



# Durham E-Theses

---

## *The low-level guidance of an experimental autonomous vehicle*

Pears, Nicholas Edwin

### How to cite:

---

Pears, Nicholas Edwin (1989) *The low-level guidance of an experimental autonomous vehicle*, Durham theses, Durham University. Available at Durham E-Theses Online: <http://etheses.dur.ac.uk/6731/>

### Use policy

---

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full Durham E-Theses policy](#) for further details.

pears

The Low-level Guidance  
of an  
Experimental Autonomous Vehicle

by Nicholas Edwin Pears B.Sc.

The copyright of this thesis rests with the author.  
No quotation from it should be published without  
his prior written consent and information derived  
from it should be acknowledged.

A Thesis Submitted for the Degree of Doctor of Philosophy

School of Engineering and Applied Science  
University of Durham

1989



2 AUG 1990

## **Declaration**

None of the work contained in this thesis has been previously submitted for a degree in this or any other university. The work contained in this thesis is not part of a joint research project.

## **Copyright**

The copyright of this thesis rests with the author. No quotation from it should be published without his prior written consent and information derived from it should be acknowledged.

## ACKNOWLEDGEMENTS

Firstly, I would like to acknowledge the invaluable assistance and the tremendous support and encouragement I have had from my supervisor, Dr. Jim Bumby. I am also grateful to I.C.I. plc. for funding the research; in particular I would like to thank my industrial supervisor, Mike Gladders, for many helpful comments.

Gratitude is also extended to Colin Dart for re-building the sensor interface boards which, unlike my own, did not resemble either a crow's nest or a Christmas tree. Thanks also to Jack Greensmith for handling numerous orders for equipment, and for a number of interesting reminiscences. I am also grateful to the Durham University Microprocessor Centre; in particular to Jim Swift, Peter Baxendale, and Milos Kolar for their help with the on-board computing hardware and the provision of software support.

Finally, I would like to thank my parents for supporting me financially whilst this thesis was being written.

## ABSTRACT

This thesis describes the data processing and the control that constitutes a method of guidance for an autonomous guided vehicle (AGV) operating in a predefined and structured environment such as a warehouse or factory.

A simple battery driven vehicle has been constructed which houses an MC68000 based microcomputer and a number of electronic interface cards. In order to provide a user interface, and in order to integrate the various aspects of the proposed guidance method, a modular software package has been developed. This, along with the research vehicle, has been used to support an experimental approach to the research.

The vehicle's guidance method requires a series of concatenated curved and straight imaginary lines to be passed to the vehicle as a representation of a planned path within its environment. Global position specifications for each line and the associated AGV direction and demand speed for each line constitute commands which are queued and executed in sequence. In order to execute commands, the AGV is equipped with low level sensors (ultrasonic transducers and optical shaft encoders) which allow it to estimate and correct its global position continually. In addition to a queue of commands, the AGV also has a pre-programmed knowledge of the position of a number of correction boards within its environment. These are simply wooden boards approximately 25cm high and between 2 and 5 metres long with small protrusions ("notches") 4cm deep and 10cm long at regular (1m) intervals along its length. When the AGV passes such a correction board, it can measure its perpendicular distance and orientation relative to that board using two sets of its ultrasonic sensors, one set at the rear of the vehicle near to the drive wheels and one set at the front of the vehicle. Data collected as the vehicle moves parallel to a correction board is digitally filtered and subsequently a least squares line fitting procedure is adopted. As well as improving the reliability and accuracy of orientation and distance measurements relative to the board, this provides the basis for an algorithm with which to detect and measure the position of the protrusions on the correction board. Since measurements in three planar, local coordinates can be made (these are:  $x$ , the distance travelled parallel to a correction board;  $y$ , the perpendicular distance relative to a correction board; and  $\theta$ , the clockwise planar orientation relative to the correction board), global position estimation can be corrected.

When position corrections are made, it can be seen that they appear as step disturbances to the control system. This control system has been designed to allow the vehicle to move back onto its imaginary line after a position correction in a critically damped fashion and, in the steady state, to track both linear and curved command segments with minimum error.

# CONTENTS

	Page
<b>Acknowledgements</b> .....	i
<b>Abstract</b> .....	ii
<b>Contents</b> .....	iii
<b>List of Figures</b> .....	x
<b>List of Tables</b> .....	xvi
<b>List of Plates</b> .....	xvii
<b>List of Symbols</b> .....	xviii
<b>Chapter 1: Introduction</b> .....	1
1.1 Introduction .....	1
1.2 Industrial automation using AGV based systems .....	2
1.3 Vehicle autonomy .....	5
1.4 The formulation of a research plan and the generation of the aims of the research .....	6
1.5 Format of the presented work .....	10

<b>Chapter 2: A Review of Autonomous Vehicle Research</b>	.....	13
2.1	Introduction	13
2.2	A multisensory approach	14
2.3	Sensor models and sensor data fusion	16
2.4	World modelling	19
2.5	The structuring of autonomous vehicle systems	21
2.6	Discussion: The role of low level guidance	25
2.7	Summary	28

<b>Chapter 3: The Research Vehicle, Computational Resources, and the Sensor Interface Hardware</b>	.....	30
3.1	Introduction	30
3.2	The research vehicle	30
3.3	Computational resources	32
3.3.1	Overview	32
3.3.2	Program development	33
3.3.3	The MC68000 based on-board microcomputer	34
3.4	Development of the interface electronics	35
3.4.1	Development of an ultrasonic ranging unit	35
3.4.2	The optical shaft encoder interface	37
3.4.3	The power supply board	38
3.4.4	The switching and opto-isolation board	39
3.5	Summary	40

<b>Chapter 4: Sensor Functionality in the Guidance Method and Sensor Characterisation</b>	.....	41
4.1	Introduction	41

4.2	Sensor functionality in the guidance method .....	41
4.3	Position correction .....	43
4.4	Errors in odometric position measurement .....	45
4.4.1	The nature of odometric measurement errors .....	45
4.4.2	The method of maintaining odometric error bands ...	46
4.4.3	The validity of maintaining odometric error bands ..	49
4.5	Ultrasonic sensing for autonomous vehicles .....	51
4.5.1	General .....	51
4.5.2	The idiosyncrasies of ultrasonic ranging .....	53
4.6	Performance of the designed ultrasonic ranging system .....	56
4.6.1	Range of the measurement system .....	56
4.6.2	Accuracy of ultrasonic ranging .....	59
4.6.3	A characterisation of the ultrasonic ranging system ..	61
4.6.4	Offset errors in ultrasonic ranging .....	62
4.6.5	Random errors in ultrasonic ranging .....	63
4.6.6	A new method of ultrasonic ranging .....	65
4.7	Discussion .....	66
4.8	Summary and conclusion .....	69

<b>Chapter 5: AGV Command Structure, Steering Control Strategy, and Transformations from Sensing to Control .....</b>	<b>71</b>
5.1 Introduction .....	71
5.2 AGV command structure .....	71
5.3 The basis of the AGV control .....	73
5.4 Formulation of a control strategy .....	74
5.5 Transformations from internal sensing to control .....	75
5.6 Calculations required for odometry .....	77
5.7 Transformation into a path dependent local frame .....	80

5.7.1	Line following transformations	80
5.7.2	Curve following transformations	82
5.8	Independent wheel speed control: Base speed control and steering	84
5.9	Summary	87

**Chapter 6: A Modular Software Package for User Interface and AGV Control** ..... 88

6.1	Introduction	88
6.2	An overview of the software structure	89
6.3	The set up procedure	94
6.4	The menu	95
6.5	Loading and saving arrays	96
6.6	Editing and appending arrays	96
6.7	Building commands	97
6.8	Path plan option	98
6.9	Board searching	100
6.10	Command queue execution	102
6.10.1	The command preparation routine	103
6.10.2	The distance monitoring routine	105
6.11	The odometry routine	106
6.12	The data processing routine	107
6.13	Base velocity control	112
6.14	The software controller	114
6.15	Limiting the rate of steering	115
6.16	The wheel speed output routine	116
6.17	High level interrupts	117
6.18	The method of results analysis	119

6.19	Software documentation .....	120
6.20	Summary .....	121

## **Chapter 7: Design, Analysis, and Performance of the AGV Steering**

<b>Control</b> .....	122
7.1 Introduction .....	122
7.2 An analysis of the basic steering control system .....	123
7.3 Performance of the control system in line following .....	127
7.3.1 Response to a step disturbance .....	127
7.3.2 Steady state response .....	130
7.4 Performance of the control system in tracking a curve .....	131
7.5 Velocity dependent control parameters .....	136
7.6 Curvature rate limiting .....	139
7.7 Analysis of the control in the discrete domain .....	145
7.7.1 Aims and method of the discrete analysis .....	145
7.7.2 Z-transform system without curvature rate limiting ..	146
7.7.3 Z-transform of the curvature rate limited system ...	147
7.7.4 Simulation results .....	148
7.7.5 Conclusions from the discrete analysis .....	150
7.8 Summary .....	151

## **Chapter 8: Data Processing and the Performance of the AGV**

.....	154
8.1 Introduction .....	154
8.2 Digital filtering .....	155
8.3 The elimination of spurious ultrasonic range measurements ..	158
8.4 The notch as seen by an ultrasonic sensor .....	159

8.5	The use of regression in data processing .....	161
8.6	Notch detection .....	164
8.7	Measurement of the local $x$ coordinate .....	168
8.8	Compensation for offset errors .....	170
8.9	Estimation of the error in ultrasonic position measurement .	172
8.10	Performance of the research vehicle .....	176
8.10.1	A comparison of corrected and uncorrected tracking performance .....	176
8.10.2	Performance of the vehicle with a large initial misalignment .....	177
8.10.3	Vehicle performance over more complex paths .....	178
8.11	A discussion of vehicle performance .....	179
8.12	Summary and conclusion .....	181
<b>Chapter 9: Summary, General Conclusions, and Suggestions for Further Work</b> .....		182
9.1	Summary and general conclusions .....	182
9.2	The contribution of the presented work .....	187
9.3	Suggestions for further work .....	188
<b>References</b> .....		192

<b>Appendix 1: A Review of Experimental AGV and Mobile Robot Research Projects .....</b>	<b>200</b>
<b>Appendix 2: Calculations of Shaft Encoder Interface Circuit Parameters .....</b>	<b>207</b>
<b>Appendix 3: Equations Required for Basic Path Planning ..</b>	<b>209</b>
<b>Appendix 4: A Geometrical Analysis of the AGV Control System .....</b>	<b>212</b>

## LIST OF FIGURES

### Figure

- 3.1 The research AGV
  - 3.1a Plan view of the research AGV
  - 3.1b Side view of the research AGV
- 3.2 The AGV research rig
- 3.3 The ultrasonic sensor interface
  - 3.3a A schematic diagram of the ultrasonic sensor interface
  - 3.3b Waveforms in the ultrasonic sensor interface
- 3.4 The ultrasonic sensor interface (detail)
- 3.5 The shaft encoder interface
- 3.6 The power supply board
- 3.7 System interconnection, opto-isolation, and switching
  
- 4.1 The position correction method
- 4.2 The expansion of odometric error bands
- 4.3 Idiosyncrasies of ultrasonic ranging
  - 4.3a Directional ambiguity in ranging
  - 4.3b Effective beamwidth narrows with range
- 4.4 A characterisation of systematic measurement errors
- 4.5 The delay in small amplitude pulse detection
- 4.6 Fluctuations in the amplitude of the return echo
- 4.7 The detailed form of an ultrasonic echo pulse
- 4.8 The relative frequency of ultrasonic range measurements
- 4.9 Detection of the peak of the return echo

- 5.1 The basis of the AGV control
- 5.2 The control method
- 5.3 Odometric geometry
- 5.4 Transformation to the local frame of a line segment
- 5.5 An alternative transformation to the local frame
- 5.6 Transformation to a local frame in a curve segment
- 5.7 The relationship between wheel speeds and turning radius
  
- 6.1 A software overview
- 6.2 The command queue array
- 6.3 The main menu
- 6.4 Calculation of command parameters for a corner
- 6.5 Command preparation
- 6.6a Data processing routine
- 6.6b Data processing routine-continued
- 6.7 The regression routine
- 6.8 The line refit routine
- 6.9 The position correction routine
- 6.10 Base velocity control
- 6.11 The wheel speed output routine
- 6.12 The high level interrupt timing routine
  
- 7.1 Block diagram of the control system
- 7.2 The line following process
- 7.3 Critically damped response:  $k_p = 4$ ,  $k_{pd} = 1$ ,  $V = 0.2m/s$ 
  - 7.3a AGV path in a step response test
  - 7.3b Graph of distance error against time
  - 7.3c Graph of local angle ( $\theta'$ ) against time

- 7.3d Graph of curvature ( $\kappa$ ) against time
- 7.4 Underdamped response:  $k_p = 1$ ,  $k_{pd} = 3$ ,  $V = 0.2m/s$ 
  - 7.4a AGV path in a step response test
  - 7.4b Graph of distance error against time
  - 7.4c Graph of local angle ( $\theta'$ ) against time
  - 7.4d Graph of curvature ( $\kappa$ ) against time
- 7.5 Steady state response
  - 7.5a Graph of distance error against time
  - 7.5b Graph of local angle ( $\theta'$ ) against time
- 7.6 Curve tracking test:  $k_p = 4$ ,  $k_{pd} = 1$ ,  $V = 0.1m/s$ 
  - 7.6a AGV path in a curve tracking test
  - 7.6b Graph of distance error against time
  - 7.6c Graph of local angle ( $\theta'$ ) against time
  - 7.6d Graph of curvature ( $\kappa$ ) against time
- 7.6 Curve tracking test:  $k_p = 8$ ,  $k_{pd} = 2$ ,  $V = 0.1m/s$ 
  - 7.7a AGV path in a curve tracking test
  - 7.7b Graph of distance error against time
- 7.8 The modified control system
- 7.9 Curve tracking performance with  $\kappa_{seg}$  input
  - 7.9a AGV path in a curve tracking test
  - 7.9b Graph of distance error against time
- 7.10 Step response tests at reduced velocity
  - 7.10a AGV path in a step response test
  - 7.10b Graph of distance error against time
  - 7.10c Graph of local angle ( $\theta'$ ) against time
  - 7.10d Graph of curvature ( $\kappa$ ) against time
- 7.11 Methods of curvature rate limiting
  - 7.11a No curvature rate limiting

- 7.11b Non-linear curvature rate limiting
- 7.11c Linear curvature rate limiting
- 7.12 Non-linear curvature rate limiting
  - 7.12a Graph of distance error against time
  - 7.12b Graph of curvature ( $\kappa$ ) against time
- 7.13 Curvature rate limiting in the  $s$ -plane
  - 7.13a  $s$ -plane representation of the limiting block
  - 7.13b Reduced form of the limiting block
- 7.14 Root locus in  $s$ -domain
  - 7.14a No curvature rate limiting
  - 7.14b Curvature rate limited system
- 7.15 Curvature rate limited system (step response test)
  - 7.15a Graph of distance error against time
  - 7.15b Graph of curvature ( $\kappa$ ) against time
- 7.16 Step response test for curvature rate limited system
  - 7.16a A comparison of unlimited and curvature rate limited responses
  - 7.16b Graph of curvature ( $\kappa$ ) against time
- 7.17 Curvature rate limiting in the  $z$ -plane
  - 7.17a  $z$ -plane representation of the limiting block
  - 7.17b Reduced form of the limiting block
- 7.18 A comparison of responses with varying sampling interval
  - 7.18a No curvature rate limiting
  - 7.18b Curvature rate limited system
- 7.19 Bode plots for control system
  - 7.19a No curvature rate limiting
  - 7.19b Curvature rate limited system

- 8.1 Ultrasonic range data as a sensor passes a notch
  - 8.1a Unfiltered ultrasonic data
  - 8.1b Filtered ultrasonic data
- 8.2 Ultrasonic data with four spurious values
- 8.3 Filter constant adjustment
  - 8.3a High suspect measurement
  - 8.3b Low suspect measurement
- 8.4 The notch as seen by an ultrasonic sensor
  - 8.4a Range data as sensor  $k$  moves past a notch
  - 8.4b The splitting of an ultrasonic echo pulse
- 8.5 The degradation of notch shape with distance
  - 8.5a Close range observation of a notch
  - 8.5b Far range observation of a notch
- 8.6 The notch detection algorithm
- 8.7 Regression lines derived off-line from ultrasonic data
  - 8.7a Regression lines with 30 data points
  - 8.7b Regression lines with 20 data points
- 8.8 Regression lines with line refitting
  - 8.8a Regression lines with 10 data points and line refit
  - 8.8b Regression lines derived in a real time run
- 8.9 Results from the notch detection algorithm
  - 8.9a Regression line gradients as the AGV moves past a correction board
  - 8.9b Operation of the notch detection algorithm
- 8.10 Performance of the vehicle with low frequency corrections
  - 8.10a Graph of distance error against time
  - 8.10b Graph of local angle ( $\theta'$ ) against time
- 8.11 Overall performance of the vehicle

8.11a AGV path

8.11b Graph of distance error against time

## LIST OF TABLES

### Table

- |     |   |
|-----|---|
| 4.1 | Mean of filtered ultrasonic distance measurements               |
| 4.2 | Standard deviation of filtered ultrasonic distance measurements |
| 7.1 | <i>CC</i> predictions for the curvature rate limited system     |
| 7.2 | A comparison of <i>s</i> -plane and <i>w</i> -plane analyses    |
| 8.1 | Errors in uncorrected position after travelling 5m              |
| 8.2 | Errors measured in odometric position for re-run tests          |
| 8.3 | Errors measured in position, with position corrections          |

## LIST OF PLATES

### Plate

- 3.1 Front view of the research vehicle
- 3.2 Rear view of the research vehicle
- 4.1 The research vehicle close to a correction board
- 4.2 The front sensor and the correction board

## LIST OF SYMBOLS

Symbol		Units
$a$	Width of an aisle	(m)
$b_0$	Constant of regression line	(m)
$b_1$	Gradient of regression line	
$c$	Clearance between AGV and a wall at critical positions on a designed path	(m)
$C$	Capacitance	(F)
$\bar{d}$	Average distance travelled by both drive wheels	(m)
$d_n$	Direction of the vehicle (forward=1, reverse=-1).	
$d_{wheel}$	Diameter of a drive wheel	(m)
$f_c$	Clock frequency	(Hz)
$f_{co}$	Cut off frequency of filter	(Hz)
$G$	Open loop transfer function of the basic control system	
$G_{lb}$	Open loop transfer function of the limiting block	
$G_{lim}$	Open loop transfer function of the curvature rate limited system	
$i$	Index of $i$ th control interval or $i$ th command segment	
$k$	Index denoting sensor $k$	
$k_{dr}$	Odometric constant	
$k_l$	Curvature rate limiting constant	
$k_p$	Local angle ( $\theta'$ ) proportional control constant	
$k_{pd}$	Local distance ( $y$ ) proportional control constant	
$l$	Length of the AGV	(m)
$L$	Distance remaining in a line or curve command segment	(m)
$M_p$	Ratio of peak overshoot to a step change	
$n$	Latched counter value	

$n_1$	Number of counts provided by the left shaft encoder over $T_s$	
$n_2$	Number of counts provided by the right shaft encoder over $T_s$	
$p_f$	Position of the front centre of the AGV	
$p_{fl}$	Position of the front left of the AGV	
$r$	Radius described by the AGV's reference point	(m)
$r_d$	Dynamic deceleration rate	(m/s <sup>2</sup> )
$r_{seg}$	Radius of a curved command segment	(m)
$r_0$	Standard acceleration or deceleration of the vehicle	(m/s <sup>2</sup> )
$r_1$	Radius described by the left drive wheel	(m)
$r_2$	Radius described by the right drive wheel	(m)
$R$	Resistance	( $\Omega$ )
$R_f$	Feedback resistance	( $\Omega$ )
$s_f$	Distance of front sensor from AGV reference point	(m)
$s_r$	Distance of rear sensor from AGV reference point	(m)
$ss$	Subscript denoting steady state	
$s_x$	Software variable dependent on vehicle heading	
$s_y$	Software variable dependent on vehicle heading	
$t$	Time	(s)
$t_p$	Time to reach first overshoot	(s)
$t_r$	Time required to attain 95% of a step disturbance	(s)
$T$	Time constant of an exponential rise	(s)
$T_a$	Ambient temperature	( $^{\circ}C$ )
$T_s$	Sampling interval	(mS)
$V$	Base velocity: Velocity of the AGV's reference point	(m/s)
$V_{cc}$	Supply voltage	(V)
$V_d$	Demand base velocity	(m/s)
$V_h$	Hysteresis voltage band	(V)
$V_{rel}$	Difference in velocity between left and right drive wheels	(m/s)

$V_s$	Velocity of sound in air	(m/s)
$V_x$	Base velocity component in the local $x$ direction	(m/s)
$V_1$	Velocity of the left drive wheel	(m/s)
$V_2$	Velocity of the right drive wheel	(m/s)
$w$	Width of the AGV	(m)
$W_b$	Wheel base of the AGV	(m)
$x, y, \theta'$	Coordinates in a path dependent local reference frame	
$x_i$	Local coordinates over which regression lines are fitted	(m)
$x_r$	Deceleration distance required to reach new demand speed	(m)
$x_0$	Local starting coordinate for position corrections	(m)
$x_1$	Local terminating coordinate for position corrections	(m)
$X, Y, \theta$	Coordinates in a global reference frame	
$X_c, Y_c$	Global coordinates of the centre of a curve segment	
$X_{fl}$	X coordinate of point $p_{fl}$	(m)
$X_{fl_{max}}$	Maximum of the coordinate $X_{fl}$	(m)
$X_{line}, Y_{line}$	Global coordinates on an imaginary line	
$X_s, Y_s$	Starting coordinates of a command segment	
$X_0, Y_0, \theta_0$	Position of local coordinate frame in a global reference frame	
$y$	Perpendicular distance to path in a local frame	(m)
$y_{k_i}$	$i$ th filtered measurement from sensor $k$	(m)
$\hat{y}_{k_i}$	$i$ th value on the current regression line of sensor $k$	(m)
$\hat{y}_{k_n}$	Final value on the current regression line of sensor $k$	(m)
$y_{lw}$	Separation of demand path and correction board	(m)
$y_{ss}$	Steady state value of perpendicular distance error	(m)
$y_1$	Measurement from rear ultrasonic sensor (local frame)	(m)
$y_2$	Measurement from front ultrasonic sensor (local frame)	(m)
$y_{1_{off}}$	Offset error in reading from rear sensor	(m)
$y_{2_{off}}$	Offset error in reading from front sensor	(m)

$\alpha$	Digital filtering constant	
$\alpha, \beta$	Constants to expand odometric error bands	
$\delta y$	Estimated error in the local y coordinate measurement	(m)
$\delta \hat{y}_{k_n}$	Estimated error in processed position measurement provided by sensor $k$	(m)
$\delta \theta'$	Estimated error in local heading measurement	(rads)
$\Delta d_1$	Incremental distance travelled by the left drive wheel	(m)
$\Delta d_2$	Incremental distance travelled by the right drive wheel	(m)
$\Delta n$	Difference in count provided by left and right shaft encoders	
$\Delta t$	A small interval of time	(s)
$\Delta X$	Change in AGV X coordinate over an odometric interval	(m)
$\Delta Y$	Change in AGV Y coordinate over an odometric interval	(m)
$\gamma_{k_i}$	$i$ th unfiltered range measurement from sensor $k$	(m)
$\kappa$	Actual vehicle turning curvature	(m <sup>-1</sup> )
$\kappa_{dem}$	Demand (unlimited) AGV turning curvature	(m <sup>-1</sup> )
$\kappa_{lp}$	Demand curvature derived from local position	(m <sup>-1</sup> )
$\kappa_{seg}$	Curvature of current command segment	(m <sup>-1</sup> )
$\kappa_{ss}$	Steady state value of turning curvature	(m <sup>-1</sup> )
$\sigma$	Standard deviation of raw ultrasonic measurements	
$\sigma_f$	Standard deviation of filtered ultrasonic measurements	
$\theta$	Heading in a global reference frame	(rads)
$\theta'$	Heading in a path dependent local reference frame	(rads)
$\theta_{quad}$	Orientation of the local frame in the first quadrant	(rads)
$\theta'_r$	Demand heading relative to demand path	(rads)
$\theta'_{ss}$	Steady state value of local angle	(rads)
$\theta_s$	Demand heading at the start of a curve segment	(rads)
$\theta_0$	Global orientation of the local frame	(rads)
$\xi$	Damping factor	

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

The term "autonomous vehicle" is applied to a vehicle that can guide itself from its current location to some destination within its environment without the aid of a driver or operator. Such vehicles are sometimes referred to as "automatic guided vehicles" (AGVs), whilst at other times they are labelled "mobile robots". The term AGV is usually used for vehicles which have a well defined task and that operate in a well defined and structured environment. Mobile robot research, however, is directed towards vehicles that exhibit intelligent behaviour in more complex environments that may be partially or even completely unknown.

Although in the above definition the term "autonomous vehicle" is seemingly self-explanatory, it will be seen that it is used to describe machines which have a wide range of function, structures, and complexity. In this chapter, the different forms which such vehicles can take will be explored and from this general inferences can be drawn concerning autonomous vehicle guidance. For example, what attributes are common to all types of autonomous vehicle and how do these attributes vary in accordance with the function of the vehicle?

Autonomous vehicles are employed, or are being developed, for a variety of applications such as planetary exploration, operation in the home and office, and handling in hazardous environments (reactor maintenance for example). Currently, however, their widest use is in the automation of warehouses and factories where such vehicles are termed 'AGVs'. Partly because of this fact, and partly because this project was created as a result of



the sponsors of this research (ICI) wanting to generate ideas about automating warehouses with AGVs, the work presented here is largely concerned with AGVs operating in an industrial environment.

The next section in this chapter elucidates the benefits to be gained in automating industrial materials handling using an AGV system. The operation of AGVs is then discussed in general terms; in particular the requirements for the autonomy of a vehicle are described. This provides a perspective on both current AGV and mobile robot research and the direction of the research described in this thesis. In further sections a formulation of a research plan is described, the main aims of this research are given, and the format of this thesis is delineated.

## 1.2 Industrial Automation Using AGV Based Systems

AGVs are becoming increasingly common in British and European industry for both warehouse automation and providing a flexible linkage between manufacturing processes in the factory. In 1985 over 500 AGV systems were in use in Europe involving some 5000 vehicles [*Engineering*, 1985].

In general, an automated materials handling installation can be thought to consist of a hierarchy of computing systems, a number of materials handling elements (conveyors, stacker cranes, AGVs), and some means of communication between the various parts of the system. The highest level, level 1, of the computing hierarchy is the corporate computing system which provides a warehouse supervisor (level 2 in the hierarchy) with its daily workload [JACKSON, 1986]. This supervisory computing level communicates with, and coordinates, various equipment such as conveyor controllers, stacker crane controllers, and an AGV scheduler. This scheduler is the third level in the computing hierarchy and has the function of local planning and coordination

of a number of AGVs. The final level of computation may reside with the warehouse equipment itself which, in the case of the AGV, is its on-board controller.

The extensive use of AGVs in industrial applications can be justified by considering, as an example, the benefits to be gained by automating a conventionally managed warehouse with an AGV based system. The first obvious advantage is a large reduction in running costs, since as well as not requiring drivers, there is the elimination of clerical work such as stock checking, taking orders, and issuing instructions. Much of this "clerical work" can take place over the communication links between the various levels of computing hierarchy. In addition, maintenance costs can be greatly reduced because the vehicles of an installed AGV system should not collide with either warehouse walls or stock.

Apart from reducing running costs, the function of automation is to maximise efficiency and this is manifested on three counts. Firstly, space utilisation is maximised: unlike conventional mechanisation in warehouses, such as stacker cranes, an AGV system does not tie up space. In addition to this, the communication links allow the immediate updating of stock records, which means there should never be any instances when spaces are recorded as occupied when they are really empty. Secondly, there is a reduction in errors. As well as eliminating human clerical errors, orders are not issued when stock has been removed from a particular location (again because of immediate updating of stock records). Finally, the AGV movements themselves can be optimised by algorithms residing in the central coordinating computer. The function of such a set of algorithms would be to minimise the time to complete a certain number of transactions but not necessarily to minimise individual transaction times. An optimisation of AGV movements would have to determine the best order in which to execute the required tasks

and how these tasks should be allocated to individual AGVs. Once a task had been allocated, an optimum route would be calculated for the AGV. This may not be the shortest in distance; a preferable criterion may be 'shortest time' taking into account overlapping of routes which may involve waiting at junctions. This amounts to a rather more complicated version of the "travelling salesman" problem, where there is more than one salesman and a number of additional constraints on the problem required to minimise transaction times and avoid collisions.

Although introducing AGVs into a warehouse may improve efficiency and reduce running costs, these advantages could be offset by a reduction in flexibility. A conventional warehousing system is both flexible and adaptable. There are no problems with installation or when the layout of the warehouse is changed. In addition vehicles can be added when required or replaced if repair is needed. A warehouse automated by AGVs must, as far as possible, retain the flexibility and adaptability of a conventional warehouse, otherwise it is not much of an improvement on static conveyor and stacker installations which tie up space.

Most AGV systems currently employed in industry rely on wires buried under the warehouse or factory floor for their guidance. In such installations, the difference in induced voltage in two coils on the AGV provides the control signal for steering as the vehicle moves along the wire. Although this system is simple, proven, and reliable, a great deal of flexibility is lost in automation. Such AGVs are not what is termed 'free ranging' since their choice of routing is dictated by the layout of the guide wires. This does not allow the AGV's movements within its working environment to be optimised. In addition, the layout of the warehouse cannot easily be changed and its constituent parts (bays, conveyors, etc.) must, in general, remain fixed and in proximity to the installed wires. If alterations are required, or a new system is to be

installed, difficulties can arise because often concrete warehouse floors are reinforced with steel. Such disruptive alterations can be costly in terms of lost production time. An additional problem with steel reinforced floors is the distortion of the magnetic field around the guidance wire which the AGV uses to generate its steering instructions. A major problem here is that if an AGV loses track of the wire it is following, it has no ability to find it again. In short the wire guidance arrangement is neither flexible nor adaptable and desirable benefits can be gained by employing free ranging vehicles.

### **1.3 Vehicle Autonomy**

Although autonomous vehicles differ greatly in their details of hardware and software, their behaviours and intelligence, and their structures and complexity, there are features common to all such vehicles that are essential for autonomy.

Firstly an autonomous vehicle must have means of sensing both its environment and its own actions. Sensory information gathered must be processed, integrated, and assimilated to yield positional information within the environment. If the vehicle is equipped with high level and diverse sensing capabilities it may also be able to detect certain situations or activities within its immediate vicinity. Secondly, in simple terms, the vehicle must have a set of algorithms within its controlling software that enable it to 'understand' its current perception of the working environment. Such algorithms may constitute forms of learning, expert systems, or other manifestations of artificial intelligence. Finally, this 'understanding' must be devolved into some combination of low level activities which ultimately actuate a behaviour, through movement and manipulation, appropriate to the overall goal of the vehicle.

The function of an autonomous vehicle and the environment it operates

in are inextricably linked to both the vehicle's sensory capabilities and the complexity of its reasoning and controlling algorithms. For example, lower level sensory information and simpler algorithms are required for an industrial AGV operating in a fixed, structured environment about which it has *a priori* knowledge than are required for a prototype planetary exploration vehicle whose environment would be both unstructured and unfamiliar.

Again the distinction is made between a vehicle termed an 'AGV' whose autonomy lies in its ability to guide itself to a known destination in a known environment and a vehicle which has more extensive sensory capabilities and more sophisticated algorithms which support a degree of behavioural competence and intelligence in unfamiliar environments. Such vehicles are commonly termed 'mobile robots'. This distinction is made since much of the literature on autonomous vehicles addresses such areas as environmental learning, artificial intelligence techniques, and the structuring of behaviours. Some of the projects containing work in these areas will be reviewed in the next chapter, but ultimately attention will be focused upon the lower level problems of sensing and guidance.

#### **1.4 The Formulation of a Research Plan and the Generation of the Aims of the Research**

In the next section of this chapter, the format of this thesis is set out. Before this is done, it is appropriate here to delineate the thoughts and circumstances that cast the initial direction of the research.

The project was created as a result of ICI wanting to generate ideas concerning the automation of warehouses using AGVs. Most current AGV systems use the wire guidance system and a central controlling computer which can communicate with the AGVs through the same or an adjacent

wire. Problems inherent in the wire guidance system have been expounded earlier in this chapter and the initial aim was to consider alternative sensing methods that could provide automation with so called 'free-ranging' AGVs.

The initial system proposal was to have an automated warehousing system consisting of a central computer communicating with a number of free-ranging AGVs. Since there would be no guide-wire, communication would be over a radio link. The communication link is required for the central computer to issue commands to individual AGVs and for the AGVs to provide the central computer with status reports and periodic updates of its position. The first part of the data sent from the central computer would contain the address of the vehicle(s) for which the command was intended and only the selected vehicles would respond to the remainder of the command message if that message were error free [BROPHY, 1985] [GIORDANO, 1981].

The next stage was to devolve warehouse automation into a number of basic tasks which were

- (1) Stock inventory.
- (2) Task generation.
- (3) Task designation.
- (4) Route calculation. (Navigation.)
- (5) AGV guidance to destination bay.
- (6) Control of the loading/unloading operation at a bay.

The idea behind this was to decide which functions the AGV software should support and which functions should be based centrally. In assigning tasks either to central control or an individual AGV the criteria were that

(1) Complexity and expense should be minimised by sharing common information through a central computer.

(2) The central computer can only communicate with an individual AGV every so often and so should not be burdened with 'low level' data transfers.

With these criteria it is clear that a representation of the current stock should be held within the central computer. With this representation and perhaps a 'clearing system' or some algorithm that takes into account the current location of individual AGVs and their destinations, the central processor can generate a list of tasks which need to be executed. The central computer would then designate tasks to individual AGVs. One basis for deciding which AGV should do which task may be from its current position and which task bay a particular free AGV could reach first. Such algorithms should plan well ahead of the current situation in the warehouse and have the ability to modify plans on the basis of updated AGV positions.

Obviously each AGV would have to guide itself and control its own loading and unloading at bays, but it was not as clear where software for navigation should reside. Should each AGV have its own 'warehouse map' and navigate on shortest route algorithms? In this case individual AGVs would have to consult the central computer before entering junctions or zones. (The central computer would be required to operate "traffic light" systems at such junctions and zones.) Alternatively, route calculations for all AGVs could be made by the central computer. The advantages of this over the former method are threefold. Firstly, optimum routes can be calculated as opposed to shortest routes. Since the central computer knows where each AGV is (AGVs are required to signal their position at critical positions such as when they are approaching a junction, or at given time intervals), it can

calculate alternative routes if an AGV is already in a particular zone or aisle. Secondly, since only the central computer has a warehouse map, more AGVs can be added, if required, with a minimum amount of reprogramming. Finally, changes made to the warehouse can be accounted for by a single reprogramming of the central computer rather than all of the AGVs.

Having reached the conclusion that, in a multi-vehicle environment, navigation can be handled more effectively by a central control, the research was directed towards the vehicle guidance itself. It was proposed that the information sent from the central computer to individual AGVs could be a string or queue of commands representing the central computers 'safe' planned path as a series of concatenated curved and straight lines. In essence then, a line following system was required as with wire guidance, but in order to make the system more flexible and the vehicles "free ranging", the lines were imaginary and had to be followed using some alternative method of sensing.

At this point it was noted that

- (1) A warehouse is a known, fixed, and structured environment.
- (2) The AGV's task in a warehouse is fixed and well defined.

The implication of these facts was that, since we have *a priori* knowledge about both the task and environment, many of the higher level functions such as environmental learning and map building used in current mobile robot research projects were not required and low cost, low level sensors could be employed.

In summary, the main aim of the research was to investigate ways in which AGVs could be made more "free ranging" by employing low cost, low level sensors. More specifically, a research AGV was built to facilitate an experimental approach to a number of initial aims which can be delineated

(1) The investigation of the processing of low level sensor data. In particular ultrasonic telemetry and odometry were chosen as two complementary low level sensing systems. These choices focused thought on how best to process aggregated telemetric results, using odometry in addition to the results themselves, in order to measure position both accurately and reliably.

(2) The examination of some of the control problems that arise when the experimental vehicle is asked to follow a sequence of curved and straight lines that constitute a planned route.

(3) The design and testing of modular software algorithms which integrate both the data processing and control aspects of the guidance method thus allowing the AGV to move through its environment in a smooth fashion.

### **1.5 Format of the Presented Work**

In the next chapter, a review of research concerned with both AGVs and mobile robots is given. In particular, an overview of what are considered to be the main areas of current research is presented. Although many aspects of the mentioned autonomous vehicles are not used in this research, examining the breadth of the field and current trends provides a good framework and perspective on the direction of this research and indicates what future work might be done.

In the third chapter, a description of the hardware used in this

research is given. This includes the mechanical hardware of the research vehicle, the various computing facilities that were employed, and the circuits developed to interface the vehicle's sensors and drive system to the on-board computer.

In the fourth chapter, the function of the two sensor systems in the proposed guidance method is described. In particular, position correction using ultrasonics, position estimation using odometry, and the way in which the two systems operate in a complementary manner is described. This chapter also details the performance of the sensing systems and notes and explains their observed characteristics. In addition, a method of estimating the current error in odometric position estimation is described.

In the next chapter, the various commands that the research vehicle can execute are described, a control strategy is formulated, and the calculations and geometrical transformations required to transform sensory data into controlled action are derived.

The sixth chapter documents a software package written in 'c' which was designed to integrate the data processing and control aspects of the guidance method in a modular fashion and provide the user with a flexible menu-driven interface. Initially an overview of the software structure is given and this is followed by a series of brief outlines of the constituent routines. In essence the chapter is concerned more with the functionality and structure of the software rather than the derivation and explanation of what is contained within the routines.

In the next chapter, the design, analysis, and performance of an efficient method of steering control is presented along with results which show the vehicle's behaviour when tracking both linear and curved paths. In addition, the results illustrate the behaviour of the vehicle when the control system is subject to step disturbances such as those experienced when position

corrections are made. Further sections in this chapter describe modifications to the basic control system that were required to improve curve tracking performance, to reduce rise time at low velocities, and to prevent stalling of the drive motors when the control system demands large torques.

In chapter 8, results are presented which show the data processing that allows the vehicle to correct its position as it moves through its environment. The methods described include recursive digital filtering, line regression, feature detection, compensation for offset errors in ultrasonic ranging, and the estimation of the effect of the random component of error in ultrasonic ranging using a 'confidence interval' basis. The final sections of this chapter draw the work to a conclusion by examining the performance of the system in more general terms using the results of a reasonably large scale test in which the vehicle is asked to follow a path from its base position to a goal position in the laboratory.

In the final chapter general conclusions are drawn and suggestions are made for further work.

## CHAPTER 2

# A REVIEW OF AUTONOMOUS VEHICLE RESEARCH

### 2.1 Introduction

Although wire guidance AGVs were in use as early as 1954 [HAMMOND, 1986], mobile robot research really began in the late sixties when an autonomous vehicle called SHAKEY was developed [NILSSON, 1969]. During the seventies, the impetus of research in this field was lost. Giralt [GIRALT, 1984] attributes this to a number of factors: lack of on-board computing power, lack of sensor technology, too remote real world applications and a direction of funds into manipulator type robots. Activity picked up again towards the end of the seventies when these factors were mitigated with improved microprocessor and sensor technology and the anticipation of new applications in both industrial and hostile environments.

It was the aim of this research to examine the use of low level sensors for the guidance of an AGV in a known, structured warehouse environment. Nothing, however, has been said about the vehicle performing its designated task, which is normally loading or unloading stock, once the vehicle reaches its destination location. Commonly wire guidance systems require the AGV to be located sufficiently accurately to allow static mechanisms, which are usually called loader tables, to carry out the loading and unloading operations. It should be noted here that the system proposed in this research would also require such a loading system. Given that an AGV has been provided with a means of free ranging guidance, which was the aim of this research, the next stage in the development of industrial automation with AGVs is to do away with static space consuming loaders and to provide the AGV with sufficient autonomy to handle free standing, palletted stock. This represents

a substantial step in increased vehicular autonomy which implicitly brings with it an increase in the number, sophistication, and complexity of AGV facets. As has been stated before, operational AGV systems are becoming increasingly common and this has been met with a corresponding increase in AGV and mobile robot research activity directed towards the degree of autonomy required for such a task as handling free standing pallets in a warehouse [BRADY, 1987].

It is the purpose of this chapter to give an overview of some of the important areas in autonomous vehicle research and to highlight current areas of research which may be directed towards providing industrial vehicles with a increased degree of autonomy.

In the following sections, a number of experimental systems developed over recent years are referred to. Synopses of several major experimental projects are given in appendix 1. In particular, sensory abilities, computing facilities, vehicle construction, aims and important areas of research are mentioned. As well as providing a readily available reference for this chapter, appendix 1 provides a perspective on how experimental systems have been developed over recent years.

The penultimate section in this chapter is used for a discussion. In particular, the relative merits of a low level and high level sensing methods are discussed in terms of application.

## **2.2 A Multisensory Approach**

One of the key ideas in current autonomous vehicle research is the desirability of a multisensory approach to the acquisition of environmental information. In general, the use of a variety of sensors is considered to be preferable to attempting to acquire large amounts of information from a single or small number of high level sensors (such as TV cameras).

A multisensory approach is attractive when the requirements of an autonomous vehicle's sensory system are considered. More specifically these requirements may be defined as

- (1) Speed of data acquisition.
- (2) Speed of data processing.
- (3) Globality of acquired data.
- (4) Accuracy and consistency of interpreted data.
- (5) Robustness to poor or spurious sensory data.

Since an AGV must perform its designated task of loading and unloading stock in real time, a premium is placed on both the speed of data acquisition and the speed of data processing. A multiplicity of sensors is obviously beneficial by allowing the collection of a large amount of data in a short time (i.e. when they operate in parallel). In addition to this, a multi-sensor system naturally lends itself to a distributed parallel processing architecture which allows the processing of large amounts of raw data to take place simultaneously.

It has been well noted in the research literature that any single sensor can only operate well in a given context. For example, TV cameras can only work well if good contrasts are engineered by structured lighting. Ultrasonic ranging devices require flat and almost orthogonal reflecting surfaces to work well. Also, whilst they give good depth measurements, they cannot easily measure angle. By contrast infrared and laser based sensors give good angular resolution but are poor when measuring depth. Since single sensors are always deficient in a number of respects, they can only measure specific properties concerning the state of the environment. In general different types of sensor do not suffer the same deficiency and so a multiple sensor system allows both

a more complete and a more accurate description of the environment.

Multiple sensor systems lend themselves to redundancy and robustness because multiple sets of observations on the same object can be used to form a consensus of opinion. This provides a reference with which to detect and reject both spurious data and data from sensors whose observations have been made under unfavourable conditions.

Considerations of globality, accuracy, and robustness of integrating sensory data can be discussed more conveniently under the heading of sensor models and sensor data fusion.

### **2.3 Sensor Models and Sensor Data Fusion**

Any AGV or mobile robot system that adopts a multisensory approach must address the problem of integrating the information provided by each sensor into a single consistent and coherent "best estimate" of the current state of the environment. This integration is often referred to as "sensor data fusion" and represents one of the main areas of current autonomous vehicle research.

In all cases of research concerning sensor fusion, it is recognised that sensor data is inherently uncertain and so it is natural to represent sensor behaviour using stochastic models. The problem of sensor fusion then becomes the determination of appropriate methods of manipulating these models to allow separate data sets from a number of different sensors, operating under different conditions and from different perspectives, to be merged into a single representation or "world model" of the environment.

Before giving an overview of some of the current ideas concerning sensor fusion, it is appropriate here to describe the ways in which multiple sensor data sets may interact. Firstly, sensor data may be non overlapping if, for example, two different objects are measured with the same or different

sensors. Such data can be thought to be complementary in that it provides a more global knowledge of the environment [DURRANT-WHYTE, 1987]. Note that, in these non overlapping cases, a fusion problem still exists since, if the vehicle has a representation in its world model of the object being measured, that model must reflect the integrated form of all previous measurements made on the object. Here the similarity between sensor models and geometric (as opposed to topological) world models becomes evident. Since a geometric world representation must be built up with stochastically modelled noisy sensors, that model itself must be stochastic. In effect, sensor and geometric world models must be one of the same kind since that world model is simply a spatio-temporal history of uncertain sensor observations. Such a compatibility is required to allow new sensor data to be integrated appropriately into the vehicle's knowledge of its environment.

Secondly, sensor data can be described as cooperative if the information provided by one sensor allows another sensor to extract information more efficiently from the environment. Harmon [HARMON, 1987] describes this process as guiding and recognises that high level sensors should be guided by coarser, lower-level sensors to areas of interest. The need for sensor guidance arises from the high speed requirement of a system operating in real time. High level sensors, such as TV cameras, provide large amounts of information which takes time to process. If these sensors are not appropriately guided, many situations may be deemed to be "uninteresting" rendering the processing required to get to the position of making that decision useless and the time taken for that processing is wasted.

Finally in many cases several data or sets of data may measure the same geometric feature in the environment. In all such cases, a degree of discrepancy is expected due to the uncertain nature of all sensor observations. It is the function of sensor fusion to determine the most appropriate method

of resolving this discrepancy using the stochastic sensor models it has available to it. If the discrepancy is large, it may be decided that one set is in error and should be ignored completely.

Luo [LUO et al, 1988] models sensor characteristics as Gaussian probability density functions which allows the use of confidence distance measurements in sensor fusion. If two observations are made of the same feature, each observation being associated with a different Gaussian probability density function (pdf), then the confidence of one observation with respect to the other can be expressed as a percentage which is dependent on the distance apart of the two measurements and the reference pdf. Such confidence distance measurements are thresholded to produce a so called relational matrix between sensors. This matrix is simply a binary representation indicating that a particular sensor either does (1), or does not (0), support another sensor's observations. Sensor observations can then be clustered and this is done by determining the set of observations in which each observation is said to support every other observation in the clustered group. These data are then fused and any other (unfused) data sets which are supported by this fused data are deemed to be correct and are fused into the existing result of data fusion. Any remaining data that supports this new fused data is deemed to be slightly deviant from the actual data value whilst remaining data sets that do not support the fused data are considered to be erroneous. Both deviant and erroneous data sets are rejected for the fusion process.

Durrant-Whyte [DURRANT-WHYTE, 1987, 1988] has described methods using Bayes procedures to integrate disparate sensor observations. Objects in the environment are modelled as a topological network of geometric features. Each of these features has an uncertainty associated with it, which is represented by a probability density function on the parameter vector of that feature. Methods are described for transforming these pdfs between different

coordinate systems. This is important since, if the invariant topology of objects in the AGVs environment are known, observations made on different features of the same object by the same sensor relocated, or by different sensors, can be integrated to give an estimate of that object's position in six dimensional space. The use of invariant topology makes maximum use of sensor information since it provides a method to integrate observations made in different locations on different parts of an object.

In this section, it has been noted that the type of "world model" associated with sensor models and sensor data fusion is one which contains the detail of geometric details and their associated uncertainty. Often such detailed information is abstracted to produce simpler representations on which to base global plans and reasoning. Such aspects are discussed in the following section.

## **2.4 World Modelling**

Much recent research has been concerned with finding appropriate forms of representation of environmental knowledge. Such representations are commonly termed "world models" or "world maps". Knowledge of the environment is required on many levels of behaviour from high level global decisions to low level guidance and, as a consequence of this, environmental maps may take more than one form.

A typical example of this is an AGV in a factory which has been given the task of visiting a number of workstations in a set sequence. In order to search for an appropriate path, a map is required which provides information about the topology or global connectivity of the workstations and paths between workstations. Although such topological maps are necessary for route planning, a different type of map may be required when the AGV is near a workstation performing its designated task. In this situation, the

vehicle requires more detail concerning the position and geometry of objects in its environment and the static environment itself. Such environmental knowledge resides in representations which are often called local or geometric maps.

Chatila and Laumond [CHATILA and LAUMOND, 1985] define the concepts of places and connectors for the mobile robot HILARE [Appendix 1]. Places are functional or topological units joined by connectors and both places and their topology may exist on several levels. This idea becomes clear if the analogy of finding a particular house in a particular town is considered. On an initial level the places are towns and the connectors are major roads. Once the destination place is reached, a new more detailed topology is entered which is a street map in which streets can be thought of as both places and connectors. Once the street is reached, the lowest level of place is the house which exists within the place "street". The geometric model employed by HILARE is developed by structuring space into places which are uncertain planar convex polygons. In the absence of unexpected obstacles an AGV can travel freely to any position within its current convex polygon thus the lowest or most detailed topological model is constructed by connecting each convex place with all those that border it.

A research project at CMU [ELFES, 1986, 1987] [Appendix 1] has been concerned with environmental mapping using a ring of 24 Polaroid ultrasonic transducers. The aim was to overcome the problems of ranging ambiguity due to the transducers wide beam width by making observations from many different perspectives and employing a probabilistic approach to building maps. In this project many different representations of sonar maps are derived from the initial probabilistic local maps and these are conceptually thought to lie along "representation axes". Firstly there is the abstraction axis, along which a geometric local map can be derived by merging localised areas which have

a high probability of being occupied into polygonal boundaries. Information from this geometric level can be abstracted further to yield a topological map which is required for higher level decision making. The geographical axis is simply a hierarchy of maps with increasing levels of coverage; three types are defined: views are what is in the "visible" range, local maps represent places, and global maps are sets of local maps covering the whole of the mapped environment. Finally the resolution axis represents the fact that maps can exist at different levels of resolution. This can be useful for the process of guiding as was described in the previous section.

This and the previous two sections have been concerned with how an AGV might derive and represent information from its environment. In determining the overall goals and subgoals that manifest themselves as "intelligent behaviour", knowledge other than that derived from sensory observations (*a priori* semantic knowledge, for example) may be employed by a mobile robot. This draws in questions of knowledge representation and artificial intelligence which are beyond the scope of this research in terms of application. (A warehouse AGV's destination and route would be determined by a central controlling computer.) It is sufficient to note here that irrespective of the degree of "intelligent behaviour" that is proposed for an AGV or mobile robot, such a vehicle can only function well if modularisation, communication, and other aspects of the structuring of complex computational systems are carefully thought out. This is the concern of the following section.

## **2.5 The Structuring of Autonomous Vehicle Systems**

AGV and mobile robot projects are being undertaken in which an increased degree of autonomy is aimed for, allowing increasingly complex tasks to be delegated to vehicles operating in increasingly unstructured environments. These new levels of autonomy necessitate an increase in the number of

components in the system (sensors, manipulators, and so on) and a large increase in the amount of software processing. Despite these increases in complexity, the system must still operate quickly, efficiently, and robustly in real time and thus the necessity of efficient methods of structuring and integrating such a complex system is paramount.

A number of requirements are evident when structuring a prototype AGV system. Firstly, since the field of AGV research is changing rapidly, research vehicles must be highly extensible and easily alterable. Secondly, the requirement of robustness and an ability to debug errors demands a modular approach [HARMON, 1987]. This is particularly important when one considers that, in general, several people will be working on the same project at the same time. Finally, there are the performance requirements of speed and efficient operation in real time.

As has been noted, many current research projects are concerned with vehicles that have a number of sensors available to them. In general, for speed of operation, a number of sensors will observe the environment simultaneously. Since these sensory systems are themselves distributed and since many of these systems require a substantial amount of processing to extract useful information, it is natural that sensory data processing should also be distributed into units operating in parallel.

Once it is appreciated that an AGVs computational structure is likely to consist of a number of intelligent processing units operating in parallel, the problem of system organisation must be addressed. The most common approach to any goal-orientated computational system is a hierarchical decomposition of the problem. Here an overall goal, which may be self generated or simply delegated, is decomposed into a number of high level tasks which themselves are divided into lower level functions. This decomposition continues through a number of levels in the hierarchy until parts of the

AGVs mechanical system are actuated to effect the action specified by the initial goal. Sensory processing, which on a number of levels can be regarded as world model building, may also be hierarchical but here the abstraction increases on passing through the layers of hierarchy until a representation is produced upon which the AGV can make decisions.

The advantage of hierarchical organisations is that their behaviour is relatively easy to assess which makes them easy to control and debug. However, such a structure as that proposed above, which is completely hierarchical, only allows information to flow through levels of the hierarchy and thus does not permit sensory interactive behaviour. In the single hierarchical loop of sensing the environment and operating on the environment, long processing times will necessarily be encountered and much sensory data is likely to be lost in the process of abstraction.

Albus [ALBUS, 1981] states that in order to cope with uncertainty and recover from errors as new sensory data becomes available, sensory data of different levels of abstraction must be able to interact with the control system at many different levels. An interesting point made here is that although high levels of data abstraction take up a substantial amount of processing time, the behavioural decisions that require such representations need to be made less frequently than for example primitive low level servo feedbacks, and so can tolerate the longer processing times.

These communication abilities between different levels of sensor data and control layers, if carried to its limit, would result in a heterarchical control. In such a system there is no overall supervision of the task and any element can communicate with any other in the system. Although such an arrangement can make maximum use of sensor data, it is very difficult to control.

The physical architecture adopted by many autonomous vehicle com-

puting systems is based around a so called "blackboard" structure. A distributed implementation of a blackboard is sometimes called a whiteboard [SHAFER et al, 1986] [HARMON, 1986, 1987] and is one of a number of recent distributed AI architectures. A "blackboard" is essentially a database and the flow of control between different modules (which are simply sections of code) is handled by a blackboard manager. Modules can only execute if a number of conditions exist and the manager directs the system by selecting the most appropriate module that is ready to execute. If the modules exist in a distributed architecture, then they can run in parallel and use their local intelligence to decide what information to retrieve from the database and what (condensed) information to store in the database.

Brooks [BROOKS, 1986] has proposed a so called "layered" decomposition of a mobile robot control system in which the system decomposition is based upon task-achieving behaviours as opposed to a traditional hierarchical decomposition of the problem. In this approach a complete robot system is built which is competent in a certain task. The lowest level of task in this case is to avoid objects and 'run away' from approaching mobiles. Once such a 'layer of control' has been debugged and is working, other layers can be added on top of that layer. Such subsequent layers may include wandering, route following, path planning, exploring and building maps. In effect, each layer executes its own function with the added condition that inputs to modules can be suppressed and outputs inhibited in order to allow higher levels to subsume the role of lower levels. Such a structure is robust since if any layer fails, all those below it can still function and thus the system can maintain a degree of intelligent behaviour. In addition, it is extensible since additional layers of behaviour can be added without any changes being made to the original system. This type of layered architecture does have its drawbacks in that additional layers rapidly become more complex and careful design is

required in order to make the communications within the system effective. In spite of this, Brooks' so called 'subsumption architecture' has achieved a good measure of popularity in recent research literature, particularly amongst those researchers who like to parallel the behaviour of mobile robots with animate behaviour. The behaviour of a vehicle which has an 'avoid and run away' level of competence in a subsumption architecture is more reflexive and more elegant than an explicit implementation of obstacle avoidance whereby the vehicle has to stop and think about its next course of action. In addition, an important advantage that the arrangement has over blackboard architectures is that bottlenecks in data access are not encountered which impose a limitation on the bandwidth of the system [BRADY et al, 1989].

As has been indicated above, it is generally accepted that the computational architecture of a multisensory autonomous vehicle should consist of a number of processing units operating in parallel. In coupling such a system together, it seems that the restrictions imposed by strict hierarchies should be avoided, yet the communications between the different parts of the system should be carefully thought out, and have sufficient definition to allow goal-orientated behaviour.

## **2.6 Discussion: The Role of Low Level Guidance**

Although many points brought up in this chapter will not be discussed further in this thesis, they have been introduced because of their relevance in terms of application. In the next stage of autonomy, an industrial AGV may have to locate itself accurately at a bay and handle free standing palleted stock. Such a developed system would have had to address the problems of a complex multisensory distributed system that have been mentioned in this chapter.

If it is possible to provide an AGV with such a degree of autonomy,

then why not do away with artificial objects introduced into the environment as reference points? This may be possible to some extent, but again one has to consider the implications in terms of real-time. When a vehicle is moving into position in order to perform its delegated task, its movements and manoeuvres may be slow enough to permit real-time multisensory cooperation thus eliminating the need for an external device for loading and unloading. However, when the vehicle is moving between bays, it needs to move quickly (1m/s) in order to be efficient and, in addition, precise location is not crucial as it is for manoeuvring and handling pallets. In this case, positional information needs to be generated rapidly in order to implement a steering control algorithm that does not limit the speed of the vehicle.

Throughout this chapter, sensing has been discussed in terms of mapping the environment in some fashion. This map, or abstracted forms of this map, may then be used for relatively high level functions such as determining the next destination and how to navigate to that destination. On the lowest level of functionality, however, sensory information must provide the control feedback necessary to actuate speed and steering changes that move the vehicle nearer to its desired course whilst at the same time avoiding collisions with obstacles.

In many autonomous vehicle research projects, high level sensors such as TV cameras, which provide a large amount of perceptual information, have been employed. A typical approach in vision analysis is to analyse grey levels for edge detection and other features. This yields a vast amount of data to be processed. For example, for an array with 512 by 512 pixels each with 16 grey levels (4-bits), over a million bits (131000 bytes) of information is required. The obvious consequence of this, as Moravec [MORAVEC, 1985] noted, is long data processing times. This may be tolerable for scene analysis in, for example, a 'learning' mode when the vehicle is stationary or moving

slowly, but may limit the speed of a vehicle that needs to move around its known environment quickly and efficiently.

In a project conducted at the Carnegie-Mellon University, vision has been employed to implement a robot road following guidance, and speeds of 1.08 km/h (0.3 m/s) have been reported [WALLACE et al, 1986]. However, the environment that the vehicle operates in is open and uncluttered and can thus accommodate a reported image processing and steering servo loop time of 3 seconds (in which time the vehicle will have travelled 0.9m). Such a scheme may not be suitable for a warehouse environment, firstly because it may be difficult to engineer good contrasts at the edges of the vehicle's path, and secondly a warehouse environment is naturally more constrained and cluttered, which demands a higher sampling frequency for the steering control, or conversely a lower travelling speed.

In contrast to high level sensors, low level sensors provide small amounts or single pieces of information that require little processing and thus they can operate at a rapid rate. However, since their information content is low, they are often used in conjunction with artificial reference points added to the environment. For example, one common method of determining absolute position is position triangulation in which the vehicle's position is measured relative to two fixed reference points. (The vehicle and each of the reference points form three points of a triangle.) Both HILARE and the GEC-Caterpillar system employ this method. The GEC-Caterpillar system [BEVAN, 1986] implements this method by rotating a laser beam in the same plane as a number of fixed bar codes. If the AGV knows the exact position associated with each code and can measure its angle relative to two such bar coded targets it can determine absolutely its planar environmental coordinates  $x, y$  and  $\theta$ .

As well as triangulating global position from two fixed reference points,

global position can be measured using the distance away from and angle to a single reference point. Range measurements can be made by timing reflected beams of ultrasound or light/infrared. Since ultrasound is considerably slower, the time of flight of the beam is longer giving more accurate depth readings. Although ultrasonic methods give reasonably accurate depth measurements, their angular resolution is very poor making it difficult to reference position from a single target. Another possibility is to reflect an infrared beam off the reference target. In contrast to ultrasonic ranging, infrared has poor depth resolution but good angle resolution. Flynn [FLYNN, 1985, 1988] experimentally characterised both an ultrasonic sensor and an infrared sensor and then formulated a set of rules for how the information from each sensor should be combined to build up a picture of the environment. The conclusion, then, is that it should be possible to reference AGV position from a single known target using the combined information from complementary low level sensors.

## 2.7 Summary

In this chapter an attempt has been made to give an overview rather than the detail of some of the current ideas in the field of autonomous vehicle research. In particular these can be summarised as

- (1) Finding new or improved methods of both sensing itself and sensory data processing.
- (2) Modelling both sensor behaviour and the environment itself.
- (3) Methods of manipulating sensory data via sensor models in order to provide an accurate and up to date world model.
- (4) The conceptual and physical structuring of complex mul-

tisensory autonomous vehicle systems.

In addition to the research areas mentioned in this chapter, a number of other areas such as artificial intelligence, knowledge representation, and expert systems will affect the future development of autonomous vehicles. Limitations may be set by the rate at which new or improved methods of processing become available. One particular example is the development of neural nets. It may be found that future autonomous vehicle systems employ a variety of combinations of conventional control architectures, neural net structures, and expert systems.

Work in this thesis is concerned with using low level sensor systems, namely odometry and ultrasonic telemetry, in conjunction with so called 'correction boards' added to the vehicle's environment. Obviously, the use of such a minimal set of sensors can not provide a vehicle with a high degree of autonomy. Rather, the aim is to investigate the use of low level sensors in the implementation of a method of free ranging guidance which does not limit the speed or efficiency of the vehicle's movements.

The following chapter describes the hardware which was designed to support an experimental approach to the research.

## CHAPTER 3

# THE RESEARCH VEHICLE, COMPUTATIONAL RESOURCES, AND THE SENSOR INTERFACE HARDWARE

### 3.1 Introduction

To implement the proposed AGV research, it was decided that a simple vehicle and research rig should be built rather than employing a software simulation. The philosophy behind this approach has been expressed several times in the research literature. Nilsson [NILSSON, 1969] referring to the mobile automaton SHAKEY states "Whereas much can be learned by simulating certain of the necessary functions, many issues are not likely to be anticipated at all in simulations." Also Flynn [FLYNN, 1985] states that a software simulation is often inappropriate because algorithms need to be designed for poor data and not perfect data, and it is difficult to anticipate the ways in which the data is poor.

In this chapter the construction of a research AGV, the available computational resources, and the development of the subsidiary hardware needed to conduct this research are described.

### 3.2 The Research Vehicle

A simple vehicle has been constructed to support this AGV research. A plan and side elevation are shown in figures 3.1a and 3.1b respectively whilst front and rear views of the vehicle are shown in plates 3.1 and 3.2 respectively. Steering and propulsion are provided by two rear wheels, whilst a swivel castor gives the necessary support at the front. Steering is implemented by controlling the speed differential between the two independent rear drive wheels. These are driven by Superior Electric 'Slo-Syn' 3V, 4A

stepping motors geared down in the ratio 7:36 through timing belts. The drive motors are arranged so that their current supply can be switched between *normal* and *boost*. This allows battery power to be conserved whilst running standard tests, yet full power is available for demonstrations in which large demand torques are generated from paths of high curvature. A three wheeled configuration for the research vehicle was chosen for the following reasons

- (1) Since only three wheels are used, there is no need for suspension.
- (2) The mechanical arrangement itself is very simple. If the front wheel were steered, a gear differential would be required on the back wheels.
- (3) The arrangement affords high manoeuvrability in the tight confines of the laboratory.
- (4) Speed control of the stepper motors is straightforward.

The plan of the AGV in figure 3.1a also shows how the sensors are mounted. Optical shaft encoders are used to generate odometric information and are mounted on each drive wheel providing one hundred pulses per revolution. (Employing shaft encoders was considered more reliable than counting the pulses supplied to the stepper motors since their use facilitates a means of testing for stepper motor stall.) Five sets of ultrasonic transducer are mounted around the perimeter of the vehicle, one at each corner to generate control and position correction information, and one at the front to sense obstacles.

The MC68000 microcomputer with its associated interface boards are housed in a '6u' sized cabinet which is racked into the front of the AGV. The interface electronics for the sensors and the stepper translators are in a

'3u' sized cabinet. This is racked above the microcomputer.

Both of the stepper motors and all of the interface electronics are driven by two 12V, 36Ah car batteries. In order to reduce the rotational inertia of the AGV in the horizontal (yaw) plane, these batteries are mounted close to the vehicle's centre of rotation, as is illustrated in figures 3.1a and 3.1b.

### 3.3 Computational Resources

#### 3.3.1 Overview

A diagram of the AGV research rig and its computational resources is shown in figure 3.2. Software development took place on a Cambridge Micro Systems 'CODATA'. This is a UNIX/MC68020 based minicomputer which has resident software to support the development of 'c' programs (compiler, verifier, beautifier, and so on). The AGV itself has an on-board microcomputer which was developed by the Durham University Microprocessor Centre to provide specialist support for the computing requirements of a variety of scientific research in the university. This is based on the Motorola MC68000 16-bit microprocessor and has 192k of memory and a number of interface boards associated with it (digital input/output, ADC, timers).

A TVI 905 is used as a dumb terminal which either allows the operator to access the host computer for program development or allows communication with the on-board computer for data entry and menu options during experimentation. The communications links are realised by two RS-232 leads which, along with a mains cord to supply the on-board computer, form a trailing lead to the research vehicle.

The method of results analysis requires the AGVs on-board computer to store data on a floppy disc after a test run. This disc is then transferred

to a personal computer which runs a program specially written to read the structure of the file and provide general data analysis and graph plotting facilities.

These components will now be discussed in more detail.

### 3.3.2 Program Development

Most of the software for the AGV was written in the 'c' programming language. This choice was made as it has many inherent features accordant with the requirements of real-time control. Firstly, the richness of its operators and control structures yields a high enough programming level to facilitate an efficient transcription of algorithms into 'c' code. Conversely, it is of a low enough level to access specific memory addresses and be compiled into efficient MC68000 source code. (It is possible to stop the compilation at the source code level and verify this.) Secondly, robust software can be created because functions are afforded a high degree of modularity through their 'call by value' nature.

Although most of the software was written in 'c' and cross compiled to MC68000 object code, some routines were written directly in MC68000 source code. In the first instance this was to manipulate registers in the microcomputers interface chips, initially in order to configure them and then to handle input/output signals. In addition, when interrupt software had to run as fast as possible, the service routines were written directly in assembler to ensure the code had optimum efficiency.

Since function names are just global addresses, transfer of control between 'c' and assembler is simple. In addition data passed by 'c' to any routine is put onto the stack which assembler routines can access by addressing via the stack pointer.

### 3.3.3 The MC68000 Based On-board Microcomputer

The on-board MC68000 based microcomputer runs at a clock frequency of 8MHz and has four interface boards associated with it.

The first of these is a 64 bit digital input/output card. This uses four Motorola 6522 VIA chips to provide eight ports each giving eight bits of input/output and two handshake/interrupt lines. These chips also provide eight timers of sixteen bits which can count the clock. (The peripheral clock frequency is divided by ten to 0.8MHz.) Four of these can optionally control an output whilst the other four can count pulses on an input.

Another of the microcomputers interface cards is an eight channel, twelve bit A to D converter. This provides eight analogue inputs, a sample and hold amplifier, and 12-bit A to D conversion. The board can be programmed to scan through any of the eight channels each time the conversion is started and it can be configured to interrupt the processor when either a single reading is available or a scan has been completed.

The third interface card is based on the AM9513 System Timing Controller (STC) which is made by Advanced Micro Devices Inc. The AM9513 IC contains five independent sixteen bit counters with a number of options for source and gate inputs. Four of these chips are available on the board and each of these can be used in a number of different modes of operation. The board is constructed so that two counter pairs from each chip have their inputs and outputs buffered and taken to a connector giving eight 32-bit counters in all.

Finally there is a disc controller board to control the on-board disc drive.

### 3.4 Development of the Interface Electronics

Four circuits have been designed for the research AGV: one to interface the ultrasonic sensors to the microcomputer, one to interface the optical shaft encoders to the microcomputer, a power supply board, and a switching and opto-isolation board. A brief description of each will be given in the following sections.

#### 3.4.1 Development of an Ultrasonic Ranging Unit

Initially two small 7 volt RS transducers were purchased; an ultrasonic transmitter and an ultrasonic receiver. With preliminary tests, it was found that a signal could be detected at reflecting ranges of up to two metres or four metres direct travelling distance. The added noise on the signal and the critical factor of orientation of the target object, however, suggested that object range may be measured reliably at around 1.5m.

At this point it was noted that a number of research projects have made use of Polaroid's ultrasonic ranging device which claims a range of up to ten metres. In this device, the transducer itself has a larger diameter and can accept much larger voltages across it. This makes for better range of operation and greater measurement accuracy. One major disadvantage with the device, however, is the fact that measurements are made at a frequency of 10Hz and it was envisioned that faster pulse rates would be required. Although this problem may have been solved by modifications to the device, it was felt that a 1.5m range was sufficient to support investigations into the use of ultrasonic sensing in vehicle guidance. In addition, a number of sensor pairs were to be used around the vehicle which placed a premium on the requirement of a low cost device. For these reasons, it was decided that cheaper low voltage transducers should be employed initially.

The basic operation of the ultrasonic ranging system is as follows. The transmit electronics provides the transmitter with a short burst of 40KHz at regular intervals, typically 10mS. At the same time, a high level interrupt signal is sent to the computer which starts a peripheral timing device. The ultrasound then bounces off the nearest object in its path that reflects the signal back in the same direction as it came and, a short time later, is detected by the receiver. The receive electronics then sends another high level interrupt signal to the computer which stops and reads the peripheral timing device. The time delay between transmitting and receiving the pulse is converted to a range measurement which is stored for use by data processing software which is entered on a low priority timed interrupt every 100mS.

The schematic diagram in figure 3.3a shows the circuit operation in more detail. The output of a variable frequency, variable pulse width TTL square wave generator is combined with the output of a tunable 40KHz square wave generator through an AND gate. When combined in this fashion, the low frequency wave is effectively the envelope of the high frequency wave giving pulses of 40KHz with alterable frequency and pulse length. Figure 3.3b shows some of the waveforms generated by the transmit electronics. Amplification is required after the AND gate since the chip can only provide 5V which does not make full use of the transducers power capability.

The received reflected signals are of the order of tens of millivolts and so amplification is required before any further signal processing can take place. Rectification and then filtering take place to form a voltage envelope that follows the positive peaks of the 40KHz amplified signal. This is then detected by a simple comparator circuit with a threshold of 200mV, and squared using a schmitt trigger to provide the stop signal. (This schmitt trigger also acts as a TTL buffer for the VIA.)

A complete circuit diagram of the interface board is shown in figure

3.4. It can be noted that two stages of amplification are employed after the ultrasonic receiver. This is because of the limited bandwidth of the basic 741 operational amplifier. The bandwidth is typically 1MHz and so with a 40kHz signal the maximum gain is 25. The gain needed was about 100 and so two stages of gain 10 were employed. Two other points are noteworthy. The 100  $\mu F$  capacitor on the third operational amplifier in the 'receive' circuit was required to ensure no fluctuations in reference level could occur. This is necessary for consistent measurements. Note also that transmitters on the board share the same basic transmit signal. As well as saving space and effort, this has a fundamental role in the control of the vehicle since range measurements need to be made simultaneously by sensors at the front and rear of the vehicle.

### 3.4.2 The Optical Shaft Encoder Interface

Each shaft encoder provides two pseudo-sine waves with 100 cycles per revolution of a drive wheel. The sine waves are approximately 90 degrees out of phase with each other, they have an offset voltage of 1V and an amplitude of 0.2V. Two signals are generated so that a signal can be developed to indicate direction of rotation.

The simplest way to produce square pulses from the output of the sensors is to compare the signal with a fixed voltage level of 1V. This was found to be unsatisfactory because the sensor signals picked up 40KHz noise from the ultrasonics board which caused extra triggering of the comparator and a pulse count of more than 100 per revolution of a drive wheel.

To prevent false counts from noise, a hysteresis band was incorporated in the comparator as shown in figure 3.5. The width of the band was designed to be greater than the peak to peak value of the noise, but less than the peak to peak value of the fundamental signal. Calculations for this band-

width and other circuit parameters are given in appendix 2. To facilitate the incorporation of a hysteresis band, the sensor signal first had to be amplified; additionally a capacitor was included in the operational amplifier feedback to prevent amplification of high frequency noise. Prior to amplification, the dc level (1V) on the signal is filtered out with a  $10 \mu F$  capacitor to prevent it saturating the amplifier. This filtering has a second beneficial effect in that the setting of a comparator level is no longer required. Once square waves had been developed, they simply had to be diode limited and buffered through a schmitt trigger to provide pulses for the computer to count.

### 3.4.3 The Power Supply Board

The stepper motor translators for the drive wheels require a 24V unregulated DC supply. The cheapest and simplest way of providing this voltage with the required current capability was with two 12V, 36Ah car batteries connected in series. This, then, is the fundamental power supply for the AGV.

A power supply card has been designed to derive the ultrasonic interface and shaft encoder interface board supplies (+15V, -15V, +5V) from the two batteries. This is shown in figure 3.6. To provide  $\pm 15V$ , voltage conversion was obviously required since the two batteries in series can only provide a total voltage swing of 24V. It was already necessary to regulate down to +5V and so 5 to 15V converters were employed. Unfortunately, DC-DC converters can only provide a relatively small current, in this case 100mA. Additionally they are expensive. In anticipation of further operational amplifier based circuits, a variable bipolar supply was incorporated on the power supply card. This was to be used for additional operational amplifiers not requiring the full  $\pm 15V$  supply and because it is regulator based, it has

a much better maximum current output than the converters. To facilitate the generation of this bipolar supply the centre point of the batteries had to be grounded to the computer. (This explains why 5V is regulated down from 12V and not 24V.)

It can be seen in figure 3.6 that there are two sets of 5V regulation followed by voltage conversion to  $\pm 15V$ . There are two reasons for this. Firstly, more than 100mA total current output was required and, secondly, when both sensor interface boards ran off the same converter, they interfered with each other. In fact, it was found that the converters were generally susceptible to noise in the system. To help suppress this noise, two  $1000\mu F$  capacitors are attached to the input pins of the converters.

#### 3.4.4 The Switching and Opto-isolation Board

Figure 3.7 shows the system of opto-isolation which allows a 24V supply to the stepper translators simultaneously with the basic  $\pm 12V$  supply necessary to drive the sensor interface boards via the power supply board. It can be seen that all of the signals required to control the stepper motors (clock pulses, direction, power enable) have to pass through opto-isolators because the common of the translators is 12V below the computer ground from which the control signals are derived.

After opto-isolation, the signals pass through darlington drivers which prevent voltage levels dropping because of current drainage. Since the translator's input signals all require 12V logic, pull up resistors are attached between the open collector outputs of both the opto-isolation and current amplification stages, and a 12V regulated supply (from 24V) available on the translator boards.

### 3.5 Summary

This chapter has described the construction of a research AGV, the computational resources employed, and the interface hardware that has been designed. Such information is appropriate at this point, firstly because the way in which the ultrasonic interface was designed affects its performance. This sensing performance is analysed in the following chapter. In addition, the vehicle construction dictates some of the lower level geometrical calculations required to implement the proposed control. These are described in the fifth chapter.

Dimensions in mm (not to scale)  
 Frame width is 25mm.

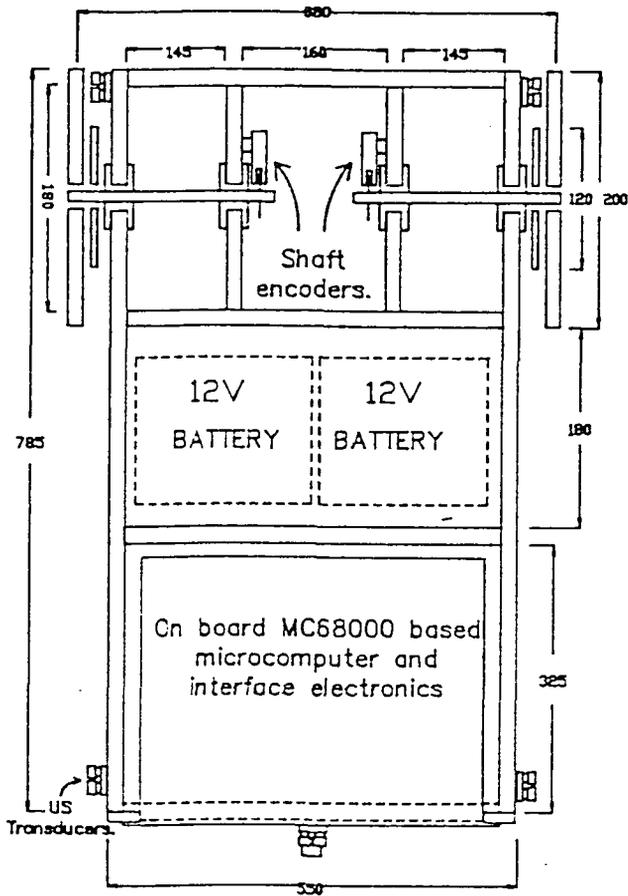


Figure 3.1a Plan view of the research AGV

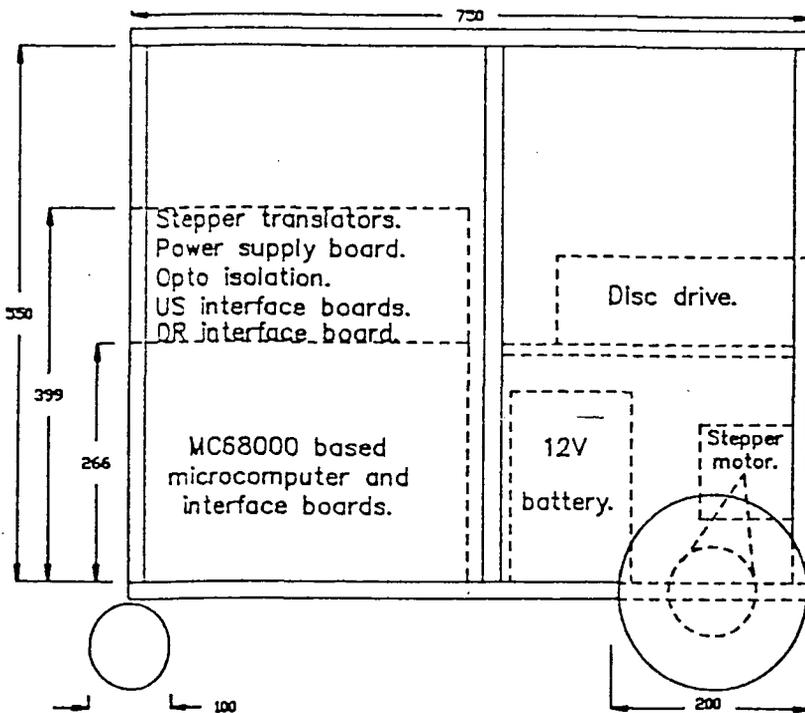


Figure 3.1b Side view of the research AGV

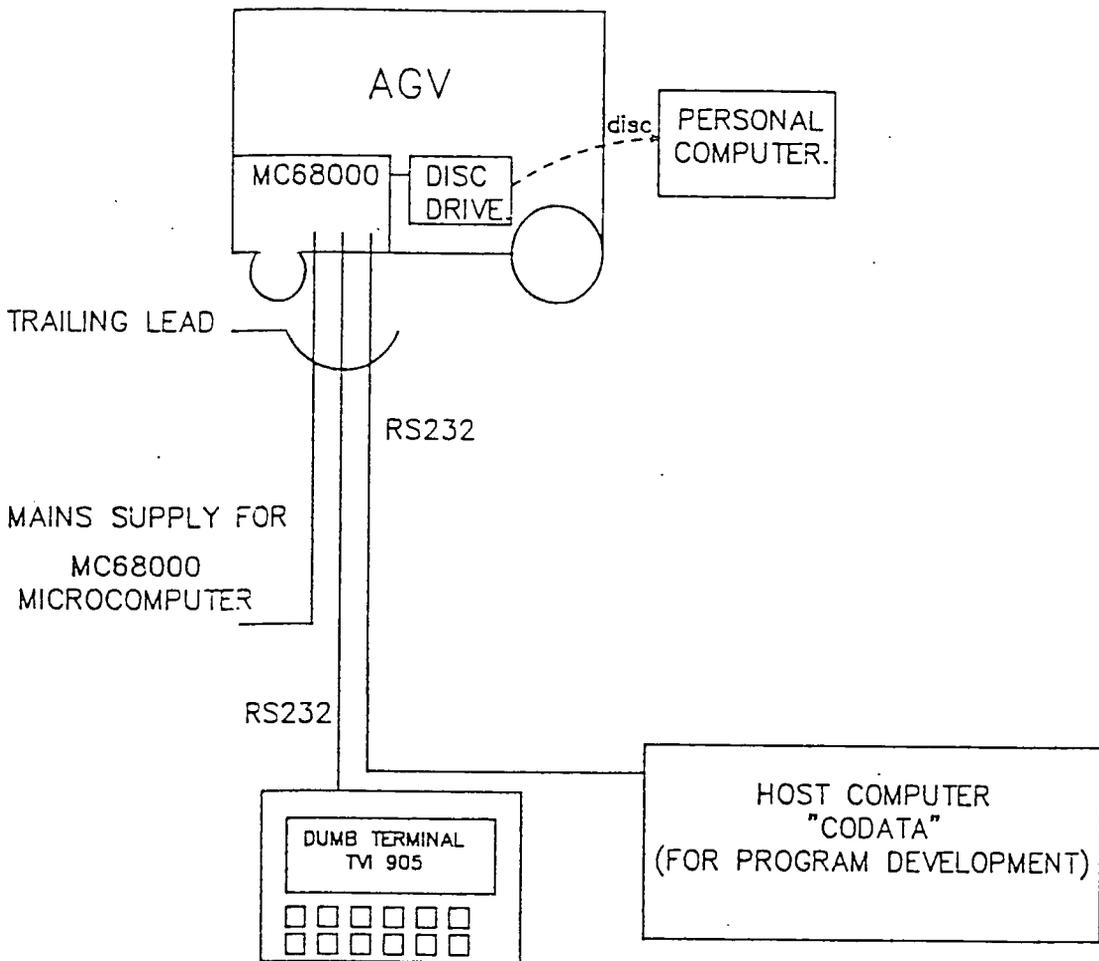


Figure 3.2 The AGV research rig

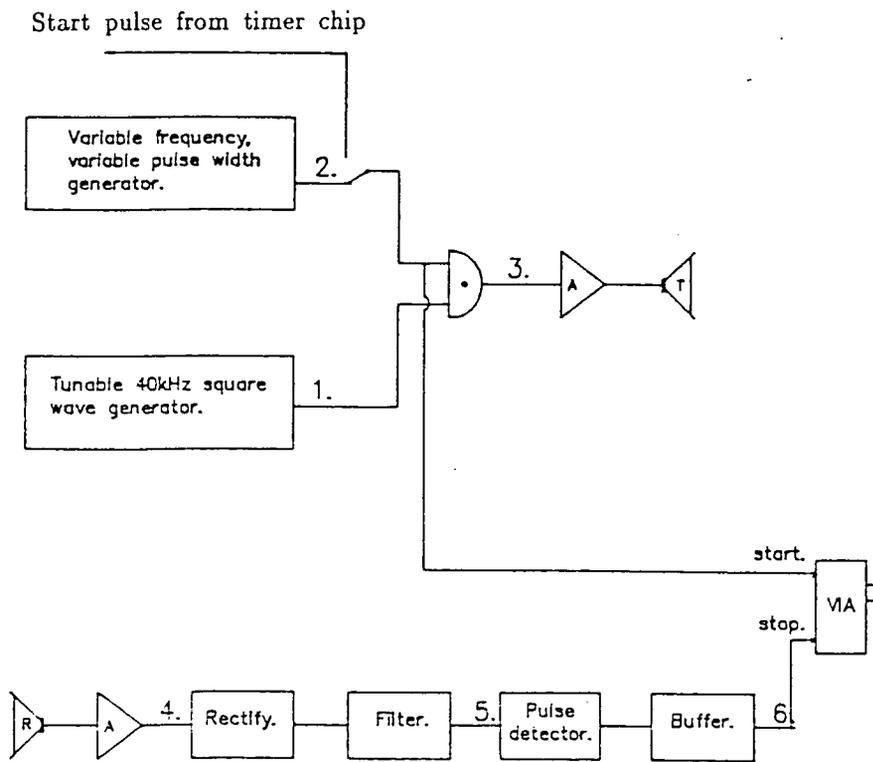


Figure 3.3a A schematic diagram of the ultrasonic sensor interface

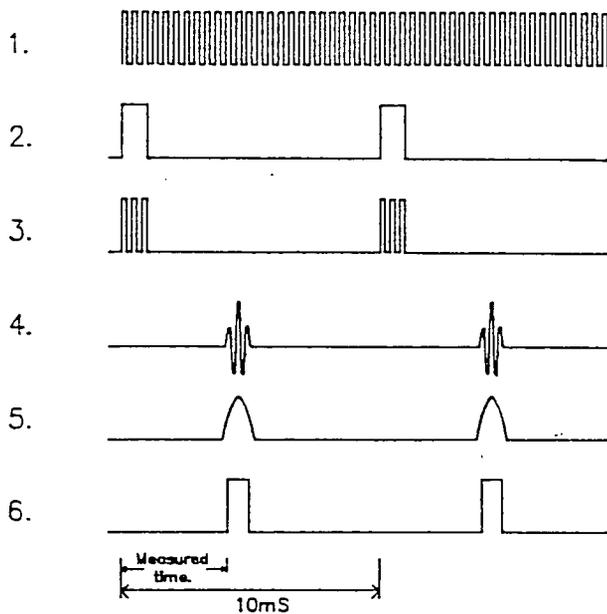


Figure 3.3b Waveforms in the ultrasonic sensor interface

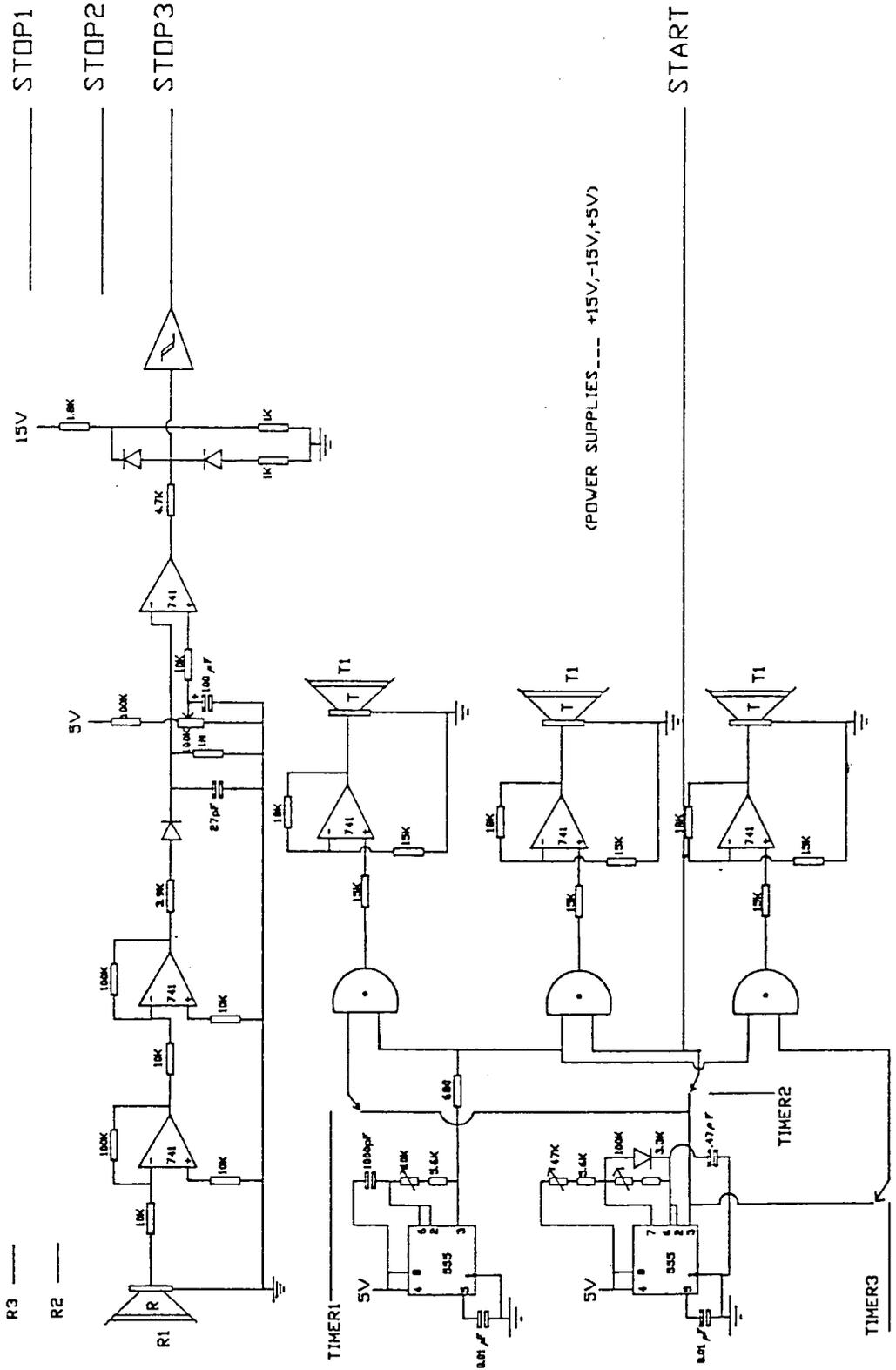


Figure 3.4 The ultrasonic sensor interface

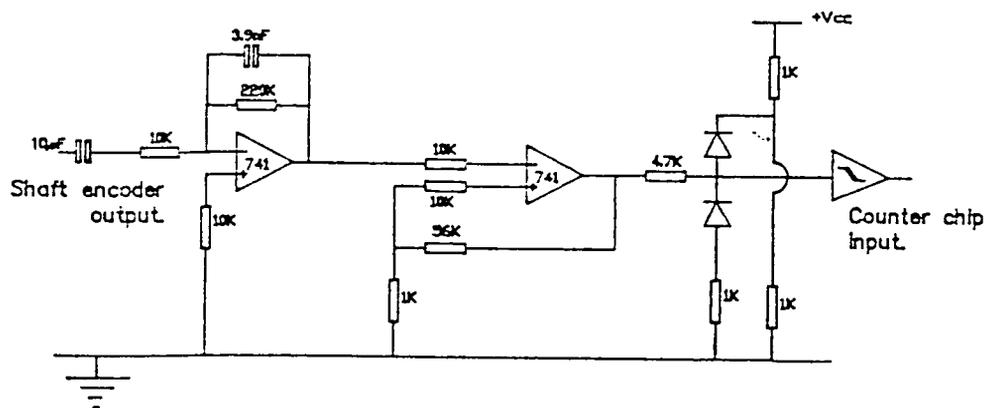


Figure 3.5 The shaft encoder interface

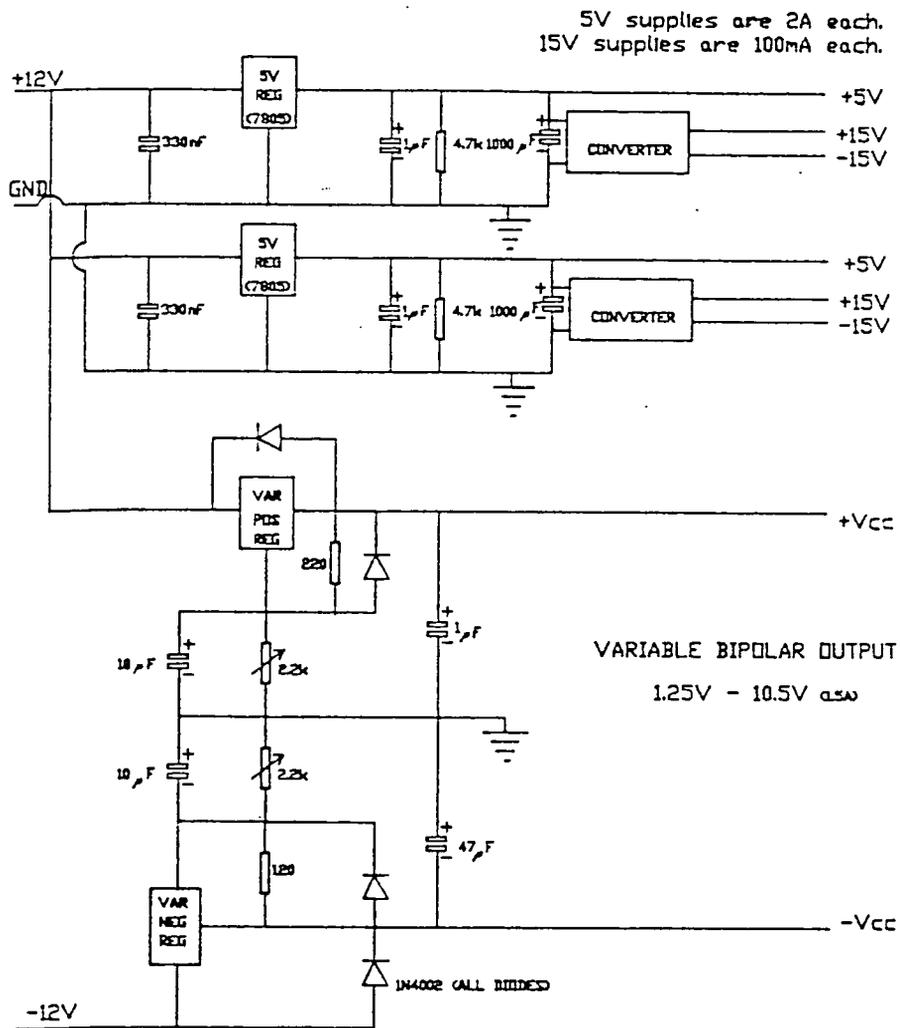


Figure 3.6 The power supply board

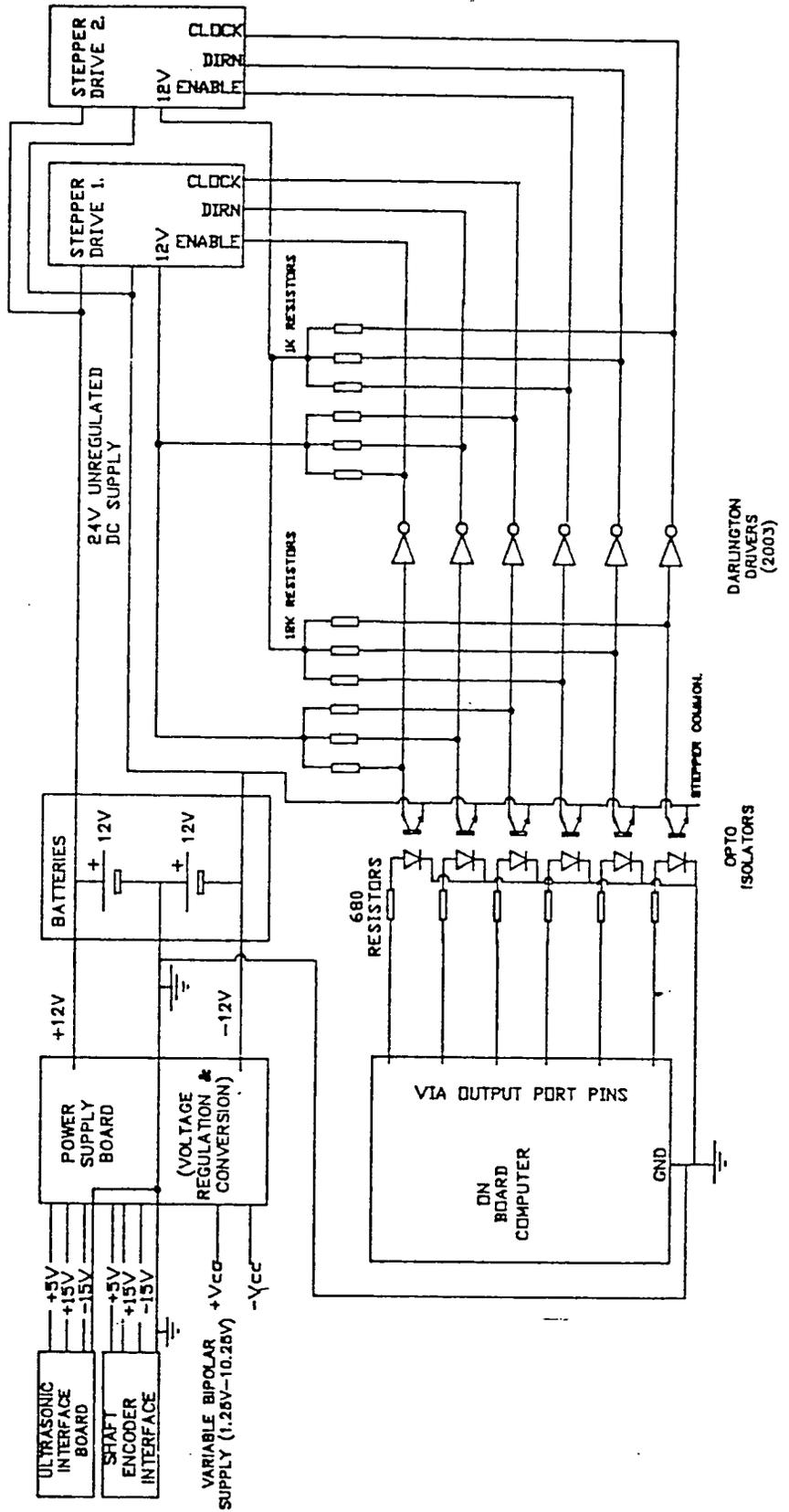


Figure 3.7 System interconnection, opto-isolation, and switching

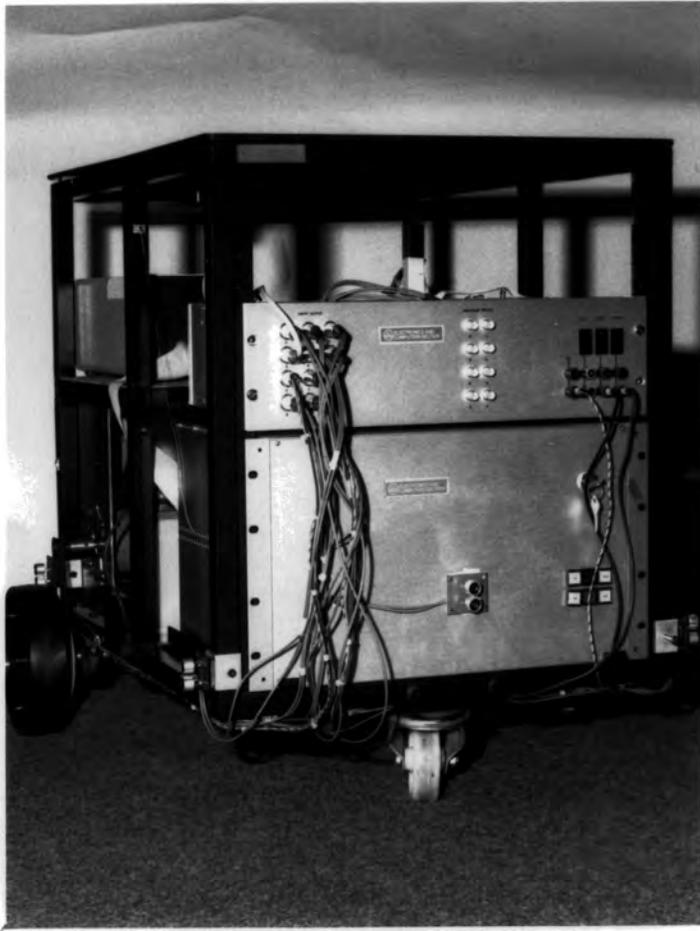


Plate 3.1: Front view of the research vehicle

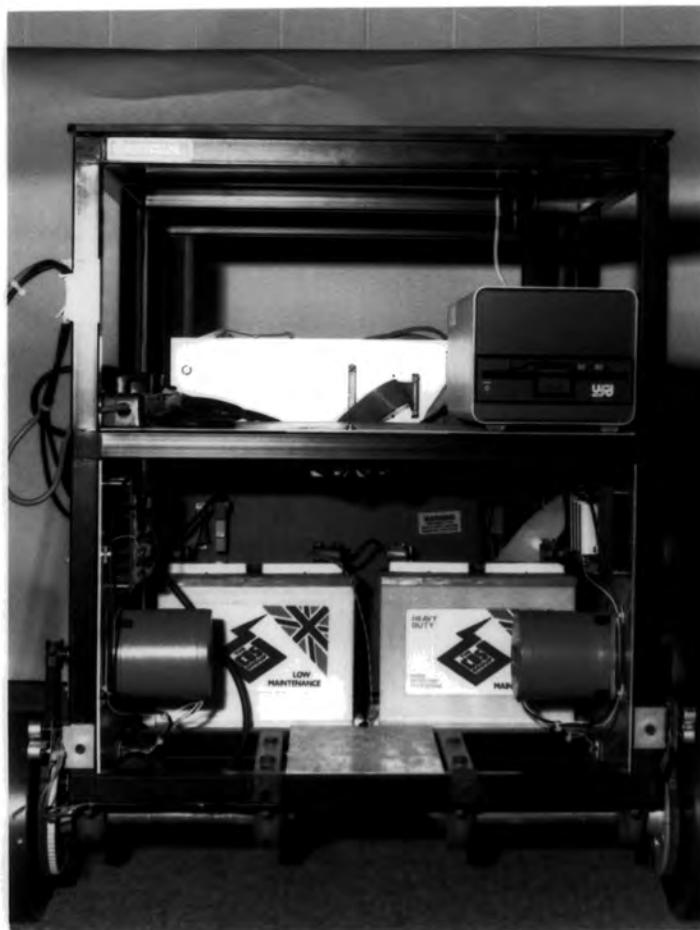


Plate 3.2: Rear view of the research vehicle

## CHAPTER 4

# SENSOR FUNCTIONALITY IN THE GUIDANCE METHOD AND SENSOR CHARACTERISATION

### 4.1 Introduction

Initially this chapter describes the functionality of the two low cost, low level sensors in a proposed guidance method. It then goes on to describe the implementation of position correction and a method of maintaining an estimate of odometric error bands. Subsequently, the manner in which the two sensor systems operate in a complementary fashion is highlighted. The remainder of the chapter is devoted to ultrasonic sensing for AGVs and mobile robots. This includes the general use of ultrasonic telemetry in autonomous vehicle research, the idiosyncrasies of ultrasonic measurement and the performance of the measurement system described in section 3.4.1. Such a characterisation of the performance of the ultrasonic measurement system is necessary in order to compensate for systematic measurement errors and assess the accuracy to which the AGV can measure its position, using the measured spread of the random error component in measurement.

### 4.2 Sensor Functionality in the Guidance Method

On the lowest level of functionality, an AGV must measure its position to provide the necessary feedback for its guidance control system. This may be effected by either

(1) External sensing when the AGV measures its position relative to a known fixed reference point. Such reference points are often called 'beacons' and may be either natural or artificially added to the environment. External sensing includes wire following ('continuous reference point'), triangulation from two reference points, or measuring both angle and distance from a single reference point.

(2) Internal sensing when the AGV measures its own movements from an initial known position. This includes inertial guidance systems, and odometry systems which monitor the vehicle's wheel movements.

In real, non-ideal environments it is unlikely that odometry can provide sufficiently accurate positional information by itself, as position errors inevitably build up with time, primarily due to wheel slip and irregularities in the floor surface although several other effects aggravate the problem (such as wear on wheels, initial measurement inaccuracies of the wheelbase and wheel diameters, and computational inaccuracies). In certain applications employing only external sensing may lead to inflexibility since the reference targets would have to be seen at all times to provide a feedback to the AGV's guidance control system.

Flexibility can be achieved by employing simultaneously both external and internal sensing. To achieve this flexibility, the AGV employed in this research can estimate its global position by monitoring the incremental rotation of each drive wheel at regular sampling intervals. In the next chapter it is described how, geometrically, these incremental wheel rotations can be used to establish the incremental change in the AGV's global position during the last sample period and hence the new updated global position. However, because

of the cumulative nature of the errors inherent in this measurement system, it is necessary to make periodic corrections of global position. Thus the AGV employs ultrasonic external sensing to sense the position of correction boards which are placed at specific, known positions within the operating environment. These correction boards are simply wooden boards approximately 25cm high and between 2m and 5m long with protruding features, which have been labelled "notches", 4cm deep and 10cm long at regular (1m) intervals along its length. (Plate 4.1 shows the research vehicle in the vicinity of one such correction board whilst plate 4.2 shows a close up of the front ultrasonic sensor after it has observed a notch.)

Essentially then, low level ultrasonic sensors are employed for reference point detection which, along with odometry, forms an estimate-correct guidance scheme.

### 4.3 Position Correction

As the AGV passes a correction board, the global position of which is known, the AGV measures its position relative to the board using two ultrasonic sensors, one at the rear of the vehicle near to the drive wheels and the other at the front of the vehicle. These transducers provide a measurement of the lateral range of the front and rear of the vehicle from the board ( $y_1, y_2$ ) and from these values a measurement of the local coordinates ( $y, \theta'$ ) can be derived such that

$$\theta' = \sin^{-1} \left[ \frac{y_2 - y_1}{s_r + s_f} \right] \quad (4.1)$$

$$y = y_1 + \frac{w}{2} \cos \theta' + s_r \sin \theta' \quad (4.2)$$

as is illustrated in figure 4.1. Measurement of the third coordinate,  $x$ , in

the local frame of the correction board (i.e. an 'along board' measurement) can only be generated when either sensor passes a notch and the resulting data is processed to yield a value for the distance travelled along the board. Since measurements in three degrees of freedom have now been made, a global position correction can be made from a transformation of the AGV's measured position  $(x, y, \theta')$  in the local frame of the correction board to the global frame  $(X, Y, \theta)$  using the matrix equation

$$\begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ \theta_0 \end{bmatrix} + \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & 0 \\ \sin\theta_0 & \cos\theta_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta' \end{bmatrix} \quad (4.3)$$

where  $(X_0, Y_0, \theta_0)$  is the position of the current command's local coordinate frame with respect to the global coordinate frame and is directly available from the software list of command specifications (the command queue array).

Equation 4.3 indicates that corrections in  $\theta$  are independent of any other coordinate and thus can be made at any time when neither sensor is opposite a notch. However, unless a correction board lies along either the X or Y axis, corrections in X and Y are not independent and, therefore, can only be made after a notch has been detected and identified.

Using the position correction mechanism described by equations 4.1 to 4.3, the error in estimated position is constrained within certain limits which depend on both the frequency and the accuracy of the corrections. Obviously these limits must be inside a value above which the AGV is deemed to be lost. In this case the positional error is so large that it picks up a correction point which it thinks is defined to be in a different position than it actually is! \* More realistically, the maximum errors allowable are dictated by the closeness of objects in the environment to the vehicle's designed path.

---

\* Notches are not individualised.

## 4.4 Errors in Odometric Position Measurement

### 4.4.1 The Nature of Odometric Measurement Errors

In order to update global position, the incremental count values from the shaft encoders on each wheel are read every 100mS. If the two values are equal, the vehicle assumes that it has travelled along a straight line, otherwise it assumes the path is an arc of constant radius. Errors in odometry occur both because the wheels may travel a different distance from that indicated by the count value, and the effective wheelbase may be different from that measured. More specifically, these errors may be from a combination of

- (1) An uneven or non-planar floor surface in the AGV's operating space.
- (2) Wheel slippage and shearing, particularly when the AGV executes tight curvatures.
- (3) Errors in the AGV's measured dimensions (wheelbase and wheel diameter). This may be due to inaccuracies in their initial measurement or wear on the wheels or flexibility in the floor surface.
- (4) Quantisation errors due to the discrete measurement of wheel rotations (i.e. the limited resolution of the shaft encoders).
- (5) Computational inaccuracies such as truncation errors in integer arithmetic.

In order to generate estimated errors in odometric position measurement it is necessary to conjecture the manner in which the discrepancy between the measured distance and true distance travelled of a drive wheel varies with distance travelled. Quantisation errors are non-cumulative and

are manifested as 'variable offsets' to the true distance. Such offsets may lie between plus and minus one count which equates to a distance error of  $\pm 6.31\text{mm}$ . Errors in wheel radius due to initial measurement, wear, or a pliable floor surface cause distance errors that are directly proportional to the true distance travelled. Errors due to unevenness in floor surface are unpredictable since the degree of unevenness is likely to vary at different points in the AGV's environment. All that may be said is that an apparently random but monotonic increase in the travelling distance compared with that over a flat surface will be induced.

Although in the list of itemised errors, errors 3, 4 and 5 are quantifiable, errors 1 and 2 are a function of both the form of the path executed by the AGV and the location of this path in the operating space. The way in which odometric measurement errors accumulate is therefore largely unpredictable. Despite this, a method of maintaining an estimate of odometric error bands is implemented in the AGV software; the reasons for doing this, and the method itself, are described in the following section.

#### 4.4.2 The Method of Maintaining Odometric Error Bands

In order to detect a *particular* notch on the correction board, positional information must be employed since notches are not individualised. Unfortunately, only the AGV's local  $x$  position calculated from odometric data is known, so it is vitally important to maintain a continually updated estimate of odometric position error bands. These error bands expand in a *path dependent* manner as the AGV travels through the environment in order to reflect the cumulative nature of position estimation errors. Effectively, such error bands are used to predict the region of odometric position estimate in which a specific notch on the correction board may be encountered. Thus the AGV is facilitated with a means of determining

- (1) Instances of a missing a specific notch.
- (2) Instances when a valid correction has not been made for a long time causing the positional error to be so large ( $\pm 0.5\text{m}$ ) that a detected notch may lie in one of two positions in the *a priori* definition of the position of the correction board. In this case, the AGV is deemed to be lost.
- (3) Instances of valid position corrections when a specific notch and correction board are found to lie within the current three dimensional error band of odometric position estimation.

Thus, in maintaining an estimation of odometric error bands, spurious data which may be interpreted as the correction board, or a notch on the correction board, can be rejected, since correction is only permitted within the current three dimensional error band. In addition, correction points should rarely be missed as they are permitted only when the nearest edge of the expanded description of position contacts the programmed position of the notch.

In order to maintain an estimate of odometric errors bands, the measured distance that a drive wheel has travelled over a sampling interval is assumed to be in error by an amount proportional to that measured distance. It is recognised here that such an assumption has only limited validity due to the nature of odometric errors as described previously. The effect of the assumption is that the estimated error in AGV heading increases in proportion to the average distance travelled by the drive wheels. This heading error then generates errors in the  $X$  and  $Y$  coordinates to a greater or smaller degree which is dependent on the current vehicle heading, and the estimated error in heading. Thus the expansion of the vehicle's odometric error bands is a function of the vehicle's path in its workspace (i.e. the expansion is

path dependent).

More specifically, if the incremental distance travelled by the left and right drive wheels is  $\Delta d_1$  and  $\Delta d_2$  respectively, then the actual incremental distances are assumed to lie within the ranges  $\beta\Delta d_1 \rightarrow \alpha\Delta d_1$  and  $\beta\Delta d_2 \rightarrow \alpha\Delta d_2$  ( $\alpha$  and  $\beta$  are constants where  $\alpha \geq 1$  and  $\beta \leq 1$ ).

It is obvious that, within such proposed error bands, there are an infinite number of combinations with which to implement the odometric calculations and the problem is to choose the correct combination of parameters that will expand the error bands of a given global coordinate by the greatest degree. This is simple for the global heading coordinate,  $\theta$ : here the combinations  $(\alpha\Delta d_1, \beta\Delta d_2)$  and  $(\beta\Delta d_1, \alpha\Delta d_2)$  generate high and low heading error bands respectively. The expansion of odometric error bands in the  $X$  and  $Y$  coordinates is more difficult: in this case one has to choose the correct value of  $\theta$  within the current heading error band in addition to the correct combination of  $\Delta d_1$  and  $\Delta d_2$ .

More specifically, a value of global heading within that coordinates current odometric error band is chosen which is

- (1) closest to  $\pi$  for the lower bound of  $X$
- (2) closest to 0 for the upper bound of  $X$
- (3) closest to  $\frac{3\pi}{2}$  for the lower bound of  $Y$
- (4) closest to  $\frac{\pi}{2}$  for the upper bound of  $Y$

This heading, along with the appropriate combination of incremental wheel distances, is fed through the odometric equations 5.4, 5.6, and 5.19 (section 5.6) in order to expand either the  $X$  or  $Y$  global error band. In practice, choosing the correct incremental wheel distances involves employing either  $\alpha\Delta d_1, \alpha\Delta d_2$  or  $\beta\Delta d_1, \beta\Delta d_2$ . The estimated error in measured curvature

over a control interval is ignored other than in updating the error bands of  $\theta$ .

Since odometric errors are not directly quantifiable, the constants  $\alpha$  and  $\beta$  were chosen by trial and error to generate error bands slightly larger than the magnitude of position corrections when the AGV was asked to move along a variety of paths in the laboratory. Suitable values were found to be  $\alpha = 1.01$  and  $\beta = 0.99$ .

Figure 4.2 shows the generation of error bands over three sampling intervals in which the vehicle is travelling along the X axis and the *measured* distances from each drive wheel is equal in each case. Note that, in this case, the rate of expansion of the error in Y is faster than the expansion in the X coordinate. When a successful position correction is made, the error bands contract by being set to the estimated error in the position correction (see figure 4.2).

#### 4.4.3 The Validity of Maintaining Odometric Error Bands

At this point it is appropriate to examine the validity of the method of maintaining odometric error bands in terms of its function within the proposed guidance method. Firstly, it should be noted that the claim of the method is modest. Given that the distance travelled by each drive wheel is within the proposed error bands, and assuming that the effective wheel base is constant, then measured global position must lie within the generated odometric error bands. Secondly, it should be noted that no assumptions have been made about the probability distribution within the error bands around  $\Delta d_1$  and  $\Delta d_2$ , and consequently no statements are made about the probability distribution above the region in the X, Y plane defined by the global odometric error bands. The upshot of this is that techniques for integrating odometric measurements with processed ultrasonic range measurements, in order to determine global

position and its associated error bands, can not be employed.

Ideally one would like to be able to reduce systematic errors in the measurement of the distance travelled by a drive wheel and approximate random errors to a normal distribution around the actual measured distances  $\Delta d_1$  and  $\Delta d_2$ . From this standpoint it would be possible, at least in theory, to generate a probability density surface above the  $X, Y$  plane which encloses a region on that plane within which the vehicle is deemed to lie. Such a representation of odometric error bands contains much more information than the simple approach used by the research AGV and may permit the implementation of forms of data fusion.

Although no experiments have been conducted to determine the form of odometric errors \* it seems unlikely that any satisfactory analytical form of modelling such errors could be extracted. This problem prevents the measurements from odometry and ultrasonic ranging from being *integrated* or *fused*; rather, the minimal assumption made about odometric errors permits the generation of tentative absolute error bands. Such error bands allow the two sensor systems to operate in a *complementary* manner.

In general ultrasonic measurements are accurate (typically to 0.5cm in 1m) and so can be used to constrain the accumulation of errors in odometric position estimation. Ultrasonic measurements, however, are not always available and are occasionally spurious. Measurements from odometry, on the other hand, are always available and their relative consistency can be used to reject spurious ultrasonic measurements, identify specific notch locations, and detect instances of valid position correction. The manner in which the two sensor systems are used in a complementary manner is discussed further in chapter 6 and chapter 8.

---

\* This would require global position measurement from some other source such as a laser triangulation system.

An experimental analysis of the performance of the designed ultrasonic sensing system is given in the following sections and the results are explained with reference to the nature of ultrasound and the method used to detect echo pulses. This is firstly required to eliminate systematic errors in position measurement relative to a correction board and secondly to quantify the remaining error in that measurement. A measurement of the error in position correction may be used to represent a measurement of the error at the start of an odometric interval and, in addition, permits the detection of instances when the confidence in correction data is so low that it should be discarded.

## 4.5 Ultrasonic Sensing for Autonomous Vehicles

### 4.5.1 General

Ultrasonic sensing is a developed technology employed in many diverse fields such as medical monitoring and diagnosis, industrial process control, and the detection of flaws in materials. The use of ultrasonic perception in the field of robotics is attributable to the low cost of ultrasonic transducers, their small size and robustness, and their simple operation. Their low cost permits a number of transducers to constitute a sonar sensing system [MORAVEC, 1985], [BAUZIL, 1981]. In the case of the vehicle employed in this research, there are five pairs of transducers; one at each corner of the vehicle and one on the front face of the vehicle. These permit the simultaneous observation of two correction boards, one at each side of an aisle, which provides the system with a degree of redundancy. In addition to these points, ultrasonic systems are cheap in terms of processing. Only a minimal amount of processing is required to provide such fundamental and useful information as the range of the nearest environmental objects in the vicinity of the vehicle.

To date ultrasonics has provided a number of mobile robot and

research AGV systems with a useful means of perception. Rotating ultrasonic sensors [KANAYAMA, 1981a, 1981b], [CROWLEY, 1985], [KOMORIYA et al, 1986] and sensors arranged in a spatial pattern around the perimeter of the vehicle [BAUZIL, 1981], [BORENSTEIN and KOREN, 1988] have been used to measure a vehicle's location relative to a wall and, for example, to provide wall following functions. However, it has been noted that in many cases, little attention has been paid to the limitations and inaccuracies of sonar sensing. Interesting research has been conducted at CMU where a ring of 32 transducers is used to build sonar maps [ELFES, 1986, 1987]. A probabilistic approach to map building effectively models the idiosyncrasies and deficiencies of sonar sensing and permits the fusion of data from multiple viewpoints to generate an increasingly detailed 'floor plan' map. An interesting point to note here is the related idea of 'beam splitting' [EVERETT, 1985] which can reduce the directional ambiguity of a target.

As well as measuring position relative to known reference points and map building, ultrasonic sensing systems are useful for the implementation of obstacle avoidance algorithms. In a scheme proposed by Kroug [KROUGH, 1986], 'generalised potential fields' are created which are dependent on the position and velocity of the vehicle with respect to each obstacle. Such a field causes the vehicle to be attracted by its goal and repulsed by obstacles. A similar scheme proposed by Khatib [KHATIB, 1985] computes a force vector in an operational space formulation of Lagrangian mechanics. Such a force vector is computed using the attractive and repulsive forces of goals and obstacles respectively. Similar systems can provide a good protection regime for autonomous vehicles, and are suitable for implementation with sonar sensing. The suitability of sonar is due to its simple processing and quickness of response which is essential since many obstacles may appear unexpectedly close at hand and some may even be mobile. In addition, the

depth accuracy of sonar is high particularly when objects are close which is important since the largest 'forces' are generated by objects which are dangerously close. (Forces may be calculated as a function of the inverse square of object range.)

The method of ultrasonic telemetry is simple. In the case of a piezoelectric transducer, voltage oscillations can be converted into the mechanical oscillations of a diaphragm. This movement disturbs the air in the vicinity of the diaphragm thus generating an ultrasonic wave. This wave then travels through its medium (air) changing in both scope and intensity as it travels. If this wave hits a solid object before it is completely attenuated, most of its energy will be reflected because the acoustic impedance of solids is much larger than that of air. (Solids act as 'acoustic mirrors'.) The manner in which the ultrasound reflects off an object is dependent on both the type of reflecting surface and its orientation relative to the direction of the wave. If some of the wave is directed back towards the transducer, and has sufficient energy to be detected on reaching the transducer, a range measurement can be made by using the local velocity of sound to scale the half-time of the 'round trip' of the ultrasonic pulse.

#### 4.5.2 The Idiosyncrasies of Ultrasonic Ranging

Although in ultrasonic ranging, range processing is very simple, a number of inherent properties of ultrasonic waves, in particular the manner in which they travel through air and reflect off solid objects, generate certain idiosyncrasies in any ultrasonic measurement system. In order to utilise ultrasound effectively, an AGV or mobile robot must recognise and, if possible, account for such idiosyncrasies in its processing algorithms.

Before the ultrasonic transducers were mounted on the research vehicle, a number of bench tests were carried out using the ranging system described

in section 3.4.1. The aim of this was to determine under what conditions the system performed well. The first obvious deficiency was that the same range reading could be obtained when a target was placed at a variety of positions relative to the sensor, as is illustrated in figure 4.3a. This implies that there is directional ambiguity when a single sensor is used for ranging. Secondly, there was an observed increase in measured range as the orientation of a reflecting surface was moved away from the normal of the incident sonar beam. As this angle increased, the signal intensity was reduced to such an extent that no measurement could be made.

The form of a reflected ultrasonic wave off a solid surface is a function of the relation between the ultrasound wavelength and the depth of irregularities in the reflecting surface. If these irregularities are small compared to this wavelength, so called spectral reflection occurs in which the angle of reflection is equal to the angle of incidence. Surface irregularities which are as large (or larger) than the ultrasound wavelength cause scattering and the resulting reflected wave is diffuse. The result of this is that the relative intensity of the echo signal (i.e. the intensity relative to that of the normal signal) drops off much more sharply with angle of incidence for a 'smooth' reflecting surface than for a 'rough' one [CLEMENCE, 1983].

It should be noted here that the research vehicle's transducers operate at 40kHz which means at room temperature, the speed of sound is such that the wavelength is approximately 8.4mm. This means that most surfaces in the laboratory are spectral in nature, including the laboratory walls and the correction boards themselves.

The result of this signal attenuation is that objects become sonically invisible when the orientation of a target relative to the normal of the transducers exceeds a certain range dependent threshold. The angle over which objects are visible can be thought of as an effective beam width and,

for a given range, this is dependent on

- (1) The directional pattern of the transmitted pulse.
- (2) The directional sensitivity of the receiver and its effective sensitivity due to signal thresholding.
- (3) The 'roughness' of the reflecting surface.

This effective beam width angle contracts with increasing range in a manner dependent on the power transmitted and the degree of attenuation with range (which is a function of wave frequency and the absorption constant of the medium.) The implication is that objects which are near to the limit of the sensors range have to be both perpendicular and centered on a normal projected from the face of the transducers (see figure 4.3b) in order to be detected.

If a target is not directly facing a transducer, how is it known what range is measured if the ultrasonic beam strikes a large area of that target? If the reflecting surface is perfectly spectral then, if a measurement can be made, it must always be normal to that surface since waves striking the target at any other angle of incidence are directed away from the transducer. If the reflecting surface is diffuse then the range measurement is more ambiguous since scattering permits energy to return from a much larger part of the area of the target struck by the ultrasonic beam. In this case the measurement is likely to be the path closest to the normal (since that is the shortest) at which sufficient energy is returned to trigger the detection threshold.

The choice of correction board is now clear. Correction boards can be thought of as an artificiality added to the vehicle's environment which is tailored to, and designed to enhance, a low level sensor's performance. The boards themselves were seen to be smooth when observed using 40kHz

ultrasound and thus functioned as spectral reflectors. This was verified by observing the form of the reflected pulse on an oscilloscope as the transducers were rotated relative to the face of a correction board. Although the intensity of the signal varied, the position of the peak of the pulse on the time scale remained constant with respect to the transmission pulse. Such spectral reflections ensured that a normal relative to the board was measured irrespective of the sensor's orientation. The large beam width of the sensors, although a function of range, generally allowed corrections to be made at up to 20 degrees relative to the board.

Now that the idiosyncrasies of ultrasonic telemetry have been described, the following section examines the specific performance of the designed ultrasonic measurement system.

## **4.6 Performance of the Designed Ultrasonic Ranging System**

### **4.6.1 Range of the Measurement System**

Distance measurements are made by measuring the time delay between the start of the transmission of a burst of ultrasound and the detection of the echo pulse. It can be recalled from section 3.4.1 that to detect a echo pulse, a comparator level of 200mV is employed. This level was chosen to be high enough to prevent noise, both that coupled onto the receive signal wire from the transmit signal wire and other extraneous noise, from being interpreted as echo pulses. In general, the distance an ultrasonic pulse can travel through air is dependent on the initial power transmitted, the degree to which the transmitted cone expands (for example there may be acoustic focusing) and the degree to which power absorption by air attenuates the signal. This is proportional to the square of the transmission frequency. Due to attenuation and the expansion of the transmitted wave, the energy in the signal drops

rapidly with range. The maximum range for the designed measurement system is the distance at which the echo signal is attenuated to 2mV. (Since the amplification stage has gain 100.) This range was found to be about 2m, although the range at which measurements could be made reliably was around 1.5m. The factor governing the range of the device is the point at which the echo signal level falls into the noise level. It was found that increasing amplification in the system did not increase the effective range because of the difficulty in preventing noise increasing in almost the same proportion, either by amplification or the addition of extra circuitry, as the signal itself. Although attenuation of ultrasound is increased in proportion to the square of frequency, lowering transmitted frequency is ineffectual because the limited bandwidth of the receiver causes its sensitivity to be greatly reduced. In addition, lowering the transmission frequency would be detrimental to the accuracy of the system. It was demonstrated that signal strength could be increased by focusing the transmitted beam with a paper cone thus improving range. The problem with focusing is that only small angles relative to the normal of the reflecting surface can be tolerated before signal strength falls off rapidly. The only way to improve the range of the device is to employ larger transducers which can accept more voltage, generate higher power bursts of ultrasound, and thus improve signal to noise ratio.

At the time the AGV research system was being developed, it was thought that the 1.5m range was sufficient to conduct the proposed research. Indeed it was in the initial stages of testing when the vehicle was asked to follow a correction board at a range of 1m whilst continually making corrections in position. However, when the system software was fully developed it was found that, unless a large number of correction boards could be distributed around the laboratory, the range limitation of the sensors restricted the

flexibility of the system. This was because routes had to be chosen carefully in order that they pass correction boards closely (i.e. within about 1.5m) allowing corrections to be made at a sufficiently high frequency to constrain the error in estimated position. To increase the flexibility of the current research system, the AGV should be fitted with higher range sensors. One possibility would be to modify the triggering circuitry of a commercial Polaroid sensing system to increase its pulsing rate. (Ideally it would be software triggerable.) However, it may be found that the pulsing rate is limited by the degree of damping in the transducer.

In order to investigate the effect of the width of transmitted pulse on the form of the echo pulse, the transducers were set at a variety of ranges from a reflecting object and the transmit pulse width was increased from 0 to 2mS. In each case the amplitude of the echo pulse increased to a range dependent maximum which occurred at a pulse width of approximately 0.8mS. Any further increase in pulse width simply caused the echo pulse to grow in width with no further increase in amplitude.

This building up and flattening off of the echo pulse envelope occurs because the damping of the receiver diaphragm oscillations increases with the magnitude of the oscillations. Oscillations in the receiver can only build up if the excitation caused by the reflected sound is greater than this damping force. After a given length of time, the oscillations become so large that the two forces are equal and no further increase in amplitude can take place. The receiver will oscillate at the same amplitude for the remainder of the transmitted cycles and then fall off according to its own degree of damping.

It was noted that the minimum pulse width at which the echo pulse reached its maximum amplitude did not vary much with range. (Slightly longer transmissions were required for larger ranges.) For this reason the transmit pulse width is fixed at 1mS allowing the maximum amplitude to be

achieved at all distances in the range.

#### 4.6.2 Accuracy of Ultrasonic Ranging

The accuracy of ultrasonic measurements is dependent on a number of factors

- (1) Quantisation errors in timing.
- (2) Time delays in software.
- (3) Speed of sound measurement.
- (4) Delays in the detection of the echo pulse.

The clock used to time ultrasonic pulses has a frequency of 0.8MHz. The maximum quantisation error in timing is two cycles of the clock, one for the start pulse and one for the stop pulse. For a speed of sound of 337m/s ( $16^{\circ}C$ ) this equates to an error of 0.42mm. This is small compared with a typical variation of  $\pm 0.5$ cm over 1m.

Time delays in software may be generated because a certain number of instructions need to be executed in interrupt software between a start or stop signal being generated on the ultrasonic interface board and the appropriate VIA timer being started or read. Since the microprocessors clock runs at 8MHz, these delays are small and, additionally, they can be eliminated by ensuring that the same number of clock cycles are executed in the 'start' branch of the interrupt software as are executed in the 'stop' branch. Alternatively, the difference between the two branches can be calculated and compensated for.

The speed of sound in air varies with temperature according to the function

$$V_s = 320.469\sqrt{1.05569 + 0.00313T_a} \quad (4.4)$$

where  $V_s$  is in m/s and  $T_a$  is in  $^{\circ}C$ . Systems employing temperature compensation have been designed [CANALI et al, 1982], [EVERETT, 1985] which employ a thermistor to generate a temperature dependent voltage which in turn can be used to supply a correction factor for the range measurement. An alternative method of compensation uses two transducer sets, one to measure the distance from a known fixed object, and one to measure the target range itself. The fixed object range can then be used to calibrate the target range [CHANDE, 1984]. This method has the advantage of providing ranges which are independent of other atmospheric conditions other than temperature such as humidity and pressure. The disadvantage, of course, is that the parts and complexity of the measurement system is increased.

Although the research AGV has a number of A to D converters which would, in theory, allow it to read a thermistor, the added complexity of both of the above systems was avoided. In the case of the research vehicle, the velocity of sound has a default value calculated for  $20^{\circ}C$  which can be altered if necessary at the start of a run using the above equation. Temperatures can be read in the laboratory to an accuracy of about  $0.5^{\circ}C$ . An error of  $0.5^{\circ}C$  at a nominal temperature of  $16^{\circ}C$  produces a range error of 0.09% or around 1mm over a metre.

The greatest measurement errors by far are generated from variations in the point within the echo pulse envelope at which a 'stop' pulse is generated. Software has been written that collects, averages, and measures the spread of a number of ultrasonic range measurements every  $n$  seconds. If  $n$  is specified to be 1, then the mean and standard deviation of one hundred new observations are displayed every second. In addition, the running mean and standard deviation are displayed from the time that program was started. In

order to generate a sensor characterisation, this software was used a number of times with a transducer set placed at specific and carefully measured ranges and orientations relative to a correction board.

#### 4.6.3 A Characterisation of the Ultrasonic Ranging System

By employing the software described above in individual tests, each lasting one minute, a characterisation of the ultrasonic sensing system was obtained. The results generated are given in table 4.1, which shows mean measured distance with respect to true distance and orientation relative to a correction board, and table 4.2, which shows measured standard deviation at the same distances and orientations. The results presented derive from digitally filtered data where the digital filter is of the form

$$y_i = (1 - \alpha)y_{i-1} + \alpha\gamma_i \quad (4.5)$$

The reasons for using such a filter, and the inherent problems with such a filter, are explained fully in section 8.2. It is sufficient to note here that  $\alpha$  is chosen to be 0.2 and the use of the filter causes a reduction in the measured standard deviation by a factor of four as compared with the measured spread of the raw data.

Figure 4.4 is a plot of the results given in table 4.1, thus each point is generated from a population of 6000 ultrasonic measurements. These results characterise the accuracy of the measurement system; for example, at 60cm and 20 degrees, the measured range has an offset of 7.5cm and a standard deviation of 0.1cm. This means that 95.5% of the readings lie within a 4mm band. Chapter 8 describes how the measured offsets are treated as systematic errors, which can be largely eliminated, and how the measured component of random error (i.e. the spread of results) is used to generate an estimate of the error in position correction. Both systematic and random errors can be

explained if the method of pulse detection is considered. This is the concern of the following two sections.

#### 4.6.4 Offset Errors in Ultrasonic Ranging

If a pulse of sound is fired at an object, and the intensity of the echo signal reaching the receiver is high, as in figure 4.5b, the echo is detected at the start of the rise of the signal. This corresponds to a point near to the start of the transmitted signal, which is the point at which the timer was started. With a very weak signal, such as that in figure 4.5c, detection can only occur at or near to the peak which corresponds to the back of the transmitted pulse. Thus offset errors in timing are due to the detection point moving through the length of the transmitted pulse as the intensity of the pulse reaching the receiver decreases. Since the length of transmitted pulse is 1ms, offsets of approximately 16.85cm ( $V_s=337\text{m/s}$ ) will be generated just before the object becomes invisible.

Since echo intensity is a function of range and orientation relative to a correction board, offset errors are also a function of range and orientation. Attenuation of signal intensity with range has been explained earlier in the chapter. As far as orientation is concerned, it has already been noted that ultrasonic reflections off a correction board are spectral in nature. This means that, in the characterisation of the measurement system, the ostensible increase in range at larger angles can not be because larger distances are actually being measured, since any part of the beam that does not reflect normally from the target will be directed away from the receiver. These variations in measured range for different angles are again attributable to variations in the amplitude of the received echo. As the angle of the transducers relative to the correction board is increased, the normal reflection moves outwards towards the edge of the transmitted cone where signal strength is weaker. In

addition, the receiver itself becomes less sensitive as its angle relative to the normal echo pulse increases.

The characterisation in figure 4.4 allows these offset errors to be considered as systematic errors which can be compensated for if the distance and angle of a particular sensor is known relative to a correction board. This compensation process is obviously iterative, since the offset errors are a function of perpendicular distance and orientation relative to the board, and these offsets are contained in the initial raw measurements with which to determine that perpendicular distance and orientation. More detail on this is given in chapter 8.

#### 4.6.5 Random Errors in Ultrasonic Ranging

Noise either on the echo signal or the threshold level itself, or changes in the form or amplitude of the echo signal, inevitably causes changes in the detection point of the echo signal. It was noted that fluctuations in the height of the echo pulse occurred even when the transducers were at a fixed distance from the target. These fluctuations seemed regular (see figure 4.6) and may have been a secondary response due to pulsing ultrasound bursts at 100Hz. If the degree of damping in the receiver is such that it has insufficient time to stop vibrating after being excited by its previous echo pulse, then the effective sensitivity of the receiver is likely to be changed. The regular envelope of the amplitude of received echos would then be a function of both the excitation frequency (100Hz) and the degree of damping of the transducer. (It was found that the shape of this amplitude envelope did change with pulsing frequency.) In addition to this, local variations in the properties of air are likely to cause small fluctuations in the form of the echo pulse.

The observed random variations in ultrasonic range measurements can partly be attributed to the detailed form of the echo pulse. Figure 4.7

shows the jagged form of the echo signal which is due to the charging and discharging of the filter capacitor (27pF) on the ultrasonic interface board (figure 3.4). Because of the discharge part of the curve, a small fluctuation in height of the echo signal can cause the detection point to move by one cycle which corresponds to a measurement change of approximately 4.2mm (at  $V_s=337\text{m/s}$ ).

From the characterisation in figure 4.4 and table 4.2, it can be noted that the spread of results increases slightly with offset. This is because the slope of the echo pulse decreases as its peak is approached allowing fluctuations in height to move the detection point by a greater amount.

In general, the standard deviation of random errors was found to be 0.2cm or less at ranges up to 1m and angles up to 20 degrees. This means that, for measurements up to these limits, 95.5% of the static ultrasonic observations will lie within a band of 8mm. If this random error is added to a systematic error of 1mm, which derives from the inaccuracy in temperature measurement, the suggestion is that a filtered ultrasonic measurement is accurate to within about  $\pm 0.5\text{cm}$  of the true correction board range. In practice, it was found that range measurement is accurate to  $\pm 0.5\text{cm}$  or better for ranges below 60cm and angles within 20 degrees, whilst for a range of 1m and angles within 10 degrees, accuracy is reduced to about  $\pm 1\text{cm}$ . This is because there must be some error in the calculation of offset error, which is implemented by means of a two dimensional interpolation, as is described in chapter 8.

Some of the work on data processing presented in chapter 8 assumes that the form of the errors in ultrasonic range measurement is Gaussian. Since ultrasonic measurement errors are generated from many sources (noise on the threshold level, noise on the echo signal, amplitude variations in the echo signal, local variations in the properties of air, and so on), one might expect

random errors in measurement to be normally distributed due to the central limit theorem. This states that if an error is a sum of errors from several sources, it will tend to be normal no matter what the probability distribution of the separate errors may be [DRAPER and SMITH, 1981]. Figure 4.8 is a relative frequency plot of 27100 unfiltered ultrasonic measurements along with a Gaussian envelope of the same variance as the collected data. This figure shows that the assumption of Gaussianity is not an unreasonable one to make.

#### 4.6.6 A New Method of Ultrasonic Ranging

Rather than attempting to detect the front edge of an echo pulse, which is embedded in noise, a method was implemented which aimed to determine the position of the echo peak, which corresponds to the back edge of the transmitted pulse. This method assumed that the form of the echo is symmetrical, which implies that the echo peak is at the centre of the stop pulse, as shown in figure 4.9. Thus, knowing the length of both start and stop pulses, measured range could be deduced from the time between the back edge of the transmitted pulse and the peak of the echo pulse (see figure 4.9). This method of detection has a number of interesting properties as compared with the conventional method of measuring the front edge of the echo. Firstly, the assumption of echo symmetry becomes more valid as the intensity of the echo decreases. Thus the system has the strange property of reducing inaccuracies due to timing as object range and orientation increase. To illustrate this, consider the observation of an object at the limit of the measurement system's range. In this case the stop pulse is extremely thin, which does not leave much scope for inaccuracies in determining the instance at which the echo peak occurs. Secondly, the observed increase in measured range with object orientation in the conventional detection method does not

occur when observing the instant at which the echo peak occurs. Finally, more information is extracted from the echo in that the width of the stop pulse gives an indication of signal intensity. Thus, for a given type of reflecting surface, stop pulse width may be calibrated to give an indication of the orientation of the reflecting surface.

Prior to testing this new ranging method, it seemed likely that it would be an improvement on the conventional method, not least because it does not require a detailed characterisation of offset errors. However, it was found that the symmetry of the echo pulse did not hold true, largely because of fluctuations in the length of the transmitted pulse. \* For this reason, the new method of range measurement was abandoned and a conventional method of measurement was incorporated into the AGV software. It should be noted, however, that the method has not been discarded as useless. If improvements could be made to the transmit circuitry of the ranging system, a reappraisal of the detection method may find that it out-performs the conventional method of detection. It may be found that it is best to detect the front of the echo for strong reflected signals and the peak of the echo for weaker signals.

#### 4.7 Discussion

Problems were encountered with the ultrasonic ranging system because 'off the shelf' components were used. In particular, standard 741 operational amplifiers were cascaded to provide the necessary gain in the receiver stage. This had a detrimental effect on the range of the system since the detection threshold had to be set above any noise introduced by such an arrangement. An improvement on the current system would be to replace those amplifiers

---

\* At this stage of testing, transmit pulses were generated by comparing a triangular wave with a non-clamped reference level.

with a low-noise type which has sufficient bandwidth to allow a single stage to be used.

Figure 3.3 (chapter 3) shows that the source of transmission pulses is switchable between a fixed pulsing rate (which is generated by a 555 timer) to a pin on the interface board from which a transmission pulse can enter from one of the on-board computers interface chips. This allows the option of 'keying' transmission pulses each time an echo pulse is received. This permits an ultrasonic transducer to make range measurements at a maximum rate which increases as object range decreases. The fastest rates are thus generated by the closest objects which is sensible since in these situations more critical or urgent control is required.

Work has been conducted in Durham [JOHNSON, 1988] which indicates that the commercially available Polaroid ranging system can be modified from its nominal pulsing rate of 10Hz so that it is software triggerable. A further advantage of updating the research AGV with such transducers concerns its automatic method of filtering spurious multiple reflections and providing (amplified) echo pulses of a more constant amplitude. To do this a time varying gain circuit is employed in the 'receive' electronics [JAFFE, 1985], [POLAROID CORPORATION, 1982]. If multiple reflections occur from a single transmitted pulse, they can only be rejected if they reach the receiver before the transmission of the next pulse which is impossible if transmission pulses are keyed on the reception of the first pulse. With a time varying gain circuit, such weak reflections are ignored since the gain will not have reached a sufficiently high value for weak secondary reflections to cross the detection threshold.

The suitability of sonar sensing for use in robotics has been extensively commented upon in the research literature and there is some measure of conflicting opinion. In some cases, the idiosyncrasies of ultrasonic measure-

ment, particularly spectral reflection, are seen as deficiencies and the role of sonar is seen as little more than a 'soft bumper'. In the case of the research presented here, spectral reflections are used to advantage, in that there is no ambiguity in the direction from which an echo pulse reaches the receiver. It is argued here that sonar will continue to feature strongly in autonomous vehicle research. The reasons for this are twofold. Firstly, current ideas about multi-sensor systems advocate the use of a set of sensors which can cooperate with each other and which complement one another. Thus the limitations of any one sensor type do not limit the overall functionality or autonomy of a vehicle. Secondly, recent research has been concerned with determining ways in which more information can be extracted from a sonar echo than is in the simple time of flight measurements employed by most current systems. A recent paper has noted the efficiency with which bats can track and locate moths; this has prompted a simulation approach to understanding and improving robotic sonar echolocation [McKERROW, 1989].

It has been noted throughout this section that most measurement errors derive from the uncertainty in the position at which an echo pulse crosses the detection threshold. Recent work has been concerned with digitising the whole of the echo signal [FRANCIS, 1987]. This enables techniques such as matched filtering to be used in order to determine accurately object range. In addition, the overall range is improved since echo signals can be picked out from relatively low signal to noise ratios [STEER and ATHERTON, 1988]. As well as improving range and accuracy, digitising the echo signal enables inferences to be made about the nature of the object. For example, two signals with the same time of flight but different intensities suggests that either one surface is angled more steeply relative to the normal, or the two reflecting surface are made of different material.

## 4.8 Summary and Conclusion

This chapter has shown how a correction board, observed simultaneously by two ultrasonic transducers mounted on a research AGV, can be used to measure local (and therefore global) position.

The manner in which odometric measurements errors build up has been discussed with particular reference to the way in which errors in heading generate errors in the X and Y coordinates. The importance of estimating the error in odometric position measurement has been shown, and a simple method of making such an estimate has been described.

In this chapter the complementary nature of ultrasonic ranging and odometry has become apparent in that the *accuracy* of ultrasonic measurement can be used to constrain the accumulation of odometric measurement errors whilst the *consistency* of odometry can be used to reject the spurious ultrasonic measurements and determine the *a priori* location of non individualised features (notches) on a correction board.

It has been shown that the measurement characteristics of the ultrasonic sensing system depend both on the inherent properties of ultrasonic wave propagation and reflection, and on the design of the ultrasonic sensor interface. Such effects have been explained and a detailed observation of the sensing system's performance has been made in order to provide the AGV with an *a priori* sensor characterisation. This characterisation allows systematic sensing errors to be reduced, thus providing a more accurate measurement of local position. In addition it permits an estimate to be made of how accurate that position measurement is.

This work has brought to light an important observation concerning low level sensing: those properties of a given sensor type, which are regarded as deficiencies when that sensor operates in a given context, can be used to

advantage when operating in a different context. In the case of ultrasound, a large beam width is often regarded as a deficiency because it engenders directional ambiguity and can cause recesses to be invisible to the sensor. Spectral reflections are often quoted as a further deficiency in that they cause reflecting surfaces to become invisible when they are orientated outside the effective beam width of that sensor for a given range. However, when a correction board is observed by two sensors simultaneously, the two properties of spectral reflection and wide beam work together to advantage. Firstly, the wide beam allows correction board observations to be made over a large range of vehicle orientation. Secondly spectral reflections ensure that the measurement of vehicle orientation is accurate since, in order for any reading to be obtained, measured ranges must be taken normal to the correction board.

Now that the AGV's sensing methods have been discussed in detail, the following chapter describes how these fit into a 'command structure'. In addition, this chapter describes how a control strategy was formulated and describes the necessary geometrical calculations required for both sensing and control.

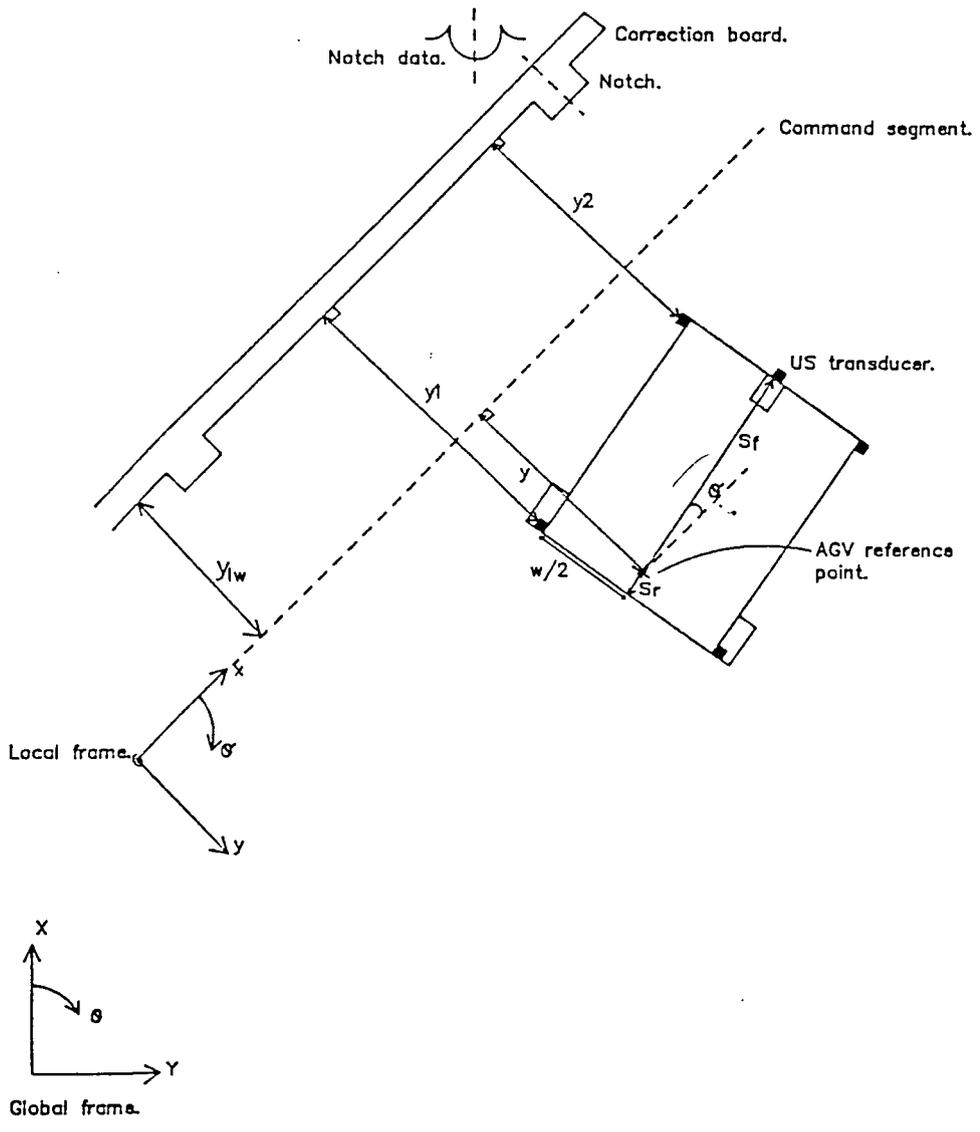


Figure 4.1 The position correction method

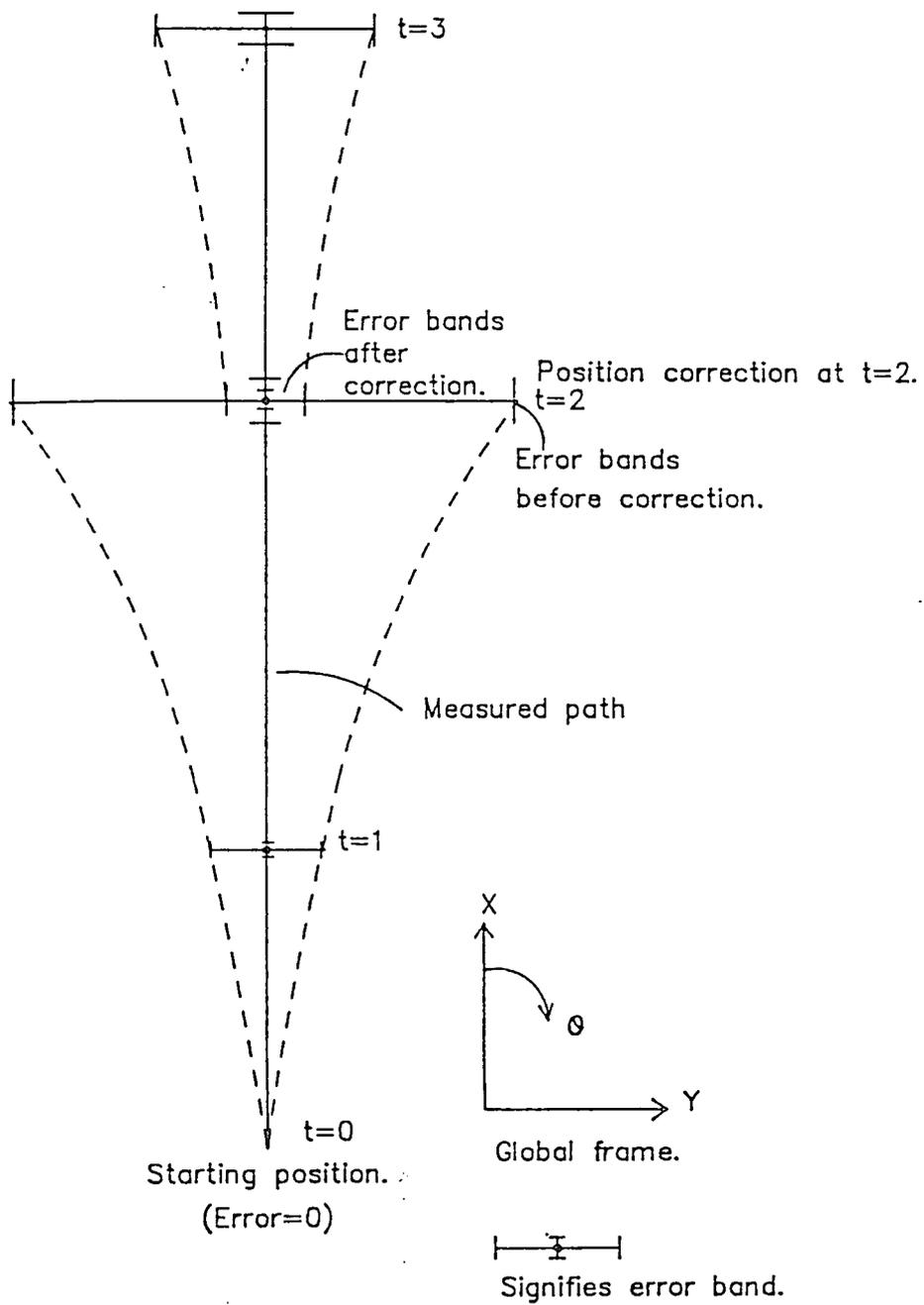


Figure 4.2 The expansion of odometric error bands

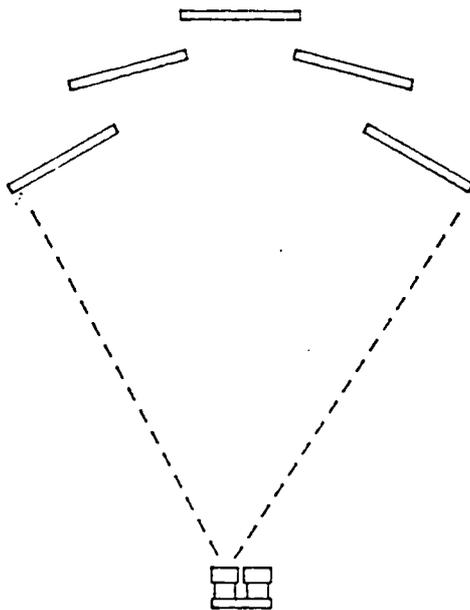


Figure 4.3a Directional ambiguity in ranging

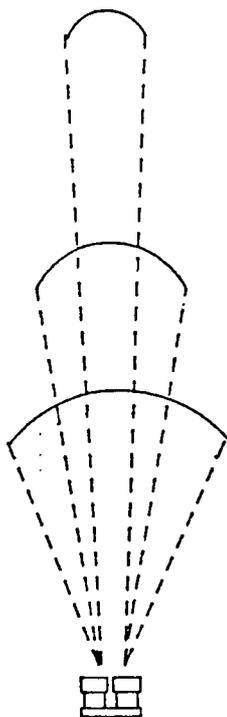


Figure 4.3b Effective beamwidth narrows with range

Figure 4.3 Idiosyncrasies of ultrasonic ranging

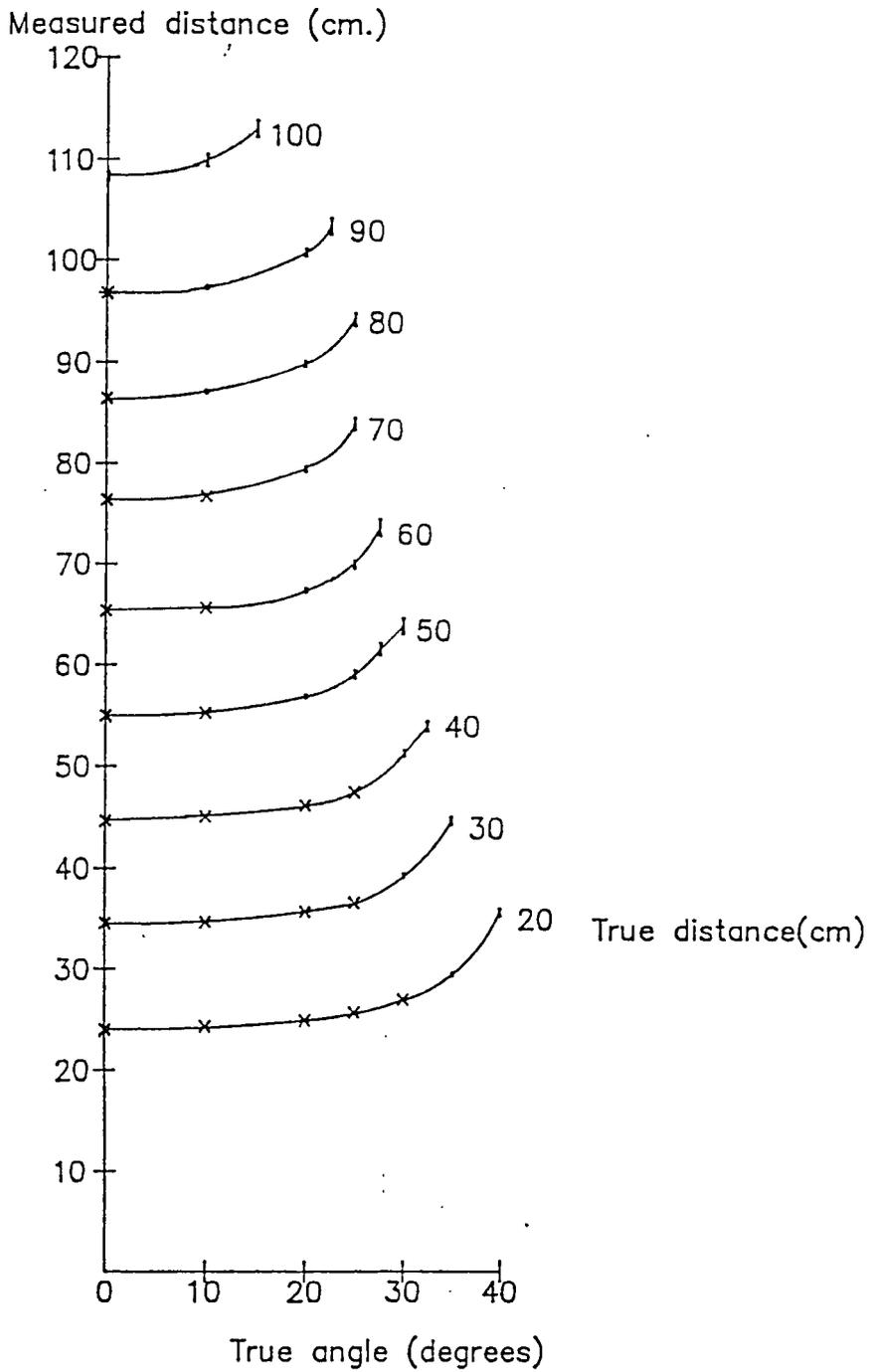


Figure 4.4 A characterisation of systematic measurement errors

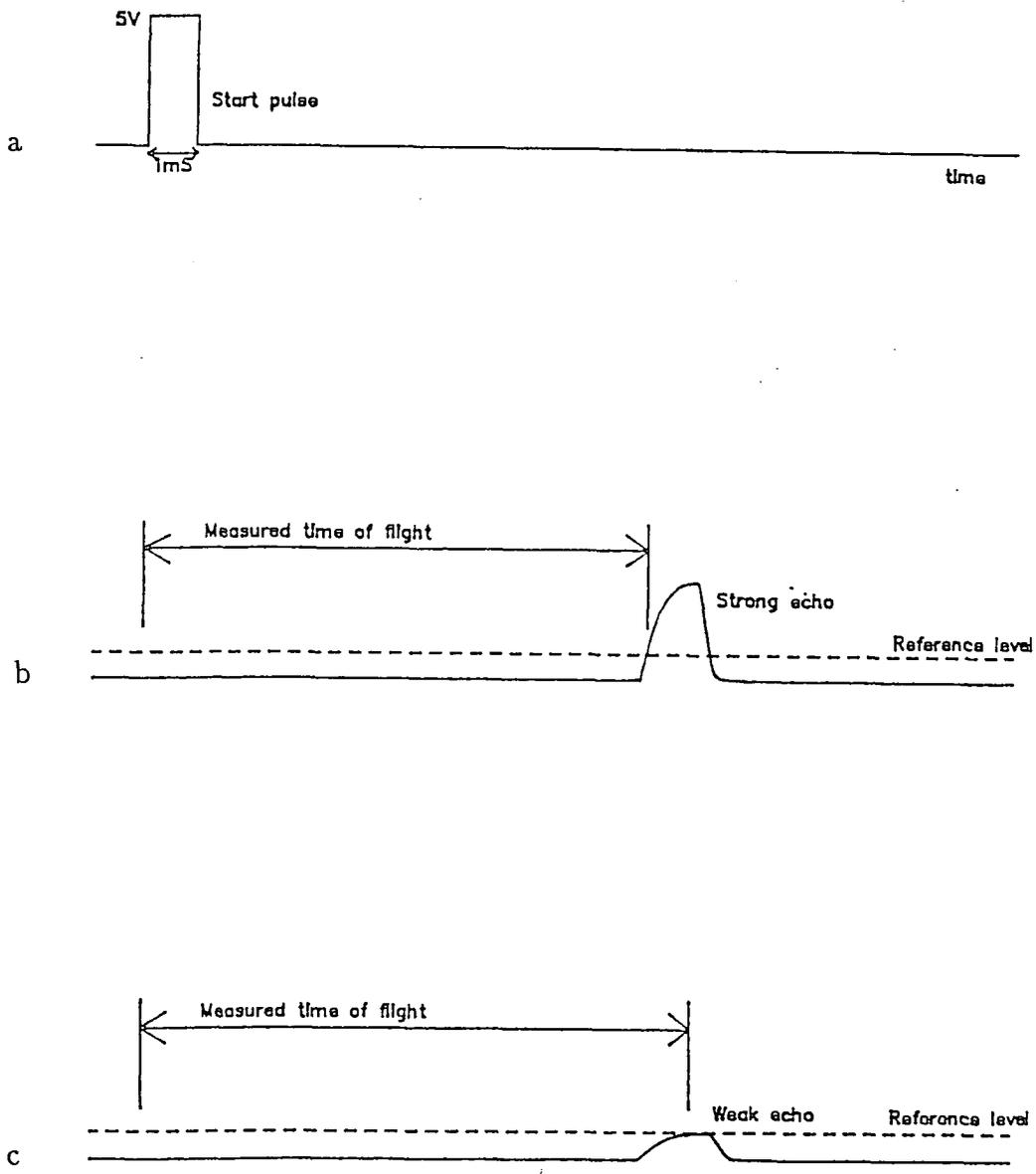
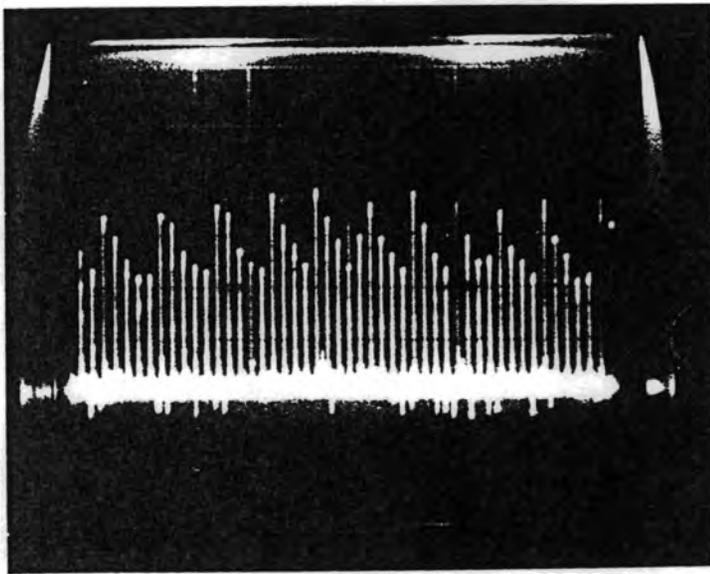


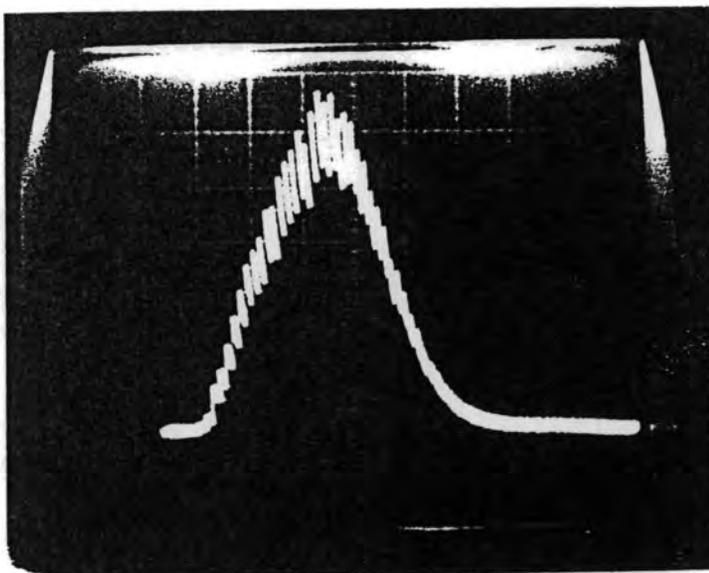
Figure 4.5 The delay in small amplitude pulse detection



Time base: 50mS/div

Gain: 0.5V/div

Figure 4.6 Fluctuations in the amplitude of the return echo



Time base: 0.2mS/div

Gain: 0.2V/div

Figure 4.7 The detailed form of an ultrasonic echo pulse

Range: 50cm

Transmit pulse width: 0.6mS

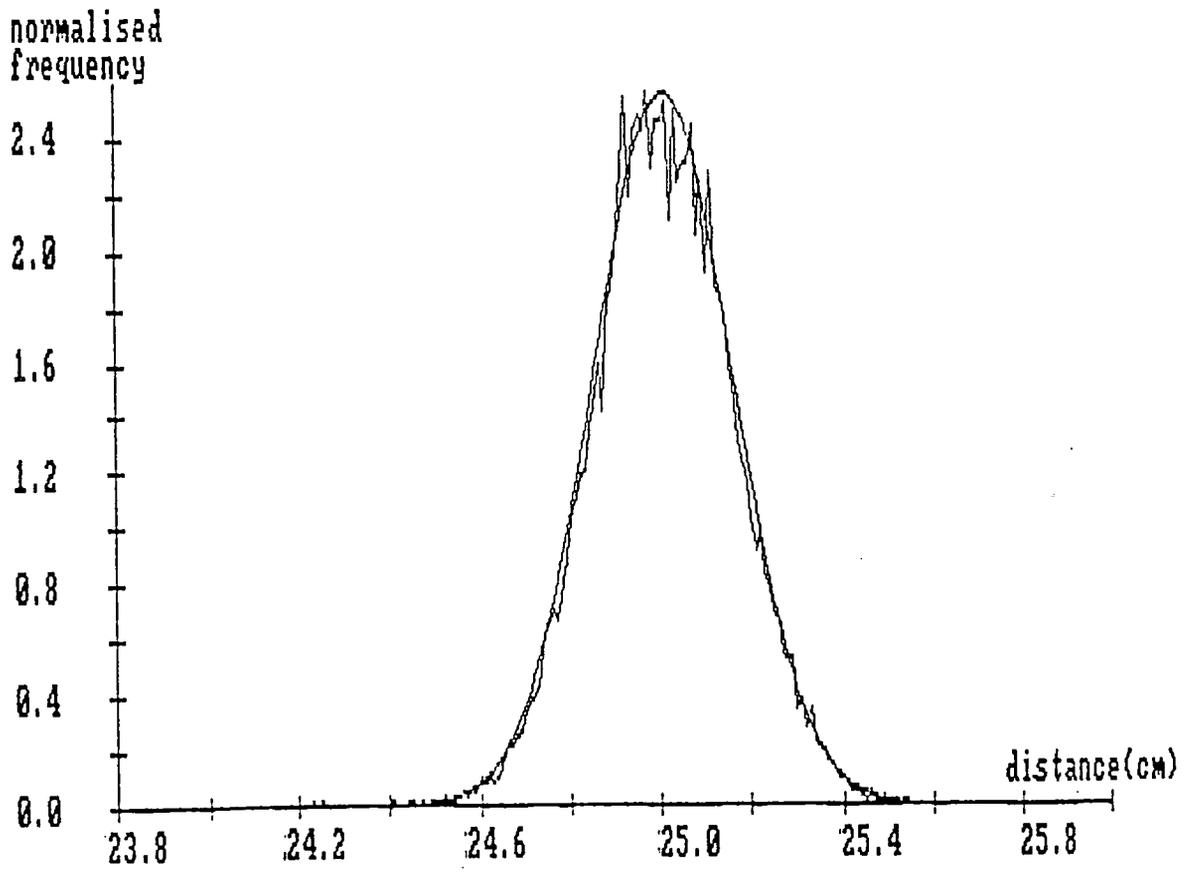


Figure 4.8 The relative frequency of ultrasonic range measurements

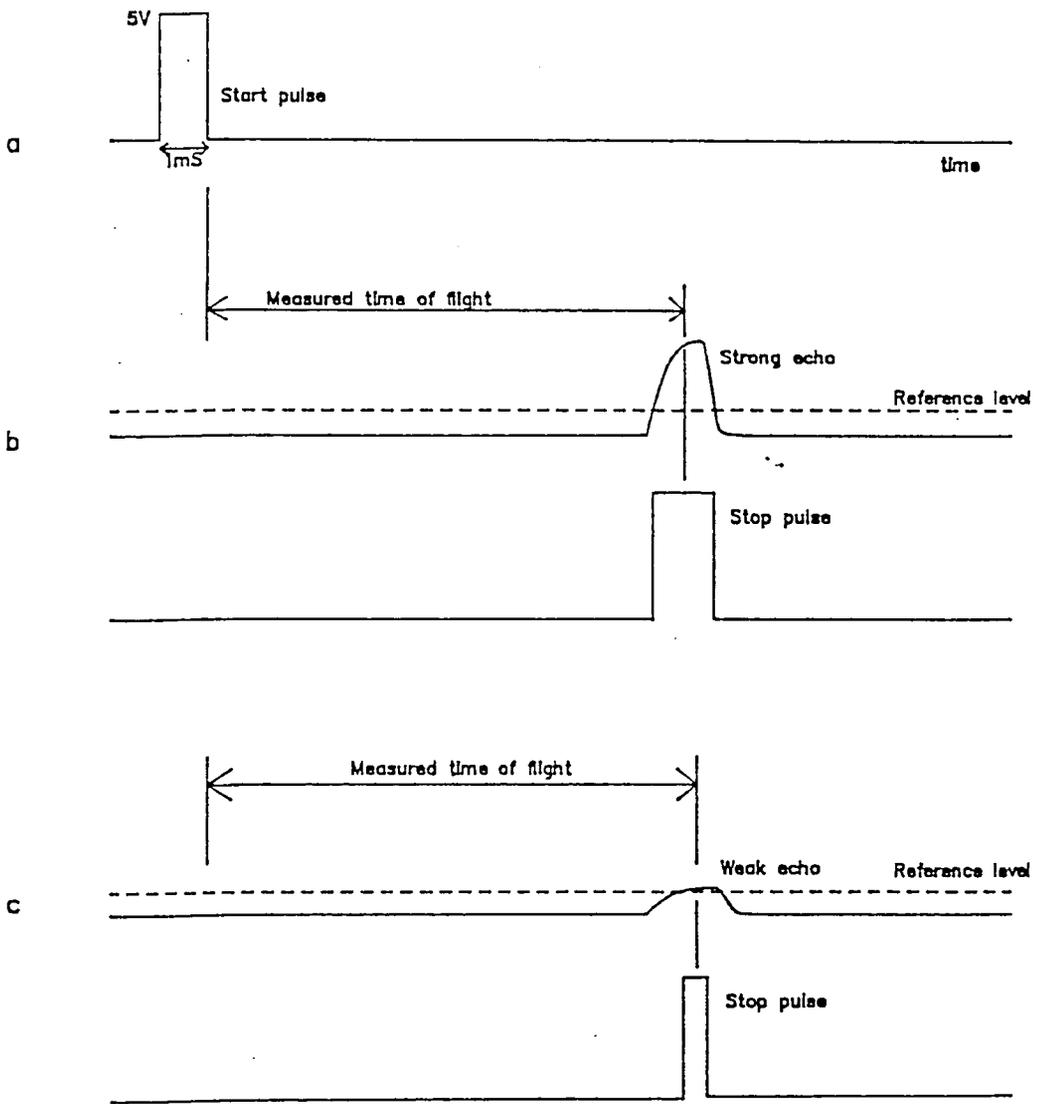


Figure 4.9 Detection of the peak of the return echo

y(cm)	$\theta$ (degrees)											
	0	10	15	20	22.5	25	27.5	30	32.5	35	37.5	40
20	24.0	24.2	-	24.9	-	25.7	-	27.0	-	29.5	-	35.8
30	34.6	34.7	-	35.8	-	36.5	-	39.2	-	43.6	-	-
40	44.8	45.1	-	46.2	-	47.5	-	51.2	54.0	-	-	-
50	55.0	55.1	-	57.0	-	59.1	61.4	63.8	-	-	-	-
60	65.3	65.7	-	67.5	-	69.7	74.0	-	-	-	-	-
70	76.3	76.9	-	79.3	-	84.0	-	-	-	-	-	-
80	86.5	87.2	-	89.9	-	94.3	-	-	-	-	-	-
90	96.9	97.3	-	100.7	113.4	-	-	-	-	-	-	-
100	108.5	109.9	113.0	-	-	-	-	-	-	-	-	-

Table 4.1 Mean of filtered ultrasonic distance measurements

y(cm)	$\theta$ (degrees)											
	0	10	15	20	22.5	25	27.5	30	32.5	35	37.5	40
20	0.023	0.035	-	0.07	-	0.075	-	0.09	-	0.1	-	0.21
30	0.05	0.06	-	0.07	-	0.076	-	0.1	-	0.29	-	-
40	0.06	0.061	-	0.073	-	0.08	-	0.13	0.25	-	-	-
50	0.06	0.065	-	0.08	-	0.11	0.21	0.22	-	-	-	-
60	0.06	0.07	-	0.1	-	0.13	0.22	-	-	-	-	-
70	0.09	0.12	-	0.14	-	0.28	-	-	-	-	-	-
80	0.1	0.12	-	0.15	-	0.22	-	-	-	-	-	-
90	0.1	0.14	-	0.17	0.26	-	-	-	-	-	-	-
100	0.17	0.22	0.32	-	-	-	-	-	-	-	-	-

Table 4.2 Standard deviation of filtered ultrasonic distance measurements



Plate 4.1 The research vehicle close to a correction board

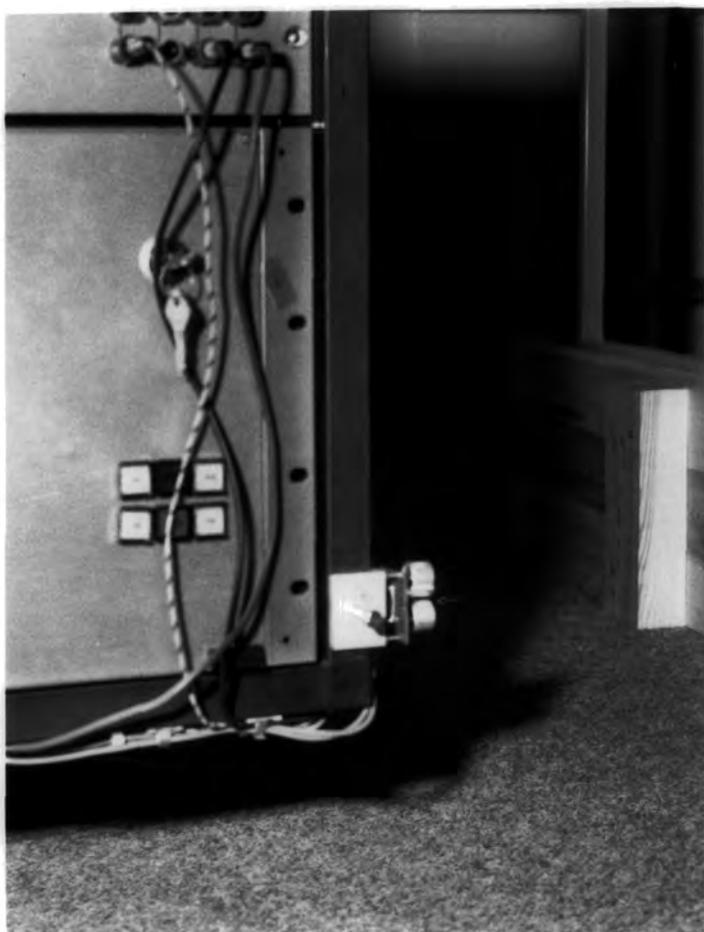


Plate 4.2: The front sensor and the correction board

## AGV COMMAND STRUCTURE, STEERING CONTROL STRATEGY, AND TRANSFORMATIONS FROM SENSING TO CONTROL

### 5.1 Introduction

In the previous chapter, the vehicle guidance method was introduced by explaining how two low-level sensor systems are employed to measure vehicle position. The first part of this chapter aims to expand on this guidance method by setting out the different types of command that the vehicle can execute and explaining how a control strategy was formulated to allow the AGV to follow so called 'imaginary lines'. In formulating this control strategy, the various sensing and control calculations that the AGV has to make become apparent. Further sections in this chapter then derive a number of equations employed by the AGV for position monitoring, speed control, and steering. This paves the way for the next chapter which describes how these equations are integrated into a flexible and modular software package that allows the AGV to move smoothly from some general position in the laboratory  $(X_1, Y_1, \theta_1)$  to some other position  $(X_2, Y_2, \theta_2)$ .

### 5.2 AGV Command Structure

In normal operation, the position feedback for the AGV's control system is provided by odometry and, as has already been stated, corrections are made by the ultrasonic sensing system when the vehicle moves close to a correction board. In addition to this mode of operation, the AGV software allows the control feedback to be taken directly from processed ultrasonic data when the vehicle's planned path runs parallel to either a wall or correction

board. This "wall following" mode is, to some extent, a software anachronism, and was initially developed to examine how well the vehicle behaved when an object was placed near to the correction board. In such instances, when the object is not in a position to cause the AGV to halt, the vehicle can be observed to steer around the object in a controlled manner. The fundamental difference between the line following and wall following modes is the source of the control algorithms feedback. All other aspects of the guidance remain the same; the same control algorithm operates, and corrections are made to the odometric estimate of position.

There are four types of command that can be executed and in the AGV software these are abbreviated as

- (1) WF. Wall follow command (using ultrasonic feedback directly).
- (2) LF. Line follow command (using corrected odometry feedback).
- (3) CF. Curve follow command (using odometry feedback directly).
- (4) ST. Stop command.

Each of these commands (excluding stop) can be thought of as representing an imaginary line which the AGV has to follow. For example, in a "wall follow" command, this line runs parallel to a wall or correction board and is displaced from it by a known distance,  $y_{lw}$ . By employing combinations of the above commands a command queue can be built up which represents a path that the AGV can follow smoothly from its current position to its destination.

Each of the above command types has a parameter string following it, each parameter quantifying some quality of that command type. In the AGV software, a typical printed command queue would be

**WF,F,L,0.2,2.0,0.0,100**

**CF,F,0.1,1.0,90.0,3.0,1.0**

**LF,F,0.2,90.0,3.0,3.0**

**ST,0**

In this example, the first command is to follow a wall or correction board (WF) in the forward direction (F) using its left set of transducers (L) at a speed of 0.2(m/s). The command terminates when the AGV reaches the (X,Y) position (2.0,0.0). With the entered terminating position and the known position of the correction board, the software returns the distance  $y_{tw}$  (which, in this case, is 100cm). The second command is to follow a curve using odometry (CF) in the forward direction (F) on a radius of 1.0 m until an angle of 90.0 degrees has been turned through. The terminating position of this command, although strictly redundant, is also displayed. The third command is a "line follow" at a speed of 0.2m/s and on a global heading of 90.0 degrees. This command terminates when the AGV reaches the (X,Y) position (3.0,3.0). The last command is always a stop, although such commands may be embedded in other parts of the command queue. This has a single parameter printed after its abbreviation, which is the number of seconds the vehicle remains in a stopped state before advancing to the next command. If the stop command is the last command in the queue, this parameter is usually zero.

### **5.3 The Basis of the AGV Control**

The function of the AGV's controller is to derive a turning radius,  $r$ , (or curvature  $\kappa = \frac{1}{r}$ ) along which the vehicle must travel, by measurement of the AGV's position relative to an imaginary line or correction board. Both perpendicular distance to the line and heading relative to the line can be

measured and either used separately or together for control purposes.

The basis of the AGV control is shown in figure 5.1. At each sampling interval, the incremental rotation of the two drive wheels is used in the odometric equations of section 5.6 to generate an update of the AGV's global position,  $(X, Y, \theta)$ . In general, the AGV's designed course does not lie along the X axis and so the odometric estimate of global position is transformed to a position estimate in the local frame of the demand path. If, rather than line following, the AGV is following a wall or correction board using its ultrasonic transducers, this relative positional information is available directly in a local frame parallel to the correction board itself. The position of the AGV relative to the demand path can now be used in a control algorithm to establish the necessary steering control. This steering information, along with the AGV's speed requirement, is fed through a transformation to produce independent speed values for each of the the two driven wheels.

#### 5.4 Formulation of a Control Strategy

This section is concerned with what positional information should be used in the control loop and how it should be used. Figure 5.2a illustrates the problem encountered in using only distance measurements in the control loop. If the AGV is in position 'A', then to follow the defined path, it needs to turn left and then right with the demand curvature gradually reducing as it approaches the line. With simple distance measurements and a proportional controller this is clearly impossible because negative (left turn) and then positive (right turn) curvatures must be generated from the same sign of distance error. Figure 5.2b illustrates this further showing that, although the two distance errors are of different sign, the controller must generate the same sign of curvature in both cases since, in this example, the AGV has to turn left in both cases.

It was also noted that simple closed loop angle control is insufficient in itself since it could only move the vehicle parallel to the desired path and not necessarily on it. Even if the vehicle were on the imaginary line before it started moving, a distance offset would inevitably build up which could not be corrected.

In other words, the control is multivariable in that both yaw angle and perpendicular distance relative to the demand path must be controlled. Since the heading of the AGV affects the way in which distance error changes, the control algorithm initially generates a demand heading relative to the heading of the imaginary line. This always points towards the imaginary line on a gradient that is proportional to the distance error and in the direction that the AGV is travelling. The error in AGV heading with respect to a demand heading derived in this fashion yields a turning curvature (through a proportional controller) that tends to diminish this heading error. In effect then, the AGV can turn left and then right by steering around each of the small heading lines shown in figure 5.2a.

## 5.5 Transformations from Internal Sensing to Control

The control scheme described in the previous section is implemented when the AGV is 'wall following' using its ultrasonic sensors directly, when the AGV uses periodically corrected odometric feedback as in 'line following', and when odometric feedback is used directly as in 'curve following'. It can be recalled that measurements  $y_1$  and  $y_2$  (see figure 4.1) are generated in a control interrupt from a processed batch of ten raw ultrasonic readings. The local position  $(y, \theta')$  required for either direct control or to correct odometric position estimation is derived from these measurements using equations 4.1 and 4.2. In contrast, odometric position monitoring works in global coordinates and in order to generate the local position information required for control

purposes using odometric data only, a transformation from the global frame to a path dependent local coordinate frame is required.

The final three sections in this chapter describe and derive the geometrical calculations required by the control algorithm to generate controlled action from sensory data. Each section can be thought of as describing a coordinate transformation. The next section describes how estimated global position is updated using the incremental rotations of the drive wheels i.e. the transformation

$$\begin{bmatrix} \Delta d_1 \\ \Delta d_2 \end{bmatrix} \rightarrow \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta \theta \end{bmatrix}$$

is described. Section 5.7 describes how this updated global position can be transformed into a local coordinate frame i.e the transformation

$$\begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ \theta' \end{bmatrix}$$

is described. The steering control algorithm itself represents the transformation

$$\begin{bmatrix} x \\ y \\ \theta' \end{bmatrix} \rightarrow \begin{bmatrix} \kappa \\ V \end{bmatrix}$$

An analysis of this control system is left until chapter 7. The final section of this chapter describes the transformation of the control system outputs to individual demand speed for the drive wheels i.e. the transformation

$$\begin{bmatrix} \kappa \\ V \end{bmatrix} \rightarrow \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

is described.

## 5.6 Calculations Required for Odometry

The first calculation required for odometry determines whether the AGV is turning and, if it is, the radius of the turn is calculated. Let  $n_1$  be the number of counts that the encoder on the left wheel has provided in a short time interval  $\Delta t$  and let  $n_2$  be the number of counts provided by the encoder on the right wheel in the same time interval.

Now let

$$\Delta n = n_1 - n_2 \quad (5.1)$$

then the extra distance travelled by one wheel is

$$\Delta d = k_{dr} \Delta n \quad (5.2)$$

where

$$k_{dr} = \frac{\pi d_{wheel}}{100} \quad (5.3)$$

The change in AGV heading is

$$\Delta \theta = \frac{k_{dr} \Delta n}{W_b} \quad (5.4)$$

and the radius of turn can be calculated as

$$r = \frac{\bar{d}}{\Delta \theta} \quad (5.5)$$

where

$\bar{d}$  = average distance travelled by drive wheels

Thus

$$r = \frac{k_{dr}(n_1 + n_2)}{2\Delta \theta} \quad (5.6)$$

and by substituting for  $\Delta\theta$  this can also be written as:

$$r = \frac{W_b(n_1 + n_2)}{2(n_1 - n_2)} \quad (5.7)$$

Equation 5.4 defines the change in heading experienced by the AGV and the new heading is obtained by adding this to the current value. If there is no change in heading, x and y coordinate changes are resolved from the distance travelled over the interval and the current heading, otherwise a calculation illustrated by figure 5.3 is made. Firstly the orthogonal components in the local frame of the AGV's initial heading are calculated as shown in figure 5.3 by:

$$x = r \sin \Delta\theta \quad (5.8)$$

$$y = r(1 - \cos \Delta\theta) \quad (5.9)$$

These two quantities are then resolved into their respective X and Y components giving four quantities in all. On the figure 5.3 these are:

$$\delta X_1 = r \sin \Delta\theta \cos \theta \quad (5.10)$$

$$\delta Y_1 = r \sin \Delta\theta \sin \theta \quad (5.11)$$

$$\delta X_2 = r(1 - \cos \Delta\theta) \sin \theta \quad (5.12)$$

$$\delta Y_2 = r(1 - \cos \Delta\theta) \cos \theta \quad (5.13)$$

These four quantities add or subtract depending on vehicle direction, vehicle heading, and the direction in which it is turning. In the particular case illustrated, the magnitudes of the coordinate changes are given by

$$|\Delta X| = r[\sin \Delta\theta \cos \theta - (1 - \cos \Delta\theta) \sin \theta] \quad (5.14)$$

$$|\Delta Y| = r[\sin\Delta\theta\sin\theta + (1 - \cos\Delta\theta)\cos\theta] \quad (5.15)$$

or in matrix form:

$$\begin{bmatrix} |\Delta X| \\ |\Delta Y| \end{bmatrix} = r \begin{bmatrix} \sin\Delta\theta & -(1 - \cos\Delta\theta) \\ (1 - \cos\Delta\theta) & \sin\Delta\theta \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \quad (5.16)$$

Note that if a left turn had been executed in the odometric interval the resolved 'X' components would be additive and the 'Y' components would subtract. This is catered for since both  $r$  and  $\Delta\theta$  are negative for left turns. It should also be noted that since  $\theta$  is defined differently from the conventional anticlockwise sense, the modulus of its sine and cosine are employed in the above equations. In order to generate the correct signs in odometry, two extra variables are introduced into the software;  $s_x$  and  $s_y$ . The first of these has the value +1 in the first and fourth quadrants and -1 in the second and third. The second variable,  $s_y$ , has the value is +1 in the first and second quadrants and -1 in the third and fourth. This directional information is included in equations 5.14 and 5.15 to give:

$$\Delta X = (d_n \cdot s_x)[\delta X_1 - (d_n \cdot s_x \cdot s_y) \delta X_2] \quad (5.17)$$

$$\Delta Y = (d_n \cdot s_y)[\delta Y_1 + (d_n \cdot s_x \cdot s_y) \delta Y_2] \quad (5.18)$$

where  $d_n$  is a variable defined as +1 in the forward direction and -1 in the reverse direction. Thus, in matrix form, the general equation for updating odometric position is:

$$\begin{bmatrix} X_i \\ Y_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} X_{i-1} \\ Y_{i-1} \\ \theta_{i-1} \end{bmatrix} + r \cdot d_n \begin{bmatrix} s_x \sin\Delta\theta & -d_n s_y (1 - \cos\Delta\theta) & 0 \\ d_n s_x (1 - \cos\Delta\theta) & s_y \sin\Delta\theta & 0 \\ 0 & 0 & \Delta\theta \end{bmatrix} \begin{bmatrix} \cos\theta_{i-1} \\ \sin\theta_{i-1} \\ 1 \end{bmatrix} \quad (5.19)$$

This section has described how an odometric estimate of global position is made. In order to implement the control described in section 5.4, the vehicle's position (perpendicular distance  $y$  and yaw angle  $\theta'$ ) relative to the line it is following has to be known at all times. In addition, the distance to the end of the current command segment has to be known in order to detect both the termination of a command and the position at which deceleration must be initiated if the demand speed of the subsequent command is lower than that of the current command. As has been stated before, this line may be either straight or a curve of constant radius. The calculations required to transform global position into a path dependent local coordinate frame are described in the following section.

## 5.7 Transformation into a Path Dependent Local Coordinate Frame

### 5.7.1 Line Following Transformations

Consider a line being followed which starts at  $(X_{0_i}, Y_{0_i})$ , ends at  $(X_{0_{i+1}}, Y_{0_{i+1}})$ , and has a heading  $\theta_{0_i}$ , with the AGV being located at  $(X, Y)$  as shown in figure 5.4. A line (a), which has the same X coordinate as the AGV is projected onto the imaginary line being followed and a Y coordinate  $Y_{line}$  is calculated as:

$$Y_{line} = \frac{(X - X_{0_i})}{(X_{0_{i+1}} - X_{0_i})}(Y_{0_{i+1}} - Y_{0_i}) + Y_{0_i} \quad (5.20)$$

The perpendicular distance from the AGV to the imaginary line is then given by

$$y = (Y - Y_{line})\cos\theta_{0_i} \quad (5.21)$$

where  $\theta_{0_i}$  is the demand heading of the  $i$ th command.

In figure 5.4, the lengths of the two right angled triangles with the

common (dotted) hypotenuse can be equated to determine the distance left to the end of a segment as:

$$L = \sqrt{(X_{0_{i+1}} - X)^2 + (Y_{0_{i+1}} - Y)^2 - y^2} \quad (5.22)$$

The equations above apply to the first and third quadrants of figure 5.4. If the heading lies in the second or fourth quadrants, the calculations are slightly different because a projection is made from the AGV to the imaginary line vertically instead of horizontally. In the general case, this change is made to improve the accuracy of the calculations and in the specific case of a 90 degree heading it is a necessity, since it is impossible to project a horizontal line onto another parallel horizontal line.

In the second and fourth quadrants the equations are:

$$X_{line} = \frac{(Y - Y_{0_i})}{(Y_{0_{i+1}} - Y_{0_i})}(X_{0_{i+1}} - X_{0_i}) + X_{0_i} \quad (5.23)$$

$$y = (X_{line} - X) \sin \theta_{0_i} \quad (5.24)$$

Note that equations 5.24 and 5.21 for the perpendicular distance  $y$  differ slightly to keep a sign convention of a positive distance being to the right of the imaginary line.

If it is recalled from section 4.3 that a position correction requires transforming measured position in a local frame into the global frame, then the transformation from global frame to local frame is simply the inverse transformation. Thus inverting equation 4.3

$$\begin{bmatrix} x \\ y \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos \theta_0 & \sin \theta_0 & 0 \\ -\sin \theta_0 & \cos \theta_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ \theta - \theta_0 \end{bmatrix} \quad (5.25)$$

where  $(X_0, Y_0, \theta_0)$  are the coordinates of the local frame in the global frame. The construction required for this transformation is shown in figure 5.5. If this method were used, the distance remaining in the command segment could easily be generated by calculating the segment length from the command queue array and subtracting the local coordinate  $x$ .

### 5.7.2 Curve Following Transformations

Consider a curve that is being followed which has radius  $r$ , centre  $(X_c, Y_c)$ , initial heading  $\theta_s$ , and arc length  $\Delta\theta$ . The current position of the AGV is  $(X, Y)$  as shown in figure 5.6. Radius, arc length, and centre are parameters obtained by accessing the current command of the command queue array. Initial heading is simply the (terminating) heading of the previous command in the queue. The centre coordinates are generated at the command entry stage with the equations

$$X_c = X_s - r \sin \theta_s \quad (5.26)$$

$$Y_c = Y_s + r \cos \theta_s \quad (5.27)$$

where  $X_s, Y_s$  are the terminating coordinates of the previous command. In this way, coordinates of the centre of the curve are simply accessed from the queue rather than recalculated at each control interrupt.

Angle monitoring requires calculating AGV position relative to the centre of the curve in the command queue regardless of what the heading of the AGV actually is. In order to implement the designed control, the AGV's position  $(y, \theta')$  in a path dependent local frame needs to be calculated. This local frame has its  $x$  axis parallel to a tangent of the curve and is situated on this curve at a point where it intersects with a line from the centre of the curve to the AGV (see figure 5.6).

In order to calculate the orientation of this local frame, an equivalent orientation in the first quadrant is generated as:

$$\theta_{quad} = \tan^{-1} \frac{|X - X_c|}{|Y - Y_c|} \quad (5.28)$$

This angle is then adjusted to the correct quadrant in accordance with the signs of  $X - X_c$  and  $Y - Y_c$ . Thus the orientation of the local coordinate frame ( $\theta_0$ ) shown in figure 5.6 is obtained when a clockwise (right) turning direction is assumed. If the vehicle is turning left (in which case the sign of the demand radius is negative),  $\pi$  is added to calculated angle which is then readjusted (if necessary) to be in the range 0 to  $2\pi$ . Subsequently the distance remaining in the curve segment can be calculated as:

$$L = |r \cdot (\theta_0 - \theta_s)| \quad (5.29)$$

(Note that, in the AGV software, instances of local angle passing through the discontinuity between 0 and  $2\pi$  are catered for.) The perpendicular distance from the curve is calculated as:

$$y = r - \text{sgn}(r) \sqrt{(X - X_c)^2 + (Y - Y_c)^2} \quad (5.30)$$

Recalling that a positive radius denotes turning to the right, equation(5.30) generates distance errors that are positive to the right of the curve.

As has already been stated, a transformation of global position to local position is required to enable the AGV's controller to derive an appropriate demand turning curvature. Once such a curvature has been developed, it must be transformed, along with the current base speed, to give independent left and right wheel speed demands. This transformation is the concern of the following section.

## 5.8 Independent Wheel Speed Control: Base Speed Control and Steering

As was described in chapter 3, the MC68000 microcomputer has a number of VIA chips (Motorola 6522) which are used to supply the pulses to the stepper motor translators. These have internal timers which can toggle the logic state of a pin on the output port when the loaded count terminates. If appropriately configured, when the logic state is toggled, the counter will reload the latched value and recommence count down. Thus by loading the counter with different values, the number of pulses per second supplied to the steppers can be varied.

For one pulse to be supplied to a stepper motor, the VIA output pin must be toggled twice; thus the latched counter value must be decremented through to zero twice, and the time ( $t$ ) for one pulse to be supplied to the stepper motor is given by

$$t = \frac{2n}{f_c} \quad (5.31)$$

where:

$f_c$  = clock frequency (0.8MHz)

$n$  = latched counter value

This gives the relation between the latched count value and the stepper motor speed at control interval  $i$  ( $V_i$ ) in steps per second (sps) as:

$$V_i = \frac{400000}{n_i} \quad (5.32)$$

Essentially the AGV drive system has two degrees of freedom which may either be described on a local basis as the individual stepper speeds ( $V_1, V_2$ ) or on a global basis as base speed and turning radius ( $V, r$ ). Here base speed is the velocity of a point midway between the drive wheels which

is the position at which all of the AGV's coordinates are defined. There is a one to one mapping between the global and local frames of reference such that:

- (1) A unique combination of stepper speeds exclusively defines a single combination of base speed and turning radius

and therefore

- (2.) the speed and steering of the AGV can be simultaneously and independently controlled by varying the stepper speeds.

Steering is effected by reducing the speed of one wheel by a fixed amount and, in order to make base speed independent of turning radius, the other wheel speed is increased by an equal amount. In figure 5.7, the AGV has turned on a constant radius,  $r$ , through an angle,  $\Delta\theta$ , in time  $\Delta t$ . Calculations based around this figure yield relationships between the turning radius and the individual wheel speeds such that:

$$r_1\Delta\theta = V_1\Delta t \quad (5.33)$$

$$r_2\Delta\theta = V_2\Delta t \quad (5.34)$$

$$r_1 - r_2 = W_b \quad (5.35)$$

$$r = r_2 + \frac{W_b}{2} \quad (5.36)$$

Combining equations 5.33 and 5.35 gives:

$$\Delta\theta = \frac{V_1\Delta t}{r_2 + W_b} \quad (5.37)$$

Subtracting equation 5.34 from equation 5.33 and substituting 5.35 gives:

$$W_b \Delta \theta = (V_1 - V_2) \Delta t \quad (5.38)$$

Substituting equation 5.37 in equation 5.38 gives

$$r_2 = \frac{V_2 W_b}{V_1 - V_2} \quad (5.39)$$

and substituting equation 5.39 into 5.36 gives:

$$r = \frac{W_b(V_1 + V_2)}{2(V_1 - V_2)} \quad (5.40)$$

Equation 5.40 shows that if  $V_1 = V_2$  then  $r = \infty$  and the AGV travels in a straight line, if  $V_2 = 0$  then  $r = \frac{W_b}{2}$  and the AGV pivots on its right wheel, and if  $V_2 = -V_1$  then  $r = 0$  and the AGV spins on the spot.

It is clear that

$$V_1 + V_2 = 2V \quad (5.41)$$

where

$V =$  base velocity

Now define  $V_{rel} = V_1 - V_2$  so that:

$$r = \frac{W_b V}{V_{rel}} \quad (5.42)$$

In order to implement steering control along some radius ( $r$ ), one wheel speed must be increased relative to the base speed by an amount  $\frac{V_{rel}}{2}$  and the other decreased by this amount. Thus the change in speed for one wheel is calculated by:

$$\frac{V_{rel}}{2} = \frac{W_b V}{2r} = \frac{W_b V \kappa}{2} \quad (5.43)$$

## 5.9 Summary

In this chapter, the various types of command that the experimental AGV can execute have been described along with an appropriate control strategy. In addition, the geometric calculations required to implement such a control have been described and derived. The following chapter describes how these calculations are integrated into a modular software structure.

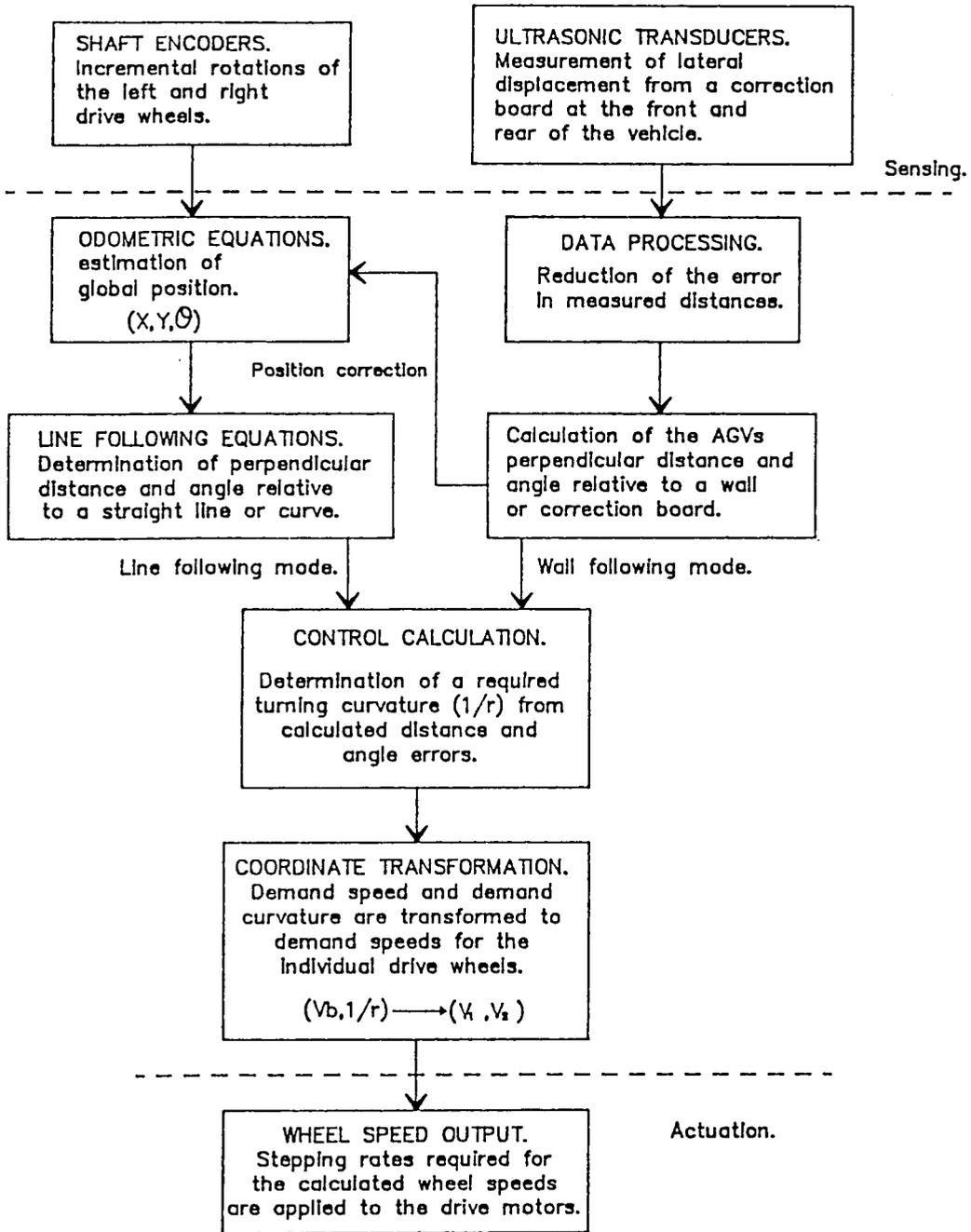


Figure 5.1 The basis of the AGV control

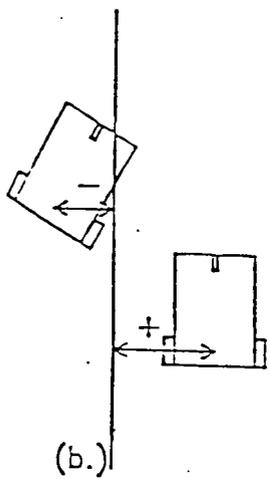
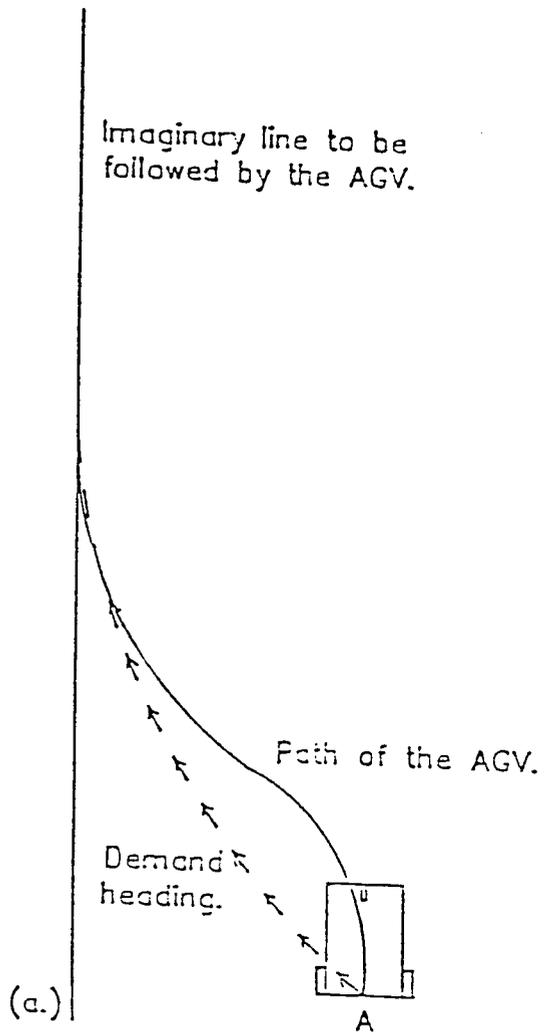


Figure 5.2 The control method

Global frame.

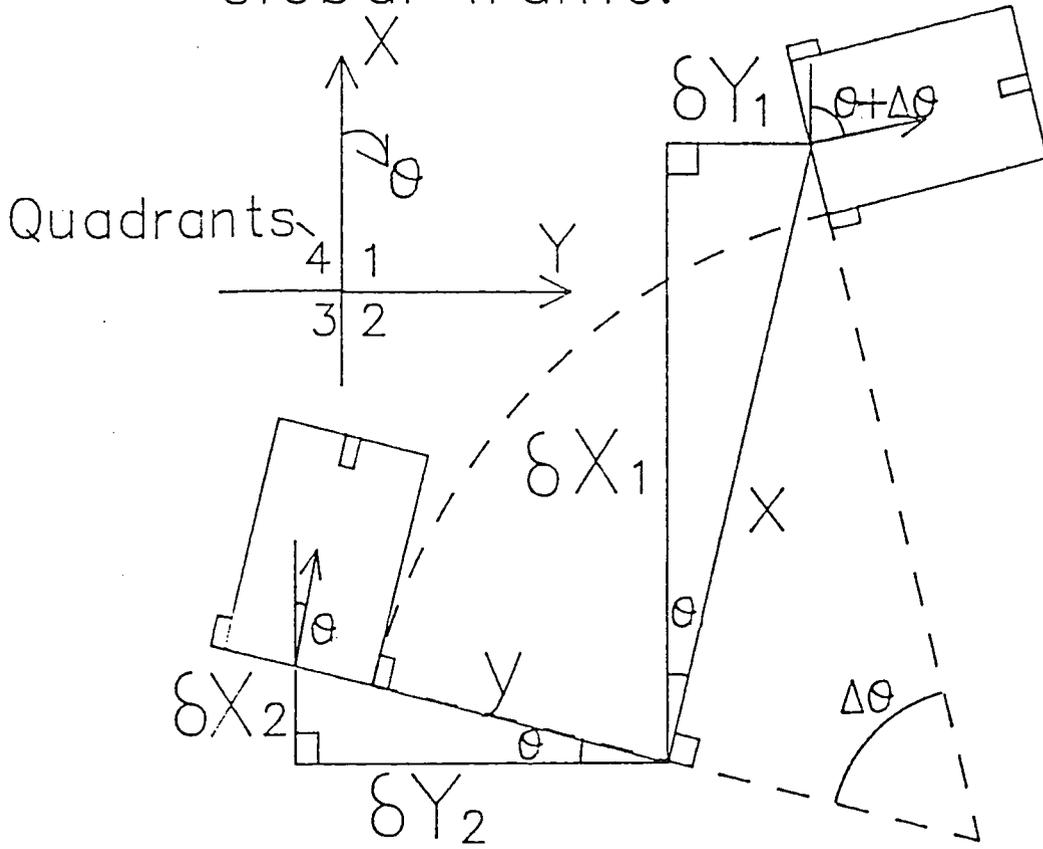


Figure 5.3 Odometric geometry

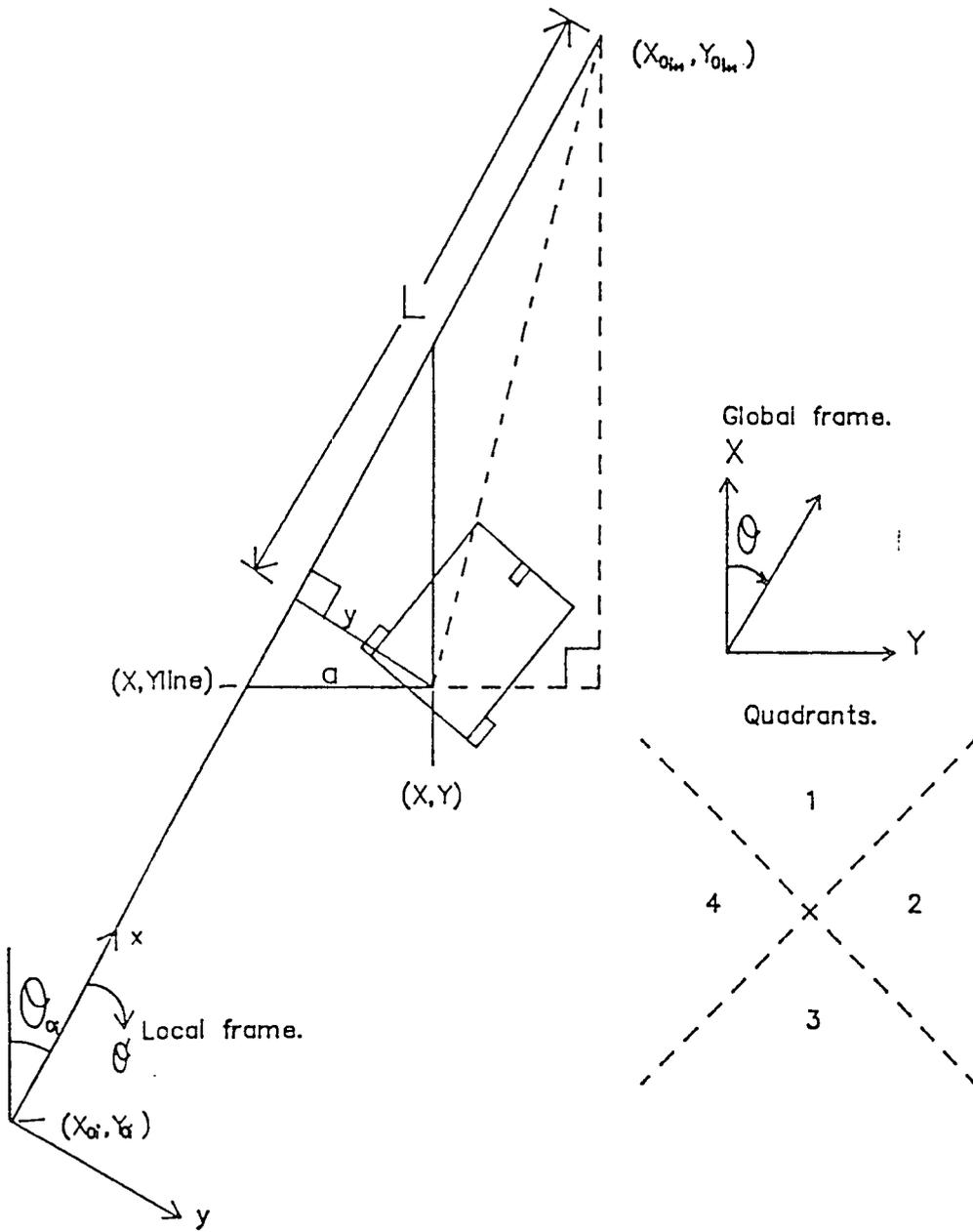


Figure 5.4 Transformation to the local frame of a line segment

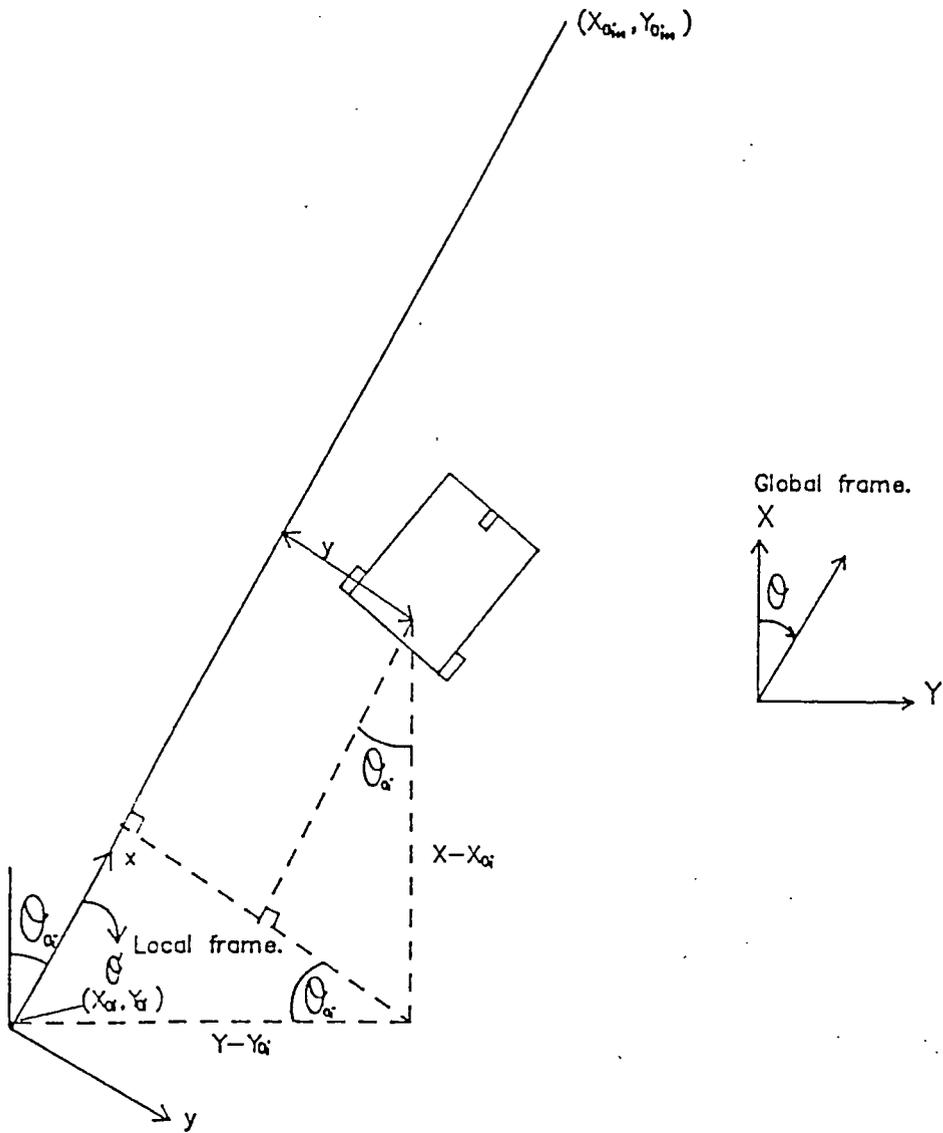


Figure 5.5 An alternative transformation to the local frame

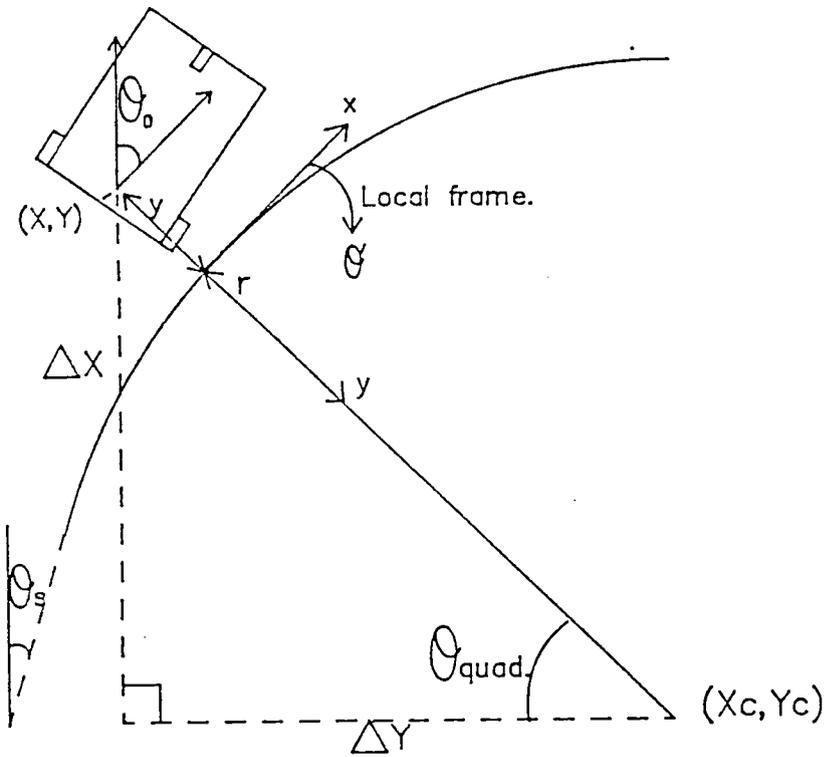


Figure 5.6 Transformation to a local frame in a curve segment

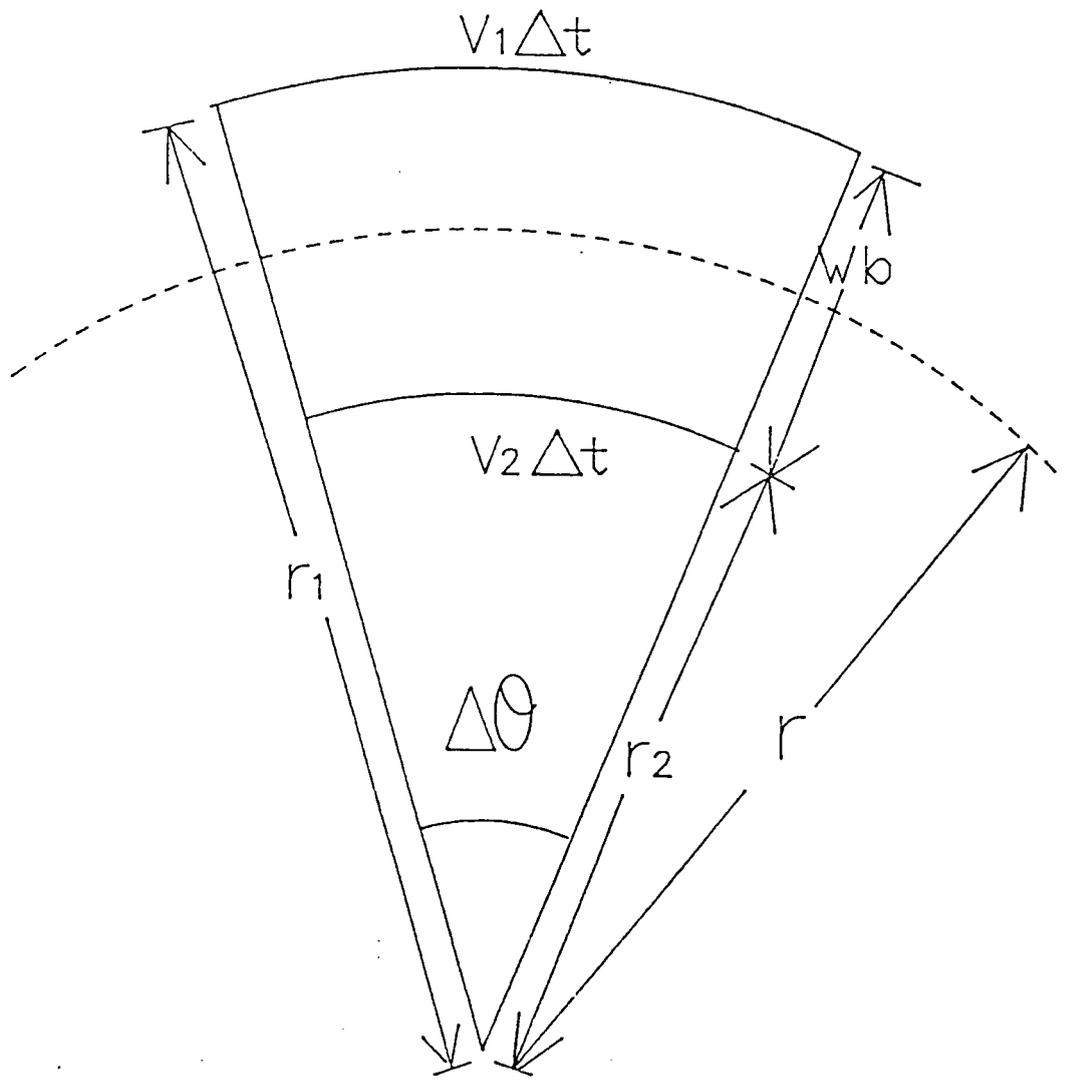


Figure 5.7 The relationship between wheel speeds and turning radius

## CHAPTER 6

# A MODULAR SOFTWARE PACKAGE FOR USER INTERFACE AND AGV CONTROL

### 6.1 Introduction

In this chapter a software package is described which is designed to allow the user to build up a queue of commands which represents a planned path that the AGV can follow. The reasons for adopting such an approach stem back to decisions made about how the systems intelligence should be divided between an individual AGV and a central computer.

It was clear that an AGV should be able to guide itself round its work area following destination commands provided by the central computer. However, it was not so obvious where the path planning required to follow these commands should reside. In summary it was decided that since there would be several AGVs working in the warehouse or factory with the obvious potential of collisions, path plans should be devised by a central computer. The results of the path planning algorithm could then be sent across a radio link as a queue of simple commands which the AGV can interpret and then execute sequentially using its low level sensing systems. This is a flexible system because

- (1) Changes in the layout of the work area only requires a reprogramming of the central computer.
- (2) More AGVs can be added without having to give them a detailed map of the environment. (However, additional AGVs would have to be given the position of the correction boards in the environment.)

(3) Collision avoidance is facilitated. For example, the central computer could send a queue of commands to an AGV for which it knew there was no potential collision with another AGV. At the end of this command queue the AGV would be in a wait state, unless the central computer has reassessed the situation and updated the queue before it actually reaches this state.

The software used in this research was designed to allow the user to construct paths and pass them to the research vehicle. Thus the user effectively takes the role of the central computer and research could be focused on the AGV itself. It should be noted here that much of this chapter is concerned with the content and structuring of the AGV software. The details and operation of a number of control and data processing algorithms are best demonstrated using collected results. This is left until chapter 7 and chapter 8 respectively.

## 6.2 An Overview of the Software Structure

A schematic diagram of the software structure is given in figure 6.1. This indicates that the AGV's routines operate on one of three levels: background, low interrupt level 3, or high interrupt level 4.

The functions of the background task are fourfold, namely

- (1) To set up the AGV rig ready for a test run.
- (2) To load, save, build or edit either a command queue array or a correction board array.
- (3) To prepare for, initiate, and monitor the execution of commands in the queue.

(4) To display collected results after a test run and/or to save them to a memory buffer ready for transfer to a disc file.

Whereas the background routines provide a 'test harness' in which to analyse the performance of the AGV, the control, data acquisition, and data processing algorithms for the vehicle reside in interrupt software.

A timer on one of the on-board microcomputer's interface chips generates a low level interrupt (level 3) every 100mS. The interrupt service routine for this level contains both the odometry algorithm and the stepper motor control algorithms. These control algorithms have to be interrupt based in order to provide a constant sample period for the control action and, since odometry is an incremental system utilised by the main control algorithm, it is sensible to have this in the same service routine prior to the control computation itself.

More specifically, the control interrupt service routine (*cisr*) contains calls to a number of routines namely

1. *odom* - Odometry (or global position update).
2. *localpos* - Local position transformation.
3. *datproc* - Data processing (including position correction).
4. *ramp* - Base speed control.
5. *controller* - Steering controller.
6. *transform* - Speed transformation and curvature rate limiting.
7. *wspout* - Wheel speed output.

Each of these is shown as a block under the heading of 'low level interrupts' in figure 6.1. The first routine, *odom*, updates the odometric estimate of the vehicle's global position by reading encoder count values from each wheel and using the equations derived in section 5.6. The following

routine (*localpos*) uses this updated global position, along with the current command dimensions read from the command queue array, to determine the corresponding position in the local coordinate frame of the current command (see section 5.7). Subsequently *datproc* processes two sets of ultrasonic range measurements, which have been stored in globally accessible arrays over the last control interval, by initially digitally filtering the results and then applying a least squares line fitting procedure. This procedure, which utilises the current and previous local  $x$  coordinate positions in order to associate an 'along board' (local  $x$ ) coordinate with each of the filtered ultrasonic measurements, is required in order to implement a 'notch detection' algorithm, to correct position, and to derive direct control information in the 'wall following' mode. The function of *ramp* is to update the vehicle's base velocity, if this is required, whilst *controller* contains the control software which generates a demand turning curvature from the current estimate of local position. Calculated values of base velocity and turning curvature are then transformed, by the routine *transform*, into individual wheel speeds. This routine may also limit the amount by which the speed of a drive wheel can change and thus prevents stalling of the stepper motors. Individual wheel speeds are then passed to the wheel speed output routine (*wspout*) in order to effect the changes required by the control.

It should be noted here that, with the exception of the somewhat arbitrary position of the routine *ramp*, which can be anywhere after *localpos* and before *transform*, the ordering of the low level interrupt routines is a consequence of necessity, in that each routine requires results from one or more of the previous routines.

External interrupts generated by the ultrasonic interface boards have the highest permitted interrupt priority level (4) to guarantee accurate timing of an ultrasonic pulse. These interrupts occur roughly twice (one start and

one stop interrupt) every 10mS for each pair of enabled transducers.

The choice of a 10mS pulsing interval was chosen to prevent reflected pulses being received after a subsequent pulse had been transmitted, thus generating a low, erroneous reading. The range of the ultrasonic ranging devices employed is around 1.5m. A pulsing interval of 10mS permits a range of 1.685m ( $V_s = 337m/s$ ) in which to detect a return echo unambiguously.

The selection of the control interrupt frequency was made by a compromise between

- (1) A high enough value relative to the response times of the control system. Chapter 7 shows that it takes approximately 12 seconds for the control system to attain 95% of a step change. The control sampling interval (100mS) is less than one hundredth of this value.
- (2) A low enough value to permit sufficient time for all the necessary data processing and control calculations that are required in the control interrupt.

Communication between the background task and the two levels of interrupt software is realised by using a globally accessible command queue array, ultrasonic data arrays, and a small number of other global variables. Typically ten ultrasonic readings have been taken from a transducer pair before a low level control interrupt occurs. These data are stored in an ultrasonic data array which can be read by low level interrupt software for processing and then used for either direct control or position correction. The command queue array forms most of the communication between the background task and the low level control interrupt software.

The command queue exists in software as a global, two dimensional

integer array in a format shown by figure 6.2. It can be seen that the different commands occupy one dimension and the various parameters for these commands occupy the other. It can also be noticed that the first six entries have been designed to be common throughout the queue. It can be argued correctly that, with this arrangement, some of the information within the array is redundant. For example, in a stop command the demand speed will always be zero and the position and heading entries are redundant because they are simply equal to the terminating position and heading of the previous command. Additionally, the terminating position of a curve is redundant because it can be derived as a function of turning radius, turning angle, and the starting position of the curve (i.e. the terminating position of the previous command). Significant improvements in modularity are obtained by incorporating such redundancy into the command queue as whenever any of the five common variables need to be either set or accessed, this can be done without reference to the command type because they occupy the same position in the command irrespective of type.

In addition to the command queue array there is a correction board array which contains information concerning the global position and orientation of the correction boards in the AGV's workspace. It also defines the distribution of notches on each of the correction boards.

Now that an overview of the AGV software has been given, the remainder of this chapter examines each part of the software in more detail, starting with the background routines, going on to the low level control interrupt software, and then high level ultrasonic timing interrupts. A final section is devoted to data logging and data analysis software. It should be noted that many of the routines are entirely sequential, or contain simple monitoring loops; therefore, only some of the more important branched routines are detailed in flowchart form.

### 6.3 The Set Up Procedure

The first 'c' routine entered (*main*) contains the set up procedure for the AGV. In their correct sequence, the preliminary operations embedded in this routine are

1. Mask all of the system's interrupts.
2. Configure the interface chips of the on-board microcomputer (VIA chips and timer chips).
3. Associate hard-wired interrupt request levels with service routine start addresses.
4. Print a software package header.
5. Create integer look up tables for a variety of functions (sin, cos, etc.).
6. Initialise global variables.

Look up tables are required because all interrupt software operates in integer arithmetic. Global variable initialisation allows certain variables to be changed from a default value. These include the nominal acceleration rate of the AGV, ambient temperature, a mode of data collection, and those parameters concerned with the software controller.

After this set up procedure, a call to the main menu (*menu*) initiates a test run and, unless an error condition is trapped or the abort key is pressed, *main* is ultimately returned to at the termination of such a run. Before control returns to the on-board microcomputer's monitor, options are provided for displaying lists of collected results on the terminal screen and saving results to a memory buffer ready for transfer to a disc file.

## 6.4 The Menu

The main menu is contained within a routine called *menu* which is called from *main*. The options available in this menu are shown in figure 6.3. It can be seen that the main body of menu options (1 to 6 and 8 to 13) primarily allow the manipulation of one of two global integer arrays, namely the command queue array and the correction board array. Note that the same routines operate on two different sets of values by passing the appropriate array name to general routines on a given menu option. The facilities available in the menu for manipulating these two arrays provides a flexible system for creating and altering AGV routes and for adding or moving correction boards in the AGV's workspace.

Other options in the menu allow basic path planning (option 7), so called 'board searching' (option 14), quit to *main* (option 0), and the execution of the command queue itself. The 'execute' option (15) is designed to execute, sequentially, each command in the queue until the bottom of the queue is detected. The main prompt in *menu* is then returned to allowing, for example, further additions to the queue whilst the AGV is in a wait state.

The menu provides for a variety of options. For example, a basic queue could be loaded from disc which may then be built up and elaborated using the 'build' option (5). This could then be tested by taking the 'execute' option at the main prompt and finally, the elaborated queue could be saved under a different filename. Alternatively, a queue could be built up and tested in small sections or even typed in and saved without running the AGV at all.

The following sections (6.5 to 6.10) briefly describe the different options available in the main menu.

## 6.5 Loading and Saving Arrays

Although default values exist in the correction board array, the command queue array is initially empty and since nothing can be done unless some form of queue exists, the first option must be to either load (1) or build (5) a command queue. (The menu informs the user when the command input pointer is 1 and, therefore, the queue is empty.) If the user takes one of the load options, a filename is entered and data in this file is moved into the appropriate array after that array has been cleared. The command queue, or correction board array, thus loaded is then printed by the appropriate display routine; an example of a displayed command queue was given in section 4.2. If the display is not what the user wanted, the menu allows repeated re-entries of filenames and will clear, load, and print the array each time.

'Save' is the option that allows either the current correction board array or the current command queue array to be written onto a disc file under a specified filename for later use.

## 6.6 Editing and Appending Arrays

If a command queue has been loaded or built, it can be altered using either the edit or append options. The edit option invokes a simple editor which has the sub menu

0. Quit (to main menu).
1. Display current command queue.
2. Edit command.

If option 2 is taken, the user enters the position of a command in the queue. This command is then displayed and the routine permits editing of any of the parameters in the displayed command. After editing a command,

it is validated to ensure that it is compatible with the previous command in the queue.

The append option in the main menu also requires the entry of the position of a command in the current queue. From this position onwards, the contents of the array are deleted (set to zero) and the command input pointer is set to the entered command position. This option is useful if a large part of a previously stored command queue is required and only the last few entries need rebuilding.

## 6.7 Building Commands

If the 'build command' option is taken, a sub menu is entered which has five options

0. Quit (to main menu)
1. Wall follow command entry
2. Line follow command entry
3. Curve follow command entry
4. Stop command entry

The response to the entry of a particular command type is a series of prompts for values of a minimum set of parameters that completely define that particular command type. The command so entered is not loaded into the queue until it has been validated by a call to the routine *validate*. This seeks to trap errors in the input of command parameters firstly by checking that they fall within sensible limits, and secondly by checking for incongruities between the current and previously entered command. Examples of such incongruities are direction changing without stopping, heading changing without a curve segment, and a stop command following another stop command. If this

validation routine returns *TRUE*, the command is loaded into the queue, the command input pointer is updated, and the command string representing the entered values is printed before the sub-menu of command types is re-entered. If the validation fails, the *validate* routine indicates the nature of the command specification incongruity and the routine returns to the command type sub-menu without changing either the command queue or the command input pointer.

Once again it can be noted here that more data is stored in the command queue than is actually entered in the routine *build*. Firstly with a curve follow entry, terminating position is not strictly required to execute a curve yet it is calculated from the entered values of demand radius and turning angle, using the equations derived for odometry in section 5.6, and is loaded into the command queue array. Additionally, demand speed is derived as a function of the entered demand radius by calling *spcal* and is entered into the array. With a stop command entry, no command parameters are entered but values of heading and terminating coordinates are copied into the array from the previous command. These are the built in redundancies mentioned earlier which maintain the software modularity.

## 6.8 Path Plan Option

The simplest solution to execute a right angled curve would be to stop and then rotate on the spot. However, this takes time and additionally conventional steering mechanisms are not normally attitude and position independent, in that they need to move in order to turn. Therefore, a path plan option is available in the main menu (option 7) which allows basic off line path planning and can calculate command parameters on the basis of entered corner positions and aisle widths. This is not the comprehensive planning routine that would reside in a central controlling computer but a simple aid to

inputting the right command parameters for right angled corners, T-junctions, and crossroads in the laboratory. The routine *planner*, called from the main menu, requires four pieces of information to be entered. Referring to figure 6.4, these are aisle width ( $a$ ), clearance ( $c$ ), and the global coordinates of the inside corner ( $X_{corner}, Y_{corner}$ ). Using these values in the function

$$f(r) = (1 + \frac{1}{\sqrt{2}})c + \frac{1}{\sqrt{2}}(\frac{w}{2} - r) + (r + \frac{w}{2})\sin\theta + s_f\cos\theta - a = 0 \quad (6.1)$$

a demand radius can be calculated using Newton-Raphson iteration. For this iteration, the differential of equation 6.1 is given by

$$f'(r) = [\frac{\psi}{1 + \psi^2}]\cos\theta + [\frac{\psi^2}{1 + \psi^2}]\sin\theta - \frac{1}{\sqrt{2}} \quad (6.2)$$

where

$$\psi = \frac{2r + w}{2s_f} \quad (6.3)$$

and:

$$\theta = \tan^{-1} \psi \quad (6.4)$$

These equations are derived in appendix 3. From the calculated value of demand radius, a starting position can be calculated so that the back right and front left corners of the AGV clear the aisle by the entered value of clearance.

The calculation of a starting position for a curve command means that the AGV must have attained a specific lateral position at a specific point before the corner itself at which the curve command starts (see figure 6.4). This means that if the vehicle is initially travelling down the centre

of an aisle, a lane changing strategy must be adopted. In addition, the exit from the corner may leave the AGV displaced from its required central lane position thus requiring a further change in lane. Such a motion can be observed when a car swings out to enter its drive and then levels up to enter its garage.

## 6.9 Board Searching

If the 'board search' option is taken in the main menu, the contents of the command queue array are matched against the contents of the correction board array in order to determine the commands in which the AGV can make position corrections. If a command is sufficiently close and parallel to a correction board, the information that the AGV requires in order to make a position correction is inserted into the appropriate columns of that commands row in the command queue array. This information relates to the position of a correction board in the local frame of that command segment. More specifically, the perpendicular separation of the command and the correction board,  $y_{lw}$ , and the local frame positions,  $x_0$ ,  $x_1$ , between which *both* sensors can observe the correction board are entered into the array as is shown in figure 6.2. (Note that the local ( $x$ ) positions of the centre of observable notches are entered into a separate 'notch array'. During development this was done simply to avoid a large number of software alterations required by expanding the command queue array.)

For all linear command segments in the current command queue, the board search routine matches command parameters against each of the correction boards in sequence. For a correction to be made a number of conditions must exist in the parameter matching, namely:

- (1) The command segment and correction board must be parallel.
- (2) The perpendicular separation must be within an observable distance (1.5m).
- (3) The segment and board must overlap by at least 10cm which is the distance,  $s_r$ , between a rear sensor and the AGV reference point.

For a correction board and command segment pair that are parallel, their perpendicular separation and the distance between the start of the command and the end of the correction board ( $L$ ) are calculated using the local transformation equations described in section 5.7.1. If the perpendicular separation is within an observable range, the linear overlap, or linear separation, is calculated by comparing the sum of the lengths of the board and command segment and the computed length,  $L$ .

Taking into account both the direction of travel of the AGV and the position of the AGV's sensors relative to its reference point (i.e. in a vehicle centred frame), the local  $x$  positions in the command segment between which *both* sensors may observe the board are determined. This may result in the original overlap between the command segment and correction board being shrunk slightly.

In order to determine which notches are observable, the length of the command segment is expanded according to the command direction and the vehicle centred sensor positions to form a window. Those notches on the correction board that fall in this window are observable and loaded into the notch array.

In executing a 'board search' real time is saved, since information concerning the position of correction boards in the local frame of specific



segments is calculated prior to the AGV executing its command queue and is available in the notch array and the command queue array itself. Note that if this option is not taken before a test run, no corrections are made since the vehicle 'thinks' it can not observe any correction boards. Thus a comparison of corrected and uncorrected (i.e. odometric) performance is easily implemented.

## 6.10 Command Queue Execution

If the 'execute' option (option 15) is taken in the main menu, a routine called *stpchk* is called before the execute routine itself. This checks to see if the last command in the queue is a stop and if this is not so, it will automatically place a stop command with a zero second wait at the end of the queue, update the command input pointer, and inform the user of the change.

Since most of the processing required to execute a command takes place in interrupt software, the *execute* routine primarily acts in a simple monitoring capacity and increments the command execution pointer when the current command is deemed terminated. Thus the AGV steps through the current queue of commands until the end of the queue is detected.

For the execution of each command, a screen of on-line information is maintained which includes global vehicle position, speed, acceleration, and so on. On entering the set up of a particular command in *execute*, the position of the command in the queue and the command parameter string are printed at the top of this screen. This is followed by a call to an interrupt enable routine (*enable*) which has the function of enabling timed low level interrupts if the vehicle is in the stopped state, and enabling the correct set of ultrasonic transducers, either the front and two left sets or the front and two right sets,

according to sign of the sixth entry in the command queue array. This refers to the separation of the command segment from the correction board and is denoted by  $y_{lw}$ . Subsequently, since low (and high) level interrupts are now active, and the command execution pointer points to a new command, the interrupt software can read the demand speed of this command and the AGV will accelerate to this speed.

If the current command is not a STOP command, a routine called *cprep* is called which is the command preparation routine described in the next section. This command preparation routine ultimately returns to *execute* when a command is deemed to be terminated. *Execute* will then increment the command execution pointer, display the new command string and, if necessary, change the enabled sets of ultrasonic transducers. Alternatively, if the end of the command queue is detected, it will return to the main menu. Since the command queue array initially contains all zeros before any commands are entered into the queue, the end of the queue is detected by the command type being equal to the null value (zero).

For the execution of STOP commands, *execute* calls a routine called *stopmon*. This simply returns when either a key is pressed or, failing that, when a timer, loaded with  $n$  seconds from the command queue array, times out. Subsequently, *execute* moves onto the next command in the queue.

### 6.10.1 The Command Preparation Routine

The command preparation routine *cprep* is called from *execute* after the current command has been set up. Its function is to ensure that there is a smooth transition between the current command and the next command.

It can be seen from figure 6.5 that if there is a reduction in the demand velocity between the current command and the next command, the *cprep* routine initiates a ramp down of speed so that the AGV is at the new

lower demand speed at the terminating position of the current command. If the next demand speed is higher than that of the current command, the speed change is made at the start of the next command (i.e. when *execute* increments the command execution pointer). This is a sensible approach because lower speeds are likely to have been chosen for a more critical control or tighter radii.

If a reduction in speed is required at the end of a command segment, the distance required to ramp down can be calculated from the two demand speeds and the nominal ramp rate,  $r_0$ , as

$$x_r = \int_0^{\frac{V_{d1}-V_{d2}}{r_0}} (V_{d1} - r_0 t) dt \quad (6.5)$$

which gives:

$$x_r = \frac{V_{d1}^2 - V_{d2}^2}{2r_0} \quad (6.6)$$

This ramping distance is then passed to the distance monitoring routine *dmon*. This is described in detail later but briefly it monitors the distance left to the end of the current command segment and returns when the value of this is equal to the value passed to the routine. Thus when the distance monitoring routine returns, the distance left to the end of the line segment is (approximately) equal to the required ramping distance and a ramp down to the demand speed of the subsequent command is initiated.

Whether a ramp down has been initiated or not, the end of a curve or line segment must be detected before control ultimately returns to *execute* which increments the command input pointer. To facilitate this, a call to the distance monitoring routine is made with the value 0.0 passed to it, as is shown in figure 6.5.

### 6.10.2 Distance Monitoring Routine (*dmon*)

This routine monitors the distance remaining in the current command segment and returns when this is equal to the distance value passed to the routine.

The distance remaining in a command segment is calculated in the routine *localpos*, which is part of the control interrupt software, from the local  $x$  coordinate and the length of the command. It should be noted that since these calculations employ squared values to determine the distance left to the end of a segment (equation 5.22), a magnitude is actually calculated. Thus there is a potential problem when the end of a segment is being monitored because the magnitude of the remaining distance can never be less than zero. This particular case is trapped in *dmon* by checking that the distance remaining in a segment is always decreasing. If an increase is detected, then *dmon* will exit and return to *cprep*.

As well as distance monitoring in the loop of *dmon*, a call to a screen updating routine called *ptvar* is made. This does two things. Firstly it prints the current value of all the important global variables in columns on the terminal screen. These include the global position  $(X, Y, \theta)$ , demand speed, base speed, acceleration, and the ultrasonic distance measurements. Additionally, it checks to see a key on the terminal has been pressed. If it has, then a 'run abort' has been requested and an error routine is called. This stops the AGV dead and switches off the power to the stepper motors. Subsequently, the AGV status, which is a list of global variable values, is printed, and a trap instruction is executed which causes the AGV software to exit and thus returns control to the on-board microcomputer's monitor.

This section concludes the description of the background routines. Subsequent sections describe both the low level control and data processing interrupt software and the high level timing interrupt software. Since the

geometrical transformations employed by the low level interrupt software were described in the previous chapter, the following sections concentrate on aspects of software structure only.

### 6.11 Odometry Routine (*odom*)

The odometry routine, *odom*, is the first major routine to be called after a low level control interrupt has occurred. Its function is to update the global position coordinates  $(X, Y, \theta)$  of the AGV by reading the count values provided by the optical shaft encoders and transforming the incremental rotations of each drive wheel into incremental changes of  $X, Y$  and  $\theta$ .

There are two types of calculation for the changes in the  $X$  and  $Y$  coordinates: one for travelling along a straight line, the other for a curve of some constant radius. The former is implemented if there is no change in heading and simply resolves the length of the straight line into incremental  $X$  and  $Y$  components using the current heading and AGV direction. If there is a change in heading (equation 5.4), the calculations assume that the change occurred along a constant radius (equation 5.6). This is justified if the change ( $\Delta\theta$ ) is small. Next the magnitudes of the orthogonal components relative to the AGV heading are calculated. These are subsequently resolved into their  $X$  and  $Y$  components and combined according to equation 5.19.

The fundamental odometric calculations, which are described by the equations derived in section 5.6, actually reside in a subroutine called by the main odometry routine. This is because a number of variations in the measured drive wheel rotations are employed to maintain an estimate of odometric error bands, as was described in section 4.4.2. Three variables are passed for each subroutine call: the most appropriate value within the current band of heading error, and the two variables  $(\Delta d_1, \Delta d_2)$ , which

describe the particular variation in conjectured path. For each subroutine call, the global array representing odometric position and its error bands is updated accordingly. (One dimension of this array is used for each of the coordinates  $X$ ,  $Y$ , and  $\theta$ , and one for the three values: low error band, measured odometric location, and high error band.)

## 6.12 Data Processing Routine (*datproc*)

After the odometry routine exits, local position coordinates  $(x, y, \theta')$  are generated from the updated global position  $(X, Y, \theta)$  and the local frame of the current command  $(X_0, Y_0, \theta_0)$  by a call to *localpos*, which contains the transformations described in section 5.7. Subsequently, the data processing routine (*datproc*) is called. The function of this routine is to examine collected ultrasonic range measurements and to process them so that useful information can be extracted for position correction and, in the case of the "wall following" mode, for direct control.

In order to provide the user with on-line information concerning the state of data processing, a method of generating diagnostic messages on the terminal screen is employed. This method requires *datproc* to pass back a symbolic constant when conditions arise which deem it impossible to make a position correction (see figures 6.6a and 6.6b). More specifically, the data processing routine is called as *status = datproc()*, where *status* is a global variable used to index an array of diagnostic messages in the background monitoring routine *execute*.

Since *datproc* is called from the low level interrupt service routine which occurs once every 100mS, and the frequency of ultrasonic measurements is 100Hz, approximately 10 raw ultrasonic readings are available from each enabled sensor at each routine call. High level interrupt software stores the

distance measurements from each separate sensor in its own array in a rotating fashion. These arrays can store fifty distance measurements which allows the data processing routine to examine data up to half a second before *datproc* was called. Each time an ultrasonic array becomes full, data at the start of the array begins to be overwritten. In order to determine which data in the ultrasonic array is new, the data processing routine can access a pointer in the high level interrupt software which points to the array position at which the last distance measurement was entered. In this way valuable memory space is not used up and, by comparing the current pointer value with its value on the previous entry to *datproc*, instances of 'wrap round' can be detected and the appropriate distance measurements can be read from the array.

A flow chart of the data processing routine is given in figures 6.6a and 6.6b. When this routine is called, the command type is first examined and the routine simply returns if the current command type is a curve follow or stop. (A useful modification to the *datproc* routine would be to allow averaging of range measurements whilst the vehicle is in a wait state.) If the vehicle is line following, the routine determines whether the AGV is in the vicinity of a correction board by examining data in the command queue array. This data refers to the position of an observable correction board in the local frame of the current command and would have been inserted by taking the 'board search' option in the main menu.

Next, the distance that the vehicle has travelled parallel to a correction board since the ultrasonic data arrays were last read is determined using the current and previous local  $x$  coordinates. This is checked against certain limits in order to reject instances when the AGV is travelling so slow that the small distances generate unreliable gradient measurements in the line fitting process. In addition, the first call of *datproc* after a curve follow

command gives a meaningless value to the parallel travelling distance since the previous recorded 'along board' distance is that of the previous correction board. Such instances of non-contiguous line fitting must also be rejected.

The routine then enters a loop, which it executes twice, in order to capture the new data provided by the front and rear ultrasonic sensors. This routine accesses the appropriate ultrasonic data array pointer for sensor  $k$  and thus determines the number of new range measurements provided by that sensor and the indices within the array between which these new values may be accessed. \* In the 'capture' loop of *datproc*, a check is made to ensure that there are less than twenty and more than two readings provided by that particular sensor. This is required to protect the integer regression calculations from either overflow or divide by zero.

After the new data provided by each sensor has been captured, the main data processing loop is entered. Initially the routine *filter* is called which reads the appropriate range measurements from the data storage array and digitally filters them to provide a data set with reduced noise. In addition, singular spurious readings are eliminated by appropriate adjustment of the filter constant. This process of filter constant adjustment is described in chapter 8.

The next routine called in the processing loop loads the two arrays upon which the regression equations operate. These contain the set of filtered range measurements,  $y_{k_i}$ , and their associated local  $x$  coordinates,  $x_i$ . A local  $x$  coordinate is associated with each new range measurement by dividing the change in  $x$  over a control interval by the number of results collected in that same interval. This is valid since base velocity is constant over any such interval and, if the vehicle is travelling almost parallel to a correction board,

---

\* It should be noted here that all of the specifications concerning the research vehicles sensors are organised into an array of 'c' structures. Thus capture updates `sen[k].n`, `sen[k].istart` and `sen[k].iend`.

the effect of turning curvature can be neglected. The routine then subtracts the minimum value from each data set to prevent integer overflow in the regression summations. Obviously this does not affect the gradient calculation, and these minimum values are simply re-introduced in the calculation for the line constant,  $b_0$ . In addition, the routine records the local  $x$  position of the maximum in the data set of  $y_{k_i}$ . This is required in order to implement a line refitting procedure.

After the appropriate arrays have been loaded, a routine called *regression* is called for a particular data set. The function of this routine is to calculate the gradient ( $b_1$ ) and the constant ( $b_0$ ) of a line ( $y = b_0 + b_1x$ ) generated from a least squares line fitting procedure. It can be seen from figure 6.7 that embodied in the regression routine is a call to a routine that calculates the residuals for each point in current data set of a particular sensor. The maximum residual in the data set is thresholded in order to detect instances of poor fit in the line fitting process. It is shown in chapter 8 that such instances occur when a sensor passes a notch on the correction board. If it is found that the modulus of the maximum residual is above 1cm, a line refit procedure, illustrated in figure 6.8, is called, which splits the current data set at the maximum of that set, and thus calculates new indices with which to access the data storage array. Subsequently, the regression arrays are reloaded and the regression routine is recalled in order to generate a new equation for the fitted line. Thus there is a recursive arrangement in the software whereby the routine *regression* may call *refit* and vice-versa. Note that there are two precautionary measures in the line refitting process. Firstly, in the regression routine, *refit* can only be called once. Although such a constraint may generate poor fitting lines for a sensor passing a more complex shape than one of the correction board notches, a single level of recursion was found to be all that was required. A second precaution is

contained in the *refit* routine. Here a check for the maximum range measurement being at the end of the regression line captures cases where the fit is 'good' yet a large maximum residual is detected. This can sometimes occur when the gradient of the fitted line is very large which exacerbates the inaccuracies of integer arithmetic.

After the line fitting process has terminated, a notch detection routine (*notch*) is called. The function of this routine is to determine whether a particular sensor is opposite a notch. If a notch is successfully detected, the routine calculates the position of the centre of the notch according to the AGV's current estimate of local position. It should be noted here that if a 'wall follow' command is executed without the notch detection algorithm running, the AGV can be seen to weave around the notches as the control tries to keep the vehicle both parallel and a constant distance from the correction board. Thus, in addition to position correction, notches must be detected reliably in order that they can be filtered as far as the control is concerned, when the vehicle operates in the 'wall follow' mode. A description of the algorithm used to detect notches is best illustrated using collected results. This is left until chapter 8.

After the notch detection algorithm, the data processing routine calls a routine called *correct* which is illustrated in figure 6.9. In the first part of this routine, a check is made to determine whether either sensor is currently opposite a notch. If this is the case, then no position correction can be made and the routine passes back a symbolic constant to inform the data processing routine. If neither sensor is deemed to be opposite a notch, local position  $(y, \theta')$  is calculated using the terminating points on the fitted regression lines of the front and rear sensors. These two local position coordinates are then checked against the current odometric error bands to ensure that the processed ultrasonic measurements are suitable for position correction. Subsequently,

the routine can correct either one two or three of the global variables  $X$ ,  $Y$ , and  $\theta$ . Whenever any of the three global coordinates is corrected, the error band of that coordinate is reset according to the estimate of error in ultrasonic position measurement. (This is detailed in chapter 8.) If a notch has just been detected, the current odometric error band in the local  $x$  direction is used in conjunction with data in the notch array to determine the true position of the notch. Thus local  $x$  position can be calculated and, using equation 4.3, a full position correction can be made. However, examination of the notch array may find that there is no notch with its local  $x$  coordinate within the current odometric error band, thus the detection of the notch must be deemed to be spurious. If this is the case, or if a notch has not been detected in the first place, then, generally, only vehicle angle can be corrected since there are non zero elements off the matrix diagonal in equation 4.3. In the specific case when the vehicle's demand path is parallel to either the global  $X$  axis or the global  $Y$  axis, the coordinates  $X$  and  $Y$  become independent thus enabling either  $(X, \theta)$  or  $(Y, \theta)$  to be corrected.

### 6.13 Base Velocity Control

After the data processing routine exits, a routine to update base speed is called. This routine updates the base speed by a fixed amount determined by the ramp rate,  $r$ , so that it is closer to the current demand speed. This demand speed may be the demand speed of the current command, the demand speed of the subsequent command, or an imposed demand speed of zero in cases when an obstacle is detected. In the case when base speed is equal to demand speed, no change is made, and when the ramp rate causes base speed to overshoot demand speed, base speed is set to demand speed. Since this routine is entered every 100mS, a staircase ramping effect is produced

which is ostensibly smooth due to the small time interval.

A flowchart for the routine is shown in figure 6.10. Firstly demand speed is compared with base speed to see if deceleration is required. If an obstacle has been detected in the path of the AGV, the demand speed will have been set to zero, thus initiating the ramp down. In this special case, the ramp rate is calculated using the current base speed and the measured range to the obstacle.

It can be noted from the figure that the *ramp* routine operates differently when the AGV is accelerating from when it is decelerating. When accelerating, the base speed is changed by a constant amount which is defined by a value of nominal ramp rate,  $r_0$ , entered at the start of a test run. However, if a reduction in base speed is required, the ramp rate is calculated as:

$$r_d = \frac{(V^2 - V_d^2)}{2x_r} \quad (6.7)$$

which is simply equation 6.6 rearranged. The ramping distance  $x_r$  is the distance left to the end of a command segment. This has already been calculated in the routine *localpos* from the local  $x$  position and the length of the current command segment.

Equation 6.7 effectively calculates the amount by which the base speed must be reduced at each interrupt in order that it will be equal to the demand speed when the ramping distance becomes zero. In other words, it ensures that the AGV will be at its new lower demand speed at the end of a command segment which is particularly important if the AGV has to stop at some specified position.

It can be seen from figure 6.10 that before the calculated value of  $r$  is applied to reduce the base speed, it is passed through a limit block. The upper limit prevents excessive changes in speed from stalling the motors and

the lower limit prevents truncation of the ramp rate to zero, in which case the AGV could never reach the demand speed.

The routine *ramp* also has the ability to switch the power to the stepper motors on and off. If, whilst decelerating, the base speed reaches zero, the stepper power is switched off. For example, this may be when the AGV has reached the end of its command queue and new commands are being entered by the user. When acceleration is subsequently required, a check is made which will cause the stepper power to be switched back on if the AGV speed is found to be zero (see figure 6.10).

The following section is concerned with deriving a turning curvature from the current estimate or measurement of position.

#### 6.14 The Software Controller

After base speed has been updated by *ramp*, a demand turning curvature is derived by a call to the routine *controller*. In this routine resides the the software controller which switches between two sections of code in accordance with the current command type. If the current command is a 'wall follow', perpendicular distance and heading errors in a local frame are taken as those derived in *datproc* from processed ultrasonic distance measurements. Otherwise, the perpendicular distance to the imaginary line it is following is simply the  $y$  coordinate of the AGV position in its local frame and the heading error is the local frame heading,  $\theta'$ . These values have already been generated by the routine *localpos*.

As well as deriving a demand turning curvature, the routine checks that both distance and heading errors are within reasonable values (1m and 45 degrees). If this is not the case, the AGV stops dead and aborts its run.

## 6.15 Limiting the Rate of Steering

When the controller has calculated a demand turning curvature, demand wheel speeds are calculated in the routine *transform* as a function of this curvature and the required base speed generated by *ramp*. From the equations in section 5.8, demand speeds can be written as:

$$V_1 = V \left( 1 + \frac{W_b \kappa}{2} \right) \quad (6.8)$$

$$V_2 = V \left( 1 - \frac{W_b \kappa}{2} \right) \quad (6.9)$$

These demand velocities are not necessarily the velocities that the drive wheels assume because the change in stepper speed at each interrupt interval is sometimes limited to prevent stalling of the motors.

In order to make efficient use of the stepper motors, this degree of limiting should be variable and in each case just sufficient to prevent a stall. This optimum degree of limiting is dependent on the torque-speed characteristic of the motor itself and the extra torque demand on the motors which would, in the worst case, be developed by the AGV moving onto a tighter radius at a higher speed. The limit would have to be calculated with respect to the linear and angular accelerations of the vehicle that can be supplied by the torque capacity of the steppers at a given speed. This is clearly a problem which is specific to the research AGV dynamics and the stepper motor torque-speed characteristic. \* Therefore, in order to simplify the problem and, to some extent, mimic a conventional steering mechanism, it was decided to have a fixed limit on the rate of steering which would be determined empirically. This method was chosen firstly because the rate at

---

\* Further complications arise since, when the vehicle is turning, the two stepper motors operate on different parts of the characteristic.

which base speed can increase is already limited to the nominal ramp rate entered at the start of a test run and, secondly, the angular accelerations caused by step changes in demand curvature are likely to produce the largest demand torques on the stepper motors.

The AGV software provides a choice between two methods of curvature rate limiting. In the first method, the amount by which the velocity of a drive wheel relative to the base velocity,  $V_{rel}$ , can change over a control interval is limited to a pre-set value. By expanding either equation 6.8 or 6.9, this quantity can be written as:

$$V_{rel} = \frac{VW_b\kappa}{2} \quad (6.10)$$

If the demand change in  $V_{rel}$  is greater than some fixed value (typically 10 sps), then  $V_{rel}$  is only allowed to change by that fixed value. This effectively sets a maximum limit on the rate at which turning curvature can change (i.e. the *rate of steering* is limited).

The result of limiting  $V_{rel}$  as opposed to the change in curvature itself is that higher base velocities generate a tighter limit on turning curvature which effectively produces constant limiting of the vehicle's angular acceleration.

A second method of curvature rate limiting, allows the vehicle to assume a given proportion of the demand change in curvature over a control interval. The advantage of this method over the first one is that its effect on the control is linear and predictable and, therefore, can be included in the control analysis which is given in the following chapter.

## 6.16 The Wheel Speed Output Routine (*wspout*)

Whether curvature rate limiting of any form occurs or not, individual drive wheel speeds are passed to the wheel speed output routine (*wspout*),

illustrated in figure 6.11, which checks for 'tight turns'. Tight turns occur when the radius of the turn is small compared to the base speed and thus the calculated speed of the slower wheel is negative. This means that the direction of the wheel associated with this value must be reversed. The *wspout* routine is fully independent in that it detects and deals with such instances of 'tight turn'. If a tight turn is not detected, the routine resets wheel directions according to travel direction before actually executing the wheel speed changes. This is necessary because a tight turn may have been executed on the previous call of *wspout*. Finally, after setting wheel directions accordingly, assembler routines are called which load VIA timer registers with values that effect the calculated wheel speeds.

This section concludes the description of low level control interrupts. The following section is concerned with the high level interrupts generated by the ultrasonic interface boards.

### 6.17 High Level Interrupts

The interrupt service routine *usisr* is an interrupt service routine that starts and stops timers on the basis of received ultrasonic signals. These signals, indicating when a burst of ultrasound is transmitted and when the echo of the same burst is detected, are on enabled lines of a VIA, which can interrupt the processor on the highest permitted level (level 4).

The processor can determine which signal caused the interrupt by examining the VIA interrupt flag register. A flow chart for handling two sets of ultrasonic transducers with a common start signal is shown in figure 6.12. Initially the registers a0 and d0 are moved onto the stack because these are employed in the routine. Next the base address of the VIA chip is loaded into register a0 so that the appropriate VIA registers can be addressed as this

plus an offset. The routine then determines which of three signals interrupted the processor. It is possible that both *STOP1* and *STOP2* occurred (almost) simultaneously in which case only *STOP1* would be processed since this is detected first. However, as is shown in the flow diagram, only the *STOP1* flag is cleared and so the routine would immediately be re-entered after it returned allowing the second signal to be processed. The static variables called stop ('stop1' and 'stop2') were introduced to prevent confusion due to multiple reflections of the ultrasonic signal. A measurement is only made if the appropriate stop variable is equal to zero, which means only the first pulse after the start signal is accepted. Timers are never stopped, they are just read and moved into the appropriate storage array. This means that the common start signal only needs to start one timer running. In the final part of the routine, the saved values of a0 and d0 are retrieved from the stack before the processor returns to its previous task.

Other noteworthy points, not shown in figure 6.12, are

(1) As well as filtering out multiple reflections, the routine ignores readings below 10cm, since these can be sometimes caused by noise coupled from the transmitter line to the receiver line.

(2) Timer values are stored in a rotating fashion in global arrays which can be accessed by the control interrupt software.

(3) The software supports five sets of ultrasonic transducer rather than two. The ultrasonic sensor at the front of the vehicle is used as a 'soft bumper' to detect obstacles in its path. If an obstacle is detected within 50cm of the AGV, a warning horn is switched on, and a status flag is set which can be accessed by the low level interrupt software to initiate a deceleration, as was described in section 6.13. Unless the AGV reaches the end of its current command, the AGV will stop approximately 10cm from the obstacle and the warning horn will remain active until one second after the

object has been removed. After the horn is switched off, the obstacle status flag is cleared, which causes *ramp* to accelerate the vehicle to the demand speed of the current command.

The following section is concerned with collecting and analysing results.

### 6.18 The Method of Results Analysis

In order to develop both data processing and control algorithms and in order to make an assessment of the performance of the research AGV, a method of collecting and analysing results had to be devised. This method involves four steps; the first two steps are taken in the AGV software, the third involves running a program provided by the Durham University Microprocessor Centre, and the fourth uses a general purpose program which has been developed on a personal computer.

#### (1) Selecting a mode of data collection.

The selection of a data collection mode is made at the start of a test run along with a data collection interval,  $n$ , and an *overwrite* mode. A specific data collection mode causes a specific set of ten variables to be saved every  $n$ th control interval by passing their values to the data collection routine *datcol*. This routine loads a single row of a general two dimensional data array with the values passed to it. If the *overwrite* mode is *ON*, *datcol* will start refilling the data array once it is full, otherwise it simply returns causing the passed data to be discarded.

#### (2) Saving the data array to a memory buffer.

The option of saving the collected data to a memory buffer is available at the end of a test run. If this option is taken, the *buffsave* routine initially copies a number of important global variables, the command queue array, and arrays of characters which are a set of labels, each label referring to a

particular variable in the data collection set, to the memory buffer before the data array itself is copied.

(3) Transferring the memory buffer to a disc file.

In order to transfer the memory buffer to disc using the on board disc drive, a separate program must be loaded and executed. This was provided by Durham University Microprocessor Centre.

(4.) Analysing the data on a personal computer.

A program which runs on a personal computer has been specially written to read the structure of the stored data file and provide general graph plotting and data analysis routines. On running this program, file specifications are displayed along with the command queue array and the labels associated with the stored data. This informs the user of the exact conditions under which the experiment was carried out and indicates what data has been collected. The program permits any two of ten data sets to be plotted against each other, or any data set to be plotted against a time scale. It also allows two graphs to be plotted against the selected abscissa. In addition, the program can zoom in on a section of a plot, employ a variety of line types, automatically scale and label axes, and automatically provide a title. If the program is being employed to plot the AGV's global position, the demand path can also be plotted, since these specifications are available in the command queue array.

## 6.19 Software Documentation

In order to keep the text concise, software listings are not included in this thesis. However, documentation of the AGV software and the data analysis software is available, along with instructions on how to operate the AGV research rig, within the School of Engineering and Applied Science,

## 6.20 Summary

This chapter has described a software package which is composed of a flexible menu-driven user interface and the interrupt based sensing and control algorithms required to implement a low level guidance scheme. In addition, the software required to implement data logging and data analysis has been briefly described.

The software package has proven to be flexible in that it allows vehicle routes to be created, or loaded, and edited with ease. In addition, it allows correction boards to be either added to, or moved to different positions within, the vehicle's workspace. When used in conjunction with the research vehicle and the data analysis program, it constitutes an efficient tool with which to conduct autonomous vehicle research.

The final two chapters of the main text present results collected from the AGV research rig during a variety of test runs. The following chapter is concerned with the design, analysis, and performance of a steering control system whilst chapter 8 details aspects of data processing and the overall performance of the research vehicle.

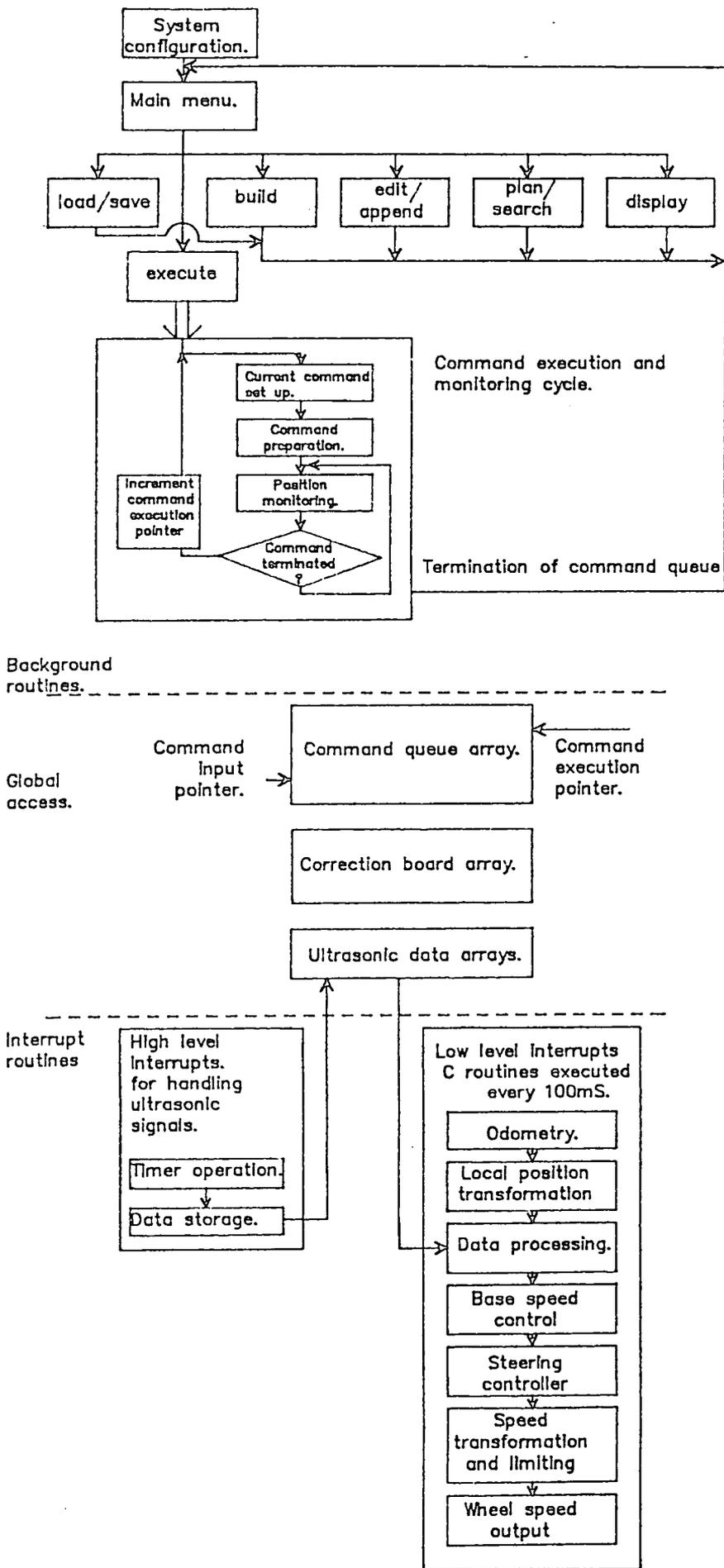


Figure 6.1 A software overview

0 1 2 3 4 5 6 7 8 9

Wall follow.	Type (1)	Direction	Demand speed.	Demand heading	Xterm	Yterm	$X_w$ Distance of line from wall (or cb)	—	—	—
Line follow.	Type (2)	Direction	Demand speed.	Demand heading	Xterm	Yterm	$X_w$ Distance of line from wall (or cb)	$X_0$ Starting local x pos for corrections	$X_1$ Terminating local x pos for corrections	Distance offset.
Curve follow.	Type (3)	Direction	Demand speed.	Demand heading (At end of curve)	Xterm	Yterm	Demand curvature.	Angle of curve segment.	$X_c$	$Y_c$
Stop.	Type (4)	Direction (of previous command)	Demand speed. (0)	Demand heading	Xterm	Yterm	n seconds wait	—	—	—

← Common parameters. →

<0> -Null type signifies end of queue

Figure 6.2 The command queue array

MAIN MENU.

0. Quit

Command queue array.

Correction board array.

1. Load

8. Load

2. Save

9. Save

3. Edit

10. Edit

4. Append

11. Append

5. Build

12. Build

6. Display

13. Display

7. Path plan

14. Board search

15. \*EXECUTE\*

\*\*\*No command queue loaded...use load or build options.\*\*\*

Enter option: \_

Figure 6.3 The main menu

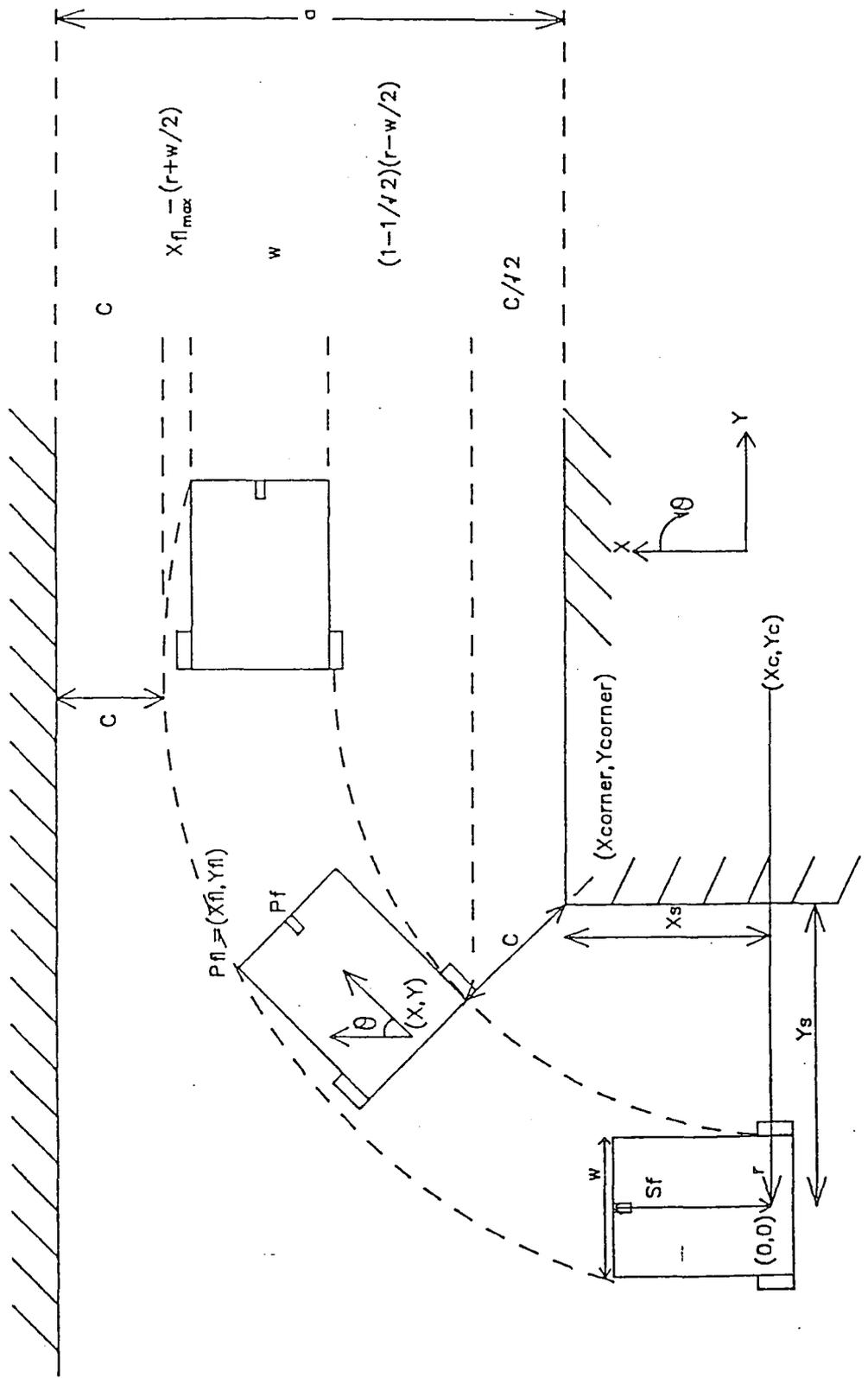


Figure 6.4 Calculation of command parameters for a corner

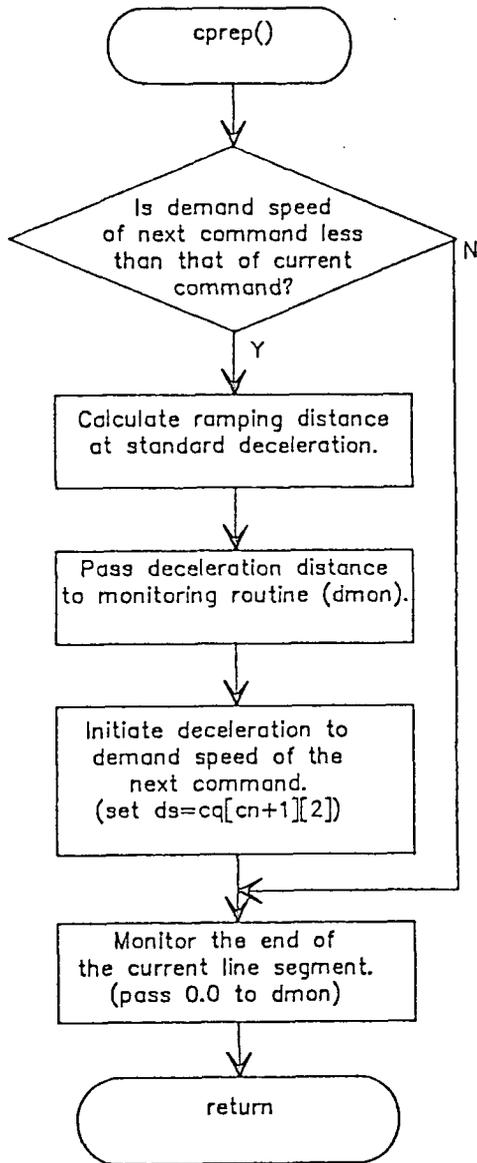


Figure 6.5 Command preparation

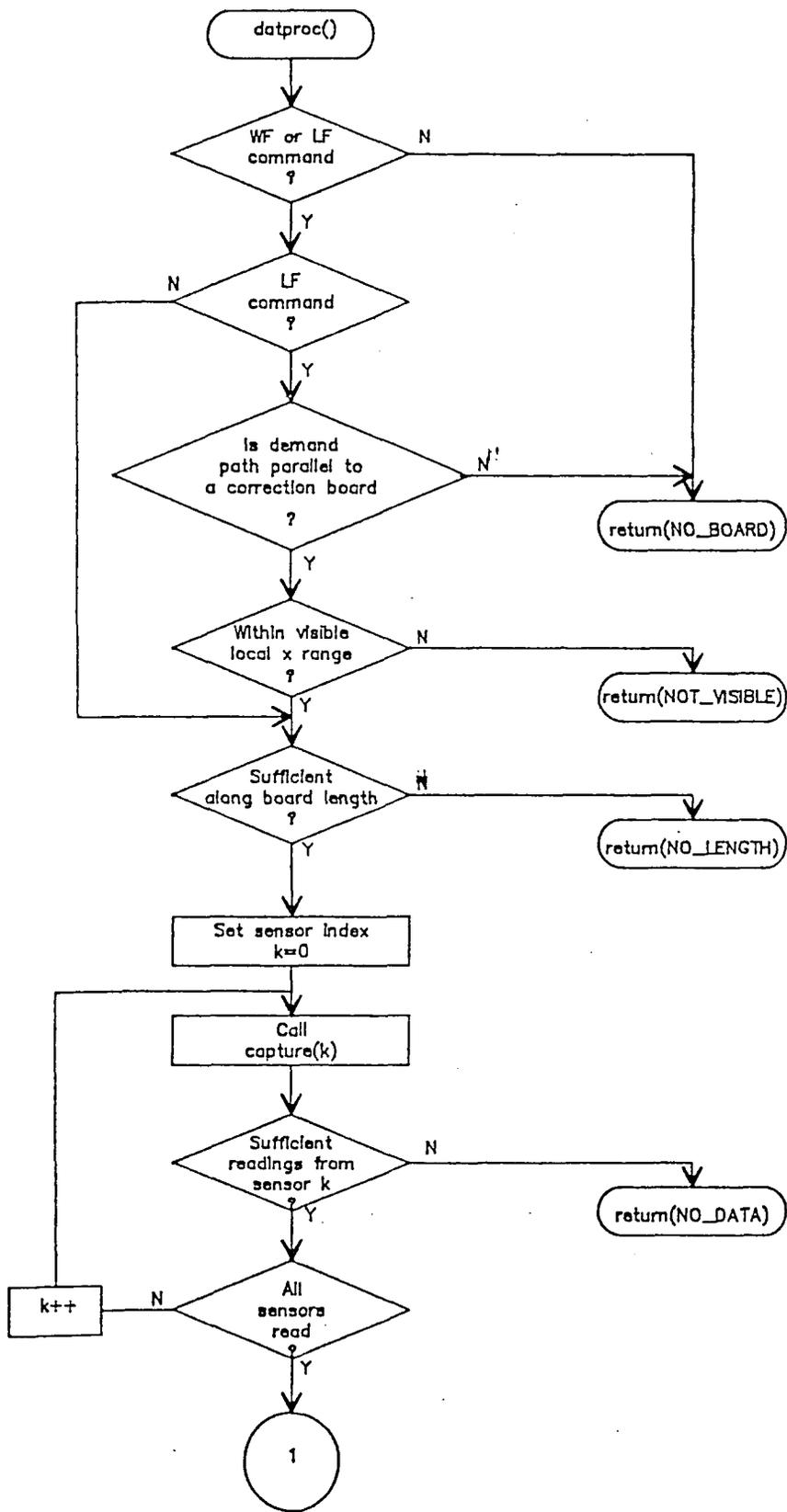


Figure 6.6a The data processing routine

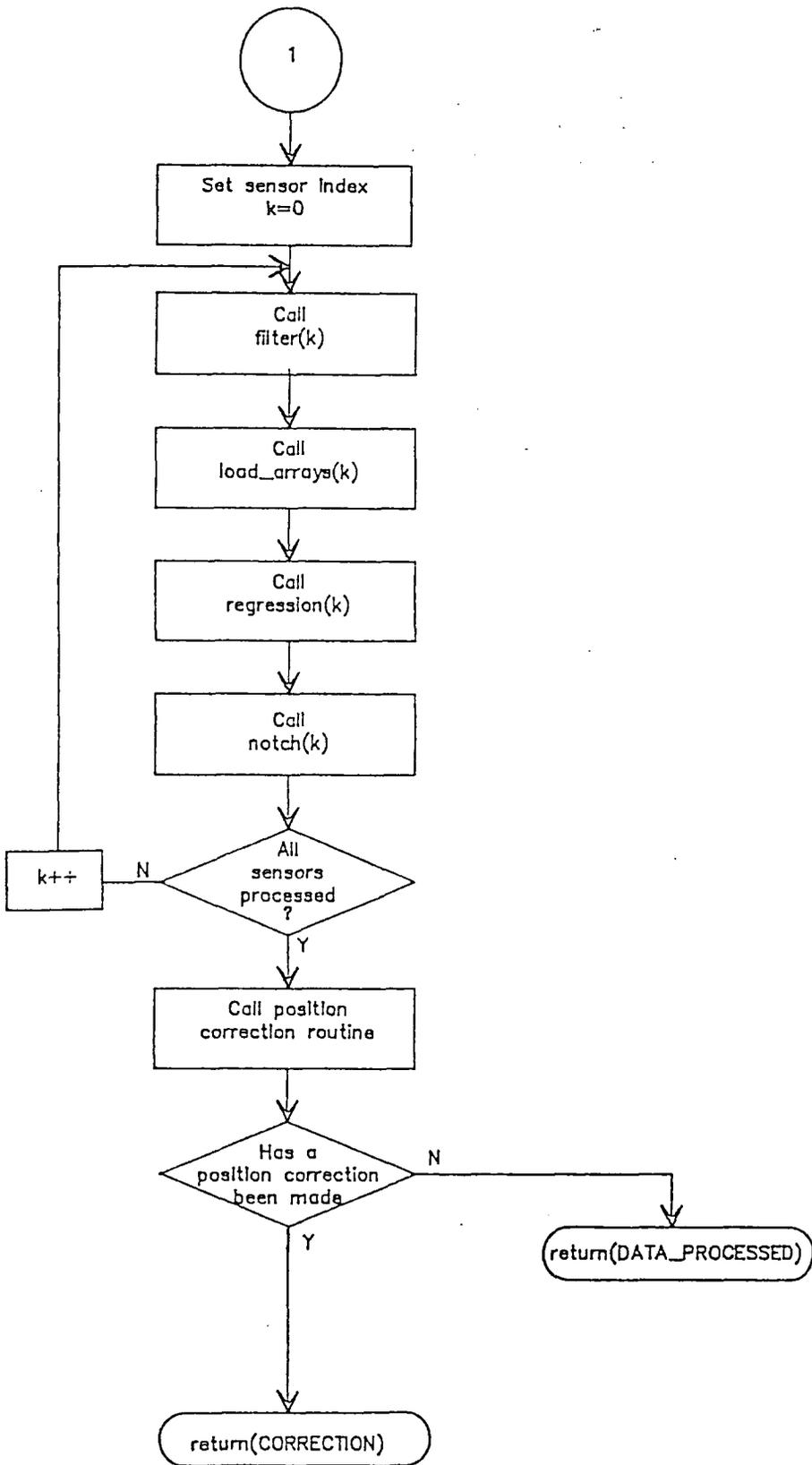


Figure 6.6b The data processing routine-continued

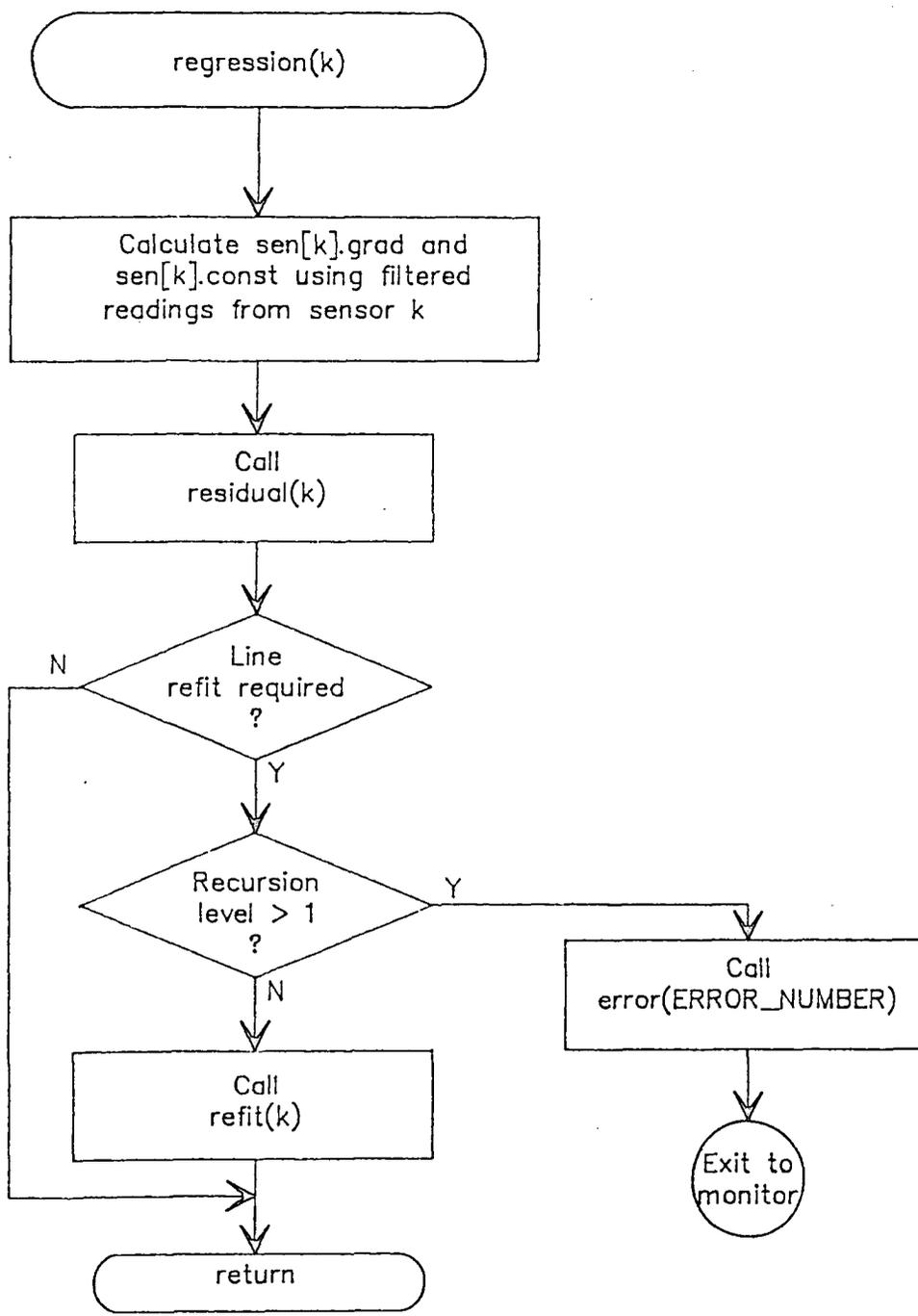


Figure 6.7 The regression routine

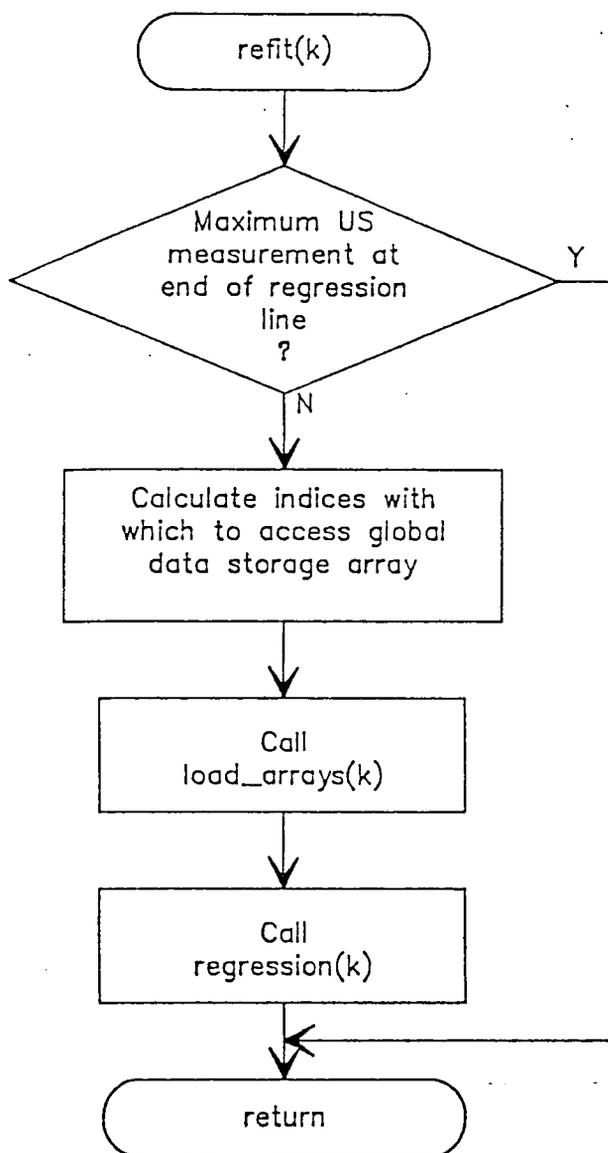


Figure 6.8 The line refit routine

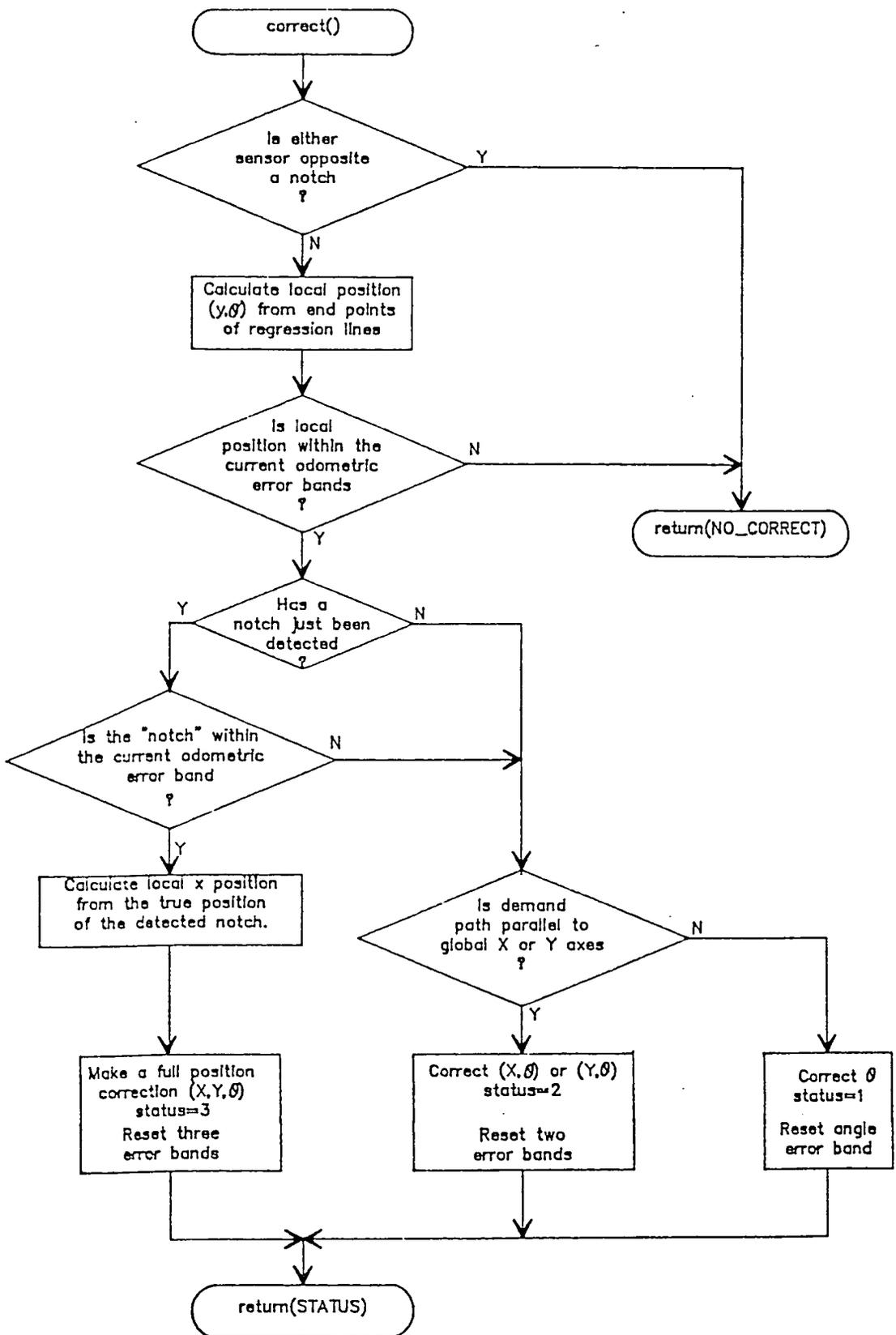


Figure 6.9 The position correction routine

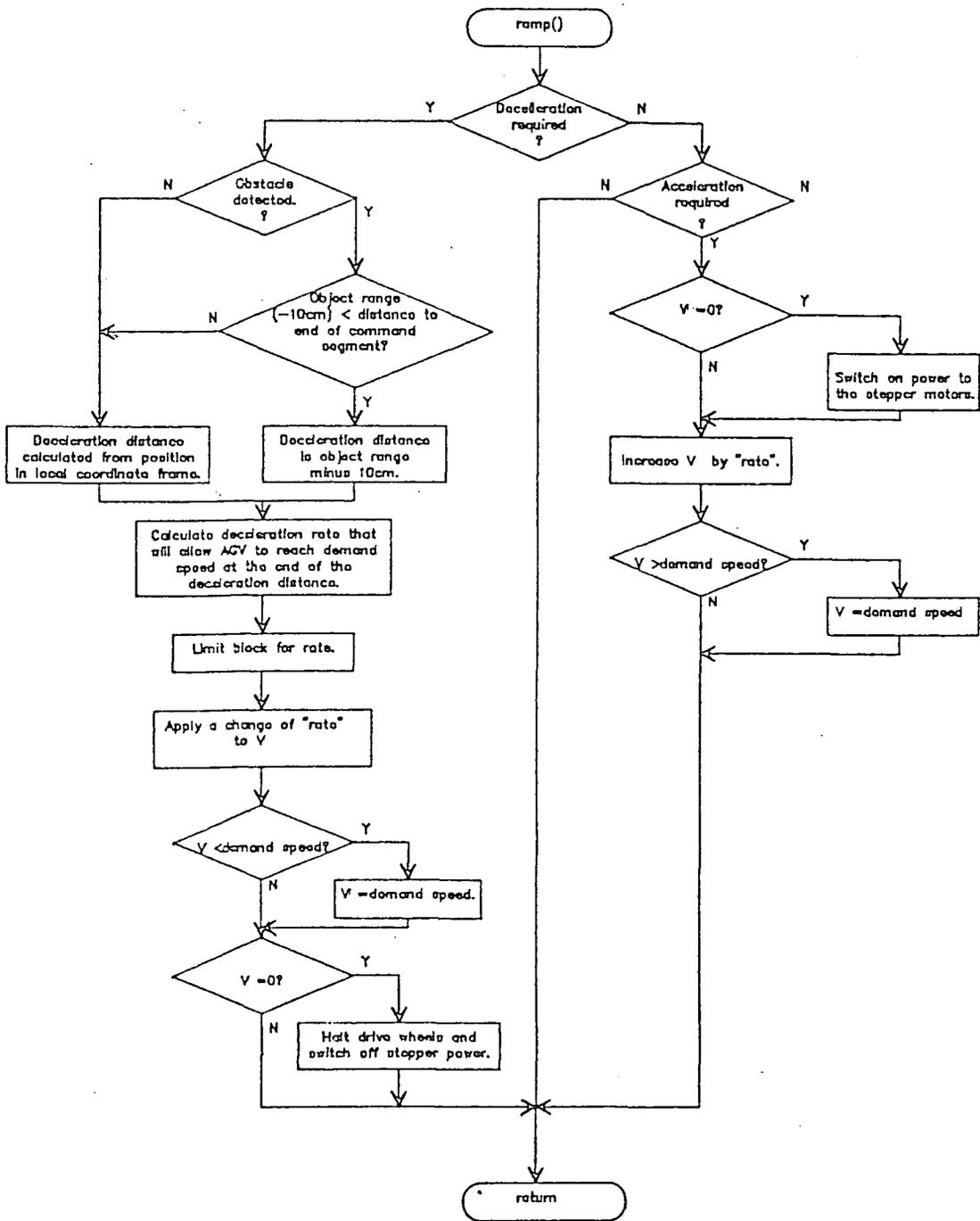


Figure 6.10 Base velocity control

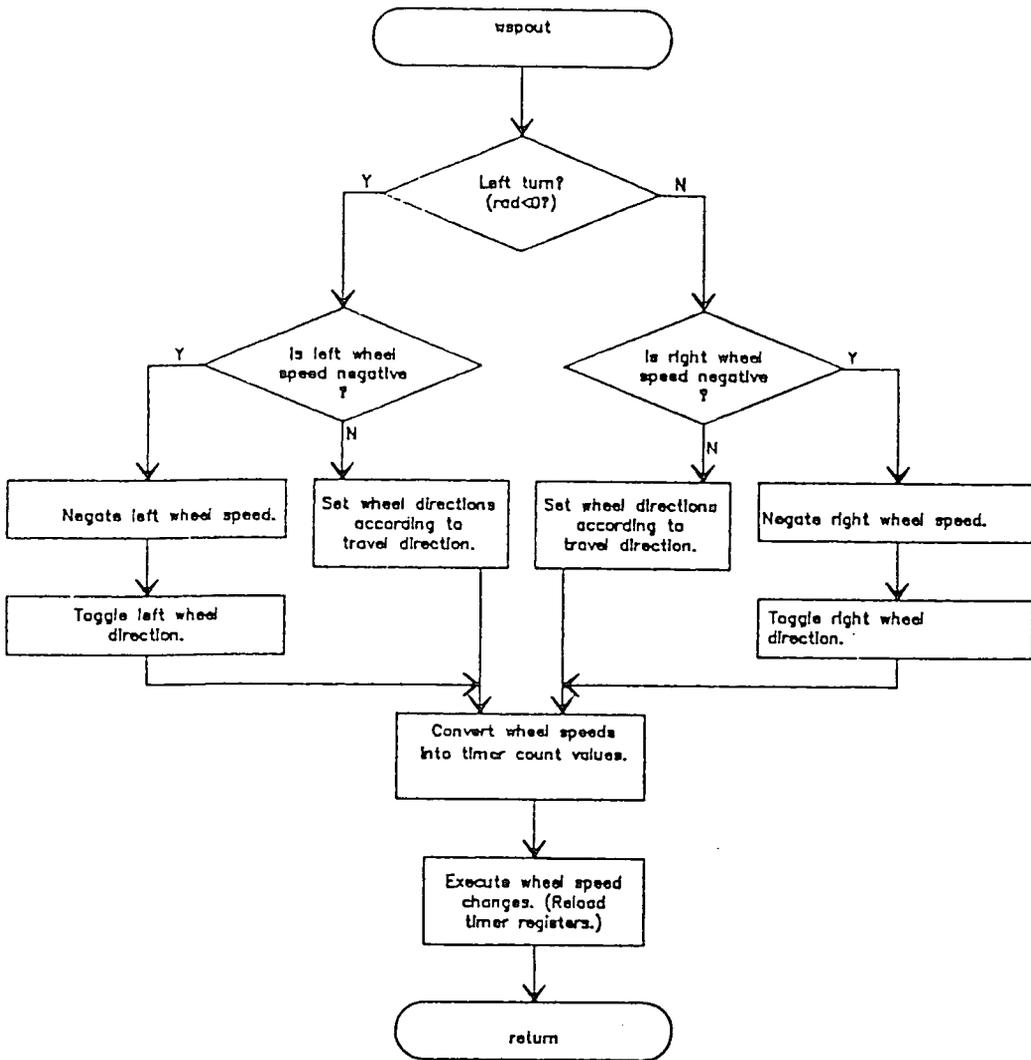


Figure 6.11 The wheel speed output routine

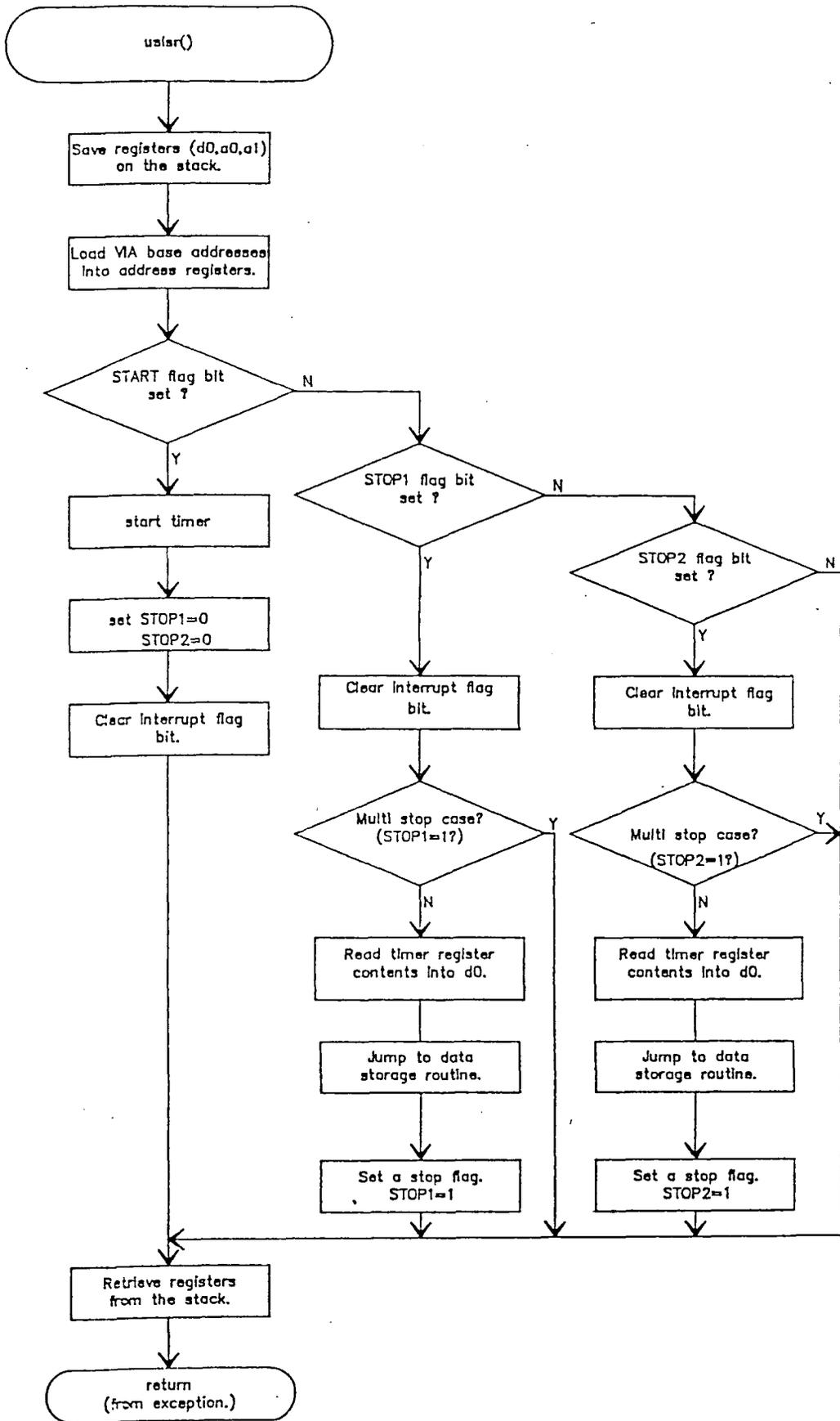


Figure 6.12 The high level interrupt timing routine

## CHAPTER 7

# DESIGN, ANALYSIS, AND PERFORMANCE OF THE AGV STEERING CONTROL

### 7.1 Introduction

The performance of the AGV's control system in steady state is required to determine how close the vehicle stays to its designed course as it moves through the environment. Since position corrections periodically subject the vehicle to step changes in apparent perpendicular distance and orientation relative to its demand path, a knowledge of the system's response to step output disturbances is also required.

In the first part of this chapter, the control method outlined in section 5.4 is defined more rigidly and subsequently analysed in a state space format. This analysis is required in order to relate second order criteria such as damping factor and rise time of the system to the parameters of the control system. Thus the control parameters can be chosen to give critical damping and the best realisable rise time. Further sections in this chapter describe modifications to the basic control that are required to

- (1) Improve curve tracking performance.
- (2) Make the vehicle's rise time, to a large extent, independent of velocity. This has the effect of making use of the stepper drive's increased torque at lower base velocities.
- (3) Prevent stalling of the stepper motors when either large position corrections occur or when the vehicle enters a command with a high curvature.

A final section is used to determine the degree of approximation in analysing a sampled control system in the continuous time domain, and the amount by which the sampling interval may be expanded without excessively depreciating the control system's performance.

## 7.2 An Analysis of the Basic Steering Control System

Several control studies have been carried out on a number of different autonomous vehicles. For example, automated passenger vehicles in which conventional, four wheeled, front steered cars have been made to follow a guide-wire have been reported [FENTON et al, 1976], [CORMIER and FENTON, 1980]. Stephens [STEPHENS et al, 1983] has described a scheme whereby incremental reference vectors are moved at demand speed through a defined trajectory to form a kind of 'tow-along' procedure. Steer [STEER, 1985] investigated wall following using a single ultrasonic sensor on the front corner of a vehicle with front wheel steering. In this work, a simple model was examined in which curvature was made to be proportional to the difference in measured distance and a demand distance. Kim [KIM, 1987] has presented a theoretical optimal (and suboptimal) proportional-plus-integral controller for a wire-guidance vehicle. Hongo [HONGO et al, 1987] has described a control method whereby the accelerations of the left and right drive wheels are controlled such that the vehicle's state (position, orientation, and velocity) will agree with that of the current command after a given time period. In this work, accelerations were used as the controlling variables to avoid sudden changes in velocity. In comparison to most of these systems, the steering control employed in this research is simple on account of the nature of the vehicle's drive system and the effectiveness of the basic control rule.

The research vehicle's drive system employs independent stepper motor

drives. The behaviour of a stepper motor is such that it either adopts a speed in correspondence with the pulse rate sent to its translator or stalls due to excessive torques developed from acceleration and loading. The implication of this is that, in the absence of stalling, mass, rotational inertia, and other vehicle dynamics are eliminated from the equations of motion and they are left dependent upon simple geometric considerations.

In section 5.4, a control system was outlined which could be used for both direct wall following using two ultrasonic sensors or imaginary line following using estimated position in a local frame. The basic control rule states that a local demand heading ( $\theta'_r$ ) pointing towards the vehicle's current demand path should be generated in proportion to the local perpendicular distance error ( $y$ ). A demand turning curvature is then derived in proportion to the error the vehicle's local heading with respect to a demand heading derived in this fashion. The initial control system configuration used to implement such a scheme is shown in figure 7.1. Using this figure, and initially ignoring the limits on the rate of curvature change, the basic control equations can be written as:

$$\theta'_r = -k_{pd}(y - y_{lw}) \quad (7.1)$$

$$\kappa = k_p(\theta'_r - \theta') \quad (7.2)$$

Since the AGV control is multivariable, in that yaw angle relative to the designed path and distance perpendicular to the designed path must be controlled, a derivation of the closed loop transfer function will now be made in a simple state space format. This can then be tested for different values of  $k_p$  and  $k_{pd}$  and compared to the predicted results. An alternative derivation which gives the same result, can be made using equations 7.1 and

a modification of the geometrical model employed by Steer [STEER, 1985] [PEARS and BUMBY, 1989]. This is given in appendix 4.

Combining controller equations 7.1 and 7.2 and taking  $y_{tw} = 0$ , as in the case of line following, gives:

$$\kappa = -k_p k_{pd} y - k_p \theta' \quad (7.3)$$

If two of the three quantities  $y$ ,  $\theta'$ , and  $\kappa$ , are known, then the third can be derived from equation 7.3. Thus any pair of variables are independent and can be chosen to represent the system. Let the system states initially and arbitrarily be chosen as:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta' \\ \kappa \end{bmatrix} \quad (7.4)$$

Consider a vehicle moving towards its demand path as shown in figure 7.2. The rate of change of perpendicular distance,  $y$ , in the local coordinate frame of the command is given by

$$\dot{y} = V \sin \theta' \quad (7.5)$$

and if  $\theta'$  is sufficiently small:

$$\dot{y} \approx V \theta' \quad (7.6)$$

In an experiment that is described in the following section, the vehicle was subjected to a step disturbance in  $y$  of 20cm and the largest angle ( $\theta'$ ) attained by the AGV was 8.5 degrees. In this case the maximum error incurred by the above approximation is less than one percent.

Note also that

$$\dot{\theta}' = V \kappa \quad (7.7)$$

and differentiating 7.3 gives:

$$\dot{\kappa} = -k_p k_{pd} \dot{y} - k_p \dot{\theta}' \quad (7.8)$$

Substituting equations 7.6 and 7.7 into equation 7.8 gives

$$\dot{\kappa} = -k_p k_{pd} V \theta' - k_p V \kappa \quad (7.9)$$

and writing equations 7.7 and 7.9 in the matrix form

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} \quad (7.10)$$

gives:

$$\begin{bmatrix} \dot{\theta}' \\ \dot{\kappa} \end{bmatrix} = \begin{bmatrix} 0 & V \\ -k_p k_{pd} V & -k_p V \end{bmatrix} \begin{bmatrix} \theta' \\ \kappa \end{bmatrix} \quad (7.11)$$

The characteristic equation of the system is given by

$$|\mathbf{A} - s\mathbf{I}| = 0 \quad (7.12)$$

which in this case is:

$$s^2 + k_p V s + k_p k_{pd} V^2 = 0 \quad (7.13)$$

Note that if the AGV is travelling backwards, velocity is negative, and so, in order to conform to the Routh-Hurwitz criterion of stability, both  $k_p$  and  $k_{pd}$  must be made negative to keep coefficients positive and the system stable. To account for negative velocities, the variable  $d_n$  is included in the software controller. This has the value -1 when the vehicle travels in the reverse direction and the value +1 for travelling in the forward direction.

The roots of the characteristic equation derived from equation 7.13 are given by:

$$s_{1,2} = \frac{-k_p V}{2} \pm \frac{V}{2} \sqrt{k_p^2 - 4k_p k_{pd}} \quad (7.14)$$

To obtain the fastest response without inducing overshoots these roots should be chosen in the correct ratio to give critical damping ( $\xi=1$ ). This requires that the discriminant of equation 7.14 be zero, thus:

$$k_p = 4k_{pd} \quad (7.15)$$

If  $k_p > 4k_{pd}$  the AGV's motion is overdamped and if  $k_p < 4k_{pd}$  the motion is underdamped and overshoots should be observed.

If the characteristic equation of the system (equation 7.13) is written in the normalised form

$$\frac{s^2}{\omega^2} + Ts + 1 = 0 \quad (7.16)$$

then  $T$  is given by

$$T = \frac{1}{k_{pd} V} \quad (7.17)$$

and the damping factor,  $\xi$ , is:

$$\xi = \frac{T\omega}{2} = \frac{\sqrt{k_p k_{pd}}}{2k_{pd}} \quad (7.18)$$

## 7.3 Performance of the Control System in Line Following

### 7.3.1 Response to a Step Disturbance

To test the accuracy of the theory outlined in section 7.2, the AGV was asked to execute the command queue:

LF,F,0.2,0.0,0.5,0.0,0

LF,F,0.2,0.0,5.0,0.0,20

ST,0

The first command allows the AGV 0.5m to reach a speed of 0.2m/s and the following command has a 20cm offset from the imaginary line which acts as a step disturbance on the output of the control system (see figure 7.1). The results of the vehicle's performance for different combinations of control parameter, using odometry control, are shown in figure 7.3a-7.4d. The experiments were conducted using uncorrected odometric measurements, derived from internal shaft encoders, as a feedback. (If position corrections were permitted during the experiment, it would have been difficult to assess the performance of the control system, as corrections would have acted as additional step disturbances.) Note that in this test a flag was set in the AGV software to remove the curvature rate limit block from the control, and the stepper motor drives were switched to current boost to prevent stalling.

The first graph, figure 7.3a, shows the path of the vehicle while figure 7.3b shows how the distance error, which is the distance between the midpoint of the AGV's drive wheels and the vehicle's demand path, varies with time when the control constants are chosen to give critical damping ( $k_p=4$ ,  $k_{pd}=1$ ). From figure 7.3b it can be seen that distance error is reduced to under 5% of the initial error (i.e. within 1cm) between 11.5 and 12 seconds after the step change was introduced.

With  $k_p = 4$ ,  $k_{pd} = 1$ , and  $V = 0.2m/s$ , and using equation 7.14, a double closed loop pole is obtained with value -0.4. Thus to attain an accuracy of 5% it takes the vehicle  $t_r$  seconds after the step change where  $t_r$  is defined by

$$[1 + 0.4t_r]e^{-0.4t_r} = 0.05 \quad (7.19)$$

which gives  $t_r = 11.9s$ . \* Figure 7.3c shows that the AGV reaches angles of up to 8.5 degrees relative to the line which partly accounts for the discrepancy between measured and predicted results. Figure 7.3d shows the large initial change in curvature ( $0.8m^{-1}$ ) that the step disturbance in  $y$  generates, and indicates that the vehicle turned right (positive curvature) and then left (negative curvature) with curvature reducing gradually as it approached its designed course.

Figures 7.4a to 7.4d show a similar set of results when the control constants are chosen to give underdamping ( $k_p = 1$ ,  $k_{pd} = 3$ ). The damping factor calculated using these constants in equation 7.18 is 0.289 which corresponds to an overshoot of

$$M_p = \exp\left[-\frac{\pi\xi}{\sqrt{1-\xi^2}}\right] = 0.39 \quad (7.20)$$

at  $t_p$  seconds after the step change was introduced where:

$$t_p = \frac{\pi}{w\sqrt{1-\xi^2}} = 9.47s \quad (7.21)$$

These values can be seen to be in close agreement with those measured on figure 7.4b where an overshoot of 40% can be observed 9 seconds after the step change was introduced.

The relation between the control constants can be explained qualitatively. If  $k_{pd}$  is large, the vehicle's demand path relative to the new line (i.e. relative to the parallel 20cm offset) is steep and only at very small distances will the demand angle decay to zero. Since  $k_p$  is small, only relatively small

---

\* Note that in the remainder of this chapter, rise time,  $t_r$ , is defined as the time in which it takes the research vehicle to attain 95% of a step change.

curvatures are generated and the AGV cannot attain the large changes in demand angle set by  $k_{pd}$ . This lag, generated because  $k_p$  is small compared to  $k_{pd}$ , causes the vehicle to cross its demand path and oscillations will ensue until demand angles are small enough to be attained by the value of  $k_p$ .

Given that the control constants are chosen in the correct ratio to give critical damping, the rise time of the AGV's motion is dependent on the value  $\frac{1}{k_{pd}V}$  in equation 7.17. This is not surprising since a higher  $k_{pd}$  generates steeper demand angles as the AGV approaches its line and it hardly needs to be stated that if the vehicle is travelling faster it will complete its designed course in a shorter time!

It should be noted here that predictions of the results presented in this section are based on a representation of the system in the continuous time domain. The control operates at intervals of 0.1s which means that it takes approximately 120 samples to attain 95% of a step change. At this point, this is sufficient to validate a continuous domain analysis. The degree of approximation in continuous domain analysis and the amount by which the sampling interval may be expanded without excessive depreciation of the system's performance are investigated later in this chapter.

### 7.3.2 Steady State Response

Using Laplace's final value theorem and the characteristic equation (7.13), the steady state error of the control system is theoretically zero. However, steady state errors do exist, as can be seen in figure 7.5a which shows the measured steady state distance error for odometric line following when the vehicle was started at position  $X, Y = 0, 0$  and asked to follow the  $X$  axis for a distance of 6m. (In this test, position corrections were suppressed.)

Steady state errors often arise in control systems because the actuating

signal generated by the controller is just below some threshold required to effect a change in the value being controlled. If a simple proportional controller is employed and gain is increased, then steady state error can decrease in the same proportion for the output of the controller to remain on the threshold of actuation. Obviously gain can not be increased indefinitely as it is likely to lead to instability especially if the system is subject to large disturbances. In this case, integration of the error signal generates actuating signals that reduce steady state error in an exponential fashion. The smallest curvature that the AGV can actuate is dependent on the quantisation of stepper speeds precipitated by the method of generating source pulses. At a base speed of 0.2m/s, this is  $0.00227 \text{ m}^{-1}$  which is approximately the curvature demand at a distance error of 0.5mm. This, therefore, is not the main source of steady state error.

The main cause of steady state errors in odometric line following is the limited resolution of the shaft encoders. If the left shaft encoder returns one more count than the right in a control interval, the estimate of AGV heading increases by 0.5 degrees (0.00872 rads). This quantisation is shown in figure 7.5b. The implication of this is that, if  $k_{pd}$  is set to 1, distance errors within the range of 8.7mm either side of the imaginary line will generate demand headings as small or smaller than the resolution of quantised angle measurement so that once within this band, accuracy can not be improved.

#### 7.4 Performance of the Control System in Tracking a Curve

The results of the previous section show that, in the absence of curvature rate limiting, the control system performs well in line following, both when subject to step output disturbances and when tracking in the steady state. In order to determine whether the control system performs

equally well whilst tracking a curve, the AGV was asked to execute the command queue:

LF,F,0.2,0.0,1.0,0.0,0

CF,F,0.08,0.9,360.0,1.0,0.0

LF,F,0.2,0.0,3.0,0.0,0

ST,0

In executing this queue, the AGV moves on a heading of 0.0 degrees to a position,  $X, Y = 1.0, 0.0$ , executes a full (360 degrees) circle on a radius of 0.9m, and finally follows an imaginary line to position  $X, Y = 3.0, 0.0$ . Figure 7.6a shows the path of the vehicle and the vehicle's perpendicular displacement from the curve is shown in figure 7.6b. It is clear that a large steady state offset quickly develops and the control system, as it stands, is clearly inadequate for curve following.

For the case of curve following, first order equations for the system states may be written as

$$\dot{y} = V\theta' \quad (7.22)$$

as before; however, the rate of change of local yaw angle is now given by

$$\dot{\theta}' = V\kappa - V\kappa_{seg} \quad (7.23)$$

where  $\kappa_{seg}$  is the curvature of the command segment. This command segment curvature generates a demand yaw angle rate and the state equation is now

$$\begin{bmatrix} \dot{\theta}' \\ \dot{\kappa} \end{bmatrix} = \begin{bmatrix} 0 & V \\ -k_p k_{pd} V & -k_p V \end{bmatrix} \begin{bmatrix} \theta' \\ \kappa \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} - V\kappa_{seg} \quad (7.24)$$

or in an alternative state representation:

$$\begin{bmatrix} \dot{y} \\ \dot{\theta}' \end{bmatrix} = \begin{bmatrix} 0 & V \\ -k_p k_{pd} V & -k_p V \end{bmatrix} \begin{bmatrix} y \\ \theta' \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} - V \kappa_{seg} \quad (7.25)$$

Figure 7.6b indicates that the disturbance caused by the curvature of the segment precipitates a steady state error in the state  $y$  of approximately -23cm. Rearranging equation 7.3, and using the subscript  $ss$  to denote steady state values, the steady state value of the local perpendicular distance to the demand path is given by:

$$y_{ss} = -\frac{1}{k_p k_{pd}} (\kappa_{ss} + k_p \theta'_{ss}) \quad (7.26)$$

In the steady state of tracking a curve, the vehicle must be travelling parallel to, if not on, its demand path, as can be seen in figure 7.6a. This implies that the steady state of local angle error must be near to zero. Figure 7.6c shows the local angle disturbances at entry and exit of the curve follow command and verifies that, in the steady state of the curve follow command, local angle error is near to zero. Thus:

$$\theta'_{ss} = 0 \quad (7.27)$$

The steady state error in  $y$  can be used to form the steady state curvature as

$$\kappa_{ss} = \frac{1}{r_{seg} - y_{ss}} \quad (7.28)$$

where  $r_{seg}$  is the radius of the command segment. Substituting equations 7.27 and 7.28 into equation 7.26 gives:

$$y_{ss} = \frac{1}{k_p k_{pd}} \left[ \frac{1}{y_{ss} - r_{seg}} \right] \quad (7.29)$$

This gives a quadratic in  $y_{ss}$  as

$$y_{ss}^2 - r_{seg}y_{ss} - \frac{1}{k_p k_{pd}} = 0 \quad (7.30)$$

thus:

$$y_{ss} = \frac{r_{seg} \pm \sqrt{r_{seg}^2 + \frac{4}{k_p k_{pd}}}}{2} \quad (7.31)$$

Equation 7.31 indicates that the steady state offset in the state  $y$ ,  $y_{ss}$ , is dependent on the command segment curvature ( $\kappa_{seg} = \frac{1}{r_{seg}}$ ) and the choice of control constants. Substituting values of  $r_{seg} = 0.9$ ,  $k_p = 4$ , and  $k_{pd} = 1$  gives  $y_{ss} = -22.27cm$ , which is in close agreement with the value obtained from figure 7.6b. The steady state curvature, given by equation 7.28, is  $0.89m^{-1}$  which is consistent with figure 7.6d. It can be seen from equation 7.31 that as the control constants are increased, the steady state error tends to zero. Figure 7.7a and 7.7b show results when the control constants are doubled. The steady state error generated by equation 7.31 with  $r_{seg} = 0.9$ ,  $k_p = 8$ , and  $k_{pd} = 2$  is  $-6.48cm$  which is in close agreement with the result obtained from figure 7.7b.

Since a non-zero value of  $y_{ss}$  must exist in order to generate the curvature of a path parallel to that of the command segment, it can be concluded that the use of proportional controllers and local position ( $y, \theta'$ ) *only* is inadequate for tracking a curve. There are two possible methods by which to eliminate this steady state error:

- (1) Employ higher order controllers.
- (2) Consider  $\kappa_{seg}$  as an input to the control system.

By including an integrating term in addition to  $k_{pd}$  and/or including a differentiating term in addition to  $k_p$  it was found that it was possible to greatly reduce steady state errors in curve tracking. Differentiating local angle

error is effectively the same as measuring the curvature of the segment, if the differential constant is chosen in accordance with vehicle velocity,  $V$ . The preferable method, therefore, is to consider the command segment curvature as an input to the control system.

If  $\kappa_{seg}$  is considered as an input, then this curvature is added to that derived from the measured local position  $(y, \theta')$  before the required differential wheel speed is calculated. This is shown in the schematic overview of the control system in figure 7.8. Since specific command segments have specific constant segment curvatures, a transition between one command and the next applies a step input to the control system. Also shown in figure 7.8 are positions corrections acting as step disturbances on the outputs of the system. In this modified system, the curvature that is applied to the vehicle's drive wheels is redefined as the difference between the curvature derived from local position  $(y, \theta')$  and the curvature of the command segment, rather than being derived from local position only. The state equation for curve following is now the same as that for line following, which is simply a special case where  $\kappa_{seg} = 0$ .

More formally, if curvature is redefined as

$$\kappa = \kappa_{lp} - \kappa_{seg} \quad (7.32)$$

where

$$\kappa_{lp} = k_p [-k_{pd}, 1] \begin{bmatrix} y \\ \theta' \end{bmatrix} \quad (7.33)$$

then the original state equation

$$\begin{bmatrix} \dot{y} \\ \dot{\theta}' \end{bmatrix} = \begin{bmatrix} 0 & V \\ -k_p k_{pd} V & -k_p V \end{bmatrix} \begin{bmatrix} y \\ \theta' \end{bmatrix} \quad (7.34)$$

describes the same performance in both line and curve following. The analysis of the previous section showed that, by Laplace's final value theorem, this system has a theoretical steady state error of zero and can operate in a critically damped fashion if the control constants are chosen in the correct (4:1) ratio.

To test the accuracy of this modification to the control, an experiment illustrated by figure 7.9a was conducted. This shows the path of the AGV when it was asked to follow a line at a perpendicular displacement of 50cm so that when it entered the subsequent curve follow command, there was an initial error of 50cm. This simulated an instance when a large position correction occurs just before the vehicle enters a curve segment. Figure 7.9b shows that perpendicular distance error is reduced in an almost critically damped fashion whilst tracking in the steady state gives almost zero error.

## 7.5 Velocity Dependent Control Parameters

To keep the performance of the control system constant and velocity independent, the system rise time as well as the damping factor should be held constant. In effect, this requires maintaining the position of the system's open loop poles and adjusting the system gain to maintain critical damping while the velocity of the vehicle changes.

Using the characteristic equation (equation 7.13), the open loop transfer function of the basic control system,  $G(s)$ , can be derived from

$$1 + G(s) = s^2 + k_p V s + k_p k_{pd} V^2 \quad (7.35)$$

which intuitively gives

$$G(s) = \frac{k_{pd} k_p V^2}{s(s + k_p V)} \quad (7.36)$$

(An explicit derivation of  $G(s)$  in terms of state space matrices can be found in a recent paper [PEARS and BUMBY, 1989b]) Thus the open loop transfer function of the basic control system has poles at 0 and  $-k_p V$ , and a system gain of  $k_{pd}$ . In order to keep the position of the open loop poles constant and, therefore, the critically damped rise time of the system constant,  $k_p$  must be chosen in inverse proportion to velocity. Gain,  $k_{pd}$ , must then be chosen to be a quarter of the value of  $k_p$  in order to maintain critical damping. More specifically, the control constants  $k_p$  and  $k_{pd}$  are made dependent on velocity such that:

$$k_p = 5/4V \quad (7.37)$$

$$k_{pd} = k_p/4 \quad (7.38)$$

Thus for a velocity of 0.2m/s control constants of  $k_p = 4$  and  $k_{pd} = 1$  are employed, whereas if the vehicle were travelling slower (0.1m/s), a higher gain would be employed ( $k_p = 8$ ,  $k_{pd} = 2$ ). These velocity dependent control parameters cause the vehicle to approach its path more steeply, since  $k_{pd}$  is higher, when travelling more slowly. This is a sensible approach since, for a given rate of change of curvature, smaller angular accelerations are developed at a lower base velocity ( $\ddot{\theta}' = V\dot{\kappa}$ ). With this scheme operating, the torque demand on the stepper motors increases in inverse proportion to velocity since each time velocity is reduced by half, both  $k_p$  and  $k_{pd}$  are doubled. Thus the initial demand rate of change of curvature after a step disturbance is multiplied fourfold and both the initial angular acceleration and demand torque are doubled. As a consequence of this, the control method is particularly suited to a stepper motor drive which can provide much larger torques at low stepping rates.

In addition to being suited to a stepper motor drive system, the constant rise time feature of the control system becomes particularly important as the vehicle approaches its destination. If a reasonably sized position correction occurs just before the goal position is reached, a poor rise time may not allow the vehicle to reach its goal position with a good accuracy (i.e. some control error may still exist at the goal position.) The use of velocity dependent control parameters means that, since rise time is held constant with respect to velocity, the forward travelling distance required to attain 95% of a lateral position correction is proportional to velocity. Thus, by reducing demand base velocity on the command segment leading up to the goal position, the constancy of rise time ensures a good control accuracy at the vehicle's destination. This can be compared to the common experience of severely reducing speed, whilst parking a car, in order to steer more sharply.

It is obvious from equations 7.37 and 7.38 that at low base velocities, large control constants are generated. A large value of  $k_{pd}$  suggests that the vehicle's approach to its demand path is steep. Such an approach invalidates the small angle approximation made in equation 7.6 and the vehicle's behaviour would deviate from that predicted. To prevent this happening there is, in the AGV software, the added constraint that if base velocity is below 0.1m/s, then the control constants are held at  $k_p = 8$  and  $k_{pd} = 2$ .

Figures 7.10a to 7.10d show results of the line following step disturbance test described in section 7.3 when demand speed is reduced to 0.1m/s. Comparing figures 7.10a and 7.10b with 7.3a and 7.3b shows that the vehicle follows a much steeper path yet the rise times are almost identical. A comparison of figure 7.10c and 7.10d with figures 7.3c and 7.3d show the vehicle's increased local angle with respect to the demand path and the increased initial change in curvature when the research vehicle moves at a lower base velocity.

## 7.6 Curvature Rate Limiting

If the control constants ( $k_p$ ,  $k_{pd}$ ) are chosen to be high in order to give a short rise time, and the system is subjected to a large step change, the controller may generate demand curvatures high enough to stall the stepper motors. Section 7.5 explained how the control constants were chosen according to base velocity in order to give a rise time that was attainable for step disturbances of 20cm or less.

In order to make the control more robust, the rate of change of curvature, or rather the amount by which curvature can change at a control interval, is limited to prevent stepper motor stalling in instances of position corrections of a greater magnitude. Whereas, in the usual case, a controller is designed to provide an actuating signal that causes the system to follow some prescribed input as closely as possible, the curvature rate limit block is designed to prevent the controlling signal from placing demands on the AGV's stepper drives that are not realisable. The AGV software provides a choice of one of two types of curvature rate limiting, one non-linear method and one linear method; a third option permits the limit block to be removed from the control. The effect of each of these blocks to a step input is illustrated in figure 7.11.

The first type of limiting determines the demand change in curvature relative to the previous low level control interrupt and if this is greater than some permitted limit, the curvature is only permitted to change by that limit value. This method of setting a *maximum* rate of change of curvature introduces a non-linear block into the control system which was not included in the original analysis. In order to test the effect of this limit block, the curvature was only allowed to change by a small amount ( $0.009m^{-1}$ ) when a step change of 20cm was applied to the control system. The result obtained

in figure 7.12a, with the previously defined values of  $k_p$  and  $k_{pd}$  for critical damping, now shows an underdamped, almost oscillatory response because the limited curvature does not allow the vehicle to reach the demand headings set by  $k_{pd}$ . Figure 7.12b shows the vehicle curvature moving at a constant rate ( $0.09m^{-1}s^{-1}$ ) between its maximum and minimum values. If curvature changes of  $0.8 m^{-1}$  are permitted between consecutive control interrupts (i.e. if the curvature rate limit is set high), then the controller, with this form of non-linear limiting, exhibits critically damped behaviour for step disturbances up to 20cm, and underdamped behaviour for larger step disturbances.

In contrast, the second available method of curvature rate limiting is linear in that a limited curvature is derived as a proportion of the difference between the current demand curvature (as derived from the vehicle's local position and the command segment curvature) and the curvature that was applied over the last control interval. Such a scheme was chosen since it is the simplest form of rate limiting where the steady state value of the output eventually reaches that of a non-changing input and where the output is derived as a function of input. The advantage of a linear limit block is that its effect is predictable and can be included in the control analysis in order to maintain critical damping and a good, realistically attainable rise time.

In the AGV software, the rate limited curvature applied to the vehicle's drive wheels at control interrupt  $i$  is calculated as

$$\kappa_i = \kappa_{i-1} + k_l(\kappa_{dem_i} - \kappa_{i-1}) \quad (7.39)$$

where

$$\kappa_{dem_i} = \kappa_{lp_i} + \kappa_{seg} \quad (7.40)$$

as is shown in figure 7.8. Alternatively equation 7.39 can be written as

$$\kappa_i = (1 - k_l)\kappa_{i-1} + k_l\kappa_{dem_i} \quad (7.41)$$

which is the form of a simple recursive digital filter. Equation 7.39 describes a method of curvature rate limiting that has the form of a closed loop integrator. Figure 7.13 shows the transfer function of the limit block in a closed loop form and a reduced form. The constant ( $k_l$ ) is multiplied by ten to account for the 0.1s sampling time of the integration. Closing the loop round the integration yields the transfer function:

$$G_{lb}(s) = \frac{\kappa_i}{\kappa_{dem_i}} = \frac{10k_l}{(s + 10k_l)} \quad (7.42)$$

Note, however, that the feedback is delayed by one control interval relative to the limit block input. This means that the feedback path of the limit block has a transfer function  $e^{-0.1s}$  and equation 7.42 is an approximation of the limit block's overall transfer function. (A more accurate derivation of equation 7.42 can be found in a recent paper [PEARS and BUMBY, 1989b].) A delay of one sampling interval is more effectively handled in the discrete domain where it is represented by  $z^{-1}$ . This is dealt with in the following section.

In order to choose control parameters that will maintain critical damping and still give a reasonable rise time, this limiting block must be included in the control analysis. The introduction of the limiting block, modifies the system's open loop transfer function (equation 7.36) to:

$$G(s) = \frac{10k_p k_{pd} V^2 k_l}{s(s + k_p V)(s + 10k_l)} \quad (7.43)$$

The effect of the limit block is to introduce an additional pole on the

real axis of the root locus at  $-10k_l$ . Again, if  $k_p$  is generated in proportion to  $1/V$  in order to fix one open loop pole of the system, the choice of  $k_l$  fixes another open loop pole, and  $k_{pd}$  is left to maintain critical damping. Figure 7.14a shows the root locus of the original (unlimited) system whilst 7.14b shows the effect of the additional pole at  $s=-2$ . For this second root locus, the values

$$k_p = 4, V = 0.2m/s, k_l = 0.2$$

were employed, which require that  $k_{pd}=0.81$  for critical damping. Thus the correct ratio of  $\frac{k_p}{k_{pd}}$  for critical damping is modified from 4 to 4.93 by the inclusion of the limit block.

It is difficult to determine precisely what the best choice of  $k_l$  is. The aim of the control design is to maintain critical damping and have the fastest rise time possible without generating loading torques that may stall the stepper motors. The 'worst case' situation is when a large position correction occurs just before the vehicle has an additional (additive) step change imposed upon it by entering a curve follow command.

Table 7.1 shows a number of results generated by a control design and analysis package (*CC*-'Comprehensive Control' developed by *STI* inc.) The columns in this table are the curvature rate limiting constant, the required gain to maintain critical damping, the time to reach 95% of a step disturbance, and the magnitude of the step disturbance that is required to generate a demand change of curvature of  $0.8m^{-1}$  at the first control interrupt after the change. (This was quoted as the approximate change in curvature at a control interval required to stall the drive system when travelling at a base velocity of 0.2m/s.) The results were generated with the assumption that  $k_p V=0.8$ .

$k_l$	$k_{pd}$	$t_r(s)$	$\frac{0.8}{k_p k_{pd} k_l} (m)$
0.1	0.656	17.05	3.047
0.2	0.812	13.99	1.231
0.5	0.922	12.60	0.434
0.8	0.951	12.30	0.263
1.0	0.960	12.22	0.208
-	1.000	11.90	0.200

Table 7.1: *CC* predictions for curvature rate limited system.

Table 7.1 shows that although the system rise time improves as the degree of curvature rate limiting is relaxed (i.e. as  $k_l$  increases), the step disturbance required to stall the drive system gradually approaches that of the unlimited system (20cm). Since the magnitude of step disturbance can not exceed the range of the vehicle's ultrasonic sensors, a value of 0.1 for  $k_l$  provides a greater degree of limiting than is strictly necessary. A value of 0.2 for  $k_l$  allows position corrections to be made that are compatible with the ultrasonic measurement range. In addition, if corrections over 1m are required, the AGV software deems that the vehicle is lost which causes it to stop dead.

Thus the inclusion of a limit block with  $k_l$  set at 0.2 provides a more robust system and increases the rise time of the system by just over 2s (11.9s to 14s). Figures 7.15a and 7.15b show results of the test run described in section 7.3 when  $k_l=0.2$  and  $k_{pd}=0.823$ . \* Figure 7.15a shows that it takes the vehicle approximately 14s to attain 95% of the step change which is in close agreement with the predicted time in table 7.1. Figure 7.15b shows

---

\* A gain of 0.823 was employed rather than 0.812 since this gain was suggested by the discrete analysis given in the following section.

the curvature of the vehicle and indicates that the vehicle turned right and then left *gradually* as it approached its new demand path. This figure can be compared with figure 7.3d where there is an initial large change of curvature followed by a series of much smaller changes. Thus the effect of the limit block is to distribute the curvature changes more evenly over the whole of the steering process.

A rise time of around 14s means that, for a velocity of 0.2m/s, the vehicle is within 5% of its desired lateral position approximately 2.8m after the step disturbance occurred. In general, this has proved to be acceptable for the research rig. However, it is possible to improve the rise time of the system and yet maintain robustness with respect to step disturbances. This requires increasing  $k_p$  and thus moving the pole  $-k_p V$  away from the imaginary axis whilst reducing  $k_i$  and thus moving the pole  $-10k_i$  closer to the imaginary axis.

Using the values

$$k_p=10, V=0.2\text{m/s}, k_i=0.1$$

a gain of  $k_{pd}=0.962$  is required for critical damping and *CC* predicts a rise time of 11.73s. Such a rise time is compatible with the non curvature rate limited system, yet a step disturbance of 0.832m is required to generate an immediate curvature change of  $0.8\text{m}^{-1}$ . Figure 7.16a shows a simulated comparison of a curvature rate limited system, with the above selection of parameters, and the original unlimited control system. Although the two systems have similar rise times, the form that curvature takes in the curvature rate limited system is much smoother. This curvature is shown in figure 7.16b and can be compared with the curvature of the original system in figure 7.3d. It can be seen that, rather than having a large step change in curvature immediately after an output step disturbance, the limiting block spreads changes in curvature much

more evenly over the whole of the steering process, thus reducing the peak torque demand on the stepper motors for a given rise time.

## 7.7 An Analysis of the Control in the Discrete Domain

### 7.7.1 Aims and Method of the Discrete Analysis

In the analyses of the previous sections, it was assumed that the sampling time ( $T_s=100\text{mS}$ ) is small enough in comparison to the time constants of the control process for design and analysis in the continuous ( $s$ ) domain to prevail. Although inspection of figure 7.3b validates this assumption, since it takes the AGV approximately 12 seconds or about 120 samples to attain 95% of its step change, no attempt has been made to quantify the accuracy of an  $s$ -domain analysis of the discrete control system. In addition, it was found that, with the limited speed of on-board processing, 100mS was only just sufficient time to do all the necessary data processing and control calculations that constitute the guidance method.

The discrete analysis presented in this section has two aims

- (1) To determine by how much predictions of the control performance differ when the system is modelled in the discrete domain rather than the continuous domain. Any observable differences in the system's behaviour when modelled in this manner may suggest a more accurate choice of the required control parameters
- (2) To determine by how much the sampling interval may be increased before the continuous domain predictions depart from those made from a discrete model of the system, and by how much the sampling interval may be increased before the performance of the control system becomes severely

depreciated.

In order to model the control system in the discrete domain, the  $s$ -plane representation was transformed to a zero-order hold equivalence  $z$ -plane representation. This discrete representation was then employed in a control design and analysis package \* in order to make a comparison of the two performances both in the time and frequency domains. Any deviations from the critically damped ( $\xi = 1$ ) performance which the chosen control constants give in the  $s$ -plane may be eliminated by fine tuning the control constants and thereby appropriately adjusting the closed loop poles in the  $z$ -plane. It should be noted, however, that the location of closed loop poles has a more complex form on the  $z$ -plane than the  $s$ -plane; for example,  $\xi = \text{constant}$  is a logarithmic spiral as opposed to a straight line at a given orientation relative to the imaginary axis. For this reason, the  $z$ -domain representation may be transformed to the  $w$ -domain which has a similar form to the  $s$ -plane making it easy to assess the effect of specific closed loop pole locations.

### 7.7.2 Z-Transform System Without Curvature Rate Limiting

Firstly consider the original control system in which there was no limit on curvature rate. During the control interval,  $T_s$ , the curvature applied to the AGV's drive wheels is held constant. This may be represented by adding a sampler and zero order hold in series with the system's open loop transfer function:

$$G(s) = \left[ \frac{1 - e^{-sT_s}}{s} \right] \cdot \frac{k_p k_{pd} V^2}{s(s + k_p V)} \quad (7.44)$$

Taking  $k_p=4$ ,  $k_{pd} = 1$ , and  $V = 0.2m/s$  gives:

---

\* *CC*-'Comprehensive Control' developed by *STI* inc.

$$G(z) = (1 - z^{-1})Z\left(\frac{0.16}{s^2(s + 0.8)}\right) \quad (7.45)$$

A partial fraction expansion and use of  $z$ -transform tables gives:

$$G(z) = \frac{0.00079(z + 0.974)}{(z - 1)(z - 0.923)} \quad (7.46)$$

### 7.7.3 $Z$ -Transform of the Curvature Rate Limited System

It can be recalled that the curvature rate limit block has the form of a closed loop integrator. The integrator ( $\frac{k_I}{s}$ ) in the forward path of the loop transforms to:

$$G_{lb_f}(z) = \frac{k_I}{1 - z^{-1}} \quad (7.47)$$

Note that the factor of ten in equation 7.42 does not appear here, since it is taken care of on entry of the sampling rate in the program *CC*. It should also be noted that the feedback value of curvature has to be taken as the curvature applied at the previous control interrupt (see equation 7.39); thus the feedback path of the limit block has a transfer function

$$H_{lb}(z) = z^{-1} \quad (7.48)$$

(see figure 7.17a). Since there is a sampler between the forward and feedback paths, the overall transfer function of the curvature rate limit block can be written as

$$G_{lb}(z) = \frac{G_{lb_f}(z)}{1 + G_{lb_f}(z)H_{lb}(z)} \quad (7.49)$$

which gives

$$G_{lb}(z) = \frac{k_I}{1 - (1 - k_I)z^{-1}} \quad (7.50)$$

or alternatively

$$G_{lb}(z) = \frac{k_l z}{z - (1 - k_l)} \quad (7.51)$$

(see figure 7.17b). The form of this  $z$ -transform is consistent with that of a simple recursive digital filter which serves as a good check, since equation 7.41 can be thought of as such a filter.

Substituting the values  $k_p = 10$ ,  $k_{pd} = 1$ , and  $V = 0.2m/s$  into equation 7.36 and adding a sampler with a zero-order hold gives

$$G(z) = (1 - z^{-1})Z\left(\frac{0.4}{s^2(s + 2)}\right) \quad (7.52)$$

or

$$G(z) = \frac{1.873 \times 10^{-3}(z + 0.9355)}{(z - 1)(z - 0.819)} \quad (7.53)$$

Substituting the value  $k_l=0.1$  into equation 7.51 and combining with equation 7.53 yields the open loop transfer function for the curvature rate limited system:

$$G(z) = \frac{1.873 \times 10^{-4}(z + 0.9355)}{(z - 1)(z - 0.819)(z - 0.9)} \quad (7.54)$$

#### 7.7.4 Simulation results

The  $z$ -plane representations described by equations 7.46 and 7.54 were transformed into the  $w$ -plane and, subsequently, the root loci of the original and curvature rate limited systems were re-plotted. These plots suggested that small modifications should be made to the values of gain calculated for maintain critical damping in an  $s$ -plane analysis. Table 7.2 compares results provided by  $CC$  for these two analyses, and a further two analyses, when base velocity is taken as 0.2m/s.

		Gain, $k_{pd}$	
$k_I$	$k_p$	s - plane	w - plane
-	4	1.000	0.980
0.1	4	0.656	0.670
0.2	4	0.812	0.823
0.1	10	0.962	0.999

Table 7.2: A comparison of  $s$ -plane and  $w$ -plane analyses

Table 7.2 shows that the  $w$ -plane and  $s$ -plane analyses differ by around 2% in their estimation of the gain required for critical damping.

Using the inverse of the  $z$ -transforms in equations 7.46 and 7.53,  $CC$  can generate time responses for any defined sampling interval and compare them to the responses obtained from the original (inverse) Laplace transforms. Figures 7.18a and 7.18b show a comparison of digital and analogue responses for the original unlimited and curvature rate limited systems respectively. It can be seen that the curvature rate limited system is much more susceptible to increases in the sampling interval, since large overshoots are observed if  $T_s$  is set above 0.2s.

Figures 7.19a and 7.19b show bode plots generated by  $CC$  for the original unlimited and curvature rate limited systems respectively when these systems are described in the  $s$ ,  $z$ , and  $w$  domains and  $T_s$  is set to 0.1s. In these figures, magnitude is represented by a continuous line whilst phase is represented by a dashed line. From the first plot, the phase lag of the  $s$ -domain representation asymptotically approaches the -180 degree line indicating that the system should never be unstable. However, the point at which the phase of both  $z$  and  $w$ -domain representations cross the -180 degree

line, the gain margin is 100. Thus the system gain has to be 100 before the sampled system becomes unstable. In contrast, the gain margin of the curvature rate limited plot suggests that gain has to be around 15 for the system to become unstable in both discrete and continuous domains. Also it can be noted that in both plots, the various representations diverge severely as the fold-over frequency ( $\frac{\pi}{T_s}$ ) is approached.

### 7.7.5 Conclusions from the Discrete Analysis

A discrete analysis of the control system has led to the following conclusions

(1) When a sampling interval of 100mS is employed, a continuous domain representation of the system is a good approximation to a zero-order hold discrete representation of the system. Although the discrete analysis did suggest minor adjustments of the gain calculated in the continuous domain, and it is a simple matter to make such adjustments in the AGV software, the effect of such adjustments is small, particularly when inaccuracies such as angle quantisation in the system are considered.

(2) It is possible to expand the sampling interval to 200mS without excessive depreciation of the control systems performance. For the original unlimited control system, such an expansion of  $T_s$  generates a response almost identical to that when the sampling interval is 100mS. However, in the case of the curvature rate limited system, gain for critical damping would need to be recalculated from the  $w$ -plane root locus. If  $T_s$  were doubled, there would be sufficient time for data from four sensors to be processed as the vehicle observed two correction boards (one either side of the current aisle). Thus the AGV would be provided with a degree of redundancy. However, expansion of the sampling interval to as much as 0.5s is undesirable on three counts. Firstly,

although the vehicle can still, theoretically, be controlled, larger changes of curvature would be demanded over consecutive control intervals which may stall the stepper motors. Secondly, the control would no longer appear to be smooth, since there is a sufficient interval between consecutive changes in base velocity and curvature to be heard by an observer. Finally, the validity of using first order regression in processing ultrasonic range measurements would be threatened, and the increased amount of data processing may generate computational problems with overflows.

## 7.8 Summary

In this chapter, the initial formulation, the analysis, and the subsequent modification of the AGV's control system have been described. The success of the controller is largely due to the formulation of a good control rule. This states that a local demand heading pointing towards the vehicle's current demand path should be generated in proportion to the local perpendicular distance error. A demand turning curvature is then derived in proportion to the error the vehicle's local heading with respect to a demand heading derived in this fashion. Although this control rule is very simple, it has proved successful, largely because it establishes a link between the vehicle's two degrees of freedom  $y$  and  $\theta'$  relative to its demand path.

After translating the control rule into a specific equation, the vehicle's motion was described by a pair of first order differential equations and the control analysed in a state space format. This analysis was required in order to generate equations relating the control constants to second order criteria such as the damping factor and rise time of the system.

The control system was then tested on the research AGV and the results have shown that although the system performs well when the vehicle is asked to follow a line, command segment curvature, which is zero for a

straight line, needs to be treated as an input to the control system in order to obtain a compatible performance in curve following.

It has also been shown that in order to keep the control system's performance (rise time and damping) independent of velocity, the controller parameters should be chosen in inverse proportion to velocity. This generates sensible behaviour whereby the AGV turns more steeply towards its demand path when it is travelling more slowly.

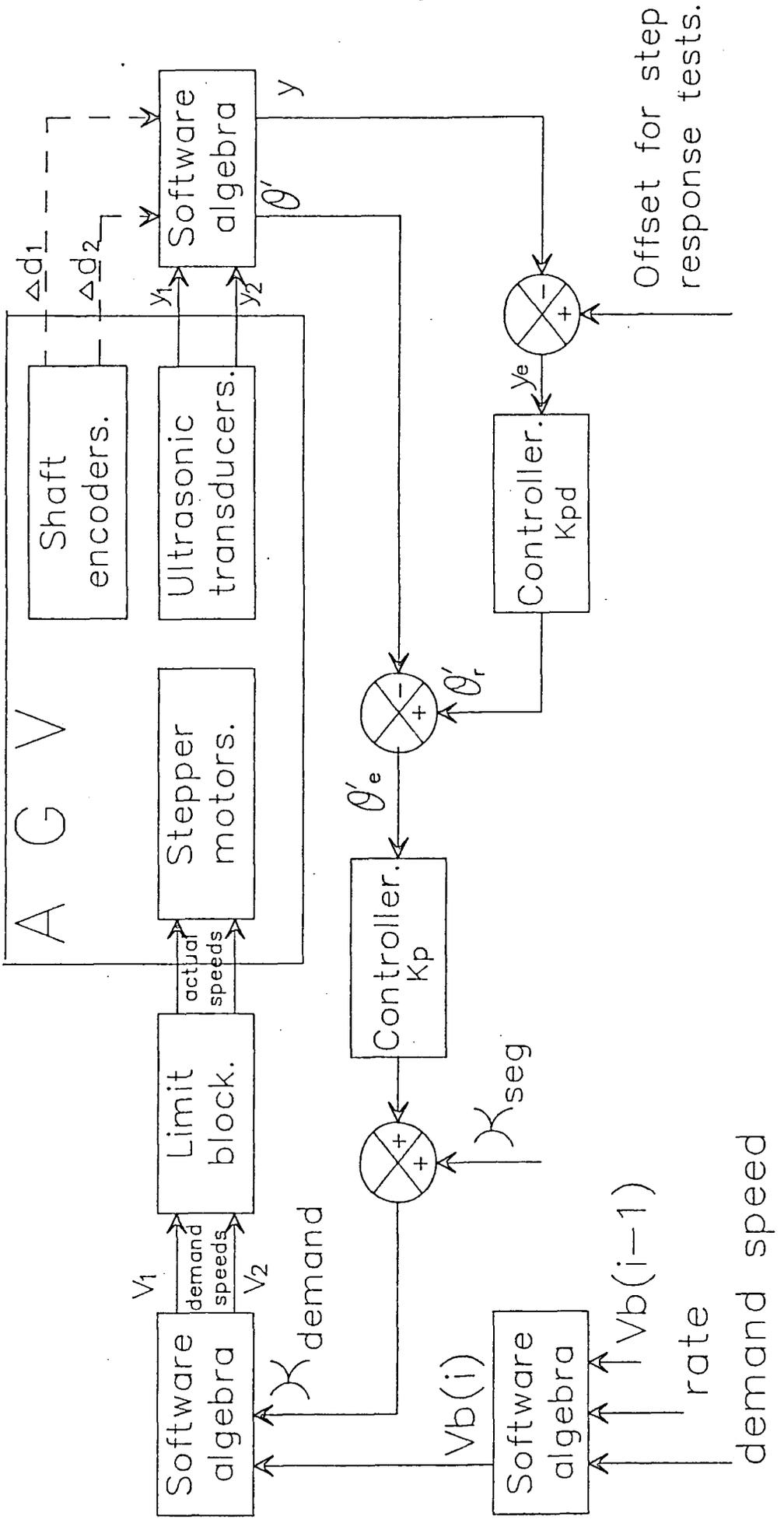
A further modification to the control system was to include a method of curvature rate limiting that prevents stalling of the stepper motors when the vehicle is subjected to a large position correction, and yet behaves in a linear and therefore predictable manner. With such curvature rate limiting included in the control system, it has been shown how the system response is changed. It has also been shown how the control parameters should be tuned to give a system that has a similar performance to the original system without curvature rate limiting, but has the added advantage of placing smaller demands on the stepper motor drives. The curvature rate limiting block achieves this by spreading the large torque demand immediately after a position correction over the whole of the steering process.

Finally an analysis of the control performance in the discrete domain has predicted responses similar to those in the continuous domain. Although this analysis recommended slight changes in the chosen control parameters, these changes were insignificant when inaccuracies such as angle quantisation in the system are considered. In addition it has been shown that the sampling interval may be expanded to 200mS without excessive depreciation of the control system's performance.

Although some of the control aspects described in this chapter, such as curvature rate limiting, are specific to the independent stepper drive steering mechanism, the initial analysis of the basic control rule may be applied to

other steering mechanisms such as vehicles steered by wheels that do not provide propulsion and that need to move in order to turn. In such cases, vehicle dynamics may be included if they affect the way in which curvature can change; the specific transfer function of the steering mechanism and some form of compensation would replace that of the curvature rate limit block, and the system would be reassessed in order to choose parameters that give critical damping and the best realisable rise time for that specific steering mechanism.

Figure 7.1 Block diagram of the control system



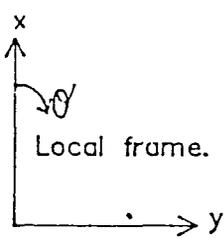
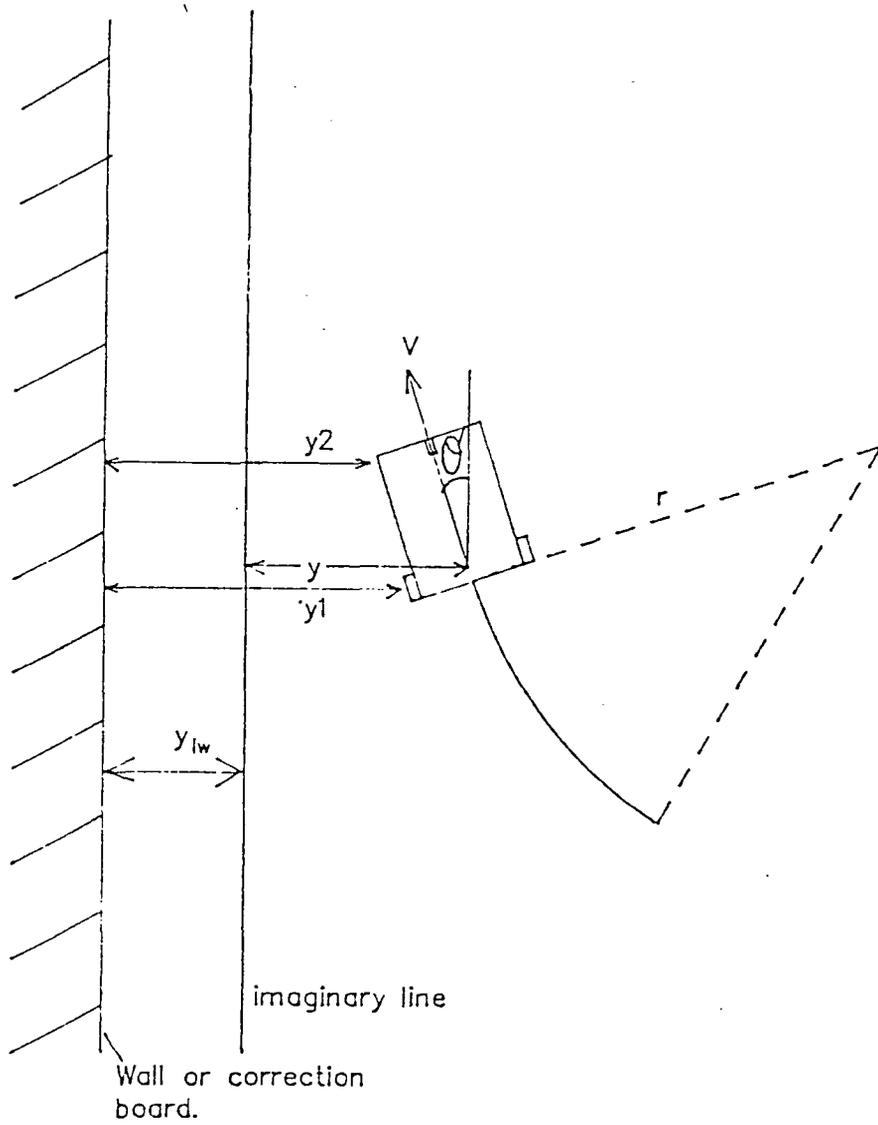


Figure 7.2 The line following process

Figure 7.3a AGV path in a step response test

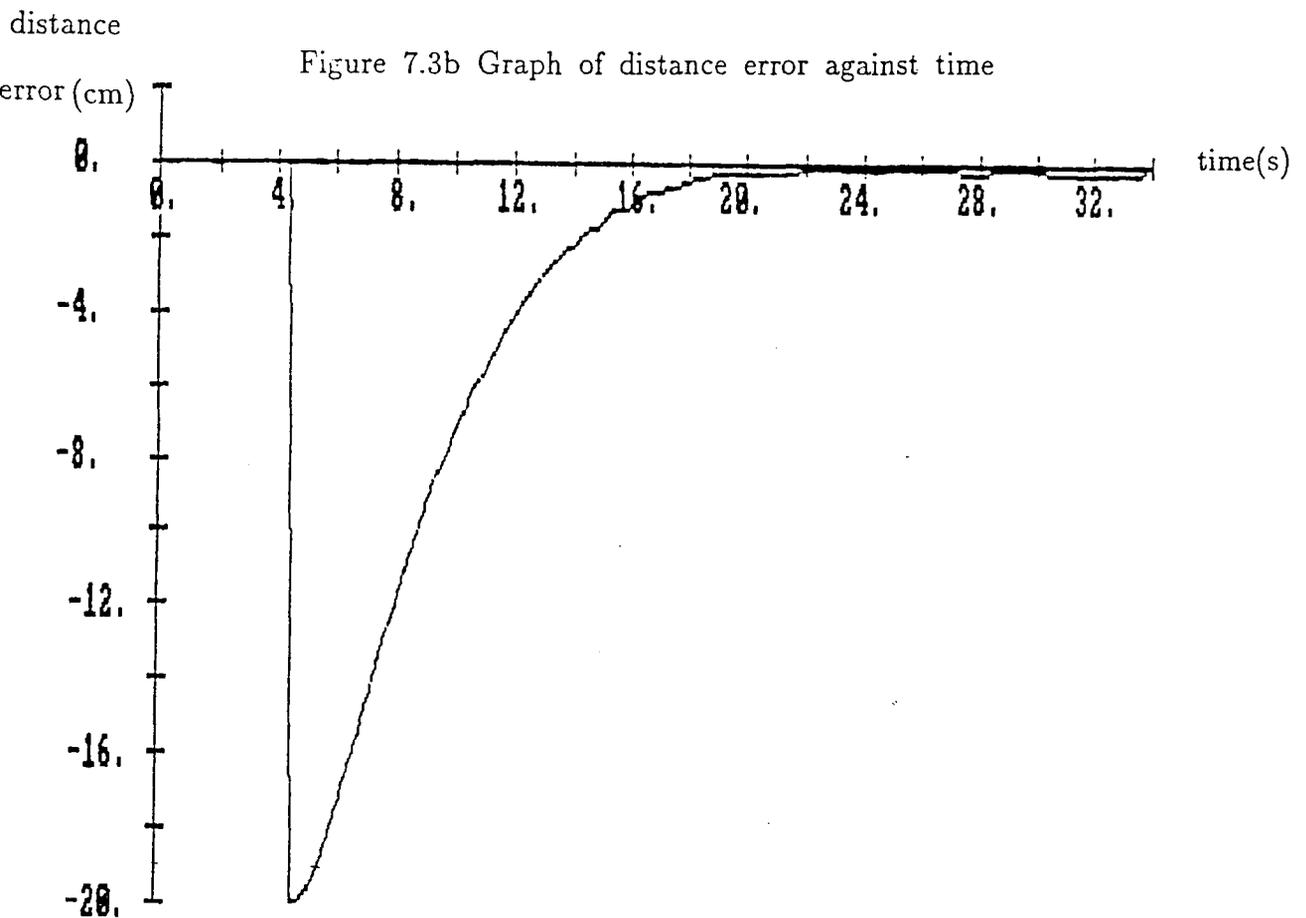
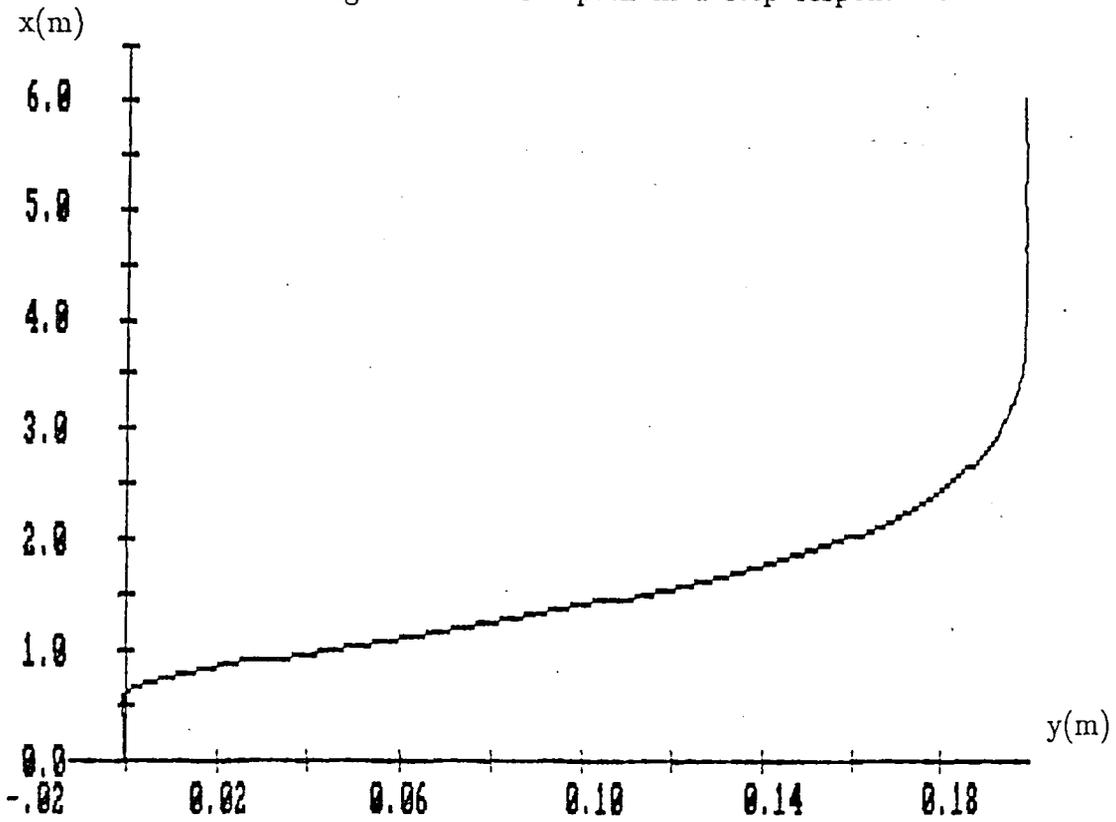


Figure 7.3 Critically damped response:  $k_p = 4$ ,  $k_{pd} = 1$ ,  $V = 0.2m/s$

Figure 7.3c Graph of local angle ( $\theta'$ ) against time

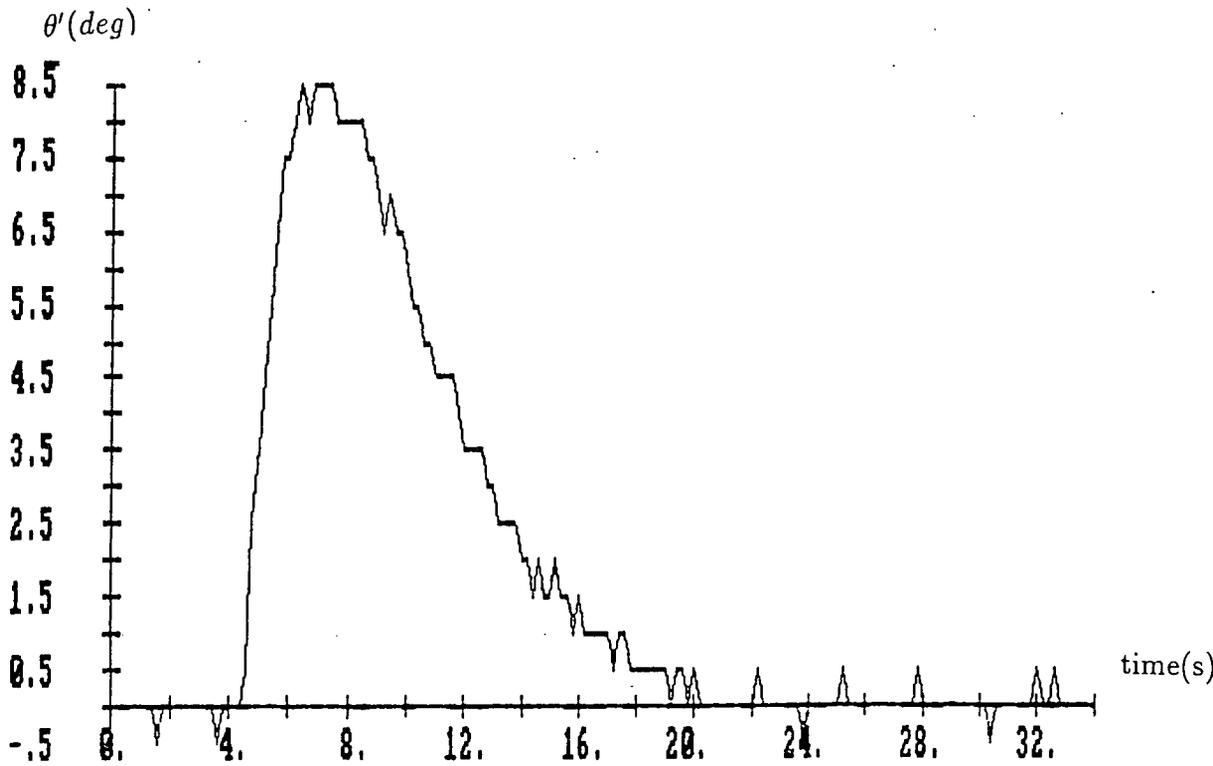


Figure 7.3d Graph of curvature ( $\kappa$ ) against time

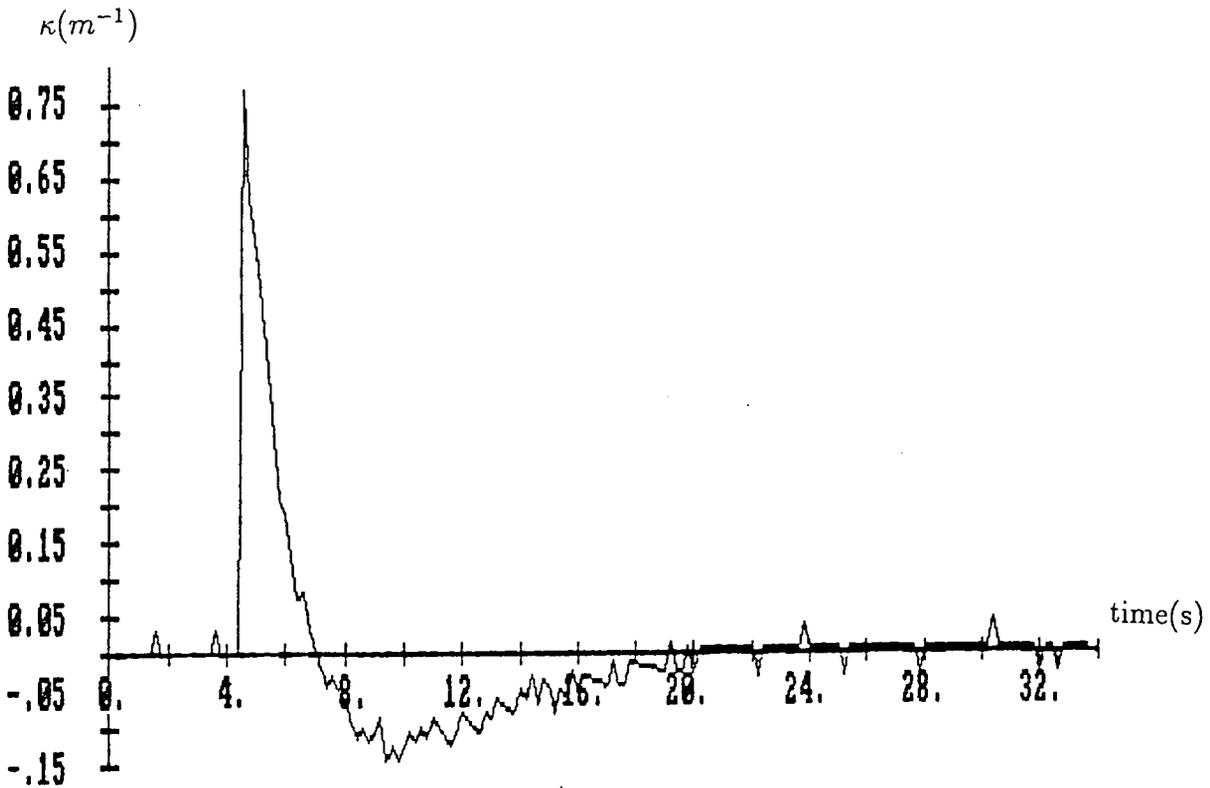
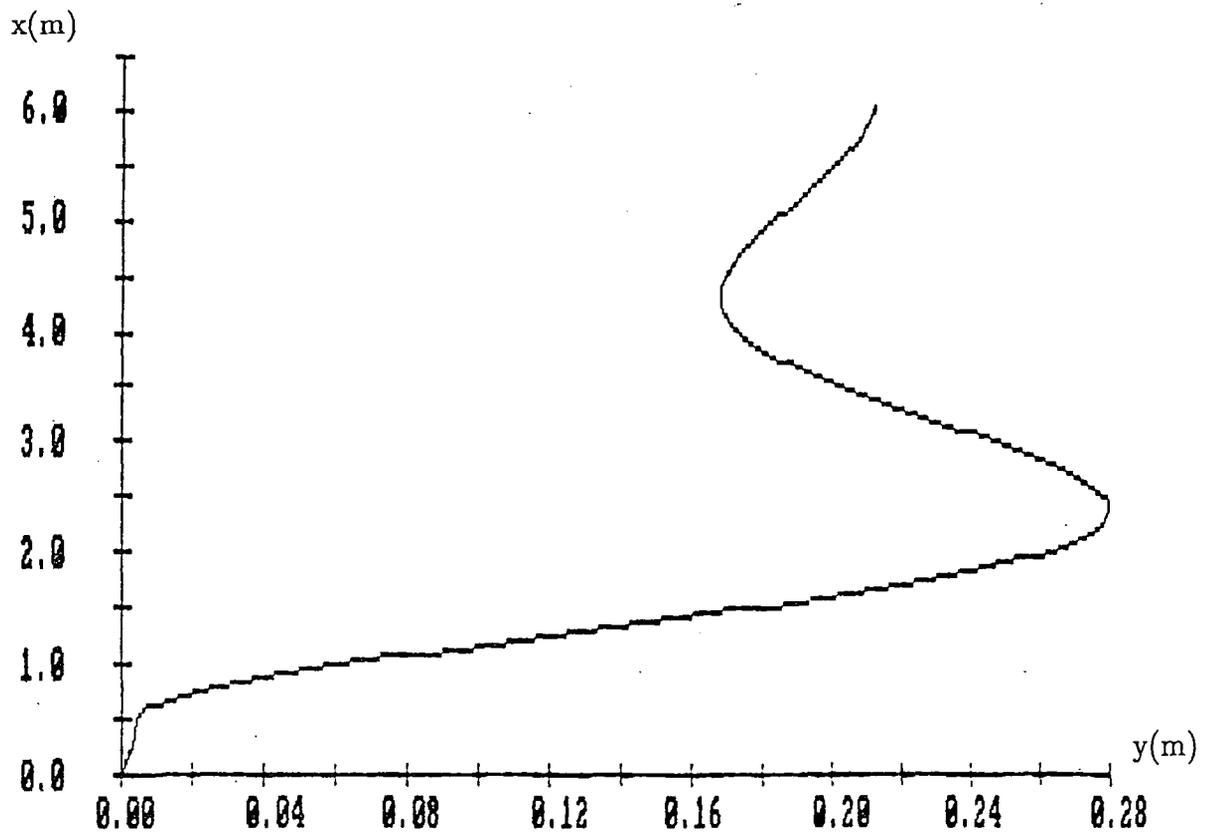


Figure 7.3 Critically damped response:  $k_p = 4$ ,  $k_{pd} = 1$ ,  $V = 0.2m/s$

Figure 7.4a AGV path in a step response test



distance

Figure 7.4b Graph of distance error against time

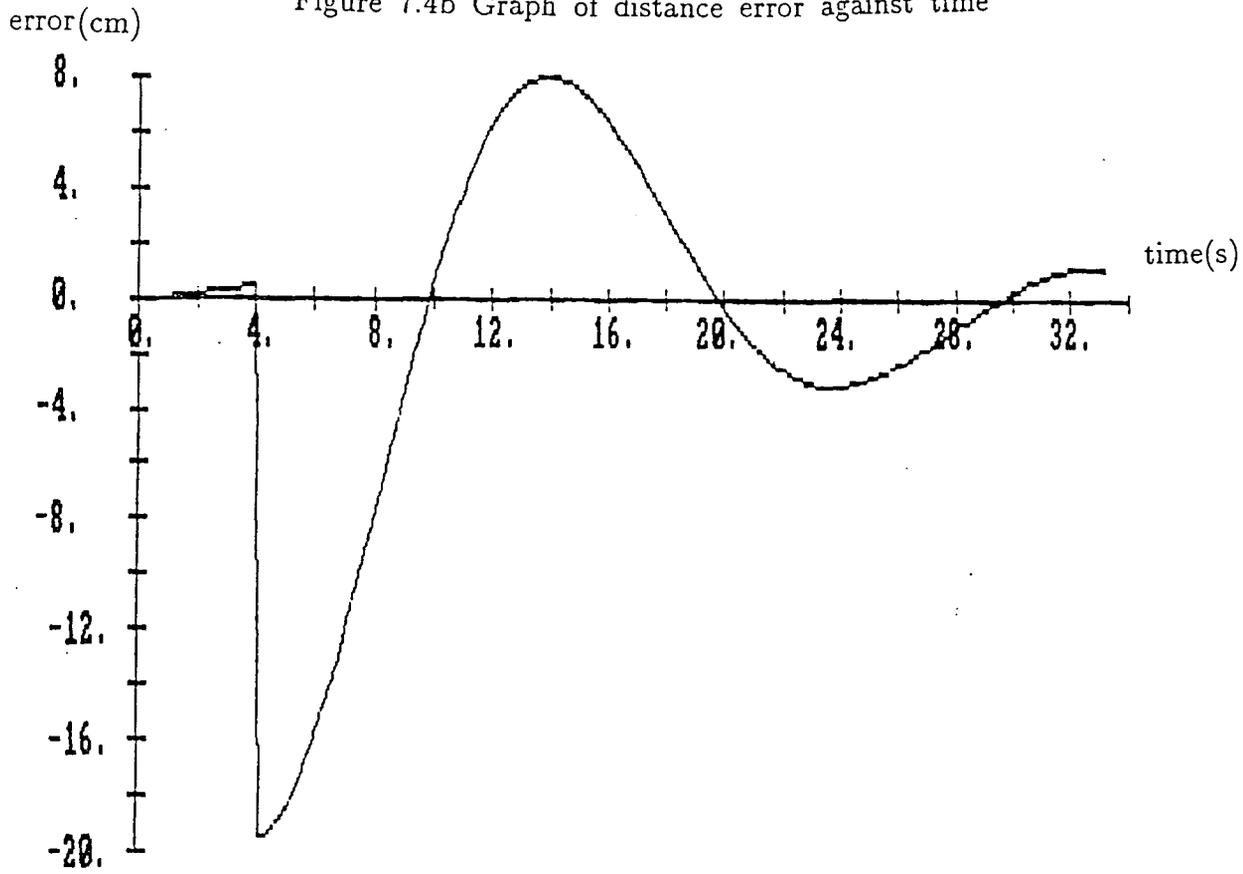
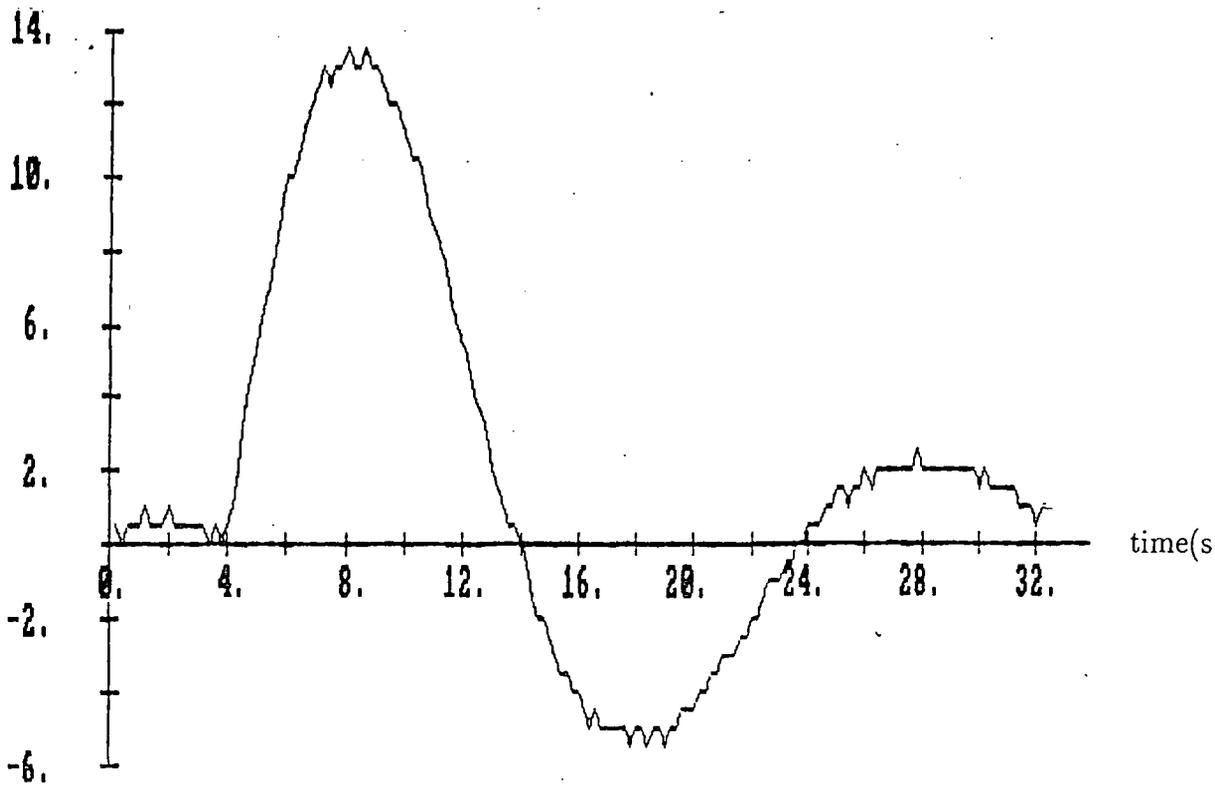


Figure 7.4 Underdamped response:  $k_p = 1$ ,  $k_{pd} = 3$ ,  $V = 0.2m/s$

$\theta'$ (deg)

Figure 7.4c Graph of local angle. ( $\theta'$ ) against time



$\kappa(m^{-1})$

Figure 7.4d Graph of curvature ( $\kappa$ ) against time

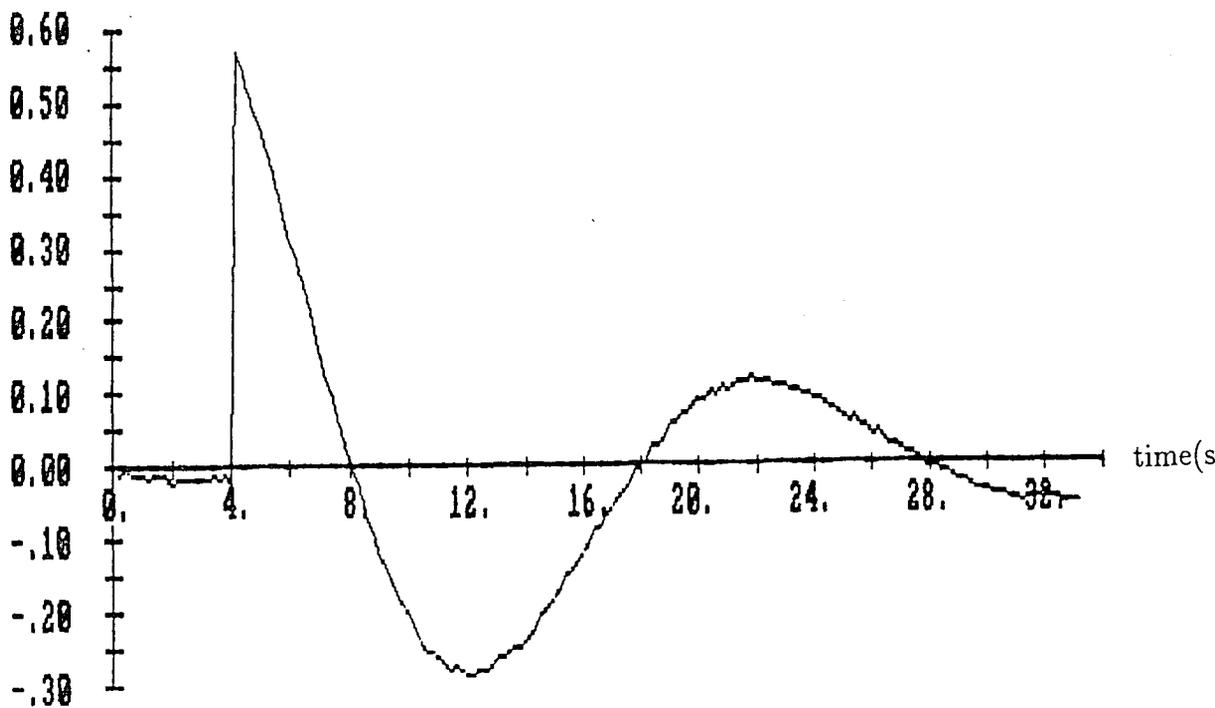


Figure 7.4 Underdamped response:  $k_p = 1$ ,  $k_{pd} = 3$ ,  $V = 0.2m/s$

distance  
error( m)

Figure 7.5a Graph of distance error against time

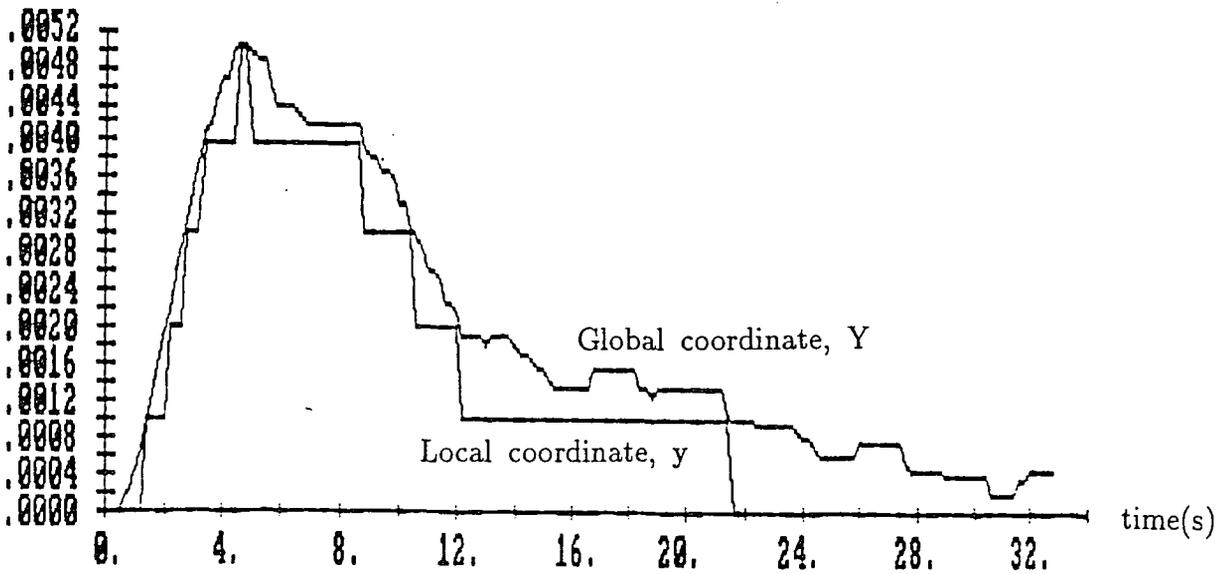


Figure 7.5b Graph of local angle ( $\theta'$ ) against time

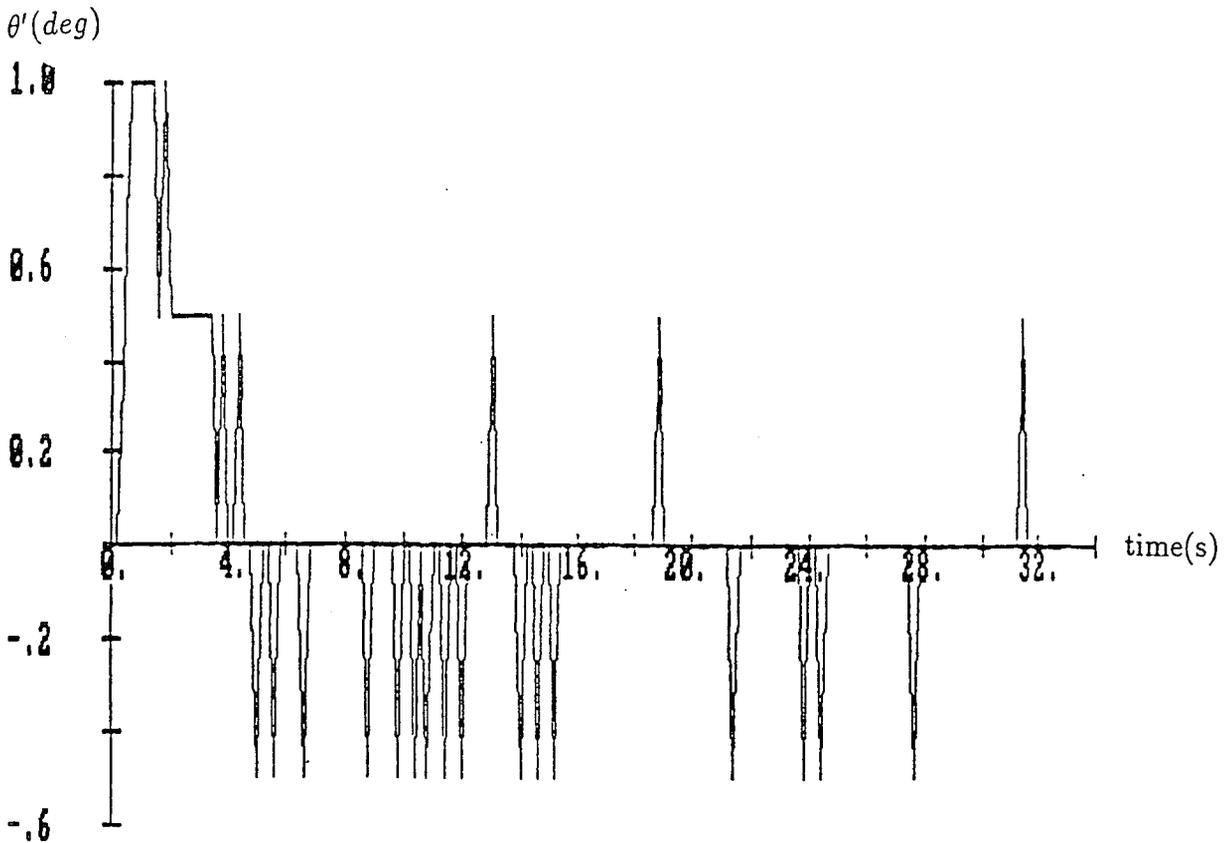


Figure 7.5 Steady state response

Figure 7.6a AGV path in a curve tracking test

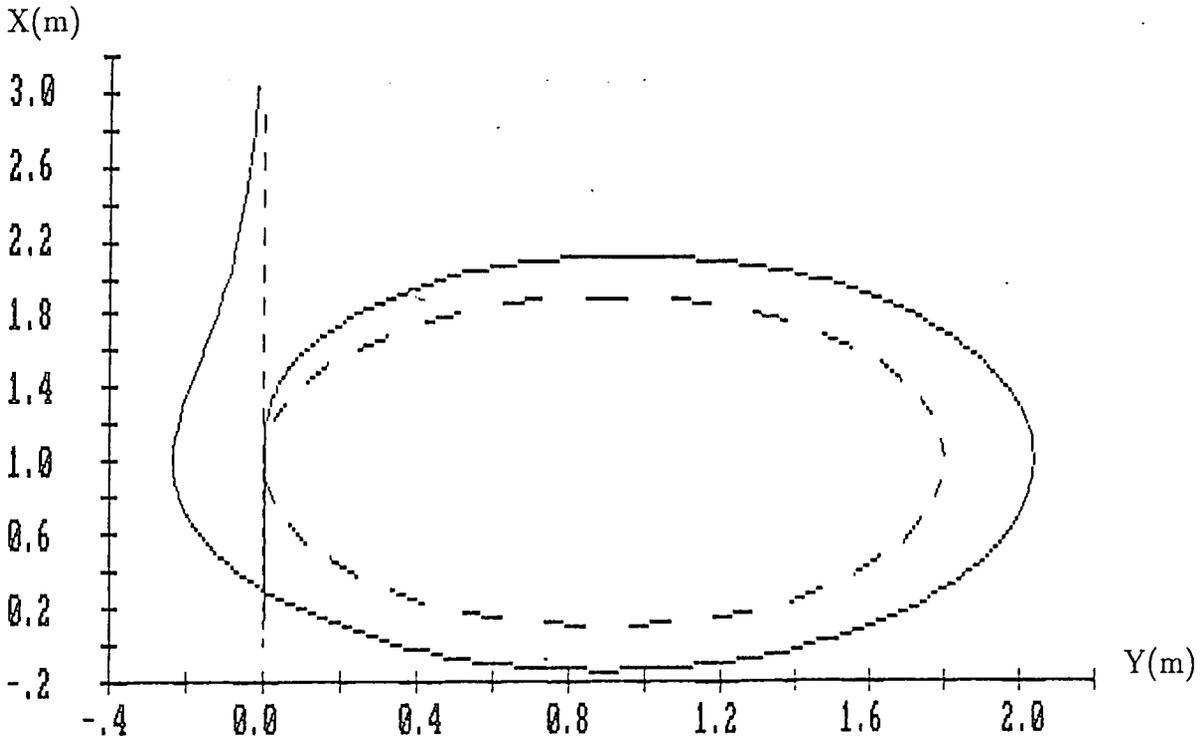


Figure 7.6b Graph of distance error against time

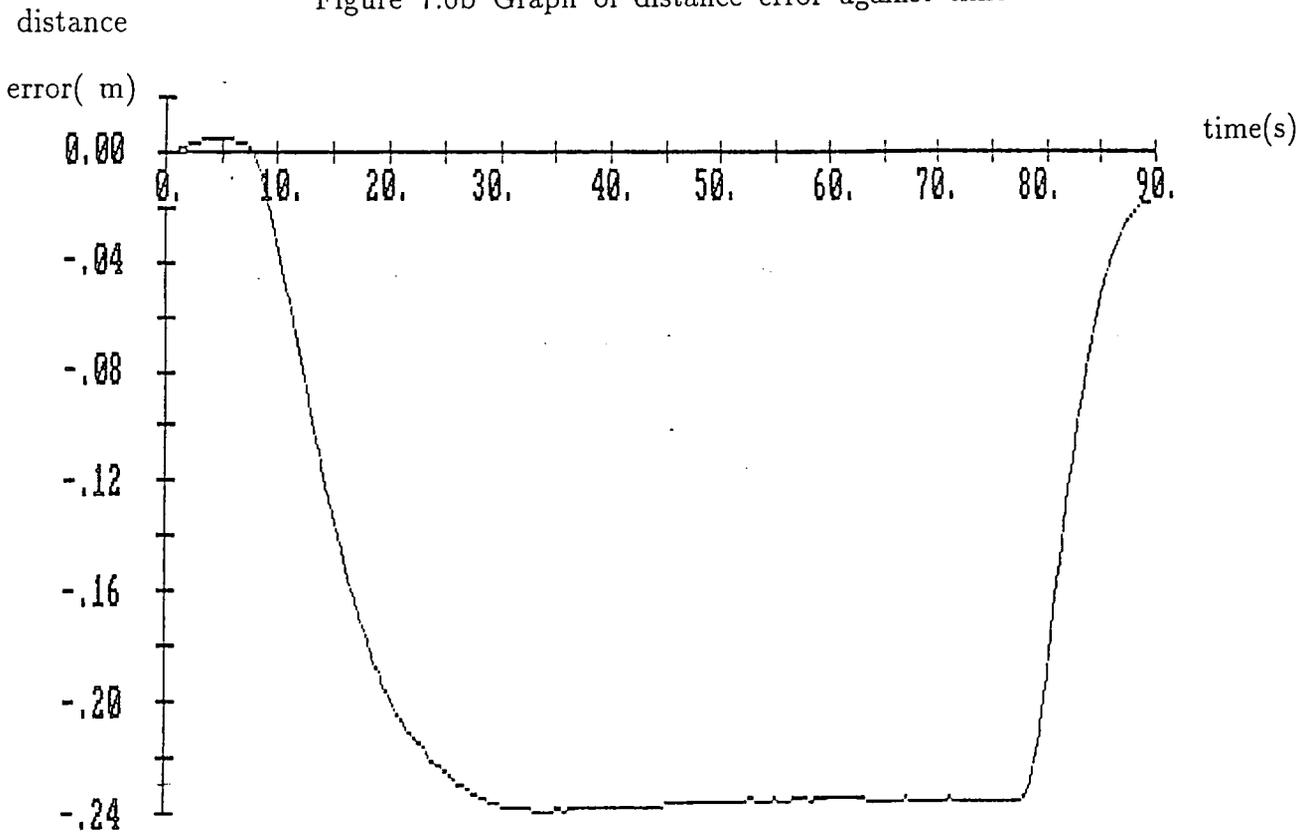


Figure 7.6 Curve tracking test

$$k_p = 4, k_{pd} = 1, V = 0.1m/s$$

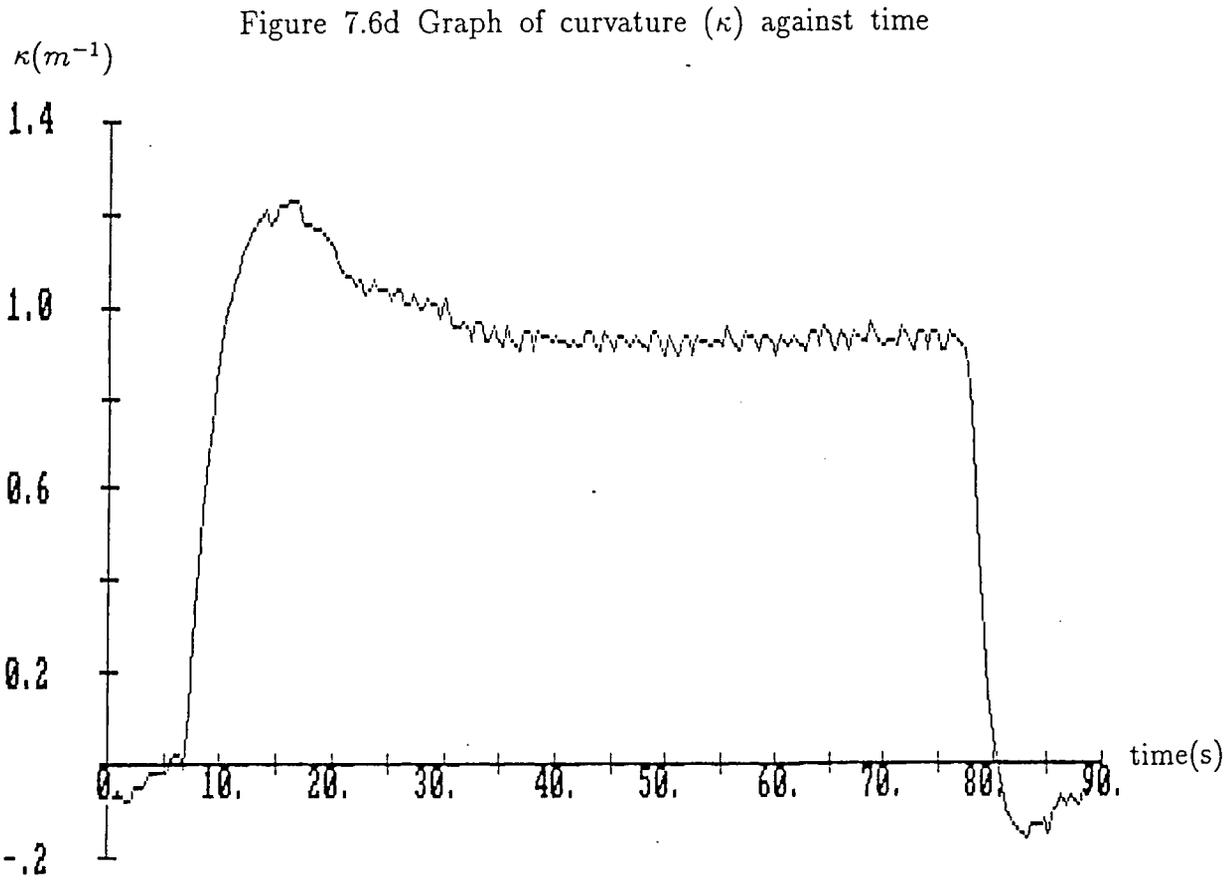
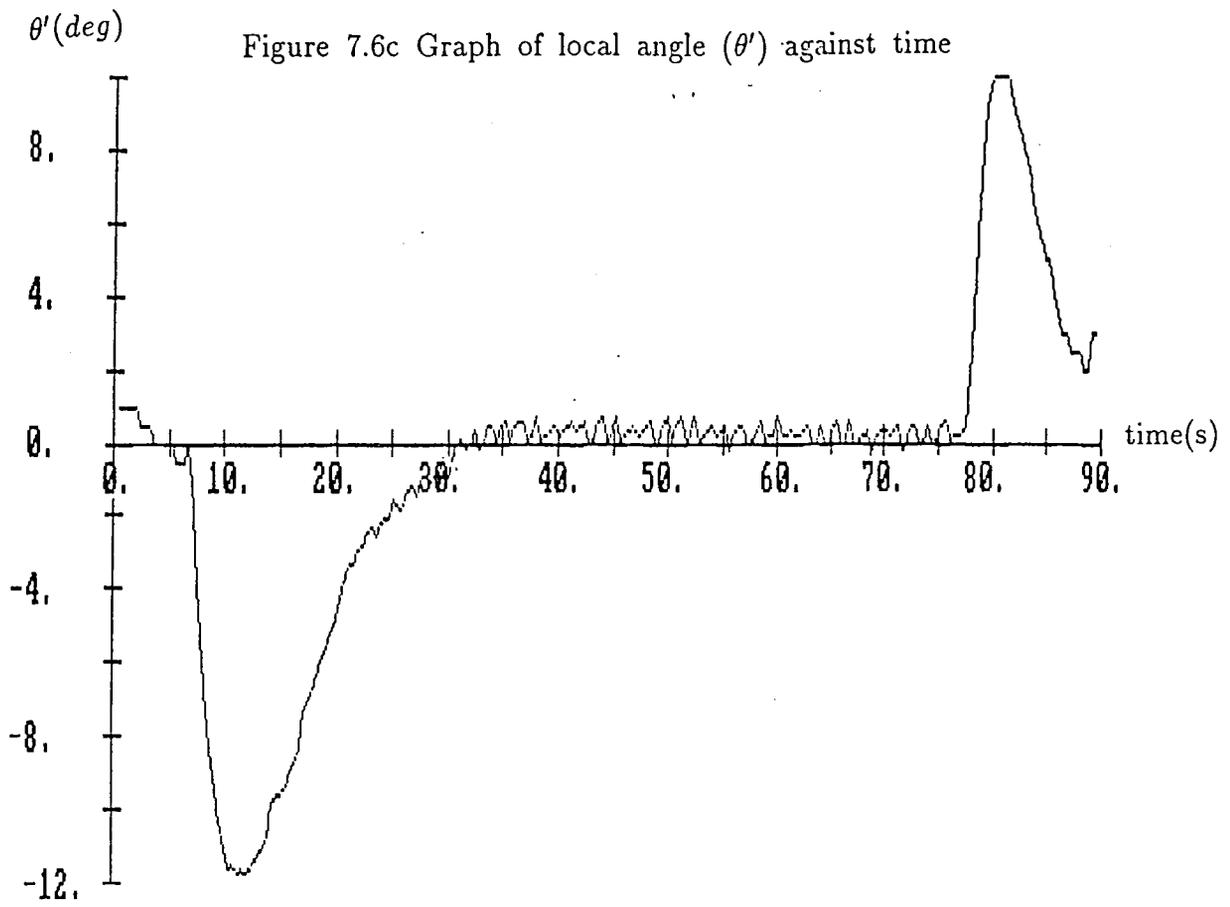
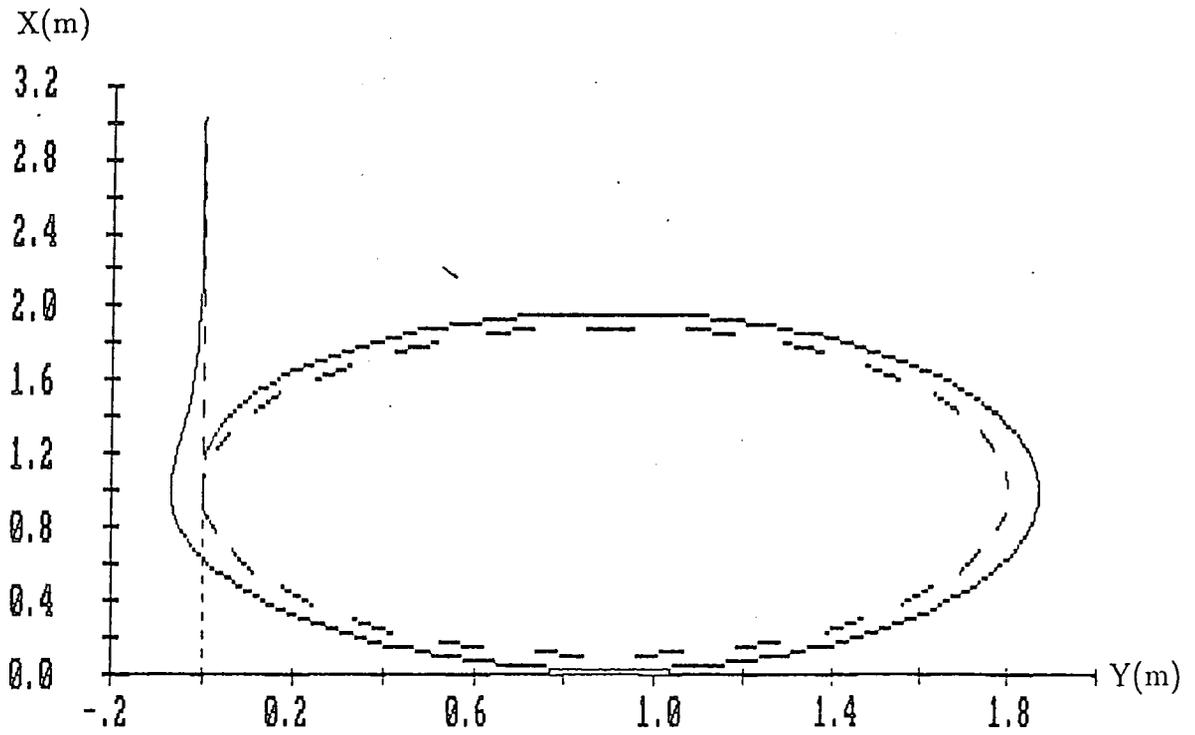


Figure 7.6 Curve tracking test

$$k_p = 4, k_{pd} = 1, V = 0.1m/s$$

Figure 7.7a AGV path in a curve tracking test



distance  
error( m)

Figure 7.7b Graph of distance error against time

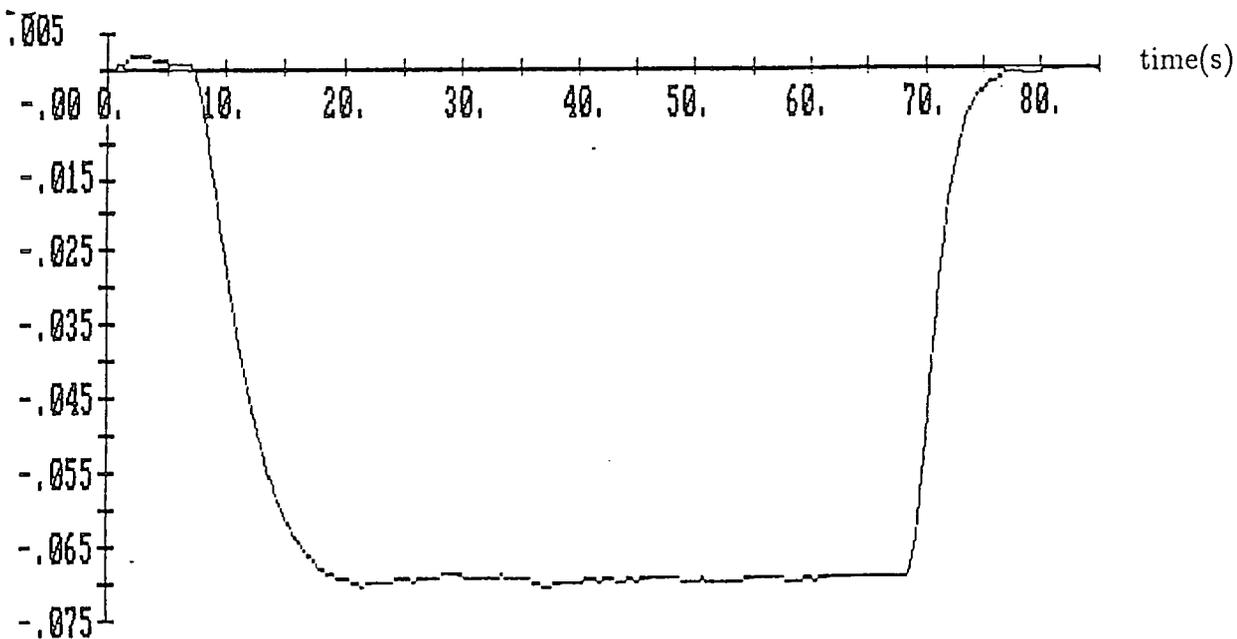


Figure 7.7 Curve tracking test

$$k_p = 8, k_{pd} = 2, V = 0.1m/s.$$

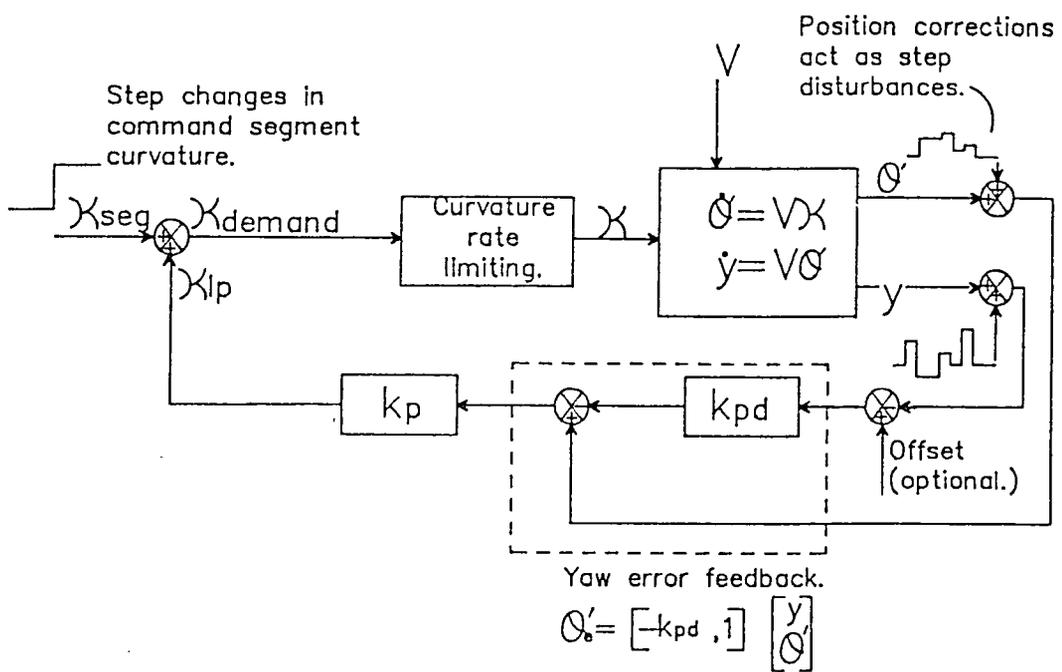
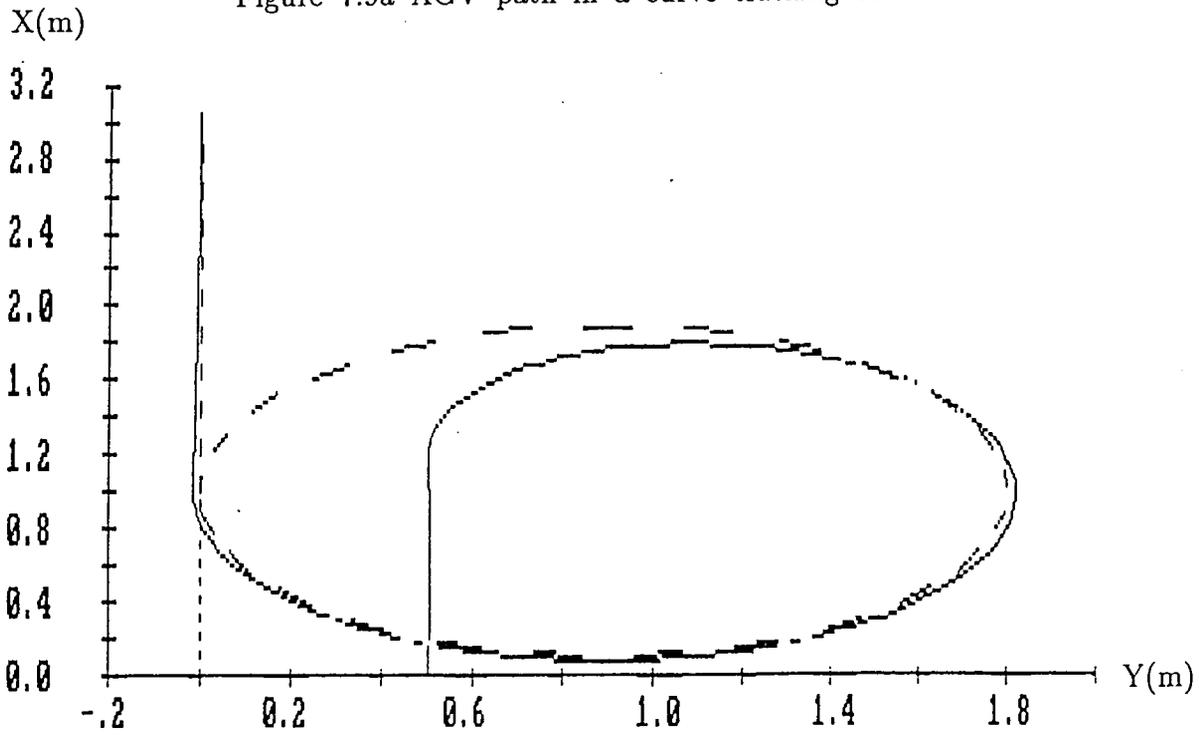


Figure 7.8 The modified control system

Figure 7.9a AGV path in a curve tracking test



distance

Figure 7.9b Graph of distance error against time

error( m)

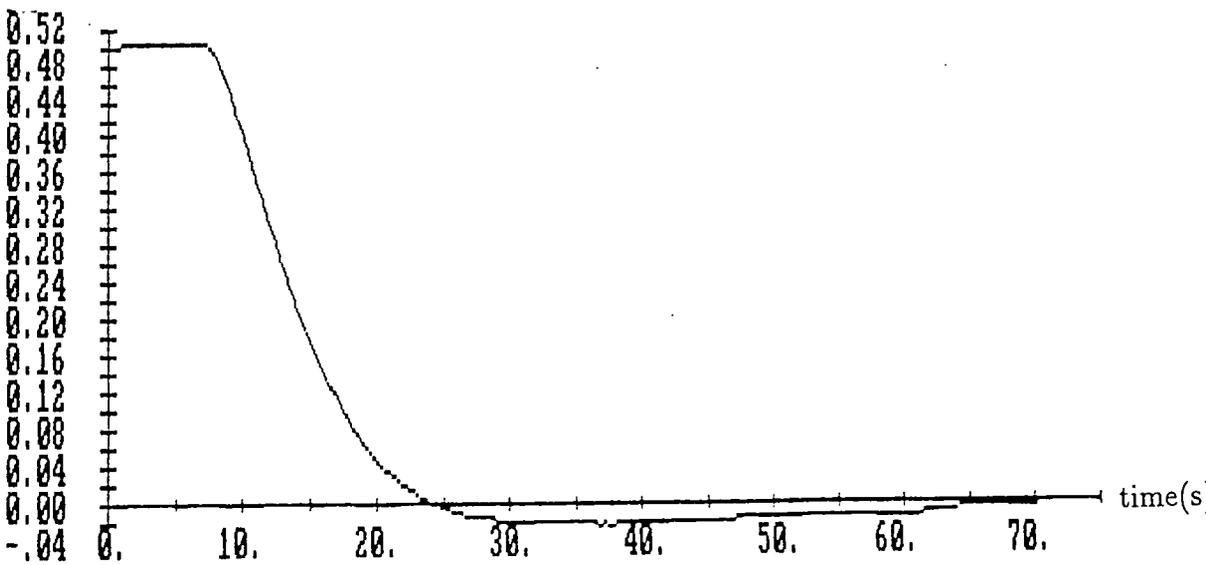


Figure 7.9 Curve tracking performance with  $\kappa_{seg}$  input

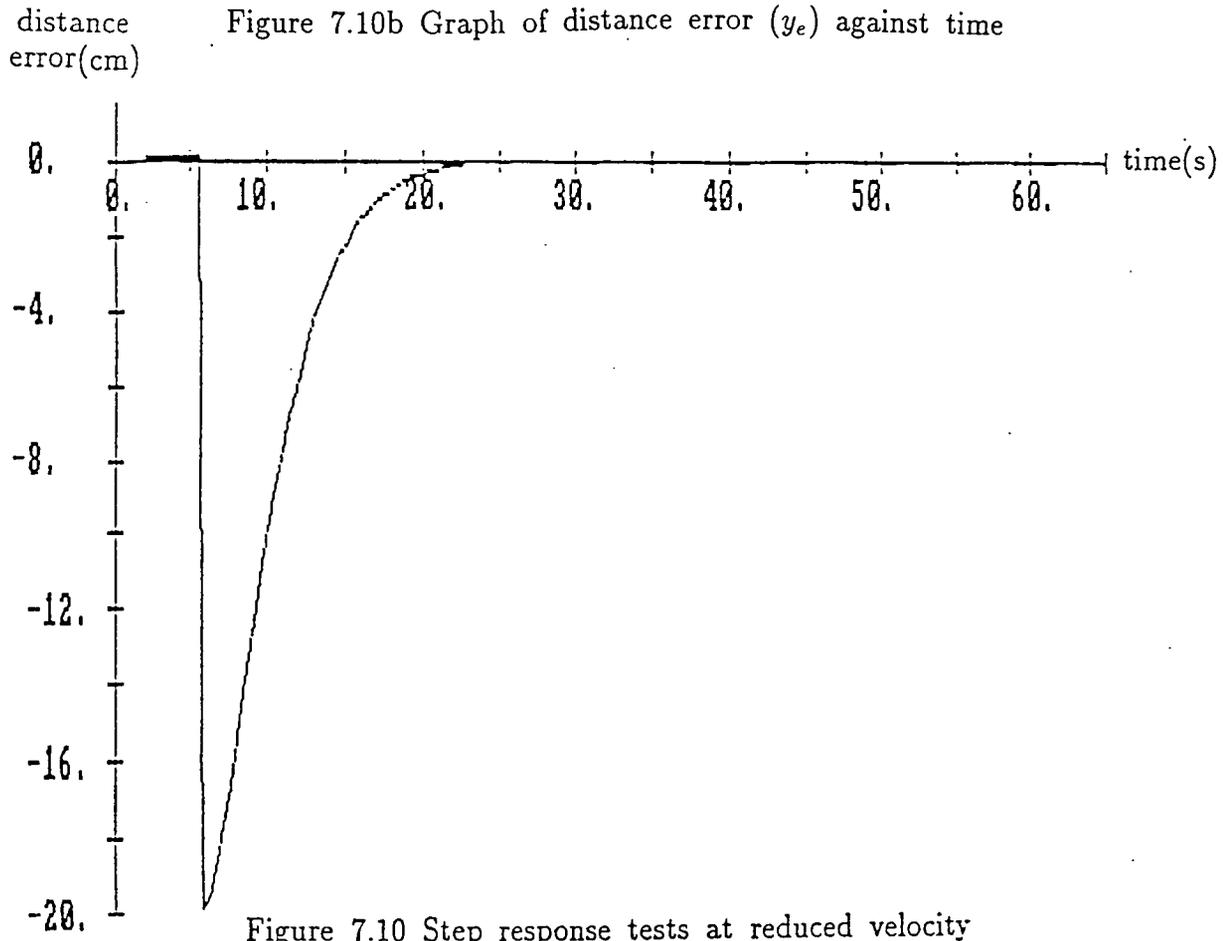
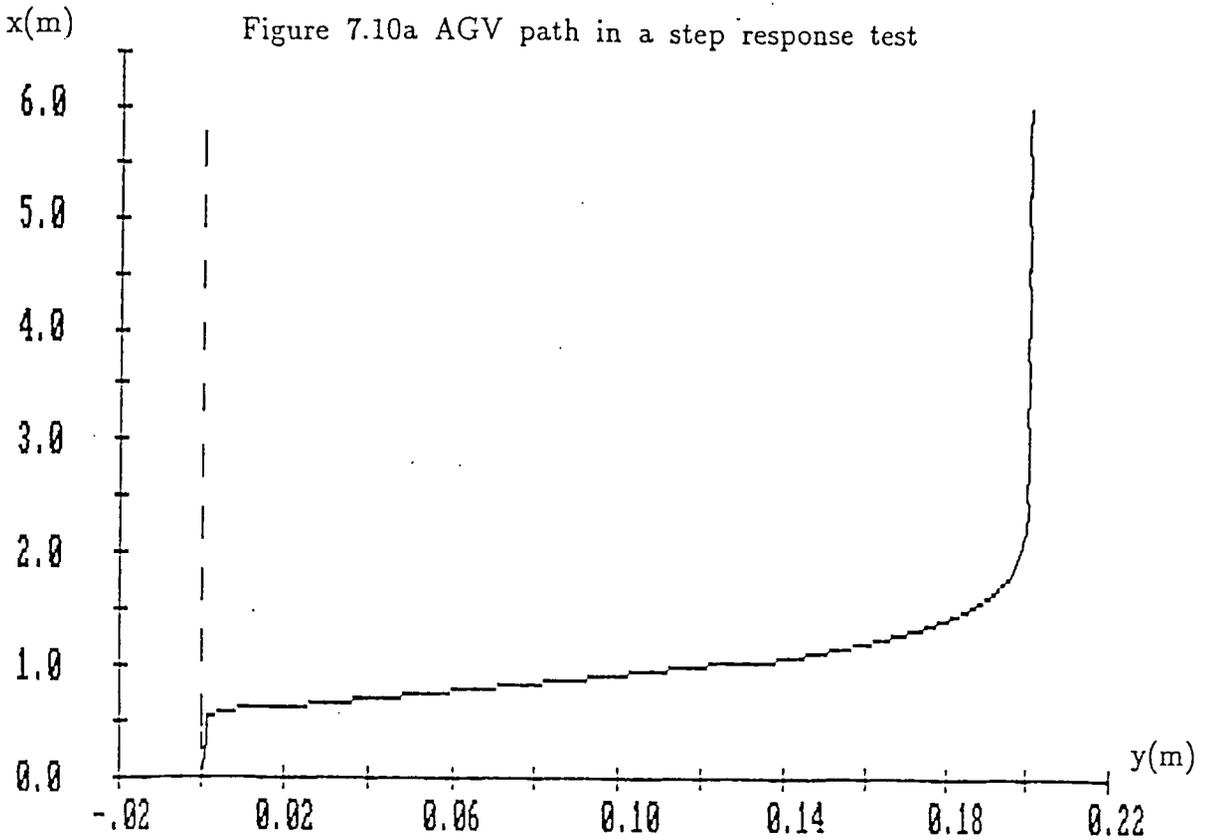


Figure 7.10 Step response tests at reduced velocity

$$k_p = 8, k_{pd} = 2, V = 0.1m/s.$$

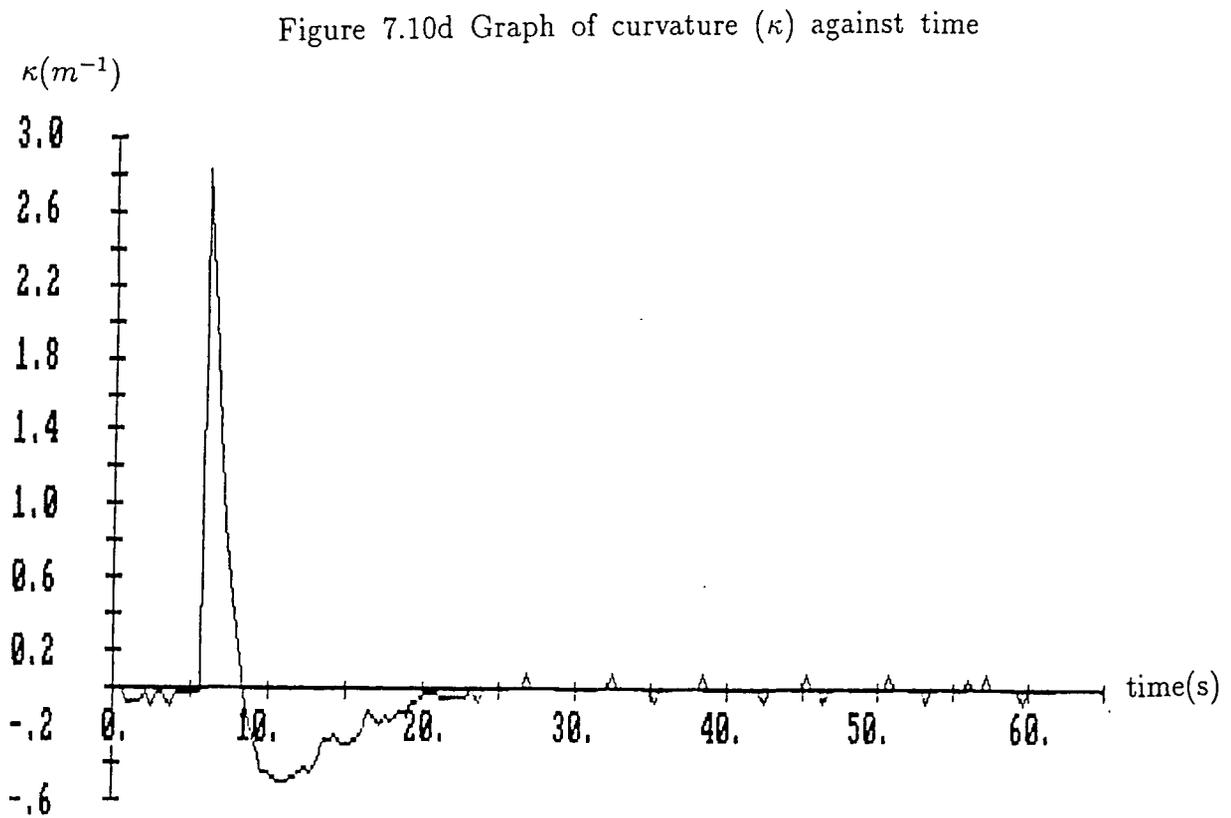
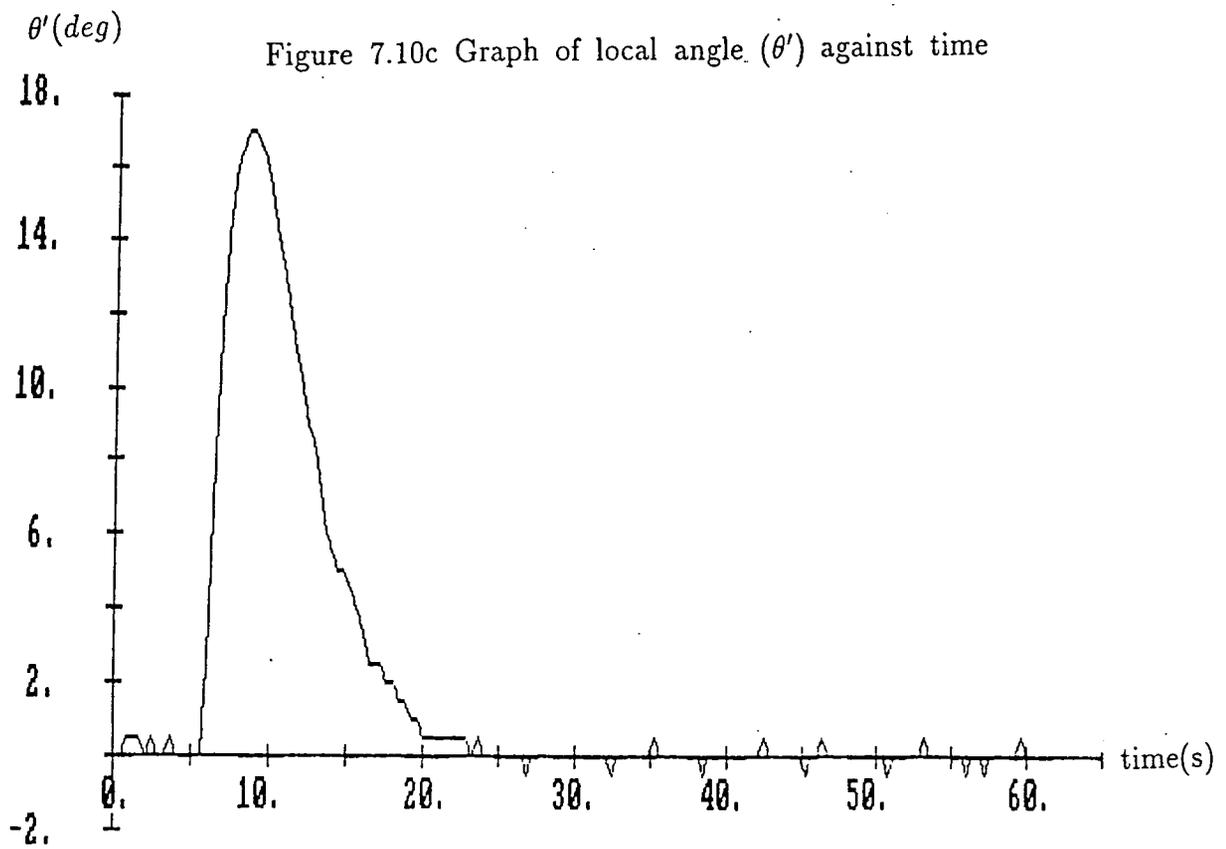


Figure 7.10 Step response tests at reduced velocity

$$k_p = 8, k_{pd} = 2, V = 0.1m/s.$$

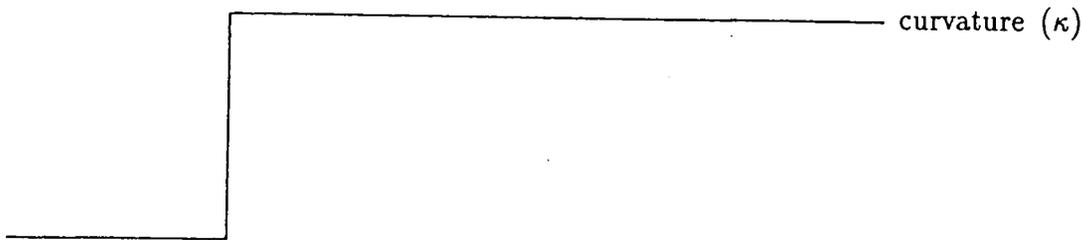


Figure 7.11a No curvature rate limiting

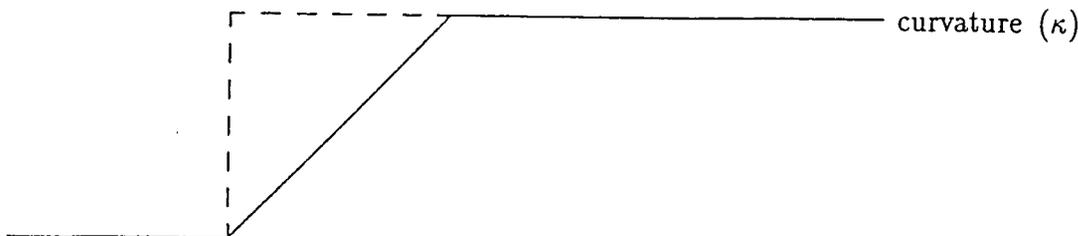


Figure 7.11b Non linear curvature rate limiting

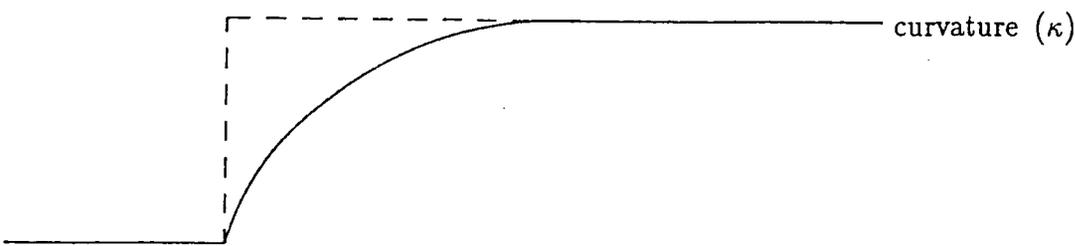


Figure 7.11c Linear curvature rate limiting

Figure 7.11 Methods of curvature rate limiting

Figure 7.12a Graph of distance error against time

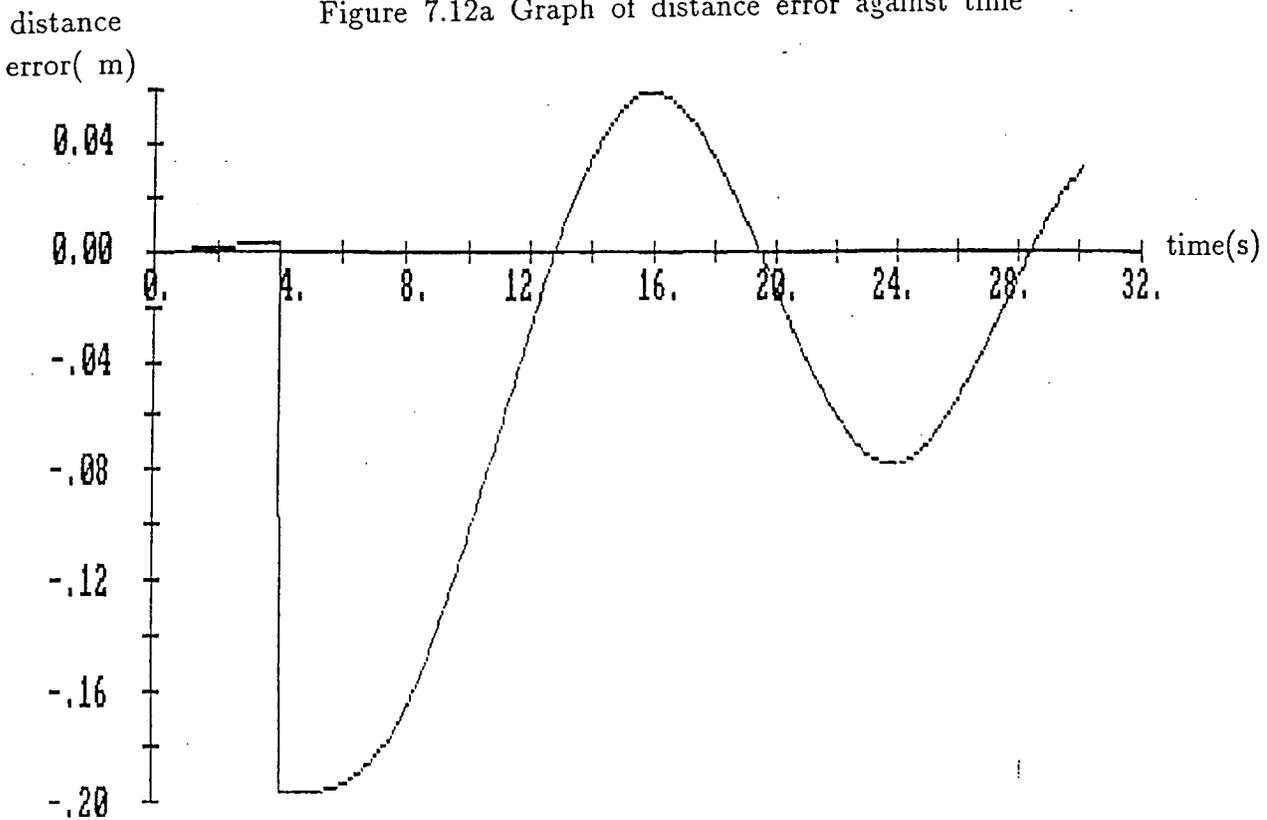


Figure 7.12b Graph of curvature ( $\kappa$ ) against time

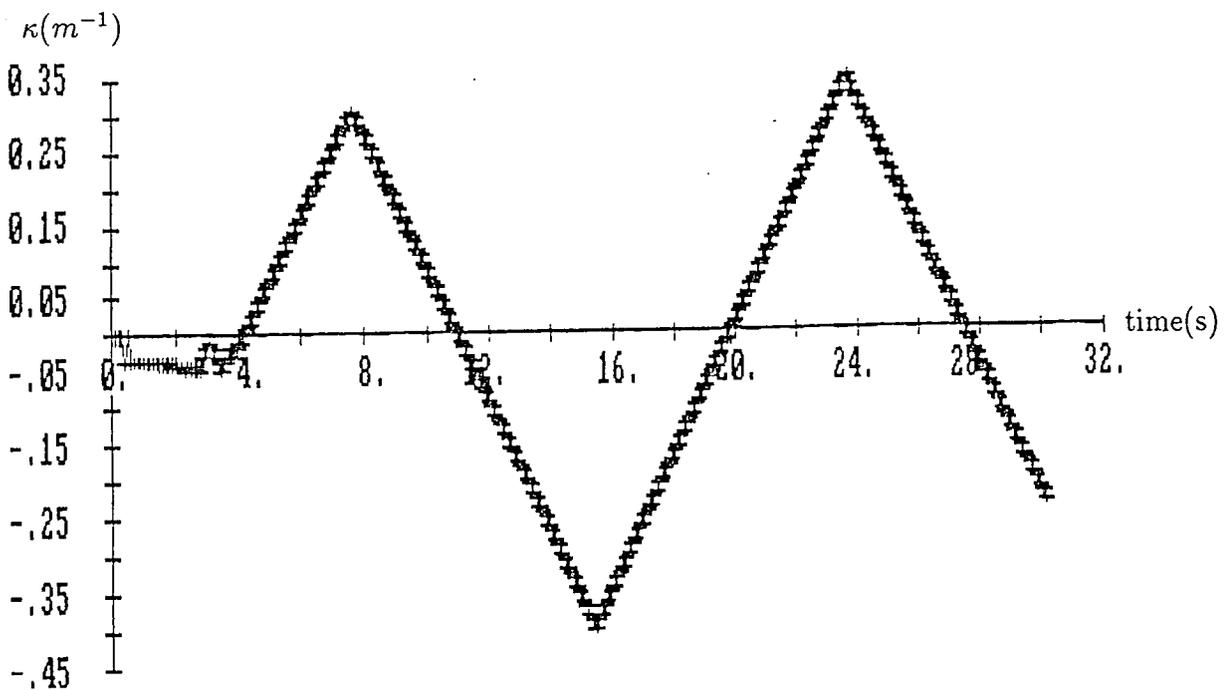


Figure 7.12 Non-linear curvature rate limiting

Curvature rate limited to  $0.09m^{-1}s^{-1}$

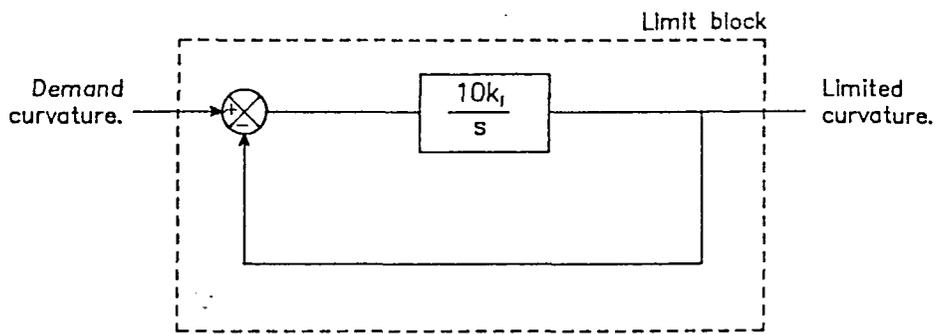


Figure 7.13a  $s$ -plane representation of the limiting block

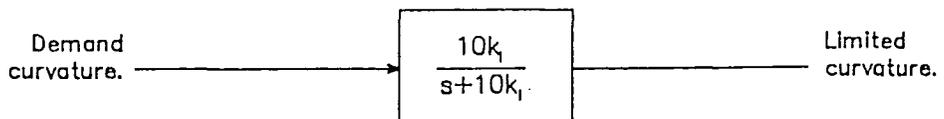


Figure 7.13b Reduced form of limiting block

Figure 7.14a No curvature rate limiting

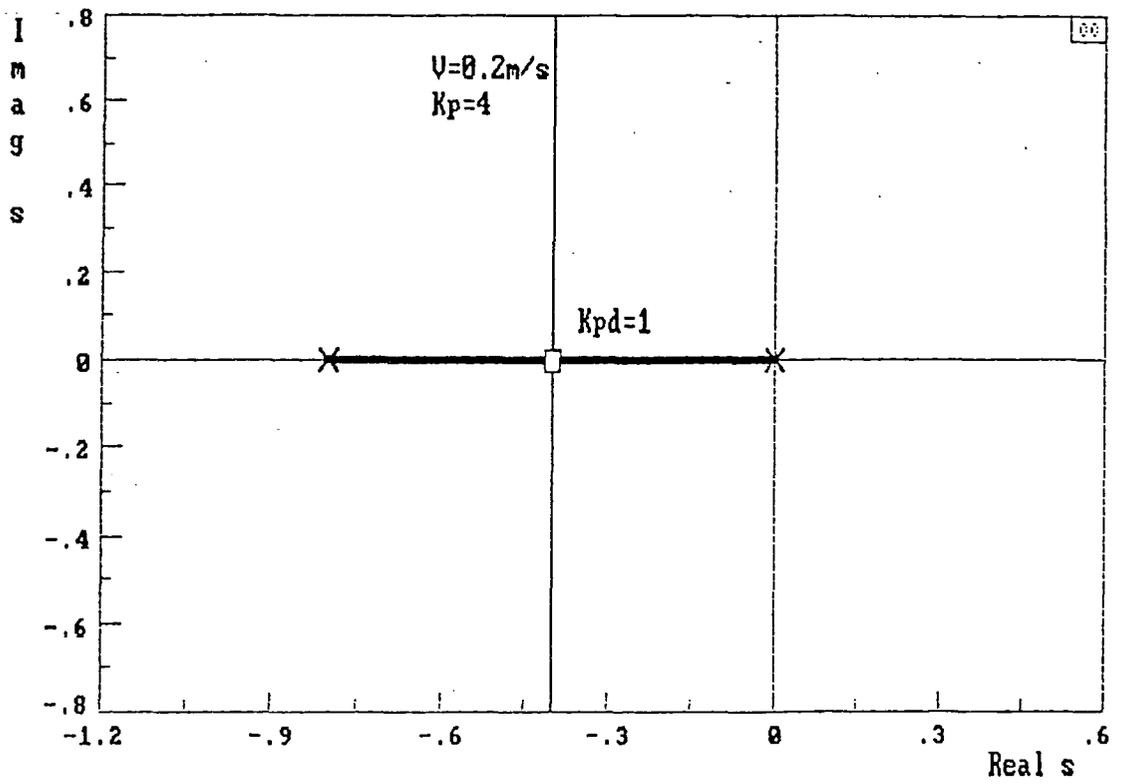


Figure 7.14b Curvature rate limited system

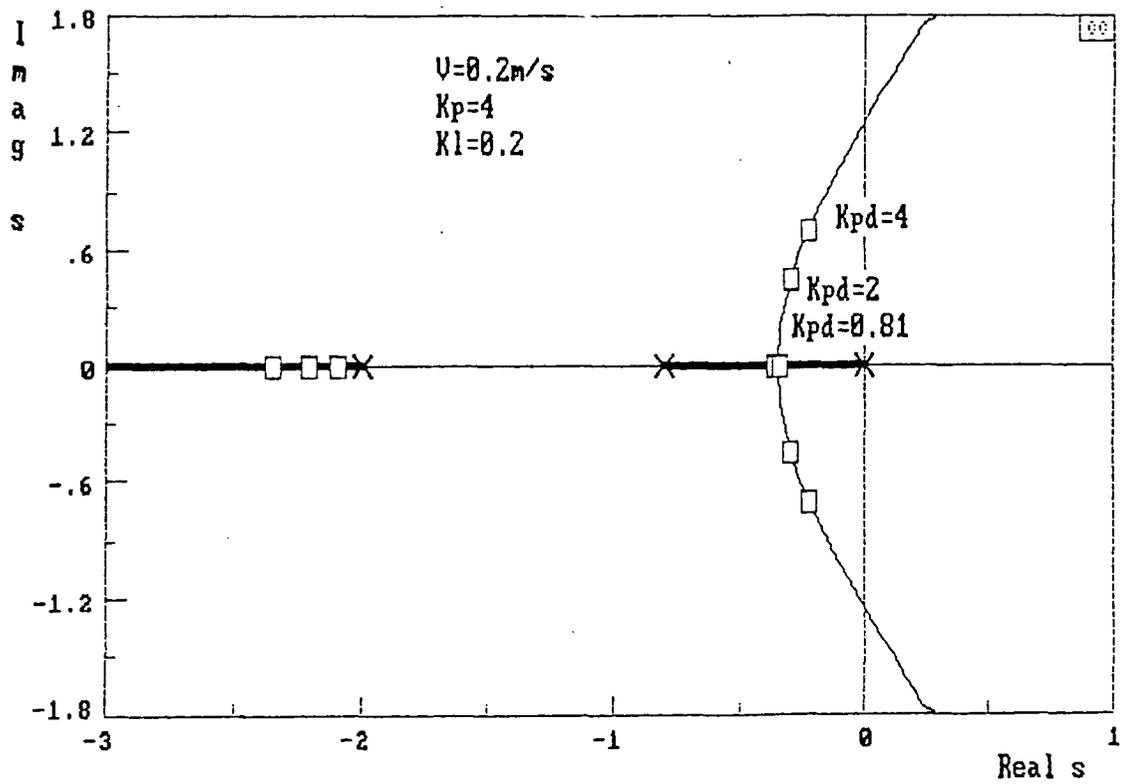


Figure 7.14 Root locus in  $s$ -domain

distance  
error( m)

Figure 7.15a Graph of distance error against time

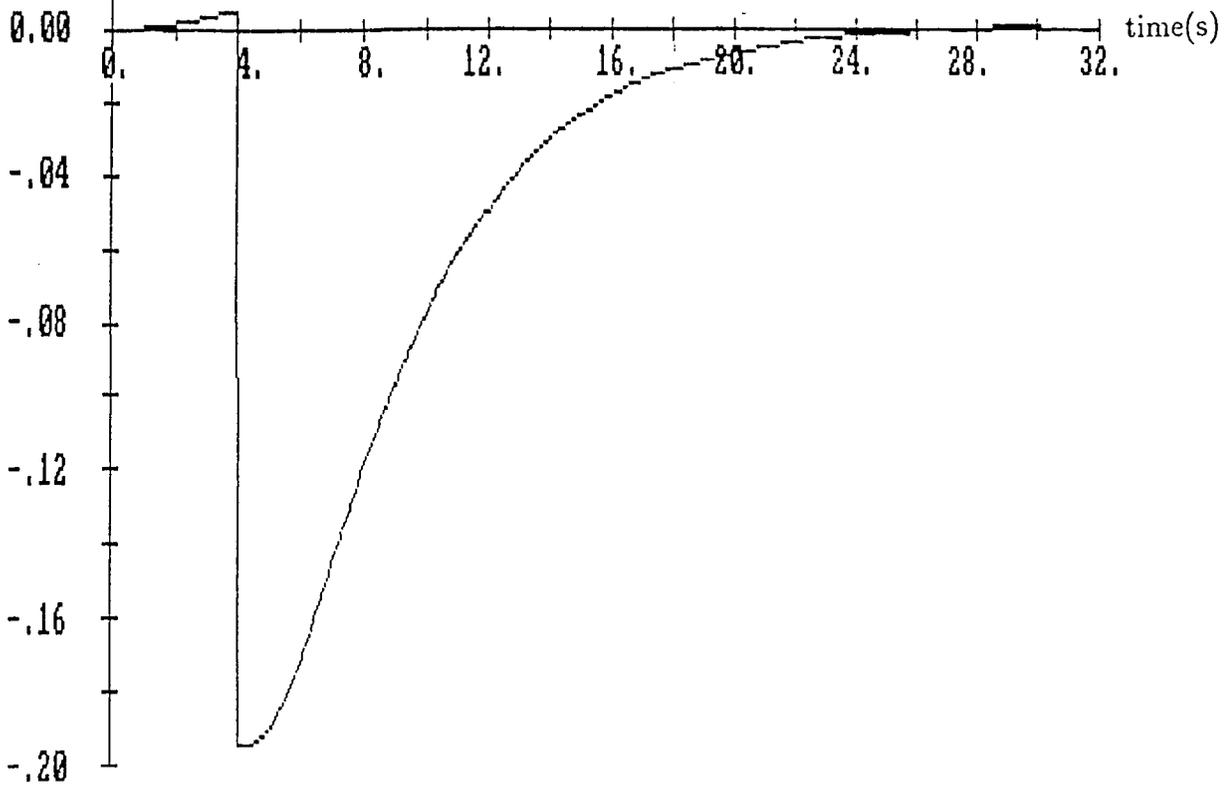


Figure 7.15b Graph of curvature ( $\kappa$ ) against time

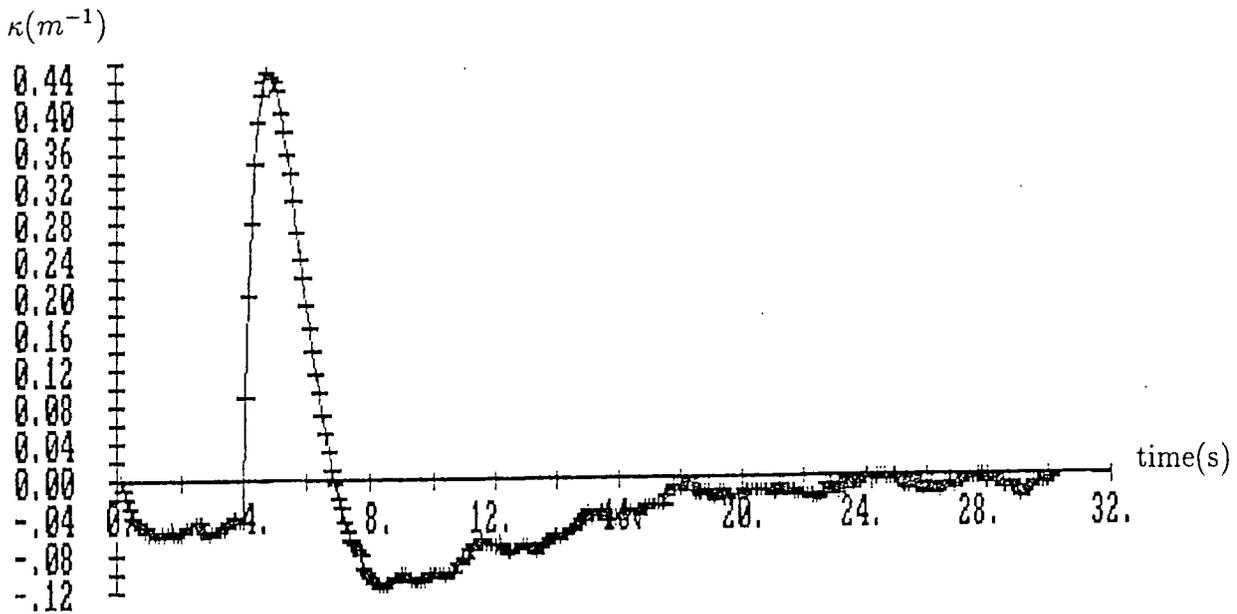
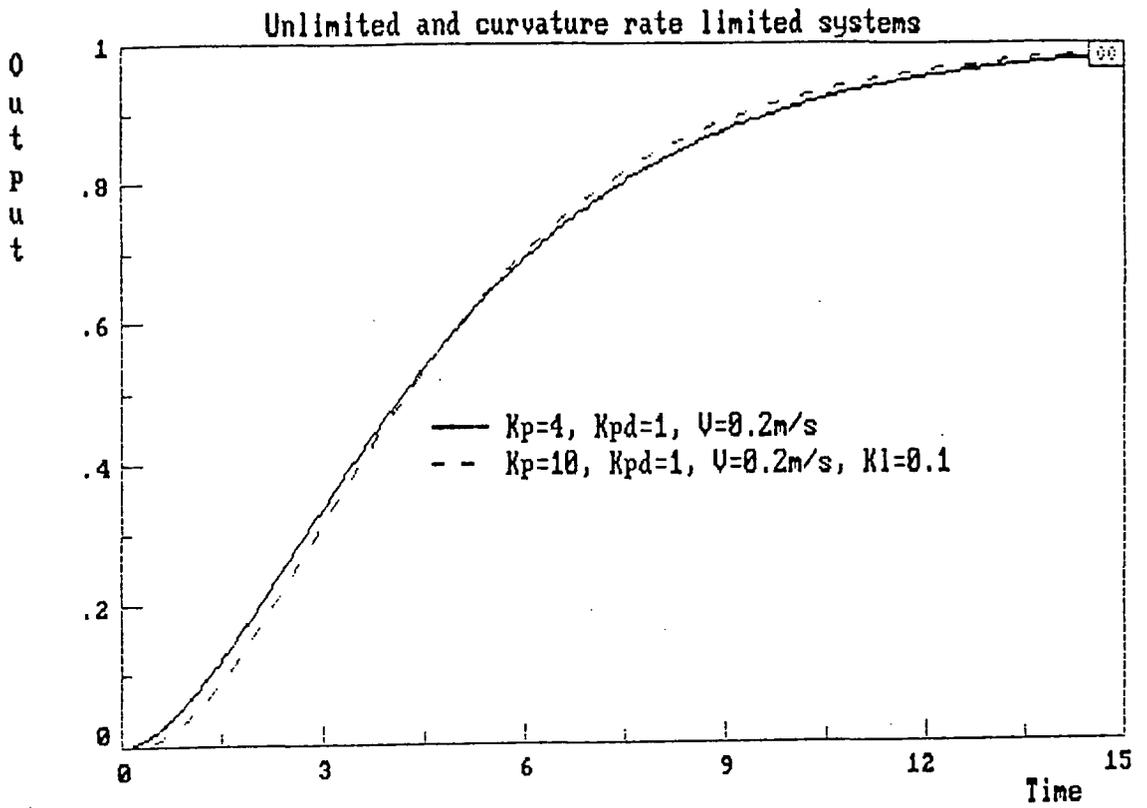


Figure 7.15 Curvature rate limited system

$$k_p = 4, k_{pd} = 0.823, k_l = 0.2, V = 0.2m/s$$

Figure 7.16a A comparison of unlimited and curvature rate limited responses



$\kappa(m^{-1})$

1.0

0.6

0.2

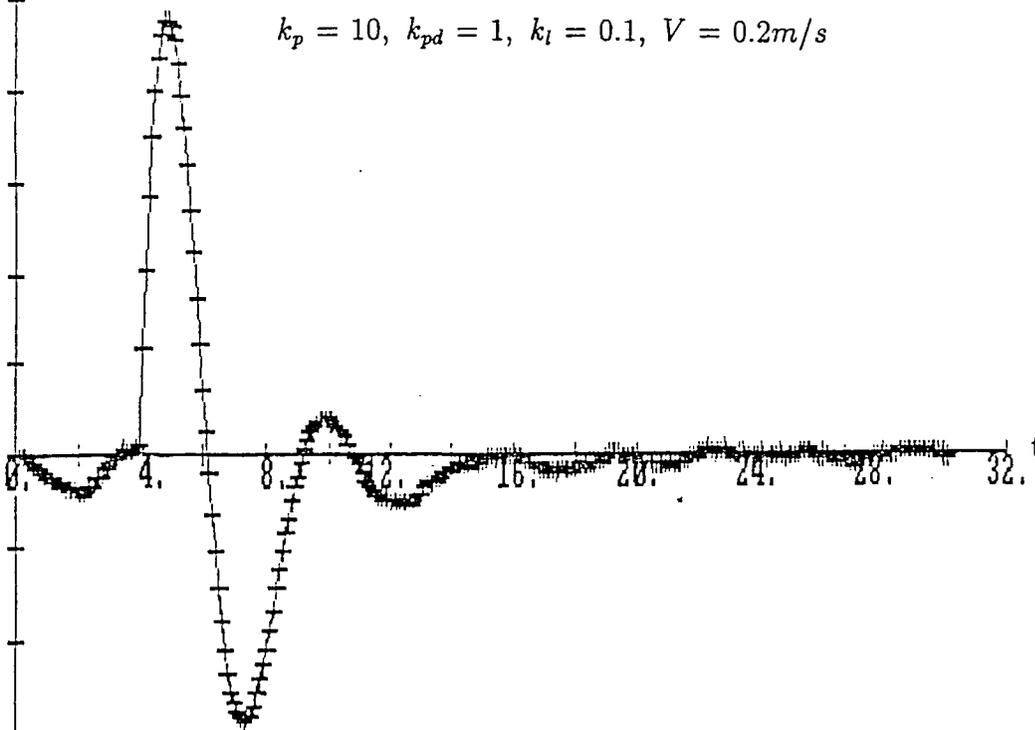
-0.2

-0.6

Figure 7.16b Graph of curvature ( $\kappa$ ) against time

$k_p = 10, k_{pd} = 1, k_i = 0.1, V = 0.2\text{m/s}$

time(s)



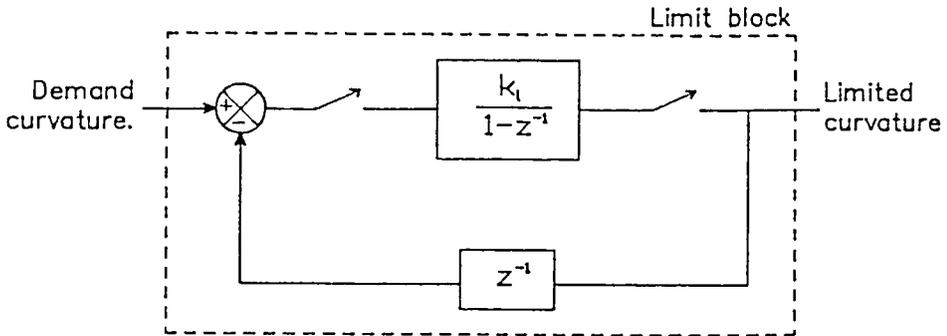


Figure 7.17a z-plane representation of the limiting block

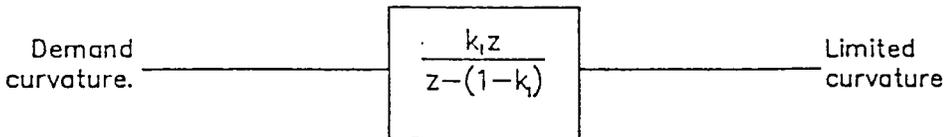


Figure 7.17b Reduced form of the limiting block

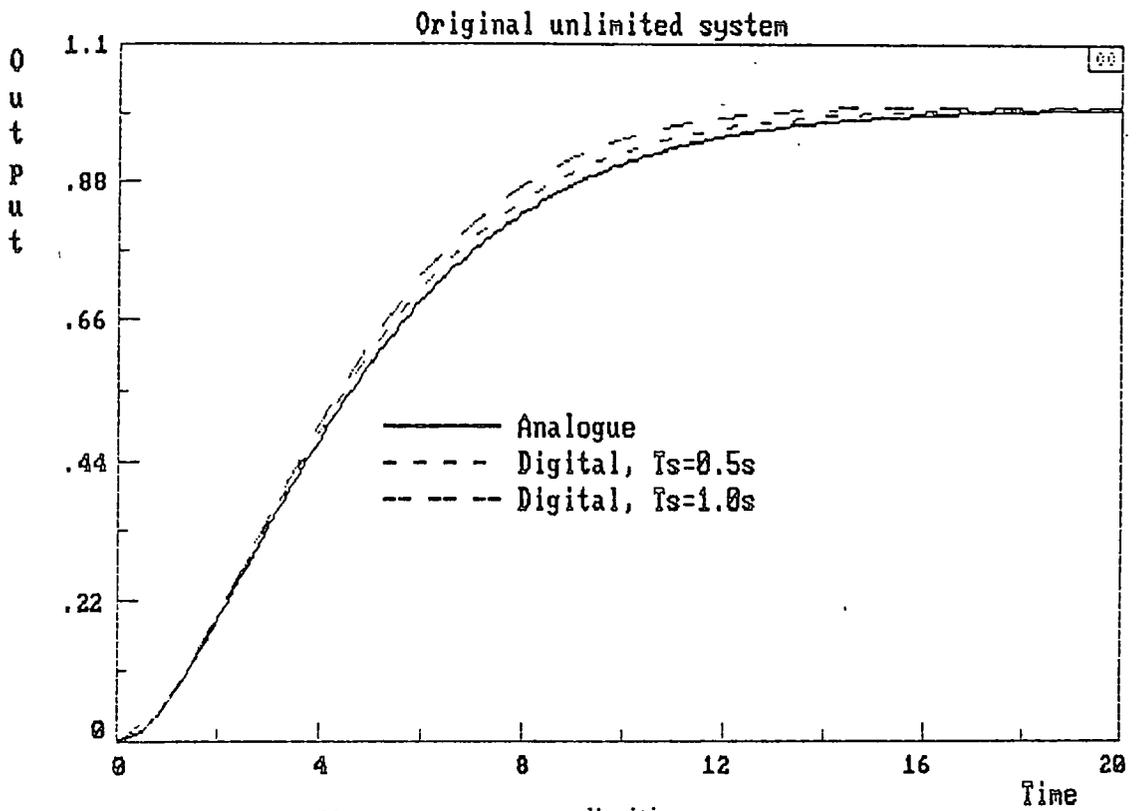


Figure 7.18a No curvature rate limiting

$$k_p = 4, k_{pd} = 1, V = 0.2m/s$$

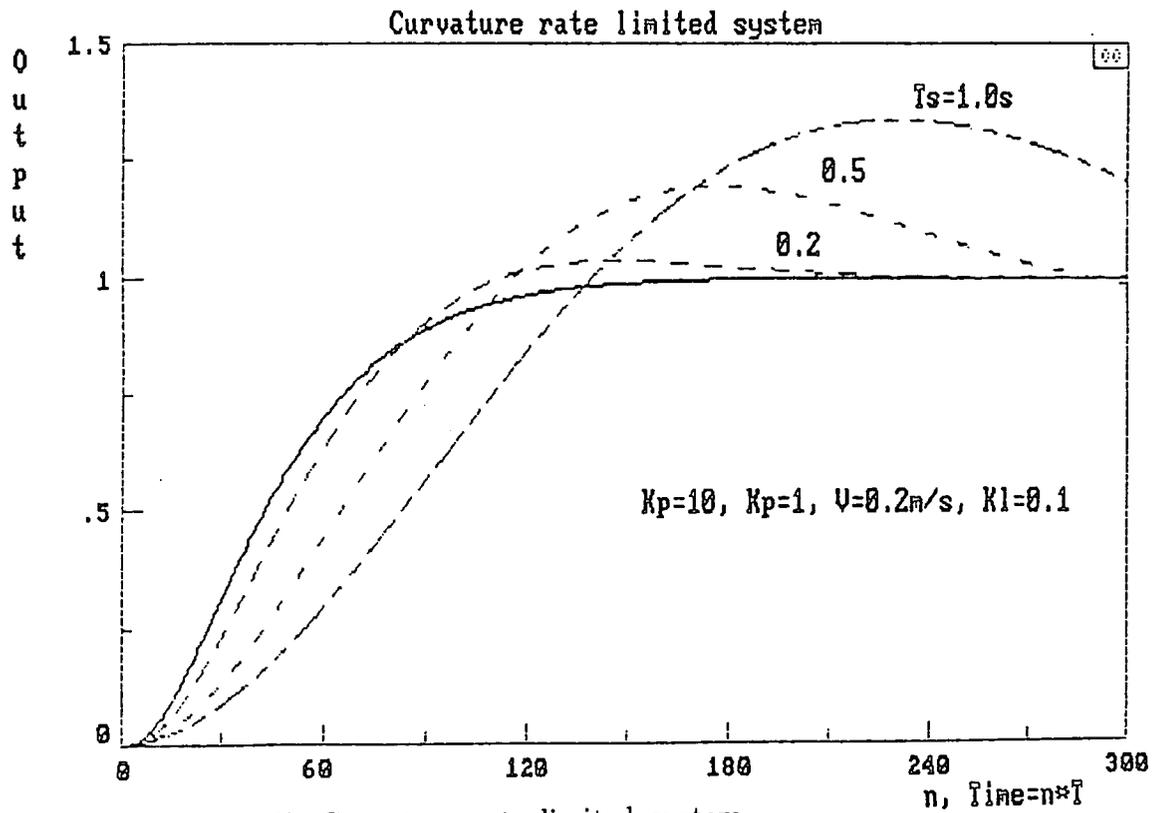


Figure 7.18b Curvature rate limited system

$$k_p = 10, k_{pd} = 1, k_l = 0.1, V = 0.2m/s$$

Figure 7.18 A comparison of responses with varying sampling interval

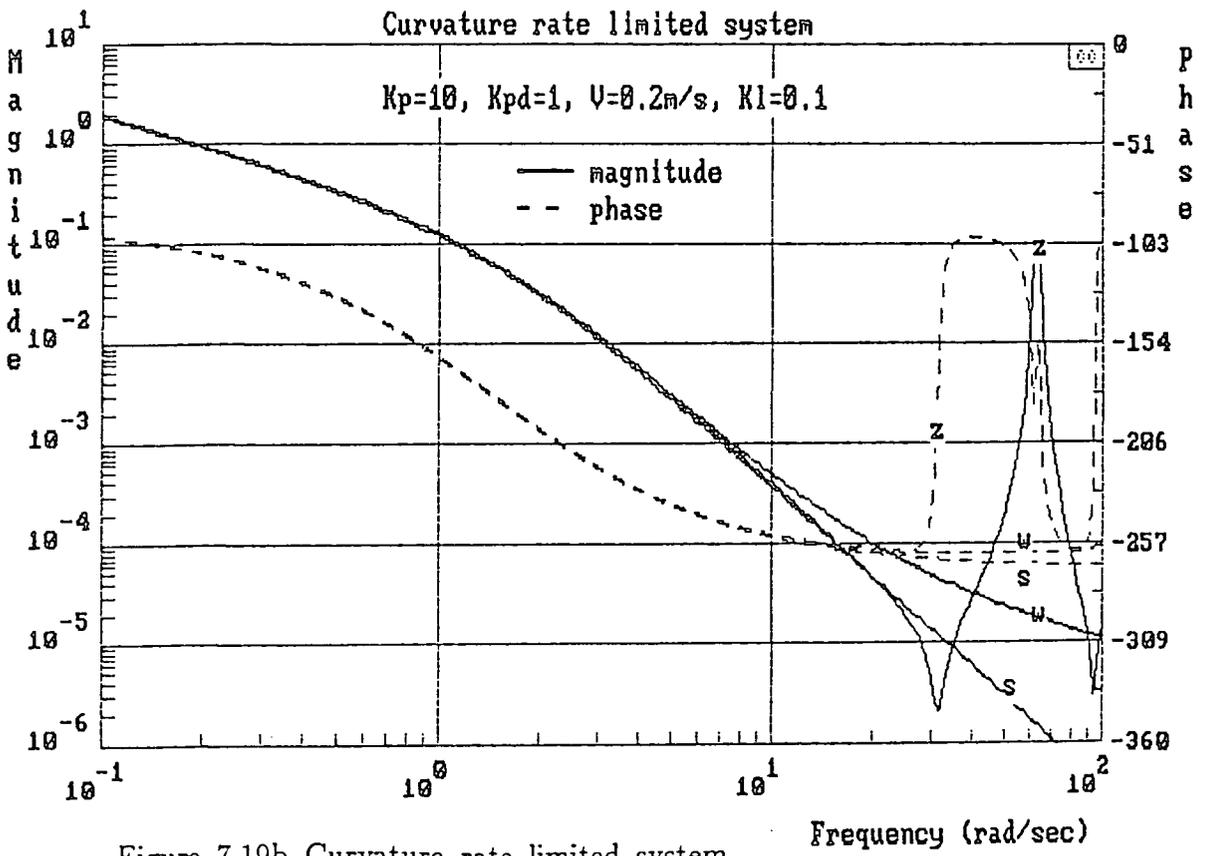
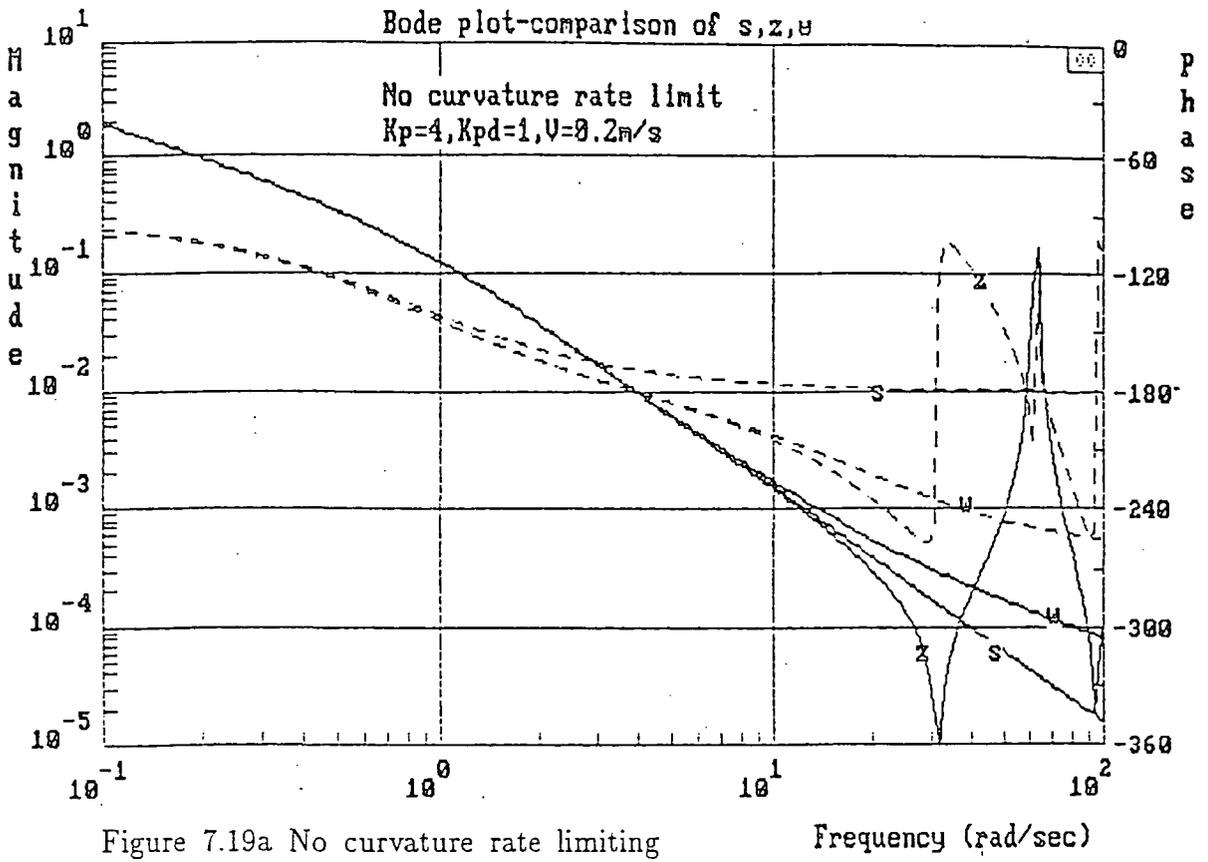


Figure 7.19 Bode plots for control system

## CHAPTER 8

### DATA PROCESSING AND THE PERFORMANCE OF THE AGV

#### 8.1 Introduction

In this chapter, the processing of ultrasonic range measurements that takes place as the AGV moves in the vicinity of a correction board is described. Initially raw ultrasonic range measurements were collected and subsequently stored on disc when the research vehicle was asked to follow a demand path parallel to a correction board. Analysis of the disc data on a personal computer enabled the form of the raw data to be examined. Thus simple data processing algorithms could be designed and tested off-line, refined, and then transferred to the AGV software.

It can be recalled that data processing routines are embedded in the main routine *datproc* which is entered on a timed low level interrupt every 100mS. Range measurements, however, are made on a high interrupt level at a rate of 100Hz. Measurements from each sensor are stored in an array of dimension 50 in a rotating fashion which means that any amount of data collected up to a time of 0.5 seconds before the current control interrupt is available for processing.

The processing applied to the ultrasonic measurements includes recursive digital filtering and the the elimination of spurious values, line regression, notch detection, compensation for systematic errors, and estimation of ultrasonic position measurement errors using a confidence interval basis. This chapter describes each stage of data processing in turn and, subsequently, results are presented which show that position corrections appear as step disturbances to the AGV's control system. Finally the results of a larger

scale test are presented in which the research vehicle is asked to follow a path, around benches in the laboratory, from its base position  $(X, Y, \theta = 0, 0, 0)$  to a defined goal position  $(X_1, Y_1, \theta_1)$ .

The performance of the system is then examined in more general terms. Inaccuracies in both odometric and ultrasonic position measurement, and the frequency at which ultrasonic measurement can correct odometric measurement, gives a picture of how positional errors build up with time. With constraints on the maximum error allowable, the manner in which errors accumulate dictates the maximum distance that the AGV can travel between correction boards.

## 8.2 Digital Filtering

The results from either the two left sensors or the two right sensors are initially passed through a simple recursive digital filter to reduce the degree of noise. The filter employed is first order and of the form

$$y_{k_i} = (1 - \alpha)y_{k_{i-1}} + \alpha\gamma_{k_i} \quad (8.1)$$

where  $\gamma_{k_i}$  is the current unfiltered range measurement in the set of observations provided by sensor set  $k$  over the last control interval, and  $\alpha$  is a constant between the values 0.0 and 1.0.

Using a stationary sensor pointing towards a correction board it has been verified that the standard deviation of random variations in distance measurement due to noise can be reduced by the factor dependent on  $\alpha$  such that

$$\sigma_f = \frac{\alpha}{(1 - \alpha)}\sigma \quad (8.2)$$

where  $\sigma_f$  is the standard deviation of ultrasonic measurement after filtering and  $\sigma$  is the standard deviation before filtering.

It is obvious, however, that any change between two consecutive measurements made by an ultrasonic sensor has two components: one due to the random error of measurement, which must be minimised; and one due to the motion of the AGV, which must be preserved. Unfortunately the filter does not discriminate between the two types of change and thus the value of  $\alpha$  was chosen to compromise between a reasonable degree of filtering (lower  $\alpha$ ) and quickness of response to changes that derive from vehicle movements (higher  $\alpha$ ).

The effect of the filter on changes in measurement due to the vehicle's motion can be assessed as follows. Consider the AGV to be travelling on a constant local heading,  $\theta'$ , relative to a correction board. If it is initially assumed that there is no random error component on either of two consecutive unfiltered measurements then

$$(\gamma_{k_i} - \gamma_{k_{i-1}}) = VT_s \sin \theta' \quad (8.3)$$

where  $T_s$  is the sampling period (0.01s) of ultrasonic observation and  $V$  is the vehicle's base velocity. When the filter is in steady state with respect to such a ramp input, the component of  $y_{k_i}$  generated by  $\gamma_{k_i}$  in equation 8.1 must be equal to that deducted from  $y_{k_{i-1}}$  plus the increase in  $\gamma_{k_i}$  over a sampling interval as described by equation 8.3. Thus

$$\alpha \gamma_{k_i} = \alpha y_{k_{i-1}} + VT_s \sin \theta' \quad (8.4)$$

or:

$$(\gamma_{k_i} - y_{k_{i-1}}) = \frac{VT_s \sin \theta'}{\alpha} \quad (8.5)$$

Additionally, for this filter, steady state implies that the rate of both input and output must be equal. This means that for  $i \rightarrow \infty$ :

$$\gamma_{k_i} - \gamma_{k_{i-1}} = y_{k_i} - y_{k_{i-1}} \quad (8.6)$$

Rearranging gives

$$\gamma_{k_i} - y_{k_i} = \gamma_{k_{i-1}} - y_{k_{i-1}} \quad (8.7)$$

and expanding gives:

$$\gamma_{k_i} - y_{k_i} = (\gamma_{k_i} - y_{k_{i-1}}) - (\gamma_{k_i} - \gamma_{k_{i-1}}) \quad (8.8)$$

Substituting equations 8.3 and 8.5 into equation 8.8 and denoting the steady state error of the filter as

$$y_{f_{ss}} = \lim_{i \rightarrow \infty} \gamma_{k_i} - y_{k_i} \quad (8.9)$$

gives:

$$y_{f_{ss}} = \frac{(1 - \alpha)VT_s \sin \theta'}{\alpha} \quad (8.10)$$

If the research vehicle were travelling at 0.2m/s on its maximum heading of 20 degrees relative to a correction board, a value of 0.2 for  $\alpha$  gives a steady state following error of 2.7mm. This value assumes that the vehicle maintains its maximum heading for some time; in practice, the AGV always tries to follow a path parallel to a correction board which means such heading errors only exist for a short time immediately after position corrections. In addition, as the vehicle gets closer to its demand path, both  $\theta'$  and the steady state error of the filter tend to zero. Empirically it was found that a suitable value for  $\alpha$  is 0.2. Such a low value can be employed because of the high rate of ultrasonic observation (small  $T_s$ ) and the small

magnitude of steady state local angle error when frequent position corrections can be made.

In figure 8.1, raw distance measurements and the same data digitally filtered with equation 8.1 are shown as the AGV moves towards a correction board and passes a notch.

### 8.3 The Elimination of Spurious Ultrasonic Range Measurements

In the initial testing of the AGV, it was found that the sensors at the rear of the vehicle and remote from the on-board computer were prone to occasional and singular spikes. This may have been due to their long connecting leads, although shielded, being more prone to noise. Difficulties were experienced due to the transmit signal coupling onto the receive signal, which could sometimes generate a low erroneous reading. Figure 8.2 shows results collected from the rear left sensor in which four spurious low readings occur. The problem with employing a recursive filter is that, since its output is dependent on previous outputs, the effect of such spurious spikes remains present long after the spike occurs. In order to filter single erroneous range measurements, and thus eliminate their sustained effect, the digital filter adjusts its constant,  $\alpha$ , for each individual ultrasonic observation by examining a successive sequence of three results in the batch of results collected over the last control interval.

Figures 8.3a and 8.3b show the basis of this filter constant adjustment for a high suspect reading and a low suspect reading respectively. If, in the digital filtering process, the modulus of the difference between the current unfiltered ultrasonic measurement,  $\gamma_i$ , and the previous filtered measurement,  $y_{i-1}$ , is greater than some threshold (set at 3cm), then the current measurement is deemed to be suspect and the next unfiltered measurement,  $\gamma_{i+1}$ , in the sequence of ultrasonic observations collected over the last control interval, is

used to determine whether the current measurement in the filtering process is a spurious singular point or the first reading of a new established level. (Which, for example, may be the first reading reflected off the face of a notch.) If  $\gamma_{i+1}$  is close to the current suspect value,  $\gamma_i$ , then that suspect value is deemed to be non-spurious and the normal value of  $\alpha$  (0.2) is used in the filtering process. However, if  $\gamma_{i+1}$  is closer to the previously filtered value  $y_{i-1}$ , then the current value is considered to be spurious,  $\alpha$  is set to 0.0, and thus, by equation 8.1, the suspect value is filtered. If  $\gamma_{i+1}$  lies in a range  $d/3$  to  $2d/3$  (see figure 8.3) between the previous filtered value and current suspect value, caution is adopted by setting the filter constant between 0.0 and the value chosen for normal filtering.

The threshold, 3cm, above which a reading is considered to be suspect, was chosen to be just above typical changes in consecutive readings as a sensor passes a notch. This prevents the filter clipping the spikes observed at the front and rear of a notch.

The method of filtering described in this section has proved successful in that it filters out spurious singular points but does not affect the changes in level generated when a sensor passes a notch.

#### 8.4 The Notch as Seen by an Ultrasonic Sensor

The dotted line shown in figure 8.4a represents the true shape of the notch. This has length 10cm and depth 4cm. The form of the distance measurements as the sensors move past the notch shows the depth and length of the notch quite accurately but its shape is rounded and has spikes at each end. These deviations from the true shape of the notch can be explained by considering figure 8.4b. As the transducer pair approaches the edge of the notch, the area on the correction board from which the ultrasonic signal is reflected contacts the notch (1). When the AGV moves on from this

point, less of the signal is reflected off the board and more off the notch (2). This causes the amplitude of the signal reflected off the board to fall rapidly giving a corresponding increase in measured distance. Very shortly after the original signal has started to fall, the portion of reflected signal from the notch reaches detection threshold and the signal portion from the back of the board is ignored because it is further away (3). Since this signal is initially very small, the first distance measurements of the notch have offset errors. These are quickly reduced as the AGV moves further forward (4) and eventually the whole of the ultrasonic beam area can reflect off the surface of the notch (5).

When the sensors move past the rear edge of the notch, the reflected signal is again split causing a reduction in the notch signal amplitude and an increase in the signal returned from the correction board. This produces another spike on the back edge of the notch (6-10).

As one might expect, the shape of a notch becomes less distinct the further a sensor is away from the correction board. Figure 8.5 shows the degradation of notch shape with range. This occurs because the detection threshold is much closer to the peak of the echo envelope and so any fluctuations in the amplitude of that echo generate larger fluctuations in measured range. This situation is exacerbated when the reflected beam is split which generates large fluctuations before and after the notch itself.

Section 8.6 develops a method of detecting notches which employs the gradient changes between consecutive regression lines and is robust to the degradation of the notch shape with range. The use of regression in the research vehicle's data processing software is described in the following section.

## 8.5 The Use of Regression in Data Processing

In order to reduce further the error in measured perpendicular distance and local orientation ( $y, \theta'$ ) relative to a correction board, and to facilitate a robust method of notch detection and thus local  $x$  coordinate measurement, the filtered ultrasonic range measurements undergo further processing. At each low level control interrupt, least squares regression lines are fitted to the two new sets of filtered ultrasonic observations which have been provided by the front pair and rear pair of sensors.

Again it should be noted that changes in consecutive measurements made by an ultrasonic sensor have two components: one due to the random error of measurement, which must be minimised; and one due to the motion of the AGV, which must be preserved. In deciding what form of data processing to employ, the aim is to utilise *all* the data collected between the  $i - 1$  and  $i$ th control interval in minimising the position measurement error at the  $i$ th control interval. Obviously a straightforward averaging process is inappropriate since each ultrasonic distance measurement is made at a slightly different position. The use of a regression curve \* can be thought of as a method of discriminating between the component of ultrasonic measurement variation due to the motion of the vehicle from that component due to random error in measurement. If the form of the regression curve fit accurately models the form of the vehicle motion, then, as the number of points in the fitting process increases, the effect of random errors on the parameters of the curve is reduced and thus the fitted curve becomes a closer approximation to the vehicle's movements over the last control interval. This implies that the last position,  $y_{k_n}$ , on the regression curve is a more accurate measurement of sensor  $k$ 's true position than the final range measurement used to form that curve.

---

\* Here 'curve' is used in the general sense.

The regression used in the research AGV's software requires a number of initial assumptions to be made:

1. There is only a small amount of random error in the measured positions  $x_i$  at which the ultrasonic observations are made.
2. The vehicle moves on a constant heading over a control period.
3. The random error in ultrasonic measurement comes from a normal distribution of constant variance.

The first assumption is required to employ a standard least squares curve fitting procedure rather than employing the more complicated procedures used for fitting curves to a pair of random variables. In the case of the research AGV, the change in local  $x$  coordinate ( $x_i - x_{i-1}$ ) derived from odometry over a control interval is divided by the number of ultrasonic observations collected over that same interval in order to associate a local  $x$  position with each individual observation. Although variations do appear in the predictor variable ( $x$ ) due to bumps in the floor and quantisation effects, they are small compared to the random variations in ultrasonic observations. The first assumption is, therefore, valid.

The second assumption is required in order to employ a first order regression or, in other words, to fit straight rather than curved lines to the collected data. In reality, the vehicle executes some constant curvature, as derived by the controller, over a control interval. For small headings it has been noted that curvature can be approximated by

$$\kappa = \frac{d^2y}{dx^2} \quad (8.11)$$

(see appendix 4). This means that a second order model of the form

$$\hat{y} = b_0 + b_1x + b_2x^2 \quad (8.12)$$

may be appropriate to implement the regression. However, since the control interval is short relative to the vehicle's speed, the angle subtended by the segment of constant curvature (executed by the vehicle over the control interval) is small, and thus the segment can be approximated by a straight line. This approximation permits the simpler model

$$\hat{y} = b_0 + b_1x \quad (8.13)$$

to be employed in reducing the effect of random error on the vehicle's measurement of local position. The error incurred by the above approximation is the smallest when the vehicle is close to its designed course and curvature is very low. The largest errors occur shortly after a position correction when curvatures can reach  $1m^{-1}$  (see figure 7.16b). If the vehicle is travelling at 0.2m/s, this generates a change in heading of 1.146 degrees over a control interval.

The final assumption is required in order to employ equations which determine the accuracy of a specific value of  $\hat{y}$  predicted by the regression equation. In chapter 4 it was noted that errors in ultrasonic measurement were liable to be generated from a variety of sources and so tended to be normally distributed. The way in which error bands for the vehicle's position are derived using confidence intervals and the measured standard deviation of random error is described in section 8.8.

It can easily be shown that the parameters of a first order least squares regression can be derived as

$$b_1 = \frac{\sum_{i=1}^{i=n} x_i y_i - n \bar{x} \bar{y}}{\sum x_i^2 - n \bar{x}^2} \quad (8.14)$$

$$b_0 = \bar{y} - b_1 \bar{x} \quad (8.15)$$

[DRAPER and SMITH, 1981]. If the variance of the residuals around the regression line is found to be approximately equal to the variance of range measurements obtained in static tests, then the regression line is a good fit and equation 8.13 is a good model of the vehicle's motion over the last control interval. Other than when a particular sensor moved past the edge of a notch, this was found to be the case.

## 8.6 Notch Detection

The function of the notch detection routine (*notch*) is to determine whether a particular sensor is opposite a notch. If a notch is successfully detected, the routine calculates the position of the centre of the notch according to the AGV's current odometric estimate of position in the local  $x$  direction. As well as being required for position correction in a positive sense (i.e. to measure local  $x$  position and, therefore, correct global position), the AGV needs to know when either its front or rear sensor pair is opposite a notch in order to prevent a false angle correction from being made.

In order to implement a robust method of detecting notches, the chosen algorithm should attempt to extract as much information as is available in the definition of a notch, when such a notch is observed gradually by an ultrasonic sensor from left to right. Obviously, monitoring sudden changes in depth is insufficient in itself since such instances could easily arise from spurious results, or if a person were to walk between the correction board and the AGV. As a further illustration of this point, consider an algorithm

which only uses the criterion of large changes in regression line gradient to detect a notch. Such an algorithm was tested and it was found that several (as opposed to one) notches were detected as a sensor passed a single notch. Figure 8.5b illustrates that oscillations prior to a notch cause several basic notch shapes to be connected in series as a notch is passed.

A flowchart of the notch detection algorithm (*notch*) is shown in figure 8.6. Firstly a threshold value is set for changes in gradient between consecutive line segments, generated in the current (i) and previous (i-1) control interrupts, by using the measured perpendicular distance from a correction board in an empirically defined table. (It was found that larger changes of gradient appeared at the edges of a notch at greater distances from the correction board.) The routine then determines whether a particular sensor is currently deemed to be opposite a notch by examining that sensor's status flag. This can be either *INNOTCH*, representing the fact that a particular sensor is currently opposite a notch, or *OUTNOTCH*, representing the fact that a sensor is not currently opposite a notch. This status flag would have been set by the previous entry into *notch*.

If the sensor is not opposite a notch, the gradient of the current line segment is subtracted from that of the previous line and tested against the current threshold. Thus, if the change is sufficiently large, the front edge of a notch has been detected and the status flag is set to indicate that this particular sensor is opposite a notch. In addition, the local *x* position at which the gradient change takes place is noted as the position of the front edge of a notch.

If a sensor is already deemed to be opposite a notch (*status = INNOTCH*), a check is initially made to ensure that the distance travelled from the recorded position of the front edge is not more than 12cm. Thus, allowing for inaccuracies and the widening effect inherent in the measurement,

spurious results may be eliminated using the known length of the notch. If such a condition is not detected, the change in gradient between current and previous line segments is thresholded. If a sufficiently large change in gradient is found, then the rear end of the notch is deemed to have been detected. However, before the algorithm accepts that a notch has been detected, it checks that the distance between the front and back edge is within reasonable bounds of the true length of the notch (8-12cm). If, however, the change in gradient is less than the current threshold, the routine checks to see if the gradient changes from being negative to positive. Empirically it has been found that this always indicates that the sensor is opposite the surface of the notch (see figure 8.8b) and thus the current level of ultrasonic reading can be compared with that recorded outside the notch. If the result is not within reasonable bounds of the depth of the notch (3-5cm) then the sensor status is set to *OUTNOTCH* and the routine exits thus rejecting the initial front edge recording as being spurious. This process solves the problem of multiple recordings of a notch when a sensor passes a single notch, as was described earlier and as might have been obtained from figure 8.5b.

A major problem with the method described above was that sometimes the results would fall so that a line was fitted directly through the edge of a notch. In the initial development of the notch detection algorithm, regression lines were fitted to collected data off-line and the positions of the maximum residuals were plotted. In figure 8.7a, consecutive sets of 30 data points were employed. Here it can be seen that the degree of poor fit is large and the maximum residuals occur at the edges of the notch. Figure 8.7b shows an improvement of fit when fewer data points (20) are employed in the regression.

An instance of poor fit has two detrimental effects. Firstly, it reduces the reliability of edge detection since the measured change in gradient between two consecutive line segments is reduced. Secondly, if the edge is detected,

an error is introduced in its measured position since the edge is not actually between the two lines at which the large gradient change occurred.

In order to detect instances where the line fit is particularly bad, it is possible to examine the variance of the residuals around the regression line. If this is substantially greater than the variance of measurements in static tests, the line is deemed to be a 'bad fit'. Alternatively, the maximum residual of each line fit can be compared against some threshold derived in relation to the variance of random errors. In the case of the research AGV, if the modulus of the maximum residual is greater than 1cm, the line fit is deemed to be bad, since that result falls outside three standard deviations of the expected value.

It should be noted, however, that although the maximum residual often falls at the peak of the spike generated by the edge of a notch, this is not always the case. Since the aim is to detect the position of the notch edges and these positions have local maxima, the maximum of the data set from sensor  $k$ ,  $y_{k_{max}}$ , is used as the point at which to split the data. Subsequently the current regression line is fitted to the ten values previous point at which the data is split. Values beyond this point which are initially ignored, are included in the line fitting procedure at the following control interrupt. Figure 8.8a illustrates the results of an off-line implementation of the refitting process whilst figure 8.8b shows the fitted lines calculated on-line in a test run. In this diagram, some of the regression lines appear to be slightly displaced from the ultrasonic data. This is due to the limited resolution of the integer arithmetic embedded in the routine *regression*.

Figures 8.9a and 8.9b show results from the line fitting process and the notch detection algorithm respectively. The first figure shows the gradients of fitted line segments as the AGV moves parallel to a correction board, with its sensors displaced 50cm from that board. Notches on the correction board

are located at 0.6m, 1.6m, and 2.6m in the  $x$  direction. Figure 8.9b shows the value of a symbolic constant which indicates the point at which the front edge of a notch is detected (1), the point at which its depth is accepted as being valid (2), and the point at which the rear edge is detected and the notch is accepted (3). Note that in the first and third notches the initial front edge is rejected, since the changes in gradient were caused by oscillations prior to the notch itself.

Although the notch detection algorithm described here is somewhat *ad hoc* (since it is used to detect a specific feature!), it has proved to be successful. This is largely because it employs all of the *a priori* information about the notch (width, depth, gradient changes) in order to discriminate such features from noise and other disturbances in the string of ultrasonic observations. Tolerances in the algorithm for notch dimensions and gradient changes were obtained by examining collected results after the vehicle had moved at a variety of distances relative to a correction board.

## 8.7 Measurement of the Local $x$ Coordinate

In order to measure and thus correct position in the local  $x$  coordinate, the  $x$  coordinate of the AGV at the instant of a particular sensor passing the midpoint of a particular notch needs to be calculated. This calculation employs the measured  $x$  positions of the front and rear edges of the notch, the position of the sensor that made the measurement relative to the AGV's reference point, and the estimated angle of the AGV as the sensor entered the state *INNOTCH*. Bounds on the estimated local  $x$  coordinate generated by the odometry routine (see chapter 4) permit the true local  $x$  coordinate of the notch to be determined from the array "notch" (see chapter 6). A correction in the local  $x$  coordinate can then be made by adding to it the difference between the true  $x$  coordinate of the notch and the measured  $x$

coordinate of the notch. Having corrected local  $x$  position, a global position correction can be made since  $y$  and  $\theta'$  would have been corrected immediately before the notch.

Unlike errors in  $y$  and  $\theta'$ , errors in  $x$  can not be quantified in terms of the measured variance of the random errors in ultrasonic measurement. This is because the position measurement for which the sensor is used is perpendicular to the direction of the incident beam; the odometric position at which range variations are obtained are used as the measurement, whilst the changes in ultrasonic range measurement are simply used to identify the shape of a notch. The accuracy of  $x$  coordinate measurement is dependent on a number of factors which are more difficult to quantify, namely:

1. The symmetry of the detected shape of the notch.
2. The accuracy of  $x$  measurement over the length of the notch.
3. The measurement and constancy of  $\theta'$  over the length of the notch

Both edges were employed in the measurement of notch position since the length of the notch was observed to vary over the range of the sensor (at ranges below 50cm notch length appeared to be expanded). In order for the apparent shifts of the two edges from the notch centre to cancel one another at smaller ranges, the observed shape of the notch should be symmetrical around the true centre. \*

Since the length of the notch is small (10cm), the accumulation of odometric errors over that distance is bound to be small and have little

---

\* Initially this required that the transmitter and receiver be mounted on top of one another rather than side by side.

effect on the accuracy of notch position measurement. Similarly, because of its small length the vehicle's heading is assumed to be constant as a sensor passes a notch.

Although the error in local  $x$  coordinate measurement is difficult to assess analytically, repeated tests have shown that the local  $x$  coordinate can be measured to about  $\pm 1\text{cm}$ .

## 8.8 Compensation for Offset Errors

In chapter 4, it was demonstrated that ultrasonic measurements are subject to offset errors which are dependent on the perpendicular distance and local orientation of a sensor relative to a correction board. Strictly, such errors should be eliminated for each pair of ultrasonic measurements from the front and rear sensors before any further processing takes place. Since this represents a substantial cost in processing time, it is assumed that the measurements collected over a control period are made over such a short time interval that offset errors are equal for each member of the set. This assumption allows systematic error compensation to take place when all other data processing has finished.

Systematic error compensation requires iteratively operating on a two dimensional look up table which represents the sensor characterisation described in section 4.6.3 and contains offset errors for various combinations of perpendicular distance and orientation relative to a correction board. The values  $y_1$  and  $y_2$  representing the perpendicular distances of the front and rear sensors from the correction board are taken as initial estimates for the true values of  $y_1$  and  $y_2$ . These initial estimates are obtained from the line fitting procedure and are the *final* points ( $\hat{y}_{k_n}$ ) on the regression line itself. Estimated orientation can be calculated from these distance estimates as

$$\theta'_i = \sin^{-1} \left[ \frac{y_{2i} - y_{1i}}{s_r + s_f} \right] \quad (8.16)$$

where, in this initial case,  $i = 0$ . Subsequently offset errors are generated for each sensor using current estimates of perpendicular distance and orientation in a two dimensional interpolation in the characterisation table. If this interpolation process is denoted by  $f_n$  then:

$$y_{1_{off}} = f_n(y_{1i}, \theta'_i) \quad (8.17)$$

$$y_{2_{off}} = f_n(y_{2i}, \theta'_i) \quad (8.18)$$

These offsets are used to form revised estimates of  $y_1$  and  $y_2$  such that

$$y_{1_{i+1}} = y_{1i} - y_{1_{off}} \quad (8.19)$$

$$y_{2_{i+1}} = y_{2i} - y_{2_{off}} \quad (8.20)$$

which subsequently form a revised estimate of orientation in equation 8.16. The process is repeated and the change in calculated orientation over an iteration is used to detect convergence. Typically it takes less than five iterations to converge within 0.1 degrees. If convergence has not been reached within five iterations, the compensation routine exits as a fail-safe and in order to prevent time being wasted in the control interrupt.

It should be noted here that the inverse sine function in equation 8.2 is not performed in the AGV software. The approximation of  $\sin\theta'$  to  $\theta'$  incurs a tolerable error of 0.4 degrees (2%) when the AGV is orientated at its maximum angle (20 degrees) relative to the correction board.

## 8.9 Estimation of the Error in Ultrasonic Position Measurement

It can be recalled from section 4.4.2 that no statement can be made about the probability density within the vehicle's current three dimensional estimate of odometric error band. This means that there is no possibility of integrating data provided by the odometry system with data from ultrasonic ranging. Instead, it is assumed that the accuracy of any position correction is better than the current estimate of odometric position \* and the estimated error in the local position measurement employed for position correction is used to reset the global odometric error bands.

The error in local position  $(y, \theta')$ , as measured by the ultrasonic sensors, derives from a combination of systematic and random errors in the initial raw range measurements. Ideally, one would like to account for systematic errors and subsequently assess the effect of random errors on the final accuracy of local position measurement. Although attempts are made to reduce systematic errors to a minimum by careful measurement of ambient temperature and the determination of the offset errors which derive from the variations in the amplitude of an echo pulse, some degree of systematic error is bound to remain. Having acknowledged this, one needs to determine how large this remnant systematic error is likely to be (for example, to what accuracy can temperature be measured?) and thereby determine whether or not this is significant in comparison to the effect of random errors.

At this point it may be argued that such lengths are unnecessary, since odometric error bands are imprecise and build up quickly thus rendering initial estimate of positional error insignificant. This is true enough when position corrections are made infrequently. However, when the vehicle moves close to a correction board for some time, corrections can be made frequently and

---

\* Except in instances where position corrections are made very frequently, this assumption will generally hold true.

thus ultrasonic position errors form a significant part of the error in estimated position. In addition to having a direct role in the guidance method, the estimation of the component of error in local position measurement due to random errors in ultrasonic ranging gives an indication of the effectiveness of the data processing algorithms.

However much data processing takes place on the raw ultrasonic range measurements, random errors in the raw range measurements will still have their effect on the accuracy of local position measurement. This section describes how the research vehicle employs some statistical confidence-interval theory to assess how large that effect is. An assessment of the error in local position measurement generated by random errors can be made by

1. Assuming that the random component of measurement error is Gaussian.
2. Employing the variance of the random errors in measurement to determine the bounds of a 97.5% confidence interval on the predicted measurement at the end of the newest fitted line ( $\hat{y}_{k_n}$ ).

A 97.5% confidence was somewhat arbitrarily chosen to determine error bands for the measurements  $\hat{y}_{k_n}$ . This means that, in the absence of systematic errors, the probability of the true distance of both sensors being within the calculated error bands is 0.95 (since the set of results provided by two different sensors are independent), thus error bands on local position are derived with 95% confidence. It can be shown that the standard deviation of  $\hat{y}_{k_i}$  at a particular value of  $x_i$  is given by

$$s.d.(\hat{y}_{k_i}) = \sigma \sqrt{\frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}} \quad (8.21)$$

[DRAPER and SMITH, 1981] and the corresponding confidence interval either side of the predicted value is

$$\delta\hat{y}_{k_i} = t(n - 2, 1 - \frac{\alpha}{2})s.d.(\hat{y}_{k_i}) \quad (8.22)$$

where  $n - 2$  is the number of degrees of freedom on which  $\sigma$  is based,  $\alpha$  is 0.025, and  $t$  is *Student's-t* distribution.

In calculating these bands it can be stated with a confidence of 97.5% that the true perpendicular displacement of a particular sensor from a correction board at a given control interrupt lies in the region

$$r\{\bar{y}_{k_n}\} = \hat{y}_{k_n} \pm \delta\hat{y}_{k_n} \quad (8.23)$$

In other words, if repeated measurements were made at position  $x_n$ , the probability that the average measurement,  $\bar{y}_{k_n}$ , lies in the region calculated by equation 8.22 is 0.975.

It has been noted that equation 8.21 implies that if more points ( $n$ ) are included in the regression and if the range of  $x_i$  is extended, then the variance of  $\hat{y}_{k_i}$  will be reduced thus contracting the confidence intervals and making the prediction of true distances more accurate. Section 6.2 explained that the range of ultrasonic measurements limits the rate at which observations can be made, thus preventing much improvement in accuracy by increasing  $n$ . It can be recalled that ultrasonic data up to 0.5s before the current control interrupt is retained which means that it is possible to extend the range over which a regression line is fitted. However, the calculation of confidence intervals only holds if the distribution of residuals around the regression line is normal and their variance is roughly equal to that of measurements

made by a static sensor (i.e. if the regression line is a good fit). If the range of regression is extended further back in time, effects of the vehicle's curvature and change in heading come into play, a first order regression is an increasingly poor approximation to the vehicles movements, and the variance of residuals around the regression line exceeds that due to random errors in measurement. Thus the accuracy of position measurements is limited not only by the intrinsic accuracy of the measurement system, but also by the rate at which measurements can be made, and the range over which the proposed model of the vehicle's movements is accurate.

The confidence intervals calculated from equation 8.22 can be used to form a 95% confidence interval for the local angle relative to a correction board so that, after a position correction, local angle is assumed to lie within the region

$$r\{\theta'\} = \theta' \pm \delta\theta' \quad (8.24)$$

where, for small angles

$$\delta\theta' = \frac{\delta\hat{y}_{1n} + \delta\hat{y}_{2n}}{s_r + s_f} \quad (8.25)$$

Using the error bands of both local angle and the position of the rear ultrasonic sensor, error bands based on a 95% confidence interval can be generated for the local  $y$  coordinate so that, after a position correction, this coordinate is assumed to lie within

$$r\{y\} = y \pm \delta y \quad (8.26)$$

The value,  $\delta y$ , can be calculated using minimum and maximum values of  $\hat{y}_{k_n}$  in equations 4.1 and 4.2. Note that, since  $s_r$  is much smaller than  $s_f$ , the

error in measurement from the rear sensor has the greatest effect on the error in perpendicular distance measurement.

This section concludes the description of the methods of data processing employed by the research vehicle. All that remains now is to examine the vehicle's performance when it is asked to move along a number of specified paths in the laboratory. This is the concern of the following section.

## 8.10 Performance of the Research Vehicle

### 8.10.1 A Comparison of Corrected and Uncorrected Tracking Performance

In order to compare uncorrected (i.e. purely odometric) performance with the vehicle's performance when ultrasonic position corrections can take place, the vehicle was asked to travel five metres along the global X axis. This required loading the simple command queue

LF,F,0.2,0.0,5.0,0.0

ST,0

and running two different types of test; one in which the 'board search' option was omitted before the run, and one in which that option was taken before the run.

In the first type of test, the AGV assumes it can not observe a correction board and so relies on uncorrected odometric information only. Table 8.1 presents the recorded deviations of the research vehicle's final position from its goal position when it was asked to travel in a straight line for five metres along four different paths in the laboratory. (The surface on which these tests were conducted is covered with carpet tiles of dimension 0.5m by 0.5m. In order to make a measurement of the vehicle's final true

position, the point at which the each of the drive wheels contacted the laboratory floor was measured relative to the nearest junction of tiles and converted to a global position.) The results of these tests indicate that, although the vehicle thinks that it is tracking its demand path within a few millimetres, its final measured position deviates from its goal position in all three coordinates by varying degrees. Table 8.2 presents the results of repeated runs on the original four paths in the laboratory. The signs of these results indicate that there is some degree of repeatability in the odometric error for a given path in the laboratory; thus irregularities in the floor surface must form a large proportion of the odometric error.

In the second type of test, the *bsearch* routine inserts additional information into the command queue array before that queue is executed. This information describes the position of the correction board in the local frame of the command. Thus if the 'board search' option is taken before a test run, corrections are repeatedly made to the vehicle's odometric estimate of position. When the AGV was asked to execute the above command queue with the appropriate correction information inserted, the vehicle's final position was found to be much closer to the goal position than in the previous tests. The results presented in table 8.3 indicate that, if position corrections can continually be made before the vehicle reaches its goal position, typical errors in goal position were found to be less than 1.5cm in the X and Y coordinates, and less than 1.5 degrees in the  $\theta$  coordinate.

### **8.10.2 Performance of the Vehicle with a Large Initial Misalignment**

In order to test if the vehicle could recover its demand path, when initially positioned with a large misalignment relative to that path, the simple queue described in the previous section was executed. In this test, the

results of which are in figures 8.10a and 8.10b, position was corrected using every twentieth set of valid position correction data. This allowed time for odometric errors to build up between position corrections, thus making the effect of position corrections more pronounced. Figure 8.10a indicates that the vehicle moved approximately 0.7m before a position correction of 22cm occurred. Subsequently, as it moved towards its true demand path, which is displaced from the correction board by 85cm, position corrections were consistently made in the same direction. This is not surprising since the path taken by the AGV was that of test 1 in tables 8.1 and 8.2 where the vehicle was seen to veer continually to the left of its demand path thus generating a large negative error in the final Y coordinate.

### 8.10.3 Vehicle Performance Over More Complex Paths

In order to determine how well the different aspects of data processing and control worked together, the AGV was asked to travel along more complex path than a simple straight line. In this test the AGV was placed in the vicinity of the demand path starting position,  $X, Y = 0, 0$ , but with a large initial misalignment. It was then asked to execute the command queue:

LF,F,0.2,0.0,2.0,0.0

CF,F,0.1,0.9,90.0,2.9,0.9

LF,F,0.2,90.0,2.9,2.9

ST,0

The results collected from this experiment are shown in figures 8.11a and 8.11b. Figure 8.11a shows the vehicle's current estimate of position (the full line) with respect to the demand path (the dashed line). Superimposed on this diagram is a dotted line which represents the probable path along

which the vehicle travelled whilst moving towards its goal position. The vehicle can be seen to correct its initial misalignment and subsequently move on to its true demand path whilst position corrections are made continually. When the vehicle enters the curve follow command, it is no longer possible to make position corrections, yet the control system gradually reduces the estimate of positional error which was substantial at the start of the curve command segment. Since position corrections can not be made in a curve segment, errors in the odometric estimate of position inevitably build up and, when the vehicle enters the final line segment in the queue, this error is observed, position is corrected, and the vehicle moves back towards its true demand path. The final error in goal position for this test was measured to be 1.4cm in  $Y$ , 1.2cm in  $X$ , and 0.7 degrees in  $\theta$ .

Figure 8.11b shows the estimated error of the vehicle with respect to its demand path against time. As well as showing the main position corrections clearly, it shows the effect of the step change in demand curvature at the start of the curve command segment.

### 8.11 A Discussion of Vehicle Performance

Experiments described in this and the previous chapter, which concern the performance of the AGV, have shown that both the data processing and control strategies that constitute the proposed low level guidance scheme have proved successful. If the research vehicle can observe a correction board, it can measure its local position effectively. In addition, the control system can track linear and curved command segments with minimum error, and can move the vehicle onto its demand path when it is initially misaligned, or after a position correction, in a controlled manner.

An assessment of the overall performance of the AGV in the previous section has shown that, on certain paths in the laboratory, the vehicle can

get within one or two centimetres of its goal position. However, when the vehicle was asked to perform longer and more complex paths, it would often deem itself to be lost and abort its run. The cause of this is attributed to a combination of deficiencies in the two sensor systems: the rapid build up of odometric errors, and the limited range of the ultrasonic sensors. For example, if the vehicle's new demand path is displaced 1.2m from a correction board and an odometric error of 30cm has built up, position corrections are sometimes not possible on account of the limited range of the ultrasonic sensors and the vehicle would subsequently give up. Even if position corrections can be made after a large odometric error has built up, the length remaining in the command segment leading up to the goal position must be large enough to allow the control system to reduce the measured error in position. (This is a relatively minor problem because strategies for coping with these cases could easily be developed. For example, a reduction of demand velocity would increase the control constants thus allowing the vehicle to turn more steeply towards its demand path.)

Part of the reason for the poor odometric performance was that, during the course of experimentation, the research AGV had to be moved from a hard tiled floor to a carpeted floor. On a tiled floor, only a small central part of the drive wheels contacted the floor surface and so the effective wheelbase could be measured accurately. Since the drive wheels are appreciably wide (5cm), and a carpeted floor has some give in it, the whole width of the wheel surface contacted the floor making it impossible to determine exactly what the wheelbase was. (When turning, the wheels must shear against the carpet surface.)

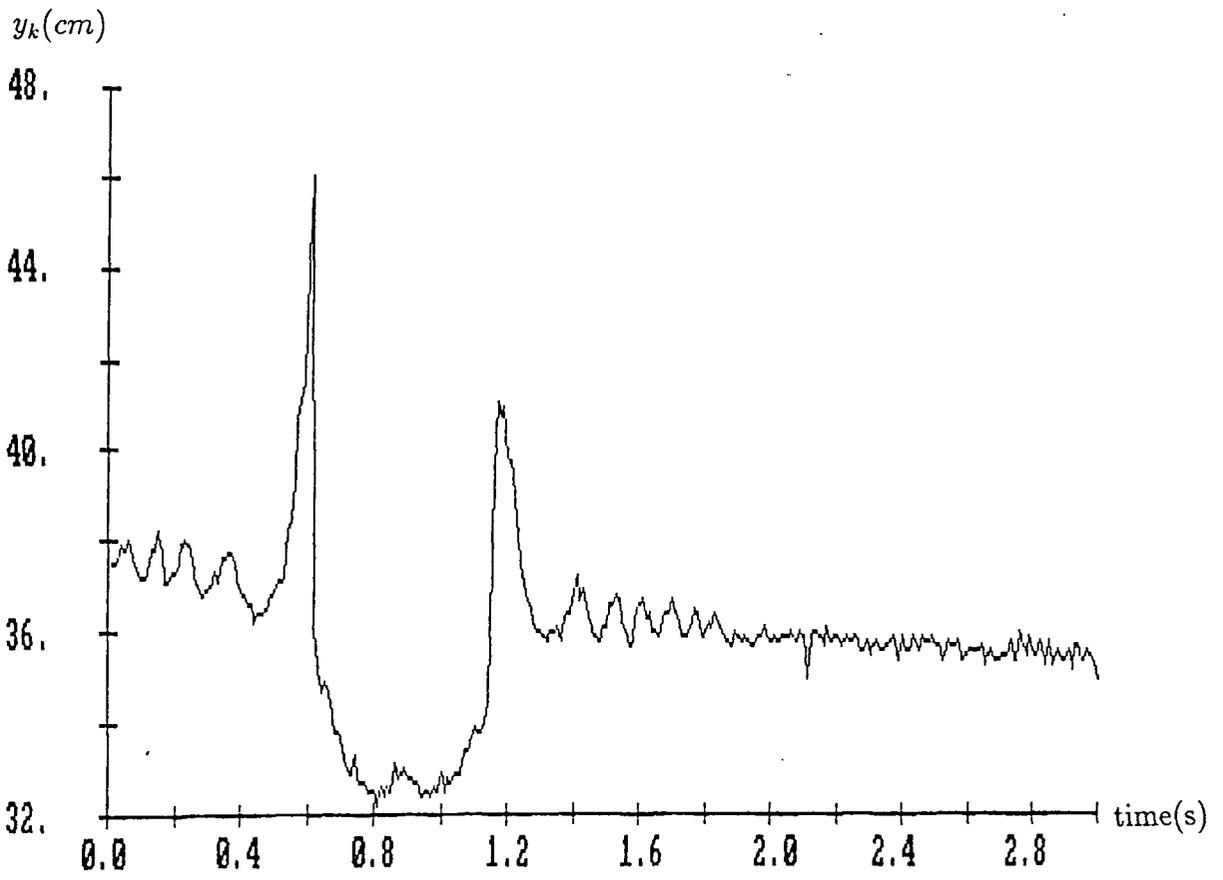
Although problems with the effectiveness of the two low level sensor systems have been experienced, such problems are merely inherent deficiencies of the equipment used on the AGV rig and, therefore, do not invalidate any of

the data processing and control methods that constitute the proposed guidance method. Proposals for improving the two sensor systems are included in the suggestions for further work (chapter 9).

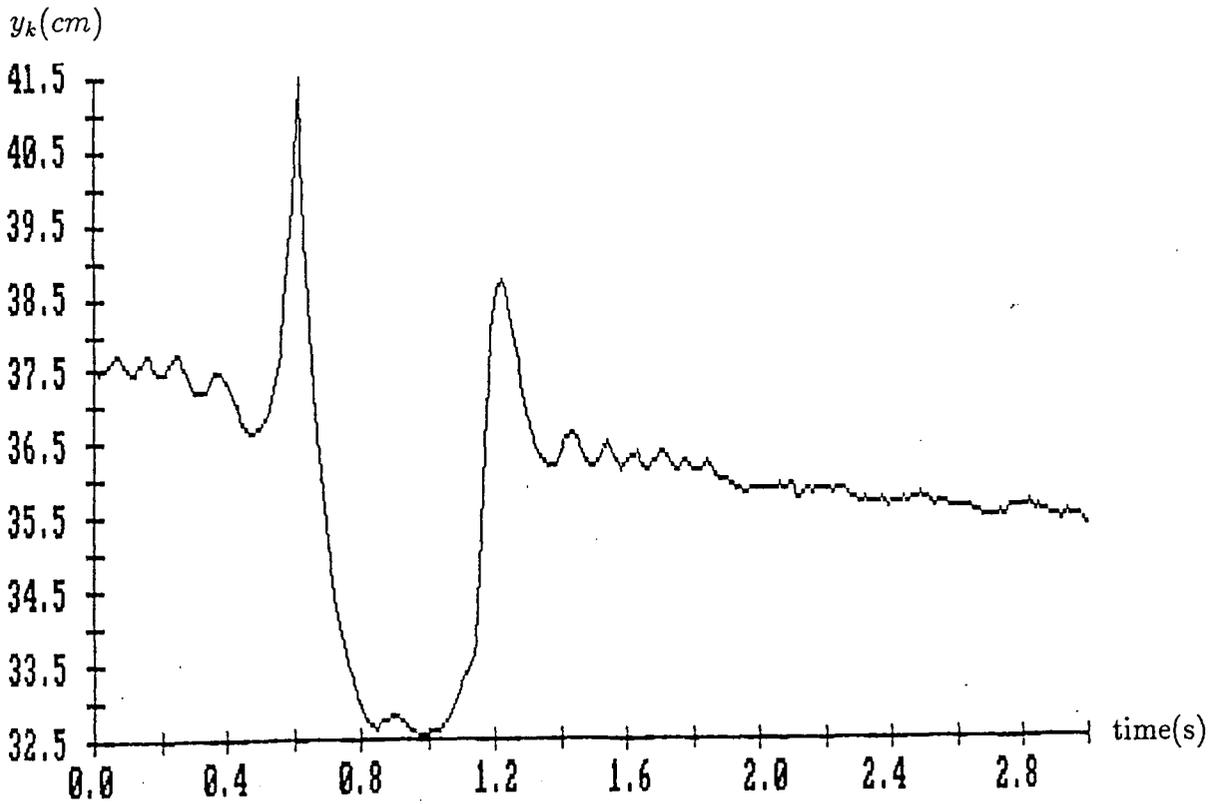
## 8.12 Summary and Conclusion

This chapter initially described a recursive digital filter which can adjust its filtering constant in order to reject spurious measurement singularities. An analysis of the operation of this filter has shown that its steady state errors are small due to the high rate of ultrasonic ranging and the fact that, in general, vehicle heading relative to a correction board is small. It was then shown how line regression can be used to implement a notch detection algorithm and reduce the effect of random errors on local position measurement. The chapter then deals with errors: in particular, a method of reducing systematic offset errors has been described along with a method of estimating the effect of random errors on the accuracy of local position measurement.

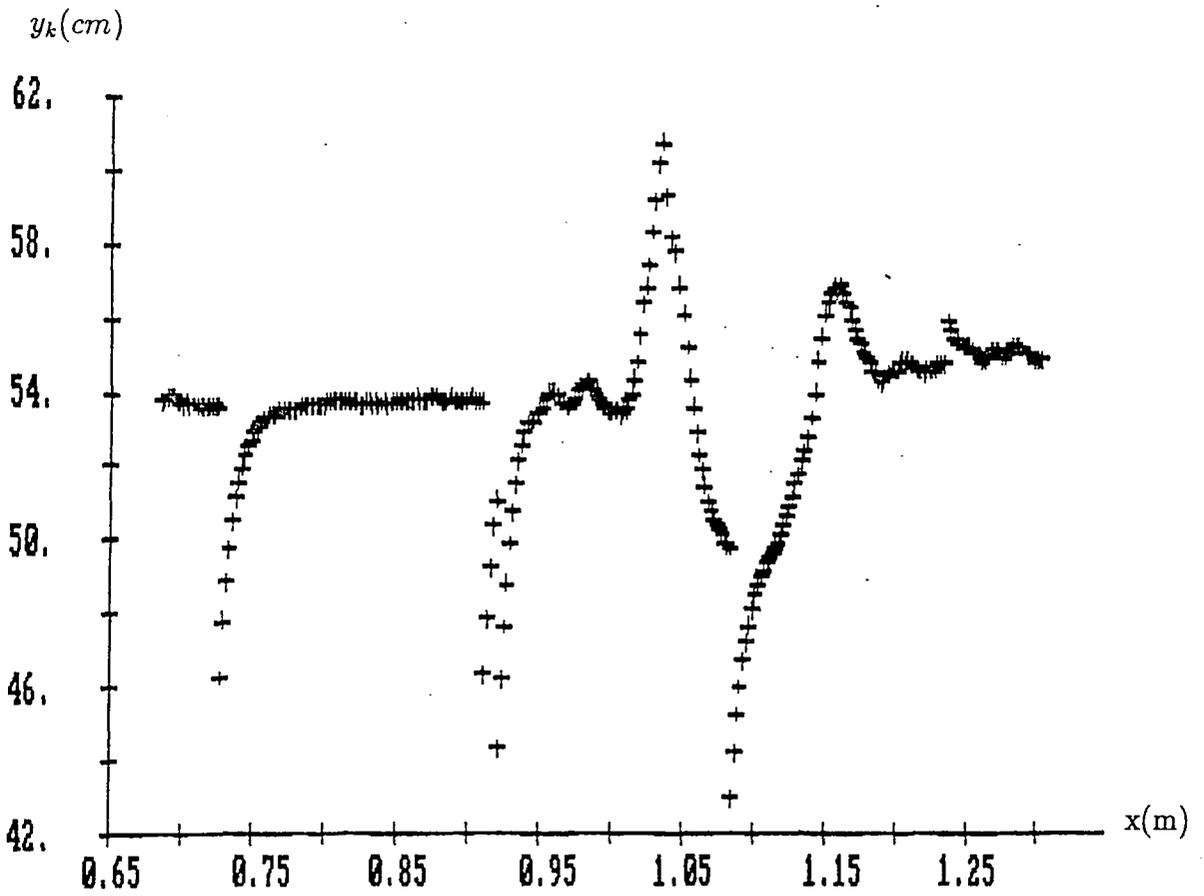
Experiments have indicated that the low level sensing systems employed can operate in a complementary fashion and, given that the research vehicle can observe a correction board on its final command segment, it has been shown that it can reach its goal position within one or two centimetres. However, the poor range of the ultrasonic sensors and the rapid build up of odometric errors can sometimes cause the vehicle to 'lose sight' of a correction board. To overcome this problem, improvements need to be made to both of the low level sensing systems.



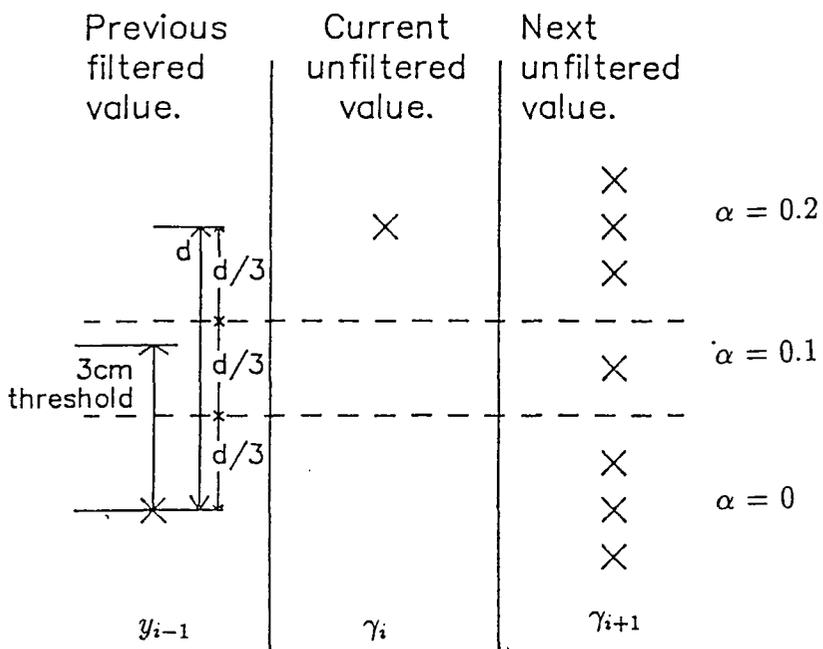
8.1a Unfiltered ultrasonic data



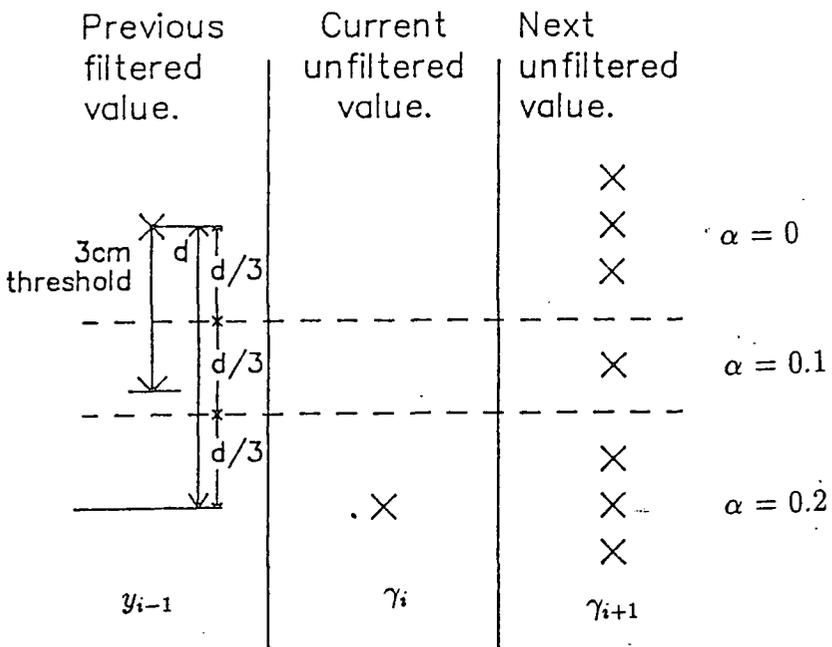
8.1b Filtered ultrasonic data



8.2 Ultrasonic data with four spurious measurements



8.3a High suspect measurement



8.3b Low suspect measurement

8.3 Filter constant adjustment

Figure 8.4a Range data as sensor  $k$  moves past a notch

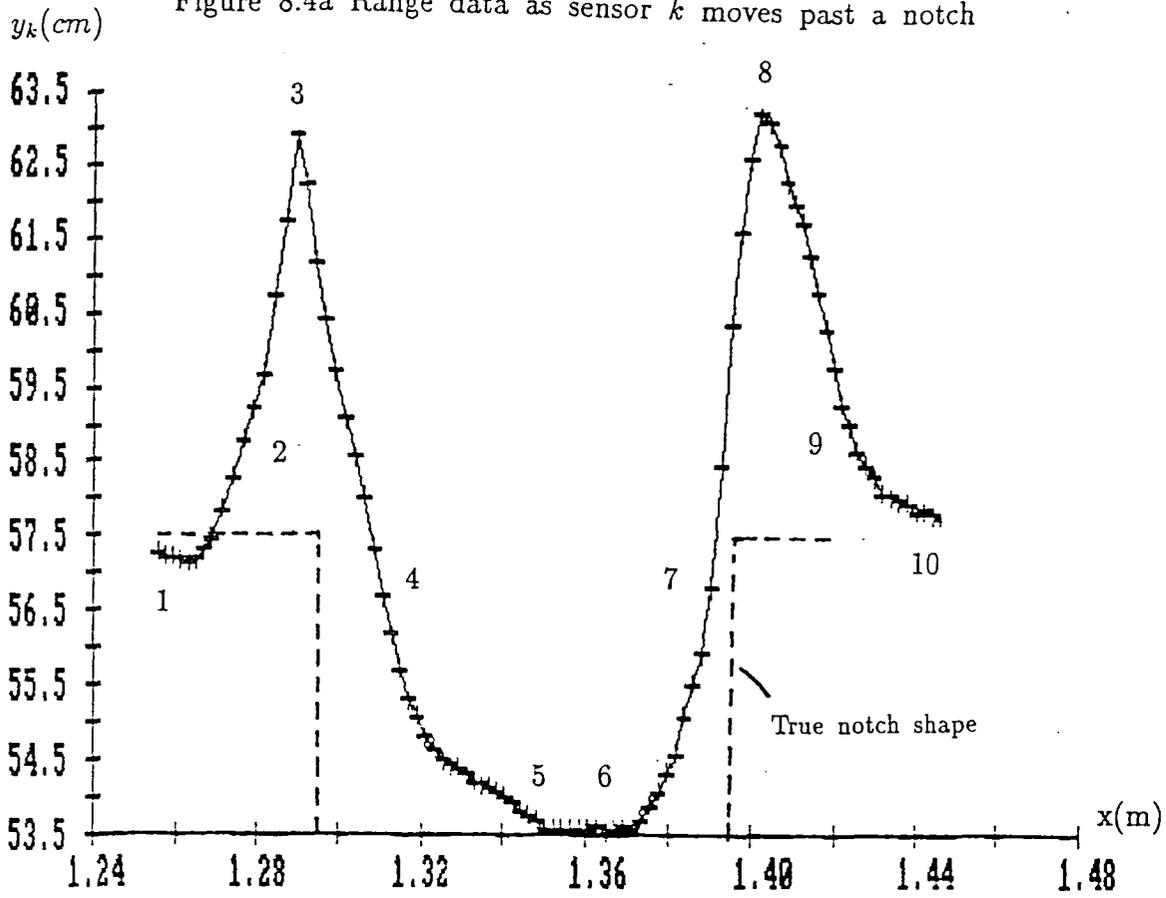


Figure 8.4b The splitting of an ultrasonic echo pulse

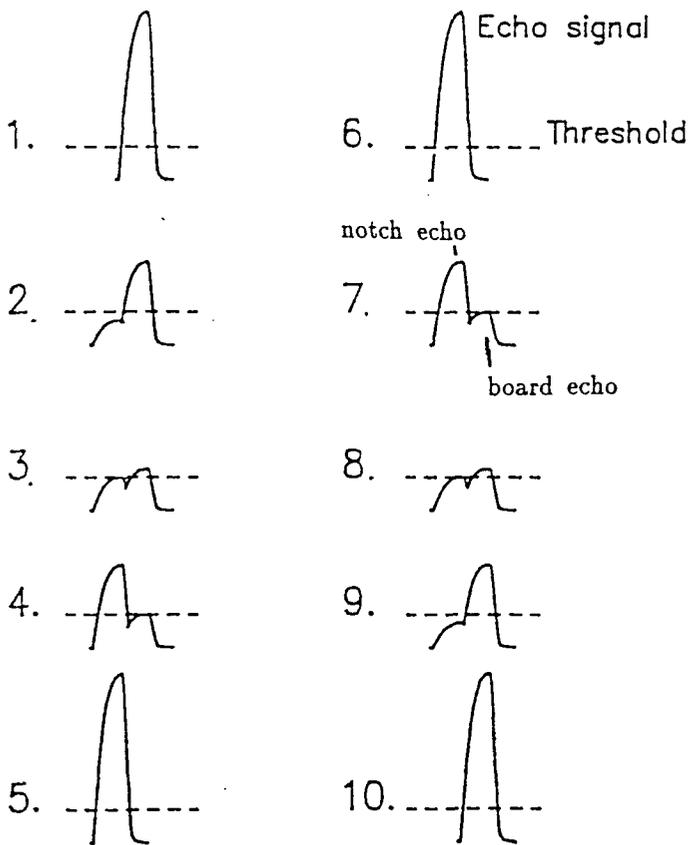


Figure 8.5a Close range observation of a notch

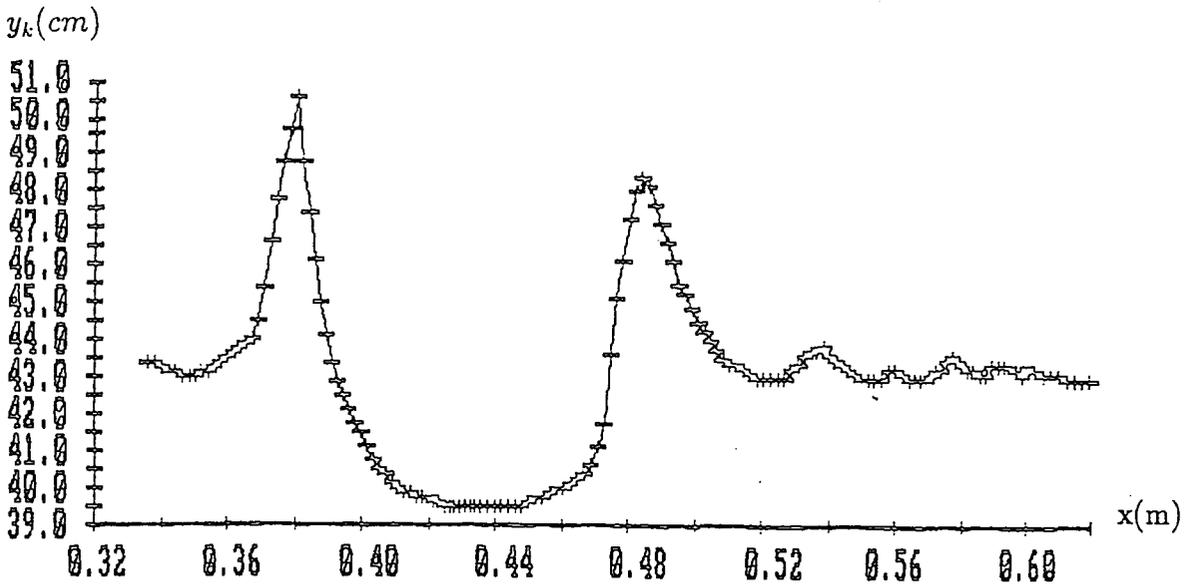
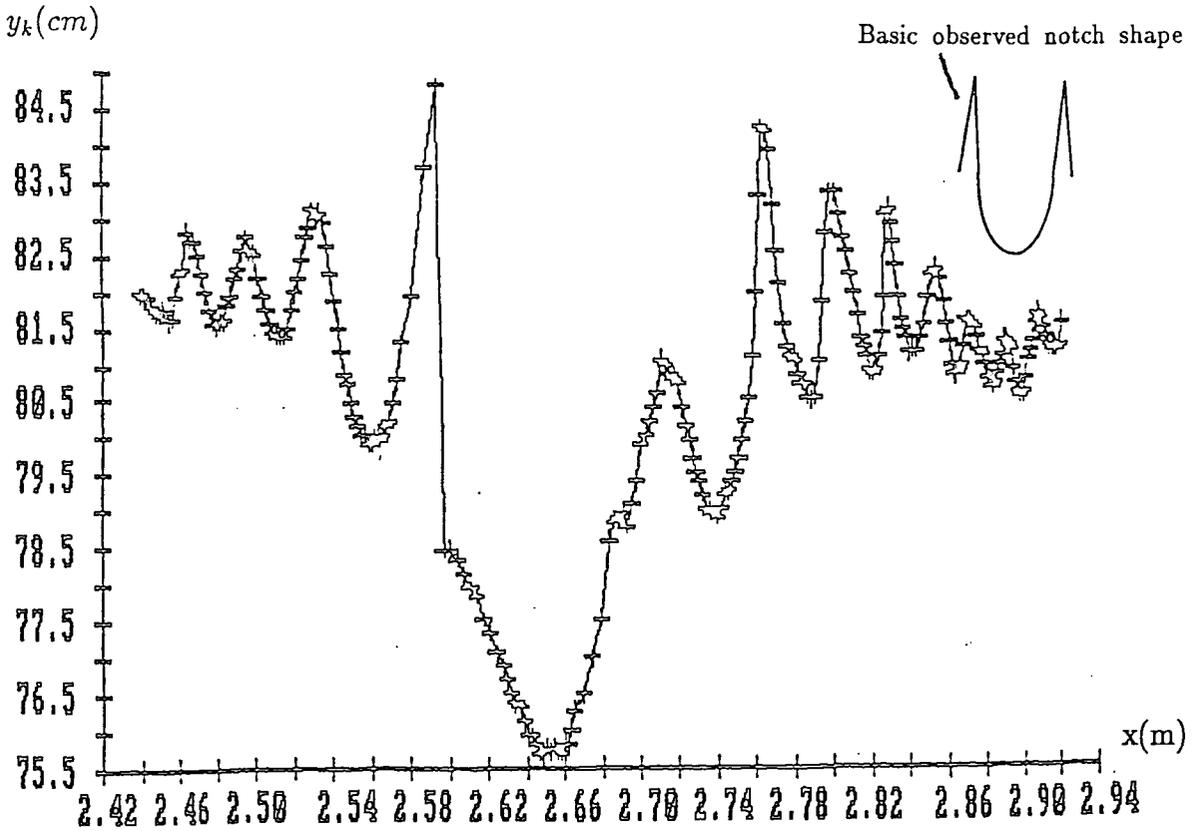


Figure 8.5b Far range observation of a notch



8.5 The degradation of notch shape with distance

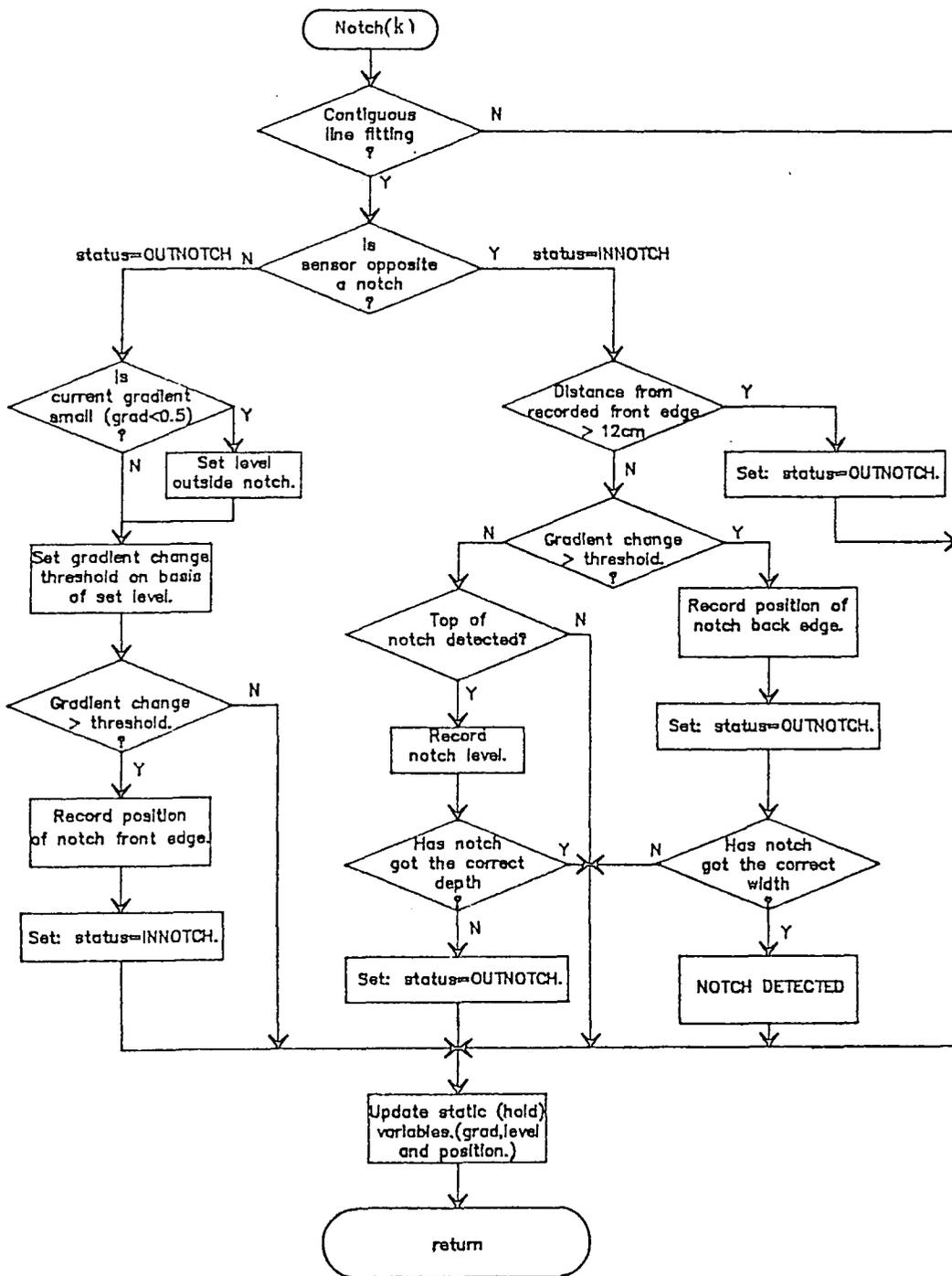
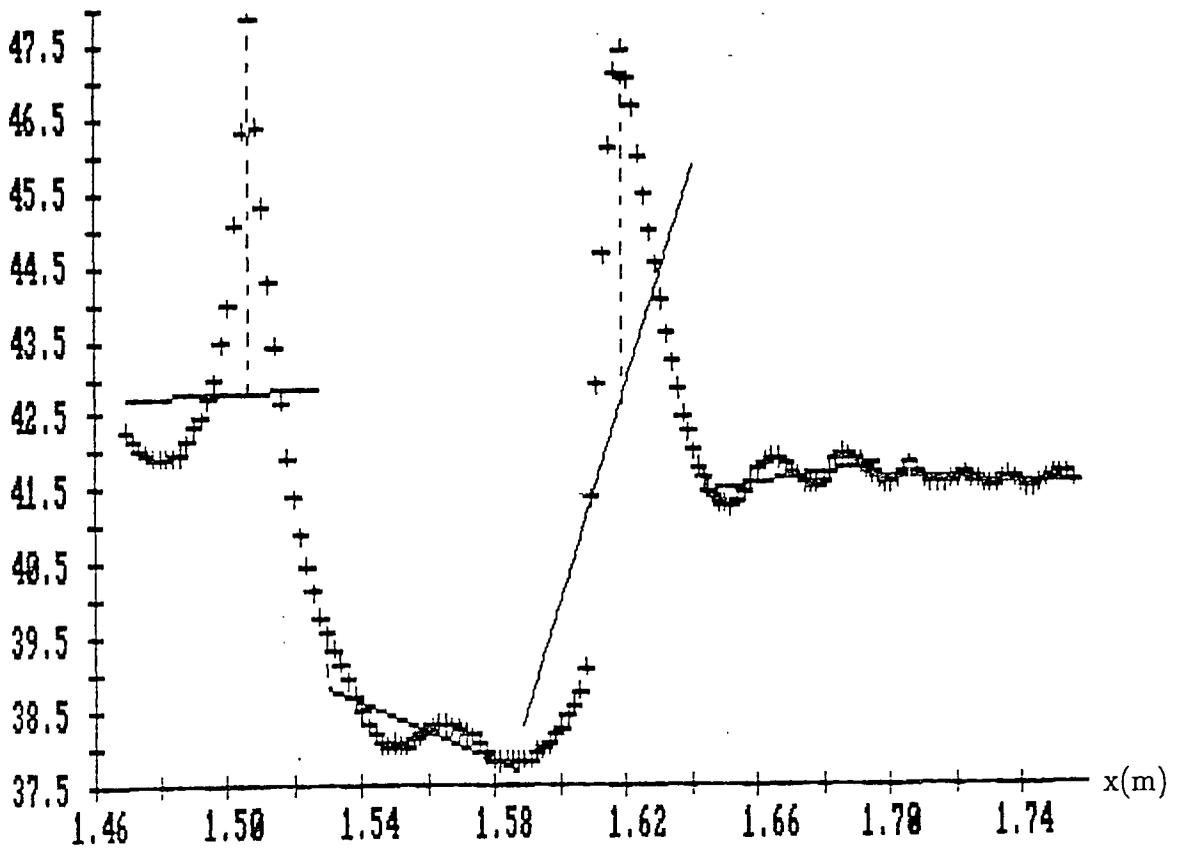


Figure 8.6 The notch detection algorithm

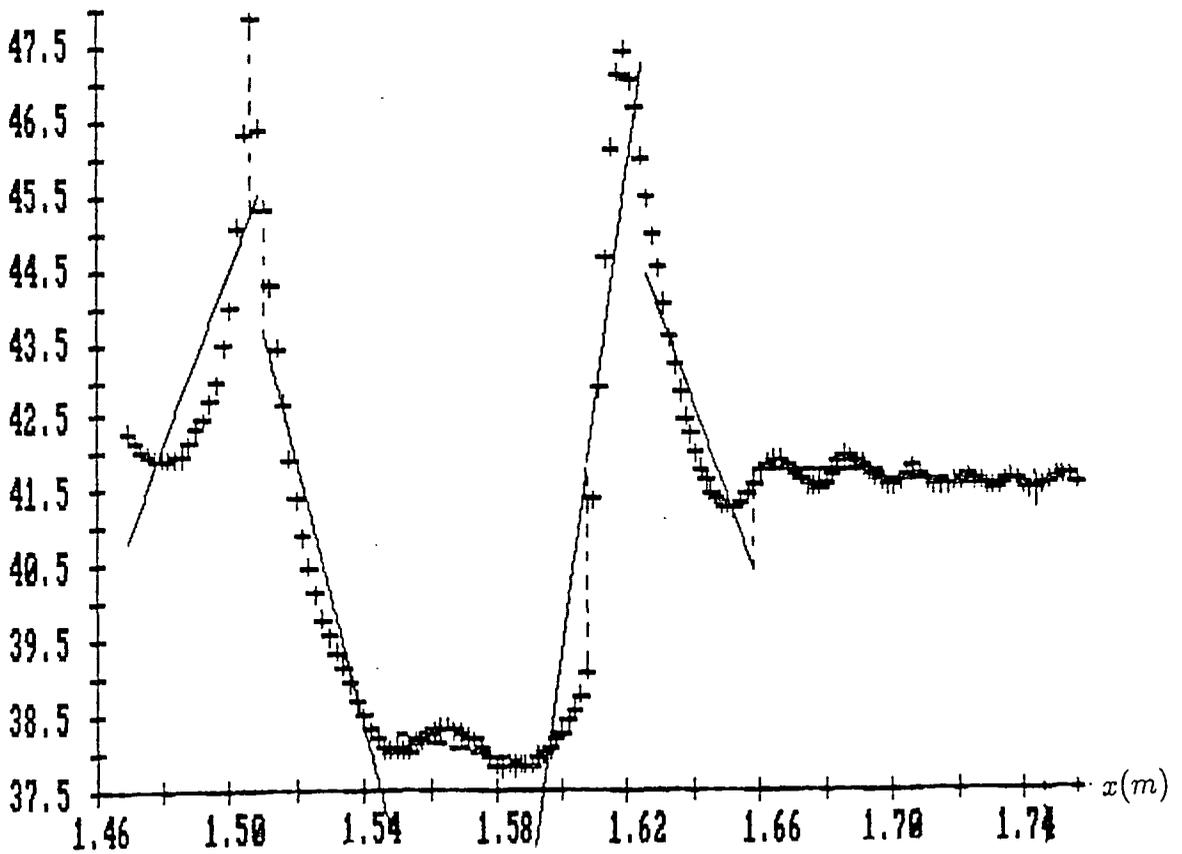
$y_k(cm)$

8.7a Regression lines with 30 data points



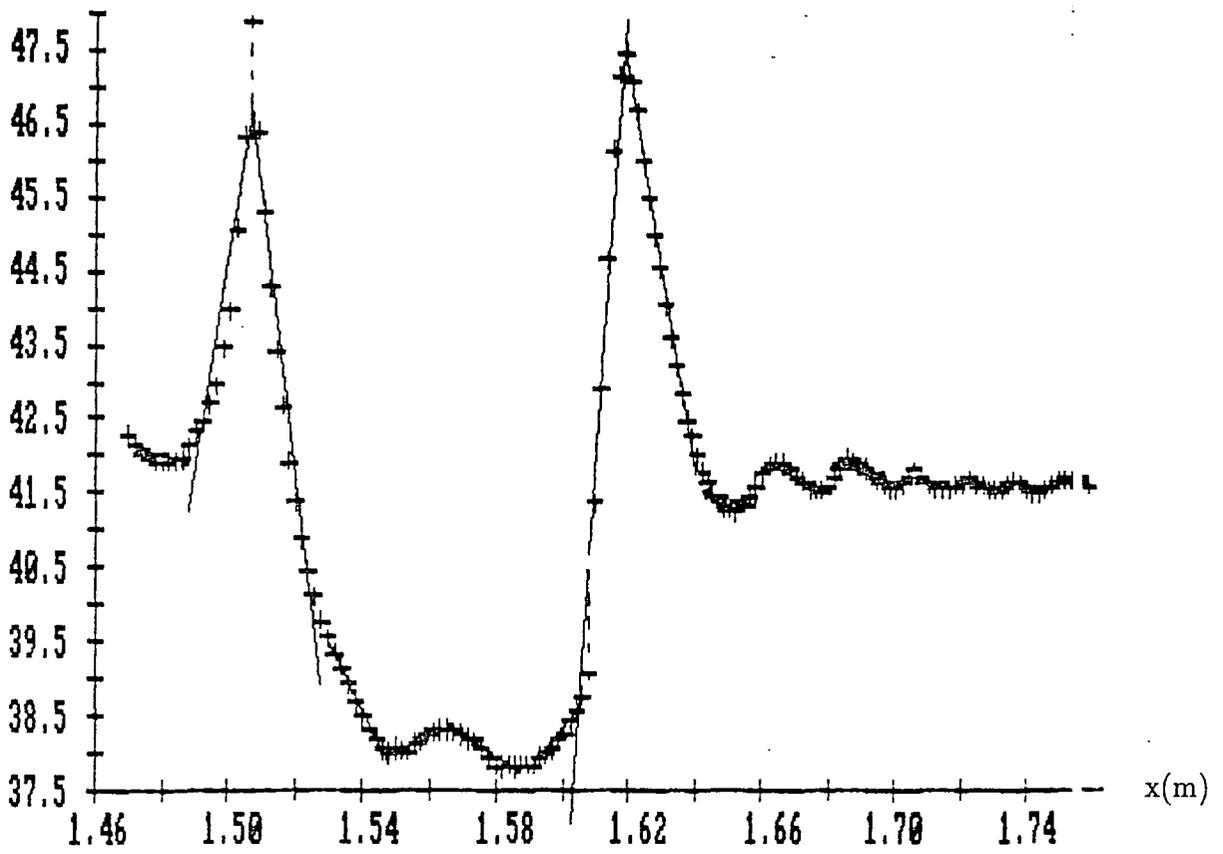
8.7b Regression lines with 20 data points

$y_k(cm)$



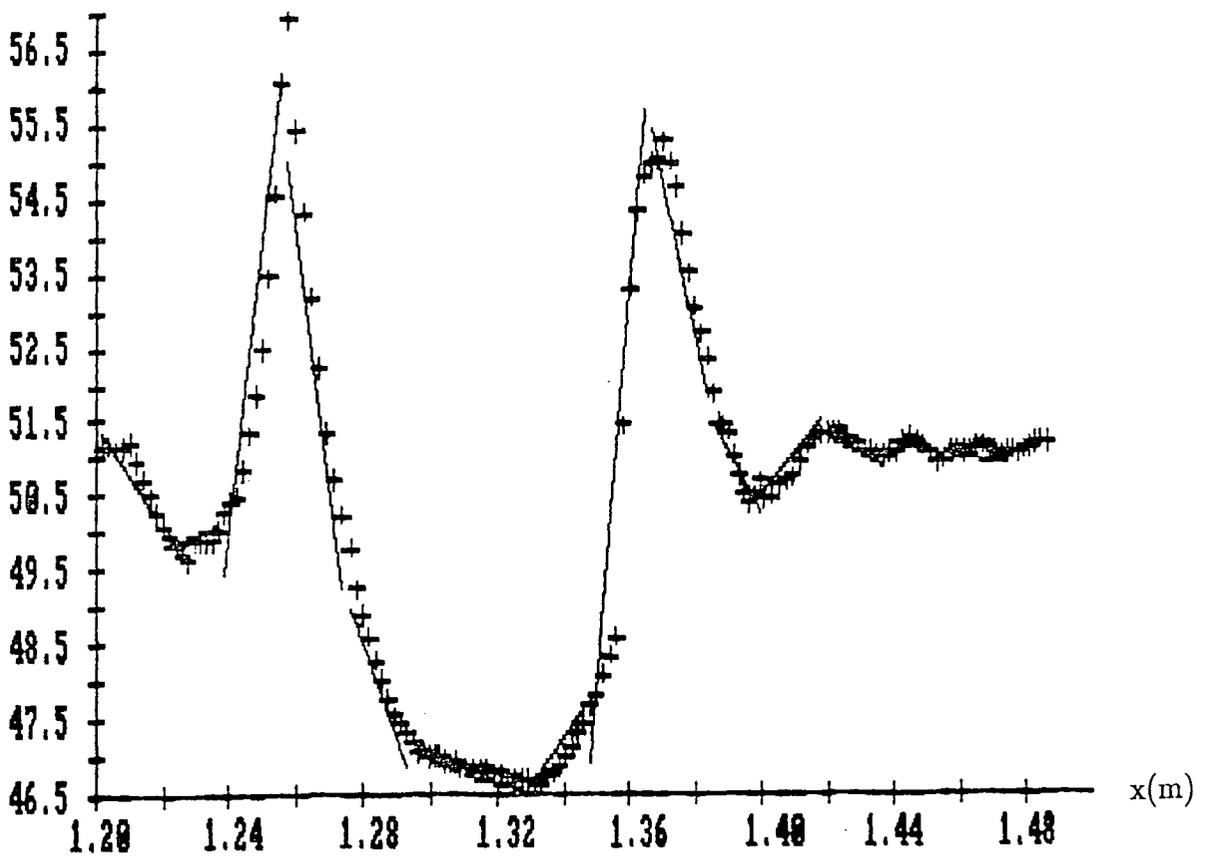
8.7 Regression lines derived off line from ultrasonic data

$y_k(cm)$



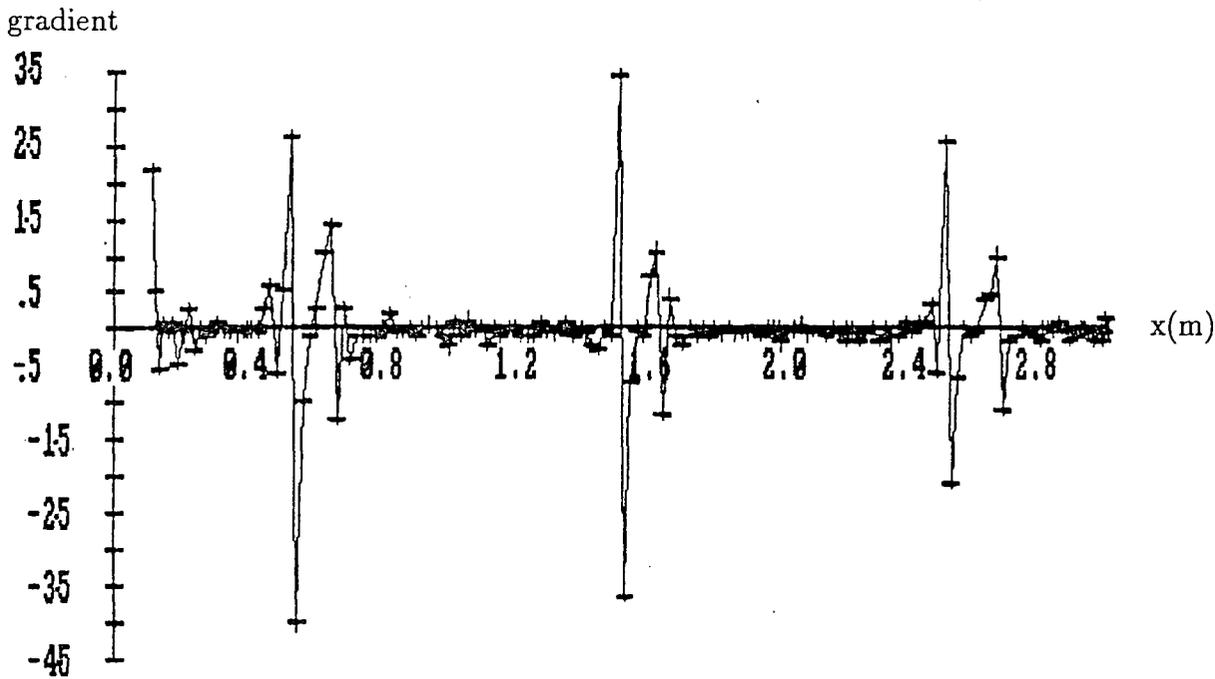
8.8a Regression lines with 10 data points and line refit

$y_k(cm)$

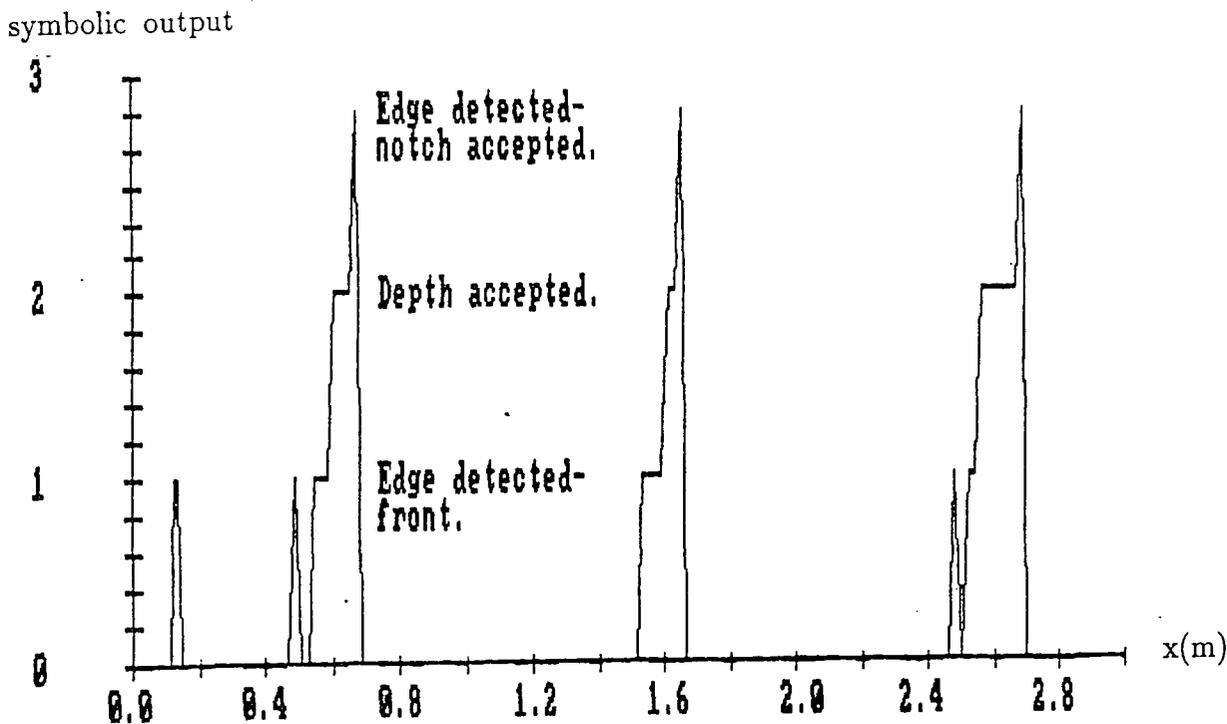


8.8b Regression lines derived in a real time run

8.9a Regression line gradients as the AGV moves past a correction board



8.9b Operation of the notch detection algorithm



8.9 Results from the notch detection algorithm

Figure 8.10a Graph of distance error against time

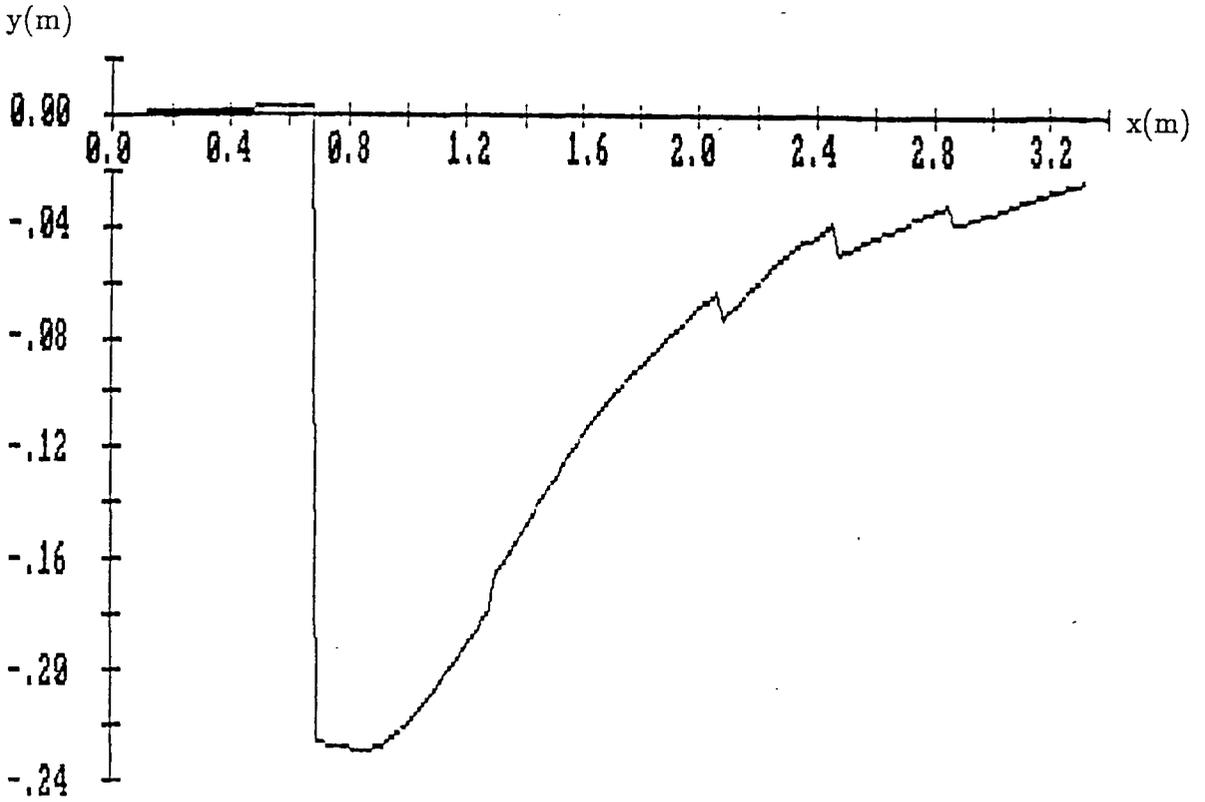


Figure 8.10b Graph of local angle ( $\theta'$ ) against time



Figure 8.10 Performance of the vehicle with low frequency corrections

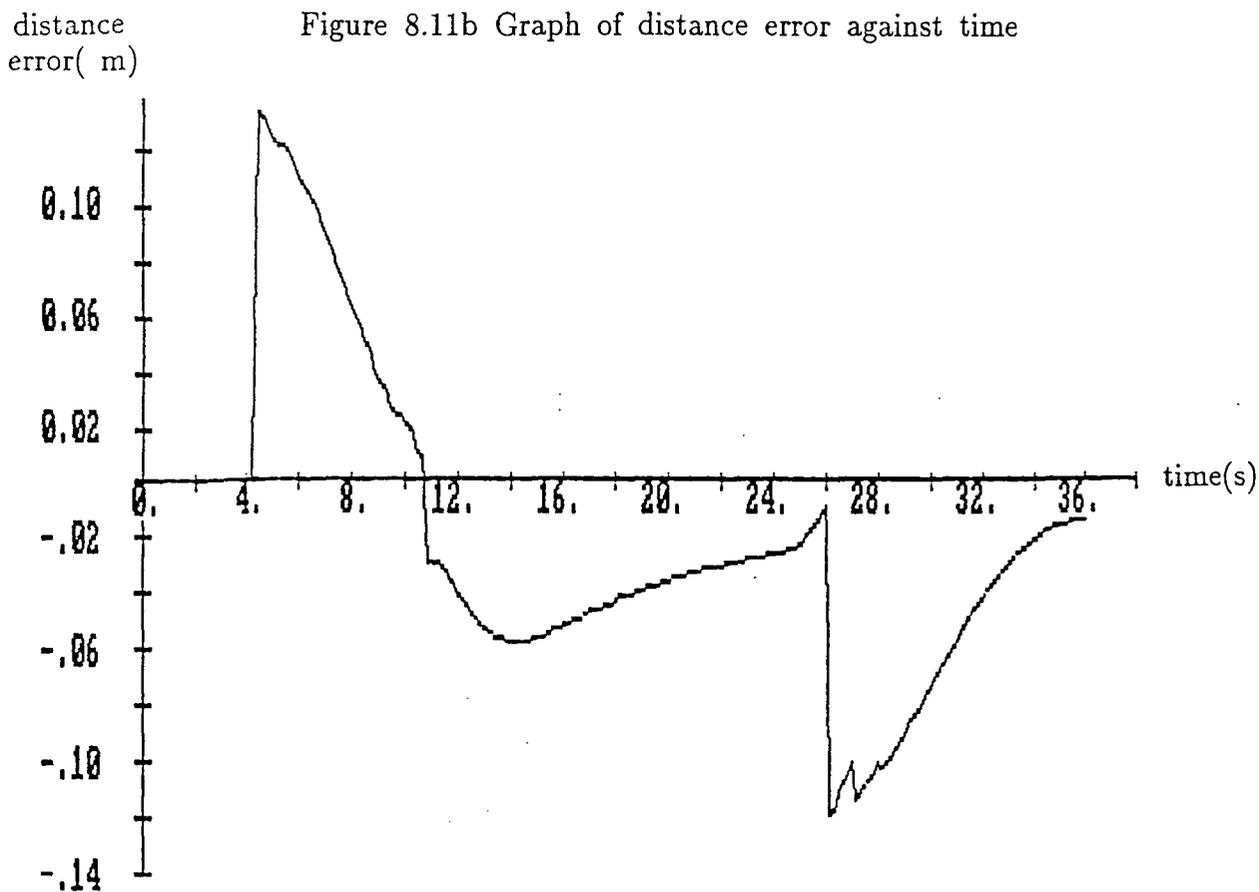
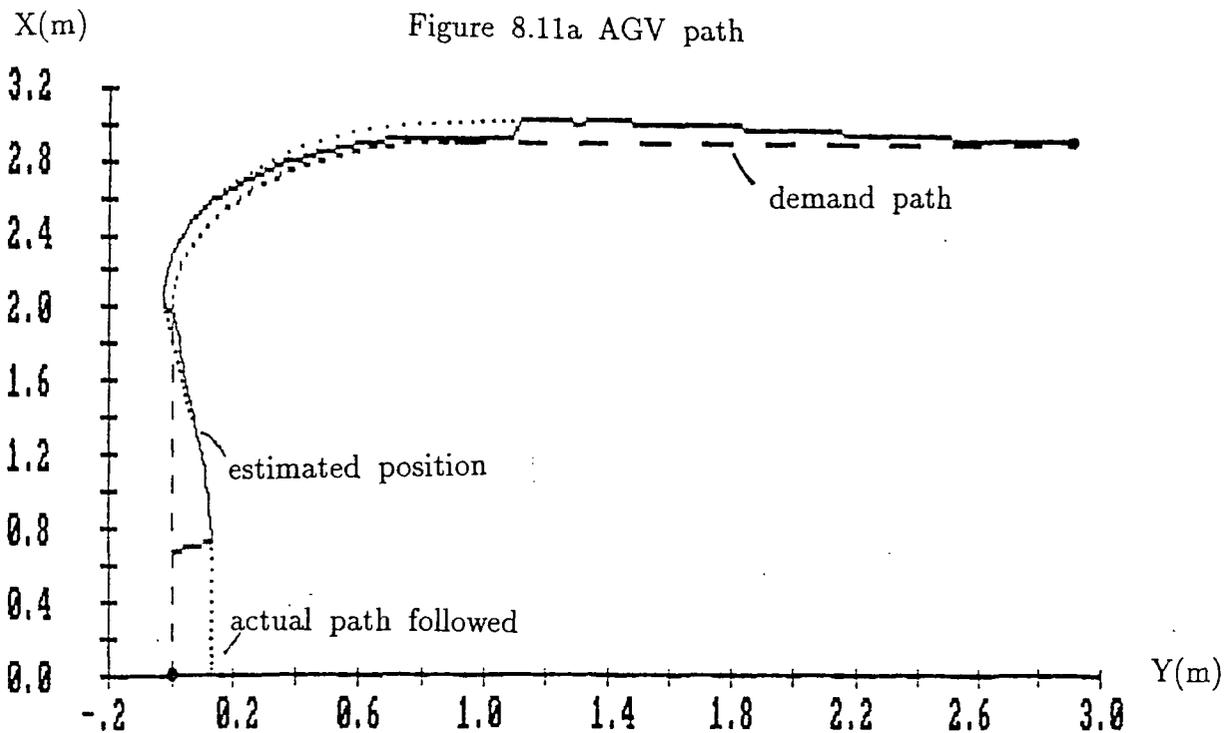


Figure 8.11 Overall performance of the vehicle

test number	$X$ error(cm)	$Y$ error(cm)	$\theta$ error(degrees)
1a	+2.0	-20.0	-3.4
2a	-2.5	-6.0	-1.3
3a	-4.5	-6.0	-2.5
4a	+1.5	+5.0	+0.8

Table 8.1: Errors in uncorrected position after travelling 5m

test number	$X$ error(cm)	$Y$ error(cm)	$\theta$ error(degrees)
1b	+1.5	-11.0	-2.5
2b	-0.75	-14.0	-1.3
3b	-4.25	-2.0	-2.1
4b	+2.25	+6.0	+0.4

Table 8.2: Errors measured in odometric position for re-run tests

test number	$X$ error(cm)	$Y$ error(cm)	$\theta$ error(degrees)
1b	+0.7	-1.6	-1.4
2b	-1.3	-1.4	-1.1
3b	-1.2	-1.5	-1.2
4b	+1.4	+0.5	+0.6

Table 8.3: Errors measured in position with position corrections

## CHAPTER 9

### SUMMARY, GENERAL CONCLUSIONS, AND SUGGESTIONS FOR FURTHER WORK

#### 9.1 Summary and General Conclusions

The work presented in this thesis has been concerned with the development of a method to guide an automated vehicle using low cost, low level sensors. In chapter 1 it was appreciated that much research has been done on a variety of topics related to the field of AGVs and mobile robots. In accordance with the interests of the sponsors of this project, attention was focused on the type of behaviour, the function, and the degree of autonomy required of AGVs operating in an industrial environment. The static nature of both an industrial AGV's environment and the type of task it has to perform engendered the formulation of a guidance method using low level sensors. The function of such a guidance system was to allow the AGV to travel to a bay or work-station to an accuracy of a few centimetres which would then permit more precise location by an alternative method of sensing.

In chapter 2 it was argued that if alternative methods of sensing are required for precise location and intelligent materials handling, then why not try and use such methods to locate the AGV without using added artificial reference points in the environment? It must be remembered that to operate efficiently, the vehicles in an automated warehousing system often move at speeds near to 1m/s. Resolving the ambiguities in the sensory data collected from observations of mapped but natural objects in the environment, in order to provide positional information sufficiently frequently to implement control of a vehicle moving at speed in a cluttered and constrained environment, represents a task which would make high, often unrealisable demands on

the sensory capabilities, processing speed, and reasoning of current research vehicles in this field. However, when an AGV is at or near the location at which it must perform its designated task and is therefore moving much more slowly, more real time is available for the more intelligent processing required to execute the finer movements and manoeuvres required by that task. The next generation of AGVs may have increased sensory capabilities and intelligence that provides them with the flexibility of operation without specialised handling equipment, whilst lower level control and sensing strategies allow the vehicle to move between the various locations for loading, unloading, and executing other designated tasks. It may be concluded then, that there is no real conflict between research into different methods of low level guidance and the various research areas expounded in chapter 2 which aim to provide mobile vehicles with a high degree of autonomy.

Since this research was the first AGV project to be conducted at Durham, and since it seemed unlikely that all of the control and sensing problems could be anticipated in a theoretical approach, a largely experimental approach was adopted. Chapter 3 described the design of a simple research vehicle, its sensory interface boards, and the computing facilities that allowed the implementation of such an approach.

A number of initial aims were outlined in chapter 1; the first of these was the investigation of the processing of low level sensor data. Chapter 4 initially described how odometry and ultrasonic telemetry are used in a complementary manner in order to implement a low level guidance scheme. It then described the results of a number of bench tests that were conducted using the designed ultrasonic ranging system operating at a variety of distances and orientations relative to a wooden reflecting board. A number of idiosyncrasies of measurement were identified. Some were attributed to the inherent properties of ultrasonic pulse propagation and reflection, and some

were found to be due to the method of pulse detection employed. A detailed characterisation of the ultrasonic sensor's performance was required in order to eliminate systematic errors in ultrasonic observations and provide the vehicle with the ability of assessing the accuracy of local position measurement.

The correction boards employed in the research can be thought of as an artificiality added to the environment which is tailored to, and designed to enhance, a low level sensors performance. The boards themselves were seen to be 'smooth' when observed by a 40kHz ultrasonic sensor. This ensured that a normal relative to the board was measured irrespective of the sensors orientation. The wide beam width of the sensors, although a function of distance, generally allowed corrections to be made at angles up to 20 degrees relative to the board.

It was also noted in chapter 4, however, that a wide beam width implies directional ambiguity. This ambiguity could be resolved by using the measurements from two sensors and an iteration based on the sensors characterisation. A flat board and measurements provided by two ultrasonic sensors can only measure position in two degrees of freedom, namely displacement ( $y$ ) and angle ( $\theta'$ ) relative to that board. Features on the correction board, which are simply blocks of wood which have been labelled 'notches', were attached to the board in order to measure the local 'along board' position ( $x$ ) and thus enable a correction to be made in the third dimension. Notches were chosen to be protruding on account of the wide beam width of the ultrasound.

This work has brought to light an important observation concerning low level sensing: those properties of a given sensor type, which are regarded as deficiencies when that sensor operates in a given context, can be used to advantage when operating in a different context. In the case of ultrasound, a wide beam width is often regarded as a deficiency because it engenders

directional ambiguity and can cause recesses to be invisible to the sensor. Spectral reflections are often quoted as a further deficiency in that they cause reflecting surfaces to become invisible when they are orientated outside the effective beam width of that sensor for a given range. However, when a correction board is observed by two sensors simultaneously, the two properties of spectral reflection and wide beam width work together to advantage. Firstly, the wide beam width allows correction board observations to be made over a large range of vehicle orientation. Secondly spectral reflections ensure that the measurement of orientation is accurate since, in order for any reading to be obtained, measured ranges must be taken normal to the correction board.

Chapter 5 described the different types of command that the AGV can execute in order to get from its current location to its destination. In addition, a control strategy was formulated qualitatively. This control system employed both yaw angle and perpendicular distance to the command segment. In formulating a control strategy, the various control and sensing calculations that the AGV needs to make became apparent. Chapter 5 then went on to describe odometric calculations, transformations to a path dependent local coordinate frame, and the transformation of base velocity and curvature into demand velocities for the individual drive wheels.

One of the main aims of the research mentioned in section 1.4 was to design and test modular software algorithms which integrate data processing and control allowing the AGV to move through its environment in a smooth fashion. Chapter 6 has described a modular software package that has been successfully implemented and which integrates data processing and control through structured interrupts. This package was also developed to provide a user interface which effectively takes the role of a central controlling and coordinating computer in the warehouse or factory. The software has proven to be flexible in that it allows vehicle routes to be created, or loaded, and

edited with ease. In addition it allows the correction boards to be either added to, or moved to different positions within, the vehicle's workspace.

Another initial aim mentioned in section 1.4 was to examine some of the control problems which exist in both line and curve following. In chapter 7, the initial formulation, the analysis, and the subsequent modification of the AGV's control system were described. Although the formulated control rule is very simple, results presented in this chapter were able to demonstrate the effectiveness of the control. The success of the control rule is largely due to it linking the vehicle's two degrees of freedom  $y$  and  $\theta'$  relative to the current demand path. The initial control analysis permitted two proportional control constants to be chosen in the correct ratio to give critical damping. It was found that for step disturbances of up to 20cm, the AGV performs well and the results are close to what is predicted. The good performance is accountable to the responsiveness of the stepper motors which effectively allow instantaneous changes of curvature. In addition, the vehicle's maximum heading for a step disturbance of 20cm is 8.5 degrees and so errors in the small angle approximations are small giving good agreement with the predicted results.

A number of modifications to the basic steering control system were also described in chapter 7. Such modifications were required to improve curve tracking performance, make use of the stepper drive's increased torque capability at low velocities, and limit the rate of steering (curvature rate) in order to prevent stepper motor stall.

In chapter 8 the various methods of data processing employed by the research vehicle were described. This processing was implemented by means of a simple recursive filter with an adjustable constant, a recursive line fitting algorithm, a notch detection algorithm, and a sensor characterisation. In the case of the notch detection algorithm, it can be concluded that it is

important to design algorithms on the basis of observed data rather than how a particular object appears to ourselves.

Experiments have indicated that the low level sensing systems employed can operate in a complementary fashion: the reliability of odometric measurements over short distances can be used to reject spurious ultrasonic measurements whilst the accuracy of ultrasonics can be used to constrain the cumulative error in odometric position estimation.

## 9.2 The Contribution of the Presented Work

It should be noted here that, due to the simplicity of deriving data from low-level sensors, odometry and ultrasonic ranging have been employed on a large number of experimental vehicles since autonomous vehicle research began twenty years ago. In addition, current ideas about multi-sensory autonomous vehicles suggest that they will continue to be used. Therefore, it is appropriate to put forward, explicitly, the contribution of the work presented in this thesis. In this work, several new ideas have been described, the main of which can be summarised as:

- (1) The use of a correction board which is tailored to the operation of ultrasonic sensors, and which is observed simultaneously by two such sensors in order to measure local position.
- (2) The generation of a smooth chain of data from the ultrasonic sensors by the use of some simple logic in conjunction with a recursive digital filter.
- (3) The use of line regression in order to improve local position measurement ( $y, \theta'$  measurement) accuracy, and in order to locate simple features on a correction board. Such features

permit the measurement of the third coordinate ( $x$ ) in the local frame parallel to the correction board.

(4) The use of an iterative compensation in order to reduce systematic offset errors in ultrasonic ranging.

(5) The generation and re-setting of odometric error bands which allows the two low-level sensor systems to operate in a complementary fashion.

(6) The formulation of a simple but effective steering control rule.

(7) The development of a novel control system that automatically steers more tightly when travelling at lower speeds, and that spreads the torque demand of the stepper motors over the whole of the steering process.

(8) The formulation of a method of generating the radius and location of simple paths through corners, T-junctions, and crossroads within the laboratory.

### 9.3 Suggestions for Further Work

Experiments described in chapters 7 and 8 concerning the performance of the AGV have shown that both the data processing and control strategies that constitute the proposed low level guidance scheme have proved successful. An assessment of the performance of the AGV (section 8.9), however, has shown that odometric errors build up quickly and thus the AGV, as it stands, has to pass close to correction boards in order to keep the frequency of the corrections high. Part of the reason for this was that, during the course of experimentation, the research AGV had to be moved from a tiled floor to a carpeted floor and the performance of the odometric system was depreciated.

On a tiled floor, only the central part of the drive wheels contacted the tile surface and so the effective wheelbase could be measured accurately. Since the drive wheels are appreciably wide (5cm), and a carpeted floor has some give in it, the whole width of the wheel surface contacted the floor making it impossible to determine exactly what the wheelbase was. (When turning, the wheels must shear against the carpet surface.)

In the light of these remarks, the first priority for future work would be to improve the quality of both of the low level sensing systems. A solution to the odometry problem could be provided by a pair of narrow, high rim friction measuring wheels. These would have to be in line with the drive wheels, since the vehicle pivots about the central point of the drive wheels. In addition they would have to be sprung to ensure that they contact the floor at all times and do not get damaged by the weight of the vehicle.

As has been stated before, it would not be difficult to replace the current low range ultrasonic sensors with ranging sets commercially available from Polaroid. If such a modification was made, the improved 'viewing' range of the vehicle would enable strategies to be developed to make position corrections off mapped but natural reference points in the environment. Cylindrical objects such as structural supports in a warehouse are particularly suited to detection by ultrasound, since they must always present a surface normal to the incident beam. (This is a necessity in the case of spectral reflections.)

If the method of detecting the peak of the echo pulse were employed (see chapter 4), or if a technique such as matched filtering were used as suggested by Steer [STEER and ATHERTON, 1988], the range of the target would not ostensibly increase as it moved away from the beam center line. Thus the need for operating iteratively on a sensor characterisation in order to eliminate offset errors would no longer be required. It has been noted in this research that the directional ambiguity of a target may be reduced by

making observations with more than one sensor. In addition, it should be possible to calculate position by observing two different targets simultaneously. For example, an AGV could fix its position if it were able to observe two perpendicular walls simultaneously.

Another area of interest would be to develop an adaptive method of generating the odometric estimate of position and its associated error bands. If position corrections were consistently made in the same direction (for example, if one wheel were slightly larger than the other), or if position corrections were always made well within the region defined by the odometric error bands, then this suggests that the parameters used to estimate the odometric position and its error bands are wrong, and the nature of the position corrections may provide a basis for the parameter adjustment.

A further modification of the current experimental system would be to improve the definition of the research vehicle's demand path. Currently, demand path consists of a concatenation of simple lines and curves of constant curvature. This means that, when the vehicle moves on to a new command segment, a step input is applied to the vehicle's control system, and the vehicle is temporarily displaced from its demand path. Thus it would be advantageous to be able to input command segments whereby demand curvature changes gradually between the various segments that define the global path.

The work described in thesis has stretched both the available memory and the processing speed of the on-board microcomputer. It would be a simple task to update the processor and increase both the clock frequency and memory of the microcomputer. Although such improvements would enable more sophisticated algorithms to be implemented on the research AGV, the microcomputer would soon become overloaded as the complexity of data processing increased. Depending on the level of sophistication of the proposed future work, it may be better to consider alternative parallel architectures

such as were described in chapter 2.

It is obvious here that future work may take any number of forms. The results of the performance of the data processing and control strategies that constitute the proposed guidance method indicate that the work done so far can provide useful support for a number of research areas in the field of autonomous vehicles.

## REFERENCES

- Albus, J.S., 1981, 'Sensory Interactive Robots' *Annals of the CIRP*, Vol 30, Part 2, pp 559-562.
- Bauzil, G. et al., 1981, 'A Navigation Sub-System Using Ultrasonic Sensors for the Mobile Robot HILARE', *Proc. 1st Annual Conference on Robot Vision and Sensory Control*, Stratford-upon-Avon, U.K., 1981.
- Bevan, M., 1986, 'Free-ranging automatic guided vehicles become reality' *Proc. Automated Guided Vehicle Executive Briefing*, Stratford upon Avon, Nov 1986, pp 151-157.
- Borenstein, J. and Koren, Y., 1988, 'Obstacle Avoidance with Ultrasonic Sensors' *Journal of Robotics and Automation*, Vol4, No2, pp 213-218.
- Brady, M. et al., 1987, 'Progress Towards a System That Can Acquire Pallets and Clean Warehouses' *4th International Symposium on Robotics Research*, MIT Press.
- Brady, M. et al., 1989, 'Sensor-based control of AGVs' *The 1st Workshop on Domestic Robots and the 2nd Workshop on Medical and Healthcare Robotics*, Newcastle-upon-Tyne, September 1989, pp173-188.
- Brooks, R.A., 1986, 'A Robust Layered Control System for a Mobile Robot' *IEEE Journal of Robotics and Automation*, Vol RA-2, No 1, March 1986, pp 14-23.

**Brophy, B.**, 1985, 'Radio-The Key to Effective Communication and control' *Logist Today (GB)*, Vol 4, No 1, pp 10-11.

**Canali, C., et al.**, 1982, 'A Temperature Compensated Ultrasonic Sensor Operating in Air for Distance and Proximity Measurements', *IEEE Transactions on Industrial Electronics*, Vol IE-29, No 4, pp 336-341.

**Chande, P.K. and Sharma, P.C.**, 1984, 'A Fully Compensated Ultrasonic Sensor for Distance Measurement', *IEEE Transactions on Instrumentation and Measurement*, Vol IM-33, No 2, pp 128-130.

**Chatila, R. and Laumond, J.P.**, 1985, 'Position Referencing and Consistent World Modeling for Mobile Robots' *Proc. IEEE Conf. on Robotics and Automation*, St Louis, pp 138-145.

**Clemence, G.T. and Hurlbut, G.W.**, 1983, 'The Application of Acoustic Ranging to the Automatic Control of a Ground Vehicle', *IEEE Transactions on Vehicular Technology*, Vol VT-32, No 3, pp 239-244.

**Cormier, W.H. and Fenton, R.E.**, 1980, 'On the Steering of Automated Vehicles-A Velocity-Adaptive Controller', *IEEE Transactions on Vehicular Technology*, Vol VT-29, No 4, pp 375-385.

**Crowley, J.L.**, 1985, 'Navigation for an Intelligent Mobile Robot', *IEEE Journal of Robotics and Automation*, Vol RA-1, No 1, March 1985, pp 31-41.

**Draper and Smith**, 1981, 'Applied Regression Analysis', *John Wiley and Sons Inc.*, Second Edition.

Durrant-Whyte, H.F., 1987, 'Consistent Integration and Propagation of Disparate Sensor Observations' *The International Journal of Robotics Research*, Vol 6, No 3, Autumn 1987, pp 3-24.

Durrant-Whyte, H.F., 1988, 'Uncertain Geometry in Robotics' *IEEE Journal of Robotics and Automation*, Vol 4, No 1, February 1988, pp 23-31

Elfes, A.E. and Talukdar, S.N., 1983, 'A Distributed Control System for the CMU Rover', *Proc. 1983 IJCAI*, pp 830-833.

Elfes, A.E., 1986, 'A Sonar Based Mapping and Navigation System' *Proc IEEE Conf on Robotics and Automation*, San Francisco, 1986, pp 1151-1156.

Elfes, A.E., 1987, 'Sonar-Based Real-World Mapping and Navigation' *IEEE Journal of Robotics and Automation*, Vol RA-3, No 3, June 1987, pp 249-265.

*Engineering.*, 1985, 'Designed to Deliver', May

Everett, H.R., 1985, 'A Multielement Ultrasonic Ranging Array' *Robotics Age*, July 1985, pp 13-20.

Fenton, R.E., et al., 1976, 'On the Steering of Automated Vehicles: Theory and Experiment', *IEEE Transactions on Automatic Control*, Vol AC-21, No 3, pp 306-315.

Fikes, R.E., et al., 1972, 'Learning and Executing Generalised Robot Plans', *Artificial Intelligence*, Vol 3, Part 4, pp 251-288.

- Flynn, A.M., 1985, 'Redundant Sensors for Mobile Robot Navigation' *MSc Thesis, Massachusetts Institute of Technology*
- Flynn, A.M., 1988, 'Combining Sonar and Infrared Sensors for Mobile Robot Navigation' *International Journal of Robotics Research*, Vol7, No 6, Dec 1988, pp 5-14.
- Francis, N.J., 1987, 'Final Year Report', Department of Computer Science, University of Warwick.
- Giordano, F., 1981, 'Two Way Radio Link Between Automatic Guided Vehicles and System Control Computer' *Proc. 1st Int. Conf. on Automated Guided Vehicle Systems*, Stratford upon Avon, pp 89-83.
- Giralt, G. et al., 1979, 'A Multi-Level Planning and Navigation System for a Mobile Robot: A First Approach to Hilare', *Proc. Int. Symp. of Robotics Research*, Tokoyo, pp 335-338.
- Giralt, G., et al., 1983, 'An Integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots', *Robotics Research 1, Cambridge MA. MIT*, pp 191-214.
- Giralt, G., 1984, 'Mobile Robots' *Nato Advanced Study Institute of Robotics and Artificial Intelligence*, pp 365-393
- Harmon, S.Y. et al., 1986, 'Sensor Data Fusion Through a Distributed Blackboard', *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, pp 1449-1454.

Hammond, G.C., 1986, 'Evolutionary AGVs-from Concept to Present Reality' *Proc. Automated Guided Vehicles Executive Briefing*, Stratford upon Avon, Nov 1986, pp 1-8.

Harmon, S.Y., 1987, 'The Ground Surveillance Robot (GSR): An Autonomous Vehicle Designed to Transit Unknown Terrain' *IEEE Journal of Robotics and Automation*, Vol RA-3, No 3, June 1987, pp 266-279

Hongo, T., et al., 1987, 'An Automatic Guidance System of a Self-Controlled Vehicle', *IEEE Transactions on Industrial Electronics*, Vol IE-34, No 1, pp 5-10.

Jackson, P., 1986, 'Supervisory control of AGVs in materials handling computer systems' *Proc Automated Guided Vehicles Executive Briefing*, Stratford upon Avon, Nov 1986, pp 59-64.

Jaffe, J.L., 1985, 'Polaroid Ultrasonic Ranging Sensors in Robotic Applications' *Robotics Age*, March 1985, pp 23-30.

Johnson, D.P., 1988, 'Development of an Ultrasonic Rangefinder' *A Final Year Honours Project, School of Engineering and Applied Science, Durham University.*

Kanade, T., et al., 1986, 'Autonomous Land Vehicle Project at CMU', *Proc. 1986 Computer Conf.*, Cincinnati, pp 71-80.

Kanayama, Y., et al., 1981a, 'A Locomotion Control System for Mobile Robots', *Proc of 7th IJCAI*, pp 779-784.

Kanayama, Y., et al., 1981b, 'Elementary Functions of a Self Contained Robot "Yamabico 3.1"', *Proc of 11th ISIR*, pp 211-218.

Khatib, O., 1985, 'Real-time Obstacle Avoidance for Manipulators and Mobile Robots', *Proc IEEE Conference on Robotics and Automation*, St Louis, March 1985, pp 500-505.

Kim, O.H., 1987, 'Optimal Steering Control of an Auto-Guided Vehicle with Two Motored Wheels', *Transactions of the Institute of Measurement and Control*, Vol 9, No 2, pp 58-63.

Komoriya, K. et al., 1986, 'A Method of Autonomous Locomotion for Mobile Robots', *Advanced Robotics*, Vol 1, No 1, pp 3-19.

Krough, B.H. and Thorpe, C.E., 1986, 'Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles', *Proc. IEEE Conf. on Robotics and Automation*, St Louis, pp 1664-1669.

Lewis, R.A. and Johnston, A.R., 1977, 'A Scanning Laser Rangefinder for a Robotic Vehicle', *Proc. IJCAI-5, Cambridge, MA.*, pp 762-768.

Luo, R.C. et al., 1988, 'Dynamic Multi-Sensor Data Fusion System for Intelligent Robots' *IEEE Journal of Robotics and Automation*, Vol 4, No 4, August 1988, pp 386-396.

**Materials Handling News.**, 1985, 'Rugged AGV Totes Three-Tonne Load', February

Miller, J.A., 1977, 'Autonomous Guidance and Control of a Roving Robot', *Proc. IJCAI-5, Cambridge, MA.*

M<sup>c</sup>Kerrow, P.J., 1989, 'Simulation of Sonar Echolocation' *The 1st Workshop on Domestic Robots and the 2nd Workshop on Medical and Healthcare Robotics*, Newcastle-upon-Tyne, September 1989, pp127-136.

Moravec, H.P. et al., 1985, 'High Resolution Maps from Wide Angle Sonar' *Proc. IEEE Int. Conf. on Robotics and Automation*, St Louis, pp 116-121.

Nilsson, N.J., 1969, 'A Mobile Automaton: An Application of Artificial Intelligence Techniques', *Proc 1st IJCAI*, pp 509-520.

Pears, N.E., 1989, 'Software Documentation and Operation Manual for the Autonomous Vehicle Research Project', *School of Engineering and Applied Science, Durham University.*

Pears, N.E. and Bumby, J.R., 1989, 'Guidance of an Autonomous Guided Vehicle Using Low Level Ultrasonic and Odometry Sensor Systems' *Transactions of the Institute of Measurement and Control*, Vol 11, No 5, pp 231-248.

Pears, N.E. and Bumby, J.R., 1989b, 'The Steering Control of an Experimental Autonomous Vehicle' Paper submitted to *Transactions of the Institute of Measurement and Control*

Polaroid corporation., 1982, 'Ultrasonic ranging system'

Shafer, S.A. et al., 1986, 'An Architecture for Sensor Fusion in a Mobile Robot' *Proc. 1986 IEEE Conf. on Robotics and Automation*, pp 2002-2011.

- Steer, B.**, 1985, 'Navigation for the Guidance of a Mobile Robot', *PhD Thesis*, pp 75-98.
- Steer, B. and Atherton, T.J.**, 1988, 'On Sonar Sensing for Mobile Robots',  
To be published
- Stephens, P. et al.**, 1983, 'Truck Location Using Retroreflective Strips and Triangulation with Laser Equipment' *Proceedings from 2nd European Conference on Automated Manufacture*, Birmingham, May 1983, pp 271-282.
- Thompson, A.M.**, 1977, 'The Navigation System of the JPL Robot', *Proc. IJCAI-5*, Cambridge, MA., pp 749-757
- Tsuji, S.**, 1985, 'Monitoring of a Building Environment by a Mobile Robot', *Robotics Research 2*, pp 349-356.
- Wallace, R., et al.**, 1986, 'Progress in Robot Road Following', *Proc. IEEE Conf. on Robotics and Automation*, San Francisco, pp 1615-1621.
- Warwick, K., Brady, J.M.**, 1986, 'A Bibliography for Autonomous Guided Vehicle Research', *Report No OUEL 1631/86*, University of Oxford
- Yakolev K.P.**, 1965, 'Handbook for Engineers', Vol 1, pp 189-191.

## APPENDIX 1

### A review of experimental AGV and mobile robot projects

In this appendix synopses are given of several major experimental autonomous vehicle projects that have taken place over recent years. In particular, sensing abilities, computational resources, vehicle construction, and the aims and important areas of research are mentioned. It should be noted that autonomous vehicle research is a rapidly expanding field and the projects mentioned here by no means constitute a comprehensive list. A more complete list of mobile robot projects undertaken before 1984 can be found in a paper by Giralt [GIRALT, 1984]. A comprehensive reference for autonomous vehicle research can be found in a bibliography by Warwick and Brady [WARWICK and BRADY, 1986].

#### A1.1 SHAKEY (The Stanford Research Institute, 1968-1973)

SHAKY was built in the late sixties in order to test the application of a number of artificial intelligence techniques and to "study processes for the real time control of a robot system that interacts with a complex environment" [NILSSON, 1969]. SHAKEY's sensory systems consisted of a television camera and optical range finder in a movable head. In addition it had 'cats whisker' touch sensors attached to its perimeter and an odometric ability. Its computing facility consisted of a time shared SDS-940 computer which could communicate with the vehicle over two special radio links. SHAKEY has been tested in a simple polyhedral world where it was given the task of rearranging (by pushing) simple objects. SHAKEY's method of visual perception employed a digitised picture which was stored on the computer as a 120 by 120 array of 4-bit (16-level) intensity values. A scene analysis using intensities then yielded

a line drawing from which open floor areas could be deduced. Information about open floor areas would then be incorporated into a 2-D grid model which could be used by the vehicle to plan routes to its goal position.

An important part of the SHAKEY project was the development of a robot problem solving system called STRIPS. The function of this was to produce a plan consisting of a sequence of pre-programmed actions that would change the world in a manner defined by the posed goal [FIKES, 1972]. A system called PLANEX was designed to supervise the execution of the sequence in order to accomplish the required task.

### **A1.2 The JPL robot (Jet Propulsion Laboratory, California, 1975-1979)**

This research program was aimed at "developing the capabilities in machine intelligence systems required for a semi-autonomous vehicle to be used in remote planetary exploration" [THOMPSON, 1977]. The vehicle's sensory systems consisted of stereo TV cameras, a laser rangefinder [LEWIS, 1977], a gyrocompass, and optical encoder odometers. It was controlled by a real-time local minicomputer which was linked to a large remote time-shared computer. The vehicle's function was to analyse scenes for traversability, generate a planned path to its goal, and then to follow that path avoiding any obstacles on the route. To do this, a segmented terrain model had to be created and maintained which was quantified in terms of traversability (i.e a two dimensional representation). The path planner used this model to implement an optimal tree structure path search which yielded a low cost route. Experiments were conducted in a simplified environment consisting of "a laboratory with a flat surface, a limited number of obstacles and constant illumination" [MILLER, 1977].

### A1.3 The CMU Rover (Carnegie-Mellon University, Pittsburg)

The CMU rover is a vehicle "intended to support a variety of research in the areas of perception, control, real world modelling, problem solving planning and related issues" [ELFES and TALUKDAR, 1983]. The sensors employed include a TV camera, an array of ultrasonic sensors, and a set of proximity sensors. Initial work done by Moravec at CMU was concerned with stereo imaging. In this method, digitised images of a view were taken from several aspects. These images were then processed and incorporated into an environmental model which, along with odometry, provided the vehicle with a method of guidance to its required destination.

Primarily because of the high computational expense, stereo vision navigation systems are very slow. Attempts were made to speed the process up and as a result, only sparse depth maps could be built. CMU's fastest system takes between 30 and 60 seconds per one metre step [MORAVEC, 1985].

These problems of high computational expenditure prompted Moravec to the use of multiple wide angle sonar range measurements which could provide much denser maps with considerably less computation. Range measurements were taken with a ring of 24 sensors mounted on the vehicle. These measurements were then integrated into a probability map, the definition of which would improve as the vehicle wandered and more readings were taken. Ultimately, a map would exist with regions which could be described as probably occupied, probably unoccupied, and empty [ELFES, 1986, 1987].

### A1.4 HILARE (Laboratoire d'Automatique et d'Analyse des Systemes du CNRS)

This project started in September 1977 at LAAS and its function

was to "provide a flexible and powerful experimental support for advanced research into robotics" [GIRALT et al, 1979]. Again the vehicle is equipped with a number of sensory devices: a video camera and laser rangefinder mounted on a two axes scanning system, a number of ultrasonic transducers around its perimeter, infrared beacons for absolute positional information, and a odometric facility [GIRALT, 1983]. HILARE's computing facilities are organised on three levels. Firstly on-board microcomputers (INTEL 8085 based) manage the vehicle's perception and locomotion systems. An off-board SEMS MITRA minicomputer is used for data processing and elaborating tasks associated with the navigation of the robot. This minicomputer is connected to a high powered IBM 370/168 which can intervene during short or particular periods in the functioning of the robot.

#### **A1.5 A Ground Surveillance Robot (NOSC, San Diego)**

A ground surveillance robot has been built with a view to navigation from one known geographic location to another over completely unknown natural terrain [HARMON, 1987]. The vehicle is a modified M114 armoured personal carrier: a tracked vehicle with a gasoline engine and a 10kW generator set. The vehicle has attitude sensors to determine roll, pitch and heading angle, forward speed, and rotational velocity. In addition there are odometry sensors, a satellite navigation system, seven Polaroid ultrasonic sensors, a laser rangefinder, and two cameras (one high resolution and one colour).

The system is programmed in PLM and is partitioned into a number of intelligent subsystems for sensing, mapping, and control. These subsystems can access a local area network via an intelligent communications interface and thus gain access to several pieces of shared memory distributed through those subsystems. This paradigm is conceptually termed a blackboard structure; the blackboard itself (shared memory) permits the communication of data between

subsystems, holds a "class tree" world model, and provides a mechanism for data fusion.

### **A1.6 An Autonomous Land Vehicle (CMU 1985-)**

At CMU an autonomous vehicle intended for research has been created by modifying a commercial truck [KANADE, 1986] [SHAFER, 1986]. The vehicle has an odometric and inertial navigation system, a pair of stereo cameras, a laser rangefinder, and sonar sensors. The on-board computing facilities consist of four SUN workstations connected by Ethernet and a general purpose systolic array computer used for high speed vision processing.

### **A1.7 The terragator (CMU)**

The terragator is a six wheeled vehicle constructed at CMU [WALLACE et al,1986]. The vehicle has two TV cameras mounted at the front which it can use to execute a road following algorithm. This algorithm is performed on a WARP array and SUN workstation. The road to be followed is segmented from the background using both colour and shape. The results of image processing are used to steer the vehicle down the centre of the road. The terragator communicates with WARP using either a radio or cable link.

### **A1.8 The M.I.T. AI Lab mobile robot (M.I.T.)**

A mobile robot has been built at MIT in order to implement and test a control system based around "task-achieving behaviours" as opposed to a traditional functional decomposition of the control problem [BROOKS, 1986]. This robot employs an Intel 8031 as its main on board processor, although most of the processing occurs off board on a Lisp machine. The vehicle is driven by three parallel drive wheels which are servoed to a single

microprocessor. The vehicle has a ring of twelve Polaroid ultrasonic sensors, drive wheel encoders, and two Sony CCD cameras. The aim of this research is to build up modular layers of a control architecture; each module is an instance of a finite state machine and is programmed in Lisp. It is proposed that such an approach serves the requirements of multiple goals, multiple sensors, extensibility, and robustness.

#### **A1.9 An Intelligent Mobile Platform (1985)**

An experimental mobile robot called an intelligent mobile platform (IMP) supported an independent research effort by Crowley [CROWLEY, 1985]. The vehicle is three wheeled, has two on board 16 bit microprocessors and employs an odometric system, touch sensors and a rotating ultrasonic transducer for guidance and navigation. The navigation principle is based around maintaining a local model of the immediate environment by integrating information from the rotating range sensor, the touch sensor, and a pre-learned global map. Position corrections are made by comparing an interpretation of the environment from the most recent sensor scan with the current local model.

#### **A1.10 The Osaka University mobile robot (1985)**

A simple vehicle has been constructed at Osaka university with a view to the monitoring of a large man made area such as plants, buildings and factories [TSUJI, 1985]. A TV camera, two lights, an odometric system and micro-computer are mounted on board the mobile platform. The image from the camera is converted into a 256 by 256 digital picture (4 bits) and sent to an off board mini computer for processing. The vehicle operates by moving a few metres along a given route, stopping every few metres to analyse an

image, and reporting any changes to a human operator. Initially the vehicle uses mismatches in analysed scene and work space to correct position and then with this known new position, it can detect obstacles and changes in the scene.

## APPENDIX 2

### Calculations of shaft encoder interface circuit parameters

#### A2.1 DC filter

A  $10 \mu F$  capacitor was required to filter a 1V dc level on the shaft encoder signal. However this must not filter out the signal itself when the AGV is moving slowly. The cut off frequency for a simple first order RC high pass filter is

$$f = \frac{1}{2\pi RC} \quad (A2.1)$$

now  $R = 1M\Omega$  (an op-amp property) so:

$$f = 0.016Hz$$

The slowest that a stepper motor can move is 6sps. Since there are 200 steps per revolution, and the gear-ratio is 7:36, this equates to a minimum drive wheel angular velocity of  $5.8 \times 10^{-3}$  revs/sec. Thus the lowest shaft encoder pulse frequency is 0.58Hz and the DC filter cut off is low enough.

#### A2.2 Amplifier / low pass filter

The amplitude of the pseudo-sine wave encoder signal was measured to be approximately 0.2V. (The data sheet for the encoders quotes a higher amplitude, which may be obtained by proper alignment of the encoder and slotted disc.) The degree of amplification is given by the ratio of feedback resistance to input resistance and was arbitrarily chosen to give a reasonably sized signal. This low frequency gain is 22 giving a signal amplitude of 4.4V.

A small capacitor was paralleled with the feedback resistor to filter high frequency noise. This capacitor had to be large enough to filter noise

but not so large as to filter the highest frequency signals. At the time of developing the shaft encoder interface, the maximum velocity of the AGV was anticipated to be around 1m/s whilst, in practice, maximum velocity was found to be around 0.5m/s. A maximum speed of 1m/s gives a frequency from the encoder of 159Hz whilst a 3.9nF feedback capacitor with the 220k $\Omega$  feedback resistor gives a cut off frequency of 185.5Hz.

### A2.3 Comparator with hysteresis band

High frequency ringing was found on the reference input and signal input of the comparator despite the filtering included in the amplifying op-amp stage. The amplitude of this was 200mV peak to peak which meant that a hysteresis band of at least 400mV was required. It can be shown that for the op-amp comparator, the voltage swing of the hysteresis band is:

$$V_h = 2V_{cc} \frac{R}{R + R_f} \quad (A2.2)$$

In the shaft encoder interface circuit

$$R = 1k\Omega$$

$$R_f = 56k\Omega$$

$$V_{cc} = 15V$$

which gives:

$$V_h = 526mV$$

This is sufficient to prevent extra triggering from noise.

## APPENDIX 3

### Equations required for basic path planning

In order to aid the input of the correct command parameters in the command sequence, a simple method of off-line planning can be implemented at right angle corners, T-junctions, and crossroads. As explained in section 6.8 a "planning" routine can be called when a command queue is being built which requests the coordinates of the inside corner ( $X_{corner}, Y_{corner}$ ) and the width of the aisle ( $a$ ) into which the vehicle is entering. From this data and a pre-defined clearance ( $c$ ), the required radius, centre of curvature, and starting position of the required curve segment are calculated. The method of radius calculation is illustrated in figure 6.4 and from this it is clear that the vehicle passes closest to a wall at the two critical positions labelled with the predefined clearance ( $c$ ). It can be seen that one of these critical positions occurs at 45 degrees whilst the position of the second critical position is less obvious. Assuming that the AGV has turned through an angle  $\theta$  and the starting position in a global coordinate frame (0,0) then:

$$p_f = (X + s_f \cos\theta), (Y + s_f \sin\theta) \quad (A3.1)$$

$$p_{fl} = (X + s_f \cos\theta + \frac{w}{2} \sin\theta), (Y + s_f \sin\theta - \frac{w}{2} \cos\theta) \quad (A3.2)$$

Substituting

$$X = r \sin\theta \quad (A3.3)$$

and

$$Y = r(1 - \cos\theta) \quad (A3.4)$$

gives the 'X' coordinate ( $X_{fl}$ ) of the AGV point  $p_{fl}$  as

$$X_{fl} = \left(r + \frac{w}{2}\right)\sin\theta + s_f\cos\theta \quad (\text{A3.5})$$

By differentiating this quantity, the angle in the curve at which the second critical position occurs can be found.

$$\frac{dX_{fl}}{d\theta} = \left(r + \frac{w}{2}\right)\cos\theta - s_f\sin\theta \quad (\text{A3.6})$$

Equating this to 0 yields

$$\tan\theta = \frac{2r + w}{2s_f} \quad (\text{A3.7})$$

By adding quantities in the aisle of width (a) in figure 6.4

$$a = \left(1 + \frac{1}{\sqrt{2}}\right)c + \frac{1}{\sqrt{2}}\left(\frac{w}{2} - r\right) + X_{fl_{max}} \quad (\text{A3.8})$$

where  $X_{fl_{max}}$  is the value of  $X_{fl}$  given by equation A3.5 at an angle of  $\theta$  defined by equation A3.7.

Rearranging equation A3.8 and substituting for  $X_{fl_{max}}$  gives the function in r:

$$f(r) = \left(1 + \frac{1}{\sqrt{2}}\right)c + \frac{1}{\sqrt{2}}\left(\frac{w}{2} - r\right) + \left(r + \frac{w}{2}\right)\sin\theta + s_f\cos\theta - a = 0 \quad (\text{A3.9})$$

which has the differential

$$f'(r) = \left[\frac{\psi}{1 + \psi^2}\right]\cos\theta + \left[\frac{\psi^2}{1 + \psi^2}\right]\sin\theta - \frac{1}{\sqrt{2}} \quad (\text{A3.10})$$

where

$$\psi = \frac{2r + w}{2s_f} \quad (\text{A3.11})$$

and

$$\theta = \tan^{-1} \psi \quad (\text{A3.12})$$

Newton-Raphson iteration is used to find a solution,  $r$ , for entered dimensions of the aisle ( $a$ ) and the specified clearance ( $c$ ).

The starting position for the execution of the curve can be written down by examining figure 6.4. The distance before the corner at which the AGV should start turning is

$$X_s = \frac{1}{\sqrt{2}} \left( r - c - \frac{w}{2} \right) \quad (\text{A3.13})$$

The distance from the right hand wall to the centre of the AGV is

$$Y_s = r - \frac{1}{\sqrt{2}} \left( r - \frac{w}{2} - c \right) \quad (\text{A3.14})$$

If the AGV is nominally travelling down the centre of the aisle, this calculated starting position requires the vehicle to have attained a specific lateral position ( $Y_s$ ) at a distance ( $X_s$ ) before the corner which is when it starts turning. In addition, after the curve has been executed, the AGV may move back to the centre of the aisle. This motion can be observed when a car swings out to enter a drive and then levels up to enter its garage.

## APPENDIX 4

### A geometrical analysis of the AGV control system

The equations employed by the AGV control system for line and wall following can be written as:

$$\theta'_r = -k_{pd}(y - y_{lw}) \quad (A4.1)$$

$$\kappa = k_p(\theta'_r - \theta') \quad (A4.2)$$

From these equations and a modification of the geometrical model employed by Steer [STEER, 1985], an approximate closed loop transfer function representing the control can be derived.

Consider a vehicle moving towards its imaginary line on a given curvature developed by the controller and at a constant velocity ( $V$ ) as in figure 6. The standard result for curvature in an x-y plane is given by [YAKOLEV, 1965]:

$$\frac{d\theta'}{ds} = \kappa = \frac{\frac{d^2y}{dx^2}}{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{3}{2}}} \quad (A4.3)$$

where  $d\theta'$  is an infinitesimal angle subtending an infinitesimal arc length  $ds$  on the curve. The first differential in the x-y plane is simply:

$$\frac{dy}{dx} = \tan\theta' \quad (A4.4)$$

In direct wall following a measurement of  $\sin\theta$  is made using distance measurements from the front and rear sensors such that:

$$\sin\theta' = \frac{y_2 - y_1}{l} \quad (A4.5)$$

In this analysis it is assumed that the angle of the AGV to a wall, correction board, or imaginary line is small enough to make the approximation:

$$\tan\theta' \approx \sin\theta' \approx \theta' \quad (A4.6)$$

In an experiment that is described in section 7.4.1, the vehicle was subjected to a step disturbance in  $y$  of 20cm and the largest angle ( $\theta'$ ) attained by the AGV was 8.5 degrees. In this case the maximum error incurred by the above approximation is less than one percent.

Using the approximation described in equation A4.6 in equation A4.4:

$$\theta' = \frac{dy}{dx} \quad (A4.7)$$

Since for small angle approximations  $\frac{dy}{dx}$  is considerably less than one, its higher powers are even smaller and equation A4.3, describing curvature, reduces to:

$$\kappa = \frac{d^2y}{dx^2} \quad (A4.8)$$

Combining controller equations A4.1 and A4.2 gives

$$\kappa = -k_p k_{pd}(y - y_{lw}) - k_p \theta' \quad (A4.9)$$

and substituting the approximate differentials for curvature and angle gives:

$$\frac{d^2y}{dx^2} + k_p \frac{dy}{dx} + k_p k_{pd} y = k_p k_{pd} y_{lw} \quad (A4.10)$$

Now

$$\frac{dy}{dx} = \frac{dy}{dt} \frac{dt}{dx} \quad (A4.11)$$

and:

$$\frac{d^2y}{dx^2} = \frac{d^2t}{dx^2} \frac{dy}{dt} + \left(\frac{dt}{dx}\right)^2 \frac{d^2y}{dt^2} \quad (A4.12)$$

As the angle  $\theta$  is assumed small and absolute velocity (V) is constant, then velocity in the x direction is constant and equation A4.10 can be converted to a differential equation in the time domain and Laplace transformed to give:

$$s^2y + k_p V s y + k_p k_{pd} V^2 y = k_{pd} V^2 y_{lw} \quad (A4.13)$$

This is the same result as was obtained from the state space analysis made in section 7.2.

