Guidance of an autonomous guided vehicle using low-level ultrasonic and odometry sensor systems

N.E. Pears and J.R. Bumby

Transactions of the Institute of Measurement and Control 1989 11: 231
DOI: 10.1177/014233128901100503

Guidance of an autonomous guided vehicle using low-level ultrasonic and odometry sensor systems

by N. E. Pears, BSc and J. R. Bumby, BSc, PhD, CEng, MIEE

A method of guidance is described for an autonomous guided vehicle (AGV) operating in a pre-defined and structured environment such as a warehouse or factory. The guidance system requires a series of data describing concatenated curved and straight-line segments to be passed to the vehicle as a representation of a planned path within its environment. The coordinate position of each line segment, the associated AGV direction, and the demand speed constitute a series of commands which are queued and then executed in sequence. In order to execute commands, the AGV is equipped with two-level sensor systems, ultrasonic transducers and odometry, 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. When the AGV passes such a correction board, it can measure its position relative to the board in the local frame of the board. This is then transformed into a global position correction. Position corrections appear as step disturbances to the control system. The control system is arranged so that it allows the vehicle to move back on to its planned trajectory after a position correction in a critically damped fashion.

Keywords: Automated guided vehicles; complementary sensors: AGV control.

List of symbols

a	Width of an aisle (m)
c	Clearance between AGV and a wall at critical positions on a designed path (m)
d_{wheel}	Diameter of a drive wheel (m)
$\bar{d}$	Average distance travelled by both drive wheels (m)
k_{dr}	Odometric constant
k_p	Local angle (θ') proportional control constant
k_{pd}	Local distance (y) proportional control constant
f_c	Clock frequency (Hz)
L	Distance remaining in a line or curve command segment (m)
l	Length of the AGV (m)
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_1	Radius described by the left drive wheel (m)
r_2	Radius described by the right drive wheel (m)
s_f	Lengthwise distance of front sensor from AGV reference point (m)
s_r	Lengthwise distance of rear sensor from AGV reference point (m)
t	Time (s)
T	Time constant of an exponential rise (s)
V_b	Base velocity: velocity of the AGV's reference point (m/s)
V_1	Velocity of the left drive wheel (m/s)
V_2	Velocity of the right drive wheel (m/s)
V_s	Velocity of sound in air (m/s)
V_x	Base velocity component in the local x direction (m/s)
w	Width of the AGV (m)
W_b	Wheel base of the AGV (m)
x, y, θ'	Coordinates in a path-dependent local reference frame
X, Y, θ	Coordinates in a global reference frame
X_0, Y_0, θ_0	Position of local coordinate frame in a global reference frame
x_0	Local starting coordinate for position corrections (m)
x_1	Local terminating coordinate for position corrections (m)
X_{agv}, Y_{agv}	Coordinates of the AGV in a global reference frame
X_c, Y_c	Global coordinates of the centre of a curve segment
$X_{fl_{max}}$	Maximum of the coordinate X_{fl} (m)
X_{line}, Y_{line}	Global coordinates on an imaginary line
X_s, Y_s	Position relative to a corner at which to start turning
X_{start}, Y_{start}	Global starting coordinates of an imaginary line
X_{term}, Y_{term}	Global terminating coordinates of an imaginary line
y_e	Perpendicular distance error from demand path (local frame) (m)
y_{lw}	Separation of demand path and correction board (m)
y_1	Measurement from rear ultrasonic sensor (local frame) (m)

School of Engineering and Applied Science, Durham University, South Road, Durham DH1 3LE, UK.

y_2	Measurement from front ultrasonic sensor (local frame) (m)
Δd_1	Incremental distance travelled by the left drive wheel (m)
Δd_2	Incremental distance travelled by the right drive wheel (m)
ΔX	Change in AGV X coordinate over an odometric interval (m)
ΔY	Change in AGV Y coordinate over an odometric interval (m)
κ	AGV turning curvature (m^{-1})
κ_{demand}	Demand AGV turning curvature (m^{-1})
κ_{seg}	Curvature of current command segment (m^{-1})
θ	Heading in a global reference frame (rad)
θ'	Heading in a path-dependent local reference frame (rad)
θ_{quad}	Orientation of the local frame in the first quadrant (rad)
θ'_r	Demand heading relative to a wall, correction board or imaginary line (rad)
θ_s	Demand heading at the start of a curve segment (rad)
$\theta_{x,y}$	Global orientation of the local frame on a curve (rad)
ξ	Damping factor

1. Introduction

The term 'autonomous guided vehicle' (AGV) 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. AGVs 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. In 1985, over 500 AGV systems were in use in Europe involving some 5000 vehicles (*Engineering,* 1985).

Although AGVs differ greatly in their behaviours and intelligence and in their structure and complexity, there are features common to all AGVs that are essential for autonomy. First, an AGV must have methods of sensing both its environment and its own actions. Sensory information gathered must be processed, integrated and assimilated to yield positional information within the AGV's 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. Second, the vehicle must have a set of algorithms within its controlling software that enables it to 'understand' its current perception of the working environment. Finally, this 'understanding' must be developed into some combination of low-level activities which ultimately actuate a behaviour (movement or manipulation) with a degree of intelligence.

The function of an AGV and the environment it operates in are inextricably linked to both the AGV's sensory requirements and the complexity of its 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. Here it is necessary to make a distinction 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 (Nilsson, 1969; Thompson, 1977; Elfes and Talukdar, 1983; Giralt *et al*, 1979). In this paper, the former type of guided vehicle is considered – i e, one operating a pre-defined and structured environment such as a warehouse or factory.

Typically, a warehouse automated with guided vehicles may consist of a central stock-taking and controlling computer, a number of AGVs, and some means of communicating between the two. Currently, most of these systems rely on wires buried under the warehouse or factory floor for their guidance. The AGV follows the wires by monitoring the difference in induced voltage between two coils on the vehicle and uses this information to generate steering instructions (*Materials Handling News,* 1985; *Materials Handling Engineering,* 1985). Although this system is simple, proven and reliable, a great deal of flexibility is lost in automation. Such AGVs are not 'free-ranging' since their choice of routing is dictated by the layout of the guide wires. Because of this, the AGV movements can not truly be optimised. Moreover, the layout of the warehouse must remain fixed and in proximity to the installed wires.

It is the purpose of this paper to examine how AGVs can be made more 'free ranging', particularly in the warehouse environment, and to examine some of the control problems which exist. In particular, the way in which low-cost low-level sensors can be used in complementary manner is discussed.

To examine the control behaviour of an AGV, an experimental vehicle has been constructed which uses, as its low-level sensors, ultrasound to generate distance information and shaft encoders to measure incremental wheel rotations. The details of this AGV are given in the following section. Further sections then describe how the information from the ultrasonic and odometry systems is complementary and can be used to provide the necessary positional information and how the AGV software is structured to cope with the many demands placed on it.

2. The research AGV

The research AGV is a three-wheeled vehicle with two rear wheels providing vehicle propulsion and steering, while a swivel castor gives the necessary support at the front. A plan and side elevation of the vehicle are shown in Figs 1a and 1b, respectively. Steering is implemented by controlling the speed differential between the two independent drive wheels. The drive wheels themselves are driven by stepping motors geared down through timing belts. Such a configuration 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 differential gear would be required on the back wheels.

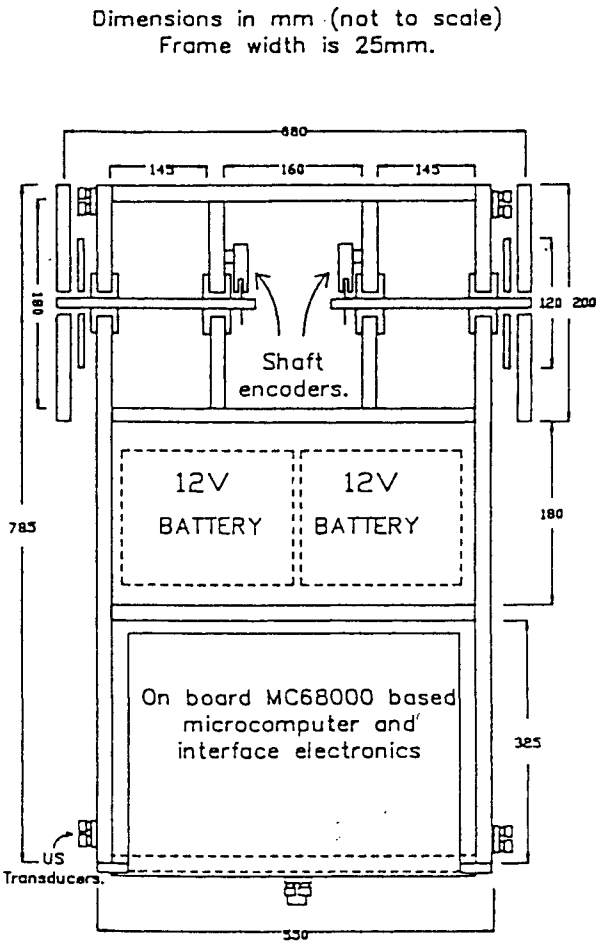

Fig 1a Plan of the research AGV

(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 Fig 1a shows how the sensors are mounted. Optical shaft encoders are used to generate odometric information and are mounted on each drive wheel to provide 100 pulses per revolution. 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 of the vehicle to sense obstacles.

The AGV is controlled by a Motorola MC68000 based computing system which runs at a clock frequency of 8 MHz and has 192k of memory and four interface boards associated with it. The first of these interface boards is a 64-bit digital input/output card which uses four Motorola 6522 VIA chips. Another is an eight-channel, 12-bit A–D converter. The third is a system timing/counting board which employs four AM9513 STC chips. Finally, there is a disc controller for the on board disc drive.

Both the stepper motors and the interface electronics are driven by two 12 V (36 Ah) car batteries. In order to reduce the rotational inertia of the AGV in the horizontal plane (yaw plane), these batteries are mounted close to the vehicle's centre of rotation (see Fig 1b).

3. The functions of the sensor systems in AGV guidance

3.1 Sensor functionality

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. This includes wire following ('continuous reference

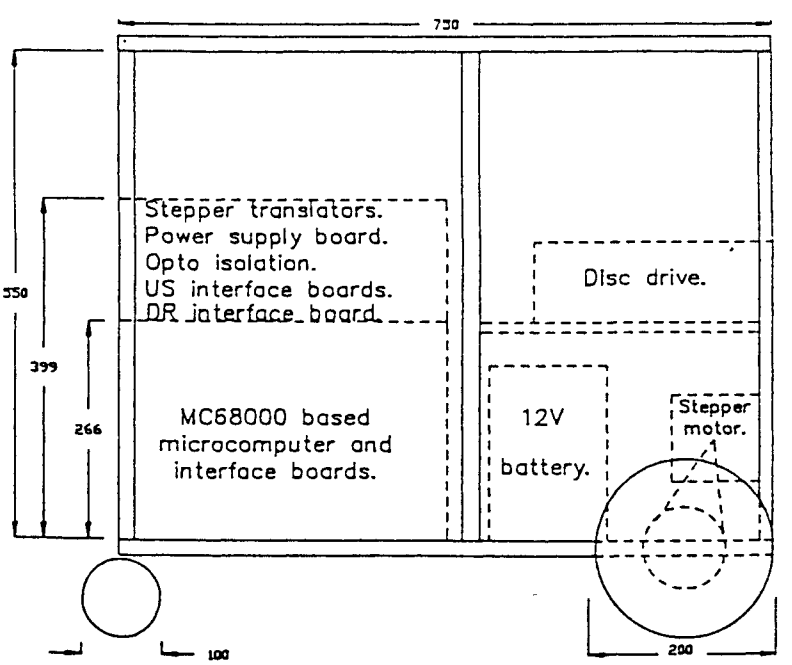

Fig 1b Side elevation of the research AGV

point'), triangulation from two reference points, or measuring both angle and distance from a single reference point.

or

(2) Internal sensing when the AGV measures its own movements from an initial known position. This includes inertial guidance systems and odometry by monitoring wheel movements.

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 aggrevate the problem (such as wear on wheels, initial measurement inaccuracies of the wheelbase and wheel diameters, and computational inaccuracies). In certain applications the employment of only external sensing may lead to inflexibility since the reference targets 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 described in this paper estimates global position by monitoring the incremental rotation of each drive wheel at regular sampling intervals. Appendix 1 describes how, geometrically, these incremental wheel rotations can be used to establish the incremental change in the AGV global position during the last sampling interval and hence the new updated global position. However, because 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 AGV's operating environment. These correction boards are simply wooden boards which are approximately 25 cm high and between 2 and 5 metres long. On these correction boards there are protrusions ('notches') 4 cm deep and 10 cm long at 1-metre intervals along its length.

Correction boards can be thought of as an artificiality added to the environment which is tailored to, and designed to enhance, a low-level sensor's performance. The boards themselves are smooth when observed using 40 kHz ultrasound and thus function as spectral reflectors. This ensures that the distance normal to the board is measured irrespective of the sensor's orientation. The wide beamwidth of the sensors, although a function of distance, generally allows corrections to be made at up to 20 degrees relative to the board. 'Notches' on a correction board are chosen to be protruding on account of the wide beamwidth of the ultrasound.

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

3.2 Position correction

As the AGV passes a correction board, the global position of which is known, the AGV measures 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 distance of the front and the rear of the vehicle from the board (y_1, y_2) and from these values a measurement of the local coordinates (y, θ') can be derived as:

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

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

(see Fig 2). Measurement of the third coordinate, x, in the local frame of the board (i e, an 'along-board' measurement) can only be generated when either sensor passes a notch and the resulting data are processed to yield a value for the distance travelled along the board. A global position correction can now be made from a transformation of the AGV's measured position in the local frame of the correction board (x, y, θ') to the global space (X, Y, θ) using the matrix equation:

$$\begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ \theta_0 \end{bmatrix} + \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & 0 \\ \sin\theta_0 & \cos\theta_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta' \end{bmatrix} \quad \ldots(3)$$

where (X_0, Y_0, θ_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).

Unfortunately, the ultrasonic measurement system is also subject to errors which must be accounted for in the global position correction. In the first instance, detailed observations of the performance of the ultrasonic system are used to provide the experimental AGV with an *a priori* characterisation of its sensing ability. This characterisation contains information concerning the dependency of both systematic and random errors in ultrasonic observations on the local position (perpendicular distance and orientation) of a sensor relative to a correction board. To a large extent, systematic errors can be eliminated by compensation. The standard deviation of random errors was found to be 0.1 cm or less at ranges up to 1 m when the sensor was normal to a correction board and less than 0.2 cm for angles up to 20 degrees relative to a correction board. This means that 95% of the ultrasonic observations lie within a band of ±4 mm relative to a mean value. Temperatures in the laboratory can be read to an accuracy of at least 1 deg C. An error of 1 deg C at a nominal temperature of 16°C produces an error of 1.4 mm over 1 m. This error, in combination with the random component of error, suggests that an ultrasonic measurement is accurate to within about ±0.54 cm of the true correction board range.

Although corrections in θ can be made at any time when neither sensor is opposite a notch, corrections in X and Y are not, in general, independent (see Eqn 3) and, therefore, can only be made after a notch has been detected and identified.

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 position calculated from uncorrected odometric data is known, so it is vitally important to maintain a continually updated estimate of the odometric position error bands. These error bands expand as

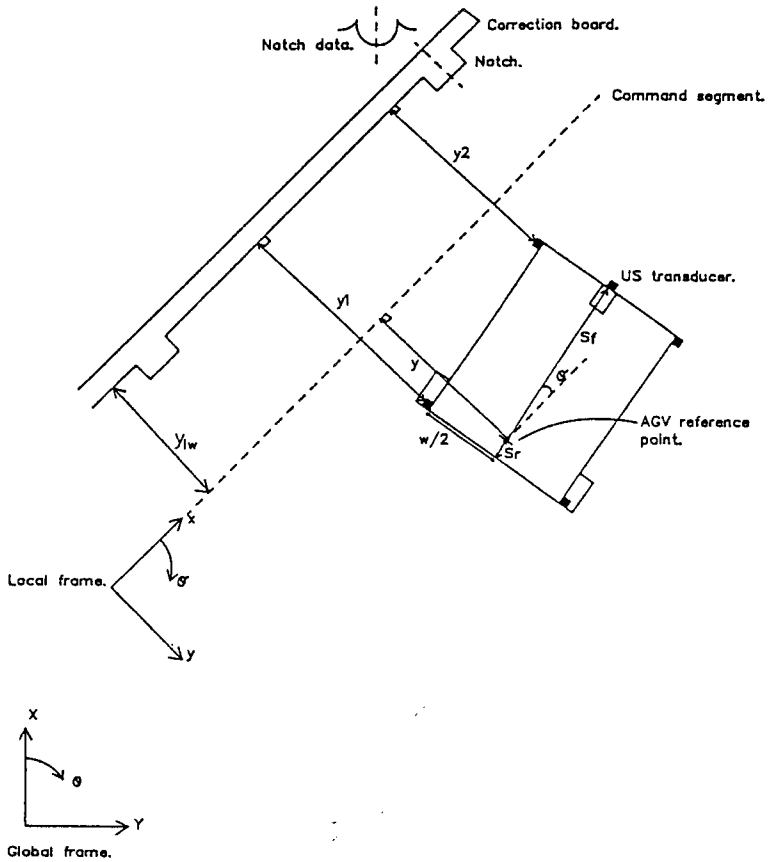

Fig 2 The position-correction method

the AGV travels through the environment in order to reflect the cumulative nature of position-estimation errors. If there is no estimate of positional error, the detection system may be enabled after the sensor has passed the notch and so correction could not take place. Conversely, if the detection system were enabled all of the time, corrections may be made on spurious data (for example, a person may walk between the vehicle and the correction board) which would cause the vehicle to lose position.

In maintaining an estimation of odometric error bands, spurious data which may be interpreted as a notch can be rejected since correction is only permitted when the position measured using the ultrasonic sensors is within the estimated three-dimensional (ΔX, ΔY, $\Delta\theta$) error band. Correction points should not be missed as they are looked for when the nearest edge of the expanded description of position contacts the programmed position of the notch.

It is now clear that the two systems are complementary. The consistency of odometry is used to reject spurious ultrasonic measurements, while the accuracy of (non-spurious) ultrasonic measurements is used to constrain the accumulation of errors in odometric position estimation.

In order to maintain an estimate of 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. More specifically, if the incremental distance travelled by the left and right drive wheels is d_l and d_r, respectively then the four combinations:

1. αd_l, αd_r
2. αd_l, βd_r
3. βd_l, αd_r
4. βd_l, βd_r

are fed through the odometric equations (25), (37), and (38) (Appendix 1) to determine, in each case, the changes in the global position coordinates (X, Y, θ) (α and β are constants where $\alpha \geqslant 1$ and $\beta \leqslant 1$). Since errors in the θ coordinate expand with distance travelled by the AGV and the build up of errors in the X and Y coordinates is dependent on this angle error, estimated errors in the X and Y coordinates are calculated using the bound of θ error that expands the X and Y error estimates by the greatest degree. For example, if it is assumed that the left wheel has actually travelled further than is measured, and the right wheel has travelled a shorter distance than is measured (case 2 above), then the upper bound of θ estimation would be used in the forward travelling direction, whilst the lower bound of theta would be used in the reverse direction. Fig 3 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.

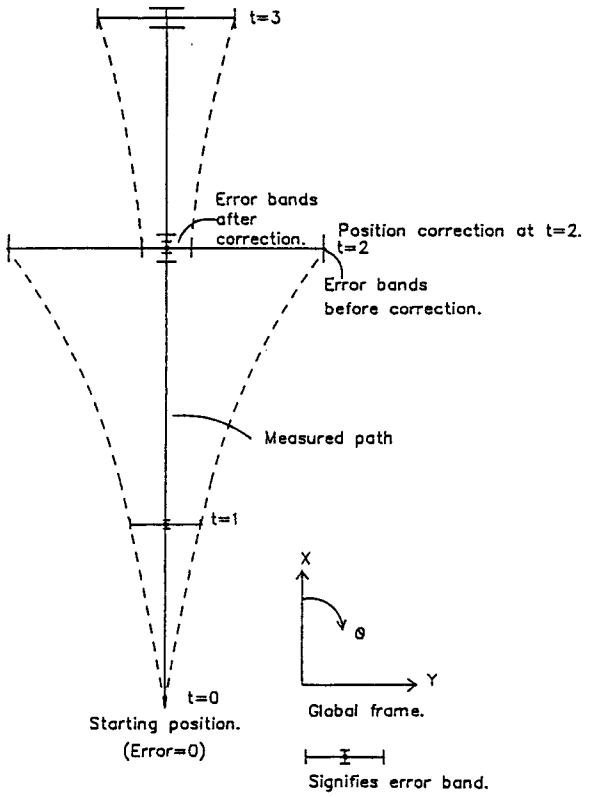

Fig 3 Expansion and contraction of odometric error bands

Since odometric errors are not directly quantifiable, the constants α and β 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 $\beta = 0.99$. When a successful position correction is made, the error bands contract by being set to the estimated error in the position correction (see Fig 3).

3.3 AGV operation

In normal operation, the position feedback for the AGV's control system is provided by odometry and, as explained in section 3.2, corrections are made by the ultrasonic sensing system when the vehicle is 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. In a warehouse environment, the AGV may use its ultrasonic transducers to steer down an aisle in which a low correction board forms the base of the racking system and the notches represent bay positions. Although odometry is not being used directly as a control feedback, position corrections are still repeatedly made to reduce the error in the measured position along the aisle. This information is necessary in order to initiate correctly a deceleration for a bay position or the exit of an aisle.

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 telemetry feedback directly);
(2) LF. Line follow command (using corrected odometry feedback);
(3) CF. Curve follow command (using odometry feedback directly); and
(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 attached to 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
```

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 program returns the distance y_{lw} (which in this case is 100 cm). 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 0.2 m/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. This has no parameters printed after its abbreviation.

4. Software

Although most of the AGV control software is written in 'C' and compiled to MC68000 object code, some routines are written directly in MC68000 source code, primarily to manipulate registers in the microcomputer's interface chips. In addition, when rapid operation of interrupt routines is required, the interrupt service routines are written directly in assembler to ensure that the code has optimum efficiency.

The method of results analysis requires the AGV's onboard 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.

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

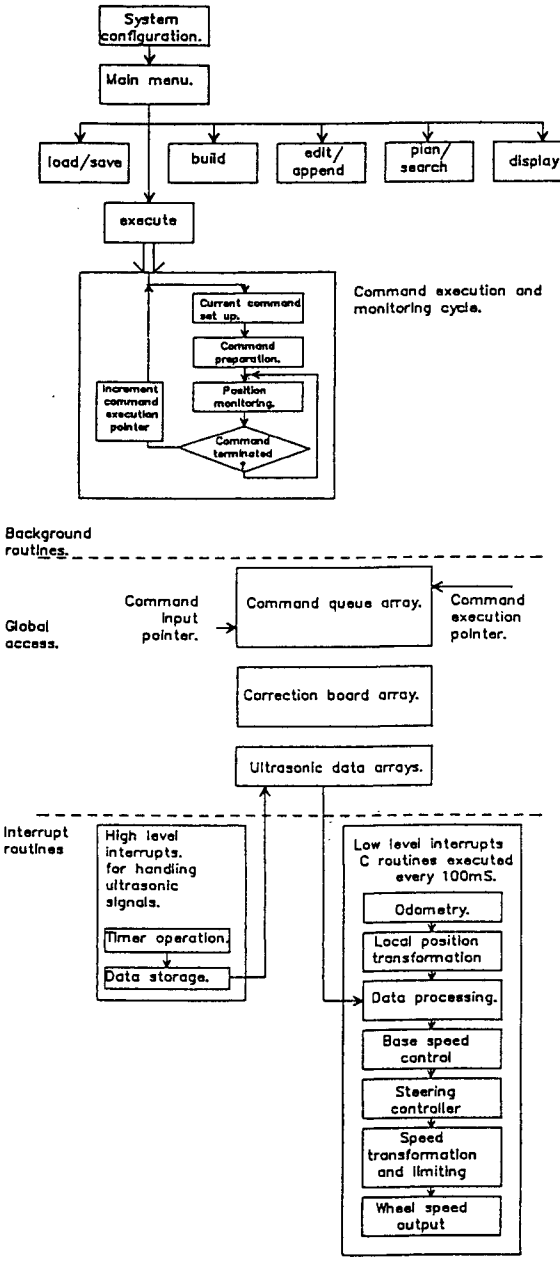

Fig 4 A software overview

The function of the background task is fourfold, namely:

(1) To initialise the AGV ready for a test run;
(2) To load, save, build, or edit a command queue or correction board array;
(3) To prepare for, initiate, and monitor the execution of commands in the queue; and
(4) To display collected results after a test run and/or to save them to a memory buffer ready for transfer to a floppy disc file.

Whereas the background routines provide a 'test harness' in which to initiate runs and analyse the performance of the AGV, both the control and sensing algorithms for the vehicle reside in interrupt software.

A timer on one of the microcomputer's interface chips generates a low-level interrupt (level 3) every 100 ms. 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 (i e, 100 ms) 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 low-level control interrupt service routine ('cisr') contains calls to a number of routines, namely:

1. 'odom' – Odometry. (Global position update);
2. 'localpos' – Local position transformation;
3. 'datproc' – Data processing;
4. 'ramp' – Base-speed (V_b) control;
5. 'controller' – Steering controller;
6. 'transform' – Speed transformation and curvature rate limiting; and
7. 'wspout' – Wheel speed output.

Each of these is shown as a block under the heading of 'low-level interrupts' in Fig 4. The first routine, 'odom', updates the odometric estimate of the vehicle's global position by reading encoder count values from each wheel. 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. Subsequently, 'datproc' processes ultrasonic distance measurements stored in globally accessible arrays 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) while '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. 'Transform' may also limit the amount by which the speed of a drive wheel may change and thus prevent 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.

External interrupts generated by the ultrasonic interface boards have the highest interrupt priority level (4) to guarantee accurate timing of an ultrasonic pulse. These interrupts occur roughly twice (one start and one stop interrupt) every 10 ms for each pair of enabled transducers.

The choice of a 10 ms 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 1.5 m. A pulsing interval of 10 ms permits a range of 1.685 m (V_s = 337 m/s) in which to detect a return echo unambiguously.

	0	1	2	3	4	5	6	7	8	9
Wall follow.	Type (1)	Direction	Demand speed.	Demand heading	Xterm	Yterm	y_w Distance of line from wall (or cb)	Transducer set enabled.	—	—
Line follow.	Type (2)	Direction	Demand speed.	Demand heading	Xterm	Yterm	y_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.	Xc	Yc
Stop.	Type (4)	Direction (of previous command)	Demand speed. (0)	Demand heading	Xterm	Yterm	—	—	—	—
(0)	← Common parameters. →									

Fig 5 The command-queue array

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

(1) A high value relative to the response times of the control system. Section 7 shows that a typical rise time of the system is 5 seconds. The control sampling interval (100 ms) is a fiftieth of this value.
(2) A value low enough to permit sufficient time for all the necessary data processing and control calculations required in the control interrupt to be processed. In addition, sufficient ultrasonic data points must have been collected in order to implement the line-fitting algorithm in the interrupt routine 'dataproc'.

Communication between the background task and the 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 are 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 the 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 (3) interrupt software.

The command queue exists in software as a global two-dimensional integer array in a format shown by Fig 5. It can be seen that the different commands occupy one dimension and the various parameters for these commands occupy the other. The first 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, angle turned through and start position of the curve (i e, 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 (e g, when they are all set at the start of a command execution), 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 (and their notches) in the AGV's workspace. If an option called 'board search' is taken in the main menu, the contents of the correction board array are matched against the contents of the command queue array in order to determine the commands in which the AGV can make position correction. If a command is sufficiently close and parallel to a correction board, the perpendicular separation (y_{lw}) and the linear overlap of that command and the board are calculated. Taking into account both the direction of travel of the AGV and the position of the AGV's sensors relative to its reference point, the local positions (x_0, x_1) in the command segment between which a (y, θ') measurement may be made are determined. This may result in the overlap segment being shrunk slightly. The values x_0, x_1, y_{lw} are loaded into the command queue array as shown in Fig 5. In order to determine which notches are observable, the command segment is expanded, according to the AGV direction and the position of the AGV's sensors relative to the midpoint between its drive wheels, to form a window. Those notches on the correction board that fall in this window are observable and their positions in the local frame of the command are loaded into a separate notch array.

As shown in Fig 4, the background task initialises the AGV for a test run. This involves configuring interface chips for the on-board microcomputer, loading or building look-up tables, and requesting certain global variables such as control variables and the mode of data collection. After the initial set-up procedure, the AGV

software allows the operator to either manipulate or execute the command queue.

A command queue can be loaded from the on-board disc drive, or a queue can be built up by entering individual commands and their associated parameters. On entry of an individual command, a simple planning option is available which allows calculation of command parameters on the basis of measured aisle widths (see Appendix 2). After a command has been entered or changed, it is validated. This seeks to trap errors by checking for incongruities between the currently entered and previously entered commands in the queue. An editing routine is available which prompts the operator for a position in the queue and then allows editing of that particular command. A coordinating menu which permits calls to any of these routines permits routes to be loaded, appended and altered with ease.

When a displayed command queue represents the desired route, the 'execute' option can be taken in the main menu. Interrupts are now enabled, a 'command pointer' is set to point to the first command in the queue, and thus a speed ramp is initiated to accelerate the AGV to its demand speed. For each command, a call to a 'command preparation' routine is made. The function of this is to ensure that there is a smooth transition between the current and the next command. For example, if there is a reduction in demand velocity between the current command and the next command, the routine will initiate a ramp 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 the current command, the speed change is made at the next command (i e, when the command pointer is incremented). This is a sensible approach because lower speeds are likely to have been calculated for more critical control or tighter radii.

5. Formulation of a control strategy

The function of the AGV's controller is to derive a turning radius $(r)\left(\text{or curvature } \kappa = \frac{1}{r}\right)$ along which the vehicle must travel, by measurement of the AGV's local 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 Fig 6. At each sampling interval, the incremental rotation of the two drive wheels is used in the odometric equations of Appendix 1 to generate an update of the AGV's global position, (X, Y, θ). If the AGV is following an imaginary line, the AGV's position relative to that line is computed

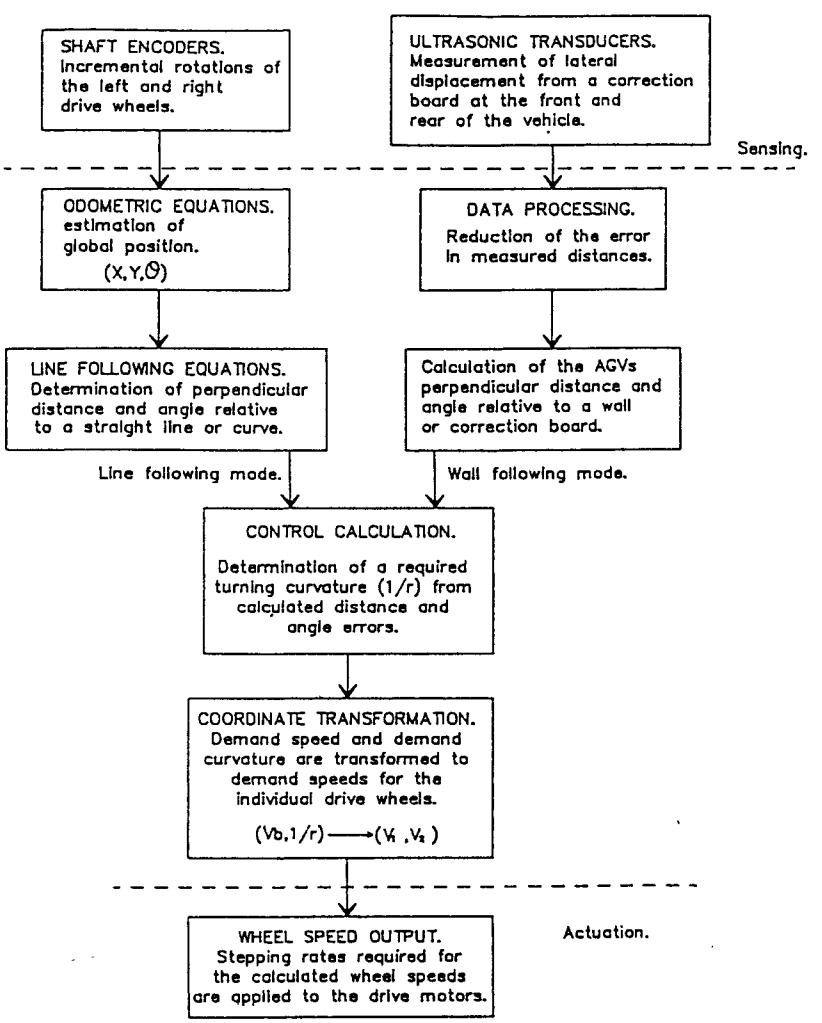

Fig 6 Schematic of the AGV control

using the updated global position information. Alternatively, if the AGV is following a wall or correction board using its ultrasonic transducers, this relative positional information is available directly. 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 two driven wheels as described in Appendix 3.

The control itself 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 heading relative to the imaginary line. This always points towards the imaginary line on a slope 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 Fig 7.

This control scheme 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 Fig 2) are generated in a control interrupt from a processed batch of ten raw ultrasonic readings. The local position (y, θ') required for either direct control or to correct odometric position estimation is derived from these measurements using Eqns (1) and (2) (section 3). 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, the equations described in Appendix 4 are employed.

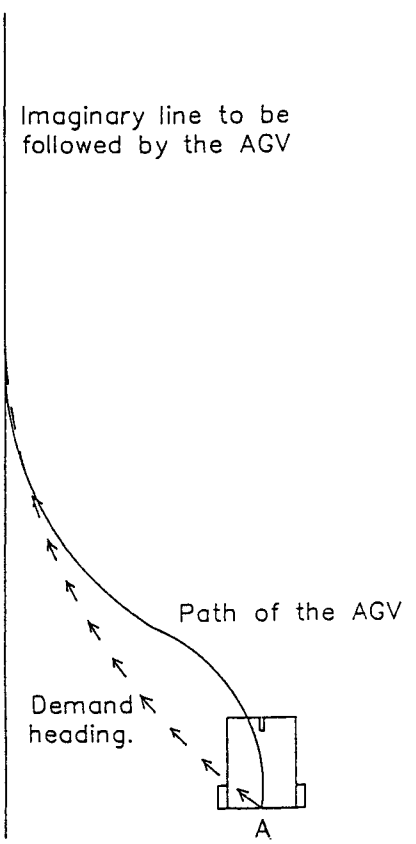

Fig 7 The control method

6. Performance of the control system

In order to assess the performance of the control system, a knowledge of steady-state performance 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 the line it is following, a knowledge of the response to step output disturbances is also required.

The behaviour of a stepper motor is such that either it adopts a speed in correspondence with the pulse rate sent to its translator or it stalls owing to excessive torques developed from acceleration and loading. The implication of this is that, in the absence of stalling, the mass, rotational inertia and other vehicle dynamics are eliminated from the equations of motion and a simple geometrical model of the line-following process may be considered.

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. To test this, a proportional control loop was implemented in which the steered angle was turned in proportion to this error. It was found that this method of control was only just adequate, and stable control could only be achieved with low speeds and small initial distance errors.

In section 5, a control system was described that could be used for both direct wall following using two ultrasonic sensors or imaginary line following using estimated position. This employs both distance and angle measurements. The basic structure of the control system is shown in Fig 8. Experience has shown that simple proportional controllers function adequately such that:

$$\theta'_r = -k_{pd}(y - y_{lw}) \quad \ldots(4)$$

$$\kappa = k_p(\theta'_r - \theta') \quad \ldots(5)$$

(see Fig 8). Note that κ in Eqn (5) is the required turning curvature relative to that specified in the command segment (the 'CF' command in section 3 has a curvature of $1.0\ m^{-1}$) and is therefore added to the curvature of the command segment (κ_{seg}) before being applied to the drive wheels. For the special case of following a straight line, the command-segment curvature is zero and so, in this instance, κ represents the actual AGV turning curvature.

A modification of the geometrical model employed by Steer (1985) can now be employed to derive an approximate closed-loop transfer function of the control represented by Eqns (4) and (5). This can then be tested for different values of k_p and k_{pd} and compared to the predicted results.

Consider a vehicle moving toward its imaginary line on a given curvature developed by the controller and at

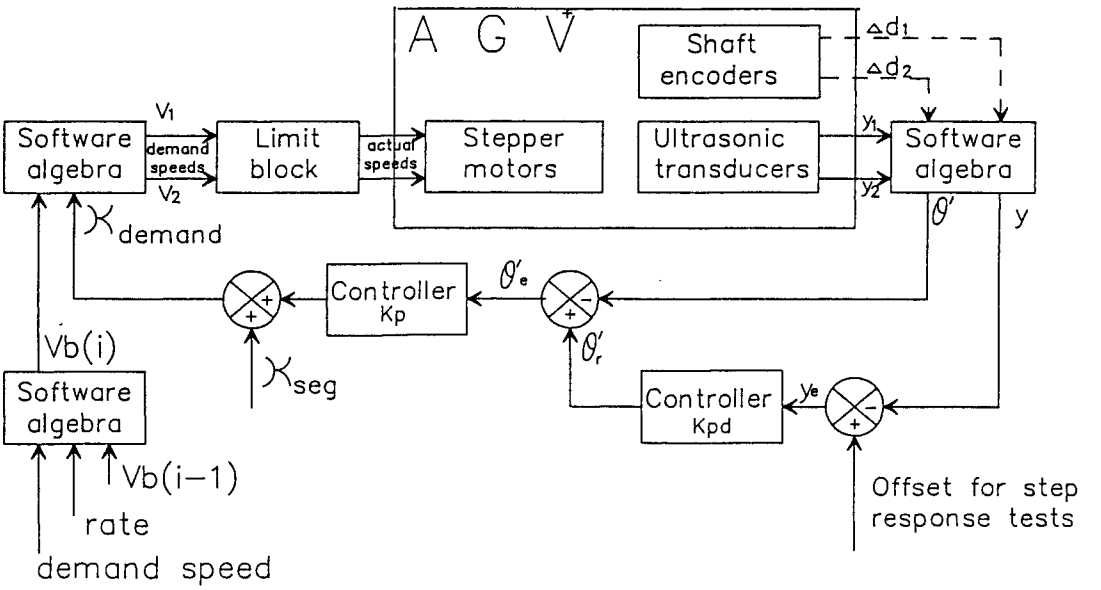

Fig 8 Block diagram of the control system

a constant velocity (V) as in Fig 9. The standard result for curvature in an x-y plane is given by Yakolev (1965)

$$\frac{d\theta'}{ds} = \kappa = \frac{\frac{d^2y}{dx^2}}{\left[1+\frac{dy^2}{dx}\right]^{3/2}} \quad \ldots(6)$$

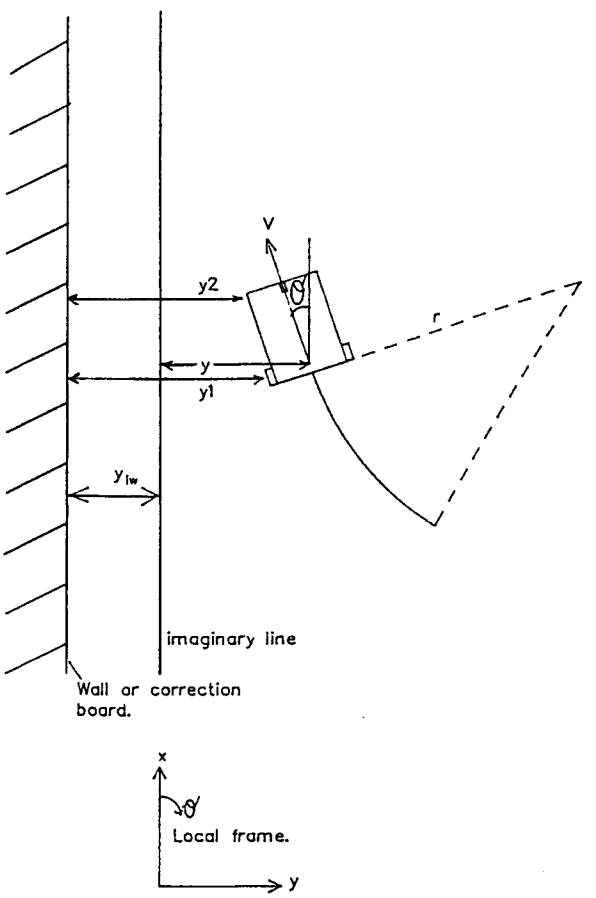

Fig 9 Wall and imaginary-line following

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

$$\frac{dy}{dx} = \tan\theta' \quad \ldots(7)$$

In direct wall following, a measurement of sin θ' is made using distance measurements from the front and rear sensors such that:

$$\sin\theta' = \frac{y_2 - y_1}{l} \quad \ldots(8)$$

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

$$\tan\theta' \approx \sin\theta' \approx \theta' \quad \ldots(9)$$

In an experiment that is described in the next section, the vehicle was subjected to a step disturbance in y of 20 cm and the largest angle (θ') attained by the AGV was 9 degrees. In this case the maximum error incurred by the above approximation is less than 1%.

Using the approximation described in Eqn (9) in Eqn (7):

$$\theta' = \frac{dy}{dx} \quad \ldots(10)$$

Since for small angle approximations $\frac{dy}{dx}$ is considerably less than one, its higher powers are even smaller, and Eqn (6), describing curvature, reduces to:

$$\kappa = \frac{d^2y}{dx^2} \quad \ldots(11)$$

Combining controller Eqns (4) and (5) gives:

$$\kappa = -k_p k_{pd}(y - y_{lw}) - k_p\theta' \quad \ldots(12)$$

and substituting the approximate differentials for curvature and angle gives

$$\frac{d^2y}{dx^2} + k_p\frac{dy}{dx} + k_p k_{pd} y = k_p k_{pd} y_{lw} \quad \ldots(13)$$

Now

$$\frac{dy}{dx} = \frac{dy}{dt}\frac{dt}{dx} \quad \ldots(14)$$

and

$$\frac{d^2y}{dx^2} = \frac{d^2t}{dx^2}\frac{dy}{dt} + \left(\frac{dt}{dx}\right)^2\frac{d^2y}{dt^2} \quad \ldots(15)$$

As the angle θ' is assumed to be small and absolute velocity (V) is constant, then velocity in the x direction is constant and Eqn (13) can be converted to a differential equation in the time domain and Laplace transformed to give

$$s^2y + k_pV_xsy + k_pk_{pd}V_x^2y = k_{pd}V_x^2y_{lw} \quad \ldots(16)$$

Note that if the AGV is travelling backwards, velocity is negative and so both k_p and k_{pd} are made negative to keep coefficients positive and the system stable.

The roots of the characteristic equation derived from Eqn (16) are given by:

$$s_{1,2} = \frac{-k_pV_x}{2} \pm \frac{V_x}{2}\sqrt{k_p^2 - 4k_pk_{pd}} \quad \ldots(17)$$

To obtain the fastest response without inducing overshoots, these roots should be chosen in the correct ratio to give critical damping. This requires that:

$$k_p = 4k_{pd} \quad \ldots(18)$$

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 (derived from Eqn (16)) is written in the normalised form:

$$\frac{s^2}{w^2} + Ts + 1 = 0 \quad \ldots(19)$$

then the rise time, T, is:

$$T = \frac{1}{k_{pd}V_x} \quad \ldots(20)$$

and the damping factor, ξ, is given by:

$$\xi = \frac{T\omega}{2} = \frac{\sqrt{k_pk_{pd}}}{2k_{pd}} \quad \ldots(21)$$

7. Results

To test the accuracy of the theory outlined in section 6 the AGV is asked to execute the command queue:

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

LF,F,0.2,0.0,6.0,0.0,20

ST

The first command allows the AGV 0.5 m to reach a speed of 0.2 m/s and the following command has a 20 cm offset from the imaginary line which acts as a step disturbance to the control system. The experiments were conducted using uncorrected odometric measurements, derived from internal shaft encoders, as a feedback. (If position corrections had been permitted, it would have been difficult to assess the performance of the control system as corrections would have acted as additional step disturbances.) When the AGV was allowed to travel 3 m along the global X axis, typical errors in uncorrected odometric position were found to be 1 cm in X, 6 cm in Y, and 5 degrees in θ. Position measurement using ultrasonic telemetry was found to be accurate to 1 cm in y and 1 degree in θ' for perpendicular distances less than 1 m and angles less than 20 degrees relative to a correction board.

The first graph, Fig 10a, shows how the distance error, which is the perpendicular distance between the midpoint of the AGV's drive wheels and the line, varies with time when the control constants are chosen to give critical damping. ($k_p = 4, k_{pd} = 1$.) From this graph it can be seen that the distance error is reduced to under 5% of the initial error (i e, within 1 cm) between 11 and 12 seconds after the step change was introduced. With $k_p = 4$, $k_{pd} = 1$ and $V_x = 0.2$ m/s, and using Eqn (17), the vehicle is expected to attain this accuracy in 11.9 seconds. (It was noted that the AGV reached angles of up to 9 degrees relative to the line which partly accounts for the discrepancy between measured and predicted results.)

Fig 10b shows the result when constants are chosen to give underdamping. ($k_p = 1$, $k_{pd} = 3$.) The damping factor calculated from Eqn (21) is 0.289 which corresponds to an overshoot of 39%. This is in close agreement with the measured value of 40%.

The relation between the control constants can be explained qualitatively. If k_{pd} is large, the vehicle's approach to the new line (i e, to the 20 cm offset) is steep and only at very small distances will the demand angle decay to zero. Since k_p is small, only relatively small 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 the line, 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 $\frac{1}{k_{pd}V_x}$ in Eqn (20). 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!

If the control constants 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. Normally the control constants k_p and k_{pd} are set to 4 and 1, respectively, which, for a base velocity of 0.2 m/s, gives a reasonable rise time (5 s) but does not cause stalling for step inputs of 20 cm and less.

In the more general case, the control constant k_{pd} is chosen to be in inverse proportion to base velocity such that:

$$k_{pd} = 0.2/V_b \quad \ldots(22)$$

with the added constraint that if V_b falls below 0.1 m/s, k_{pd} is held at 2.

If k_p is always chosen to be four times this value of k_{pd}, then both rise time and damping factor are held constant. Thus the control is velocity dependent which causes the vehicle to approach its path more steeply when travelling slower. This is a sensible approach, since smaller angular accelerations are developed at a lower

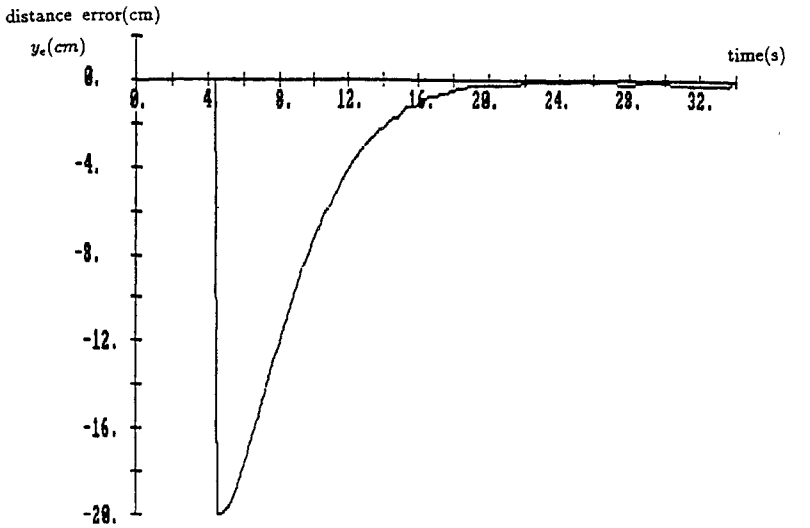

Fig 10a Graph of distance error y_e (cm) against time; critically damped response $k_p = 4$; $k_{pd} = 1$; $V = 0.2$ m/s

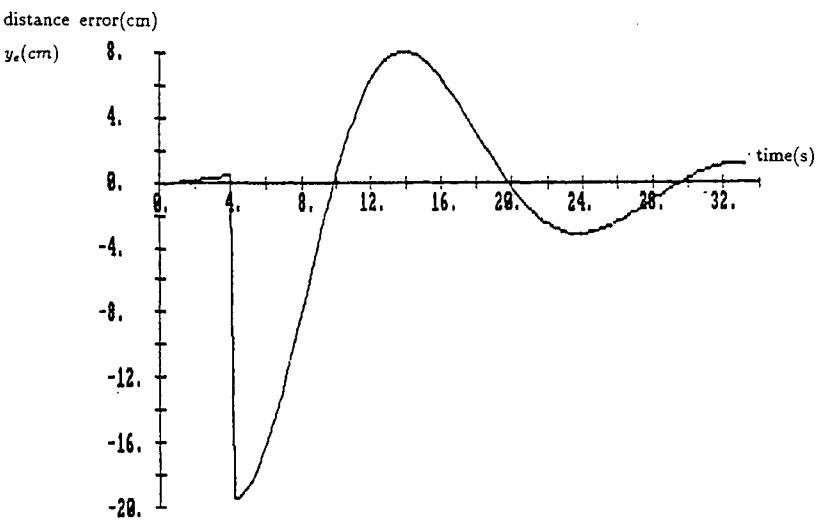

Fig 10b Graph of distance error y_e (cm) against time; underdamped response $k_p = 1$; $k_{pd} = 3$; $V = 0.2$ m/s

base velocity and, in addition, there is more torque available from the stepper-motor drives.

Using Laplace's final value theorem and the characteristic equation derived from Eqn (16), the steady-state error of the control system is theoretically zero. However, steady-state errors do exist, mainly on account of quantisation effects due to the limited resolution of the shaft encoders. For example, 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 rad). The implication of this is that, if k_{pd} is set to 1, distance errors within the range of 8.7 mm 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 ±8.7 mm band, accuracy cannot be improved.

Step disturbances larger than 20 cm do cause a deviation from the predicted behaviour since the limit block shown in Fig 8 limits the amount by which curvature can change at each control interrupt. This limiting of the curvature rate introduces a non-linear block into the control system which was not included in the original analysis. In order to test the effect of the limit block, the curvature was only allowed to change by a small amount ($0.009\ m^{-1}$) over a control interval when a step change of 20 cm was applied to the system. The result obtained in Fig 11, with the previously defined values of k_p and k_{pd} for critical damping, now shows an underdamped response because the limited curvature does not allow the vehicle to reach the demand headings set by k_{pd}.

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 exhibits critically damped behaviour for step disturbances up to 20 cm, and underdamped behaviour for larger step disturbances.

8. Conclusions

The AGV software has proven to be flexible in that it allows vehicle routes to be created, loaded, and edited with ease and allows correction boards to be moved around, or added, to the AGV's workspace. A control scheme has been devised which can be used for either wall following using ultrasonics or line following using odometry. For step inputs of up to 20 cm, the AGV performs well and the results are close to what is predicted. The good performance is accountable to the responsiveness of the steeper motors which effectively allow in-

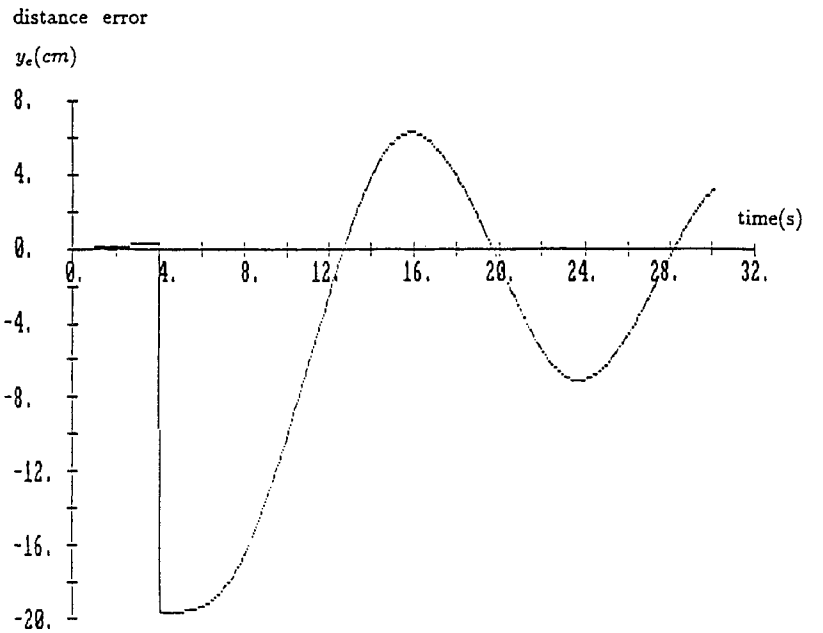

Fig 11 Graph of distance error y_e (cm) against time; curvature limited to 0. 9 m^{-1} s^{-1}

stantaneous changes of curvature. In addition, the vehicle's maximum heading for a step input of 20 cm is 9 degrees and so errors in the small angle approximations are small giving good agreement with the predicted results.

9. Acknowledgment

The authors gratefully acknowledge the financial support received from ICI PLC, in support of this work. The authors would also like to acknowledge the helpful advice and guidance received from Mr M. Gladders of ICI PLC.

10. References

Elfes, A. and Talukdar, S. N. 1983. 'A distributed control system for the CMU rover', *Proc IJCAI.*

Engineering. 1985. 'Designed to deliver', May.

Giralt, G., Sobek, R. and Chatila, R. 1979. 'A multi-level planning and navigation system for a mobile robot; A first approach to HILARE', *6th IJCAI,* Tokyo, Japan.

Materials Handling Engineering. 1985. 'Computer manufacturer meets automation boom with material handling', July.

Materials Handling News. 1985. 'Rugged AGV totes three-tonne load', February.

Nilsson, N. J. 1969. 'A mobile automaton: An application of artificial intelligence techniques', *Proc 1st IJCAI.*

Steer, B. 1985. *Navigation for the guidance of a mobile robot,* PhD thesis, 75–98.

Thompson, A. M. 1977. 'The navigation system of the JPL robot', *JPL Pub* 77–20.

Yakolev, K. P. 1965. *Handbook for Engineers,* **1,** 189–191.

Appendices

1. 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 *cntl* be the number of counts that the encoder on the left wheel has provided in a short time interval Δt and let *cntr* be the number of counts provided by the encoder on the right wheel in the same time interval.

Now let

$$\Delta cnt = cntl - cntr$$

Then the extra distance travelled by one wheel is

$$\Delta d = k_{dr}\Delta cnt \quad \ldots(23)$$

where

$$k_{dr} = \frac{\pi d_{wheel}}{100} \quad \ldots(24)$$

The change in angle of the AGV is

$$\Delta\theta = \frac{k_{dr}\Delta cnt}{W_b} \quad \ldots(25)$$

From the change in AGV heading, the radius of turn can be calculated as

$$r = \frac{\bar{d}}{\Delta\theta} \quad \ldots(26)$$

Where

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

Thus

$$r = \frac{k_{dr}(cntl+cntr)}{2\Delta\theta} \quad \ldots(27)$$

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

$$r = \frac{W_b(cntl+cntr)}{2(cntl-cntr)} \quad \ldots(28)$$

Eqn (25) 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 Fig 12 is made.

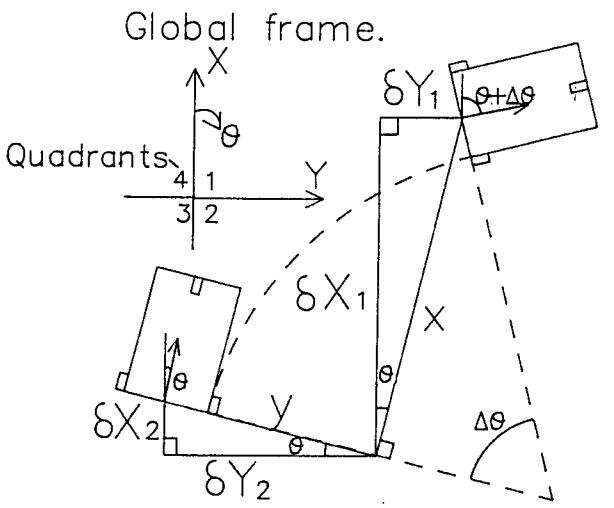

Fig 12 Odometric geometry

First, the orthogonal components in the local frame of the AGV's initial heading are calculated as shown in Fig 12 by:

$$x = r\sin\Delta\theta \quad \ldots(29)$$

$$y = r-r\cos\Delta\theta \quad \ldots(30)$$

These two quantities are then resolved into their respective X and Y components giving four quantities in all. On the Fig 12 these are:

$$\delta X_1 = r\sin\Delta\theta\cos\theta \quad \ldots(31)$$

$$\delta Y_1 = r\sin\Delta\theta\sin\theta \quad \ldots(32)$$

$$\delta X_2 = r(1-\cos\Delta\theta)\sin\theta \quad \ldots(33)$$

$$\delta Y_2 = r(1-\cos\Delta\theta)\cos\theta \quad \ldots(34)$$

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

$$|\Delta X| = r\sin\Delta\theta\cos\theta-r(1-\cos\Delta\theta)\sin\theta \quad \ldots(35)$$

$$|\Delta Y| = r\sin\Delta\theta\sin\theta+r(1-\cos\Delta\theta)\cos\theta \quad \ldots(36)$$

Note that if a left turn had been executed in the control 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 θ 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; *'sgnx'* and *'sgny'*. The first of these has the value +1 in the first and fourth quadrants and −1 in the second and third. *'Signy'* is +1 in the first and second quadrants and −1 in the third and fourth. This directional information is included in Eqns (35) and (36) to give:

$$\Delta X = (dirn.sgnx)[\delta X_1-(dirn.sgnx.sgny.)\delta X_2] \quad \ldots(37)$$

$$\Delta Y = (dirn.sgny)[\delta Y_1+(dirn.sgnx.sgny.)\delta Y_2] \quad \ldots(38)$$

where *'dirn'* is a variable defined as +1 in the forward direction and −1 in the reverse direction.

2. 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-angled corners, T-junctions, and crossroads. As explained in section 4, 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 Fig 13, 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, while the position of the second critical position is less obvious. Assuming that the AGV has turned through an angle θ and the starting position in a global coordinate frame was (0,0) then:

$$p_f = (X+s_f\cos\theta),\ (Y+s_f\sin\theta) \quad \ldots(39)$$

$$p_{fl} = \left(X+s_f\cos\theta+\frac{w}{2}\sin\theta\right),\left(Y+s_f\sin\theta-\frac{w}{2}\cos\theta\right) \quad \ldots(40)$$

Substituting

$$X = r\sin\theta \quad \ldots(41)$$

and

$$Y = r(1-\cos\theta) \quad \ldots(42)$$

gives the 'X' coordinate (X_{fl}) of the AGV point p_{fl} as

$$X_{fl} = \left(r+\frac{w}{2}\right)\sin\theta+sf\cos\theta \quad \ldots(43)$$

By differentiating this quantity, the angle in the curve at which the second critical position occurs can be found.

$$\frac{dX_{fl}}{d\theta} = \left(r+\frac{w}{2}\right)\cos\theta-s_f\sin\theta \quad \ldots(44)$$

Equating this to 0 yields

$$\tan\theta = \frac{2r+w}{2s_f} \quad \ldots(45)$$

By adding quantities in the aisle of width (a) in Fig 13:

$$a = \left(1+\frac{1}{\sqrt{2}}\right)c+\frac{1}{\sqrt{2}}\left(\frac{w}{2}-r\right)+X_{fl_{max}} \quad \ldots(46)$$

where $X_{fl_{max}}$ is the value of X_{fl} given by Eqn (43) at an angle of θ defined by Eqn (45).

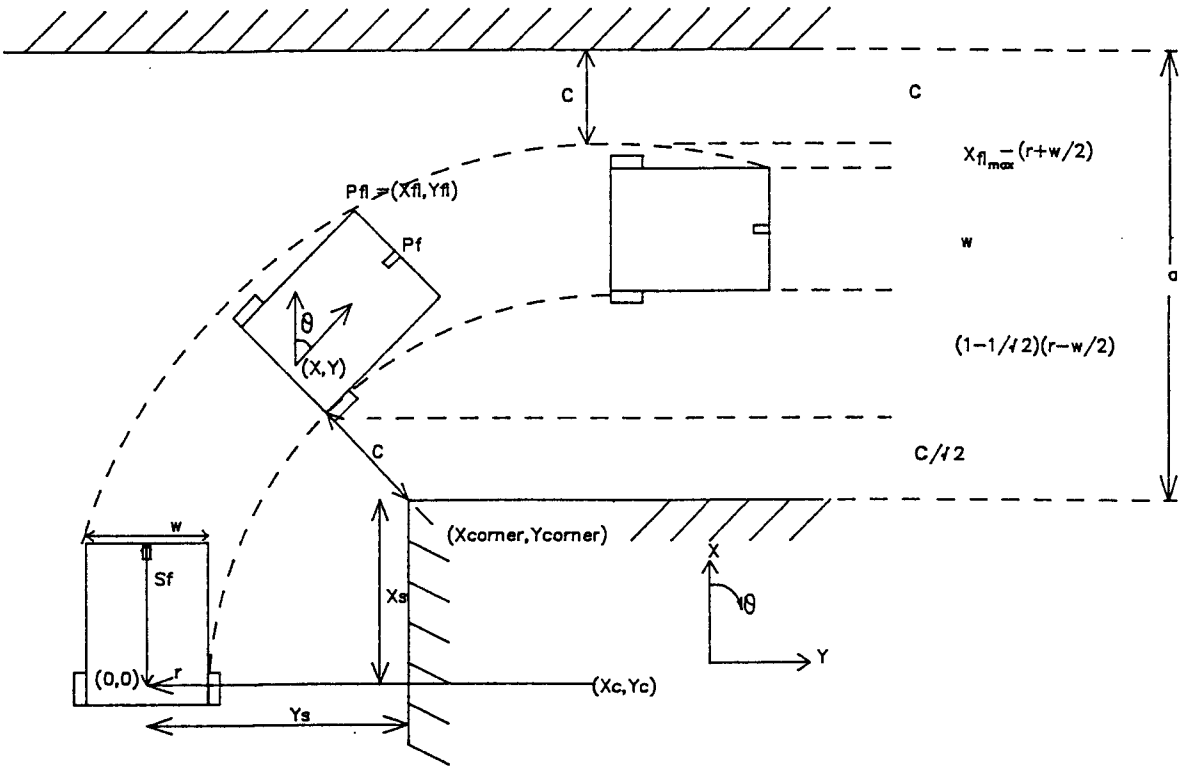

Fig 13 Calculation of command parameters for a corner

Rearranging Eqn (46) and substituting for $X_{fl_{max}}$ gives the function in r:

$$f(r) = \left(1+\frac{1}{\sqrt{2}}\right)c+\frac{1}{\sqrt{2}}\left(\frac{w}{2}-r\right)+\left(r+\frac{w}{2}\right)\sin\theta + s_f\cos\theta - a = 0 \quad \ldots(47)$$

which has the differential

$$f'(r) = \left[\frac{\psi}{1+\psi^2}\right]\cos\theta+\left[\frac{\psi^2}{1+\psi^2}\right]\sin\theta-\frac{1}{\sqrt{2}} \quad \ldots(48)$$

where

$$\psi = \frac{2r+w}{2s_f} \quad \ldots(49)$$

and

$$\theta = \arctan\psi \quad \ldots(50)$$

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 Fig 3. The distance before the corner at which the AGV should start turning is

$$X_s = \frac{1}{\sqrt{2}}\left(r-c-\frac{w}{2}\right) \quad \ldots(51)$$

The distance from the right hand wall to the centre of the AGV is

$$Y_s = \left(1-\frac{1}{\sqrt{2}}\right)\left(r-\frac{w}{2}\right)+\frac{c}{\sqrt{2}} \quad \ldots(52)$$

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.

3. Independent wheel speed control: base speed control and steering

As described in section 2 the MC68000 microcomputer has a number of VIA chips (Motorola 6522) which are used to supply the pulses to the stepper translators. These have internal timers which can toggle the logic state of a pin on the output port on clock-count expiry. If appropriately configured, when the logic state is toggled, the counter will reload the latched value and recommence countdown. 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 – i e, the latched counter value must be decremented through to zero twice. Thus the time (t) for one pulse to be supplied to the stepper motor is given by:

$$t = \frac{2n}{f_c} \quad \ldots(53)$$

where

f_c = clock frequency (0.8 MHz) and
n = latched counter value.

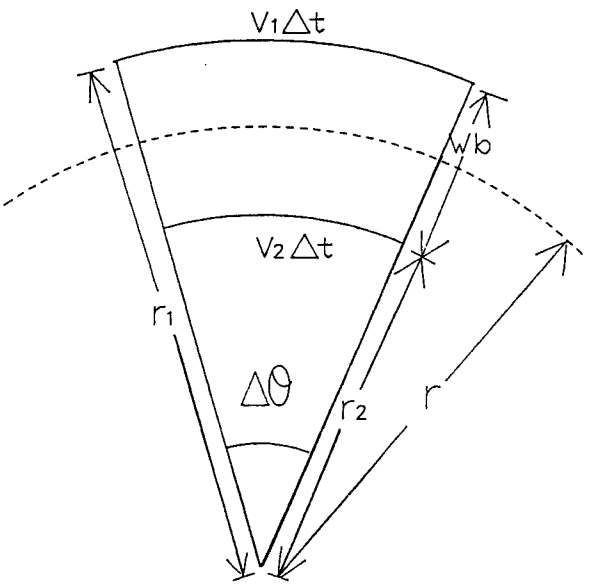

Fig 14 The relationship between wheel speeds and turning radius

This gives the relation between the latched count value and the stepper speed (V) in steps per second (sps) as:

$$V = \frac{400000}{n} \quad \ldots(54)$$

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_b,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. Calculations based around Fig 14 yield relationships between the turning radius and the individual wheel speeds such that:

$$\Delta\theta r_1 = V_1\Delta t \quad \ldots(55)$$

$$\Delta\theta r_2 = V_2\Delta t \quad \ldots(56)$$

$$r_1 - r_2 = W_b \quad \ldots(57)$$

where

$\Delta\theta$ = angle turned in time Δt.

Substituting for r_1 in Eqn (55):

$$\Delta\theta = \frac{V_1\Delta t}{r_2+W_b} \quad \ldots(58)$$

Subtracting Eqn (50) from Eqn (55) and substituting Eqn (57) and Eqn (58) gives:

$$r_2 = \frac{V_2 W_b}{V_1 - V_2} \quad \ldots(59)$$

Now

$$r = r_2 + \frac{W_b}{2} \quad \ldots(60)$$

Hence

$$r = \frac{W_b(V_1+V_2)}{2(V_1-V_2)} \quad \ldots(61)$$

Eqn (61) 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 one wheel; and if $V_2 = -V_1$ then $r = 0$ and the AGV spins on the spot.
It is clear that

$$V_1 + V_2 = 2V_b \quad \ldots(62)$$

where

V_b = base velocity

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

$$r = \frac{W_b V_b}{V_{rel}} \quad \ldots(63)$$

In order to implement steering control along some constant 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 given by

$$\frac{V_{rel}}{2} = \frac{W_b V_b}{2r} \quad \ldots(64)$$

4. Position-monitoring calculations

4.1 Line-following calculations

Consider a line being followed which starts at (X_{start}, Y_{start}), ends at (X_{term}, Y_{term}) and has a heading θ with the AGV being located at (X_{agv}, Y_{agv}) as shown in Fig 15a. 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} calculated as:

$$Y_{line} = \frac{(X_{agv}-X_{start})}{(X_{term}-X_{start})}(Y_{term}-Y_{start}) + Y_{start} \quad \ldots(65)$$

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

$$y = (Y_{agv}-Y_{line})\cos\theta \quad \ldots(66)$$

where θ is the demand heading of the command.

In Fig 15a, 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.

$$L = \sqrt{(X_{term}-X_{agv})^2+(Y_{term}-Y_{agv})^2-y^2} \quad \ldots(67)$$

Since the routines employ squared values to calculate the distance left to the end of a segment, a magnitude is actually calculated. Thus there is a potential problem

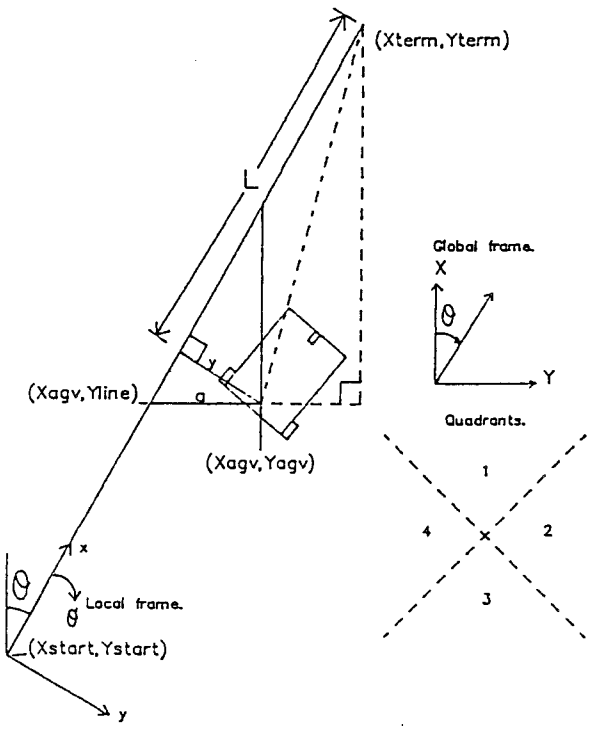

Fig 15a Transformation to local position – line following

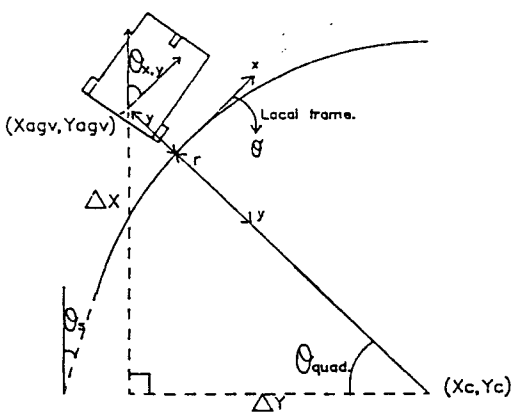

Fig 15b Transformation to local position – curve following

when the end of a segment is being monitored, because the remaining distance can never be less than zero. This particular case can be recognised by checking that the remaining distance in a segment is always decreasing.

The equations above apply to the first and third quadrants of Fig 15a. 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 on to another parallel horizontal line.

In the second and fourth quadrants the equations are:

$$X_{line} = \frac{(Y_{agv}-Y_{start})}{(Y_{term}-Y_{start})}(X_{term}-X_{start})+X_{start} \quad \ldots(68)$$

$$y = (X_{line}-X_{agv})\sin\theta \quad \ldots(69)$$

Note that Eqns (69) and (66) for the perpendicular distance y differ slightly to keep a sign convention of a positive distance being to the right of the imaginary line.

4.2 Curve-following calculations

Consider a curve that is being followed which has radius r, centre (X_c, Y_c), initial heading θ_s, and arc length $\Delta\theta$. The current position of the AGV is (X_{agv}, Y_{agv}) as shown in Fig 15b. Radius, arc length, and centre are parameters obtained by accessing the current command of the command queue array. Initial heading is simply the heading of the previous command in the queue. The centre coordinates are generated at the command entry stage with the equations:

$$X_c = X_{start}-r\sin\theta_s \quad \ldots(70)$$

$$Y_c = Y_{start}-r\cos\theta_s \quad \ldots(71)$$

where X_{start}, Y_{start} 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 the calculation of AGV position relative to the centre of the curve in the command queue regardless of what the heading of the AGV actually is. This calculation requires an equivalent heading in the first quadrant to be generated as:

$$\theta_{quad} = \text{atan}\frac{|X_{agv}-X_c|}{|Y_{agv}-Y_c|} \quad \ldots(72)$$

This angle is adjusted to the correct quadrant in accordance with the signs of $X_{agv}-X_c$ and $Y_{agv}-Y_c$. Thus the orientation of the local coordinate frame shown in Fig 15b $(\theta_{x,y})$ 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), π is added to the calculated frame orientation which is then readjusted (if necessary) to be in the range 0 to 2π.

A curve-follow command is deemed to be terminated when the condition

$$|\theta_{x,y}|-\theta_s \geq \Delta\theta \quad \ldots(73)$$

is detected. The remaining distance in the curve needs to be calculated if a reduction of speed is required before the termination of the command. This is simply the modulus of the current demand radius multiplied by the remaining angle in the segment.

The perpendicular distance from the curve is calculated as:

$$y = r-sgn(r)\sqrt{(X_{agv}-X_c)^2+(Y_{agv}-Y_c)^2} \quad \ldots(74)$$

Recalling that a positive radius denotes turning to the right, Eqn (74) generates distance errors that are positive to the right of the curve.