A Real Time DSP Sonar Echo Processor

Andrew Heale and **Lindsay Kleeman**

Intelligent Robotics Research Centre
Department of Electrical and Computer Systems Engineering
Monash University, Victoria 3800, Australia
Andrew.Heale@eng.monash.edu.au Lindsay.Kleeman@eng.monash.edu.au

Abstract

This paper describes a new highly accurate fast self-contained sonar sensor. Transmission and echo analysis are performed at repetition rates exceeding 27 Hz for ranges to 5.4 metres. The sensor contains low noise variable gain preamplifiers, two 1 MHz 12 bit ADC receivers and a DSP echo processor. Optimal arrival time estimation is performed by the DSP using matched filtering of echoes with short duration and wide bandwidth. With the sensor mounted on a mobile robot, map results are derived from scanning with the sensor as the robot continuously moves.

1. Introduction

Robotic systems are often sensor bound. The performance of tasks such as map building, localisation and obstacle avoidance are critically dependent on the availability of high accuracy, reliable and fast sensor systems. Sonar sensing traditionally is seen as the cheap alternative that is unreliable and inaccurate due the popular use of sensor technology, such as the Polaroid ranging module [1,2]. In particular, sonar is often characterised as a poor angle sensor due to the reliance on resolving angle merely to within the beamwidth a single transducer [1,2]. With the decreasing cost and increasing performance of Digital Signal Processors (DSP), it is now realistic to perform computationally intensive sonar echo processing in real time *and* with optimal results *and* with greater bearing/range accuracy than laser range finders. By using a DSP, this paper presents the first self contained sonar sensor that achieves optimal signal processing of echoes at close to real time rates – that is transmit repetition rates limited mainly by the time of flight to the furthest range of interest. For a 5.4 metre range, this limitation is 30 Hz. We achieve a sustained 27 Hz repetition rate in this paper.

Sonar signal processing bears similarity to that of RADAR and this is borne out by papers that using RADAR based techniques [3,4,5]. From RADAR theory [6], the minimum variance arrival time estimator in the presence of white gaussian receiver noise is the matched filter – also called template matching. This filter is based on finding the peak of the cross correlation of the echo with *a priori* stored pulse shape templates. This approach has been employed extensively in [3,7,8,9] and no other sonar system is known to perform at greater range or bearing accuracy than the template match approach. Previous template match implementations have been based on transferring complete receiver echo data to a PC. More recently this has been achieved at twice real time receiver data rates via the PCI bus [8]. Not only is this technically complex but invariably this leads to increased receiver noise and significant computational latency and communication burden on the main robot computer. The work presented in this paper solves both these problems by *local* DSP processing of the echo then communication of just the higher level data over a serial line which relieves computational burden. Moreover, more sophisticated signal processing can be implemented with the DSP than was possible with the PCI sonar system reported earlier [8].

The paper is organised in a bottom up fashion. Where possible each section is supported by realistic experimental results that highlight both the benefits and limitations of the techniques under discussion. Section 2 introduces the sensor and DSP hardware. Returns are extracted on-the-fly using assembly coded signal processing onboard the DSP, as discussed in Section 3; while Section 4 discusses the implementation of template matching within a DSP context, yielding very accurate range estimates. Receiver echo arrival times and correlation coefficients are used to associate the data from the two receiver channels to form high level range/bearing estimates to targets with an indication of their reliability in Section 5. Finally, data association across a windscreen wiper scan cycle of the sensor is discussed in Section 6 where preliminary map results are presented from the sensor. Conclusions and future work form the last section of the paper.

2. DSP Sonar Sensor Hardware

The DSP sonar sensor is shown in Figure 1 mounted on the panning mechanism of the mobile robot Werrimbi. The sensor has two upper transmitters and two lower receivers configured 40 mm between centres. Polaroid 7000 series transducers are used in the transmitters and receivers. The DSP sonar communicates with a 233 MHz Pentium on Werrimbi via a high speed serial link. The DSP sonar is responsible for transmitting pulses from either transmitter (although only the central transmitter is deployed in this paper) and processing the returned echoes on both receivers. Arrival times, amplitudes and correlation coefficients (as discussed in section 4.2) are relayed to Werrimbi along the serial cable for higher level processing.

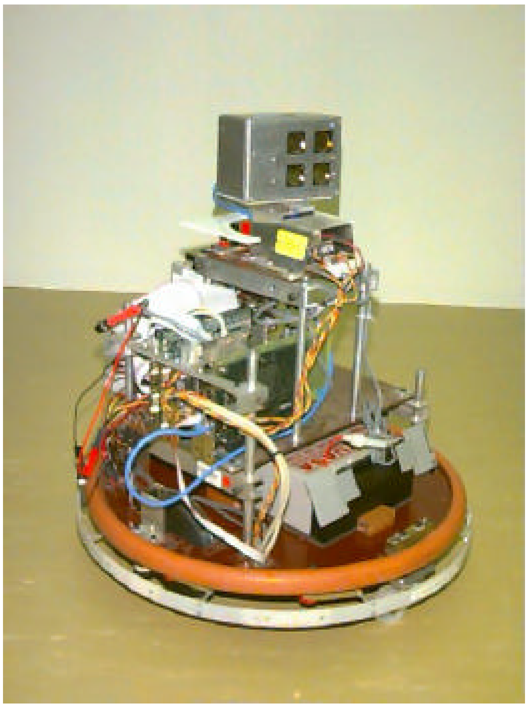

Figure 1 – DSP Sonar mounted on Werrimbi.

Figure 2 shows the block diagram of the hardware design. An Analog Devices 2181 DSP was chosen due to its large single clock cycle on-chip RAM of 80 k bytes, allowing receiver echo pulses to be extracted and stored directly on the DSP chip. The receiver channels are amplified and low pass filtered before sampling with 12 bit ADCs at 1 MHz. Separate transmitter and receiver transducers (Polaroid 7000 series) were employed to lower receiver noise levels. The 300 Volt transducer bias is generated within the sensor by a high voltage DC-DC converter. A single 5 Volt supply powers the sensor.

By tightly coupling the high speed DSP to the transmit and receive hardware within the sensor, significant simplifications can be made in the hardware design over previous generations of sonar sensors [3,7,8,9]. The transmitter circuitry allows a programmable digital pulse train to be sent to the transducer without the need for preloaded memory buffers as required previously [8]. Instead the DSP directly writes to the transmit logic every microsecond under interrupt control. Another simplification is that the gain control circuitry in the receiver preamplifier is digitally controlled to make the transition between low gain and high gain via an RC network feeding the analogue gain input of AD600 chips.

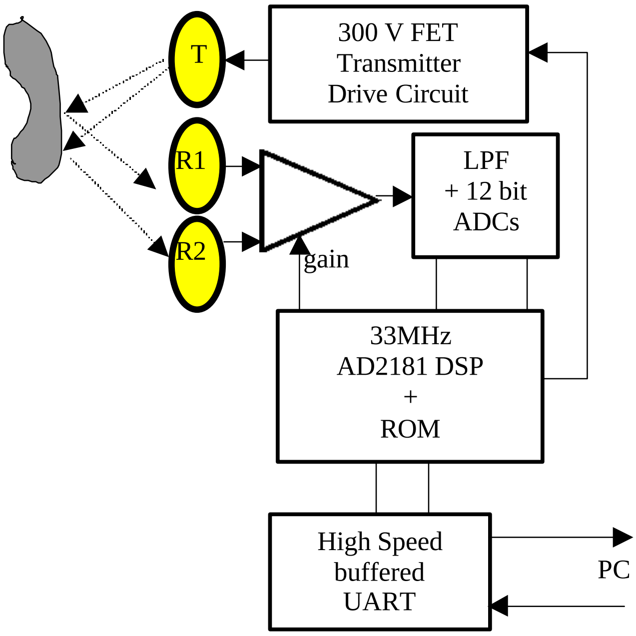

Figure 2: DSP Sonar Hardware Block Diagram

3. On-the-fly Pulse Capture DSP Assembly Code

The DSP software is organised into two stages. During the first stage, highly optimised assembly code performs on-the-fly processing of the samples from the two receivers to extract discrete pulses that exceed the noise floor. The second stage processes the extracted pulses with C code to partition closely spaced yet separable pulses and to extract arrival times using matched filtering. The next section of the paper describes this second stage.

Stage one has a main program and a timer interrupt routine that runs every microsecond. The timer interrupt routine fetches the next 12 bit ADC samples from the two receiver channels and places them into a circular buffer. The interrupt routine is also responsible generating the transmit pulse and receiver gain changes.

The main program runs in a loop where each iteration processes the block of data acquired since the previous iteration. The two channels are processed independently through four stages: DC bias removal, thresholding, aggregation and storing into a pulse data structure.

3.1. Thresholding

Each receiver sample is compared with a threshold to classify the sample as noise or part of an echo pulse. Since an echo pulse can legitimately pass through zero, a windowing technique is used. Samples greater than the threshold are deemed to be part of an echo, along with the 30 samples before and after them. If another above-threshold sample exists in this range, the two ranges are merged, creating a aggregate range that contains a single pulse of several oscillations.

A threshold level is obtained from a lookup table. The level is based on the noise level expected at the current time since firing, with 512 sample granularity. The time dependency of the threshold is necessary due to the time varying gain in the receiver preamplifiers.

3.2. Aggregation

When there is a run of between 30 to 60 samples not exceeding the threshold, two separate returns are identified, but their extremities overlap. Overlapping ranges are merged to prevent repetition, then the resulting ranges are saved for later template matching.

4. DSP Stage 2 Processing

Stage 2 processing by the DSP occurs after the receiver channels have been logged and stored in a pulse table. The processing time in stage 2 occurs in addition to the time of flight between transmitting and receiving the furthest echo – 32 milliseconds (approximately 5.4 metres) in our implementation. This processing time then directly impacts on the real time performance of the sensor since it takes place sequentially with respect to the capture time. In contrast, the stage 1 processing occurs concurrently with respect to the capture time and consequently must keep up with the incoming data to avoid buffer overflow errors.

Stage two processing consists of two major sequential tasks: pulse splitting and template matching.

4.1. Pulse Splitting

The threshold algorithm for extracting pulses works well when received pulses are sufficiently well spaced so that the signal level drops below the threshold for at least 30 samples (microseconds) between pulses. There are situations where this is unrealistic:

- Closely spaced targets – partial pulse overlap occurs.
- A pulse preceded by a large amplitude pulse – the ringing tail of a large amplitude pulse extends into the next pulse period above the threshold amplitude.

Pulse splitting aims to identify these situations and partition the pulse into two or more pulses that can be processed by the template matching approach described below to yield distinct sonar targets. If pulses overlap sufficiently to grossly distort the pulse shape, the template matching approach fails due to a low correlation coefficient described below. Pulse splitting does not claim to resolve this situation. Separating significantly overlapping pulses is a potentially intractable problem not addressed in this paper.

The approach taken in the pulse splitting algorithm is based on recognising that pulses have a unique maximum absolute value, usually in the centre of the pulse and that the pulse duration is limited. A pulse is found if a local maximum exists that is maximum over an interval at least as wide as the nominal duration. The algorithm finds all local maxima within the pulse and checks from the largest of these to the smallest for a clear maximum half a nominal duration each side. Once declared a clear maximum, the local maxima within the newly declared pulse are removed from consideration. The procedure continues until all maxima are either checked or removed from consideration. The pulse start and end points are chosen to be zero-crossing points if possible. Referring to Figure 4, the first and second pulses from the left pulses have been demarcated by pulse splitting.

4.2. Template Matching

To determine the echo pulse arrival times, matched filtering is performed on the echo pulses extracted during stage one and pulse splitting during stage 2 of the processing. Matched filtering obtains the arrival time by cross correlating the received echo pulse with an echo template stored in the sensor. A template is a noise free pulse shape computed offline from a calibration pulse obtained from a plane at 1 metre range straight ahead. The template is shifted across the echo to find the maximum correlation. By fitting a parabola to the maximum three correlations and their shift times, a very accurate arrival time estimate is obtained [3].

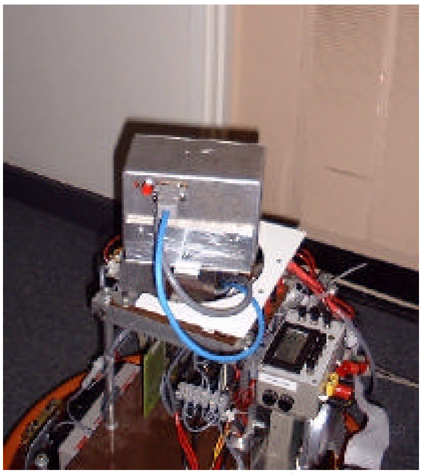

Figure 3 – Sonar Sensing of a Doorjamb.

A complication in the template match process is that the pulse shape depends on the angle of arrival and the range. This change of pulse shape with range and angle can be computed offline [3]. For each metre of range several different template pulses are stored within the DSP RAM corresponding to all the possible arrival angles. To speed up computation and to save precious DSP RAM memory, fewer templates are stored for more distant ranges since the possible arrival angles reduce with range. The template match procedure implemented on the DSP exploits the DSP's capability to perform a cross-correlation operation extremely quickly. Template matching is tried across all possible angles at the given range and the closest match is selected for arrival time estimation purposes.

4.3. Stage 2 Experimental Results

To illustrate the DSP processing, the sonar sensor was pointed at a complex set of reflectors formed by a doorjamb, as shown in Figure 3. The doorjamb consists of three small concave right angled corners facing towards the sensor. Concave corners reflect incoming pulses back in the opposite direction towards the receivers as can be observed by the pulses extracted from the right-hand receiver shown in Figure 4.

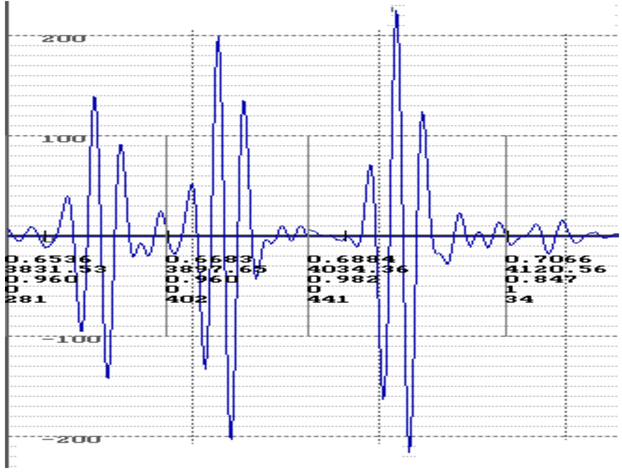

Figure 4 – Right-hand Receiver Pulses Extracted by DSP Code. The start of each pulse is marked by a short vertical gray line; and the first number under each pulse is its range in metres.

The DSP sonar sensor can be commanded to continuously report parameters on up to 5 pulses per receiver. Tables 1 and 2 are these parameters for the doorjamb example. The limit of 5 pulses per receiver was imposed so that the DSP is not delayed by serial communications to the PC. The parameters for 10 pulses fit within 128 bytes which is the size of the UART buffer. Note that pulses from targets just 11 mm apart in range have been identified in the first two entries in Tables 1 and 2. The amplitude column represents the maximum minus the minimum of the pulse. Amplitude information is useful in classifying targets based on their reflectivity and also can be exploited in the association of pulses between two receivers discussed below. The duration column in Tables 1 and 2 represents the number of samples in the pulse after splitting. The correlation coefficient lies between –1 and +1 and is an important outcome of template matching [3]. It represents how well the received pulse matches the closest shaped template. Correlation coefficients above 95% indicate that a reliable arrival time estimate has been obtained. Values below 95% are usually a result of pulse overlap between targets closer than 10 mm difference in range. These pulse arrival times are unreliable for bearing estimation purposes but still give an indication that an obstacle is present.

In our current implementation, stage 2 DSP processing can be performed in about 500 microseconds per pulse processed on a 33 MHz 2181 DSP from Analog Devices. In continuous fire mode where only the first 5 pulses on each channel are processed, this translates to a maximum sensor cycle time of 32+5=37 milliseconds or a 27 Hz repetition rate. Thus the stage two processing represents a 13.5% overhead compared to real time performance where all processing occurs during reception.

Table 1 – Left Receiver Pulse Data from Doorjamb

Range (m)	Arrival time (usec)	Amp.	Dur. (usec)	Correl. Coeff. %
0.6514	3831.99	284	87	96.5
0.6627	3898.15	386	75	96.5
0.6859	4034.91	397	114	98.5
1.3629	8017.29	30	61	95.0
2.8101	16530.29	32	62	96.5

Table 2 – Right Receiver Pulse Data from Doorjamb

Range (m)	Arrival time (usec)	Amp.	Dur. (usec)	Correl. Coeff. %
0.6521	3836.09	255	83	96.5
0.6628	3898.75	427	104	96.5
0.6854	4031.53	679	72	98.5
0.6946	4086.08	81	63	95.0
1.3327	7839.68	35	41	96.5

5. Receiver Data Association

In order to derive the bearing angle to physical targets, pulse arrival times must be associated between the left and right receiver channels. Ambiguities are possible in this process when there are many closely spaced pulses. To guard against this, we adopt a conservative approach and actively seek out possible ambiguities and flag the results appropriately. The effect of an incorrect association between two pulses is that the bearing angle is grossly in error.

Every pulse extracted from the left channel is compared with every pulse from the right. An association is declared *reliable* if the following conditions are met:

- Arrival times are consistent with the receiver physical spacing, SEP (=40 mm) – that is arrival times differ by less than *sin(max_angle)*SEP/speed_sound*
- Pulse amplitude ratio is between 0.5 and 2.0.
- Both correlation coefficients over 95%

If only the last condition is not met, the association is flagged as *unreliable*. Range and bearing of all associations are calculated [3] and called targets. If any pulse is associated with more than one pulse on the other receiver, all associations involving these pulses are demoted to *unreliable*. Pulses that are not associated with any pulse are retained and are assumed to correspond to targets in the pointing direction of the receiver and are marked *extremely unreliable*. These targets combined with amplitude information across several measurements taken at different directions can give some indication of the roughness and position of a surface [10].

5.1. Target Association Results

The results of associating the measurements in Tables 1 and 2 are presented in Figure 5. The dark crosshairs represent reliable targets and these can be seen to correspond well with an overlaid tape measured representation of the doorjamb. An ‘extremely unreliable’ target is shown to the right of the doorjamb on the centre axis of the diagram and results from a pulse on just one receiver with no association. This pulse is most likely caused by a double reflection off the robot and the doorjamb.

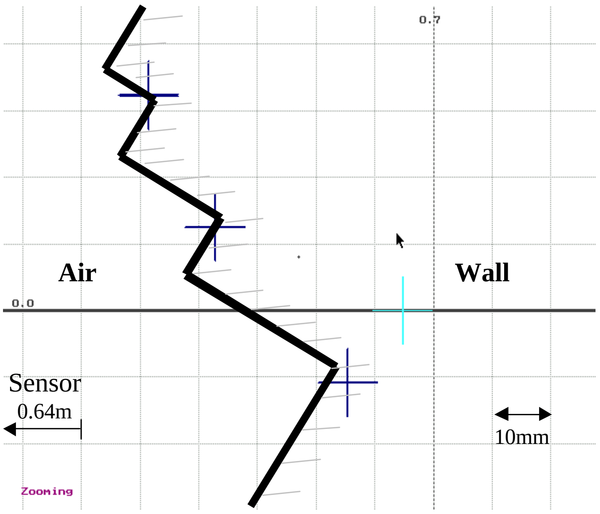

Figure 5 – Sonar range bearing measurements as cross hairs overlaid with tape measurements of the doorjamb in solid lines. The sonar sensor is 640 mm off to the left. Figure 3 shows this scene looking from behind the sonar.

6. Scanned Data Association

The DSP sonar system has been mounted on the mobile robot Werrimbi reported elsewhere [7]. The sensor is scanned back and forth in front of the robot using a PID servo mechanism with encoder angle resolution of 0.18 degrees. One scan goes from 0 to –90 to +90 to 0 degrees where 0 is straight ahead. On every sensor reading, odometry and pan angle information is recorded. Within one scan target information is merged based on simple heuristics. To achieve real time performance, a Kalman filter based map approach [7] will be employed only at a higher level to account for odometry errors and map feature merging between scans. Within a scan, relative odometry accuracy is assumed to be sufficient to merge results reliably.

Each target detected within a scan is given a weighting, heuristically calculated from reliability, amplitude and bearing. The scan is searched for a previously located feature corresponding to the target. The matching strategy is to find the existing feature that is ‘closest’ to the new target. In calculating the distance, the range coordinate is given ten times the importance of the bearing, corresponding to the uncertainty in each. The final distance must fall below a threshold to be considered a match. If a match is made, the new target’s position is merged with the feature using a weighted average in Cartesian space, and the weight of resulting feature is increased. If no match is found, a new feature is added to the scan. Finally, weak features, those with low weightings are considered unreliable and stripped from the scan.

Results from moving a robot down the corridor in Figure 6 are shown in Figures 7 and 8. Note that the moldings and door jams are good reflectors.

Figure 6 – Scanned data was taken from this corridor, where the robot starts at the middle bottom of the photo and heads away.

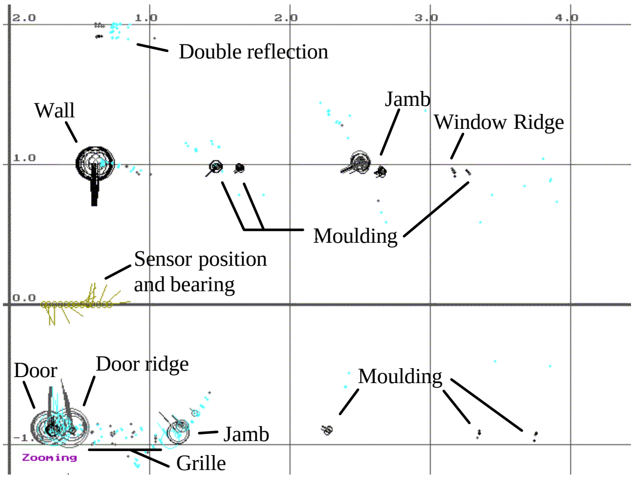

Figure 7 – Raw sonar data from the robot moving along corridor in Figure 6. The corridor runs left to right. The size of the circles indicates the amplitude and the line points back to the sensor. Black targets are reliable, gray unreliable and light gray (blue) are very unreliable. The robot position starts at (0,0) and moves right as shown.

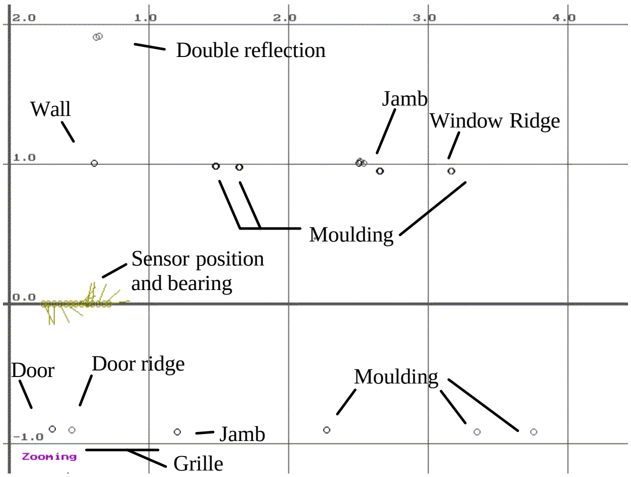

Figure 8 – Fused targets from figure 7.

7. Conclusions and Future Work

The paper has presented a new DSP real time sensor design based on the template matched arrival time estimator that has proven accuracy and robustness characteristics [3]. This paper has not re-iterated these accuracy measurements, but rather illustrated the DSP sensor performance by results from real environments.

The first advantage of the DSP implementation is that processing can be done locally – obviating the data communication problem to a central computer. The second major advantage is that very fast signal processing can be achieved, enabling central processing to be devoted to higher level computing activities such as simultaneous localisation and map building. Sustained repetition rates of 27 Hz have been achieved with the DSP sonar system, enabling *accurate* real time map building and localisation tasks to be feasible with sonar.

Future work will be performed on high level merging of scan data to form map features, such as planes and corners, in real time. Another project underway is to design a DSP based sonar ring to enable simultaneous sonar sensing surrounding the robot.

Acknowledgment

Steve Armstrong is gratefully acknowledged for technical support. The financial support of a large Australian Research Council grant is acknowledged.

References

[1] T. Tsubouchi, “Nowadays trends in map generation for mobile robotics”, Proceedings IROS’96, Osaka Japan 1996, pp 828-833.

[2] J. Budenske and M. Gini, “Why is it so difficult for a robot to pass through a doorway using ultrasonic sensors?,” IEEE Conference on Robotics and Automation, San Diego CA, 1994 pp. 3124-3129.

[3] L. Kleeman and R. Kuc, "Mobile robot sonar for target localization and classification", International Journal of Robotics Research, Volume 14, Number 4, August 1995, pp 295-318.

[4] K. Jorg and M. Berg, “Mobile robot sonar sensing with pseudo-random codes”, Proceedings 1998 IEEE Conference on Robotics & Automation, Leuven, BelgiumMay 1998, pp 2807-2812.

[5] H. Peremans, K. Audenaert, and J. M. V. Camperhout, “A high-resolution sensor based on tri-aural perception,” IEEE Transactions on Robotics and Automation, vol. 9, pp. 36-48, 1993.

[6] P M Woodward, Probability and Information Theory with Applications to Radar. 2nd ed. 1964, Oxford: Pergamon Press.

[7] K S Chong and L. Kleeman, ”Feature-based mapping in real, large scale environments using an ultrasonic array”, International Journal Robotics Research, Vol 18, No. 1, Jan 1999, pp. 3-19.

[8] L. Kleeman, “Fast and accurate sonar trackers using double pulse coding”, IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyongju, Korea, October 1999, pp.1185-1190.

[9] H. Akbarally and L. Kleeman, "3D robot sensing from sonar and vision", IEEE International Conference on Robotics and Automation 1996, Minneapolis, Minnesota, April 1996 pp. 686-691.

[10] Bozma and R Kuc. "Characterizing the environment using echo energy, duration, and range: the ENDURA method", in IEEE/RSJ International Conference on Intelligent Robots and Systems. 1992. Raleigh, NC: pp. 813-820.