A Feature Based Navigation System for an Autonomous Underwater Robot

John Folkesson, Jacques Leederkerken, Rob Williams,
Andrew Patrikalakis, and John Leonard

Massachusetts Institute of Technology

Summary. We present a system for autonomous underwater navigation as implemented on a Nekton Ranger autonomous underwater vehicle, AUV. This is one of the first implementations of a practical application for simultaneous localization and mapping on an AUV. Besides being an application of real-time SLAM, the implemtation demonstrates a novel data fusion solution where data from 7 sources are fused at different time scales in 5 separate estimators. By modularizing the data fusion problem in this way each estimator can be tuned separately to provide output useful to the end goal of localizing the AUV, on an a priori map. The Ranger AUV is equipped with a BlueView blazed array sonar which is used to detect features in the underwater environment. Underwater testing results are presented. The features in these tests are deployed radar reflectors.

1 Introduction

Navigating underwater requires localization without GPS. Stationary features in the environment can be used to help the robot stay localized [1], [2], [3], [4]. Examples of sonar mapping can be seen in [5], [6] and [7]. This is commonly known as simultaneous localization and mapping, SLAM . Some underwater robots have the advantage of rather expensive sensors such as Doppler velocity logs for measuring the movement of the robot. In this application the robot must be expendable which limits the choice of sensors. Instead of a single high quality sensor of the motion we have a number of weaker sensors that must be fused to estimate the motion of the robot.

The problem of fusing data from dissimilar sensors is challenging for a number of reasons [8], [9], [10], [11], [12]. First there is a temporal issue in that some sensors need to be incorporated with a simple calculation at a high update rate while other information might need intensive calculation but at a much lower update rate. Another issue is observablity. Generally each sensor provides only a partial observation of the total state. The problem is complicated by the fact that some sensors provide measurements in the relative frame of the robot while others are in the absolute earth frame. Measurement errors that are correlated over time due to bias or environmental disturbances can lead to over confident estimates if the errors are treated as independent.

A more practical issue is that of getting the system to work robustly. This requires that it be possible to examine each sensor's contribution to the final

C. Laugier and R. Siegwart (Eds.): Field and Service Robotics, STAR 42, pp. 105–114, 2008.
springerlink.com

estimate separately. Only then can one hope to understand the system well enough to tune the parameters.

2 Project Overview

The task of clearing a lane from sea to shore through a mine field has traditionally been carried out by human divers. This method is associated with very high casualty rates but in many cases there are no alternatives. It is therefore very desirable to come up with methods that do not cost lives.

The system presented here is part of a larger project. The larger mission scenario starts with a target field of underwater mines along a shoreline. The mine field will be scouted first by a REMUS AUV. The data collected from the REMUS will be used to create an a priori map of the mines and other strong sonar reflecting objects along the shore. A particular target mine will be selected. The Nekton Ranger will then be sent on a mission to locate and attach to the target. As the Ranger is a less expensive AUV than REMUS it is meant to represent an expendable robot that could be set to detonate an explosive charge after attaching itself to the target.

The a priori map will not be perfect. It will have an absolute accuracy of about 5 meters. The relative accuracy between nearby features will be considerably better. The Ranger will be be able to get to within 70 meters of the target on the surface and to dive to the target depth. The task then becomes one of distinguishing between several nearby targets to find the designated target. The AUV must then navigate to that target accurately.

The approach taken is to have the Ranger carry out simultaneous localization and mapping, SLAM, were the SLAM state is initialized with the a priori map including its uncertainty. By then matching the features detected by the Ranger to the a priori map the robot pose will become correlated to the a priori map. When this correlation is sufficiently accurate the target can be acquired with confidence.

The a priori map will initially have relatively tight correlations between features that go like the inverse of the distance between the features. The absolute uncertainty of the features will be much looser allowing the map to shift as a whole to agree with where the Ranger finds it. Thus upon matching a feature seen by the Ranger with an a priori map feature both the Ranger pose and the map will shift to reconcile any differences. The side scan sonar used by the REMUS to collect the data for the a priori map is different than the forward looking blazed array sonar on the Ranger. It is therefore likely that the maps will look different both in feature positions and numbers.

3 System Overview

Figure 1 is a schematic over the navigation system. Its inputs, raw motion data and sonar detections, are written to as data arrives. These writes then trigger the filters downstream to update as data is pushed and pulled through the system.

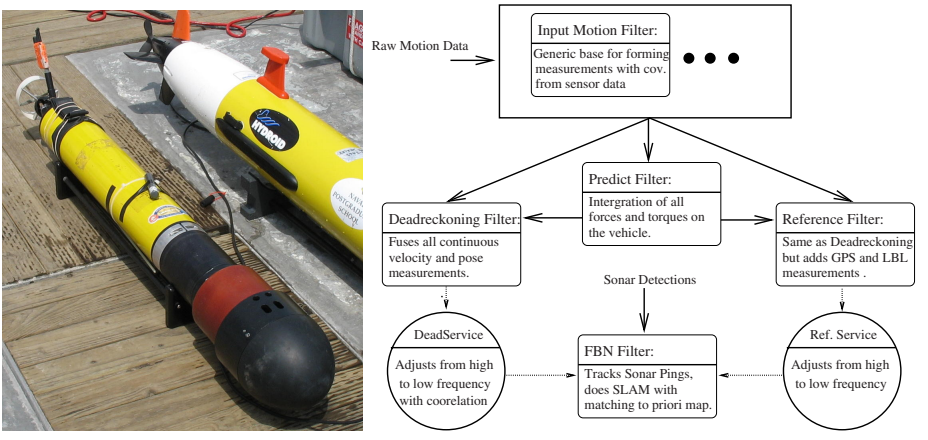

Fig. 1. The Nekton Ranger is shown to the left. The separate modules of the navigation system are shown on the right.

The data from the motion sensors is fed to sensor specific modules called input motion filters in figure 1. These model the individual sensors error characteristics. This sensor data is fused using five separately tuned estimators. Each estimator has a specific purpose and frequency as summarized in table 1 The outputs of the input motion filters are pushed to the predict filter and to two separate Extended Kalman Filters (EKF). The predict filter uses a model of the forces and torques acting on the vehicle to estimate the motion over short periods of time. Each of the EKF filters has a specific function. The dead-reckoning filter provides the best continuous estimate of the robot pose. This estimate is used as the frame for accumulating data from the sonar.

The reference EKF is identical to the dead-reckoning EKF aside from its inputs and outputs. GPS and long base line transponder data (LBL) are the additional inputs. These give measurements of the absolute x and y position of the robot. They also cause the estimated trajectory to have discontinuities in it.

Table 1. The five estimators that fuse the various sensor data are summarized

Estimator	Output	Rate	Inputs
Predict Filter	Motion between measurements	100 Hz	Propeller Actuation signals, Various parameters
Deadreck. EKF	Smooth pose trajectory for tracking	$\approx$ 30 Hz async	3-Axis Compass, Depth, Altitude, Predict output
Reference EKF	Corrected xy and parameter estimates	$\approx$ 30 Hz async	3-Axis Compass, Depth, Altitude, GPS, LBL, Predict output
Tracking Filter Squareroot SAM	Composite measurements	1 to 2 Hz	Sonar Detections, Dead-reckoning output
Global EKF	Localization on an a priori map	.01 to .1 Hz	Composite Measurements and Reference output

These 'jumps' are why the reference estimate is not used for feature tracking. The reference EKF also estimates more parameters than the dead-reckoning EKF. The additional parameter estimates such as water current velocity can only be estimated using the GPS or LBL. The corrected parameters and their covariance are then passed to the predict filter to be used in prediction for both EKF filters.

The feature based navigation, FBN, filter consists of two separate estimators. The tracking filter and the global EKF. The tracking filter accumulates information from the sonar detections over short snippets of the robot path, (SLAM). The local map and trajectory from these snippets are then used to form *composite measurements*. These are then used to update the global EKF. Matching to the prior map is done on the global EKF.

4 The Predict Filter

A propeller on the aft end of the robot is mounted on a universal joint that can be pan/tilt rotated relative to the robot. An estimate of the pan/tilt angle is used to decompose this propeller thrust into robot pitch/yaw torques and a force forward. The movements of the robot through the water are resisted by forces and torques proportional to the linear and angular velocities. These forces then produce a convergent velocity for a given RPM. The parameters are tuned to give the empirical .75 m/s terminal velocity at 1,200 RPM, Fig. 2. The prediction model includes damping, actuation transfer, water velocity, buoyancy, and wave surge parameters, all modeled as Gaussian.

These various forces and torques are integrated at a rate of 100 Hz using Newton's equations to provide the EKF filters with estimates of the motion between the sensor measurements. The EKF filter can be configured to estimate and pass to the predict filter values for any of the parameters. The predict filter will then used those along with its own defaults for the others when making predictions. The right plot of Fig. 2 shows the improvement in the dead-reckoning from using the reference EKF estimated parameters.

5 Dead-Reckoning EKF

The dead-reckoning EKF estimates the robot pose, its velocity and angular actuation parameters, a total of 15 dof. As input it has depth, altitude velocity, and orientation from a 3-axis compass. The output is a smooth and accurate estimate of the motion between two sonar pings.

The dead service module does the translation from the EKF outputs at around 30 Hz to two pose estimates at the ping rate (1-2 Hz). The two pose estimates are a 6 dof incremental change estimate P_i and a 4 dof earth frame measurement, P_e. The two estimated poses are then able to account for the correlations between estimates taken at different times. The EKF provides snapshots of the uncertainty at the time of each sonar ping. We must break this uncertainty up into a part relative to the previous pose and a part relative to the earth. This is done by accumulating the predictions used in the EKF. Both poses' mean are

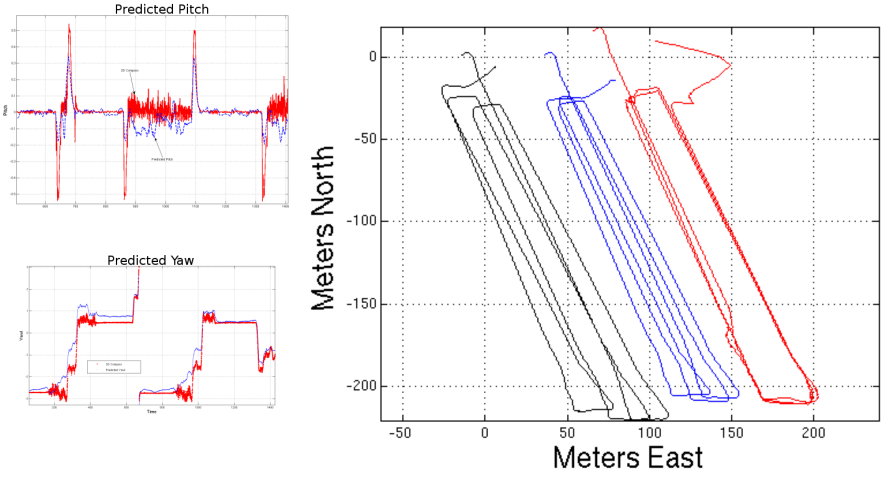

Fig. 2. The tuning of the prediction for the yaw and pitch angle is done by comparing to the 3D compass as shown on the bottom left (yaw) and. upper left (pitch). The figure to the right compares the dead-reckoning and reference estimates of the robot trajectory. The three paths have been offset to separate them for comparison. The path to the right is the reference EKF output. It uses the GPS during the turn arounds on the surface. The middle is the dead-reckoning EKF, sharing the predict filter with the reference EKF. This allows it to benefit from the parameter estimate made in the reference EKF. The left is the EKF, without any parameter sharing.

the EKF values. The covariance of P_i is taken as the accumulated prediction value. This then gives a system of matrix equations that can be solved for the covariance of P_e. In this way we were able to have the tracking filter run at a much slower rate than the EKF filters. At the same time we had the exact pose and covariance estimate that the EKF provided.

6 Reference EKF

The reference EKF adds estimates of velocity damping, linear actuation, and water velocity parameters to the state vector estimated by the dead-reckoning EKF. A total of 27 dof. The purpose of the reference filter is quite different than that of the dead-reckoning EKF. The reference EKF is used to smooth the GPS and LBL measurements for use by the global EKF which operates at a much lower frequency. Only the xy parts are actually used. The global EKF will only use the reference xy estimate for an update if the covariance in xy has decreased (ie. when the Reference EKF has had a GPS update).

The reference EKF will only use a GPS measurement if its estimated covariance is less than that of the reference EKF. The reason for this seemingly odd update rule is that the GPS measurements have time correlated errors. Using consecutive GPS measurements and treating them as independent will cause the

filter to become over confident very quickly. The rule softens this and essentially ensures that the covariance of the EKF will never get much lower than that of the single best GPS measurement.

7 The Tracking Filter

The tracking filter has the function of tracking features in the sonar data and using them to correct the motion of the robot over times of 10 to 100 seconds. These times correspond roughly to the time a feature is in the field of view of the sonar. The output of this filter is a composite measurement that combines the dead-reckoning and feature tracking information over this interval. So a composite measurement gives the positions of any features tracked, the estimated pose change for the robot and the absolute (z,θ,ϕ,ψ) of the robot at the end of the interval.

The tracking is done in a small scale but highly detailed SLAM filter. This SLAM filter captures all the characteristics of the blazed array sonar. It estimates the entire path of the robot over the interval. Thus it adds 6 dimensions to its state for the robot pose on each new ping. The features are parameterized in the sensor frame of the first pose they were seen from. In that way the feature will move when this pose is re-estimated. The parameterization in that frame is the range and bearing to the feature. This ensures that the uncertainty in these parameters will be closer to Gaussian than simple Cartesian coordinates would be.

The tracking estimator is a square root filter [13]. This is a Gaussian estimator similar to the Extended Information Filter, EIF, which in turn is equivalent to the EKF. It has several important advantages. The most important is that as the square root of the information matrix is used the condition number of this matrix will be smaller. A smaller condition number means that the matrix equations that must be solved are numerically more well-posed. This will virtually always lead to an improvement over EKF or EIF for SLAM. Secondly, one can in our case do the updates in constant time. This is because the short snippets of path we use have no loops in them. Partial covariances can be computed in constant time as well. Estimating the full covariance would take a longer time but is actually never needed.

The details of this filter would take us beyond the scope of this paper and will appear in a later publication. Here we present the system as a whole and concentrate on how the parts work together in a particular application.

8 The Global EKF

The global EKF is updated with the composite measurements. This begins by augmenting the state with the pose and features of the composite measurement. The EKF state can then be updated using the composite measurement. The older previous pose can then be marginalized out. This implies simple deleting

the rows and columns of the older pose. The new features can then be matched and merged with the older features including the a priori map features.

The matching of the features is done in three ways. First any features that were tracked in the tracking filter from one composite measurement to another are matched based on a key provided by the tracking filter. Second a threshold is placed on the mahanolobis distance between single features that are relatively well correlated to one another to determine matching features. Third multiple feature matches are search for. All of the pairs of possibly matched features are used a a basis over a hypothesis space. This space is search exhaustively if small enough or selectively if too large. The details of the selective search and match criteria are original and require an entire paper to describe. One part of our criteria is the joint compatibility test [14]. This forms a distance between sets of points which is essentially the Mahanolobis distance of a pseudo measurement equating the point in each set.

$$d = \eta S^{-1} \eta. \tag{1}$$

Where:

$$S = C_{\eta\eta} + J_{\eta,\mathbf{x}} C_{\mathbf{xx}} J_{\eta,\mathbf{x}}^T. \tag{2}$$

Here $C_{\eta\eta}$ is the covariance of this pseudo measurement (a parameter) and should reflect the finite size of the actual features, $C_{\mathbf{xx}}$ is the covariance of the EKF state and $J_{\eta,\mathbf{x}}$ is the Jacobian of the measurement. This is balanced against a threshold that is based on certain statistical measurements accumulated over the path of the robot.

The problem of finding candidate combinations to apply this test to is where the difficulty lies. The number of possibilities quickly becomes unmanageable as the number of features in the map increases. For this application we deal with a relatively small number of features and a very slow update rate for the maps. However we still need to limit this search at times.

We start by forming $matchdata_{ij}$ objects between each pair of features, labeled i and j. This is a N^2 operation where N is the number of features. The match data consists of r_{ij}, the Euclidean distance between the points,c_{ij}, the covariance of rij, and a_{ij}, the angle of the vector between them.

$$rij = |\mathbf{x}_i - \mathbf{x}_j|, \tag{3}$$

$$c_{ij} = J_{r,x} C_{\mathbf{xx}} J_{r,x}^T \tag{4}$$

$$aij = arctan(y_i - y_j, x_i - x_j) \tag{5}$$

$$matchdata_{ij} = (r_{ij}, c_{ij}, a_{ij}) \tag{6}$$

Remember that we are using a compass and will take full advantage of knowing the orientation of the robot and thus these angles in some cases. The cases that we feel reasonably confident about the angles between two features is when the two features are relatively tightly correlated. This means that they were either seen near each other in time as the robot moved or that they were matched to features that were in turn tightly coupled to one another.

We see that this concept of two features being tightly coupled is important. We define tightly coupled as $c_{ij} <$ some threshold. Having this we can form N lists of $matchdata_{ij}$ objects where the i^{th} list contains all the j that are tightly coupled to i. These are the local patches that we will try to find multiple matches between.

Continuing to try and limit the number of possible matches we can note all pairs that are possibly the same point feature by either r_{ij} or rij^2/c_{ij} being relatively small. These will be stored on a list of possible matches. We can then iterate through this list and search for multiple matches between the local patches associated with each of the possibly matched features. We check if there are other possible matched features between the local patches by checking the list.

9 Experiments

The experiments consisted of a series of underwater tests. A target field was deployed consisting of 12 radar reflectors anchored 1 meter off the sea floor. These were meant to represent a mine field obstructing access to a part of the shoreline. The targets were deployed in three rows perpendicular to the shore. Two of the rows had five reflectors and a center row had 2 reflectors. This target field was then surveyed by the REMUS AUV (see Figure 1). This data was then used to form a hand made a priori map for the Ranger AUV to use, Figure 3 left.

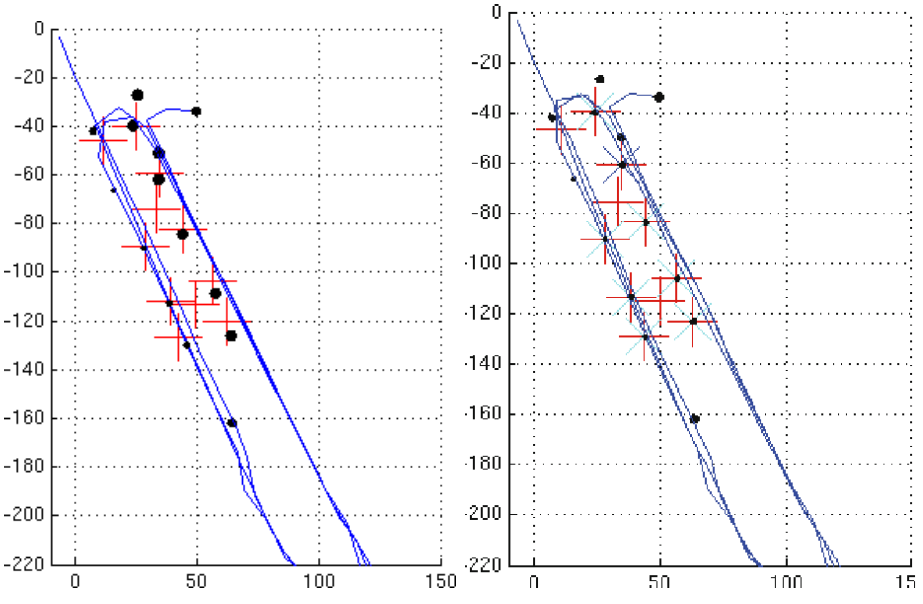

Fig. 3. On the left is the show the SLAM map as black ellipsoids and the a priori map as red crosses. This map made no attempt to match the two maps to one another. On the right is a map made with matching. The robot succeeded in matching on all 6 dives demonstrating the localization of the robot is working.

The final map made by the Ranger is seen to be very similar to the a priori map however it is shows some features displaced from the priori positions. The absolute xy of the map is of no significance here the important point is that the robot pose is correlated to the a priori map. This despite significant differences in the GPS location fix. The robot here dove six separate times and was able to correctly match features to the map each time. In at least one instance, (the first return to start location), the ranger FBN Filter gave better estimated positions then the Reference EKF had.

10 Conclusion

The system was able to localize the robot on the a priori map in real time. This will allow an autonomous AUV to home in on a pre-designated target. The approach of fusing the different sensor data using five separate estimators works well. This approach allows us to modularize the system which adds flexibility and presumably some robustness. The various estimates give us a more precise view of the overall estimation process than a single integrated estimator could give. We can then identify the source of any errors more quickly. These error then give us a valuable guide to adjustments and refinements of the estimators. The GPS was able to be used to improve the dead-reckoning estimate without introducing any discontinuities in the pose estimate. That thanks to a dual EKF implementation.

In the future we plan to work on the final control of the robot to the designated target. We would also like to add some active perception to the system by having the robot rotate slightly both in pitch and yaw when targets are in view. This will help the convergence of the tracked feature's covariances.

References

1. Dissanayake, M.G., Newman, P., et al.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Trans. on Robotics and Automation 17(3), 229–241 (2001)
2. Newman, P., Leonard, J., Rikoski, R.: Towards constant-time slam on an autonomous underwater vehicle using synthetic aperture sonar. In: Proc. of the International Symposyum on Robotics Research (ISRR 2003) (2003)
3. Leonard, J., Rikoski, R., Newman, p., Bosse, M.: Mapping partially observable features from multiple uncertain vantage points. IJRR International Journal on Robotics Research 7(3), 943–975 (2002)
4. Leonard, J.J., Rikoski, R.J.: Incorporation of delayed decision making into stochastic mapping. In: Experimental Robotics VII. lecture Notes in Control and Information Sciences, vol. 271, pp. 533–542. Springer, Heidelberg (2001)
5. Ribas, D., Ridao, P., Neira, J., Tardos, J.: Slam using an imaging sonar for partially structured underwater environments. In: Proc. of the IEEE International Conference on Intelligent Robots and Systems(IROS 2006), IEEE, Los Alamitos (2006)
6. Leonard, J.J., Carpenter, R., Feder, H.J.S.: Stochastic mapping using forward look sonar. Robotica 19, 467–480 (2001)

7. Tena, I., de Raucourt, S., Petillot, Y., Lane, D.: Concurrent mapping and localization using sidescan sonar. IEEE Journal of Ocean Engineering 29(2), 442–456 (2004)
8. Castellanos, J.A., Neira, J., Tardós, J.D.: Multisensor fusion for simultaneous localization and map building. IEEE Transactions on Robotics and Automation 17(6), 908–914 (2001)
9. Durrant-Whyte, H.F., Rao, B.Y.S., Hu, H.: Toward a fully decentralized architecture for multisensor datafusion. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA 1990) (1990)
10. Moutarlier, P., Chatila, R.: Stochastic multisensory data fusion for mobile robot location and environmental modelling. In: Proc. of the International Symposium on Robotics Research, pp. 85–94 (1990)
11. Guivant, J.E., Masson, F.: Using absolute non-Gaussian non-white observations in Gaussian SLAM. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA 2005), vol. 1, pp. 338–343 (2005)
12. Sukkarieh, S., Nebot, E., Durrant-Whyte, H., Corba, M.: A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. IEEE Trans. on Robotics and Automation 15(3), 572–578 (1999)
13. Dellaert, F.: Square root sam: Simultaneous location and mapping via square root information smoothing. In: Robotics: Science and Systems (2005)
14. Neira, J., Tardós, J.D.: Data association in stocastic mapping using the joint compatibility test. IEEE Transaction on Robotics and Automation 17(6), 890–897 (2001)