A decoupled, distributed AUV control architecture

Stefan B. Williams, Paul Newman, Gamini Dissanayake, Julio Rosenblatt, Hugh Durrant-Whyte
Australian Centre for Field Robotics
University of Sydney NSW 2006, Australia

Abstract

Current work on undersea vehicles at the Australian Centre for Field Robotics concentrates on the development of terrain-aided navigation techniques, sensor fusion and vehicle control architectures for real-time platform control. Accurate position and attitude estimation and control methods use information from scanning sonar to complement a limited vehicle dynamic model and unobservable environmental disturbances. In this paper we present the vehicle control architecture currently running on the Oberon submersible. This architecture is based on the Distributed Architecture for Mobile Navigation. We use a distributed, decoupled control paradigm to facilitate the tuning of individual control modes. A number of behaviours have been created to direct the motion of the vehicle. We show that when coupled with a low-level terrain-aided navigation scheme, effective control of the vehicle can be achieved.

I. Introduction

Current work on undersea vehicles at the Australian Centre for Field Robotics concentrates on the development of terrain-aided navigation techniques, sensor fusion and vehicle control architectures for real-time platform control. Accurate position and attitude estimation and control methods use information from scanning sonar to complement a limited vehicle dynamic model and unobservable environmental disturbances. Key elements of this work include the development of sonar feature models, the tracking and use of these models in mapping and position estimation, and the development of low-speed platform models used in vehicle control.

One of the key technologies being developed in the context of this work is Simultaneous Localisation and Map Building (SLAM). The robot typically starts at an unknown location with no a priori knowledge of landmark locations. From relative observations of landmarks, it simultaneously computes an estimate of vehicle location and an estimate of landmark locations. While continuing in motion, the robot builds a complete map of landmarks and uses these to provide continuous estimates of vehicle location.

Map building represents only one of a number of functionalities that are required in order for a robot to operate autonomously in a dynamic environment. An AUV must also have the ability to make decisions about the control actions to be taken in order for it to achieve it's goals. These goals may change as the mission progresses and there must be a mechanism in place for the robot to deal with unforeseen circumstances that may occur during the course of a mission. We have developed a distributed, decoupled control architecture based on the Distributed Architecture for Mobile Navigation (DAMN) as proposed in [10].

In this paper we present the vehicle control architecture currently used on the Oberon submersible followed by results of the application of Simultaneous Localisation and Map building to estimate the motion of the vehicle. Section II presents the vehicle sensors and actuators while section III describes the distributed, decoupled control scheme currently operating on the vehicle. In Section IV we present a summary of the Simultaneous Localisation and Map building Algorithm. In Section V we describe a series of trials and show the results of applying this technique to data collected in a natural terrain environment. Finally, Section VI concludes the paper by summarizing the results and discussing future research topics as well as on-going work.

II. The Oberon Vehicle

Tests of our underwater navigation techniques are performed on a mid-size submersible robotic vehicle called Oberon designed and built at the Centre (see Figure 1). The vehicle is equipped with two scanning low frequency terrain-aiding sonars and a colour CCD camera, together with bathyometric depth sensors and a fiber optic gyroscope [12]. This device is intended primarily as a research platform upon which to test novel sensing strategies and control methods. Autonomous navigation using the information provided by the vehicle's on-board sensors represents one of the ultimate goals of the project [8].

A. Embedded controller

At the heart of the robot control system is an embedded controller. Figure 2 shows a schematic diagram of the vehicle sensors and their connections. The Oberon robot uses a CompactPCI system running Windows NT that interfaces directly to the hardware and is used to control the motion of the robot and to acquire sensor data. While the Windows operating system doesn't support hard real-time performance, it is suitable for soft real-time applications and the wide range of de-

Fig. 1. Oberon at Sea

velopment and debugging tools make it an ideal environment in which to test new navigation algorithms. Time-critical operations, such as sampling of the analog to digital converters, are performed on the hardware devices themselves and use manufacturer supplied device drivers to transfer the data to the appropriate processes.

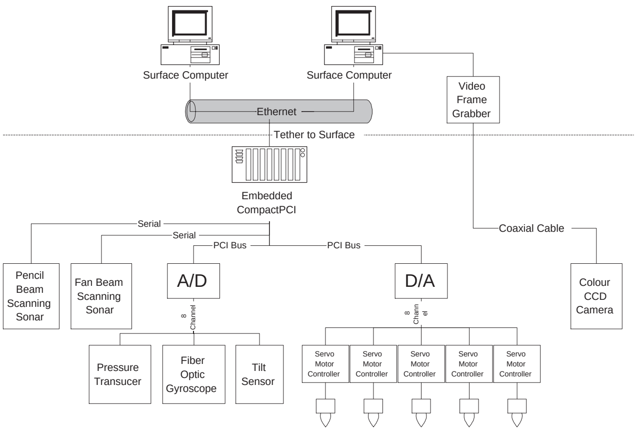


Fig. 2. Vehicle System Diagram

The sensor data is collated and sent to the surface using an ethernet connection where a network of computers are used for further data processing, data logging and to provide the user with feedback about the state of the sub. Communications between the computers at the surface and the sub are via a tether. This tether also provides power to the robot, a coaxial cable for transmitting video data and a leak detection circuit designed to shut off power to the vehicle in case of a leak. While some effort might have been spent on eliminating the tether, it was felt that the development of the navigational techniques was of more immediate interest.

B. Sonar

Sonar is the primary sensor of interest on the Oberon vehicle. There are currently two sonars on the robot. An Imagenex sonar unit operating at 675 kHz has been mounted at the front of the vehicle. It is positioned such that its scanning head can be used as a forward and downward looking beam This enables the altitude above the sea floor as well as the proximity of obstacles to be determined using the wide angle beam of the sonar.

The second sonar is a Tritech SeaKing. This is an imaging sonar and has a dual frequency narrow beam sonar head that is mounted on top of the sub and is used to scan the environment in which the sub is operating. It can achieve 360^o scan rates on the order of 0.25 Hz. The information returned from this sonar is used to build and maintain a feature map of the environment.

C. Internal Sensors

An Andrews Laser Gyro has been included in the Oberon robot to allow the robot's orientation to be determined. This sensor provides yaw rate information and is used to control the heading of the sub. We currently estimate the bias in the gyroscope reading prior to a mission. The bias compensated yaw rate is then integrated to provide an estimate of vehicle heading. Because the yaw rate signal will inevitably be noisy, the integration of this signal will cause the estimated heading to drift with time. At present, missions do not typically run for longer than 30 minutes and yaw drift does not pose a significant problem on this time frame. In the future, a compass will be added to the system to allow us to periodically reset the heading for longer missions. This will also allow us to detect changes in the yaw rate bias that typically occur as a result of changes in the internal temperature of the unit.

A pressure sensor measures the external pressure experienced by the vehicle. This sensor provides a voltage signal proportional to the pressure and is sampled by an analogue to digital converter on the embedded controller. The pressure can then be converted to a depth below the surface of the ocean. Feedback from this sensor is used to control the depth of the sub.

D. Camera

A Panasonic camera in an underwater housing is mounted externally on the vehicle. It is used to provide video feedback of the underwater scenes in which the robot operates. This is a colour camera and sends the video signal to the surface via the tether. A Matrox Meteor card is then used to acquire the video signal for further image processing.

E. Thrusters

There are currently 5 thrusters on the Oberon vehicle. Three of these are oriented in the vertical direction while the remaining two are directed horizontally. This gives the vehicle the ability to move itself up and down, control its yaw, pitch and roll and move forwards and backwards. This non-holonomic thruster configuration does not allow for lateral motion but this does not pose a problem for the missions envisaged for this vehicle.

III. Vehicle Control System

Control of a mobile robot in six dimensional space in an unstructured, dynamic environment such as is found underwater can be a daunting and computationally intensive endeavour. Navigation and control both present difficult challenges in the subsea domain. We have developed a distributed, decoupled control architecture to help simplify the controller design.

The vehicle control architecture currently running on the sub is based on the Distributed Architecture for Mobile Navigation (DAMN) as proposed in [10]. This behaviour-based control architecture uses a centralised arbiter to combine votes from various behaviours running in the system in order to determine the optimal course of action to pursue. A behaviour encapsulates the perception, planning and task execution capabilities necessary to achieve one specific aspect of robot control, and receives only that data specifically required for that task [2].

A. Decoupled control

By decoupling the control problem, individual controller design is greatly simplified. In the case of the Oberon vehicle, vertical motion is controlled independently of its lateral motion using two separate PID controllers. These controllers are then tuned to provide the required performance in each case.

This division of control has been selected as it fits in with many of the anticipated missions to be undertaken by the vehicle. A typical mission might see the vehicle performing a survey of an area of the Great Barrier Reef while maintaining a fixed height above the sea floor [12]. The task of performing the survey can then be made independent of maintaining the vehicle altitude. This also allows us to optimize the performance of the depth controller prior to deploying the navigation algorithms used for mapping of the environment to ensure that the vehicle has little risk of hitting the sea floor.

The behaviours that run on the vehicle are also divided into horizontal and vertical behaviours and there are consequently two arbiters currently present. One is responsible for setting the desired depth of the vehicle while the other sets the desired yaw and forward offset to achieve horizontal motion.

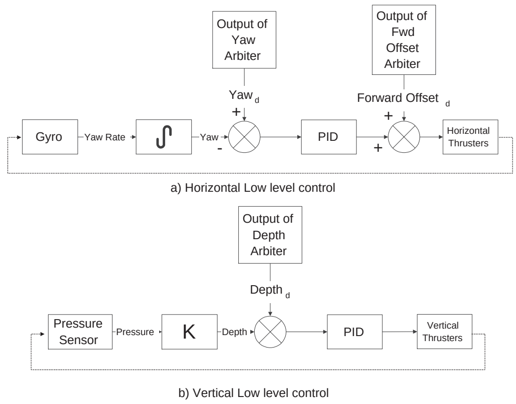


Fig. 3. The low level control processes that run on the embedded controller. These processes include processes for sampling the internal sensor readings, computing the PID control outputs and driving the thrusters.

For a survey mission, the vertical behaviours would typically be responsible for keeping the sub from colliding with the sea floor. The vertical behaviours that run on the vehicle include maintain minimum depth, maintain minimum altitude, maintain depth and maintain altitude. The combination of the outputs of these behaviours determines the depth at which the vehicle will operate. A large negative vote by the maintain minimum altitude behaviour will keep the vehicle at a minimum distance from the sea floor.

The horizontal behaviours that would run on a survey mission include follow line (sonar and/or vision), avoid obstacles, move to location and perform survey. The combination of the outputs of these behaviours determines the orientation maintained by the vehicle as well as the forward offset applied to the two horizontal thrusters. This allows the vehicle to move forward while maintaining its heading.

B. Low and High Level Control

The control of the sub is further decoupled into low-level and high-level control processes. The low-level processes run on the embedded controller and are used to interface directly with the hardware (see figure 3). In addition, the PID controllers have been implemented at this level. This allows the PID controllers to respond quickly to changes in the state of the sub without being affected by the data processing and high-level control algorithms running at the surface.

The high-level processes run on a pair of Pentium machines at the surface control station. Information from the sub's sensors is fed to a series of processes running on these machines. These processes use the data supplied by the sensors to determine the desired set points for the low-level control processes. The raw sensor data is pre-processed to produce virtual sensor information - information that is of interest to multi-

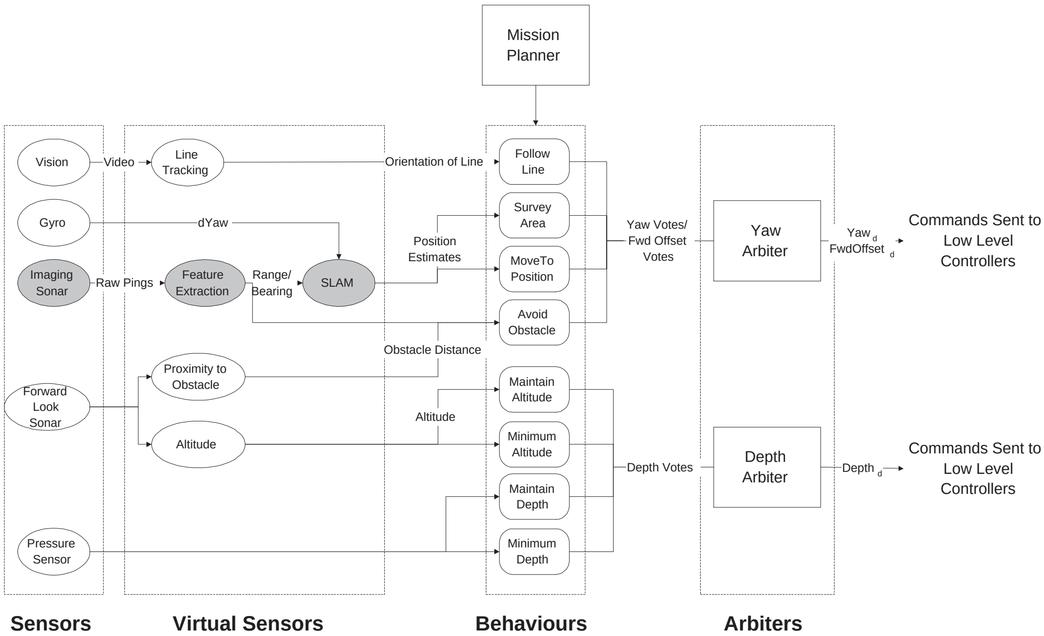


Fig. 4. The high level process and behaviours that run the vehicle. The sensor data is pre-processed to produce virtual sensor information available to the behaviours. The behaviours receive the virtual sensor information and send votes to the arbiters who send control signals to the low level controllers. The greyed out boxes are the SLAM processes that will be detailed in section IV.

ple behaviours in the system. The behaviours register their interest in the data being generated by the virtual sensors and send votes to the arbiters to specify their desired course of action. A task-level mission planner is used to enable and disable behaviours in the system depending on the current state of the mission and its desired objectives. The arbiter combines the votes from the behaviours and selects the optimal action to satisfy the goals of the system. A schematic representation of the data flow within the system is shown in figure 4. This control strategy relies on the ability of the sub to gather information about its environment and reason about its desired actions. By continuously monitoring the state of the environment, the sub is able to respond to changes as they occur.

C. Distributed control

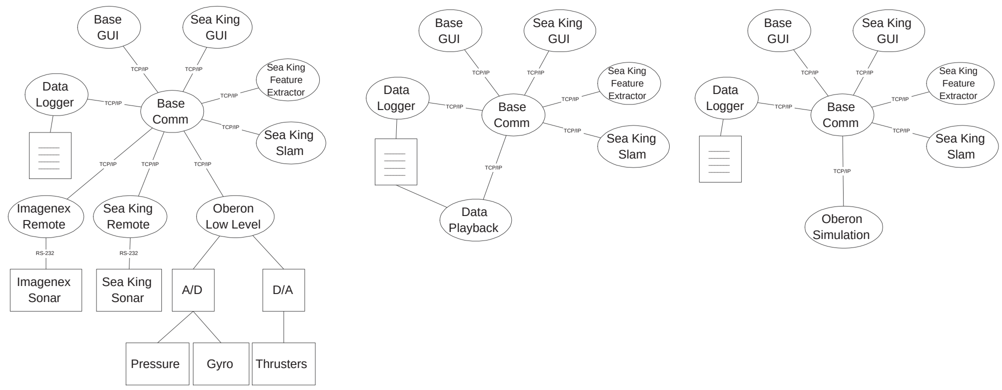
A number of processes have been developed to accomplish the tasks of gathering data from the robot's sensors, processing this data and reasoning about the course of action to be taken by the robot. These processes are distributed across a network of computers and communicate asynchronously via a TCP/IP socket-based interface using a message passing protocol developed at the Centre.

A central communications hub is responsible for routing messages between the distributed processes running on the vehicle and on the command station. Processes register their interest in messages being sent by other processes in the system and the hub routes the messages when they arrive. While this communications structure has some drawbacks, such as potential communications bottlenecks and reliance on the performance of the central hub, it does provide some interesting possibilities for flexible configuration, especially during the development cycle of the system. As shown in Figure 5, the low-level processes that control the vehicle can easily be replaced by a data playback process to replay mission data or by a simulator to develop closed-loop control algorithms prior to vehicle deployment.

IV. Simultaneous Localisation and Map Building(SLAM)

Simultaneous Localisation and Map Building (SLAM) is the process of concurrently building up a feature based map of the environment and using this to obtain estimates of the location of the vehicle [3][4][5][6][11]. The robot typically starts at an unknown location with no a priori knowledge of landmark locations. From relative observations of landmarks, it simultaneously computes an estimate of vehicle location and an estimate of landmark locations. While continuing in motion, the robot builds a complete map of landmarks and uses these to provide continuous estimates of vehicle location.

The localisation and map building process consists of a recursive, three-stage update procedure comprising prediction, observation and update steps using an



(a) Vehicle controller (b) Data Playback (c) Simulation

Fig. 5. a) The low-level processes that control the vehicle can easily be replaced by b) a data playback process to replay mission data or by c) a simulator to develop closed-loop control algorithms prior to vehicle deployment.

Extended Kalman Filter (EKF). The state estimate $\hat{\mathbf{x}}(k)$ is the augmented state vector consisting of the two dimensional pose of the vehicle $\hat{\mathbf{x}}_v(k)$, made up of the position (x_v, y_v) and orientation ψ_v, together with the estimates of the positions of the N landmarks $\hat{\mathbf{x}}_i,\ i = 1...N$ [4]. Point landmarks are currently tracked in the map although the state formulation of the estimates can accommodate higher order features such as lines, curves and polylines.

V. Results

This section describes results from the deployment of the vehicle during testing in a natural terrain environment along Sydney's coast. We show that the control architecture appears to be working effectively by showing the altitude versus depth plots over a particular run. We also show some of the results achieved using the Simultaneous Localisation and Map Building techniques. As can be seen, the vehicle is able to navigate in a straight line relative to the sonar targets deployed in the field.

A. Altitude/Depth control

By coupling the depth controller with the depth sensor, which is continuously sampling the depth, the sub can quickly respond to disturbances in its depth. One of the key behaviours envisaged for the vehicle is the ability to maintain a constant altitude above the sea floor. The Imagenex sonar that currently provides altitude estimates operates at a much slower rate than the depth sensor (on the order of 1 Hz) and would be unable to supply the information at a rate sufficiently high to reject disturbances in depth. The Imagenex sonar is also shared by both the task that maintains the altitude of the sub and by tasks interested in obstacle avoidance. Since the resources of any robot will be limited, it is important to share these resources as necessary. A task scheduler has been implemented that allows resources to be shared between various tasks based on their relative priority.

Figure 6 shows a typical example of the performance of the system. The maintain altitude behaviour is voting for an altitude of 1.5m. The Imagenex sonar returns are being processed to produce an estimate of the altitude of the vehicle. This altitude data is passed to the arbiter which takes it into account when determining the desired depth. The difference between the current altitude and the desired altitude represents the desired change in depth necessary to maintain the correct altitude. As can be seen the vehicle successfully maintains its desired altitude with a standard deviation of 0.2m despite a changing sea floor profile over the course of the run.

B. Localisation and Map Building

We have also tested the SLAM algorithms during deployment in a natural environment off the coast of Sydney. The sub was deployed in a natural inlet with the sonar targets positioned in a straight line in intervals of 10m. The sub's yaw controller maintained a constant heading while the vehicle drove parallel to the line. Since there is currently no absolute position sensor on the vehicle, we cannot measure the performance of the positioning filter against ground truth at this time.

The plot of the final map shown in Figure 7 clearly shows the position of the sonar feature targets along with a number of tentative targets. Some of the tentative targets are from the reef wall while others come

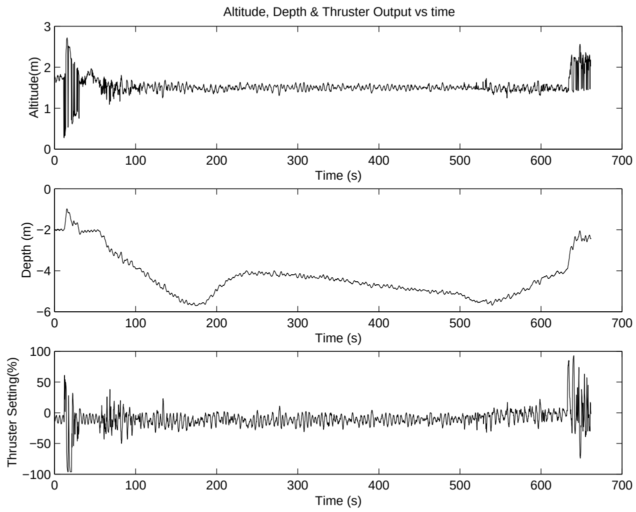


Fig. 6. Results showing the measured altitude versus depth. The desired altitude has been set to 1.5m and the vehicle is able to maintain this altitude with standard deviation of 0.2m over the entire run despite a marked change in the bottom profile. The mean vertical thruster value of -10% downward thrust is due to the positive buoyancy of the vehicle.

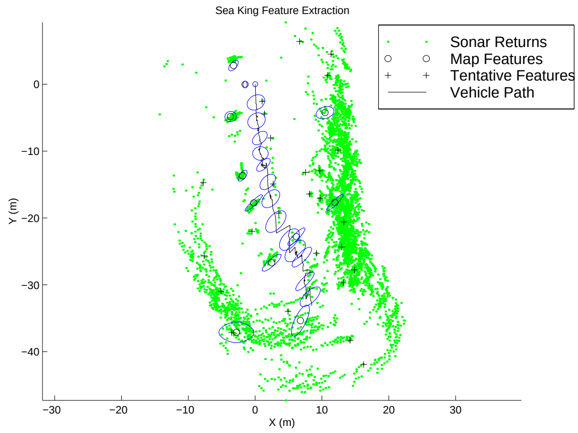


Fig. 7. Path of robot shown against final map of the environment. The estimated position of the features are shown as circles with the covariance ellipses showing their 95% confidence bounds. Tentative targets that have not yet been added to the map are shown as '+'. The series of tentative targets to the right of the image occur from the reef wall. The natural point features tend not to be very stable and are thus not incorporated into the map.

from returns off of the tether. These returns are typically not very stable and therefore do not get incorporated into the SLAM map. The sonar principal returns have been plotted relative to the estimated position of the vehicle. The reef wall to the right of the vehicle and the end of the inlet are clearly visible.

VI. Summary and Conclusions

We have been testing the Oberon system in a number of locations. At present, we have shown that the decoupled, distributed control architecture proposed here is practically feasible during deployment in a natural terrain environment on Sydney's shore-line. These control schemes have proven to be effective in controlling the motion of the sub. It appears that more complicated control schemes relying on complex models of the sub are not necessary in the context of this problem.

The focus of recent work has been on the development of robust terrain-aided navigation. We are investigating criteria for feature selection as a function of the geometric distribution of features in an environment as well as based on the current mission being run. Incorporation of the SLAM architecture into the mission planning loop will allow decisions concerning sensing strategies to be made in light of the desired mission objectives.

References

[1] Bauer R, Rencken WD. "Sonar feature based exploration". *1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*.pp.148-53 vol.1. Los Alamitos, CA, USA, 1995

[2] Brooks, RA. "A robust, layered control system for a mobile robot". *IEEE Journal of Robotics and Automation*, vol.RA-2, no.1 , pp 14-23 April, 1986

[3] Castellanos JA, Montiel JMM, Neira J, Tardos JD. "Sensor Influence in the Performance of Simultaneous Mobile Robot Localization and Map Building". *6th International Symposium on Experimental Robotics*. pp 203-212, Sydney, NSW, Australia, 1999

[4] Dissanayake MWMG, Newman P, Durrant-Whyte HF, Clark S, Csorba M. "An Experimental and Theoretical Investigation into Simultaneous Localisation and Map Building". *6th International Symposium on Experimental Robotics*. pp 171-180, Sydney, NSW, Australia, 1999

[5] Feder HJS, Leonard JJ, Smith CM. "Adaptive sensing for terrain aided navigation". *IEEE Oceanic Engineering Society. OCEANS'98*. pp.336-41 vol.1. New York, NY, USA, 1998

[6] Leonard JJ, Durrant-Whyte HF, "Simultaneous Map Building and Localisation for an Autonomous Mobile Robot", *IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '91*, pp.1442-1447, New York, NY, USA, 1991

[7] Leonard JJ, Carpenter RN, Feder HJS, "Stochastic Mapping Using Forward Look Sonar", *FSR '99 International Conference on Field and Service Robotics* , pp.69-74, Pittsburg, PA, USA 1999

[8] Newman P. and Durrant-Whyte HF, "Toward Terrain-Aided Navigation of a Subsea Vehicle", *FSR'97 International Conference on Field and Service Robotics*, pp.244-248, Canberra, Australia, 1997.

[9] Newman P., "On The Structure and Solution of the Simultaneous Localisation and Map Building Problem", *D.Phil Thesis*, University of Sydney, 1999

[10] Rosenblatt, J. "The Distributed Architecture for Mobile Navigation". *Journal of Experimental and Theoretical Artificial Intelligence*, vol.9 no.2/3 pp.339-360, April-September, 1997

[11] Rencken WD. "Concurrent localisation and map building for mobile robots using ultrasonic sensors". *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2192-7 vol.3. New York, NY, USA, 1993

[12] Williams S, Newman P, Majumder S, Rosenblatt J, Durrant-Whyte H, "Autonomous Transect Surveying of the Great Barrier Reef", *Australian Conference on Robotics and Automation (ACRA'99)*, Brisbane, QLD, 1999.