ULTRASONIC AUTONOMOUS ROBOT LOCALISATION SYSTEM

Lindsay Kleeman
Intelligent Robotics Research Centre
Department of Electrical and Computer Systems Engineering
Monash University
Clayton Victoria 3168
Australia

Abstract

This paper describes a reliable localisation system based on ultrasonic sound, that has been designed for an autonomous robot. The robot position and orientation within a 10 meter square room can be obtained to an accuracy of a few centimeters. The system uses cheap, commercially available ultrasonic transmitters placed in the environment at known positions, and receivers on board the robot that monitor incoming pulses. The transmitters send pulses in sequence with a designated first beacon sending an identifying signature. The transmitters and receivers are not connected electrically, which allows a truly autonomous robot to operate without any communication with the outside world.

In this paper hardware, algorithmic and software aspects of the system are discussed. A prototype system has been successfully constructed and performs both accurately and reliably even in the presence of obstacles, ultrasonic reflections, noise and air turbulence from air conditioning ducts. The prototype takes a few seconds to produce each result with all calculations performed by an 8 bit microprocessor within the receiver on board the robot.

1. Introduction

The localisation system described in this paper has been designed out of necessity, since no other system has been available to provide an autonomous robot with reliable position and orientation information, to the author's knowledge. The overall aims of the autonomous robot project [1] are to enable a robot to map the environment of a room containing obstacles and use that information to navigate through the room using minimum distance (or other measure) paths without colliding with obstacles. The research has applications in many fields, including domestic service, warehouses, hazardous environment work, cleaning, agriculture, mining industry and office automation. The work is particularly suited to situations where the environment and obstacles are not known in advance or may change without notice. Other researchers have also contributed to the project with an infra-red communication system [2, 3]; an infra-red mobile range floor based finder [4]; and distance transform navigation algorithms [1, 5].

Localisation systems reported in the literature are not suitable for our application. MELODI [6] is an ultrasonic system using on board angular detection of three different frequency transmitters in the environment. The system is susceptible to obstacles blocking beacons; accurate to only 1 part in 80; requires 3 rotating arms on the receiver and is expensive. Another system [7] is impractical because it requires a distributed ultrasonic microphone to receive waves orthogonally from a spark gap transmitter. An ultrasonic system [8] for tracking robot manipulators provides accuracies of 0.5mm in 2000 mm at a sample rate of 300 Hz. However, this system requires a high speed connection from the robot to the environment for synchronisation. No such link can be trailed behind an autonomous robot.

The localisation system tested originally in our project [5] employed dead reckoning based on wheel rotations of the robot and its internally stored orientation. This has been found to be unsatisfactory due to the wheels slipping and slight errors in orientation accumulating with time. Dead reckoning could only be used over short distances and, consequently, an absolute position technique was required.

This paper presents an overview of the system in section 1.2. Section 2 gives details of the transmitter design and timing, and the receiver hardware design is presented in section 3. Section 4 examines algorithms to solve for the robot position and software design is discussed in section 5. Prototype performance is given in section 6. Conclusions are presented in section 7.

1.2 Overview of System

In order to keep the cost and complexity down, the system contains physically separate receiver and transmitters. The direct path between a transmitter and the receiver is used, and reflections off objects are ignored.

Since the robot aims to be autonomous, no wires between the robot and the environment can be allowed. This requirement complicates the algorithmic design since the times of transmissions cannot be determined in advance by the receiver. Also, the autonomy of the robot requires that the position and orientation information be available on the robot and not in the environment. Thus, the receiver is located on the robot.

The localisation system consists of ultrasonic beacons located in the environment and an intelligent receiver on board the robot as shown in Fig. 1. The beacons are placed around the edges of the environment allowing unambiguous position determination, in contrast to the Loran-C navigation system [11, 13]. The beacons send pulses in strict time sequence around the room. The first beacon needs to be distinguished initially to allow the receiver to correctly identify which beacon corresponds to a received pulse. This is achieved by sending a double pulse from the first beacon with known pulse widths and separation. During initialisation or resynchronisation of the system, these double pulses are sought out. Note that there is no wire connection or communication, other than ultrasonic sound, between the beacons and receiver.

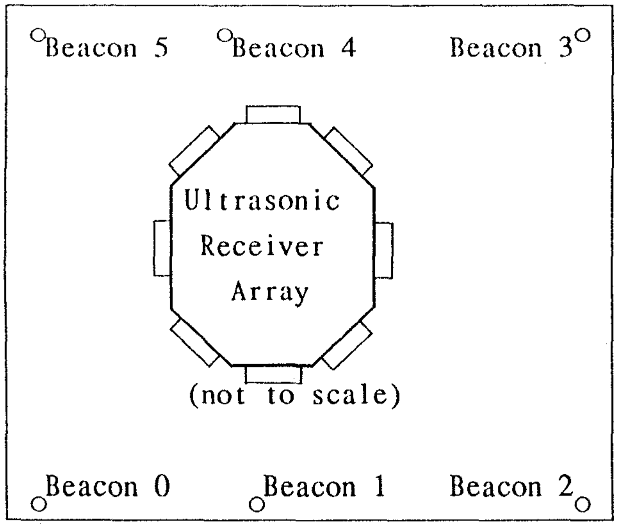

Fig. 1 Overview of Localisation System.

The beacons' pulse sequence is controlled by a central beacon control module that sends signals to each beacon on a common 4 wire telephone cable. Each beacon circuit is identical, with its sequence number set by DIP switches on the board. Fig. 1 shows 6 beacons, however this number may be varied from 3 (minimum to obtain a position) up to 16 (prototype hardware limitation). Each beacon contains circuitry to sense its turn to transmit and an asynchronous start 40 Khz oscillator at to drive the ultrasonic transmitting device(s). Two transmitters are employed to achieve sufficient angular coverage for beacons not placed in corners of the room (beacons 4 and 1 in Fig. 1).

The receiver consists of 8 ultrasonic receiving devices placed in an octagon as shown in Fig. 1. The signals from the 8 devices are amplified and filtered before being analysed by a microprocessor. The software extracts times of arrivals and the identity of pulses. Distance is determined from the speed of sound. The position can then be calculated from data which is the distances to beacons plus an unknown offset. Also the direction of arrival of each pulse is logged so that the robot orientation can be determined. The software overcomes numerous practical problems to produce reliable results. These include:

(1) Reflections and obstacles causing indirect paths for received pulses.
(2) Spurious environmental noises, causing invalid pulses to be received.
(3) Variation in the speed of sound due to temperature and humidity changes.
(4) Air turbulence causing fluctuations in received amplitude and direction of pulses.

2. Transmitter Design

2.1 Control Module Timing

The transmitter control module is responsible for generating signals to correctly sequence the firing of the beacons. In Fig 2, the timing of beacon firing is shown for the 6 beacon configuration employed in the prototype. Note that beacon 0 is identified by the receiver detecting a double pulse. Each pulse corresponds to a burst of 40 Khz ultrasonic sound.

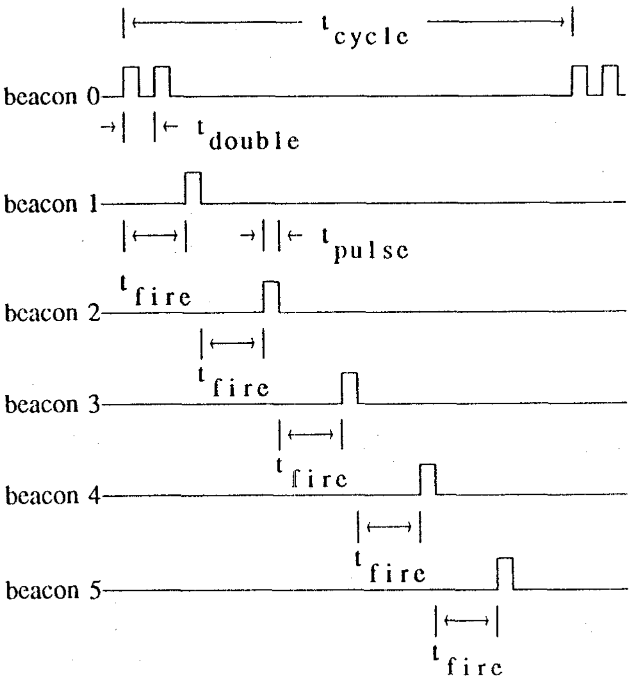

Fig. 2 Beacon Firing Sequence.

2.2 Beacon Circuit Design

Each beacon circuit and the main control module is connected to a common 4 wire telephone cable carrying the power supply, reset and strobe signals, as shown in Fig. 3. The beacon circuits are all identical with DIP switches to set the identity of each beacon (0 to 15). As shown in Fig. 4, each beacon circuit contains a 4 bit counter and comparator circuit. The reset line resets the counter to zero, and each positive edge of the strobe increments the counter. When the counter matches the DIP switch settings, the 40 Khz oscillator starts. The ultrasonic transmitter is then driven for the duration of the strobe pulse.

The logic is implemented in CMOS with a 10 V supply. The reset and strobe lines are fed into Schmitt triggers to filter noise. The oscillator frequency is trimmed to resonate the ultrasonic transmitter at approximately 40 Khz. Due to the large Q of the transmitter, the envelope rise time on the ultrasonic output is about 1 msec which corresponds to about 30 cm propagation distance. The slow rise time of the transmitter is one of the major contributions to position error in the system. The receiver software attempts to compensate for the rise time when calculating the arrival time of a pulse.

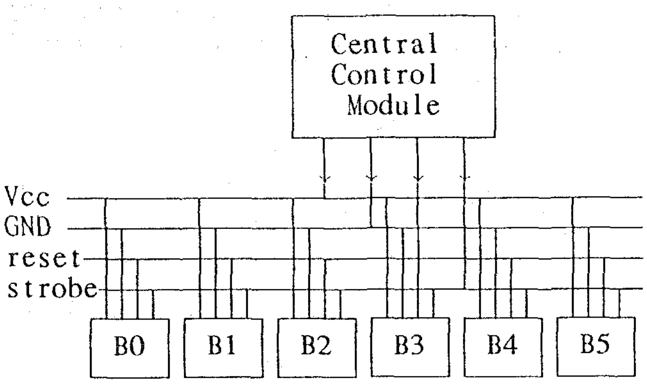

Fig 3 Transmitter Control Structure.

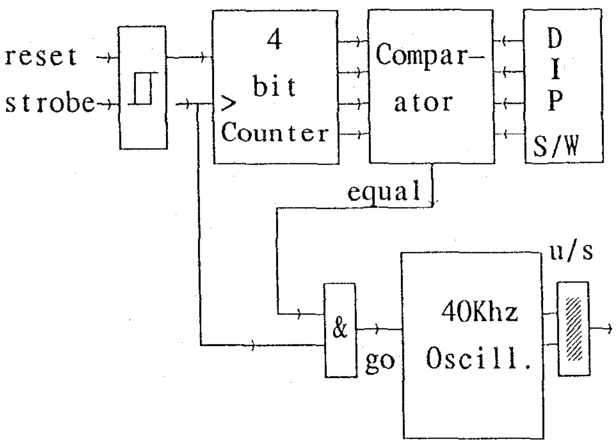

Fig. 4 Beacon Circuit.

2.3 Transmitter Control Module Circuit

The transmitter control module circuit has been implemented with a minimicro (64180 microprocessor card [9]) and an interface board for digital output, line drivers and power supply generation, as shown in Fig. 5. The timing of the reset and strobe lines (Fig. 6) is implemented in software relying on the accuracy of the crystal oscillator of the microprocessor. This may not be suitable for mass production for economical reasons. In the future, the module will be implemented with an EPLD (Erasable Programmable Logic Device) and crystal oscillator.

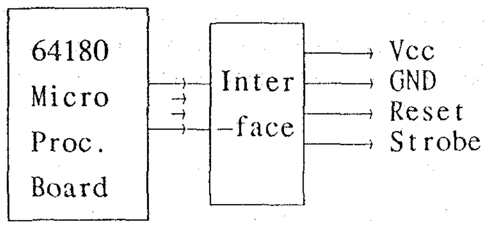

Fig. 5 Transmitter Control Circuit.

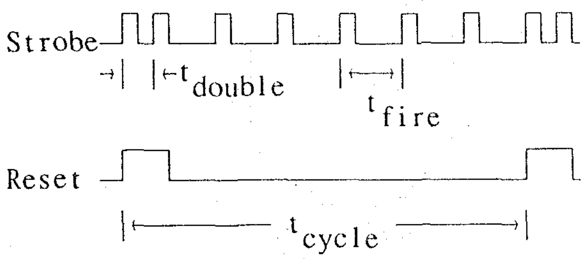

Fig. 6 Strobe Reset Timing.

The time t_{fire} is assigned a value of 2^{16} timer clock tics (approximately 150 msec) which proves convenient later in the receiver software which runs on an identical processor.

2.3 Selection of Beacon Timing Parameters

The firing timing must be very accurate since the receiver accuracy depends on a constant inter-beacon time, t_{fire}, (ie the time between firing successive beacons as shown in Fig. 2). Also care must be taken in determining the value of t_{fire}. As will be seen in section 4.3, the receiving algorithms depend on echoed pulses never being received *after* the next beacon has been fired. That is, the ultrasonic signal must be below the threshold of the receiver in the worst case when the next beacon fires. The worst case is when the centre transmitter wave front (the amplitude attenuates at angles away from the centre beam) is echoed around the room. The amplitude is assumed to decrease linearly with distance. This corresponds to an inverse square law for the power. Thus, to satisfy the condition of below threshold reception after t_{fire} we require:

$$\frac{V_{1m} \cdot (1-\alpha)^n}{t_{fire} \cdot v_{sound}} < V_{thresh}$$

$$t_{fire} > \frac{V_{1m} \cdot (1-\alpha)^n}{V_{thresh} \cdot v_{sound}} \qquad (1)$$

where V_{1m} is the receiver voltage at 1 meter, v_{sound} is the speed of sound (345 m/sec at room temperature), V_{thresh} is the threshold voltage in the receiver, and α is the coefficient of absorption on each of n reflections.

The designer has limited freedom of choice over V_{1m} and V_{thresh}. The sound pressure from the transmitting device is fixed by the device specification and the driving voltage, and so the main controlling influence for V_{1m} is the gain of the receiver circuit. The gain cannot be set arbitrarily high since saturation occurs at 5V in the receiver hardware. Saturation is a problem when the receiver attempts to determine the direction of arrival of pulses. In order to limit this saturation to a distance closer than 2.5 meters, a value of 12.5V at 1 m is employed.

The receiver threshold V_{thresh} is chosen to be above the noise floor and ideally low enough to receive the smallest valid pulse. The direction characteristics of the transmitters and receivers and the room dimensions are crucial in the worst case pulse amplitude on reception. The worst case scenario for the prototype occurs at a distance of 10 meters 45^{o} from centre of the transmitter and 22.5^{o} from centre on receiver device. These combined angles produce an attenuation of approximately 20 dB + 3 dB respectively or a factor of 14.3. Combining this with a distance attenuation of 10 over 10 meters gives a received voltage of 12.5/143 or 87 mV. Unfortunately, the threshold could not be set this low due to noise limitations and a value of 200 mV was chosen.

Another consideration is the constraint on t_{fire}. Obviously, the speed of a measurement is directly dependent on t_{fire} and with V_{thresh}=87 mV and α=0, equation (1) produces a t_{fire} of 480 msec which is unacceptably high. A compromise has been reached with V_{thresh} of 200mV and t_{fire} of 150 msec (which implies the absorption α>0.04 for n=5, which is reasonable in practice). The choice of this threshold results in some loss in pulse reception in the corners of the room, however this has proved not to be a problem in the prototype, due to the redundancy in the system.

The pulse width t_{pulse} must be sufficient to reliably register when rise and fall times of the transmitter are considered. The ultrasonic transducer data suggests a worse case of 2 msec and so a 2.5 msec has been chosen for the prototype. Naturally short pulses are preferred to prevent overlap of direct and reflected pulses during reception. The double pulse separation time has been chosen slightly larger to allow for easy recognition of double pulses. The value of t_{double} in the prototype is 5.5 msec.

3. RECEIVER HARDWARE DESIGN

3.1 Overview

The receiver consists of four modules as shown in Fig. 7. The receiver array has 8 ultrasonic receivers arranged in an octagon, 50 mm a side. The ultrasonic receiver outputs are amplified, filtered and their envelopes extracted. A fast 8 channel multiplexed ADC (Analogue to Digital Converter) digitises the envelope into 8 bits for processing by the microprocessor. The position and orientation information are calculated in the receiver software.

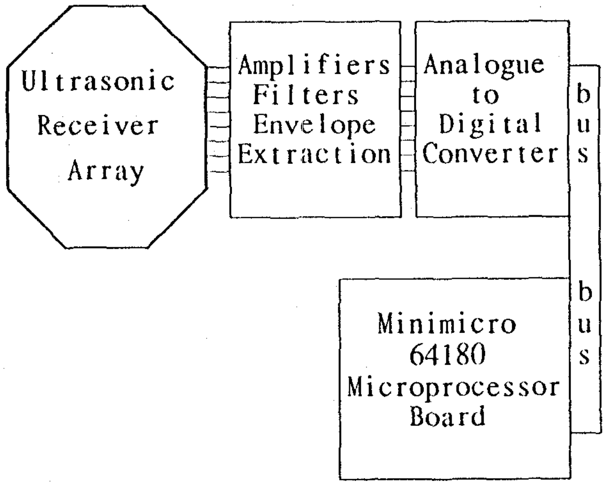

Fig. 7 Receiver Block Diagram.

3.2 Receiver Array

The receiver array should be approximately circular so that the same distance offset applies in any direction. A constant offset in measurements from all beacons has no effect on the final result, since the solution is based only on differences in distance to the beacons. The polarity of each receiver should be the same to prevent errors of half a wavelength or 4mm.

3.3 Analogue Signal Processing

The basic gain component is a trans-impedance amplifier implemented with one opamp out of a LF444 quad low power opamp IC. The gain-bandwidth product of the LF444 is 1 Mhz, and so the gain at 40 Khz is around 25. After amplifying, the signal is band pass filtered at 40 Khz, with a Q of 6 and centre gain of 8.

A half-wave rectifier and envelope filter is then applied. The envelope is followed by a peak detection circuit. The peak decays with a time constant of approximately 0.5 msec. This is shorter than the 2 msec fall time of the received envelope. Finally, the signal is buffered and clamped with a 4.7 V zener diode to protect the ADC.

An 8 channel multiplexing ADC (AD7828) with 2.5 μsec conversion time has been selected to match the receiver array and processing time requirements. The AD7828 directly interfaces to the minimicro bus.

4. ANALYSIS AND ALGORITHMS

This section describes algorithms to calculate the position and orientation accurately and reliably. Firstly, an iterative technique is presented that finds a solution given three beacon measurements which was developed with help from [10]. An error analysis is then mentioned which enables each set of three beacon measurements to be evaluated for their likely errors as a function of the position. The error analysis allows the redundant information of more than three beacon measurements to be fully exploited by the localisation system to both minimise errors and also to reject spurious measurements caused by noise or echoes. Finally, a pre-processing algorithm is presented that quickly rejects much invalid data, saving processing time later and improving the reliability of the results.

4.1 Solution for Three Beacons

The distance data obtained from beacons cannot be used directly since it contains an unknown offset. To eliminate this offset, differences in distances to beacons are used. The locus of points defined by a constant *difference in distance* is a hyperbola. Two hyperbolas intersecting define the location of the robot. Thus, data from three beacons is required to solve for position. An iterative algorithm is described which takes data from three beacons and calculates the position of intersection of the hyperbolas. The beacons are labelled b_0, b_1 and b_2, the position P and the respective distances to the beacons d_0, d_1 and d_2. The angles from the x-axis to the beacons are labelled ψ_0, ψ_1 and ψ_2, as shown in Fig. 8.

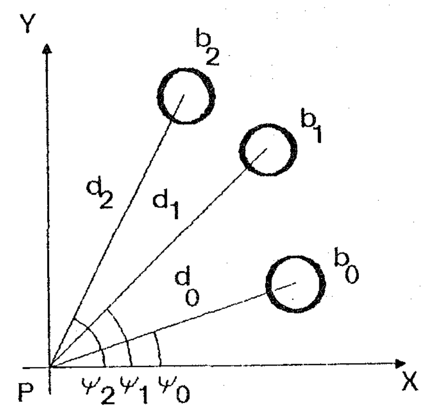

Fig. 8 Iterative Algorithm for Three Beacons.

To preserve d_1-d_0 and vary d_2-d_1 we move along the hyperbola defined by d_1-d_0. As shown in [11], the tangent to this hyperbola is in the direction $(\psi_1+\psi_0)/2$. Suppose we move a small distance δr towards the beacons, then the change in d_1 is denoted δd_1 and similarly δd_2 for d_2.

$$\delta d_1 = -\delta r \cos\left[\frac{\psi_1+\psi_0}{2} - \psi_1\right]$$
$$\delta d_2 = -\delta r \cos\left[\frac{\psi_1+\psi_0}{2} - \psi_2\right] \qquad (2)$$

Solving for δr and then resolving in the x direction gives

$$\delta x = \frac{-\delta(d_1-d_2)\cos\left[\frac{\psi_0+\psi_1}{2}\right]}{2\sin\left[\frac{\psi_0-\psi_2}{2}\right]\sin\left[\frac{\psi_1-\psi_2}{2}\right]} \qquad (3)$$

The same reasoning can be applied to moving along the hyperbola defined by d_2-d_1 constant and varying d_1-d_0.

Combining the two increments gives

$$\delta x = \frac{-\delta(d_1-d_2)\cos\left[\frac{\Psi_0+\Psi_1}{2}\right]}{2\sin\left[\frac{\Psi_0-\Psi_2}{2}\right]\sin\left[\frac{\Psi_1-\Psi_2}{2}\right]} + \frac{-\delta(d_0-d_1)\cos\left[\frac{\Psi_2+\Psi_1}{2}\right]}{2\sin\left[\frac{\Psi_2-\Psi_0}{2}\right]\sin\left[\frac{\Psi_1-\Psi_0}{2}\right]} \qquad (4)$$

The expression for δy is similar. Equation (4) forms the basis of an iterative algorithm which converges on the intersection of the two hyperbolas. Each iteration is not exact because the hyperbola is approximated by a straight line at each stage. The iterations converge slowest when the curvature of a hyperbola is large, as occurs close to beacons. Convergence is obtained within a few iterations usually. All trigonometric functions are translated into square roots, multiplications and divides to achieve higher prototype speed.

4.2 Error Analysis

An expression for the solution error as a function of measurement error for three beacons and the position is derived in [11]. Space limitations prevent presentation here. Once this error is estimated, selection and weighting of solutions can be systematically undertaken as described in section 5.2 below.

4.3 Data Validation

The receiver software has to overcome problems involving invalid data due to the relatively hostile transmission environment. The most common problem occurs when a beacon is obscured by fixed or moving obstacles in the room. An indirect path for a pulse can be followed and the receiver can then be deceived as to the time of flight of the direct path that it is expecting. There are three levels at which invalid data such as this can be rejected:

(i) assembly language signal validation
(ii) low level data validation and
(iii) high level redundancy checking.

This section describes the basis for (ii), whilst (i) is discussed in [11] and (iii) is treated in section 5.2.

A crucial assumption is made regarding pulse reception in the following analysis. *When waiting for a pulse from a particular beacon, it is assumed that the pulse can never arrive earlier than its true arrival time.* Late arrivals are expected if an indirect path is taken. However, an "early" pulse is impossible from the beacon, and must be a consequence of a pulse sent from an earlier time slice. As mentioned in section 2, the firing time between beacons is designed to prevent previous pulses exceeding the receiver threshold later than their time slice.

The validity check is based on the triangular inequality: *The sum of the distances on two sides of a triangle must exceed the third.* Suppose two beacons b_0 and b_1 send pulses to point P as shown in Fig. 9.

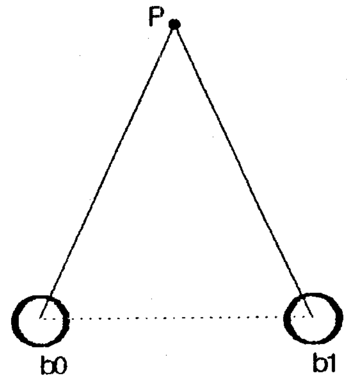

Fig. 9 Receiving Geometry

From the triangular inequality

$$|Pb_0| - |Pb_1| < |b_0b_1| \qquad (5)$$

$$|Pb_1| - |Pb_0| < |b_0b_1| \qquad (6)$$

The validation test checks these inequalities. If one fails, the appropriate measurement is ignored in later processing. Thus if (5) fails, for example, Pb_0 has to be rejected based on the assumption that a measurement can only be too long.

All possible pairs of beacon data are checked. Every effort is made early, rather than later, since data rejected at the preprocessing stage saves a large amount of high level processing and also improves the reliability of the results.

Data rejected for the coordinate and orientation solution is *not* ignored in the remainder of the validation process. The reason is that the invalid data cannot cause the rejection of valid data since the left hand side of the inequalities can only be decreased by the invalid data, and thus no false rejection can occur However, the invalid data can be still useful to reject other invalid data.

5. RECEIVER SOFTWARE DESIGN

5.1 Overview of Software

The external data consists of 8 bit digitised envelope data from the 8 ultrasonic receivers. Two timers are available within the processor. Pulses are logged for arrival time, duration and amplitude by assembly interrupt driven routines. Shared data structures with high level C code are used to signal readiness of new data for processing into coordinates and orientation. Timers 0 and 1 are 16 bits and are configured to interrupt on terminal count. Timer 1 is used as the system time keeper, resolving time to approximately 2.5 μsec. Timer 0 is used to interrupt for general timeout conditions and also to activate pulse receiving assembly code. That is, pulses are detected by polling the 8 ultrasonic receivers until a threshold is exceeded. To minimise the wasted processing time, the receive software is activated in short time windows when the system expects pulses to arrive. The scheduling of receiver interrupts is discussed in [11].

The start up procedure of the system consists of several steps. Hardware is initialised, internal timing is synchronised to incoming pulses, receiver interrupts are set for the next cycle of 6 beacons and the interrupts are then enabled. The high level routines are entered only after the receiver synchronises to the double pulse signature of beacon 0 and data from the first 6 beacon cycle is established. The speed of sound, which can vary by several percent [7] due to temperature and humidity changes, is then *derived* from the measurements by a calibration routine. This routine uses the inherent redundancy in the data to minimise the solution distance discrepancy by varying the speed of sound. The speed of sound is resolved down to 0.25% on the prototype. This calibration technique needs no extra reference devices, in contrast to the technique used in [12].

The receiver software then operates via interrupts and a background high level cycle of data processing. The interrupts for the receiver are rescheduled every beacon cycle, using information from high level solutions when possible. All pulse data in the high level C routines is first checked against the triangular inequality to remove echoes and other spurious data as early as possible. Coordinate information is then calculated by the longsolve routine described below. The orientation is then calculated using only valid data after longsolve and triangular inequality checking.

5.2 Longsolve Procedure

The longsolve routine has the following as input: distance data to all 6 beacons plus an unknown offset; and a set of validity flags for the data set by pulse receiving software and triangular inequality checking. The routine calculates the position of the robot from a carefully weighted average of the best triples of data available, rejecting spurious data. For 6 beacons, there are 20 possible triples of distance data that can determine the position. Some of the 20 possible solutions may be invalid due to erroneous data resulting from echoes or noise. Fortunately, the high degree of redundancy can be exploited to eliminate erroneous data and produce an accurate result. The longsolve routine achieves this by the following steps:

1) Use the last least error triple of beacons to calculate a "quick and dirty" solution.
2) IF the robot moved greater than 1 meter since last error calculations THEN recalculate all errors for 20 triples.
3) Average positions obtained from 7 of least error triples using weights of 1/(estimated error) for x and y axes. In the average use each beacon distance measurement a preset maximum number of times (5 in prototype).
4) REPEAT

   IF the maximum, over all triples of beacons, of the distance deviation from weighted average is greater than the value estimated due to measurement error

   THEN remove that triple from average and try next best triple

   UNTIL (no change) OR (only 3 triples left)

The selection of the preset constants has been optimised by simulations [11]. The above algorithm has been found to have a good ability to reject erroneous solutions resulting from echoes and noisy data. The rejection is achieved by examining the deviation from the weighted average of each triple in step 4.

The inverse error weighting of the solution average would give an optimally small error, if the solutions were independently Gaussian distributed. However, since there are only 5 degrees of freedom in the data, this is not the case. In order to improve the triple data independence, the number of times a particular beacon measurement is used in different triples is restricted in step 3 above. Simulation studies [11] show that this restriction improves the performance.

6. PROTOTYPE PERFORMANCE

The coordinates and orientation produced from the localisation system can be displayed in real time on a scaled map of the laboratory environment. The floor tiles are displayed on the map and a direct comparison can be made between the actual physical position of the robot and the calculated position. Results were obtained from the localisation system consistently within 50 mm of the physical position over the 10 m square laboratory floor. The orientation error was within 10 degrees, since pulse arrivals are resolved to 45 degrees only on the prototype.

This can be improved later by interpolation between receivers. Further statistical measurements were performed on the results to monitor the standard deviation of the measured position with a stationary robot. These results are summarised below:

X (mm)	Y	Samples	Std Dev. (mm) X	Y
4223	7902	532	8.0	11.0
4003	4892	1684	6.8	4.3
7339	5885	639	8.5	5.5

Each measurement took 3 to 4 seconds on a 8 bit microprocessor running at 9.11 Mhz.

Tests were also performed on the dynamic performance, by plotting the positions when the robot moves in a straight line. The positions lagged behind by a few seconds but followed the robot path almost exactly when the speed was limited to 1 m/sec.

7. CONCLUSIONS

A localisation system based on ultrasonic sound has been successfully designed and constructed. The system performs accurately for coordinate estimation and less so for orientation estimation. The key features of the system are:

(1) ability to reject spurious data due to obstacles, noise and air turbulence.
(2) the calibration for the speed of sound with no extra hardware by utilising inherent redundancy.
(3) no interconnections between the environment and the robot, allowing autonomous operation.
(4) no audible and little visible impact is made on the environment.
(5) low cost and small size.
(6) low power consumption and no moving parts.

REFERENCES

[1] R. A. Jarvis, "An intelligent robots research centre – work in progress," Proceedings of 19th ISIR, International Symposium and Exposition of Robotics, pp. 149–186, Sydney, Australia, Nov. 1988.

[2] K. Ajay, "A medium speed infra-red modem for digital communications," Monash University, Dept. Electrical and Computer Systems Engineering, Technical Report 1989.

[3] J. Byrne "A reliable communications link protocol," Monash University, Dept. Electrical and Computer Systems Engineering, Technical Report 1989.

[4] K. Ajay, "A floor line rangefinding system," Monash University, Dept. Electrical and Computer Systems Engineering, Technical Report MECSE 89–4 1989.

[5] R. A. Jarvis, J. C. Byrne and K. Ajay, "An intelligent autonomous guided vehicle: localisation, environmental modelling and collision-free path finding," Proceedings of 19th ISIR, International Symposium and Exposition of Robotics, pp. 767–792, Sydney, Australia, Nov. 1988.

[6] T. Arai and E. Nakano, "Development of measuring equipment for location and Direction (MELODI) using ultrasonic waves," Trans. ASME Journal of Dynamic Systems, Measurement and Control, Vol.105, September 1983, pp. 152–156.

[7] W. E. Moritz, P. L. Shreve and L. E. Mace, "Analysis of an ultrasonic spatial locating system," IEEE Trans. on Instrumentation and Measurement, Vol. 25, No.1 March 1976, pp. 43–50.

[8] M. Dickinson and A. S. Morris, "Coordinate determination and performance analysis for robot manipulators and guided vehicles," IEE proceedings Pt. A, No.2, February 1988, pp. 95–98.

[9] Monash University, "CS-64180 Mini microprocessor card technical manual," Revision A, March 1988.

[10] E. M. Cherry, Private communication concerning the Cherry Omega navigation receiver, Monash University, 1988.

[11] L. Kleeman, "Ultrasonic autonomous robot localisation system," Monash University, Dept. Electrical and Computer Systems Engineering, Technical Report MECSE 89–8, 1989.

[12] P. K. Chande and P. C. Sharma, "A fully compensated digital ultrasonic sensor for distance measurement," IEEE Trans. on Instrumentation and Measurement, Vol.IM-33, No.2, June 1984, pp. 128–129.

[13] R. L. Frank, "Current developments in Loran-C," Proceedings of the IEEE, Vol 71, No.10 October 1983, pp. 1127–1139.