriers under the "Guide-O-Matic" tradename and offer wire or white line guidance or radio remote control. The pallet carriers can automatically pull pallets onto themselves and discharge them at any number of pre-set stations, and the tractors feature automatic coupling. Permanent stopping points are marked by permanent magnets set in the floor, whilst reflective discs can also be placed along the track to mark temporary stations. These are read by an infra-red sensor on the vehicle.

Jungheinrich also use permanent magnets as a positioning aid to achieve ± 5cm accuracy. Navigation information is provided by two signals in

MANUFACTURER	INSTALLATIONS
Jungheinrich	Triumph International, Swindon
Malthouse Hunter	
Eaton Kenway	Keebler Distribution Centre, Alsip, Illinois
Wagner Indumat	Heinrich Heine Warehouse, Karlsruhe, W. Germany
BT Rolatruc	OVD Warehouse, Oslo, Norway
Fata Carrago	General Motors, Antwerp, Belgium
Babcock	
Komatsu Forklift Co.	Komatsu, Kawasaki, Japan

Table A1 Major Wire-guidance Suppliers.

the guide-wire: one provides the magnetic field distribution for the steering system, and the second represents a 'travel' command. Where there is a choice of routes, for instance a branch, the travel signal will be switched off in the prohibited track, thus routing the truck along the desired path. This switching may be performed manually or via magnetic switches activated as the truck passes. The vehicle may also be requested to stop simply by turning the travel signal off in a portion of track equal in length to its braking distance. Automatic positioning is achieved via an on-board microprocessor which can be programmed to stop at a number of points. In simple systems, the vehicle keeps track of its position by counting stopping points and branches in passing. More complex systems use coded markers made up of several magnets or coils and read by magnetic switches on the truck. Each marked location has a unique identifier which the vehicle transmits to the control computer. This computer works out the desired path and tells the vehicle which branch to take or whether to stop. The vehicles travel at 1.2ms^{-1}, reducing to 0.8ms^{-1} for curves, and one model is bi-directional. Magnetic switches can be vehicle activated to open doors and operate traffic lights, and the system can be installed outdoors. Various automatic load transfer devices can be fitted, including swivelling and side-shifting forks, roller conveyors or lifting platforms.

Eaton Kenway are licensees of Digitron's Robocarrier together with Schindler Lifts. Eaton's applications are pallet trucks with lifting platforms or chain conveyors which are designed to pick up loads from special fixtures. The vehicles are symmetrical and may travel in either direction at speeds up to 1ms^{-1}. Tight bends and positioning manoeuvres are negotiated at 0.1ms^{-1}. The complete system features central computer control with local control units for dividing the track into sectors and controlling switching, steering and stopping points for all vehicles in each sector. Positioning is mechanical with four conical locating pins fixed to the floor at each destination, which mate with a hydraulic position correction system on the base of the vehicle as it stops.

To avoid the problems of putting loading fixtures above the guide path, Wagner Indumat have given their pallet trucks all wheel steering. This enables the vehicles to move sideways to pick up and deposit loads, leaving the main route free for through traffic. Their fork-lift trucks have tactile sensors which detect when the pallet is correctly positioned and automatically instruct the vehicle to move off. Communication between vehicles and the central computer is via inductive transmitter/receivers in the floor and on-board. Alternatively, the vehicles can be manually operated by push-button selection of on-board programs. Locations and destinations are identified by markers along the guide-path - magnets and reflectors - or by interruption of the traction frequency in the guide-path. Different sections of the route can also be given different steering frequencies and the vehicle programmed to follow the desired signal. At an installation in the Heye Glassworks at Obenkirchen in Germany, tugs and trailers are used. Here, positioning is achieved by means of a light barrier which detects when each trailer is aligned and instructs the tug to stop. When the load transfer is complete, the tug moves on until the next trailer breaks the barrier.

Another vehicle featuring load sensors is the BT Rolatruc Automatic Low Lifter (ALL). This has radar mounted on the fork tips to sense the closeness of the pallet and slow the vehicle to creep speed during pick-up. They are also used as safety devices to detect unexpected obstructions. Feelers are mounted on either side of the truck to ensure the path is wide enough to accept the load. The ALL has castors and a single drive wheel, with a steering sensor at each end to facilitate travel in either direction. It travels at three pre-programmed speeds of 0.15, 0.6 and $1.1 ms^{-1}$. BT also produce an Automatic Pallet Mover - a flat bed carrier which can be fitted with a lifting table or a conveyor.

One of the problems with wire-guidance systems is the space required to operate traffic in both directions, since in general this requires two guide-wires or a complicated sectioned track with passing loops. Carrago Transport System AB of Sweden has overcome this problem with their pallet trucks which are capable of leaving the wire for short intervals to facilitate overtaking or passing. This also allows access to areas such as lifts where a wire cannot be laid. Each vehicle has its own on-board microprocessor which can be programmed with the task. As tasks are executed, vehicles can communicate with each other directly to update their relative positions. For larger installations, a central computer is provided to co-ordinate routing and tasks. All vehicles have optical scanners to sense obstacles, and travelling speed varies from $1.25 ms^{-1}$ forwards to $0.5 ms^{-1}$ in reverse.

Babcock market a system of flat-bed carriers which only accept special 'roll containers'. These are trolleys, mainly used in hospitals for meals distribution, laundry collection etc. Two types of carrier are produced: a drive under tug with a locating pin which locks onto the load and tows it to its destination, and the 'Automatic Transfer Robot' which is fitted with a roller conveyor. In this system, the container is programmed with the destination which is set up as a magnetic code on the base and read by inductive sensors on the vehicle. The latter then decides its own shortest route and begins the task. In simple systems, the route is divided into loops, only one of which is activated at a time. The vehicle can activate a loop through control elements set in the floor. Multi-frequency systems again have several loops, but each has a different frequency signal and is permanently activated. Route selection takes place on-board with the vehicle deciding which signal to follow. Speed varies from $0.1ms^{-1}$ when positioning, to a maximum of $1.5ms^{-1}$.

All the vehicles mentioned so far interact with special loading stations. Komatsu Forklift have developed a system to enable random stacking of pallet loads without the need for racking. Their design is a standard manual fork-lift truck fitted with a motor with chain drive to the steering column, inductive sensors on the chassis and ultrasonic sensors on the forks and mast. The guide route is set out as a series of parallel lanes and carries a steering signal and a traction signal to control acceleration and braking. When a truck enters a lane, the ultrasonic sensors scan for the presence or absence of pallet loads at various heights. If it is unloading, the forks are moved to the height of the lowest void and proximity sensors are used to guide the pallet in relation to adjacent loads. This allows for position variations due to compression of stacked loads. For loading, the forks are guided to the topmost pallet in the stack. A central computer co-ordinates the vehicles and keeps an inventory of where stocks have been stored.

Other companies involved in wire-guidance include Lansing Bagnell who use the system as an auto-pilot to aid drivers of their turret trucks. Volvo and Telelift Transcar both use metal tape as the guide route and Mazda of Japan advertise wire-guided golf trolleys. A number of Japanese companies have AGV developments, mainly for use within their own companies. A list of the companies mentioned here, together with contact addresses, may be found in the List of References.

Appendix B

PROJECT SOFTWARE

B.1 Navigation Program.

This program embodies both the sensor control and data processing, and the navigation software described in chapters 5 and 6 respectively. The sensor software controls the operation of the infra-red emitters and the sampling of the detectors, and also performs filtering and thresholding of the incoming signals before compressing the image data into an eight-bit word which is compared with a pre-programmed set of templates. Interfacing to the navigation software is achieved via a flag signifying a true match, and a pointer to the matched template. The navigation software has to deal with two situations: travelling to the next location requested by Vehicle Executive, and positioning at the task location. Each situation is dealt with by a loop which samples the sensor and decides the next action based on sensor output and Executive's instructions. Operation alternates between the two loops as the vehicle task requires. Communications with Executive take place via the Communications Handler and follows the standard protocol. In addition, the Navigation program communicates with the Traction sub-system over a direct link.

B.1.1 Program Layout.

	Memory address
Program Start (start of PROM)	0000H
Start of RAM	1800H
End of RAM	1C00H
Bottom of Stack	1C00H
Start of Tables	0003H
Interrupt Mode	2
Interrupt Service Routine Address held at	1800H

Initialisation:

```
Clear RAM.
Clear interrupts.
Load interrupt vectors and addresses.
Initialise PIOs.
Enable interrupts.
```

Navigating Loop:

```
Initialise flags.
Wait for instructions from Executive.
LOOP
    Take one sensor sample.
    Check for any change in instructions from Executive.
    If no marker has been recognised, continue from LOOP.
    If marker is requested location, tell Executive and jump to the positioning loop.
    Tell Executive current position and continue from LOOP.
```

Positioning Loop:

```
Tell Traction to slow down.
Initialise flags.
LOOP 2
    Take one sensor sample.
    Check for any change in instructions from Executive.
    If no marker has been recognised, continue from LOOP 2.
    Send to Traction the speed demand corresponding to marker view recognised.
    If the centre of the marker has not been reached, continue from LOOP 2.
If central view recognised, tell Traction to stop, tell Executive
```

the task is completed, and wait for further instructions.

The sub-routines will now be described individually, followed by the assembled program listing.

B.1.2 Sub-routines.

SETPT

Sets the PIOs to their initial conditions.

Entry Requirements: none.

Exit Status: register A modified, PIOs set up as described in listing.

BLINK

Sends out a Start Of Conversion (SOC) pulse to the ADCs and waits for the returning End Of Convert (EOC). During the conversion time, the first ADC address is output.

Entry Requirements: variable ADATA must hold current data on PIO 2 port A.

Exit Status: register A modified, variable ADATA updated.

MUX

Reads the ADC data bus and stores in register C, then increments current ADC address.

Entry Requirements: variable ADATA must hold current data on PIO 2 port A.

Exit Status: register A modified, variable ADATA updated, ADC reading in register C.

NEWTOT

Performs first adjustment of running total and stores latest 'On' reading.

Entry Requirements: register IX must point to running total for detector being sampled, IY must point to oldest reading in Sample table, C must hold latest

'On' reading.

Exit Status: register A modified, Total and Reading tables updated.

SUBT

Performs second adjustment of running total, and filtering of the current detector reading.

Entry Requirements: as NEWTOT, except C holds latest 'Off' reading.

Exit Status: as NEWTOT.

THRESH

Thresholds completed running total and stores result in array CODE.

Entry Requirements: IY points to threshold for 'White', IX points to current Total.

Exit Status: array CODE updated, bit 0 of FLAG set if sample is invalid, reset otherwise, register A modified.

DECODE

This routine is entered after a complete sample of all eight detectors. If bit 0 of FLAG is reset, the routine takes the word CODE and tries to match it with one of the templates held in store.

Entry Requirements: bit 2 of variable FLAG reset.

Exit Status: register A modified. If a match is found, bit 2 of FLAG is set, register pair HL point to the marker identity, registers B and C are modified.

SAMIC

Sets up initial conditions for sampling for location markers.

Entry Requirements: none.

Exit Status: IX and IY point to the top of the Total and Sample tables respectively, and these tables are cleared. The template table pointer TAB4 points to the location marker template table. FLAG and CODE are reset and ADATA initialised. Registers A, B, H and L modified.

EDGIC

Sets up initial conditions for sampling for a stop marker.

Entry Requirements: none.

Exit Status: same as SAMIC, except TAB4 points to the stop marker
template table.

SAMPLE

Deals with all the sensor control and data processing for one sensor sample (see chapter 5).

Entry Requirements: variable ADATA holds current data on PIO 2 port A,
IX and IY point to the current position in the
Total and Sample tables respectively.

Exit Status: registers IX, IY, A, B, C modified.

INSTRS

Halts the processor and waits for Executive to interrupt. Then it stores the information received.

Entry Requirements: none.

Exit Status: registers H, L, D, E, A modified.

CHK

Checks to see if new information has arrived from Executive.

Entry Requirements: none.

Exit Status: registers H, L, A modified, variable LOCSTP updated
with requested location. This routine changes its return
address according to the new instructions.

FLAGS

Clears CODE and FLAGS

LOCUP

Initiates communications with Vehicle Executive.

B.1.3 Communications Handler.

The handler contains an interrupt service routine for Vehicle Executive initiated interrupts, and controls the handshaking for transfer of information between sub-system and Executive. Incoming data is stored in variable INFO and outgoing data is read from variable OUTFO.

CINT

This is the interrupt service routine. All register contents are saved on the stack and routine COMIN is called to interrupt Vehicle Executive in return. Then the registers are restored, interrupts are enabled and a RETI executed.

Entry Requirements: interrupt initiated.

Exit Status: variable INFO updated.

COMIN

This routine is called from the interrupt service routine and deals with all communications requests from Vehicle Executive. First, the communications control word is read from PIO 0 port B to ascertain the direction of data transfer requested.

TX: Sub-system to Executive -	the 16 bytes of variable OUTFO are transferred to Executive serially via PIO 0 port B according to the protocol.
RX: Executive to sub-system -	16 bytes of data are read serially through PIO 0 port B according to the protocol.

ERROR

Called if fewer than 16 bytes are received during RX (this check is in VERIN). In this event, the sub-system requests message repeat through COMOUT.

COMOUT

Requests communication with the executive according to the protocol.

Entry Requirements: the correct communications control word must be in A.

Exit Status: registers A and B modified.

AGAIN

If the executive is busy servicing a higher order sub-system, COMOUT is repeated.

MNSTCK

If the correct handshake signal is not received from Executive at any time, this routine is accessed to decide on the action to be taken.

CINIT

Sets PIO 0 (communications) to quiescent conditions stated in the communications protocol. It is called during initialisation and on exit from either COMOUT or COMIN.

Entry Requirements: none.

Exit Status: a register modified, PIO 0 set to initial conditions.

```
                  ;"PC6/1" POSITION CONTROL
                  ; WITH COMMUNICATIONS
                  ;*******************************
                  ;* PROGRAM TO RECOGNIZE BINARY *
                  ;* CODED MARKERS USING THE     *
                  ;* INFRA-RED DETECTION UNIT    *
                  ;* 45016/2. MK1 SENSOR.        *
                  ;* A.BAKER              20-4-82 *
                  ;*       THIS VERSION    2-3-83 *
                  ;*******************************
                  ;
                  ;CONSTANTS
                  ;
0000              X       EQU 0000H         ;MEMORY OFFSET
0000              SROM    EQU 0H+X
1800              RAMTOP  EQU 1800H+X
1C00              RAMEND  EQU 1C00H+X
0000              P1      EQU 0
0001              P1C     EQU 1
0002              P2      EQU 2
0003              P2C     EQU 3
0004              P3      EQU 4
0005              P3C     EQU 5
0006              P4      EQU 6
0007              P4C     EQU 7
0008              P5      EQU 8
0009              P5C     EQU 9
000A              P6      EQU 0AH
000B              P6C     EQU 0BH
                  ;
0000                      ORG SROM
0000                      IDNT $,$
0000 183B                 JR GO
                  ;
                  ;****************************************
                  ;TABLES
                  ;
0002 2201A201     THRON   DEFW 290,418,473,434
000A 3401C801             DEFW 308,456,707,406
0012 8800C700     THROFF  DEFW 136,199,242,156
001A A0000601             DEFW 160,262,344,130
                  ;
0022 14           SLOW1   DEFB 20
0023 08           EDGE    DEFB 8            ;0000/1000
0024 8C                   DEFB 8CH          ;1000/1100
0025 46                   DEFB 46H          ;0100/0110
0026 A3                   DEFB 0A3H         ;1010/0011
0027 51                   DEFB 51H          ;0101/0001
0028 20                   DEFB 20H          ;0010/0000
0029 10                   DEFB 10H          ;0001/0000
002A 14                   DEFB 14H          ;0001/0100 20% D.C.
002B 14                   DEFB 14H
002C 14                   DEFB 14H
002D 00                   DEFB 0
002E 14                   DEFB 14H
002F 14                   DEFB 14H
0030 14                   DEFB 14H
0031 78           LOCTAB  DEFB 78H          ;0111/1000
0032 5A                   DEFB 5AH          ;0101/1010
0033 3C                   DEFB 3CH          ;0011/1100
```

```
PolyZap V2.0      PC6/1 FOR MK1 SENSOR 2-3-83 A. BAKER

0034 1E                     DEFB 1EH          ;0001/1110
0035 A3                     DEFB 0A3H         ;1010/0011
0036 00                     DEFB 0
0037 02                     DEFB 2
0038 01                     DEFB 1
0039 03                     DEFB 3
003A 0E           EDIM      DEFB 0EH
003B 05           LEN       DEFB 5
003C 06           EDLEN     DEFB 6
                  ;****************************************
                  ;
003D 210000       GO        LD HL,0
0040 0647                   LD B,71
0042 318C18                 LD SP,RAMTOP+140
0045 E5           L1        PUSH HL
0046 10FD                   DJNZ L1           ;CLEAR RAM
                  ;
0048 31001C                 LD SP,RAMEND
004B 215100                 LD HL,CLRI
004E E5                     PUSH HL
004F ED4D                   RETI              ;CLEAR INTERRUPTS
0051 216503       CLRI      LD HL,CINT
0054 220018                 LD (ISRAD),HL
0057 3E18                   LD A,18H
0059 ED47                   LD I,A            ;HIGH BYTE OF ISRAD
005B CD2301                 CALL SETPT        ;INITIALIZE PIOs
005E 216118                 LD HL,OUTFO       ;INITALIZE OUTFO
0061 360A                   LD (HL).0AH
0063 23                     INC HL
0064 36FF                   LD (HL),0FFH
                  ;
0066 ED5E                   IM 2
0068 3E00                   LD A,0
006A D301                   OUT (P1C),A       ;LOW BYTE OF ISRAD
006C 3E77                   LD A,77H
006E D301                   OUT (P1C),A       ;AND & HIGH
0070 3E7B                   LD A,7BH          ;0111/1011
0072 D301                   OUT (P1C),A       ;INTERRUPT MASK
0074 3E83                   LD A,83H
0076 D301                   OUT (P1C),A       ;ENABLE PIO INTERRUPTS
0078 FB                     EI
0079 3EFF                   LD A,0FFH
007B 328218                 LD (INCHK),A
007E CDEE02                 CALL INSTRS       ;PICK UP INSTRUCTIONS
                  ;
                  ;****************************************
                  ;                NAVIGATING
                  ;****************************************
                  ;
0081 CD1F02       OPLOP     CALL SAMIC        ;ICs FOR SAMPLING
0084 3E40                   LD A,40H
0086 D304                   OUT (P3),A        ;RET SPEED CTL TO EXEC
0088 3E00                   LD A,0
008A 326118                 LD (OUTFO),A      ;UPDATE STATUS
008D 3A0C18                 LD A,(LOCN)
0090 FE0E                   CP 0EH
0092 2803                   JR Z,01
0094 326218                 LD (OUTFO+1),A
0097 CD5903       01        CALL LOCUP        ;TELL V. EXEC
                  ;
```

```
009A CD7502      SAM      CALL SAMPLE
009D 3A0218               LD A,(FLAG)       ;TEST FOR MATCH
00A0 CB57                 BIT 2,A
00A2 2008                 JR NZ,ACTION      ;JUMP IF MATCH
                 ;
00A4 CD4B03      J3       CALL FLAGS
00A7 CD2503               CALL CHK          ;CHECK FOR INSTR CHANGE
00AA 18EE                 JR SAM            ;CONTINUE SAMPLING
                 ;
00AC 3A3B00      ACTION   LD A,(LEN)
00AF 4F                   LD C,A
00B0 0D                   DEC C
00B1 0600                 LD B,0
00B3 09                   ADD HL,BC         ;HL^ MATCHED TEMPLATE
00B4 7E                   LD A,(HL)
00B5 320C18               LD (LOCN),A       ;UPDATE LOCN
00B8 3A8118               LD A,(LOCSTP)     ;LOCN^ STOP POINT
00BB BE                   CP (HL)
00BC 2012                 JR NZ,J8
00BE 3A0218               LD A,(FLAG)       ;WHICH EDGE TO STOP AT?
00C1 CB67                 BIT 4,A
00C3 281D                 JR Z,CLOP         ;***JUMP TO POSITIONING
                 ;                                    IF FIRST EDGE
00C5 CBA7                 RES 4,A
00C7 320218               LD (FLAG),A
00CA 3A3A00               LD A,(EDIM)
00CD 328118               LD (LOCSTP),A     ;DES LOCN := 1st EDGE
00D0 3A0C18      J8       LD A,(LOCN)
00D3 FE0E                 CP 0EH
00D5 28CD                 JR Z,J3           ;CONTINUE SAMPLING
00D7 326218               LD (OUTPO+1),A
00DA CD5903               CALL LOCUP
00DD CDA304               CALL WAIT
00E0 18C2                 JR J3             ;CONTINUE SAMPLING
                 ;
                 ;
                 ;*****************************************
                 ;                POSITIONING
                 ;*****************************************
                 ;
00E2 3A2200      CLOP     LD A,(SLOW1)
00E5 D304                 OUT (P3),A        ;TAKE OVER SPEED CTL
00E7 3E03                 LD A,3            ;0000/0011
00E9 326118               LD (OUTPO),A      ;UPDATE STATUS
00EC 3A7118               LD A,(INFO)
00EF 326218               LD (OUTPO+1),A
00F2 CD5903               CALL LOCUP
00F5 CD4A02               CALL EDGIC        ;INIT CONDS FOR EDGSAM
00F8 CD7502      EDGSAM   CALL SAMPLE
00FB 3A0218               LD A,(FLAG)
00FE CB57                 BIT 2,A
0100 2008                 JR NZ,ACT2        ;JUMP IF MATCH
0102 CD4B03      C3       CALL FLAGS
0105 CD2503               CALL CHK          ;CHECK FOR INSTR CHANGE
0108 18EE                 JR EDGSAM         ;CONTINUE SAMPLING
                 ;
010A 3A3C00      ACT2     LD A,(EDLEN)
010D 4F                   LD C,A
010E 0600                 LD B,0
0110 09                   ADD HL,BC         ;HL^ MATCHED TEMPLATE
```

```
FolyZap V2.0     FC6/1 FOR MK1 SENSOR 2-3-83 A. BAKER

0111 7E                   LD A,(HL)           ;A HAS SPEED DEMAND
0112 D304                 OUT (F3),A          ;SEND TO SCU
0114 FE00                 CF 0                ;CONTINUE SAMFLING
0116 20EA                 JR NZ,C3            ;IF NOT AT FINAL FOSN
                 ;
0118 3E0B                 LD A,0BH            ;0000/1011
011A 326118               LD (OUTFD),A        ;UFDATE STATUS
011D CD5903       CALL    LOCUF               ;TELL V. EXEC
0120 CDEE02               CALL INSTRS         ;WAIT FOR INSTRS
                 ;
                 ;          *** END OF FROGAM ***
                 ;
                 ;
                 ;***************************************
                 ;                 FROCEDURES
                 ;***************************************
                 ;
                 ;1)SETFT: SETS UF FORTS
                 ;
0123 3EFF        SETFT    LD A,0FFH
0125 D301                 OUT (F1C),A
0127 3EFF                 LD A, 0FFH
0129 D301                 OUT (F1C),A         ;FORT 1 ALL INFUTS
012B 3EFF                 LD A, 0FFH
012D D303                 OUT (F2C),A
012F 3EFF                 LD A, 0FFH
0131 D303                 OUT (F2C),A         ;FORT 2 ALL INFUTS
                 ;
0133 3EFF                 LD A, 0FFH
0135 D309                 OUT (F5C),A
0137 3E20                 LD A,20H
0139 D309                 OUT (F5C),A
013B 3E10                 LD A,10H
013D D308                 OUT (F5),A
                 ;***************************************
                 ;* FIO 2 FORT A                        *
                 ;*                                     *
                 ;*     LED           EN      MUX     1 *
                 ;* / 0 / 0 / 1 / 0 // 0 / 0 / 0 / 0 /  *
                 ;*        EOC  SOC       ADDRESS     0 *
                 ;***************************************
                 ;
013F 3EFF                 LD A,0FFH
0141 D30B                 OUT (F6C),A
0143 3EFF                 LD A,0FFH
0145 D30B                 OUT (F6C),A
                 ;***************************************
                 ;* FIO 2 FORT B                        *
                 ;*                                     *
                 ;*   DET        ADC DATA          DET  *
                 ;* / 1 / 1 / 1 / 1 // 1 / 1 / 1 / 1 /  *
                 ;*   7                             0   *
                 ;***************************************
                 ;
0147 3EFF                 LD A,0FFH
0149 D305                 OUT (F3C),A
014B 3E00                 LD A,0
014D D305                 OUT (F3C),A         ;FORT 3 ALL OUTFUTS
014F 3E00                 LD A,00H            ;SFD CTL TO FOS.
0151 D304                 OUT (F3),A          ;ZERO SFEED
```

```
PolyZap V2.0    PC671 FOR MK1 SENSOR 2-3-83 A. BAKER

0153 3EFF                LD A,0FFH
0155 D307                OUT (P4C),A
0157 3E00                LD A,0
0159 D307                OUT (P4C),A        ;PORT 4 ALL OUTPUTS
015B 3E07                LD A,7
015D D306                OUT (P4),A
015F C9                  RET
                ;
0160 3A0518     BLINE    LD A,(ADATA)
0163 CBA7                RES 4,A            ;SOC LINE
0165 D308                OUT (P5),A
0167 CBE7                SET 4,A
0169 D308                OUT (P5),A
016B E6F0                AND 0F0H           ;RESET MUX ADDRESS
016D 320518              LD (ADATA),A
0170 CBDF                SET 3,A
0172 D308                OUT (P5),A         ;MUX ENABLE
0174 320518              LD (ADATA),A
0177 DB08       EOC      IN A,(P5)
0179 CB6F                BIT 5,A
017B 20FA                JR NZ,EOC
017D C9                  RET
                ;
017E DB0A       MUX      IN A,(P6)
0180 4F                  LD C,A
0181 3A0518              LD A,(ADATA)
0184 CB9F                RES 3,A
0186 3C                  INC A
0187 D308                OUT (P5),A
0189 320518              LD (ADATA),A
018C CBDF                SET 3,A            ;MUX ENABLE
018E D308                OUT (P5),A
0190 320518              LD (ADATA),A
0193 C9                  RET
                ;
0194 DD7E00     NEWTOT   LD A,(IX)          ;LOW BYTE OF TOTAL
0197 FD9600              SUB (IY)           ;SUBTRACT OLDEST
019A 3003                JR NC,N1
019C DD3501              DEC (IX+1)         ;ADJUST HIGH BYTE
019F DD7700     N1       LD (IX),A          ;UPDATE LOW BYTE
01A2 FD7100              LD (IY),C          ;STORE NEW 'ON'
01A5 C9                  RET
                ;
01A6 FD7E00     SUBT     LD A,(IY)          ;A HAS LAST ON READ
01A9 91                  SUB C              ;SUBTRACT OFF READING
01AA 380F                JR C,S2            ;J IF OFF>ON
01AC FD7700              LD (IY),A          ;STORE NEWEST
01AF DD8600              ADD A,(IX)         ;UPDATE TOTAL
01B2 3003                JR NC,S1
01B4 DD3401              INC (IX+1)         ;ADJUST HIGH BYTE
01B7 DD7700     S1       LD (IX),A          ;STORE NEW TOTAL
01BA C9                  RET
01BB FD360000   S2       LD (IY),0
01BF C9                  RET
                ;
01C0 FD7E01     THRESH   LD A,(IY+1)        ;HIGH BYTE OF THRON
01C3 DDBE01              CP (IX+1)          ;(THRON-TOTAL)
01C6 380A                JR C,T1            ;'ON' STATE
01C8 2011                JR NZ,T2
01CA FD7E00              LD A,(IY)          ;LOW BYTE OF THRON
```

```
01CD DDBE00                  CP (IX)           ;(THRON-TOTAL)
01D0 3009                    JR NC,T2
01D2 3A0718        T1        LD A,(CODE)       ;'ON' STATE
01D5 07                      RLCA
01D6 3C                      INC A
01D7 320718                  LD (CODE),A
01DA C9                      RET
01DB FD7E11        T2        LD A,(IY+17)      ;HIGH BYTE OF THROFF
01DE DDBE01                  CP (IX+1)         ;(OFFLEV-TOTAL)
01E1 300A                    JR NC,T3          ;'OFF STATE'
01E3 2010                    JR NZ,INVAL       ;INVALID STATE
01E5 FD7E10                  LD A,(IY+16)      ;LOW BYTE OF THROFF
01E8 DDBE00                  CP (IX)
01EB 3808                    JR C,INVAL
01ED 3A0718        T3        LD A,(CODE)
01F0 07                      RLCA
01F1 320718                  LD (CODE),A
01F4 C9                      RET
01F5 3A0218        INVAL     LD A,(FLAG)
01F8 CBC7                    SET 0,A
01FA 320218                  LD (FLAG),A
01FD C9                      RET
                   ;
01FE 3A0218        DECODE    LD A,(FLAG)
0201 CB47                    BIT 0,A
0203 2801                    JR Z,TARGET
0205 C9                      RET
                   ;
0206 3A3B00        TARGET    LD A,(LEN)
0209 4F                      LD C,A
020A 0600                    LD B,0
020C 2A0A18                  LD HL,(TAB4)
020F 3A0718                  LD A,(CODE)
0212 EDB1                    CPIR
0214 3A0218                  LD A,(FLAG)
0217 2005                    JR NZ,R1
0219 CBD7                    SET 2,A
021B 320218                  LD (FLAG),A
021E C9            R1        RET
                   ;
021F 3E10          SAMIC     LD A,10H          ;0001/0000
0221 320518                  LD (ADATA),A      ;INIT CONDS FOR ADATA
0224 DD215018                LD IX,TOTAL       ;IX^TOTAL(0)
0228 FD210F18                LD IY,OLDEST      ;IY^OLDEST(0)
022C 213100                  LD HL,LOCTAB      ;DECODE TABLE=LOCATION
022F 220A18                  LD (TAB4),HL      ; MARKER TEMPLATES
0232 0654                    LD B,84           ;CLEAR TOTAL &
0234 210D18                  LD HL,IYREF       ;OLDEST TABLES
0237 3600          CLR       LD (HL),0
0239 23                      INC HL
023A 10FB                    DJNZ CLR
023C 3A0218                  LD A,(FLAG)
023F E610                    AND 10H
0241 320218                  LD (FLAG),A
0244 3E00                    LD A,0
0246 320718                  LD (CODE),A
0249 C9                      RET
                   ;
024A 3E10          [illegible]    LD A,10H
024C 320518                  LD (ADATA),A
```

```
PolyZap V2.0     PC6/1 FOR MK1 SENSOR 2-3-83 A. BAKER

024F 0654                 LD B,84
0251 210D18               LD HL,IYREF
0254 3600        ECLR     LD (HL),0
0256 23                   INC HL
0257 10FB                 DJNZ ECLR
0259 DD215018             LD IX,TOTAL
025D FD210F18             LD IY,OLDEST
0261 212300               LD HL,EDGE
0264 220A18               LD (TAB4),HL
0267 3A0218               LD A,(FLAG)
026A E610                 AND 10H
026C 320218               LD (FLAG),A
026F 3E00                 LD A,0
0271 320718               LD (CODE),A
0274 C9                   RET
                 ;
0275 3A0518      SAMPLE   LD A,(ADATA)
0278 CBF7                 SET 6,A            ;LED ON
027A D308                 OUT (F5),A
027C 320518               LD (ADATA),A
027F CD6001               CALL BLINK
0282 3A0518               LD A,(ADATA)
0285 CBB7                 RES 6,A            ;LED OFF
0287 D308                 OUT (F5),A
0289 320518               LD (ADATA),A
028C 0608                 LD B,8
028E FD220D18             LD (IYREF),IY      ;STORE IY^
0292 CD7E01      LOOP1    CALL MUX
0295 CD9401               CALL NEWTOT
0298 DD23                 INC IX
029A DD23                 INC IX
029C FD23                 INC IY
029E 10F2                 DJNZ LOOP1
02A0 CD6001               CALL BLINK
02A3 FD210200             LD IY,THRON        ;INITIALIZE POINTER
02A7 FDE5                 PUSH IY            ;FOR THESHOLDS
02A9 FD2A0D18             LD IY,(IYREF)      ;IY:=IY-8
02AD DD215018             LD IX,TOTAL        ;RESET IX^
02B1 0608                 LD B,8
02B3 CD7E01      LOOP2    CALL MUX
02B6 CDA601               CALL SUBT
02B9 FDE3                 EX (SP),IY
02BB CDC001               CALL THRESH
02BE FD23                 INC IY
02C0 FD23                 INC IY
02C2 FDE3                 EX (SP),IY
02C4 DD23                 INC IX
02C6 DD23                 INC IX
02C8 FD23                 INC IY
02CA 10E7                 DJNZ LOOP2
02CC DD3400               INC (IX)           ;CT FOR RUNNING MEAN
02CF 3E04                 LD A,4
02D1 DDBE00               CP (IX)
02D4 200A                 JR NZ,J5
02D6 DD360000             LD (IX),0          ;RESET IY^ COUNT
02DA FD210F18             LD IY,OLDEST       ;RESET IY
02DE 1804                 JR J6
02E0 00          J5       NOP
02E1 00                   NOP
02E2 00                   NOP
```

```
PolyZap V2.0      PC671 FOR MK1 SENSOR 2-3-83 A. BAKER

02E3 00                   NOP
02E4 DDE1         J6      POP IX
                  ;                        ;TAKE THRON ADDRESS
02E6 DD215018                               OFF STACK
02EA CDFE01               LD IX,TOTAL      ;RESET IX^
02ED C9                   CALL DECODE
                          RET
                  ;
02EE 76           INSTRS  HALT
02EF 2A7118               LD HL.(INFO)     ;WAIT FOR V.E. TO INT
02F2 ED5B8218             LD DE.(INCHK)
02F6 37                   SCF
02F7 3F                   CCF
02F8 ED52                 SBC HL,DE
02FA 28F2                 JR Z,INSTRS      ;CHECK FOR CHANGE
                  ;
02FC 2A7118               LD HL.(INFO)     ;WITH INSTRUCTIONS
02FF 3A8218               LD A.(INCHK)     ;CHANGE IN LOCATION?
0302 BD                   CP L
0303 228218               LD (INCHK),HL
0306 2006                 JR NZ,NEWLOC
0308 E1                   POP HL
0309 21E200               LD HL.CLOP       ;LOCN UNCHANGED ->
030C E5                   PUSH HL          ;STOP AT NEXT EDGE
030D C9                   RET
                  ;
030E 7D           NEWLOC  LD A,L
030F 328118               LD (LOCSTP),A
0312 CB44                 BIT 0,H          ;WHICH EDGE?
0314 210218               LD HL,FLAG
0317 2804                 JR Z,I1
0319 CBE6                 SET 4,(HL)
031B 1802                 JR I2
031D CBA6         I1      RES 4,(HL)
031F E1           I2      POP HL
0320 218100               LD HL,OPLOP
0323 E5                   PUSH HL          ;RET TO OPEN LOOP
0324 C9                   RET
                  ;
0325 217118       CHK     LD HL,INFO
0328 3A8218               LD A,(INCHK)
032B BE                   CP (HL)
032C C8                   RET Z            ;RETURN IF NO CHANGE
                  ;
032D E1                   POP HL
032E 218100               LD HL,OPLOP      ;RET ADDR = OPLOP IF
0331 E5                   PUSH HL          ;CHANGE
0332 2A7118               LD HL,(INFO)
0335 228218               LD (INCHK),HL
0338 7D                   LD A,L
0339 328118               LD (LOCSTP),A    ;UPDATE DESIRED LOCN
033C CB44                 BIT 0,H          ;WHICH EDGE?
033E 210218               LD HL,FLAG
0341 2803                 JR Z,J7
0343 CBE6                 SET 4.(HL)
0345 C9                   RET
                  ;
0346 CBA6         J7      RES 4.(HL)
0348 C9                   RET
                  ;
0349 7F           [illegible] LD A,A
```

```
034A C9
                          RET
                 ;
034B 3A0218      FLAGS    LD A,(FLAG)
034E E6F8                 AND 0F8H
0350 320218               LD (FLAG),A          ;RES MATCH & VALID FLAGS
0353 3E00                 LD A,0
0355 320718               LD (CODE),A          ;CLEAR CODE ARRAY
0358 C9                   RET
                 ;
0359 3A6218      LOCUP    LD A,(OUTFO+1)
035C D306                 OUT (F4),A
035E 3E85                 LD A,85H             ;CCW:TX
0360 CD7E03               CALL COMOUT
0363 76                   HALT
0364 C9                   RET
                 ;
                 ;
                 ;******** COMMUNICATIONS HANDLER ********
                 ;
0365 F5          CINT     PUSH AF
0366 C5                   PUSH BC
0367 D5                   PUSH DE
0368 E5                   PUSH HL
0369 CDEC03               CALL COMIN
036C E1                   POP HL
036D D1                   POP DE
036E C1                   POP BC
036F F1                   POP AF
0370 FB                   EI
0371 ED4D                 RETI
                 ;
0373 3EFF        CINIT    LD A,0FFH
0375 D301                 OUT (P1C),A
0377 D301                 OUT (P1C),A
0379 D303                 OUT (P2C),A
037B D303                 OUT (P2C),A
037D C9                   RET
                 ;
037E 320418      COMOUT   LD (CCW),A
0381 060A        WTTEN    LD B,10
0383 DB00        WAITN    IN A,(P1)
0385 E681                 AND 81H
0387 FE01                 CP 1
0389 20F6                 JR NZ,WTTEN
038B 10F6                 DJNZ WAITN
                 ;
038D 3EFF                 LD A,0FFH
038F D301                 OUT (P1C),A
0391 3EFE                 LD A,0FEH
0393 D301                 OUT (P1C),A
0395 D300                 OUT (P1),A
0397 0600                 LD B,0             ;WAIT FOR BITS 1-6 HIGH
0399 DB00        BITS16   IN A,(P1)
039B E67E                 AND 7EH
039D FE7E                 CP 7EH
039F 280D                 JR Z VEREP
03A1 DDE5                 PUSH IX
03A3 DDE1                 POP IX
03A5 FDE5                 PUSH IY
03A7 FDE1                 POP IY
```

```
03A9 10EE                   DJNZ BITS16
03AB C38704                 JP MNSTCK

                   ;
03AE 3EFF          VEREF    LD A,0FFH          ;SET BIT 1 LOW
03B0 D301                   OUT (P1C),A
03B2 3EFA                   LD A,0FAH          ;BIT 2 & 0 OUTPUT
03B4 D301                   OUT (P1C),A
03B6 D300                   OUT (P1),A         ;BIT 2 & 0 LOW
03B8 DB00                   IN A,(P1)          ;CHECK FOR HIGHER
03BA E602                   AND 2              ;PRIORITY SUB-SYSTEM
03BC FE02                   CP 2
03BE 2023                   JR NZ,AGAIN
03C0 3EFF                   LD A,0FFH
03C2 D303                   OUT (P2C),A
03C4 3E00                   LD A,0
03C6 D303                   OUT (P2C),A
03C8 3A0418                 LD A,(CCW)         ;SEND CCW
03CB D302                   OUT (P2),A
                   ;
03CD 3EFF                   LD A,0FFH          ;SET BITS 1-6 AS I/P
03CF D301                   OUT (P1C),A
03D1 3EFE                   LD A,0FEH
03D3 D301                   OUT (P1C),A
03D5 0600                   LD B,0
03D7 DB00          FINLP    IN A,(P1)          ;WAIT FOR BITS 1-6 LOW
03D9 E67E                   AND 7EH
03DB CA7303                 JP Z CINIT
03DE 10F7                   DJNZ FINLP
03E0 C38704                 JP MNSTCK
03E3 CD7303        AGAIN    CALL CINIT         ;RE-INIT PIOs AND
03E6 3A0418                 LD A,(CCW)         ;TRY AGAIN
03E9 CD7E03                 CALL COMOUT
                   ;
03EC DB02          COMIN    IN A,(P2)
03EE CB47                   BIT 0,A
03F0 2843                   JR Z,RX
                   ;
03F2 CD8D04        TX       CALL REPLY
03F5 3EFF                   LD A,0FFH
03F7 D301                   OUT (P1C),A
03F9 3EF4                   LD A,0F4H
03FB D301                   OUT (P1C),A
03FD D300                   OUT (P1),A
                   ;
03FF 3EFF                   LD A,0FFH
0401 D303                   OUT (P2C),A
0403 3E00                   LD A,0
0405 D303                   OUT (P2C),A
                   ;
0407 216118                 LD HL,OUTFO
040A 0E10                   LD C,16
040C 7E            SEND     LD A,(HL)
040D D302                   OUT (P2),A
040F 3E02                   LD A,2
0411 D300                   OUT (P1),A         ;SET DATA VALID
0413 1E04                   LD E,4
0415 CD7F04                 CALL BITNZ         ;WAIT FOR DATA RFT
                   ;
0418 3E01                   LD A,1
041A B9                     CP C               ;CHECK FOR LAST BYTE
```

```
041B 280D                JR Z,EXIT
                ;
041D 3E00                LD A,0
041F D300                OUT (P1),A          ;RESET DATA VALID
0421 23                  INC HL
0422 1E04                LD E,4
0424 CD7504              CALL BITZ
0427 0D                  DEC C
0428 20E2                JR NZ,SEND
                ;
042A 3E08       EXIT     LD A,8
042C D300                OUT (P1),A          ;SEND DATA FINISHED
042E 060E                LD B,14
0430 10FE       EXLOP    DJNZ EXLOP
0432 C37303              JP CINIT
                ;
0435 CD8D04     RX       CALL REPLY
0438 3EFF                LD A,0FFH
043A D301                OUT (P1C),A
043C 3EFA                LD A,0FAH
043E D301                OUT (P1C),A
0440 D300                OUT (P1),A
0442 3EFF                LD A,0FFH
0444 D303                OUT (P2C),A
0446 D303                OUT (P2C),A
                ;
0448 217118              LD HL,INFO
044B 0E10                LD C,16
044D 1E02       GET      LD E,2
044F CD7F04              CALL BITNZ
                ;
0452 DB02                IN A,(P2)
0454 77                  LD (HL),A
0455 3E04                LD A,4
0457 D300                OUT (P1),A
0459 1E02                LD E,2
045B CD7504              CALL BITZ
                ;
045E DB00                IN A,(P1)
0460 CB5F                BIT 3,A
0462 2008                JR NZ,VERIN
0464 3E00                LD A,0
0466 D300                OUT (P1),A
0468 23                  INC HL
0469 0D                  DEC C
046A 20E1                JR NZ,GET
                ;
046C 3E01       VERIN    LD A,1
046E B9                  CP C
046F C49C04              CALL NZ,ERROR
0472 C37303              JP CINIT
                ;
0475 0600       BITZ     LD B,0
0477 DB00       BTZLP    IN A,(P1)
0479 A3                  AND E
047A C8                  RET Z
047B 10FA                DJNZ BTZLP
047D 1808                JR MNSTCP
                ;
047F 0600       BITNZ    LD B,0
```

```
0481 DB00        BNZLP   IN A,(P1)
0483 A3                  AND E
0484 C0                  RET NZ
0485 10FA                DJNZ BNZLP
                 ;
0487 213D00      MNSTCK  LD HL,GO
048A E5                  PUSH HL
048B ED4D                RETI
                 ;
048D 3EFF        REPLY   LD A,0FFH
048F D301                OUT (P1C),A
0491 3EFE                LD A,0FEH
0493 D301                OUT (P1C),A
0495 D300                OUT (P1),A
0497 1E80                LD E,80H
0499 C37504              JP BITZ
                 ;
049C 3E82        ERROR   LD A,82H             ;CCW:1000/0100
049E CD7E03              CALL COMOUT          ;ASK FOR MESSAGE
04A1 76                  HALT                 ;REPEAT
04A2 C9                  RET
                 ;
04A3 3E64        WAIT    LD A,100             ;WAIT 50mS
04A5 069F        WAIT1   LD B,159
04A7 10FE        WAIT2   DJNZ WAIT2
04A9 3D                  DEC A
04AA 20F9                JR NZ WAIT1
04AC C9                  RET
                 ;
                 ;***** END OF COMMUNICATIONS HANDLER *****
                 ;
                 ;************ VARIABLES ************
                 ;
1800                     ORG RAMTOP
1800 + 0002      ISRAD   DEFS 2
1802 + 0001      FLAG    DEFS 1
1803 + 0001      REF     DEFS 1
1804 + 0001      CCW     DEFS 1
1805 + 0001      ADATA   DEFS 1
1806 + 0001      STATUS  DEFS 1
1807 + 0001      CODE    DEFS 1
1808 + 0002      LAST    DEFS 2
180A + 0002      TAB4    DEFS 2
180C + 0001      LOCN    DEFS 1
                 ;
180D + 0002      IYREF   DEFS 2
180F + 0040      OLDEST  DEFS 64
184F + 0001              DEFS 1
1850 + 0010      TOTAL   DEFS 16
1860 + 0001              DEFS 1
1861 + 0010      OUTFO   DEFS 16
1871 + 0010      INFO    DEFS 16
1881 + 0001      LOCSTP  DEFS 1
1882 + 0002      INCHI   DEFS 2
1884                     END
```

```
PolyZap V2.0     PC671 FOR MFI SENSOR 2-3-83 A. BAKER

ACT2      0106     ACTION    00AC     ADATA     1805
AGAIN     03E3     BITN7     047F     BITS16    0399
BITZ      0475     BLINK     0160     BNZLP     0481
BITZLP    0477     C3        0102     CCW       1804
CHK       0325     CINIT     0373     CINT      0365
CLOP      00E2     CLR       0237     CLR1      0051
CODE      1807     COMIN     03EC     COMOUT    037E
DECODE    01FE     ECLR      0254     EDGE      0023
EDGIC     024A     EDGSAM    00F8     EDIM      003A
EDLEN     003C     EOC       0177     ERROR     049C
EXIT      042A     EXLOP     0430     FINLP     03D7
FLAG      1802     FLAGS     034B     GET       044D
GO        003D     I1        031D     I2        031F
INCHK     1882     INFO      1871     INSTRS    02EE
INVAL     01F5     ISRAD     1800     IYREF     180D
J3        00A4     J5        02E0     J6        02E4
J7        0346     J8        00D0     L1        0045
LAST      1808     LEN       003B     LOCN      180C
LOCSIP    1881     LOCTAB    0031     LOCUP     0359
LOOP1     0292     LOOP2     02B3     LOST      0349
MNSTCH    0487     MUX       017E     N1        019F
NEWLOC    030E     NEWTOT    0194     O1        0097
OLDEST    180F     OPLOP     0081     OUTFO     1861
P1        0000     P1C       0001     P2        0002
P2C       0003     P3        0004     P3C       0005
P4        0006     P4C       0007     P5        0008
P5C       0009     P6        000A     P6C       000B
R1        021E     RAMEND    1C00     RAMTOP    1800
REF       1803     REPLY     04BD     RX        0435
S1        01B7     S2        01BB     SAM       009A
SAMIC     021F     SAMPLE    0275     SEND      040C
SETPT     0123     SLOW1     0022     SROM      0000
STATUS    1806     SUBT      01A6     T1        01D2
T2        01DB     T3        01ED     TAB4      180A
TARGET    0206     THRESH    01C0     THROFF    0012
THRON     0002     TOTAL     1850     TX        03F2
VEREF     03AE     VERIN     046C     WAIT      04A3
WAIT1     04A5     WAIT2     04A7     WAITN     0383
WTTEN     0381     X         0000
```

B.2 Traction Control Program.

This program obtains a reference speed from one of three sources, compares the reference with the current speed and controls the duty cycle applied to the drive motors according to the result of this comparison. It also controls motor direction. The program skeleton was written for a closed loop proportional speed control system using a speed sensor on the vehicle as was envisaged for 'cruise control' on the final product. For the early stages of Traction development, the skeleton was modified for open loop control by replacing the speed sensor input by the current duty cycle demand. It is this modified version that is described here.

The speed reference may be provided by Vehicle Executive (either directly or from the Manual Control Unit), or Position. Communications with Executive follows the standard protocol. Program operation falls into two phases: obtaining the reference and executing the control loop. Peripheral to these are the initialisation phase and the communications handler.

B.2.1 Program Layout.

	Memory address
Program Start (start of PROM)	0000H
Start of RAM	1800H
End of RAM	1C00H
Bottom of Stack	1C00H
Start of Tables	0400H
Interrupt Mode	2
Interrupt Service Routine Address held at	1800H

Initialisation:

Clear RAM.

Clear interrupts.

Load interrupt vectors and addresses.

Initialise PIOs.

Enable interrupts.

Enable field duty cycle output and wait for current to build up.

Main Program Loop:

This consists of six sub-routine calls which are continuously repeated:

GETREF	interrogate speed reference sources and store appropriate reference for control loop.
SPEED	read current speed from sensor.
SUM	subtract speed from reference to find error.
COMP	transform error into duty cycle demand and output to CTC.

During program operation, a pulse must be sent to the trip relay monostable at least once every 200ms. This is achieved by the routine MONO which is called twice during the main loop and is also invoked in every WAIT loop called by the various sub-routines.

At any time, the program may be interrupted with new instructions from Executive. Storage of these instructions is handled by the interrupt routine.

B.2.2 Sub-routines.

WAIT1

Variable delay specified by the contents of registers D and E. If E holds 149, the delay is 0.5 x D ms. This routine also calls MONO.

Entry Requirements: delay variables in D and E.

Exit Status: registers D and E modified.

MONO

Sends a high pulse on bit 7 of PIO 1 port B (monostable trigger).

Entry Requirements: none.

Exit Status: register A modified.

INIT

Sets PIOs to initial conditions, programs the CTC to produce the duty cycle envelope and initialises the field. All variables are cleared.

Entry Requirements: none.

Exit Status: registers A, B, H, L modified,
PIO 1 port A: bits 0-4 inputs
bits 5,6 output low
bit 7 output high (armature disabled)
PIO 1 port B: bits 0-6 inputs
bit 7 output low
CTC channel 0 programmed to output a pulse every $\frac{2}{3}$ms to give a 1.5kHz duty cycle envelope.
CTC channel 1 initialised as a counter and reset.
CTC channel 2 programmed to produce 98% field duty cycle.
Variable ADATA set such that armature and field are disabled.

GETREF

Reads the speed reference sources, decides which one has highest priority and converts the relevant reference value into a 2's complement 8-bit word which is stored as variable REF (bit 7 denotes direction $\overline{\text{Forward}}$/Reverse).

Reference sources:

a) Vehicle Executive - held in first two bytes of table INFO

byte 1 bit 7 = $\overline{\text{Forward}}$/Reverse
bits 1-6 = speed if manual
bit 0 = manual/$\overline{\text{auto operation}}$
byte 2 speed if auto (2's complement)

b) Position Control: PIO 1 port B

bit 6 = Priority to pos^n ctl/Priority to V. Exec.
bit 5 = $\overline{\text{Forward}}$/Reverse
bits 0-4 = speed if pos^n has priority.

NB. If byte 2 of INFO = 0, then emergency stop must be initiated. This program also reads bit 4, PIO 1 port A - the radio control input. If this is high, the armature output is disabled and the reference sources re-read.

Entry Requirements: valid speed requests in INFO and on port B, PIO 1

Exit Status: variable REF updated. In event of a radio control initiated stop, bit 7 of variable ADATA is set, and variable DCYCLE is set to zero.

Registers A, D, E, H, and L modified.

SUM

Current speed is subtracted from current speed reference to yield the error signal. This error is limited to the range -16 to +16 which effectively imposes a runout on acceleration.

Entry Requirements: valid data in variables SPEED and REF.

Exit Status: variable Error updated, registers A and D modified.

COMP

The error signal is transformed into a duty cycle demand through a look-up table. This value is then limited to the range -80% to +80%. If a change of direction is demanded, the sub-routine FLIP is called. The direction bit of the duty cycle demand word is then reset and this positive value is loaded as the new time constant for CTC channel 1. For the case of maximum error, a 200ms delay is executed through WAIT1 to produce the runout.

Entry Requirements: valid data in variables SPEED and ERROR.

Exit Status: variables DCYCLE and DCDEM updated, variable ADATA and PIO 1 port A status altered with respect to field and armature enable and direction. CTC channel 1 time constant re-programmed. Registers A, D, E, H and L modified.

NB. For open-loop operation, the current duty cycle (held in variable DCDEM) is added to the duty cycle demand, due to the nature of the look-up table.

SENSOR

Currently, this sub-routine simulates the sensor input by using the value in variable DCYCLE as the speed reading. This is used to access a look-up table and the result stored in variable SPEED as a 2's complement number of the same form as REF.

Entry Requirements: valid data in DCYCLE.

Exit Status: variable SPEED updated, registers A, H, L modified.

FLIP

Disables armature and field outputs. A 100ms delay is called to allow the field current to fall to zero, then the field bit in ADATA is complemented and output on PIO 1 port A. The field duty cycle is re-programmed and enabled, followed by another 100ms delay to allow the field current to build up.

Entry Requirements: none.

Exit Status: variable ADATA updated and PIO 1 port A status updated, variable DCYCLE set to zero. Registers A, D, E modified.

B.2.3 Communications Handler.

See section B.1.3.

```
PolyZap V2.0      TRACTION 3  28-1-83

                  ;T1
                  ;WITH COMM          WITH BRAKING
                                     ==============
                  ;****************************************
                  ;* TRACTION CONTROL FOR TUG TO RUN WITH *
                  ;* BOARD 45023 AND POSPROM 6 IN POS CTL *
                  ;* AND COMMUNICATE WITH VEHICLE EXEC    *
                  ;* A.BAKER                              *
                  ;* THIS VERSION                 9-7-82  *
                  ;*****************************  26-8-82 *
                  ;****************************************
0000              P1. EQU 0
0001              P1C EQU 1
0002              P2 EQU 2                ;COMMUNICATIONS
0003              P2C EQU 3               ;    PIO
                  ;
0004              P3 EQU 4
0005              P3C EQU 5
0006              P4 EQU 6                ;PIO 1
0007              P4C EQU 7
                  ;
0008              CT0 EQU 8
0009              CT1 EQU 9               ;CTC
000A              CT2 EQU 0AH
                  ;
0047              CTCLD EQU 47H
                  ;
0000              SROM EQU 0H
1800              RAMTOP EQU 1800H
1C00              RAMEND EQU 1C00H
                  ;
                  ;**************************************
                  ;                PROGRAM
                  ;**************************************
                  ;
0000               ORG SROM
0000               IDNT $,$
0000 31001C       BEGIN LD SP,RAMEND
0003 210900        LD HL,CLRI
0006 E5            PUSH HL
0007 ED4D          RETI               ;CLEAR INTERRUPTS
0009 21E201       CLRI LD HL,CINT
000C 220018        LD (ISRAD),HL
000F 3E18          LD A,18H
0011 ED47          LD I,A             ;HIGH BYTE OF ISRAD
0013 CD6700        CALL INIT         ;INITIALIZE PIO's
0016 CDA800        CALL CINIT
0019 ED5E          IM 2
001B 3E00          LD A,0
001D D301          OUT (P1C),A        ;LOW BYTE OF ISRAD
001F 3E77          LD A,77H
0021 D301          OUT (P1C),A        ;AND & HIGH
0023 3E77          LD A,77H           ;0111/0111
0025 D301          OUT (P1C),A        ;INTERRUPT MASK
0027 3E83          LD A,83H
0029 D301          OUT (P1C),A         ;ENABLE PIO INTERRUPTS
002B FB            EI
                  ;
002C 3A0218        LD A,(ADATA)
002F CBB7          RES 6,A
0031 D304          OUT (P3),A             ;ENABLE FIELD
```

```
0033 320218         LD (ADATA),A
0036 1664           LD D,100
0038 1E95           LD E,149
003A CD5100         CALL WAIT1
                   ;                         ;WAIT FOR FIELD
003D CD5E00        START CALL MONO
0040 CDB300         CALL GETREF
0043 CD9101         CALL SENSOR
0046 CD5E00         CALL MONO
0049 CDFC00         CALL SUM
004C CD1A01         GALL COMP
004F 18EC           JR START
                   ;
                   ;
                   ;
                   ;****************************************
                   ;              SUB-ROUTINES
                   ;****************************************
                   ;
0051 C5            WAIT1 PUSH BC
0052 4A             LD C,D
0053 43            W1 LD B,E
0054 10FE          W2 DJNZ W2
0056 0D             DEC C
0057 CD5E00         CALL MONO
005A 20F7           JR NZ,W1
005C C1             POP BC
005D C9             RET
005E 3E80          MONO LD A,80H
0060 D306           OUT (P4),A
0062 3E00           LD A,0
0064 D306           OUT (P4),A
0066 C9             RET
                   ;
                   ;
0067 3EFF          INIT LD A,0FFH
0069 D305           OUT (P3C),A
006B 3E1F           LD A,1FH            ;000I/IIII
006D D305           OUT (P3C),A
006F 3E80           LD A,80H
0071 D304           OUT (P3),A
                   ;
0073 3EFF           LD A,0FFH
0075 D307           OUT (P4C),A
0077 3E7F           LD A,07FH           ;0III/IIII
0079 D307           OUT (P4C),A
007B 3E00           LD A,0
007D D306           OUT (P4),A
                   ;
007F 3E47           LD A,47H
0081 D308           OUT (CT0),A
0083 3E0F           LD A,15
0085 D308           OUT (CT0),A       ;2/3mS DC ENVELOPE
                   ;
0087 3E47           LD A,47H
0089 D30A           OUT (CT2),A
008B 3E62           LD A,98
008D D30A           OUT (CT2),A       ;98% FIELD
                   ;
008F 3EC0           LD A,0C0H
```

```
PolyZap V2.0      TRACTION 3  28-1-83

0091 320218        LD (ADATA),A
                  ;                          ;ARMATURE OFF, FIELD OFF
0094 3E47          LD A,47H
0096 D309          OUT (CT1),A
0098 3E01          LD A,1
009A D309          OUT (CT1),A             ;RESET CTC 1
                  ;
009C 210318        LD HL,SPEED
009F 061E          LD B,30
00A1 3E00          LD A,0
00A3 77           CLRAM LD (HL),A
00A4 23            INC HL
00A5 10FC          DJNZ CLRAM
00A7 C9            RET
                  ;
                  ;
00A8 3EFF         CINIT LD A,OFFH
00AA D301          OUT (P1C),A
00AC D301          OUT (P1C),A
00AE D303          OUT (P2C),A
00B0 D303          OUT (P2C),A
00B2 C9            RET
                  ;
                  ;
00B3 210518       GETREF LD HL,INFO
00B6 7E            LD A,(HL)               ;CHECK AUTO/MAN
00B7 CB46          BIT 0,(HL)
00B9 2025          JR NZ,EXIT
00BB 23            INC HL
00BC 7E            LD A,(HL)
00BD FE00          CP 0                    ;CHECK FOR EM STOP
00BF 2004          JR NZ,NSTP
00C1 3EFB          LD A,0FBH               ;-5% D.C.
00C3 181B          JR EXIT
00C5 DB04         NSTP IN A,(P3)
00C7 CB67          BIT 4,A                 ;READ R/C
00C9 2019          JR NZ,STOP
00CB DB06          IN A,(P4)
00CD CB77          BIT 6,A                 ;READ V EXEC/POSN
00CF 2803          JR Z,PREF
00D1 7E            LD A,(HL)
00D2 180C          JR EXIT
                  ;
00D4 E63F         PREF AND 3FH            ;0011/1111
00D6 CB6F          BIT 5,A
00D8 2806          JR Z,EXIT
00DA CBAF         REV RES 5,A
00DC ED44          NEG
00DE 1800          JR EXIT
                  ;
00E0 321518       EXIT LD (REF),A
00E3 C9            RET
                  ;
                  ;
00E4 3A0218       STOP LD A,(ADATA)
00E7 CBFF          SET 7,A
00E9 D304          OUT (P3),A              ;STOP VEHICLE
00EB 320218        LD (ADATA),A
00EE 3E00          LD A,0
00F0 321718        LD (DCYCLE),A
```

```
00F3 1E95          LD E,149
00F5 1601          LD D,1
00F7 CD5100        CALL WAIT1
00FA 18B7          JR GETREF
                  ;
                  ;*** END OF GETREF ***
                  ;
00FC 3A0318       SUM LD A,(SPEED)
00FF 57            LD D,A
0100 3A1518        LD A,(REF)
0103 92            SUB D
0104 FE80          CP 80H
0106 3008          JR NC,MINUS
0108 FE10          CP 10H
010A 380A          JR C,STORE
010C 3E10          LD A,10H              ;MAX +VE ERROR = 16
010E 1806          JR STORE
0110 FEF0         MINUS CP 0F0H
0112 3002          JR NC,STORE
0114 3EF0          LD A,0F0H             ;MIN ERROR = -16
0116 321618       STORE LD (ERROR),A
0119 C9            RET
                  ;
011A 21B104       COMP LD HL,CPTAB
011D 3A1618        LD A,(ERROR)
0120 85            ADD A,L
0121 6F            LD L,A
0122 3A0318        LD A,(SPEED)
0125 86            ADD A,(HL)
                  ;
0126 CB7F          BIT 7,A
0128 2016          JR NZ,REV3
012A FE50          CP 80
012C 3802          JR C,J5
012E 3E50          LD A,80               ;MAX ARM DC=80%
0130 321718       J5 LD (DCYCLE),A
0133 321818        LD (DCDEM),A
0136 3A0218        LD A,(ADATA)
0139 CB6F          BIT 5,A
013B CC9E01        CALL Z,FLIP      ;SWAP FIELD OVER
013E 1816          JR J2
                  ;
0140 FEB0         REV3 CP 0B0H
0142 3002          JR NC,J7
0144 3EB0          LD A,0B0H        ;MIN ARM DC=-80%
0146 321718       J7 LD (DCYCLE),A
0149 ED44          NEG
014B 321818        LD (DCDEM),A
014E 3A0218        LD A,(ADATA)
0151 CB6F          BIT 5,A
0153 C49E01        CALL NZ,FLIP
                  ;
0156 3A1818       J2 LD A,(DCDEM)
0159 FE00          CP 0
015B 200C          JR NZ,J3
015D 3A0218        LD A,(ADATA)
0160 CBFF          SET 7,A          ;ARM DISABLE
0162 D304          OUT (P3),A
0164 320218        LD (ADATA),A
0167 181F          JR J4
```

```
                        ;
0169 57                 J3 LD D,A
016A 3E45                LD A,45H
016C D309                OUT (CT1),A
016E 3A1818              LD A,(DCDEM)
0171 D309                OUT (CT1),A
0173 3A0218              LD A,(ADATA)     ;SEND OUT DUTY CYCLE
0176 CBBF                RES 7,A
0178 D304                OUT (P3),A
017A 320218              LD (ADATA),A
017D 3A1618              LD A,(ERROR)     ;ARMATURE ENABLE
0180 FE10                CP 16
0182 2805                JR Z,DELAY
0184 FEF0                CP 0F0H                ;RUNOUT IF MAX ERROR
0186 2801                JR Z,DELAY
0188 C9                 J4 RET                  ;RUNOUT IF MIN ERROR
0189 16C8               DELAY LD D,200
018B 1E95                LD E,149
018D CD5100              CALL WAIT1
0190 C9                  RET
                        ;
                        ;
0191 3A1718             SENSOR LD A,(DCYCLE)
0194 215004              LD HL,SPDTAB
0197 85                  ADD A,L
0198 6F                  LD L,A
0199 7E                  LD A,(HL)
019A 320318              LD (SPEED),A
019D C9                  RET
                        ;
                        ;
019E 3A0218             FLIP LD A,(ADATA)
01A1 CBFF                SET 7,A
01A3 CBF7                SET 6,A
01A5 D304                OUT (P3),A          ;KILL ARMATURE & FIELD
01A7 320218              LD (ADATA),A
01AA 3E00                LD A,0
01AC 321718              LD (DCYCLE),A
01AF 1E95                LD E,149
01B1 1664                LD D,100
01B3 CD5100              CALL WAIT1          ;100ms DELAY FOR FIELD
                        ;
01B6 3A0218              LD A,(ADATA)
01B9 CB6F                BIT 5,A             ;TEST FIELD BIT
01BB 2804                JR Z,J6
01BD CBAF                RES 5,A
01BF 1802                JR J8
01C1 CBEF               J6 SET 5,A
01C3 320218             J8 LD (ADATA),A
01C6 D304                OUT (P3),A          ;FLIP FIELD OVER
                        ;
01C8 3E45                LD A,45H
01CA D30A                OUT (CT2),A         ;REPROG FIELD CTC
01CC 3E62                LD A,98
01CE D30A                OUT (CT2),A         ;SEND D CYCLE
01D0 3A0218              LD A,(ADATA)
01D3 CBB7                RES 6,A
01D5 320218              LD (ADATA),A
01D8 D304                OUT (P3),A          ;ENABLE FIELD
01DA 1E95                LD E,149
```

```
01DC 1664       LD D,100
01DE CD5100     CALL WAIT1          ;FIELD DELAY
01E1 C9         RET
               ;
               ;
               ;*** COMMUNICATIONS HANDLER ***
               ;
01E2 F5        CINT PUSH AF
01E3 C5         PUSH BC
01E4 D5         PUSH DE
01E5 E5         PUSH HL
01E6 CDF001     CALL COMIN
01E9 E1         POP HL
01EA D1         POP DE
01EB C1         POP BC
01EC F1         POP AF
01ED FB         EI
01EE ED4D       RETI
               ;
01F0 DB02      COMIN IN A,(P2)
01F2 CB47       BIT 0,A
01F4 C0         RET NZ             ;RET IF TX
               ;
01F5 CD5702    RX CALL REPLY
01F8 3EFF       LD A,0FFH
01FA D301       OUT (P1C),A
01FC 3EFA       LD A,0FAH
01FE D301       OUT (P1C),A
0200 D300       OUT (P1),A
0202 3EFF       LD A,0FFH
0204 D303       OUT (P2C),A
0206 D303       OUT (P2C),A        ;SET UP COMM PIO FOR RX
               ;
0208 210518     LD HL,INFO
020B 0E10       LD C,16
020D 1E02      GET LD E,2
020F CD3F02     CALL BITNZ         ;WAIT FOR DATA VALID
0212 DB02       IN A,(P2)
0214 77         LD (HL),A          ;STORE INFO
0215 3E04       LD A,4
0217 D300       OUT (P1),A         ;SEND DATA RECEIVED
0219 1E02       LD E,2
021B CD3502     CALL BITZ          ;WAIT FOR NO DATA VALID
021E DB00       IN A,(P1)
0220 CB5F       BIT 3,A            ;TEST FOR DATA FINISHED
0222 2008       JR NZ,VERIN
0224 3E00       LD A,0
0226 D300       OUT (P1),A         ;RESET DATA RECEIVED
0228 23         INC HL
0229 0D         DEC C
022A 20E1       JR NZ,GET          ;GET NEXT BYTE OF INFO
               ;
022C 3E01      VERIN LD A,1        ;CHECK 16 BYTES RECEIVE
022E B9         CP C
022F C46602     CALL NZ,ERR
0232 C3AB00     JP CINIT           ;SET COMM PIO TO I.C.'S
               ;
0235 0600      BITZ LD B,0         ;TEST BIT(E) FOR ZERO
0237 DB00      BTZLP IN A,(P1)
0239 A3         AND E
```

```
023A C8           RET Z
023B 10FA         DJNZ BTZLP
023D 1808         JR MNSTCK           ;RESET IF V EXEC DIES
                 ;
023F 0600        BITNZ LD B,0         ;TEST BIT(E) FOR ONE
0241 DB00        BNZLP IN A,(P1)
0243 A3           AND E
0244 C0           RET NZ
0245 10FA         DJNZ BNZLP
                 ;
0247 3A0218      MNSTCK LD A,(ADATA);STOP VEHICLE & RESET
024A CBFF         SET 7,A             ;PROG IF NO REPLY FROM
024C D304         OUT (P3),A          ;V. EXEC
024E 320218       LD (ADATA),A
0251 210000       LD HL,BEGIN
0254 E5           PUSH HL
0255 ED4D         RETI
                 ;
0257 3EFF        REPLY LD A,0FFH
0259 D301         OUT (P1C),A
025B 3EFE         LD A,0FEH
025D D301         OUT (P1C),A
025F D300         OUT (P1),A          ;SEND LOW ON BIT0,PORT2
0261 1E80         LD E,80H
0263 C33502       JP BITZ             ;WAIT FOR LOW BIT 7
                 ;
0266 3E88        ERR LD A,88H         ;CCW
0268 CD6D02       CALL COMOUT         ;ASK FOR MESSAGE REPEAT
026B 76           HALT
026C C9           RET
                 ;
026D 320418      COMOUT LD (CCW),A
0270 060A        WTTEN LD B,10
0272 DB00        WAIT IN A,(P1)
0274 E681         AND 81H
0276 FE01         CP 1               ;WAIT FOR PERMISSION
0278 20F6         JR NZ,WTTEN        ;TO SEND CCW
027A 10F6         DJNZ WAIT
                 ;
027C 3EFF         LD A,0FFH
027E D301         OUT (P1C),A
0280 3EFE         LD A,0FEH
0282 D301         OUT (P1C),A
0284 D300         OUT (P1),A         ;SEND BIT0 LOW
0286 0600         LD B,0
0288 DB00        BITS16 IN A,(P1)    ;WAIT FOR BITS 1-6
028A E67E         AND 7EH            ;     HIGH
028C FE7E         CP 7EH
028E 2805         JR Z,VEREP         ;V EXEC HAS REPLIED
0290 10F6         DJNZ BITS16
0292 C34702       JP MNSTCK          ;RESET IF NO REPLY
                 ;
0295 3EFF        VEREP LD A,0FFH
0297 D301         OUT (P1C),A
0299 3EF6         LD A,0F6H
029B D301         OUT (P1C),A
029D D300         OUT (P1),A         ;SEND BIT 1 LOW
029F DB00         IN A,(P1)          ;CHECK FOR HIGHER
02A1 E606         AND 6              ;PRIORITY SUB-SYSTEMS
02A3 FE06         CP 6
```

```
02A5 2023          JR NZ,AGAIN
02A7 3EFF          LD A,OFFH
02A9 D303          OUT (P2C),A
02AB 3E00          LD A,0
02AD D303          OUT (P2C),A
02AF 3A0418        LD A,(CCW)          ;SEND CCW
02B2 D302          OUT (P2),A
                  ;
02B4 3EFF          LD A,OFFH           ;SET BITS 1-6 AS INPUT
02B6 D301          OUT (P1C),A
02B8 3EFE          LD A,OFEH
02BA D301          OUT (P1C),A
02BC 0600          LD B,0
02BE DB00         FINLP IN A,(P1)      ;WAIT FOR BITS 1-6 LOW
02C0 E67E          AND 7EH
02C2 CAA800        JP Z,CINIT
02C5 10F7          DJNZ FINLP
02C7 C34702        JP MNSTCK           ;RESET IF V EXEC DIES
                  ;
02CA CDA800       AGAIN CALL CINIT      ;RE-INITIALIZE PIOs
02CD 3E88          LD A,88H             ;AND TRY AGAIN
02CF 189C          JR COMOUT
                  ;
                  ;*** END OF COMMUNICATIONS HANDLER ***
                  ;
0400               ORG SROM+400H
0400 B0B1B2B3      DEFB OB0H,OB1H,OB2H,OB3H,OB4H,OB5H
0406 B6B7B8B9      DEFB OB6H,OB7H,OB8H,OB9H,OBAH,OBBH
040C BCBDBEBF      DEFB OBCH,OBDH,OBEH,OBFH,OC0H,OC1H
0412 C2C3C4C5      DEFB OC2H,OC3H,OC4H,OC5H,OC6H,OC7H
0418 C8C9CACB      DEFB OC8H,OC9H,OCAH,OCBH,OCCH,OCDH
041E CECFD0D1      DEFB OCEH,OCFH,OD0H,OD1H,OD2H,OD3H
0424 D4D5D6D7      DEFB OD4H,OD5H,OD6H,OD7H,OD8H,OD9H
042A DADBDCDD      DEFB ODAH,ODBH,ODCH,ODDH,ODEH,ODFH
0430 E0E1E2E3      DEFB OE0H,OE1H,OE2H,OE3H,OE4H,OE5H
0436 E6E7E8E9      DEFB OE6H,OE7H,OE8H,OE9H,OEAH,OEBH
043C ECEDEEEF      DEFB OECH,OEDH,OEEH,OEFH,OF0H,OF1H
0442 F2F3F4F5      DEFB OF2H,OF3H,OF4H,OF5H,OF6H,OF7H
0448 F8F9FAFB      DEFB OF8H,OF9H,OFAH,OFBH,OFCH,OFDH
044E FEFF          DEFB OFEH,OFFH
0450 00010203     SPDTAB DEFB 0,1,2,3,4,5,6,7,8,9,10
045B 0B0C0D0E      DEFB 11,12,13,14,15,16,17,18,19,20
0465 15161718      DEFB 21,22,23,24,25,26,27,28,29,30
046F 1F202122      DEFB 31,32,33,34,35,36,37,38,39,40
0479 292A2B2C      DEFB 41,42,43,44,45,46,47,48,49,50
0483 33343536      DEFB 51,52,53,54,55,56,57,58,59,60
048D 3D3E3F40      DEFB 61,62,63,64,65,66,67,68,69,70
0497 4748494A      DEFB 71,72,73,74,75,76,77,78,79,80
04A1 F0F1F2F3      DEFB OF0H,OF1H,OF2H,OF3H,OF4H,OF5H
04A7 F6F7F8F9      DEFB OF6H,OF7H,OF8H,OF9H,OFAH,OFBH
04AD FCFDFEFF      DEFB OFCH,OFDH,OFEH,OFFH
04B1 00010203     CPTAB DEFB 0,1,2,3,4,5,6,7,8,9,10
04BC 0B0C0D0E      DEFB 11,12,13,14,15,16
04C2 01C8         RUNOUT DEFB 1,200
04C4 02C8                DEFB 2,200
04C6 04C8                DEFB 4,200
04C8 0864                DEFB 8,100
04CA 1064                DEFB 16,100
04CC 3264                DEFB 50,100
04CE 6264                DEFB 98,100
```

```
PolyZap V2.0     TRACTION 3  28-1-83

                     ;
1800                  ORG RAMTOP
1800 + 0002          ISRAD DEFS 2
1802 + 0001          ADATA DEFS 1
1803 + 0001          SPEED DEFS 1
1804 + 0001          CCW DEFS 1
1805 + 0010          INFO DEFS 16
1815 + 0001          REF DEFS 1
1816 + 0001          ERROR DEFS 1
1817 + 0001          DCYCLE DEFS 1
1818 + 0001          DCDEM DEFS 1
1819                  END
```

REFERENCES.

1. "The welcome intruder", H and SM, June 1982, pp 67 - 71.
2. "Computer-controlled towline serves machines automatically", Modern Materials Handling, December 1979, pp 52 - 56.
3. "Automatically operated industrial vehicles", P.E.D., February 1981, pp 49 - 54.
4. "The future's most advanced warehouse is here now", P.E.D., May 1981, pp 92 - 95.
5. M.Briot, J.C.Talou, G.Bauzil, "The multi-sensors which help a mobile robot find its place", Sensor Review, Vol No.1, January 1981, pp 15 - 19.
6. M.Ferrer, M.Briot, J.C.Talou, "Study of a video image treatment system for the mobile robot HILARE", Proc. 1st. Int. Conf. ROVISEC Stratford, April 1981, pp 59 - 71.
7. G.Bauzil, M.Briot, P.Ribes, "A navigation sub-system using ultrasonic sensors for the mobile robot HILARE", Proc. 1st. Int. Conf. ROVISEC Stratford, April 1981, pp 47 - 58.
8. L.Marce, M.Julliere, H.Place, H.Perrichot, "A semi-autonomous remote controlled mobile robot", Ind. Robot, Vol 7 No.4, December 1980, pp 232 - 235.
9. L.Marce, M.Julliere, H.Place, "An autonomous computer controlled vehicle", Proc. 1st. Int. Conf. on AGVS Stratford, June 1981, pp 113 - 122.
10. S.Mouchaud, H.Merguen, B.Lemaire, "A self adapting low cost sonair for use on mobile robots", Sensor Review, Vol 1 No.4, October 1981, pp 180 - 183.
11. "The itinerant robots", Technology, Vol 6 No.35, 27th, September 1982, pp 13 - 14.

12. IEE Colloquium on Control of Manipulators and Robotic Devices, London, 1980.

13. D.Nitzan (S.R.I.), "Assessment of robotic sensors", Proc. 1st. Int. Conf. ROVISEC Stratford. April 1981, pp 1 - 11.

14. U.K. Patent Specification 935,751 EMI Ltd., "Improvements relating to conveyor systems", 4th, September 1963.

15. U.K. Patent Specification 980,313 EMI Ltd., "Improvements relating to vehicle guidance systems", 13th, January 1965.

16. M.H.E.Larcombe, "Navigation and protection of computer controlled vehicles", IEE Coll. 'Control of manipulators and robotic devices', London, April 1980, Paper 2.

17. U.K. Patent Specification 1,086,724 EMI Ltd., "Improvements in or relating to apparatus for controlling the movement of a moveable member", 11th, October 1967.

18. W.K.Pratt, "Digital Image Processing", (Book) Wiley 1978.

19. G.J.Vanderbrug, J.S.Albus, E.Barkmeyer, "A vision system for real-time robot control", Robotics Today, Winter 1979/80, pp 20 - 21.

20. K.Stout, A.F.Thomas, "How robots are improving their eyesight", The Production Engineer, March 1980, pp 29 - 34.

21. A.Pugh, "Vision applied to robot manipulators and part inspection", IEE Coll. 'Control of manipulators and robotic devices', London, April 1980, Paper 5.

22. J.S.E.Bromley, "First Level processing of signals from a solid-state camera", IEE Coll. 'Robot sensor signal processing', London, November 1982, Paper 3.

23. P.E.Anuta, "Rail car identification using pattern recognition techniques", Joint ASME/IEEE/AAR Railroad Conf., 1979.

24. C.Braccini, G.Gambardella, G.Sandini, T.Tagliasco, "Borrowing from the eyes to create robot vision algorithms", Sensor Review, Vol 1 No.2, April 1981, pp 68 - 72.

25. A.H.Bond, "Fast vision for a low cost computer controlled robot", 2nd. Int. C.I.S.M. - IFTOMM SYMP. on 'Theory and practice of robots and manipulators", Warsaw, 1976, pp 273 - 284.

26. C.A.Rosen, N.J.Nilsson (S.R.I.), "An intelligent Automaton", IEEE Int. Conv. Rec., 1967, Part 9, pp 50 - 55.

27. M.H.E.Larcombe, "Mobile robots for industrial use", Ind. Robot, Vol 6, 1979, pp 70 - 76.

28. W.K.Taylor, G.Cro, "Real time teaching and recognition system for robot vision", Ind. Robot, Vol 7 No.2, June 1980, pp 99 - 106.

29. "Getting involved with vision", Sensor Review, Vol 1 No.2, April 1981, p 67.

30. J.Hartley, "Academia rules in Tokyo", Ind. Robot, Vol 8 No.4, December 1981, pp 255 - 259.

31. J.Hartley, "Picking parts from a bowl feeder with image sensing", Sensor Review, Vol 1 No.1, January 1981, pp 30 - 31.

32. J.H.Crenshaw, "Federated versus integrated computer systems", AGARD Conf. Proc. 'The application of digital computers to guidance and control', 1976, p 44.

33. B.Rooks, "Solving the basics can lead to better robot design", Ind. Robot, Vol 8 No.4, December 1981, pp 242 - 244.

34. "Inductive wire-guided store truck in the manufacturing industry", Industrial and Production Engineering, 1981/2, p 51.

35. "Komatsu computer controlled forklift truck", Komatsu Company Report, 1979.

36. International Materials Handling Exhibition, NEC Birmingham, 1980.

37. A.Gaillet, V.Fuertes, M.Llibre,"Optical automatic guidance system of a mobile robot for industrial manutention", Proc. 1st. Int.Conf.

on AGVS Stratford, June 1981, pp 79 - 88.

38. Nakamura, I.Fukui, "Navigation of a robot vehicle by slit pattern detection", IFAC MANUFACONT '80,Budapest.

39. L.Norton-Wayne, D.Guentri, "Vehicle guidance by automated scene analysis", Proc. 1st. Int. Conf. on AGVS Stratford, June 1981, pp 129 - 136.

40. RS Data sheet R/2135 March 1980.

41. J.Brignell, "Micro-maths -2", Electron, 11th, March 1980, pp 12 - 14.

42. Kodak high speed Infra-Red Film 2481 Black and White negative.

43. R.C.Dorf, "Modern control systems", (Book) Addison Wesley 3rd. Ed. 1980, Section 3.3, pp 87 - 90.

44. Jungheinrich: Carnarvon St., Manchester, M3 1EZ.

45. Malthouse Hunter: 5/7, Norwich Rd., Bournemouth, Dorset. BH2 5QZ.

46. Eaton Kenway: Kenway Inc., 310, South Main St., Salt Lake City, UTAH, 84101.

47. Wagner Indumat: Central Way, North Feltham Trading Estate, Feltham, Middx., TW14 OUJ.

48. BT Rolatruc: Stirling Rd., Trading Estate, Slough, Berks., SL1 4SY.

49. Carrago Transport System AB: 10044, Pianezza (To), Italy.

50. Babcock: PHB, 58/59, Gt. Marlborough St., London, W1V 1DD.

51. Komatsu Forklift Co. Ltd: No.3-4, Akasaka 2- Chome, Minato - ku, Tokyo.