Navigating and Mapping with the SPARUS AUV in a Natural and Unstructured Underwater Environment

Angelos Mallios, Pere Ridao, Marc Carreras and Emili Hernández
Institute of Informatics and Applications, Universitat de Girona,
Campus de Montilivi, 17071, Girona, Spain
Email: {amallios,pere,marcc,emilih}@eia.udg.edu

Abstract—In spite of the recent advances in unmanned underwater vehicles (UUV) navigation techniques, robustly solving their localization in unstructured and unconstrained areas is still a challenging problem. In this paper, we propose a pose-based algorithm to solve the full Simultaneous Localization And Mapping (SLAM) problem for an Autonomous Underwater Vehicle (AUV), navigating in the unknown and unstructured environment. A probabilistic scan matching technique using range scans gathered from a Mechanical Scanning Imaging Sonar (MSIS) is used together with the robot dead-reckoning displacements. The raw data from the sensors are processed and fused in-line with an augmented state extended Kalman filter (EKF), that estimates and keeps the scans poses. The proposed SLAM method has been tested with a real world dataset acquired from the Sparus AUV, guided in a natural underwater environment.

I. INTRODUCTION

During a long term mission with an autonomous robot it is necessary to keep the track of the vehicle’s position. Scan matching is a technique that can be used to estimate the vehicle displacement using successive range scans. Many applications in robotics like mapping, localization, pose tracking or SLAM use this technique to estimate the robot’s relative displacement; [1], [2], [3], to mention some but a few. In many senses, scan matching techniques have some resemblance to image mosaicking techniques [4].

This paper is a contribution in this area and the future work of [5], proposing a pose-based algorithm to solve the full SLAM problem of an AUV navigating but this time, in an unknown and unstructured environment, as it is a natural underwater tunnel. The technique incorporates probabilistic scan matching with range scans gathered from a MSIS, taking into account the robot dead-reckoning displacements estimated from a Doppler Velocity Logger (DVL) and a Motion Reference Unit (MRU).

Although a large literature exists reporting successful applications of scan matching and SLAM to mobile robots, very few attempts have been done to use sonar scan matching in underwater applications and even fewer putting them in a SLAM framework. In [6], a non-probabilistic variation of Iterative Closest Point (ICP) is proposed to achieve on-line performance for registering multiple views captured with a 3D acoustic camera. In [7], the authors proposed to use a particle filter to deal with the sonar noisy data but only simulated results are reported. An AUV equipped with an array of 56 narrow beam sonar transducers explores cenotes (i.e. sinkholes) in Mexico. The map is stored within a 3D evidence grid which uses the Deferred Reference Counting Octree (DCRO) data structure to reduce the memory requirements [8]. On a similar environment but with a sonar-equipped Remotely Operated Vehicle (ROV), in [9], four different mapping and localization techniques were tested: Sonar image mosaics using stationary sonar scans, SLAM while the vehicle was in motion, SLAM using stationary sonar scans, and localization using previously created maps. Using a MSIS in an AUV, in [10] demonstrated SLAM in the structured environment of a marina. An application combining SLAM and sonar scan matching underwater is reported in [11], were an ICP variant is used for registering bathymetric sub-maps gathered with a multibeam sonar profiler. With the same type of sensor, the authors in [12] modelled the uncertainty in the vehicle state using a particle filter and an EKF.

The paper is structured as follows. In section II, the probabilistic scan matching algorithm is described. The way to overcome the difficulties of the underwater sonar images and the scan matching SLAM technique, are detailed in section III. In section IV, we introduce the Sparus AUV that has been used for the experiments of this paper and section V, reports the experimental results before conclusions.

II. PROBABILISTIC SCAN MATCHING

The goal of scan matching is to compute the relative displacement of a vehicle between two consecutive configurations by maximizing the overlap between the range measurements obtained from a laser or a sonar sensor. That means, that given a reference scan $S_{ref}$, a new scan $S_{new}$ and an rough displacement estimation $q_0$ between them, the objective of scan matching methods is to obtain a better estimation of the real displacement $q = (x, y, \theta)$ (Fig. 1).

Several scan matching algorithms exists with most of them being variations of the ICP algorithm. The geometric representation of a scan in the conventional ICP algorithm does not model the uncertainty of the sensor measurements. Correspondences between two scans are chosen based on the closest-point rule normally using the Euclidean distance. As pointed out in [13], this distance does not take into account...
that the points in the new scan, which are far from the sensor, could be far from their correspondents in the previous scan. On the other hand, if the scan data are very noisy, two statistically compatible points could appear far enough, in terms of the Euclidean distance. Both situations might prevent a possible association or even generate a wrong one.

To overcome those problems, the authors in [13], proposed the Probabilistic Iterative Correspondence (pIC) which is a statistical extensions of the ICP algorithm, where the relative displacement as well as the observed points in both scans, are modelled as random Gaussian variables.

III. UNDERWATER SCAN MATCHING SLAM

As discussed in the previous section, scan matching is a technique that can provide a better estimation of the vehicle’s displacement. However, before the registration of two scans, the building process for each individual scan with a MSIS has to address a number of issues, because the rotation speed of the sonar head is comparable to the vehicle’s speed and this introduce distortion in the final scan image. For this reason and to help the reader to follow the algorithm, we will examine the proposed pose-based SLAM algorithm in two separate parts: the ScanGrabbing and the main SLAM algorithm.

A. ScanGrabbing algorithm

ScanGrabbing algorithm’s role, is to collect all the beams that forms a full 360° sonar image sector, analyse them and remove any motion distortion. Following, we will present the three major parts that ScanGrabbing algorithm consists of: Beam segmentation, Relative vehicle localization and Scan forming.

1) Beam segmentation and range detection: The MSIS returns a polar acoustic image composed of beams. Each beam has a particular bearing angle value and a set of intensity measurements acquired at known intervals along the beam path. The angle corresponds to the orientation of the sensor head when the beam was emitted. The acoustic linear image corresponding to one beam is returned as an array of acoustic intensities detected at a certain distance. To obtain a range measurement, the beam is then segmented using a predefined threshold to compute the intensity peaks. Due to the noisy nature of the acoustic data, a minimum distance between peaks criteria is also applied. Hence, the positions finally considered are those corresponding to high intensity values above the threshold with a minimum distance between each other.

2) Relative vehicle localization: To maximize the probability for data overlapping, we collect a complete 360° scan sector and register it with the previous one in order to estimate the robot’s displacement. Since MSIS needs a considerable period of time to obtain a complete scan, the robot’s motion induces a distortion in the acoustic image when the robot does not remain static, which is very common in water (Fig. 2). To deal with this problem it is necessary to know the robot’s pose at the beam reception time. Then, we define a reference coordinate system I, to reference all the range measurements belonging to the same scan. In order to reduce the influence of the motion uncertainties to the scan, we set this reference frame at the robot pose where the centre beam of the current scan was received.

The localization system used in this work to estimate the vehicle motion is a slight modification of the navigation system described in [14]. In this system, a MRU provides attitude measurements and a DVL unit which includes a depth sensor is used to estimate the robot’s velocity and depth during the scan. All measurements happen asynchronously with the MSIS beams arriving at 30 Hz rate, while DVL and MRU readings arriving at a frequency of 1.5 and 10 Hz respectively. It is very common that non gliding AUVs are very stable in roll and pitch, performing survey patterns at constant speed, so a simple 4 Degrees of Freedom (DoF) constant velocity kinematic model is used to predict the vehicle’s motion. An EKF is used to estimate the robot’s pose whenever a sonar beam is received and the model prediction is updated by the standard Kalman filter equations each time a new DVL or
MRU measurement arrives.

3) Scan forming: The navigation system presented above is able to estimate the robot’s pose, but the uncertainty will grow without limit due to its dead-reckoning nature. However, we are only interested in the robot’s relative position (and uncertainty) with respect to the centre of the scan (Ic frame, Fig. 3). Hence, a slight modification to the filter is introduced making a reset in position (setting x, y, z to 0 in the vector state) whenever a new beam is emitted. Note that it is important to keep the yaw (ψ) value (it is not reset) because it represents an absolute angle with respect to the magnetic north and a reset would initialise a relative angle and therefore, produce an error during the compass update. Since compass measurements can be easily integrated.

The navigation system presented above is able to estimate the robot’s pose, but the uncertainty will grow without limit due to its dead-reckoning nature. However, we are only interested in the robot’s relative position (and uncertainty) with respect to the centre of the scan (Ic frame, Fig. 3). Hence, a slight modification to the filter is introduced making a reset in position (setting x, y, z to 0 in the vector state) whenever a new beam is emitted. Note that it is important to keep the yaw (ψ) value (it is not reset) because it represents an absolute angle with respect to the magnetic north and a reset would initialise a relative angle and therefore, produce an error during the compass update. Since we are only interested in the uncertainty accumulated during the scan, the reset process also affects the x, y, and z terms of the covariance matrix P. Now, the modified filter provides the robot’s relative position where the beams where gathered (and uncertainty) with respect to the centre of the scan (Ic), removing the distortion induced by the robot’s motion.

B. SLAM algorithm

The proposed pose-based SLAM algorithm uses an augmented state EKF (ASEKF) for the scan poses estimation. In this implementation of the stochastic map [15], the estimate of the positions of the vehicle at the centre of each full scan at the time step (k), are stored in the state vector \( \hat{x}_k \), referenced at the base frame \( B \):

\[
\hat{x}_k^B = [\hat{x}_{n_k}^B \ldots \hat{x}_{n_k}^B]^T
\]

(1)

and the covariance matrix for this state is defined as:

\[
P_k^B = E((x_k^B - \hat{x}_k^B)(x_k^B - \hat{x}_k^B)^T)
\]

(2)

We remind that a full scan is defined as the final 360° polar range image obtained after compounding all the robot poses with the 200 beams needed to obtain the full sector. The scan is referenced in the centre of that path, which is the output from the ScanGrabbing algorithm.

1) Map initialisation: All the elements on the state vector are represented in the map reference frame \( B \). Although this reference frame can be defined arbitrarily, we have chosen to place its origin on the initial position of the vehicle at the beginning of the experiment and orient it to the north, so compass measurements can be easily integrated.

The pose state \( x_i \) is represented as:

\[
x_i^B = [x \ y \ \psi]^T
\]

(3)

where, \( x \), \( y \), and \( \psi \) is the position and orientation vector of the vehicle in the global frame \( B \). The state and the map are initialized from the first available heading measurement.

2) Prediction: Let,

- \( x_k^B \equiv \mathcal{N}(\hat{x}_k^B, P_k^B) \) be the last robot pose, and
- \( q_{n_k}^B \equiv \mathcal{N}(\hat{q}_{n_k}^B, P_{n_k}^B) \) be the robot displacement during the last scan, estimated through dead reckoning.

then the prediction / state augmentation equation is given by:

\[
\hat{x}_{k+1}^B = \hat{x}_k^B \circ q_{n_k}^B = \left[ \hat{x}_{n_k}^{B-1} \circ q_{n_k}^B \right] \hat{x}_{n_k-1}^B \ldots \hat{x}_{n_k}^B \ldots \hat{x}_{1_k}^B
\]

(4)

where, given that \( B \) and \( B_n \) frames are both north aligned, the operator \( \circ \) is defined as:

\[
x \circ q = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \circ \begin{bmatrix} d \\ e \\ f \end{bmatrix} = \begin{bmatrix} a + d \\ b + e \\ c + f \end{bmatrix}
\]

(5)

being \( J_{1\odot} = \begin{bmatrix} 1_{2\times2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \) and \( J_{2\odot} = I_{3\times3} \) the corresponding linear transformation matrices, then the predicted pose uncertainty \( P_{k+1}^B \) can be computed as:

\[
P_{k+1}^B = J_{1\odot} P_k^B J_{1\odot}^T + J_{2\odot} P_{n_k}^B J_{2\odot}^T
\]

(6)
3) Loop closing candidates: Each new pose of a scan is compared against the previous scan poses that are in the nearby area defined by a threshold. Whenever enough points are overlapping, a new scan matching introduce a constraint between the poses, updating the ASEKF. These constraints close the loops that correcting the whole trajectory and bounding the drift. Finally the scan matching result is used to update the filter.

4) Scan matching: In order to execute the pIC algorithm, given two overlapping scans \( (S_i, S_n) \) with their related poses \( (\hat{x}_i^B, \hat{x}_n^B) \), an initial guess of their relative displacement is necessary. This initial guess \( [q_{i}^{I}, P_{i}^{I}] \) can be easily extracted from the state vector using the tail-to-tail transformation [15]:

\[
q_{i}^{I} = \odot \hat{x}_i^B \oplus \hat{x}_n^B
\]

(7)

Since the tail-to-tail transformation is actually a non-linear function of the state vector \( \hat{x}_i^B \), the uncertainty of the initial guess can be computed by means of the Jacobian of the non-linear function:

\[
P_{i}^{I} = H_k P^B_i H_k^T
\]

(8)

where

\[
H_k = \left. \frac{\partial \odot \hat{x}_i^B \oplus \hat{x}_n^B}{\partial x_k} \right|_{x_k^B=x_k^P}
\]

(9)

Moreover, as shown in [15], the Jacobian for the tail-to-tail transformation \( x_{a_i} = \odot x_{b_i} \oplus x_{n_i} \) is:

\[
\frac{\partial \odot x_{b_i} \oplus x_{n_i}}{\partial (x_{b_i},x_{n_i})} = [J_{1\odot}, J_{2\odot}, J_{3\odot}]
\]

(10)

where the \( J_{1\odot}, J_{2\odot} \) and \( J_{3\odot} \) are the Jacobian matrices of the compounding and inverse transformations respectively.

Being in our case \( \hat{x}_i^B \) and \( \hat{x}_n^B \) components of the full state vector, the Jacobian of the measurement equation becomes:

\[
H_k = \left. \frac{\partial \odot \hat{x}_i^B \oplus \hat{x}_n^B}{\partial x_k} \right|_{x_k^B=x_k^P}
\]

\[
= \begin{bmatrix}
J_{2\odot} & 0_{3\times3(n-i-1)} & 0_{3\times3(i-1)}
\end{bmatrix}
\]

(11)

Once the initial displacement guess is available, the pIC algorithm can be used to produce an updated measurement of this displacement.

5) State update: When two overlapping scans \( (S_i, S_n) \) with the corresponding poses \( (x_i^B, x_n^B) \) are registered, their relative displacement defines a constraint between both poses. This constraint can be expressed by means of the measurement equation, which again in our case becomes:

\[
z_k = \odot \hat{x}_i^B \oplus \hat{x}_n^B
\]

(11)

where \( \hat{x}_i^B \) is the scan pose which overlaps with the last scan pose \( \hat{x}_n^B \). Now, an update of the stochastic map can be performed with the standard extended Kalman filter equations.

IV. SPARUS AUV

Sparus AUV (fig. 4), was developed in the underwater robotics lab at Universitat de Girona (UdG) mainly for participating in the Student Autonomous Underwater Challenge - Europe (SAUC-E) 2010. After becoming the winning entry of the competition, Sparus has been used as a research platform to develop and test new algorithms as well as to collect datasets in real environments. Recently, in SAUC-E 2011, it achieved the second place [16].

A. Mechanical design

Sparus AUV, was designed with the main goal of having a small and simple torpedo-shaped vehicle with hovering capabilities. It has three DoF and the propulsion consists of three thrusters: two for the surge and yaw DoFs, one for the heave DoF and are integrated in a classical torpedo shape AUV. The mechanical structure and components are therefore organised around this configuration. The front of the vehicle contains all the sensors and the battery housing while in the back there is a second housing for the electronics, the computer and the inertial navigation system.

The main structure is made of aluminium profiles and stainless steel clamps that hold the two pressure housings (fig. 5a). The electronics and battery housings were made of aluminium, rated for 100m depth. To give the vehicle the required buoyancy, technical foam is distributed all over the top part of the vehicle in order to place the buoyancy centre at the same longitudinal position as the gravity centre but above it, assuring pitch and roll stability (fig. 5b). Finally, to reduce the water drag and to protect the components, a two-part ABS skin covers the AUV (fig. 5c). The final dimensions of the vehicle are 1.22 m length by 0.23 m diameter, and the weight is around 30 kg.

B. Hardware design

The vehicle’s power module consists of one battery pack of 25.9 V and 34.4 Ah of nominal capacity, which allows for an autonomy of more than 6 hours. The pack is composed by 112 Lithium ion battery cells (18650 type, 3.7 V, 2150 mAh). Additionally, an umbilical cable can be connected to the vehicle’s computer housing for external power and Ethernet access.

The on-board embedded computer has been chosen as a trade off between processing power, size and power consumption. The ADL945HD board together with the U2500 board...
processor at 1.2 GHz provides the processing power of a Core Duo architecture together with the Ultra Low Voltage (ULV) consumption and the 3.5” small form factor.

The vehicle is equipped with a complete sensor suite composed by two colour video cameras (forward-looking and down-looking), a MRU MTi from XSens Technologies, a Micron imaging sonar from Tritech, an echosounder, a pressure sensor and the NavQuest 600 Micro DVL from LinkQuest which also includes a compass/tilt sensor. Additional temperature, voltage, pressure sensors and water leak detectors are installed into the pressure vessels for safety purposes. Besides, the vehicle hosts on the top a WiFi and a GPS antenna covered with resin, which can be detached and placed in a float on surface, keeping the connection with the AUV via a 5 meter USB cable.

C. Software architecture

The software architecture currently running on Sparus was developed from scratch for the SAUC-E 2010, although is fully compatible with the previous robots in the underwater robotics laboratory. The COLA2 architecture (Component Oriented Layer-based Architecture for Autonomy) is an hybrid control architecture divided in three layers, namely mission, execution and reactive layers. The reactive layer is composed by the Vehicle Interface, Perception and Control and Guidance modules (fig. 6) where the different software components (drivers, processing units, primitives, etc.) communicate using a custom-designed communication protocol. Over these three modules there is the Mission Control System which constitutes the mission and execution layers.

The implemented communication system is based in standard XML strings over TCP/IP connections, resulting in a lightweight protocol which allows to perform the component communication in a plain and simple way. By using this protocol, the architecture can be network-distributed among different computers allowing, for example, the execution of some components inside the vehicle’s computer and some others in external PC’s.

V. EXPERIMENTAL SET-UP AND RESULTS

A. The dataset

The method described in section III, has been tested with a dataset obtained in a natural underwater tunnel located in the Costa Brava area, Spain (fig. 7a).

Although there is enough vertical information for a MSIS sensor, it is a challenging unstructured environment as the walls and the depth of the tunnel are uneven (fig. 7b). The entrance of the tunnel starts at around 10 m depth and reaches almost 18 m at the end of a 30 m corridor.

The survey mission was carried out using the Sparus AUV guided along a 150 m path. The MSIS was configured to scan the whole 360° sector and it was set to fire up to a 50 m range, with a 0.1 m resolution and a 1.8° angular step. Dead-reckoning was computed using the velocity reading coming from the DVL and the heading data obtained from the MRU sensor, both merged using the described EKF. Standard deviation for the MSIS sensor was set as it is specified by the manufacturer, 0.1 m in range and 1.8° in angular measurements.
Fig. 7. Gathering the dataset: a) The underwater tunnel is pointed by the dotted red lines. b) The depth profile of the AUV trajectory.

Fig. 8. SLAM Trajectory and map. In red (solid) line is the dead reckoning and in blue (dash) the trajectory estimated with the SLAM algorithm.

B. Results

The vertical beam width of the MSIS is $30^\circ$, which eliminates the vertical resolution on the sensor (2D scans). That was not introducing any problem in our previous work in the marina environment [5], as the man-made wall were straight vertical. In this natural dataset, the rough and unstructured walls, as much as the vertical extension of the tunnel, breaks the continuity of the sonar images from scan to scan. However, the proposed algorithm was able to cross-register 36 from the 102 scan poses. Figure 8, shows the trajectory and the map estimated with the proposed SLAM algorithm. As was expected, the dead-reckoning estimated trajectory suffers from a significant drift which is drastically limited by the proposed SLAM algorithm. Unfortunately, in such environment is very difficult to obtain a ground truth but this is included in our future list work, as is the only way to validate the results. The whole dataset was acquired in 23 min with the vehicle travelling at 0.2 m/s average speed and the off-line execution of the proposed algorithm, implemented in MATLAB, needs around 4 min at an Intel Core2 Quad @ 3.00 GHz CPU, which gives good possibilities for real time implementation.
VI. CONCLUSION

This paper proposes a pose-based algorithm to solve the full SLAM problem for an AUV, navigating in an unknown and possibly unstructured environment. A probabilistic scan matching technique using range scans gathered from a MSIS is used together with the robot dead-reckoning displacements. The proposed method utilizes two EKFs. The first, estimates the local path travelled by the robot while forming the scan as well as its uncertainty, providing position estimates for correcting the distortions that the vehicle motion produces in the acoustic images. The second is an augmented state EKF that fuse in-line the raw data from the sensors and estimates and keeps the registered scans poses.

The algorithm has been tested with the Sparus AUV, guided along a 150 m path within an underwater tunnel, which is a challenging unstructured environment. Although the vertical extension of the tunnel is difficult to be sensed with a MSIS, our proposed algorithm was able to cross-register 30% of the total poses, and constrain the dead reckoning drift.

ACKNOWLEDGMENT

The authors would like to thank all the members of the underwater robotics lab at UdG for their efforts to make operational the Sparus AUV on time and particular Arnau and Carles for their invaluable help in the field. This research was sponsored by the Spanish project DPI2008-06548-C03 (RAUVI), and two European Commission’s Seventh Framework Program 2007-2013 projects under grant agreements: ICT-248497 (TRIDENT) and Marie Curie PERG-GA-2010-276778 (Surf3DSLAM).

REFERENCES