Interval methods based AUV localization in the context of an acoustic network with experimental results

Jan Sliwka, Andrea Munafò, Gabriele Ferri, João Alves
Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy
e-mail: {sliwka,munafo,ferri,alves}@cmre.nato.int

***Abstract*—This article presents an application of interval methods for the dynamic localization of an autonomous underwater vehicles (AUV) using acoustic measurements of range and/or bearing to one or more underwater or surface nodes. A node can be either a vehicle or a buoy equipped with an acoustic modem and positioning aid. The position and speed vector of each node is communicated acoustically underwater to all of the other nodes. Popular methods used in solving localization problems use a probabilistic approach (e.g. Kalman filtering or Particle filtering). While such methods consider the probabilistic distribution of variables, the interval methods consider sets of possible values of a variable. The interval methods are deterministic in nature and have certain advantages over probabilistic methods such as native handling of non linear equations or solving problems of high dimension, guaranteed result under the right assumptions, and handling inconsistent data. The interval localization method is tested on a dataset acquired during an experiment on the Mediterranean sea.**

I. Introduction

Navigation is fundamental to accomplish any kind of autonomous robotic mission. The navigation of an autonomous underwater vehicle (AUV) is ensured using various technologies [4], [7], [10]. One of them is the Doppler Velocity Log (DVL) which can provide a measurement of the vehicle's speed with respect to the water column or sea floor if in range. The DVL can be used to compute the current robot position by calculating its displacement from an initial known position. Other alternatives are to use acoustic positioning such as the Long Base Line (LBL), Short Base Line (SBL) or Ultra Short Base Line (USBL). The advantage of the second approach is that the positioning error does not accumulate with time. Another important factor in a mission is communication. This enables exchanging messages containing, for example, orders from the command centre, telemetry from AUVs and, more importantly, positioning data. This is typically achieved using acoustic modems. Among the objectives of the Cooperative Anti Submarine Warfare programme (CASW) at the NATO Centre for Maritime Research and Experimentation (CMRE) is the creation of underwater networks enabling simultaneous navigation and communication. In this context, multiple underwater vehicles and multiple surface nodes cooperate to detect targets or provide support to the vehicles. Not all of the vehicles are necessarily outfitted with expensive equipment (INS, DVL, USBL). Additionally, not all equipment works in all conditions. For example the DVL does not work well in deep waters when the sea bottom is out of range (no bottom lock). An LBL-like acoustic navigation aid can be then used to cope with this DVL limitation. An LBL-like scheme which is similar to a classic LBL but allowing both range acquisition and communication is explained in more detail in a companion paper in this conference [5]. The objective of the present paper is to present the method used to compute the position of any vehicle which acquires range and/or bearing measurements to at least one node with a known position. The position of the vehicle is computed using interval methods [2], [3], [8]. Existing methods generally use probabilistic approaches (e.g. Kalman filtering or Particle filtering) [1], [11], [6], [9]. While these methods consider the probabilistic distribution of variables, the interval methods consider sets of possible values of a variable. As such, the range measurements as well as vehicle speed and heading are modeled as intervals. The solution of the localization problem is the set of positions consistent with the range measurements (not necessarily all of them) and the evolution with time of the AUV. The interval methods are deterministic in nature and have different advantages over probabilistic methods such as native handling of non-linear equations or solving problems of high dimension, guaranteeing results under the right assumptions and dealing with outliers in the data. The localization method is tested on a dataset acquired during the COLLAB13 sea trial on the Mediterranean sea.

Section II of the paper makes an introduction to the interval methods. Section III deals with the implementation. Section IV presents experimental results.

II. Understanding interval methods

A. Mathematical theory

The mathematical problem solved here is the relaxed constraint satisfaction problem or relaxed CSP [3]. A relaxed CSP is a set of equations which only a part has to be satisfied.

A CSP can be denoted in the following way

$$\begin{array}{ll} i \in \{1,..,n\}, \mathbf{f_i} : \mathbb{R}^n \to \mathbb{R}^m & \\ C_i : \mathbf{f_i}(\mathbf{x}) = \mathbf{y_i} & (constraints) \\ x \in \mathbb{X}_0 & (unknowns) \\ \mathbf{y_i} \in [\mathbf{y_i^-}, \mathbf{y_i^+}]. & (data) \end{array} \quad (1)$$

Example

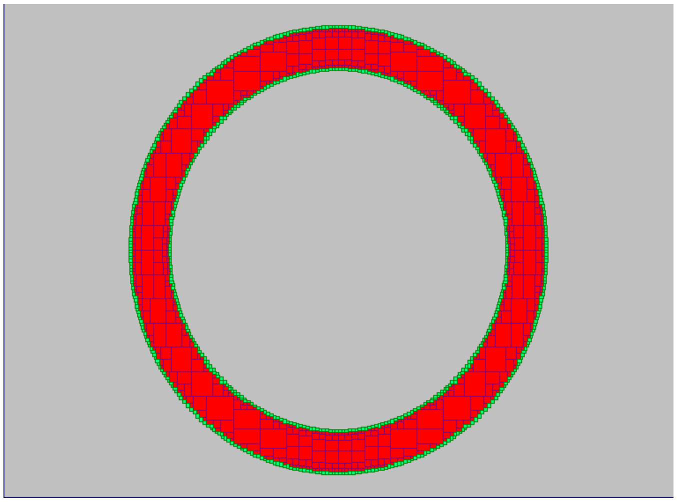


Figure 1. Solution of the ring constraint using Proj2D interval solver

$$\begin{array}{l} x_1^2 + x_2^2 = r^2 \\ (x_1, x_2) \in \mathbb{R}^2 \\ r \in [r^-, r^+]. \end{array} \tag{2}$$

another notation for that CSP would be

$$\begin{array}{l} \sqrt{x_1^2 + x_2^2} \in [r^-, r^+] \\ (x_1, x_2) \in \mathbb{R}^2. \end{array} \tag{3}$$

For each constraint of the CSP it is possible to compute the set of points (from the initial set of unknowns) which satisfy that constraint. As an example, the set of points which satisfy the constraint defined in the example is a ring. Figure 1 shows the solution of a ring constraint for a given radius interval.

The solution of a relaxed CSP with several constraints is several sets. Each of those sets is a set of points satisfying a specific number of constraints. Those sets can be obtained by intersecting in a specific way the sets of points which satisfy only one constraint. Figure 2 shows an example solution of a relaxed CSP. Consider 6 constraints where 2 of them are inconsistent with the rest. For each constraint it is possible to define the set of points which satisfy that constraint (here, the $\mathbb{X}_i$ sets). Using those sets it is possible to define the sets of points which satisfy a specific number j of constraints (here, the sets $\mathbb{Y}_j$). Each $\mathbb{Y}_j$ set corresponds in fact to a relaxed-intersection of the $\mathbb{X}_i$ sets. A q-relaxed intersection of a number of sets, q being a natural number, is a set containing points which belong to all the sets except q. In the example, the sets $\mathbb{Y}_6$ and $\mathbb{Y}_5$ are empty since 4 is the maximum number of constraints which can be satisfied by any point of space. In practice, we are interested in computing only a subset of the $\mathbb{Y}_j$ sets. A good choice is the set of points satisfying most of the constraints (here $\mathbb{Y}_4$). The algorithm is presented in the next section.

B. Algorithm Concepts

Before explaining the algorithm we define several concepts.

Box: a box is an interval in an n-dimensional space (also called hyper-rectangle or orthotope). Figure 3 shows a representation of it and its mathematical notation.

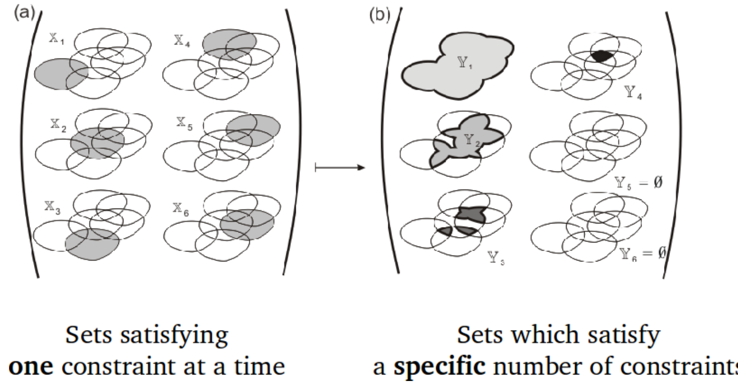
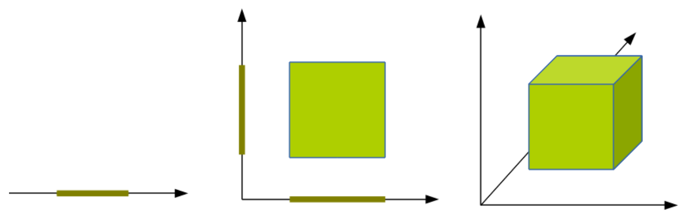


Figure 2. Relaxation concept



$$\begin{array}{rcl} [\mathbf{x}^-, \mathbf{x}^+] & = & ([x_1^-, x_1^+], [x_2^-, x_2^+], .., [x_n^-, x_n^+]) \\ [\mathbf{x}] & = & ([x_1], [x_2], .., [x_n]) \end{array}$$

Figure 3. A box

Contractor : A contractor is an operator which basically shrinks a box around a specific set. Figure 4 shows an example of a disc contractor. The contractor is an algorithmic representation of a constraint or a set of constraints. The contractor is in fact used to compute the set of points that satisfy one or a set of constraints.

A mathematical definition of a contractor can be found below.

Definition: The operator $C : \mathbb{IR}^n \to \mathbb{IR}^m$, $\mathbb{IR}$ being the set of intervals of $\mathbb{R}$, is a contractor if

(i)$\forall [\mathbf{x}] \in \mathbb{IR}^n, C([\mathbf{x}]) \subset [\mathbf{x}]$ (contractance)

(ii) $(\mathbf{x} \in [\mathbf{x}], C(\{\mathbf{x}\}) = \{\mathbf{x}\}) \Rightarrow \mathbf{x} \in C([\mathbf{x}])$ (consistency)

(iii) $C(\{\mathbf{x}\}) = \emptyset \Leftrightarrow (\exists \varepsilon > 0, \forall [\mathbf{x}] \in B(\mathbf{x}, \varepsilon), C([\mathbf{x}]) = \emptyset)$ (convergence)

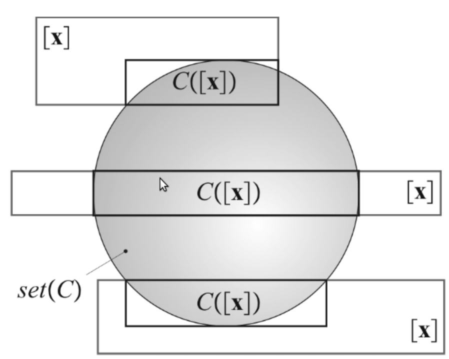


Figure 4. Three cases of contraction of a box **[x]**

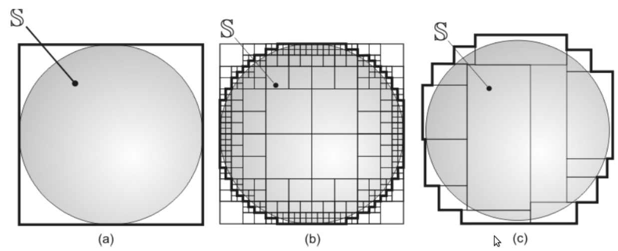


Figure 5. Subpaving

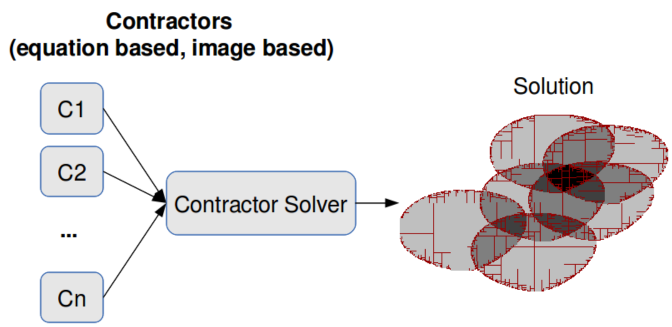


Figure 6. Building one contractor out of many then calling the solver

B(**x**,ε) is the ball of radius ε and centre **x.**

Subpaving: The mathematical theory deals with continuous sets. A generic way to represent a bounded continuous set on a computer is to use a subpaving, *i.e.* a set of non overlapping boxes as shown in Figure 5.

Solution of a relaxed CSP: The solution of the relaxed CSP is several subpavings, one for each satisfaction requirement. In fact, for each set of contractors $C_1, ..., C_n$ it is possible to construct one contractor which allows computing the subpaving approximating the set of points satisfying a specific number of constraints. This concept is represented in Figure 6.

C. Contractor solver

The contractor solver as a whole can be decomposed into several subroutines.

- The subroutine which generates a subpaving for a given contractor (the algorithm is explained below)
- The subroutine which makes one contractor out of many (using contractor intersection or relaxation in case of outliers)
- The subroutine which generates the contractors from the data received by the algorithm

Algorithm: Consider the following parameters:

- initial box
- number of bisections
- precision box
- fixed point variation threshold
- fixed point iterations
- scale box (only used when the unknowns are of different nature like positions in meters and angles in radians)

The algorithm uses those parameters as well as one contractor to build the solution in the form of a subpaving.

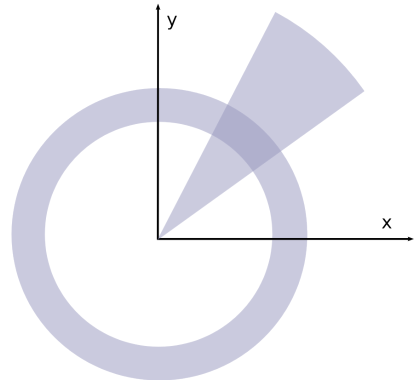


Figure 7. Contractor sets for the range and bearing constraint

1) First we build a contractor which defines the solution set we are searching for. The contractor can be build using multiple contractors each one coming from a constraint corresponding to a sensor measurement for example.
2) We consider an **initial box** and a subpaving which approximates the mathematical solution set (the subpaving is the output of the algorithm). Initially the subpaving is formed from the **initial box**.

Iteratively, the algorithm takes a box from the subpaving (which has not been processed yet) and contracts it using the contractor. The contractor does not immediately converge to the minimal box and needs to be called many times to do so. This step is stopped when the contractor has been called **fixed point iterations** times *or* when the box does not change much after contraction (box width variation less than **fixed point variation threshold**). If the box is smaller than the **precision box** it is not processed anymore and remains in the subpaving. Otherwise, the box is bisected i.e. split into two boxes along the bisection axis. The bisection axis is usually equal to the largest axis of the scaled box using the **scale box**.

1) The algorithm stops when all the boxes are small enough or when the **number of bisections** is reached.

III. Implementation

A. Contractors to solve the problem of localization

Two types of contractors are used to solve the problem of localization:

- The **distance contractor**: given r_{AB} the distance between to points A and B the contractor is based on the following constraint

$$distance(A(x_a, y_a), B(x_b, y_b)) = r_{AB} \tag{4}$$

- The **angle contractor**: given θ_{AB} the argument of the vector $\overrightarrow{AB}$ the contractor is based on the following constraint

$$arg(\overrightarrow{A(x_a, y_a)B(x_b, y_b)}) = \theta_{AB} \tag{5}$$

$x_a, y_a, x_b, y_b, r_{AB}, \theta_{AB}$ are all uncertain and belong to the intervals $[x_a], [y_a], [x_b], [y_b], [r_{AB}], [\theta_{AB}]$. The intervals $[x_a], [y_a], [x_b], [y_b]$ are contracted. Figure 7 show a representation of the contractor sets.

B. Evolution function

The contractor links the coordinates of the nodes at a time step k with the distance/bearing measurement made at the same time step. In order to use this contractor at a posterior time one has to take into consideration the transformation of coordinates between the two time-steps. This transformation is given by the evolution equation of each node.

$$\begin{aligned} x_{k+1} &= x_k + (v * cos(\psi) + v_{cx}) * dt \\ y_{k+1} &= y_k + (v * sin(\psi) + v_{cy}) * dt \end{aligned} \quad (6)$$

where (x_k, y_k) are the coordinates of an arbitrary node at a time step k, v its speed, ψ its heading, (v_{cx}, v_{cy}) the current it is subject to and dt the model integration time step. The uncertainty due to the current can be included in the uncertainty on speed and heading.

If T is the total coordinate transformation between two instants t_1 and t_2 then the contractor at a time t_2 obtained from a contractor valid for the instant t_1 is obtained using the following formula

$$C_2 = I \cap T \circ (I \cap (C_1 \circ T^{-1})). \quad (7)$$

I being the identity contractor.

All the contractors coming from sensor data are transformed to the same time t before being used in the contractor solver. The contractor solver returns the set of possible positions of the vehicle at the time t.

Note: An unknown node heading is equivalent to supposing that the heading of the node belongs to $[0, 2\pi]$. An unknown node speed is equivalent to supposing that the speed of the node belongs to $[0, v_{max}]$. v_{max} being the maximum speed the vehicle may achieve.

C. Interpretation of the output

1) Subpaving: The output of the interval solver is a subpaving for each satisfaction requirement. This subpaving can be used to compute values of interest as shown in Figure 8 such as

- **Box envelope** (yellow box in the example) : the union of the boxes in the sub-paving
- **Center of the box envelope** (yellow cross in the example) which is used the most as a punctual representation of the solution
- **Barycenter of the boxes** (red dot in the example). Rarely used since it requires some computation time and the gain is not that big.
- **Number of disjoint solutions** (there are two disjoint subpavings in the example). This might be useful to detect and deal with the ambiguities by tracking the ambiguous solutions separately.

Note that in our case, since there are no outliers, we compute the subpaving for the set of points satisfying all of the constraints.

2) Satisfaction number: If there are outliers, not all the constraints are satisfied. The solution for a satisfaction requirement of n (if n is the number of constraints) will most likely yield an empty set.

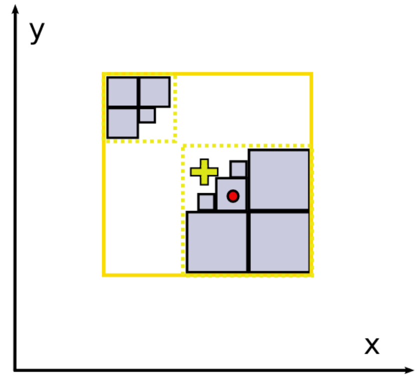


Figure 8. Subpaving interpretation

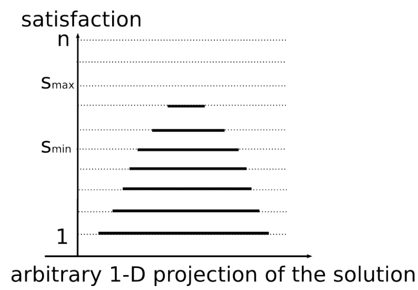


Figure 9. Solution in the presence of outliers

We define by satisfaction number an estimate of the number of constraints which are actually satisfied. Figure 9 shows 1-D projection of solver solution set for different degrees of satisfaction. Interval algorithms do not provide that number but only an upper and lower bound. The upper bound is obtained naturally using the solver with the contractor cited in this document. The lower bound requires additional computations which are usually avoided.

By tracking the satisfaction number and making sure for example it does not go lower than a certain value (s_{min} in the example) it is possible to know if the vehicle is "lost." It is also possible to detect the constraints which are inconsistent with the rest of the constraints (labeling them as outliers). Once detected, the constraints might be removed from the set of valid constraints (valid measurements) thus saving some computation time. Detecting inconsistent constraints does not necessarily improve positioning.

IV. Experimental results

A. Context

The range data was acquired during the COLLAB13 sea trial in which 2 AUVs (Harpo and Groucho) depicted in Figure 10 were deployed from CMRE's research vessel NRV Alliance. Each of the three vehicles was equipped with an acoustic modem. The status information (position, speed and heading) of all nodes are sent regularly through the acoustic network. The ground truth for the position of the AUVs is given by DVL navigation which provides a position with an error of 0.1% of the overall distance crossed by the AUV (i.e. 1 m in every 1 km). Crude estimation of the vehicle velocity vector (200 times less precise than the DVL, i.e. an

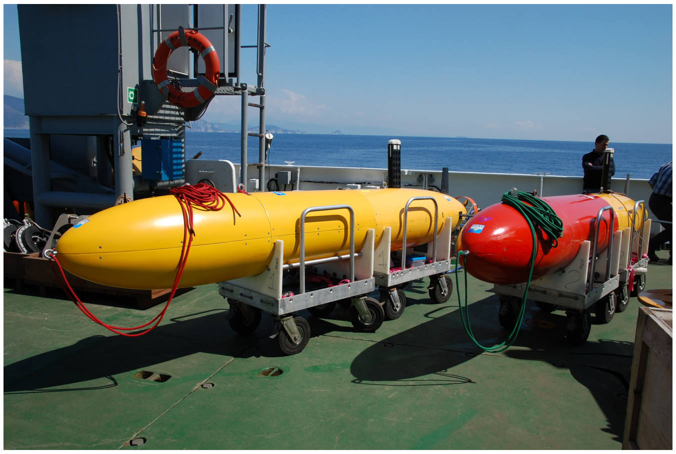


Figure 10. Harpo and Groucho AUVs on Alliance

error of about +/- 0.2 m/s) is used as input for the positioning algorithm.

B. Using 2 reference nodes

The problem solved here is computing the position of a node knowing the dynamics of the localized node and the positions of two other nodes. The curves in Figure 11 shows the reconstructed trajectories for all the nodes computed for 3 scenarios (one per node). In each scenario the position of 2 out of 3 nodes is provided and the estimated trajectory as well as the ground truth are plotted. Note that the ship was stationary.

Note: the algorithm does not just track the position, but performs a global localization as well (i.e. computes the initial position from a large initial set). In other words there is no need to know the initial position.

The algorithm execution was real-time. The example does not have very high computation requirements since the measurement rate was no more than 3 measurements per minute. The algorithm can run hundreds of times faster on a single Intel 2GHz core.

The interval algorithm parameters were the following:

- **Desired precision** for set estimation: 1 m (another parameter defining the precision is the number of bisections)
- **Number of bisections** (equivalent to algorithm resolution of the result) was equal to 100.
- **Number of measurements** taken into consideration for each step (history): 100

The curve in Figure 12 show the absolute localization errors for all nodes. The first errors are big since there are not yet enough data to localize the node and/or resolve all the ambiguities. This phase could be called the global localization phase. The error then rapidly converges to a value depending on the overall range measurement error (which was on the order of 100 meters). Note that when the position converges, it is possible to search for the solution locally, in other words, track the solution. Tracking the solution does consume less resources. The peak in the error at the acoustic measurement number 150 is due to the ambiguity of the situation (i.e. due to

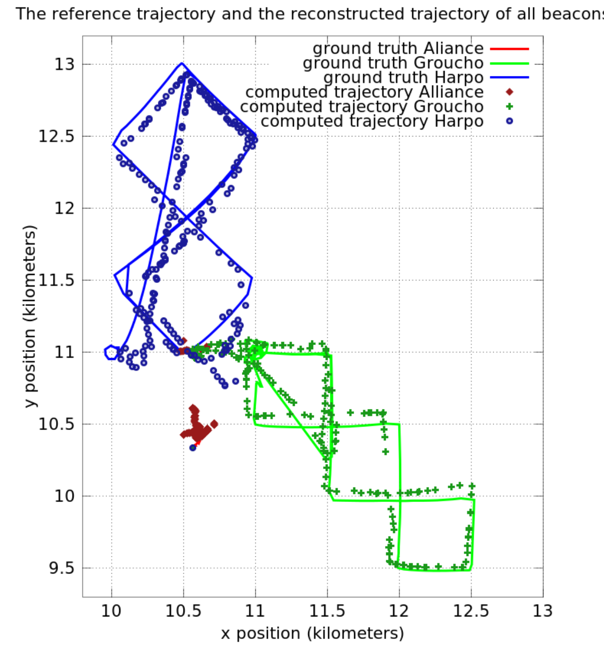


Figure 11. Trajectory reconstruction using interval methods for the COLLAB13 data set

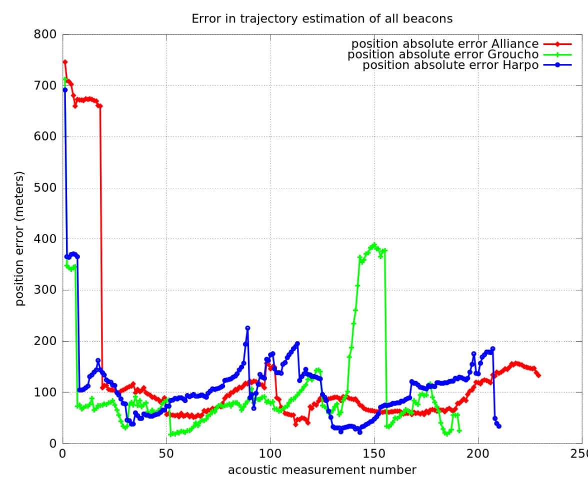


Figure 12. Errors in trajectory estimation

the existence of multiple solutions). The localized AUV was in fact exactly in between two nodes (alignment).

C. Using one reference node

Here we show that it is possible to localize an AUV using range measurements and the relative speed vector between the AUV and one node (given by the GPS of the ship and sensors on board the AUV). Figure 13 shows the result of localization for Groucho AUV using range measurements to the ship only. The solution set (cherry color) contains the real position of the AUV (light blue dot). The localization using one node works better when the trajectory of the AUV is perpendicular to the AUV/Ship line. That was the case in the Figure13.

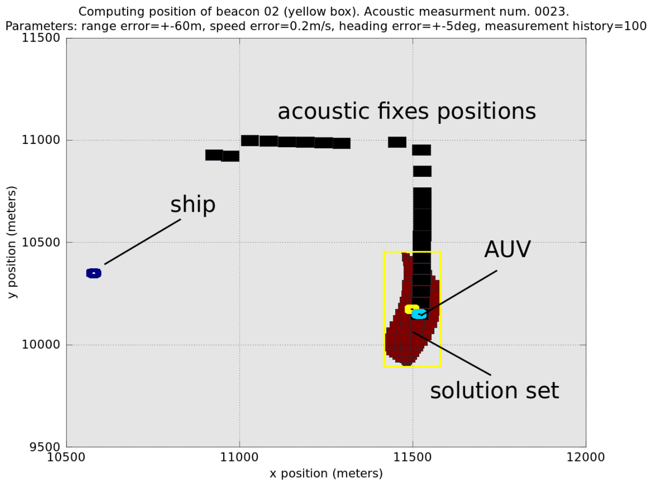


Figure 13. Localization Groucho using range measurements to the ship

V. Conclusion

In this paper we showed an application of interval methods for the localization of AUVs. We showed that a problem of localization can be set up as a set of equations which, in case of the absence of outliers, must all be satisfied. The interval algorithm computes the set of points satisfying those equations from an initial set representing the area of the sea trial.

Future work will focus on showing the full capabilities of interval algorithms. One of the steps is to augment the dimension of the problem by rising the number of parameters to estimate. An example of problems we are working on is the problem of localization of a constellation of vehicles. The point is to consider the positions of all the vehicles, represented in a wisely chosen frame of reference, as unknowns and add them to the list of estimated variables. Information such as relative range and bearing or a GPS fix when a vehicle is on the surface allow to solve the problem provided that there is enough data to do so. We are also working on an improvement to the localization method by adding vehicle speed and heading, sea current and other parameters (e.g. the speed of sound) to the list of estimated variables.

Acknowledgement: The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7-ICT-2011-7) under grant agreement no. 288704.

References

[1] Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics.

[2] Jaulin, L.,Kieffer M., Didrit O. and Walter E. (2001). Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics.

[3] Sliwka J. (2011). Using set membership methods for robust underwater robot localization. Phd Thesis. Universite de Bretagne Occidentale, Brest.

[4] Paull, L., Saeedi, S., Seto, M., & Li, H. (2014). AUV Navigation and Localization: A Review. Oceanic Engineering, IEEE Journal of , vol.39, no.1, pp.131,149, Jan 2014.

[5] Munafo' A., Sliwka J., Ferri G., Vermeij A., Goldhahn R., LePage K., Alves J., Potter J. (2014). Enhancing AUV Localization using underwater acoustic sensor networks: results in long baseline navigation from the COLLAB13 sea trial. OCEANS14 MTS/IEEE, St. John's, Canada, 2014.

[6] Clerentin, A., Delafosse, M., Delahoche, L., Marhic, B., and Jolly-Desodt, A. (2008). Uncertainty and imprecision modeling for the mobile robot localization problem. Autonomous Robots , 24(3):1573-7527

[7] Eustice, R. M., Whitcomb, L. L., Singh, H., & Grund, M. (2007, April). Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles. In Robotics and Automation, 2007 IEEE International Conference on (pp. 4257-4264). IEEE.

[8] Caiti, A., Garulli, A., Livide, F., and Prattichizzo, D. (2002). Set-membership acoustic tracking of autonomous underwater vehicles. Acta Acustica united with Acustica , 5(88):648-652.

[9] Maurelli, F., Krupinski, S., Petillot, Y., & Salvi, J. (2008, September). A particle filter approach for AUV localization. In OCEANS 2008 (pp. 1-7). IEEE.

[10] Walls, J. M., & Eustice, R. (2013). An Exact Decentralized Cooperative Navigation Algorithm for Acoustically Networked Underwater Vehicles with Robustness to Faulty Communication: Theory and Experiment. In Robotics: Science and Systems.

[11] Olson, E., Leonard, J. J., & Teller, S. (2006). Robust range-only beacon localization. Oceanic Engineering, IEEE Journal of, 31(4), 949-958.