Robotica (2001) volume 19, pp. 481–496. Printed in the United Kingdom © 2001 Cambridge University Press

Autonomous underwater navigation and control

Stefan B. Williams, Paul Newman, Julio Rosenblatt, Gamini Dissanayake and Hugh Durrant-Whyte

Australian Centre for Field Robotics, Department of Mechanical and Mechatronic Engineering, University of Sydney, NSW 2006 (Australia)

(Received in Final Form: December 20, 2000)

SUMMARY

This paper describes the autonomous navigation and control of an undersea vehicle using a vehicle control architecture based on the Distributed Architeclure for Mobile Navigation and a terrain-aided navigation technique based on simultaneous localisation and map building. Development of the low-speed platform models for vehicle control and the theoretical and practical details of mapping and position estimation using sonar are provided. Details of an implementation of these techniques on a small submersible vehicle "Oberon" are presented.

KEYWORDS: Underwater navigation; Autonomous control; Distributed architecture; Sonar use

1. 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. Position and attitude estimation algorithms that use information from scanning sonar to complement a vehicle dynamic model and unobservable environmental disturbances are invaluable in the subsea environment. Key elements of the current research 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 for vehicle control.

One of the key technologies being developed in the context of this work is an algorithm for Simultaneous Localisation and Map Building (SLAM) to estimate the position of an underwater vehicle. SLAM is the process of concurrently building up a feature based map of the environment and using this map to obtain estimates of the location of the vehicle.[1–5] 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 this to provide continuous estimates of the vehicle location. The potential for this type of navigation system for subsea robots is enormous considering the difficulties involved in localisation in underwater environments. Maps of the terrain in which the vehicle will be operating do not typically exist prior to deployment and GPS is not available at the depths at which these vehicles operate.[6]

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 its 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. A behaviour-based control architecture, based on the Distributed Architecture for Mobile Navigation (DAMN),[7] is used for control of the AUV.[8]

This paper presents the vehicle control architecture currently used on the Oberon submersible followed by results of the application of a Simultaneous Localisation and Map building algorithm to estimate the motion of the vehicle. This work represents the first instance of a deployable underwater implementation of the SLAM algorithm. Section 2 introduces the Oberon submersible vehicle developed at the Centre and describes the sensors and actuators used. Section 3 describes the distributed, decoupled control scheme currently operating on the vehicle while Section 4 summarizes the stochastic mapping algorithm used for SLAM together with the feature extraction and data association techniques used to generate the observations for the SLAM algorithm. In Section 5 a series of trials are described and the results of applying SLAM during field trials in a pool at the University of Sydney as well as in a natural terrain environment along Sydney's coast are presented. Finally, Section 6 concludes the paper by summarizing the results and discussing future research topics as well as on-going work.

2. THE OBERON VEHICLE

The experimental platform used for the work reported in this paper is 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.[9] This device is intended primarily as a research platform upon which to test novel sensing strategies and control methods. Autonomous navigation using the infor-

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

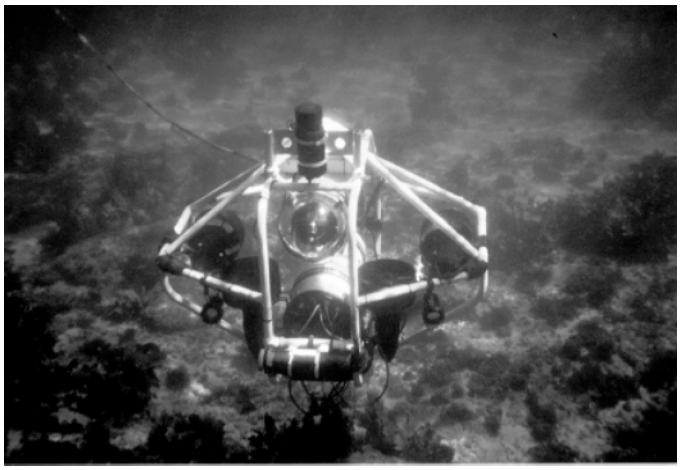


Fig. 1. Oberon at Sea

mation provided by the vehicle's on-board sensors represents one of the ultimate goals of the project.[10]

2.1. 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 development 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.

The sonar 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 submersible. Communications between the computers at the surface and the submersible 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 water is detected inside the pressure hulls using a pair of optical diodes.

2.2. Sonar

Sonar is the primary sensor of interest on the Oberon vehicle. There are currently two sonars on the robot. A Tritech SeaKing imaging sonar has a dual frequency narrow beam sonar head that is mounted on top of the submersible and is used to scan the environment in which the submersible is operating. It can achieve 360° scan rates on the order of 0.25 Hz using a pencil beam with a beam angle of 1.8°. This narrow beam allows the sonar to accurately discriminate bearing returns to objects in the environment. It has a variable mechanical step size capable of positioning the sonar head to within 0.5° and can achieve range resolution on the order of 50 mm depending on the selected scanning range. It has an effective range to 300 m allowing for long range target aquisition in the low frequency mode but can also be used for high definition scanning at lower ranges. The information returned from this sonar is used to build and maintain a feature map of the environment.

The second sonar is an Imagenex sonar unit operating at 675 kHz and 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 (see Figure 3). This enables the altitude above the sea floor as well as the proximity of obstacles to be determined using the wide angle beann of the sonar. The Imagenex sonar features a

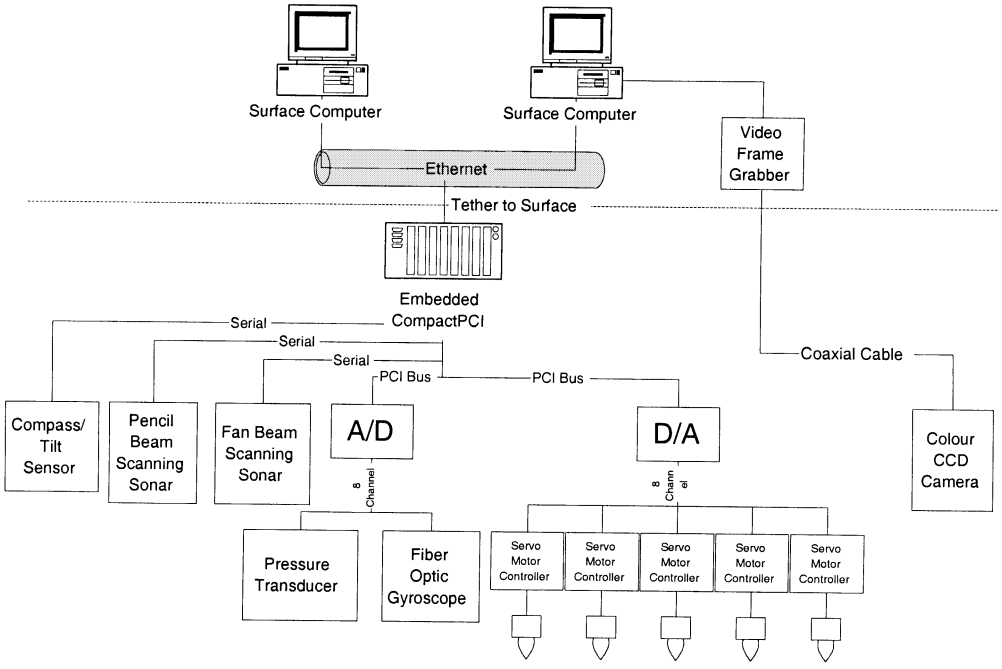
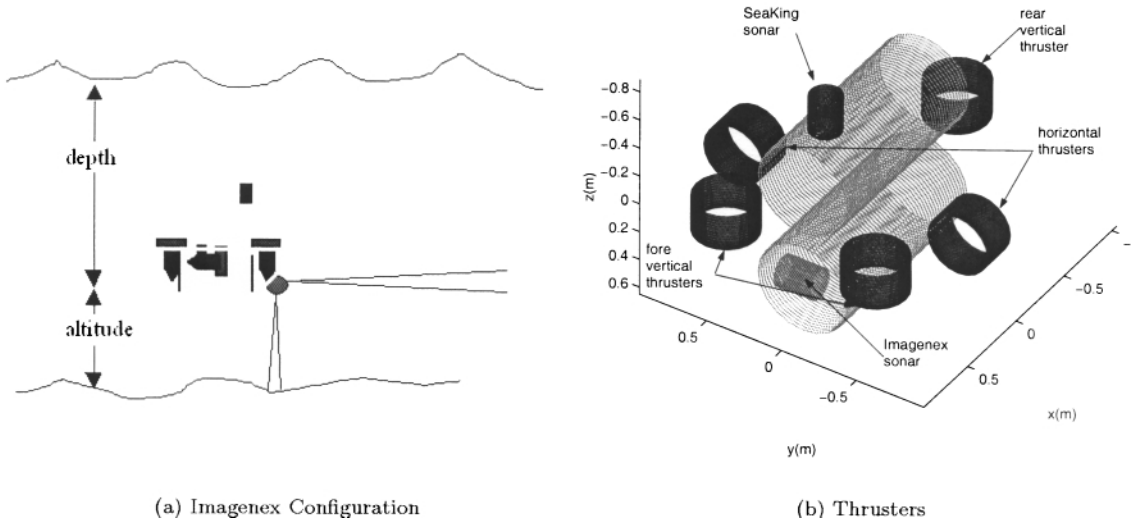


Fig. 2. Vehicle System Diagram

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press



(a) Imagenex Configuration (b) Thrusters

Fig. 3. Vehicle Configuration

beam width of $15° \times 1.8°$ allowing a broad swath to be insonified with each burst of acoustic energy ("ping") – an ideal sensor for altitude estimation and obstacle avoidance.

2.3. Internal sensors

An Andrews Fiber Optic Gyro has been included in the Oberon robot to allow the robot's orientation to be determined. This sensor provides the yaw rate and is used to control the heading of the submersible. The bias in the gyroscope is first estimated while the vehicle is stationery. The bias compensated yaw rate is then integrated to provide an estimate of vehicle heading. Because the yaw rate signal is noisy, the integration of this signal causes the estimated heading to drift with time. In addition, the bias can drift as the unit's internal temperature changes. Temperature compensation can help to overcome some of these problems and will be considered in the future. At present, missions do not typically run for longer than 30 minutes and yaw drift does not pose a significant problem Subsequent to the data collection presented here, an integrated compass and tilt sensor has been added to the vehicle. The compass signal is filtered with the output of the gyroscope to estimate the yaw rate bias of the gyroscope on-line. This will allow the vehicle to undertake longer missions than were previously feasible. The Simultaneous localisation and Mapping algorithm to be presented later also allows the yaw rate bias to be estimated by using tracked features in the environment to provide corrections to errors in the estimated yaw.

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. Feedback from this sensor is used to control the depth of the submersible.

2.4. Camera

A colour video 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. The video signal is transmitted to the surface via the tether. A Matrox Meteor frame grabber is then used to acquire the video signal for further image processing.

2.5. 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 (see Figure 3(b)). This gives the vehicle the ability to move itself up and down, control its yaw, pitch and roll and move forwards and backwards. This thruster configuration does not allow the vehicle to move sideways but this does not pose a problem for the missions envisaged for this vehicle.

3. 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. This section describes the distributed, decoupled control architecture used to help simplify the controller design for this vehicle.

3.1. Low level control

The dynamics of the Oberon vehicle are such that the vertical motion of the vehicle is largely decoupled from the lateral motion. The vehicle is very stable in the roll and pitch axes due to the large righting moment induced by the vertical configuration of the pressure vessels. A steel keel provides an added moment to maintain the vehicle in an upright pose. Two independant PID controllers can therefore be used to control horizontal and vertical motion of the vehicle. This greatly simplifies the individual controller design. Furthermore, this particular division of control fits in with many of the anticipated missions to be undertaken by the vehicle. For example, one of the target missions is to use Oberon to survey an area of the Great Barrier Reef while maintaining a fixed height above the sea floor.[11] The surveying task can then be made independent of maintaining the vehicle altitude.

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

The low-level processes run on the embedded controller and are used to interface directly with the hardware (see Figure 4). This allows the controllers to respond quickly to changes in the state of the submersible without being affected by delays due the data processing and high-level control algorithms running on the remote computers. Set points to the low-level controllers are provided by the behaviours and high-level controllers described in the next section.

3.2. High-level controller

The high-level controllers are based on the Distributed Architecture for Mobile Navigation (DAMN).[7] DAMN consists of a group of distributed behaviours sending votes for desirable actions and against objectionable ones to a centralized command arbiter, which combines these votes to generate actions. The arbiter then provides set-points to the low-level controller such that the desired motion is achieved.

Within the framework of DAMN, behaviours must be defined to provide the task-specific knowledge for the domain. These behaviours operate independently and asynchronously, and each encapsulates the perception, planning and task execution capabilities necessary to achieve one specific aspect of robot control, and receives only the data specifically required for that task.[12]

The raw sensor data is preprocessed to produce information that is of interest to multiple behaviours designed to control the vehicle's actions. These preprocessors act as virtual sensors by providing data for the behaviours that is abstracted from the raw sensor data, simplifying the individual behaviour design. The behaviours monitor the outputs of the relevant virtual sensors and determine the optimal action to achieve the behaviour's objectives. The behaviours send votes to the arbiters which combine the votes in the system to determine the action which will allow the system to best achieve its goals. 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. A command arbitration process combines the votes from the behaviours and selects the optimal action to satisfy the goals of the system.

In the present implementation, two arbiters are used to provide set-points to the low-level controllers. One arbiter is responsible for setting the desired depth of the vehicle while the other sets the desired yaw and forward offset to achieve horizontal motion. The behaviours are consequently divided into horizontal and vertical behaviours.

For a survey mission, the vertical behaviours are responsible for keeping the submersible from colliding with the sea floor. The vertical behaviours that run on the vehicle include maintain minimal 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, for example, will keep the vehicle at a minimum distance from the sea fioor.

The horizontal behaviours that run during a typical survey mission include follow line (using sonar and/or vision), avoid obstacles, move to a 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.

A schematic repsentation of the control structure of the Oberon vehicle is shown in Figure 5. The vertical behaviours rely primarily on the depth sensor whereas the horizontal behaviours use vision, gyro and the imaging sonar (Tritech SeaKing). The forward look sonar (Imagenex) is shared between both horizontal and vertical behaviours. It is used to provide periodic altitude measurements for the maintain altitude behaviour while providing

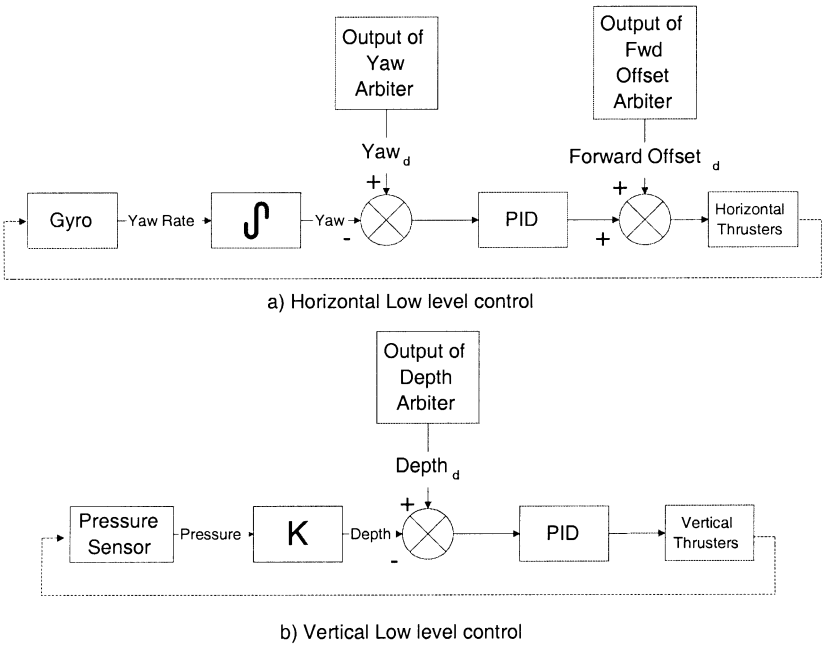


Fig. 4. Low-level processes

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

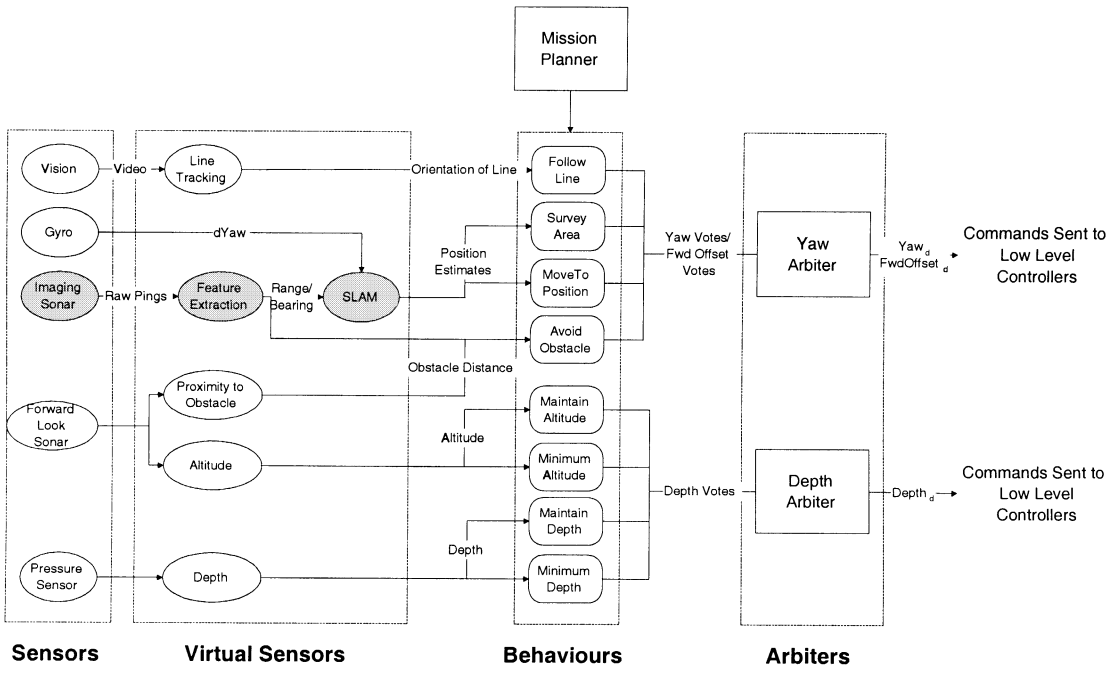


Fig. 5. High-level processes

an indication as to the presence of obstacles in front of the vehicle to the obstacle avoidance behaviour. A task scheduler that allow resources to be shared between various tasks is used to allocate this sensor to the two behaviours. Clearly, tasks that are deemed more important to the accomplishment of the vehicle's mission need to be given preferential access to the resources they require. For example, the altitude task is allowed to ping the bottom in preference to the avoid obstacles behaviour since the AUV typically operates at relatively low speeds and collision with the bottom is more of a concern than collision with obstacles in front of the vehicle.

3.3. Distributed control

The control structure described in the previous sections is implemented using a distributed control strategy. 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. In the context of the low information rates present in an underwater vehicle, the system performs well. The implementation also allow for easy system development. For example, as shown in Figure 6, 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. This enables the development of closed-loop control algorithms prior to vehicle deployment.

The control architecture presented here provides a distributed environment in which to control the robot. Processes can be developed and added to the system without major changes to the overall architecture. New mission-dependent behaviours can be introduced without requiring changes to the rest of the controller. The communications have also been abstracted away from the operation of the various processes. This provides the potential to change the inter-process communications medium without necessitating major changes to the behaviours themselves.

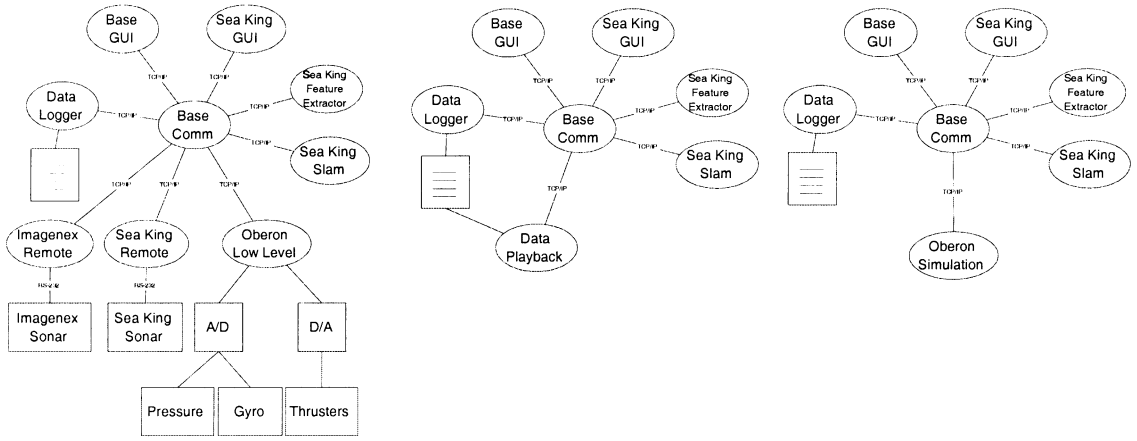
4. POSITION ESTIMATION

Many of the behaviours in the system rely on the ability of the vehicle to estimate its position. Behaviours such as move to position and avoid obstacles need to have a reliable estimate of the vehicle position in order to function correctly. While many land-based robots use GPS or maps of the environment to provide accurate position updates, a robot navigating underwater does not have access to this type of information. In typical underwater scientific missions, a priori maps are seldom available.[13] This section presents the feature based localisation and mapping technique used for generating vehicle position estimates.

4.1. The estimation process

The localisation and map building process consists of a recursive, three-stage procedure comprising prediction, observation and update steps using an Extended Kalman Filter (EKF).[2] The EKF estimates the two dimensional pose

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press



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

Fig. 6. Distributed control

of the vehicle $\hat{\mathbf{x}}_v$, made up of the position (x_v, y_v) and orientation ψ_v, together with the estimates of the positions of the N landmarks $\hat{\mathbf{x}}_i$, $i=1 \ldots N$, using the observations from the sensors on board the submersible.

4.1.1. Prediction. The prediction stage uses a model of the motion of the vehicle to compute the vehicle position at instant k using the information up to instant $k-1$, $\hat{\mathbf{x}}_v(k|k-1)$. A simple constant velocity model shown in equation 2 is used for this purpose. The vehicle velocity V is assumed to be proportional to the mean lateral thruster setting. Given the small submerged inertia, relatively slow motion and large drag-coefficients induced by the open frame structure of the vehicle, this is a reasonable vehicle model.

$$\hat{\mathbf{x}}_v(k|k-1)=\mathbf{F}_v(\hat{\mathbf{x}}_v(k-1|k-1), \mathbf{u}_v(k-1|k-1)) \quad (1)$$

where $\mathbf{F}_v$ is defined by

$$\hat{x}_v(k|k-1)=\hat{x}_v(k-1|k-1)+V\Delta T\cos(\hat{\psi}(k-1|k-1))$$
$$\hat{y}_v(k|k-1)=\hat{y}_v(k-1|k-1)+V\Delta T\sin(\hat{\psi}(k-1|k-1))$$
$$\hat{\psi}_v(k|k-1)=\hat{\psi}_v(k-1|k-1)+\Delta\psi\Delta T$$

The covariance of the vehicle and feature states, $\hat{\mathbf{P}}(k|k)$, are predicted using the non-linear state prediction equation. The predicted covariance is computed using the gradient of the state propagation equation, $\nabla\mathbf{F}_v$, linearised about the current estimate, the process noise model, $\mathbf{Q}$, and the control noise model, $\mathbf{U}$. The filter parameters used in this application are shown in Table I.

Table I. SLAM filter parameters.

Sampling period	ΔT	0.1 s
Vehicle X process noise std dev	σ_x	0.075 m
Vehicle Y process noise std dev	σ_y	0.075 m
Vehicle heading process noise std dev	σ_ψ	0.5°
Vehicle velocity std dev	σ_v	0.75 m/s
Vehicle steering std dev	$\sigma_{d\psi}$	0.5°
Gyro/Compass sid dev	σ_c	0.5°
Range measurement std dev	σ_R	0.5 m
Bearing measurement std dev	σ_B	2.0°
Sonar range		15 m
Sonar resolution		0.075 m

$$\hat{\mathbf{P}}(k|k-1)=\nabla\mathbf{F}_v\hat{\mathbf{P}}(k-1|k-1)\nabla\mathbf{F}_v^T + \nabla\mathbf{F}_v\mathbf{U}(k|k-1)\nabla\mathbf{F}_v^T+\mathbf{Q}(k|k-1) \quad ((2)$$

with

$$\mathbf{Q}(k|k-1)=diag\ [\sigma_x^2\ \sigma_y^2\ \sigma_\psi^2] \quad (3)$$

and

$$\mathbf{U}(k|k-1)=diag\ [\sigma_v^2\ \sigma_{d\psi}^2] \quad (4)$$

4.1.2. Observation. Observations are made using an imaging sonar that scans the horizontal plane around the vehicle. Point features are extracted from the sonar scans and are matched against existing features in the map. The feature matching algorithm will be described in more detail in Section 4.2. The observation consists of a relative distance and orientation from the vehicle to the feature. The predicted observation, $\hat{\mathbf{z}}_i(k|k-1)$, when observing landmark "i" located at $\hat{\mathbf{x}}_i$ can be computed using the non-linear observation model $\mathbf{H}_i(\hat{\mathbf{x}}_v(k|k-1), \hat{\mathbf{x}}_i(k|k-1))$.

$$\hat{\mathbf{z}}_i(k|k-1)=\mathbf{H}_i(\hat{\mathbf{x}}_v(k|k-1), \hat{\mathbf{x}}_i(k|k-1)) \quad (5)$$

where $\mathbf{H}_i$ is defined by

$$\hat{z}_{iR}(k|k-1)= \sqrt{(\hat{x}_v(k|k-1)-\hat{x}_i(k|k-1))^2+(\hat{y}_v(k|k-1)-\hat{y}_i(k|k-1))^2}$$

$$\hat{z}_{i\theta}(k|k-1)= \arctan\left(\frac{(\hat{y}_v(k|k-1)-\hat{y}_i(k|k-1))}{(\hat{x}_v(k|k-1)-\hat{x}_i(k|k-1))}\right)-\hat{\psi}_v(k|k-1) \quad (6)$$

The difference between the actual observation $\mathbf{z}(k|k-1)$ and the predicted observation $\hat{\mathbf{z}}(k|k-1)$ is termed the innovation $\nu(k|k-1)$.

$$\nu_i(k|k-1)=\mathbf{z}_i(k|k-1)-\hat{\mathbf{z}}_i(k|k-1) \quad (7)$$

The innovation covariance $\mathbf{S}(k|k-1)$ is computed using the current state covariance estimate $\hat{\mathbf{P}}(k|k-1)$, the gradient of the observation model, $\nabla\mathbf{H}(k|k-1)$ and the covariance of the observation model $\mathbf{R}(k|k-1)$.

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

$$\mathbf{S}(k|k-1)=\nabla\mathbf{H}(k|k-1)\hat{\mathbf{P}}(k|k-1)\nabla\mathbf{H}(k|k-1)^T + \mathbf{R}(k|k-1) \quad (8)$$

with

$$\mathbf{R}(k|k-1)=diag\ [\sigma_R^2\ \sigma_B^2] \quad (9)$$

4.1.3. Update. The state estimate can now be updated using the optimal gain matrix $\mathbf{W}(k)$. This gain matrix provides a weighted sum of the prediction and observation and is computed using the innovation covariance, $\mathbf{S}(k|k-1)$ and the predicted state covariance, $\hat{\mathbf{P}}(k|k-1)$. This is used to compute the state update $\hat{\mathbf{x}}(k|k)$ as well as the updated state covariance $\hat{\mathbf{P}}(k|k)$.

$$\hat{\mathbf{x}}(k|k)=\hat{\mathbf{x}}(k|k-1)+\mathbf{W}(k|k-1)\nu(k|k-1) \quad (10)$$

$$\hat{\mathbf{P}}(k|k)=\hat{\mathbf{P}}(k|k-1)-\mathbf{W}(k|k-1)\mathbf{S}(k|k-1)\mathbf{W}(k|k-1)^T \quad (11)$$

where

$$\mathbf{W}(k|k-1)=\hat{\mathbf{P}}(k|k-1)\nabla\mathbf{H}(k|k-1)\mathbf{S}^{-1}(k|k-1) \quad (12)$$

4.2. Feature extraction for localisation

The development of autonomous map based navigation relies on the ability of the system to extract appropriate and reliable features with which to build maps.[14,15] Point features are identified from the sonar scans returned by the imaging sonar and are used to build up a map of the environment. The extraction of point features from the sonar data is essentially a three stage process. The range to the principal return must first be identified in individual pings. This represents the range to the object that has produced the return. This is complicated by such issues as multiple and/or specular reflections, problems which are seen in the pool but less so in a natural environment. The principal returns must then be grouped into clusters. Small, distinct clusters can be identified as point features and the range and bearing to the target estimated. Finally, the range and bearing information must be matched against existing features in the map.

Sonar targets are currently introduced into the environment in which the AUV will operate (see Figure 7) in order to obtain identifiable and stable features. A prominent portion of the reef wall or a rocky outcropping might also be classified as a point feature. If the naturally occurring point features are stable they will also be incorporated into the map. Development of techniques to extract more complex natural features, such as coral reefs and natural variations on the sea floor, is an area of active research as this will allow the submersible to be deployed in a larger range of natural environments without the need to introduce artificial beacons.

The sonar targets produce strong sonar returns that can be charaterised as point targets for the purposes of mapping (see Figure 8). The lighter sections in the scan indicate stronger intensity returns. As can be seen from the figure, the pool walls act as specular reflectors causing a considerable amount of additional sonar noise as well as multiple reflections that appear to be objects 'behind' the walls.

The three stages of feature extraction are described in more detail in the following subsections

4.2.1. Principal returns. The data returned by the SeaKing sonar consists of the complete time history of each sonar ping in a discrete set of bins scaled over the desired range. The first task in extracting reliable features is to identify the principal return from the ping data. The principal return is considered to be the start of the maximum energy component of the signal above a certain noise threshold. Figure 9 shows a single ping taken from a scan in the field. This return is a reflection from one of the sonar targets and the principal return is clearly visible. The return exhibits very good signal to noise ratio making the extraction of the principal returns relatively straightforward.

4.2.2. Identification of point features. The principal returns are then processed to find regions of constant depth

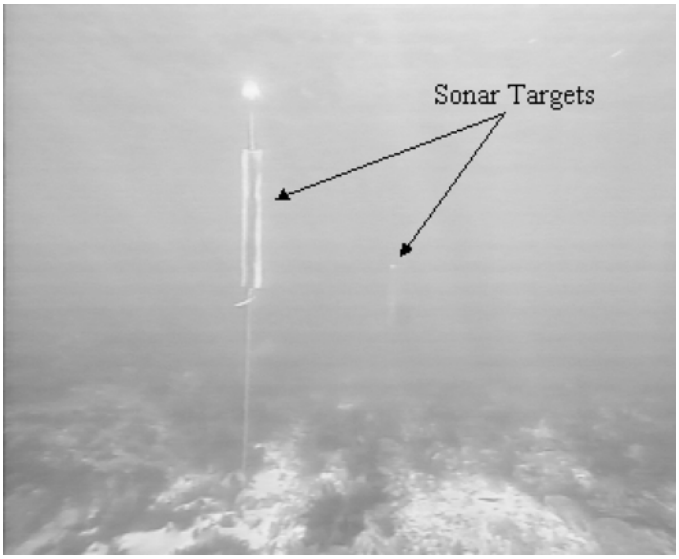


Fig. 7. Sonar targets

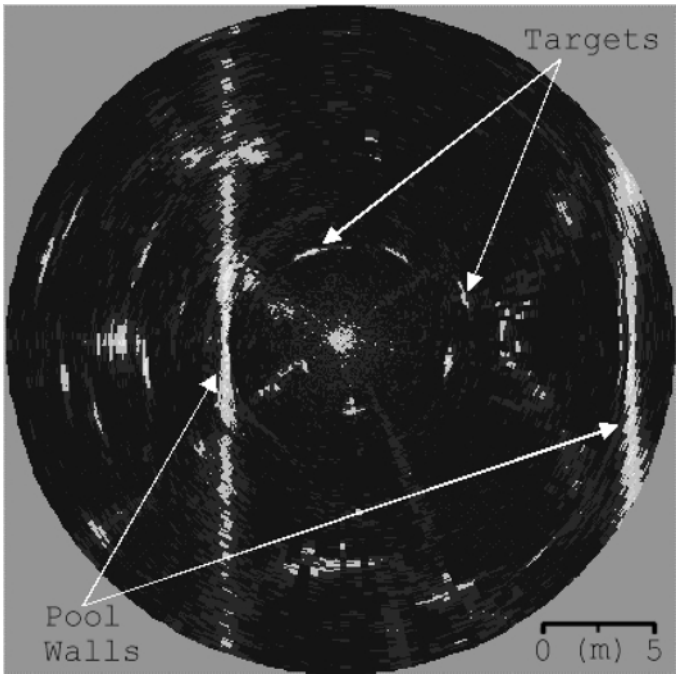


Fig. 8. Sonar scan of pool

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

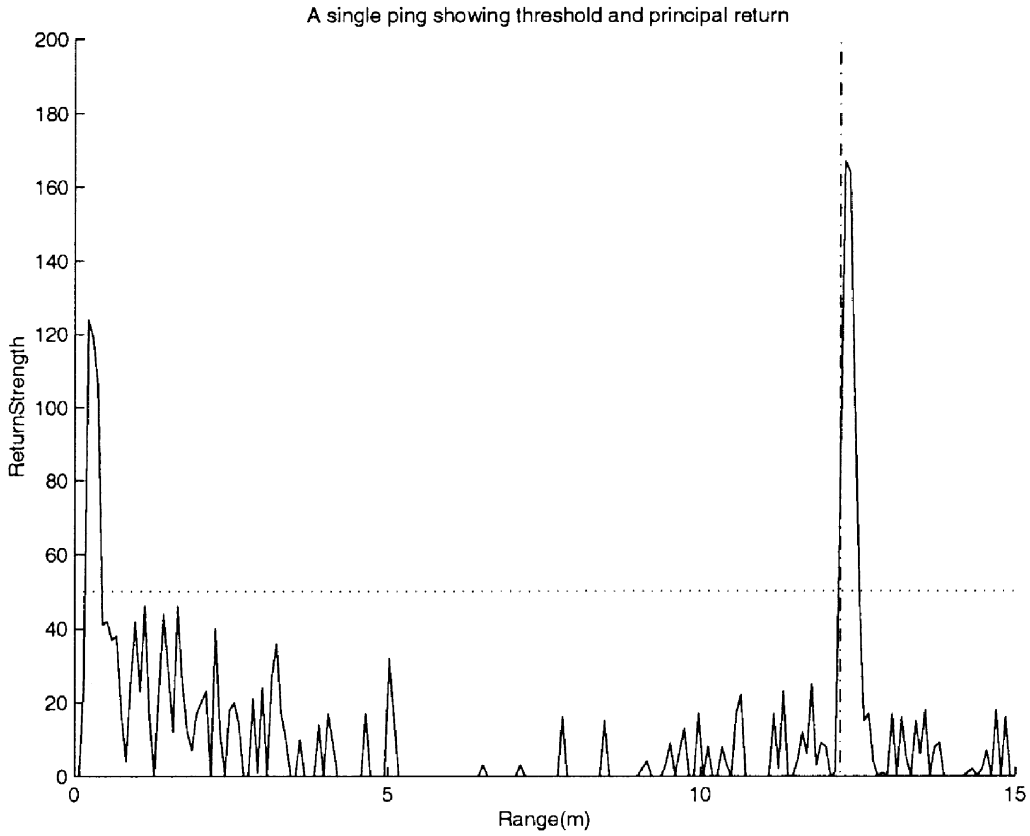
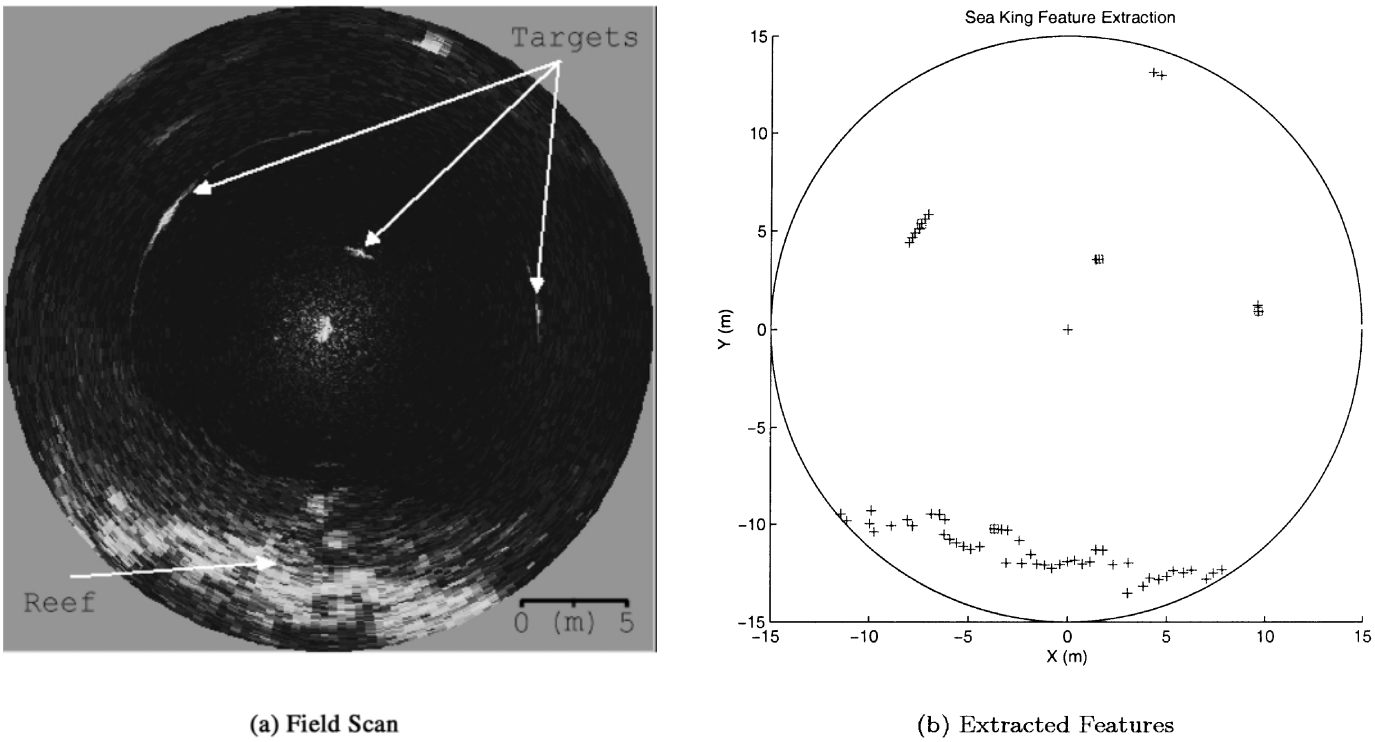


Fig. 9. Single sonar ping

within the scan that can be classified as point features. Sections of the scan are examined to find consecutive pings from which consistent principal return ranges are located. The principal returns are classified as a point feature if the width of the cluster is small enough to be characterised as a point feature and the region is spatially distinct with respect to other returns in the scan.[16] The bearing to the feature is computed using the centre of the distribution of principal returns. The range is taken to be the median range of the selected principal returns.

A scan taken in the field is shown in Figure 10(a). Three targets are clearly visible in the scan along with a section of



(a) Field Scan (b) Extracted Features

Fig. 10. Sonar feature extraction

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

the reef wall. Figure 10(b) shows the principal returns selected from the scan along with the point features extracted by the algorithm. All three targets are correctly classified as point features. A prominent portion of the reef wall is also classified as a point feature.

4.2.3. Feature matching. Once a point feature has been extracted from a scan, it must be matched against known targets in the environment. A two-step matching algorithm is used in order to reduce the number of targets that are added to the map (see Figure 11).

When a new range and bearing observation is received from the feature extraction process, the estimated position of the feature is computed using the current estimate of vehicle position. This position is then compared with the estimated positions of the features in the map using the Mahanabolis distance.[2] If the observation can be associated to a single feature the EKF is used to generate a new state estimate. An observation that can be associated with multiple targets is rejected since false observations can destroy the integrity of the estimation process.

If the observation does not match to any targets in the current map, it is compared against a list of tentative targets. Each tentative target maintains a counter indicating the number of associations that have been made with the feature as well as the last observed position of the feature. If a match is made, the counter is incremented and the observed position is updated. When the counter passes a threshold value, the feature is considered to be sufficiently stable and is added to the map. If the potential feature cannot be associated with any of the tentative features, a new tentative feature is added to the list. Tentative features that are not reobserved are removed from the list after a fixed time interval has elapsed.

5. Experimental results

Oberon was deployed in a number of environments including the pool at the University of Sydney and in a natural terrain environment along Sydney's coast, in order to evaluate the control and mapping techniques developed in the previous sections. This section presents the results obtained.

5.1. Decoupled control

This section describes the performance of the vertical control behaviours. These behaviours allow the vehicle to navigate without risk of hitting the sea floor.

5.1.1. Altitude/depth control. One of the key behaviours envisaged for the vehicle is the ability to maintain a constant altitude above the sea floor. Figure 12 shows a typical

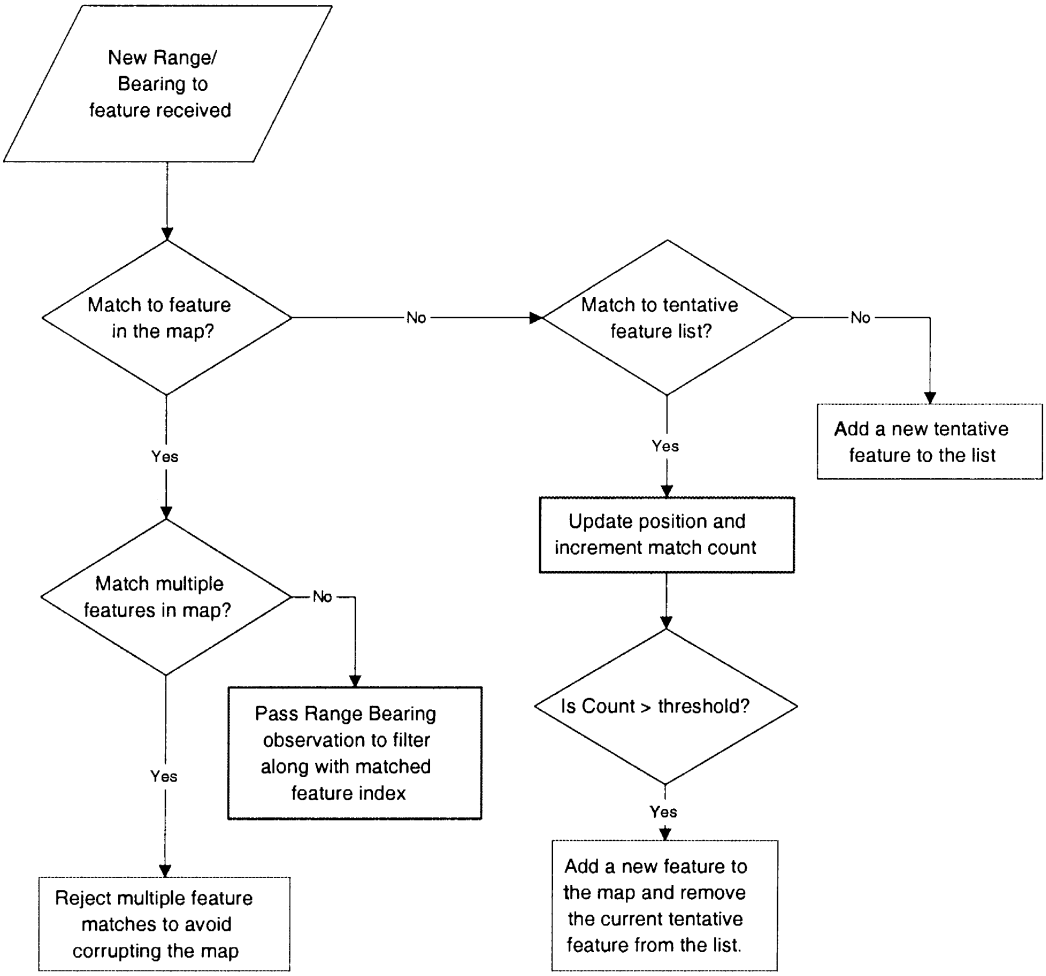


Fig. 11. The feature matching algorithm

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

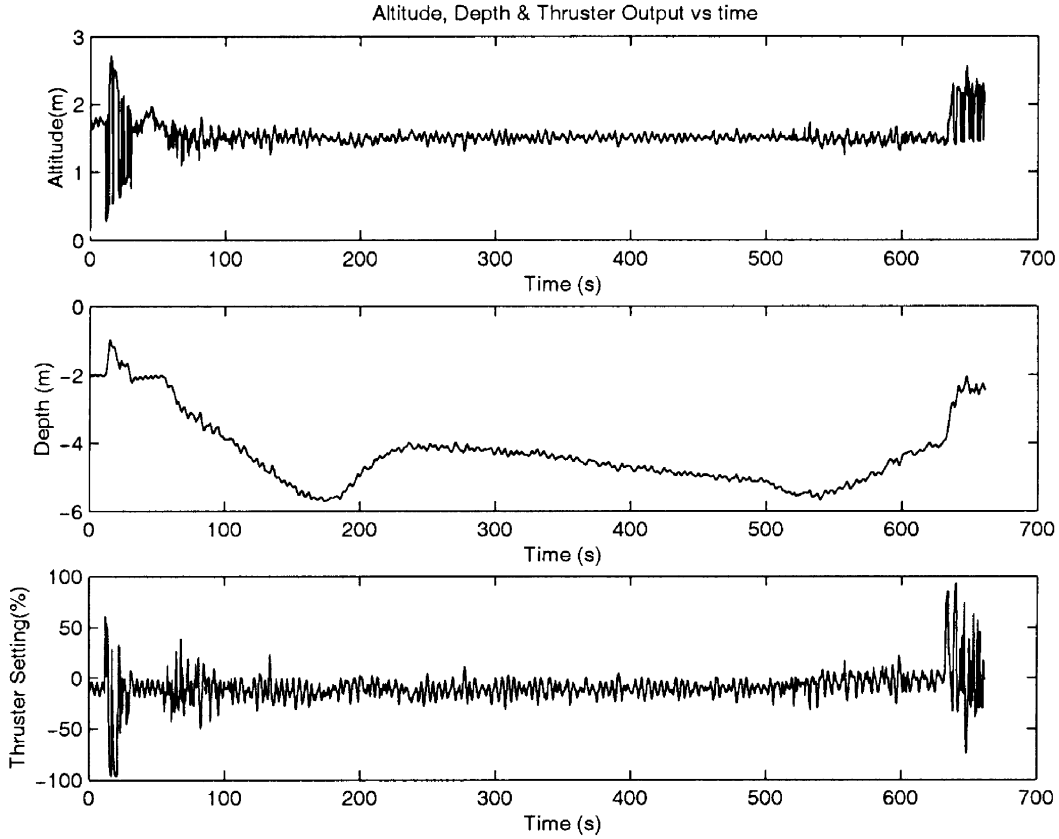


Fig. 12. Altitude versus depth

example of the performance of the system. The maintain altitude behaviour is voting for an altitude of 1.5 m. The Imagenex sonar returns are used to produce an estimate of the actual altitude of the vehicle. This altitude 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 in Figure 12 the vehicle successfully maintains its desired altitude with a standard deviation of 0.2 m despite a changing sea floor profile over the course of the run.

5.2. Localisation results

This section shows the results of the deployment of the localisation algorithm. This algorithm was evaluated in the pool and in a natural terrain environment. It is demonstrated that the simultaneous map building and localisation algorithm provides consistent estimates of the vehicle location.

5.2.1. Swimming pool trials. During tests in the pool at the University of Sydney, six sonar targets were placed in the pool such that these form a circle roughly 6 m in diameter in the area in which the submersible is operating. The vehicle starts to the left of the target cluster and is driven in a straight line perpendicular to the pool walls over a distance of approximately 5 m. The submersible is then driven backwards over the same distance and returns to the initial position. A PID controller maintains the heading of the submersible using feedback from a fiber optic gyro. During the short duration of the trials, the accumulated drift in the gyro orientation estimate is considered negligible. Forward thrust is achieved by adding a common voltage to each thruster amplifier control signal.[11]

The distance to the pool walls measured with the sonar is used to determine the true position of the submersible in the pool. The walls are clearly evident in the scans taken in the pool giving us a good approximation of the actual position of the submersible.

Figure 13 shows the position estimate along the axis perpendicular to one of the pool walls generated by SLAM during one of the runs in the pool. These results show that the robot is able to successfully determine its position relative to the walls of the pool despite the effect of the tether, which tends to dominate the dynamics of the vehicle at low speeds. With the low thruster setting, the AUV stops after it has traversed approximately 4.5 m, with the catenary created by the deployed tether overcoming the forward thrust created by the vehicle's thrusters. The SLAM algorithm is able to determine the fact that the vehicle has stopped using observations from its feature map.

The error in the perpendicular position relative to the pool walls is plotted along with the 95% confidence bounds in Figure 14. While the estimated error bounds are conservative, this allows the optimal estimator to account for environmental disturbances in the field such as current and the forces exerted by the tether catenary.

In order to test the filter consistency, the innovation sequences can be checked against the innovation covariance estimates. This is the only method available to monitor online filter performance when ground truth is unavailable. As

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

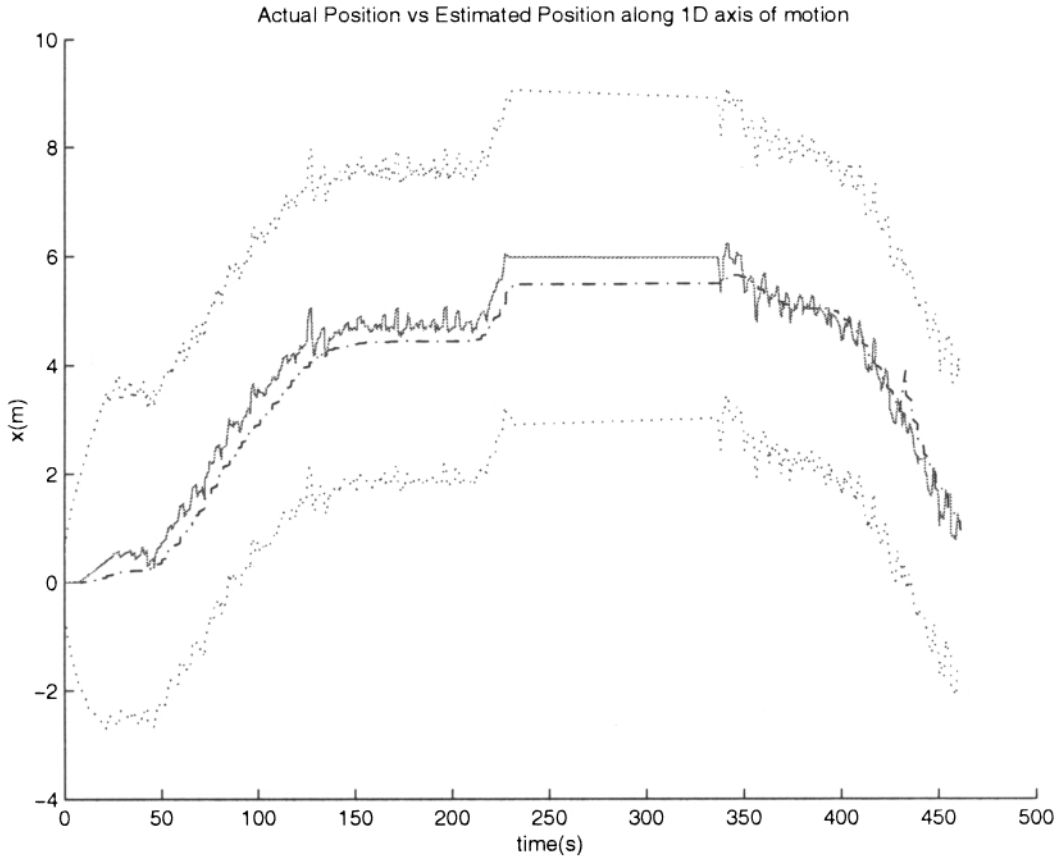


Fig. 13. Position of robot in pool

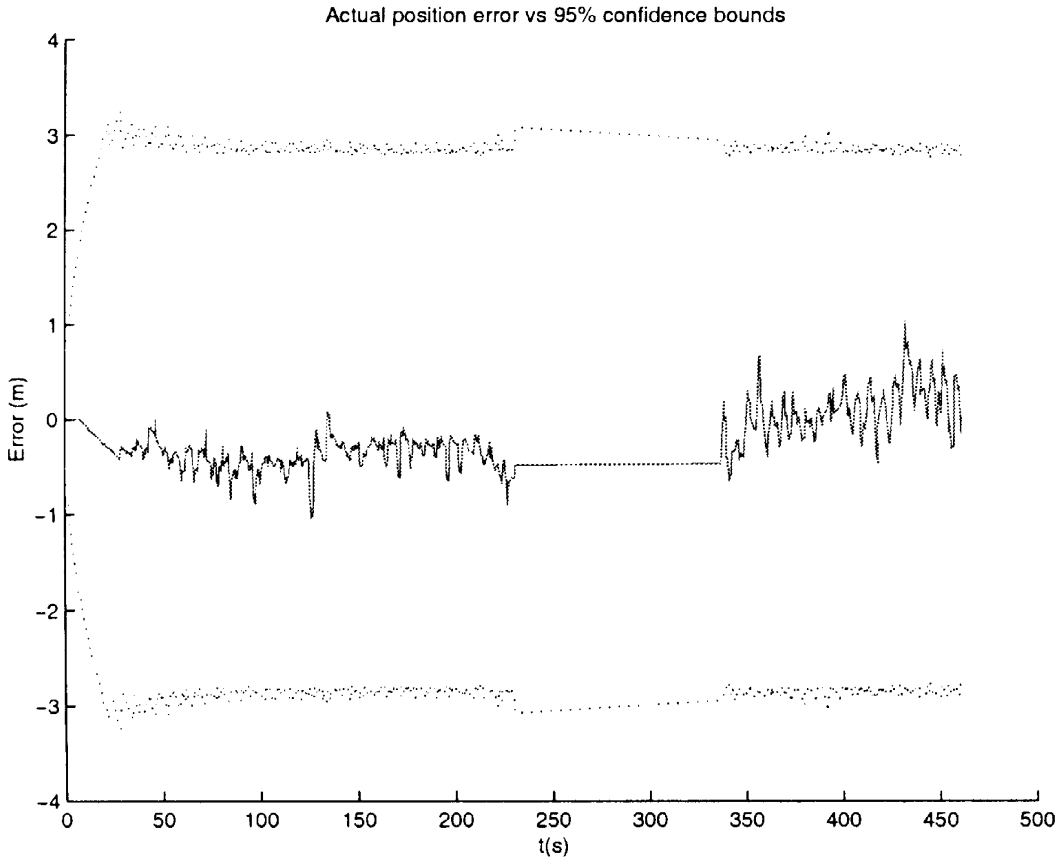


Fig. 14. X estimate error with 95% confidence bounds

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

can be seen in Figure 15, the innovation sequences for range and bearing are consistent.

The error in the estimated position of the beacons can be plotted against their respective 95% confidence bounds. This is shown in Figure 16 for Beacon 2 which was seen from the start of the run and Figure 17 for Beacon 4 which is only incorporated into the map after the first minute.

The final map generated by the SLAM algorithm is plotted in Figure 18. The true position of the sonar targets is shown and the associated map features can clearly be seen in the final map estimates. A number of point features have also been identified along the pool wall to the left of the image. The estimated path of the submersible is shown along with the covariance ellipses describing the confidence in the estimate. It is clear that the covariance remains fairly constant throughout the run.

5.2.2. Field trials. The SLAM algorithms have also been tested during deployment in a natural environment off the coast of Sydney. The submersible was deployed in a natural inlet with the sonar targets positioned in a straight line in intervals of 10 m. Since there is currently no absolute position sensor on the vehicle, the performance of the positioning filter cannot be measured against ground truth at this time. The innovation sequence can, however, be monitored to check the consistency of the estimates. Figure 19 shows that the innovation sequences are within the covariance bounds computed by the algorithm.

The plot of the final map shown in Figure 20 clearly shows the position of the sonar feature targets along with a number of tentative targets that are still not confirmed as sufficiently reliable. Some of the tentative targets are from the reef wall while others come from returns off of the tether. These returns are typically not very stable and therefore do not get incorporated into the SLAM algorithm. 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.

6. Conclusions

In this paper, it has been shown that the decoupled, distributed control architecture proposed here is practically feasible for the control of an underwater vehicle. 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 submersible. Given the nature of the anticipated missions for this vehicle, it appears that more complicated control schemes relying on complex models of the submersible dynamics are not necessary in the context of this problem.

It has also been shown that SLAM is practically feasible in both a swimming pool at the University of Sydney and in a natural terrain environment on Sydney's shore-line. By using terrain information as a navigational aid, the vehicle is able to detect unmodeled disturbances in its motion induced by the tether drag and the effect of currents.

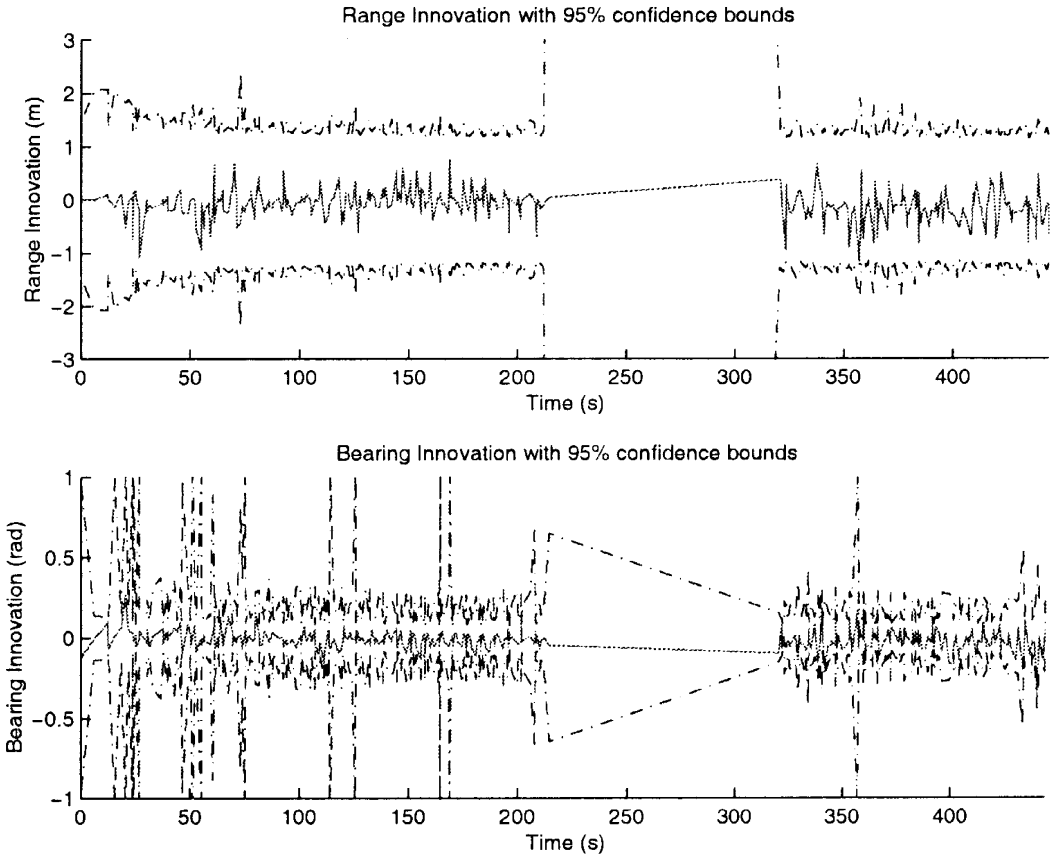


Fig. 15. Range and bearing innovation

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

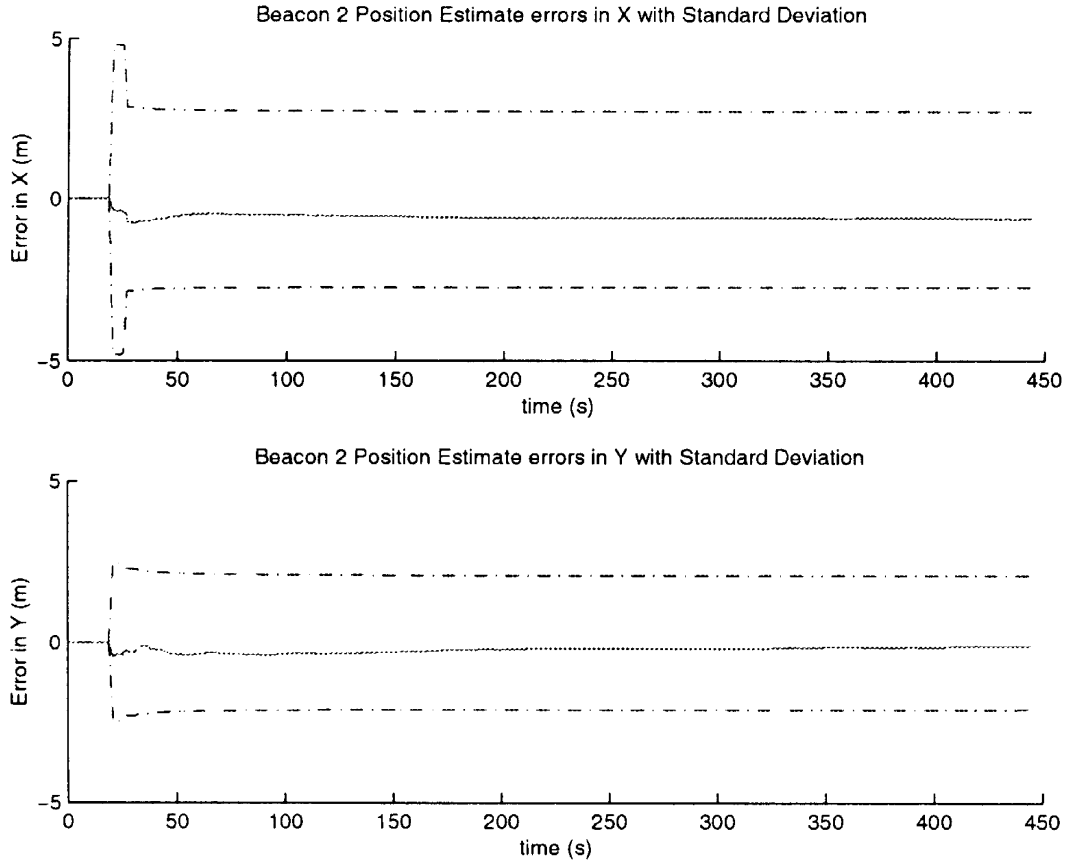


Fig. 16. Beacon 2 error estimate

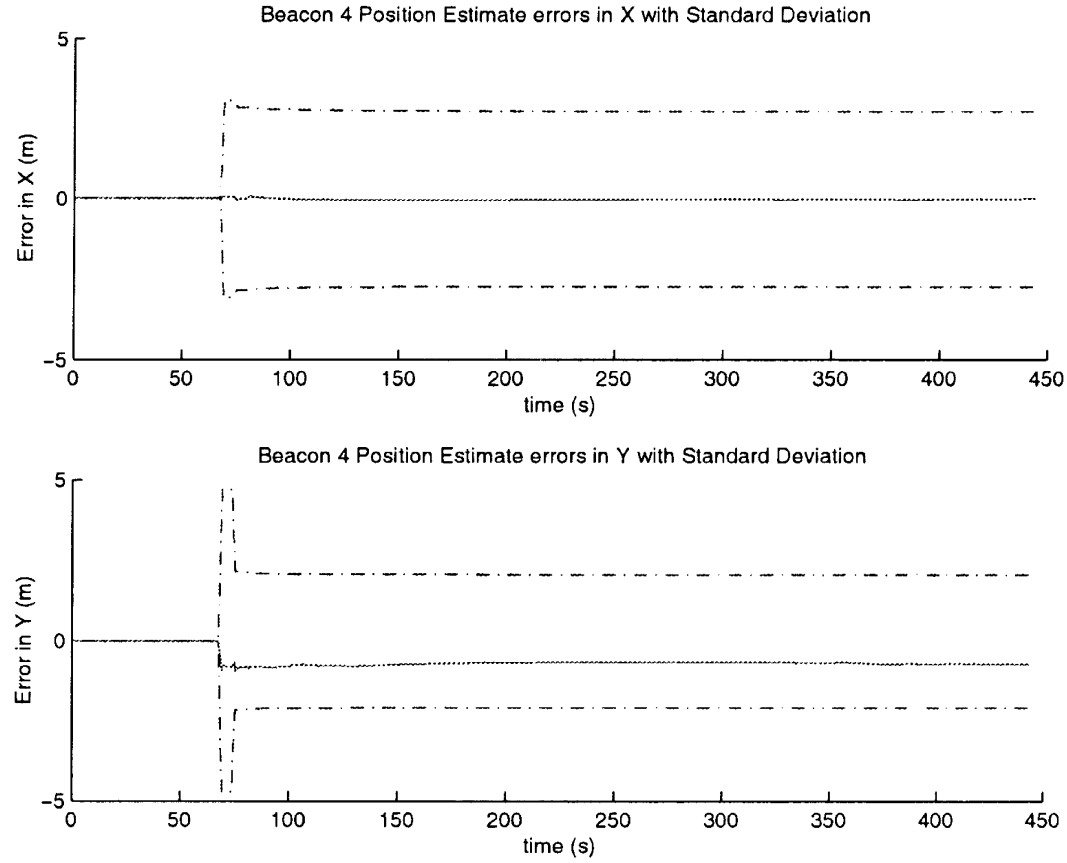


Fig. 17. Beacon 4 error estimate

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

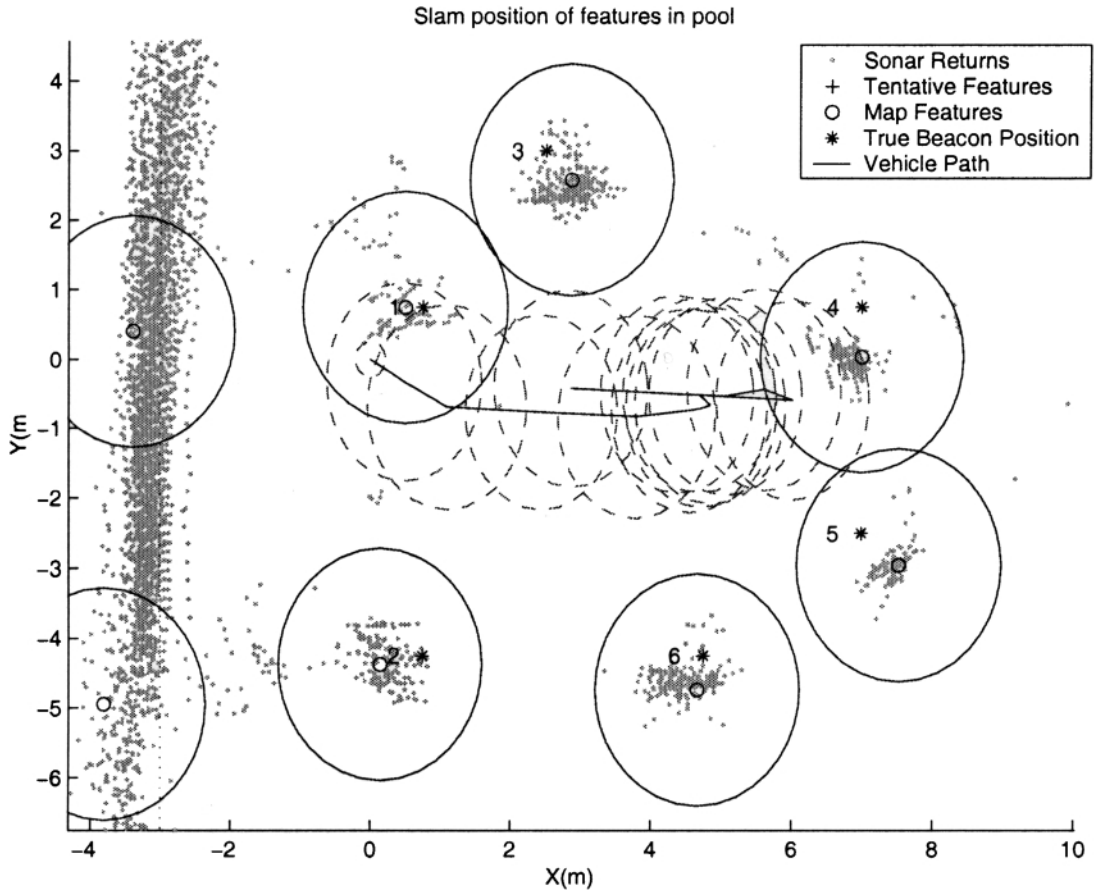


Fig. 18. Path of the robot in the pool

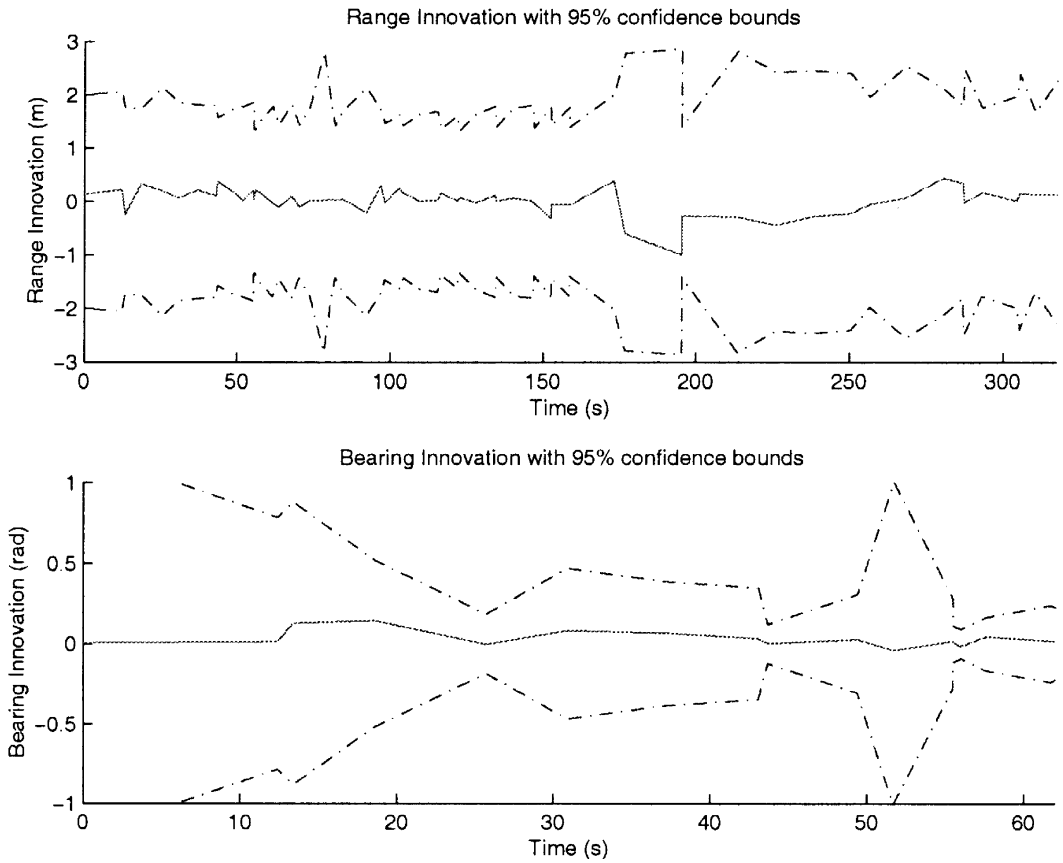


Fig. 19. Range and bearing innovation in natural terrain

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

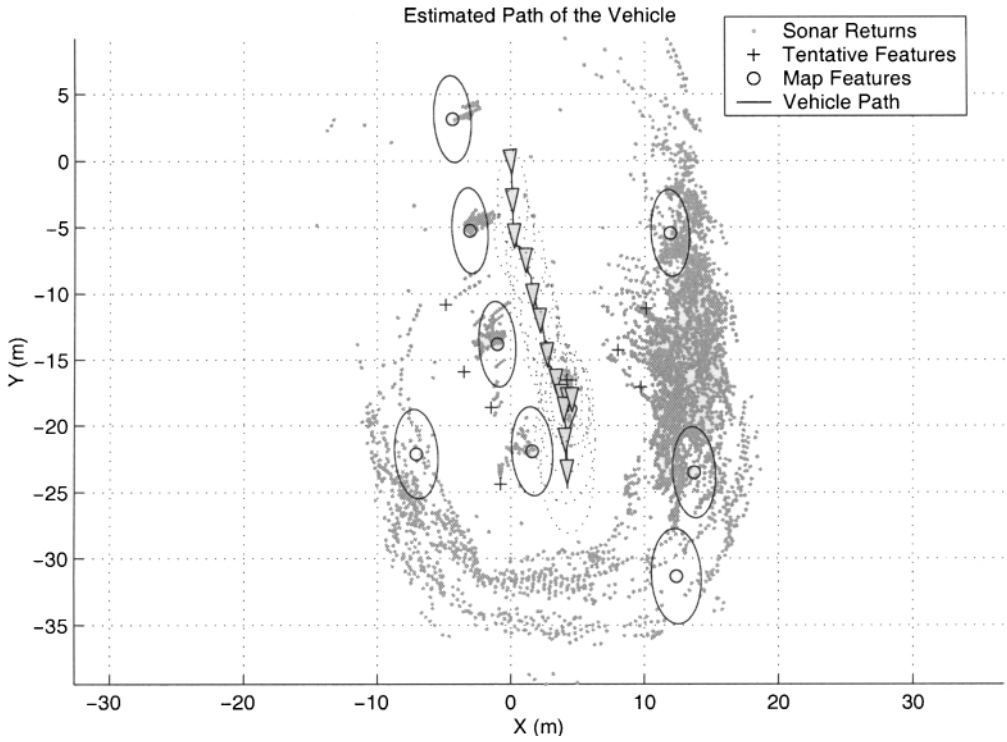


Fig. 20. Path of the robot in natural terrain

The focus of future work is on representing natural terrain in a form suitable for incorporation into the SLAM algorithm. This will enable the vehicle to be deployed in a broader range of environments without the need to introduce artificial beacons. Another outstanding issue is that of map management. As the number of calculations required to maintain the state covariance estimates increases with the square of the number of beacons in the map, criteria for eliminating features from the map as well as for partitioning the map into submaps becomes important. This is especially true for longer missions in which the number of available landmarks is potentially quite large. Finally, integration of the localisation and map building with mission planning is under consideration. This will allow decisions concerning sensing strategies to be made in light of the desired mission objectives.

References

1. J. A. Castellanos, J. M. M. Montiel, J. Neira and J. D. Tardos, "Sensor Influence in the Performance of Simultaneous Mobile Robot Localization and Map Building", *6th International Symposium on Experimental Robotics*, Sydney, NSW, Australia (1999) pp. 203–212.
2. M. W. M. G. Dissanayake, P. Newman, H. F. Durrant-Whyte, S. Clark and M. Csorba, "An Experimental and Theoretical Investigation into Simultaneous Localisation and Map Building", *6th International Symposium on Experimental Robotics*, Sydney, NSW, Australia (1999) pp. 171–180.
3. H. J. S. Feder, J. J. Leonard and C. M. Smith, "Adaptive sensing for terrain-aided navigation", *IEEE Oceanic Engineering Society. OCEANS '98*, New York, NY, USA (1998) **Vol. 1**, pp. 336–341.
4. J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous Map Building and Localisation for an Autonomous Mobile Robot", *IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '91*, New York, NY, USA (1991) pp. 1442–1447.
5. W. D. Rencken, "Concurrent localisation and map building for mobile robots using ultrasonic sensors", *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, New York, NY, USA (1993) **Vol. 3**, pp. 2192–2197.
6. L. Whitcomb, D. Yoerger, H. Singh and D. Mindell, "Towards Precision Robotic Maneuvering, Survey, and Manipulation in Unstructured Undersea Environments", *Robotics Research – The Eight International Symposium* (1998) pp. 346–353.
7. J. Rosenblatt, "The Distributed Architecture for Mobile Navigation", *J. Experimental and Theoretical Artificial Intelligence*, **9**, No. 2/3, 339–360 (April–September, 1997).
8. J. Rosenblatt, S. Williams and H. F. Durrant-Whyte, "Behavior-Based Control for Autonomous Underwater Exploration", *Proceedings of IEEE International Conference on Robotics and Automation (ICRA00)*, San Francisco, CA (2000) pp. 1793–1798.
9. S. Williams, P. Newman, G. Dissanayake and H. F. Durrant-Whyte, "Autonomous Underwater Simultaneous Localisation and Mapping", *Proceedings of IEEE International Conference on Robotics and Automation (ICRA00)*, San Francisco, CA (2000) pp. 920–925.
10. P. Newman and H. F. Durrant-Whyte, "Toward Terrain-Aided Navigation of a Subsea Vehicle", *FSR '97 International Conference on Field and Service Robotics*, Canberra, Australia (1997) pp. 244–248.
11. S. Williams, P. Newman, S. Majunder, J. Rosenblatt and H. F. Durrant-Whyte, "Autonomous Transect Surveying of the Great Barrier Reef", *Australian Conference on Robotics and Automation (ACRA '99)*, Brisbane, QLD (1999) pp. 16–20.
12. R. A. Brooks, "A robust, layered control system for a mobile robot", *IEEE Journal of Robotics and Automation*, **RA-2**, No. 1, 14–23 (April, 1986).
13. L. Whitcomb, D. Yoerger, H. Singh and J. Howland, "Advances in Underwater Robot Vehicles for Deep Ocean

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press

Exploration: Navigation, Control and Survey Operations", *9th International Symposium on Robotics Research (ISRR '99)*, Snowbird, Utah, USA (1999) pp. 346–353.
14. R. Bauer and W. D. Rencken, "Sonar feature-based exploration", *1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Los Alamitos, CA, USA (1995) **Vol. 1**, pp. 148–153.
15. J. J. Leonard, R. N. Carpenter and H. J. S. Feder, "Stochastic Mapping Using Forward Look Sonar", *FSR '99 International Conference on Field and Service Robotics*, Pittsburg, PA, USA (1999) pp. 69–74.
16. P. Newman, "On The Structure and Solution of the Simultaneous Localisation and Map Building Problem", *PhD Thesis* (University of Sydney, 1999).

https://doi.org/10.1017/S0263574701003423 Published online by Cambridge University Press