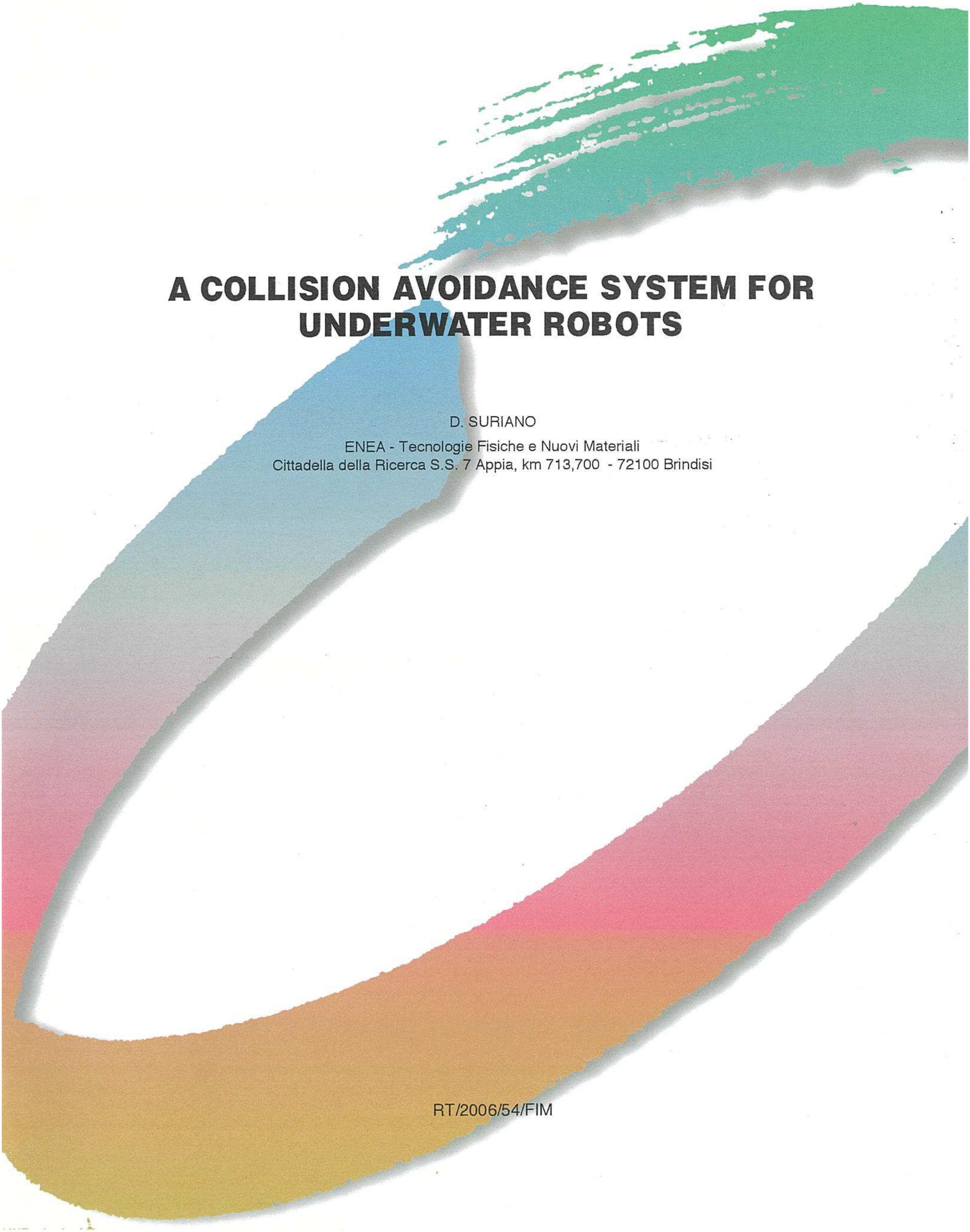
ENEA

ENTE PER LE NUOVE TECNOLOGIE,
L'ENERGIA E L'AMBIENTE

ISSN/0393-3016



A COLLISION AVOIDANCE SYSTEM FOR UNDERWATER ROBOTS

D. SURIANO

ENEA - Tecnologie Fisiche e Nuovi Materiali
Cittadella della Ricerca S.S. 7 Appia, km 713,700 - 72100 Brindisi

RT/2006/54/FIM

ENTE PER LE NUOVE TECNOLOGIE,
L'ENERGIA E L'AMBIENTE

A COLLISION AVOIDANCE SYSTEM FOR UNDERWATER ROBOTS

D. SURIANO

ENEA - Tecnologie Fisiche e Nuovi Materiali
Cittadella della Ricerca S.S. 7 Appia, km 713,700 - 72100 Brindisi

RT/2006/54/FIM

This report has been prepared and distributed by: Servizio Edizioni Scientifiche - ENEA Centro Ricerche Frascati, C.P. 65 - 00044 Frascati, Rome, Italy

A COLLISION AVOIDANCE SYSTEM FOR UNDERWATER ROBOTS

D. SURIANO

Abstract

The main goal for a mobile robot navigation system is to guide the vehicle safely in the presence of obstacles toward some goal point. In order to fulfil this aim, in this paper is proposed a solution suitable for autonomous underwater vehicles (AUV) equipped with echo sounders. The collision avoidance system described here, is based on the range readings of these ultrasonic devices and it is not dependent by the number of sensors aboard the vehicle. The system builds the environment map in which the vehicle acts through the sonar range readings, then, provided by the map informations, sets the new vehicle heading, evaluating the minimum value of a particular merit function. The efficiency of the collision avoidance system is investigated through simulations in which are reproduced several and various submarine scenarious and situations: closed submarine canyons, dense obstacles sea areas, as well as contrived situations like small labyrinths.

Keywords: *Obstacle avoidance, Underwater vehicles, Map building, Ultrasonic sensors, Autonomous navigation*

Riassunto

Il principale obiettivo per un sistema di navigazione di un robot semovente è quello di guidare il veicolo verso un punto prefissato in maniera sicura in un ambiente con presenza di ostacoli. Al fine di perseguire questo obiettivo, in questo documento viene proposta una soluzione adatta ai veicoli autonomi subacquei equipaggiati con sensori ad ultrasuoni. Il sistema anti-collisione proposto costruisce una mappa dell'ambiente in cui il veicolo interagisce mediante le rilevazioni dei sonar di bordo, quindi elabora la direzione di marcia che il veicolo intrprenderà, mediante la ricerca del minimo di una particolare "funzione di merito". L'efficacia di questo sistema anti-collisione è provata tramite simulazioni che riproducono diversi scenari sottomarini virtuali (per esempio: canyon sottomarini a fondo cieco, aree marine dense di ostacoli, o anche scenari artificiosi come piccoli labirinti subacquei).

Parole chiave: aggiramento ostacoli, veicoli subacquei, sotruzione di mappe, sensori ad ultrasuoni, navigazione autonoma

INDICE

A COLLISION AVOIDANCE SYSTEM FOR UNDERWATER ROBOTS

1. INTRODUCTION

The main goal for a mobile robot navigation system is to guide the vehicle safely in the presence of obstacles toward some goal point. A collision avoidance system (or obstacle avoidance system) is in charge of fulfil this aim. It consists of a set of sensors that provides the data concerning the current robot position, the obstacles shapes and locations, and of a procedure that computes the right robot direction in order to avoid obstacles and lead the vehicle toward its goal.

Due to the water physical properties, in the field of underwater robots, the sensorials devices that provide information about the vehicle surrounding environment are sonars, echo sounders and other acoustical devices. In several collision avoidance systems the raw data provided by the sonars or by the echo sounders are stored in a structurated way forming the map of the robot external world. The building of the vehicle surrounding environment map provides a world model on which an obstacle avoidance algorithm can be based.

This paper introduces a collision avoidance system and a map buiding method for autonomous underwater vehicles (AUV), both based on the range measurements of echo sounders devices, in an a priori unknown environment. The approach that will be presented is suitable for all the underwater vehicles equipped with echo sounders; moreover we have investigated the efficiency of our system through simulations based on the model of the Enea AUV Sara [1]. A schematic view of the collision avoidance system is depicted in figure 1.

Basically the system is composed by the map builder and the collision avoidance controller. The map builder takes the raw sonar data and the vehicle position by which builds and continuously updates the world map. The collision avoidance controller selects the desired vehicle bearing by evaluating the vehicle position, the goal position and through the world map informations.

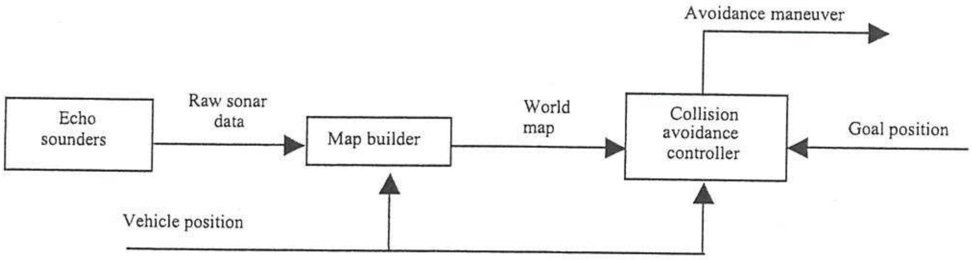


Fig. 1 - Collision avoidance system overview

This paper is organized as follows: a brief survey of relevant obstacle avoidance methods is presented in section 2, in the section 3 is introduced the map building method, then the collision avoidance algorithm is explained in the section 4. The simulator by which is tested the collision avoidance system is presented in section 5, then in section 6 will be shown some simulations results.

2. SURVEY OF OBSTACLE AVOIDANCE METHODS

There are a rich variety of algorithms for obstacle avoidance. An approach to the problem is based on the obstacles edge detection [2,3]. In this method, the algorithm tries to determine the position of the vertical edges of the obstacle and consequently attempts to steer the robot around either edge. The line connecting the two edges is considered to represent one of the obstacle's boundaries. A disadvantage with obstacle avoidance based on edge detecting is the need of the robot to stop in front of an obstacle in order to allow for a more accurate measurement. A further drawback of edge-detection methods is their sensitivity to sensor accuracy.

The Bug's algorithms [4,5] follow the easiest common sense approach of moving directly towards the goal unless an obstacle is found, in which case, the obstacle is contoured until motion to goal is again possible. In these algorithms only the most recent values of sensorial data are used.

Path planning using artificial potential fields [6] is based on a simple and powerful principle. The robot is consider as a particle that moves immersed in a potential field generated by the goal and by the obstacles present in the environment. The goal generates an attractive potential while each obstacle generates a repulsive potential. Obstacles are either a priori known, (and therefore the repulsive potential may be computed on-line) or on-line detected by the on-board sensors and therefore the repulsive potential is on-line evaluated. Besides the obstacle avoidance functionality, the potential field planning approach incorporates a motion control strategy that defines the velocity vector of the robot to drive it to the goal while avoiding obstacles.

The Vector Field Histogram [7] generates a polar histogram of the space occupancy in the close vicinity of the robot. This polar histogram, that is constructed around the robot's momentary location, is then checked to select the most suitable sector from among all polar histograms sectors with a low polar obstacle density and the steering of the robot is aligned with that direction.

Elastic bands [8] is a framework that combines the global path planning with a real-time sensor based robot control aiming at a collision free motion to the goal. An elastic band is a deformable collision-free path. According to [8], the initial shape of the elastic band is the free path generated by a planner. Whenever an obstacle is detected, the band is deformed according to an artificial force aiming to keeping a smooth path and simultaneously maintaining the clearance from the obstacles. The elastic deforms as changes in the environment are detected by the sensors, enabling the robot to accommodate uncertainties and to avoid unexpected and moving obstacles.

Others less relevant obstacle avoidance techniques developed for AUVs are based on neural networks [9,10]. In [9] K.Ishii, T. Ura and others proposed an approach based on a Kohonen Self-organizing Map [11,12]; while in [10] a multilayer neural network is provided by a range-bearing image from a forward-looking sonar, and also it outputs the rudder angle and the AUV propulsion power.

3. WORLD MAP BUILDING

A fundamental task of a collision avoidance system is the external environment map building. The methods employed to carry out this task depend strongly on the characteristics of sensors that provide raw data. Echo sounders and, more generally, ultrasonic range based sensors, provide good range data but offer poor directionality and another problem is the specular reflections from smooth surfaces. Furthermore systems using these sensors must also cope with frequent misreadings due to ultrasonic noise from external sources or stray reflections from neighboring sensors (the so called "crosstalk" phenomenon).

Obstacle avoidance systems (OAS) and path planners based on a surrounding environment map have to take into account these drawbacks in the building of their map through ultrasonic sensors; on the other hand, this these drawbacks are less critical in the OASs which don't need a world model, i.e. [13,14,15].

Another problem of the OAS based on maps builded in the inertial frame is the sensitivity to the accuracy of the vehicle position valuation: obstacles absolute positions are derived from sonar readings and from the robot position, possible misreadings in vehicle position will be reflected in the valuation of the obstacles location.

We can found some examples of OAS for underwater vehicles based on ultrasonic unidirectional sensors in [9,13,16,17].

The map builder developed for our OAS subdivides the robot surrounding space in cubes. These cubes are named cells. The cells are denoted by their centers positions expressed in the vehicle inertial frame coordinates; furthermore an integer number is associated with those cells in which the echo sounders have detected an obstacle, this number, that we call certainty value (CV accordingly with [7]), indicates the measure of confidence that an obstacle exists within the cell area.

The field of view of an ultrasonic sensor, such an echo sounder, can be represented by a cone, so, when an object is detected, is more likely that this object is closer to the acoustic axes of the sensor than to the periphery of the cone. For this reason, when an echo sounder detects an object at the distance D, only the cell CV which lies at the measured distance D along the sensor cone acoustic axes is incremented (see figure 2, the third dimension given by the Z axes is neglected to semplify things).

Furthermore the CV cells located on the line connecting the incremented cell and the cone origin are decremented. Through the continuous and rapid sampling of the environment while the vehicle is moving, is obtained an obstacle probability distribution like shown in figure 3.

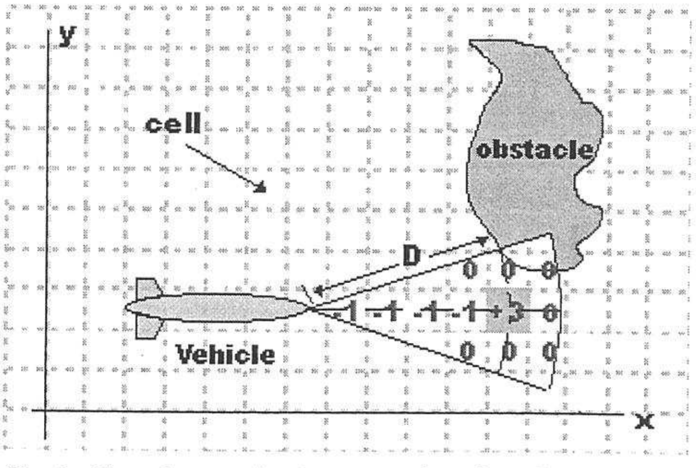


Fig. 2 - The echo-sounder detects an obstacle at the distance D

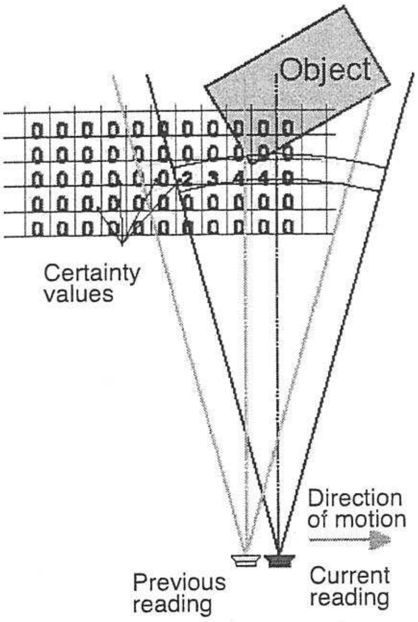


Fig. 3 - A hystogramic probability distribution is obtained by continuous and rapid sampling of the sensors while the vehicle is moving

The maximum CV of a cell is $CV_{max} = 15$, while the minimum is $CV_{min} = 0$; moreover actually the increment value (denoted I_+) is set $I_+=3$, and the decrement value (denoted I_-) is set $I_- = -1$. Note that CV_{max} and CV_{min} have been chosen arbitrarily. I+ was determined experimentally (in relation to CV), by observing that too large a value would make the robot react to single, possibly false readings, while a smaller value would not build up CVs in time for an avoidance maneuver. I_- was determined experimentally and in relation to I_+. This approach for the representation of obstacles is inspired to the one proposed by Borenstein and Koren in [18]. In our work the cells that have a CV greater than zero and adjacent each other are clustered in an entity forming an obstacle (see figure 4). Let $A(x_A,y_A,z_A)$ and $B(x_B,y_B,z_B)$ two cells; the adjacency condition is expressed as:

$$\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2} < \sqrt{3}\ L \ , \tag{1}$$

where L is the cell side length. In the AUV onboard computer memory this entity is stored in the c programming language struct called ‘obstacle’:

```
typedef struct {
      cells_list list;
      int dimension;
      double min[3];
      double max[3]; } obstacle
```

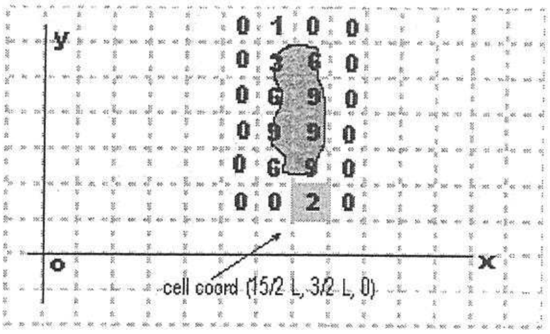


Fig. 4 - representation of a flat obstacle. Assuming L as the cell side length, the coloured cell has coordinates (15/2 L, 3/2 L, 0).

The above structure represents the obstacle entity that is composed by the adjacent cells list with CV > 0 forming the real object shape (fig. 4), by an integer value corresponding with the amount of cells in the list and by two double array defined as:

$$\min[3] = \left(\min_{i=1\ldots N} (x_i^O),\ \min_{i=1\ldots N} (y_i^O),\ \min_{i=1\ldots N} (z_i^O) \right), \tag{2}$$

$$\max[3] = \left(\max_{i=1\ldots N} (x_i^O),\ \max_{i=1\ldots N} (y_i^O),\ \max_{i=1\ldots N} (z_i^O) \right), \tag{3}$$

where O is the obstacle to which the cells belong and N is the number of cells belonging to the obstacle O (i.e. for the object in figure 4, assuming L as the cell side length, it is min[3] = (13/2 L, 3/2 L, 0) and max [3] = (15/2 L, 13/2 L, 0), supposing a flat object shape). In the cell list are stored the tridimensional coordinates of all the cells with a CV > 0 and their own CV values. The entire map is composed by an array of the obstacle structures. It has to be noted that whenever the vehicle, while it is moving, detects that two or more cells belonging to different obstacles are adjacent, those obstacles are fused in only one. In this way the environment tridimensional map is builted and constantly updated.

Finally, it has to make some considerations about the choice of the side cell length L. Suppose that we want to have a number of scans for each cell greater than N_s and the vehicle maximum speed is V_{max}; moreover suppose that the vehicle has a number of S echo sounders which maximum range is R_{max}. If the sound speed in sea water is approximatively $V_s = 1500$ m/s and we can scan only a sensor each time (due to avoid "crosstalk"), each echo sounder will be fired every

$$T_f = 2SR_{max}/V_s \tag{4}$$

seconds (R_{max} in meters). So, in the worst case, if the vehicle is moving at its maximum speed, we have that the side cell length will be:

$$L > V_{max} T_f N_s \ . \tag{5}$$

4. THE AVOIDANCE ALGORITHM

The collision avoidance system that we propose, operates in the vehicle steering plane on the basis of the informations provided by the environmental tridimensional map, by the current vehicle position and by the goal position. The aim of the collision avoidance controller is the computation of the most suitable vehicle heading in order to avoid obstacles and to lead the AUV toward its goal.

Starting from the current AUV position, the controller traces, for each heading on the vehicle steering plane, an immaginary segment (the scanning segment) which length is L_s, in order to verify if it crosses some cell belonging to some map obstacle. Infact cells which have a certainty value CV greater than a threshold T_s, are considered sea areas in which the vehicle never has to navigate. This operation is carried out by the 360° scan partitioning in discreet intervals of Δh degrees in order to limiting the computational load; for this same reason is not appropiate to select L_s much long, but it is also suitable that $L_s > R_{max}$. The Δh and L_s parameters strongly depend on the computational power onboard the vehicle, so they have to defined experimentally.

Another way to reduce the scanning time employed in our algorithm is to use the 'struct' obstacle parameters denoted as 'min' and 'max'. Let $L_1(X_1,Y_1,Z_1)$ and $L_2(X_2,Y_2,Z_2)$ the coordinates of the scanning segment L_s ends, moreover let $X_{Smin} = \min(X_1, X_2)$, $Y_{Smin} = \min(Y_1, Y_2)$, $Z_{Smin} = \min(Z_1, Z_2)$, and $X_{Smax} = \max(X_1, X_2)$, $Y_{Smax} = \max(Y_1, Y_2)$, $Z_{Smax} = \max(Z_1, Z_2)$. Also we suppose that the obstacle O that is going to be scanned has min[3] = $(X_{Omin}, Y_{Omin}, Z_{Omin})$ and max[3] = $(X_{Omax}, Y_{Omax}, Z_{Omax})$, if it results that:

$$X_{Smin} > X_{Omax} \ , \tag{6}$$

or

$$X_{Smax} < X_{Omin} \ , \tag{7}$$

or

$$Y_{Smin} > Y_{Omax} \ , \tag{8}$$

or

$$Y_{Smax} < Y_{Omin} \,, \tag{9}$$

or

$$Z_{Smin} > Z_{Omax} \,, \tag{10}$$

or

$$Z_{Smax} < Z_{Omin} \,, \tag{11}$$

the scanning segment surely don't cross any cell of the obstacle O that is going to be scanned, so the O scanning can be avoided saving computation time.

The threshold setting needs a greater care: if it is too much large, the vehicle may approach the obstacle too closely, on the other hand, if the threshold is too much low, some sonar misreadings risk to affect the vehicle path.

The collision avoidance controller remembers for each heading 'h' the distances of the nearest cells to the vehicle with CV > T_s (if there are) at the time k, denoted with $d_{h,k}$. Then the function used to determine the new vehicle heading is

$$F(h) = A(V_k/d_{h,k}^{\;2}) + B|\,h - h_k| + C|\,h - h_g| \tag{8}$$

where h is the heading being evalueted as a potential new heading, V_k is the current vehicle speed (at time k), h_k is the current heading, h_g is the heading associated with the goal point. A, B, C are gains. The heading at which F(h) is minimum is selected as the new vehicle heading (fig. 5).

The computation of the F(h) function is immediately aborted if the controller detects the presence of some obstacle cell at a distance less then d_s in order to maintain a safety distance between the AUV and the environmental obstacles. In this case the new vehicle heading is:

$$h_{new} = h_{obst} + 180° \,, \tag{9}$$

where h_{obst} is the heading of the nearest obstacle cell which distance to the vehicle is less than d_s. Naturally, the d_s parameter value depends on the geometrical features of the AUV and mostly on the vehicle dynamics such as the AUV minimum turn radius. The tuning of the A, B, C gains in the equation (8) defines the robot behaviour: increasing the 'A' gain causes the vehicle to turn away from obstacles earlier. If 'A' is too large the AUV may have difficulty finding a path out of a dense obstacle field. If it is too small the robot can tend to move straight toward obstacles when a slight turn may be more prudent. Increasing 'B' gain

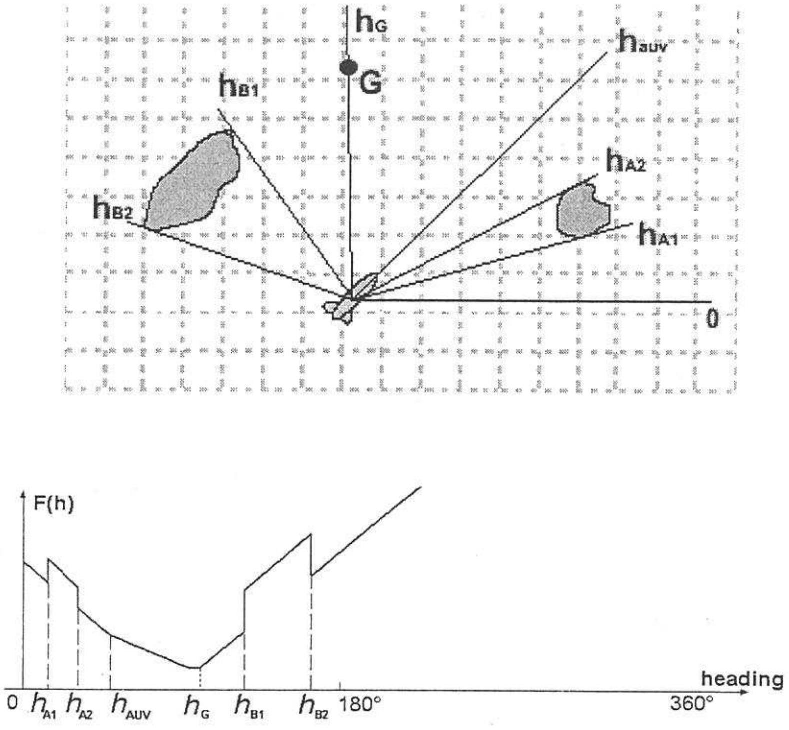


Fig. 5 – a) A typical situation during an AUV mission, here is shown the goal point heading, the current vehicle heading, the reference heading and the obstacles relevant headings. b) The qualitative graph of the merit function relative to the situation shown in fig. 5a

encourages the robot to stay on its current heading for a longer time. If ‘B’ is too small, the vehicle tends to turn too often, performing wasteful maneuvers in the pursuit of negligible benefits. If it is too large, the robot will stubbornly move on its current direction turning only when it find on its course an obstacle cell at a distance smaller than the safety one. Finally, increasing the ‘C’ gain strengthens the attraction of the goal position. If ‘C’ is too small, the robot tends to wander with little effort given towards reaching the goal. If it is too large, the AUV tries to move toward the goal ignoring viable alternative paths when an obstruction is detected.

5. THE SIMULATION ENVIRONMENT

In order to test the avoidance collision system performance, we have used an hardware in loop (HIL) graphical simulator and a model of Enea AUV Sara [1]. A schematic view of the simulator is depicted in figure 6.

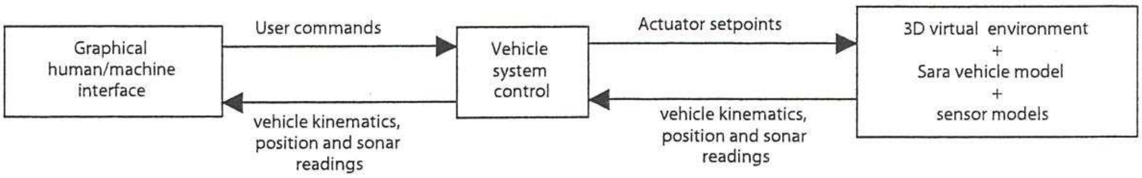


Fig. 6 - HIL simulator overview

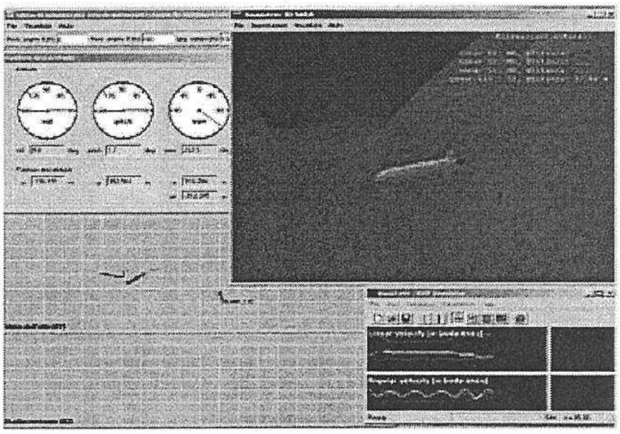


Fig. 7 - A simulator screenshot

The simulator components shown in figure 6 runs on three distinct computers. The human machine interface (HMI) is written in Java and runs on a PC Win 32, while the 3D environment, the vehicle and the sensor models are simulated by a process running on an other PC Win 32. The process source code is written in C++ through the Microsoft Visual Studio IDE. All the vehicle system control is written in C and runs on a Motorola PowerPC 604 board with the LinxOs real time operative system. All the components are networked through a TCP/IP LAN (10 Mb ethernet) (figure 7).

As described in the literature [19] and as it is implemented in our simulator, the hydrodynamic equation of motion of an underwater vehicle with 6 DOF can be conveniently described as follows:

$$(M_{RB}+M_A)\dot{\upsilon} + (C_{RB}(\upsilon)+C_A(\upsilon))\upsilon + D(\upsilon)\upsilon + G(\eta) = \tau_{Thr} \qquad (7)$$

$$\dot{\eta} = diag\{J_1(\eta), J_2(\eta)\}\upsilon \qquad (8)$$

$$\eta = \int \dot{\eta}\, dt \qquad (9)$$

where υ is the vehicle vector linear and angular speeds in the robot fixed coordinate system, M_{RB} is the inertia matrix, M_A is the added-mass matrix, C_{RB} is the rigid body coriolis matrix, C_A is the added coriolis matrix, D is the damping matrix, $G(\eta)$ is the restoring forces balance, τ_{Thr} is the thrusters forces and moments vector, $\eta=[x,y,z,\phi,\theta,\psi]$ is the vehicle absolute position in the earth fixed coordinate system (ϕ,θ,ψ are respectively the robot roll, pitch and yaw angles) and

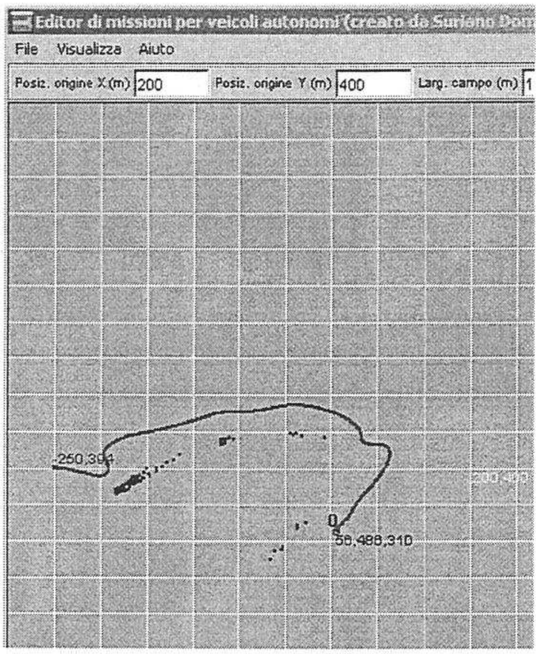
$$J_1(\eta)=\begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi+\cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi+\cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi+\sin\phi\sin\theta\sin\psi & -\cos\psi\sin\phi+\sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \quad (10)$$

$$J_2(\eta)=\begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \quad (11).$$

Echo sounders are simulated using the method suggested in [20]. The sensor beam is simulated as a cone with b degrees of aperture. While the robot moves in the virtual environment, the points belonging to this cone are explored in order to see if they impact with objects in the vehicle surroundings. In this case, the axial distance plus a gaussian error are considered as the measurement given by the transducer. Moreover, in order to consider the multi-path problem, a second error source is added periodically. This second source has a high variance and a mean greater than zero producing glitches in the echo sounder ranges.

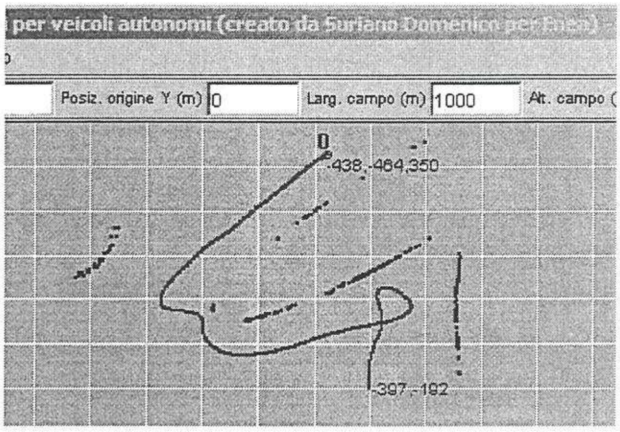
6. SIMULATIONS AND RESULTS

In order to test the collision avoidance system several simulations are carried out, they are set in various scenarious: closed submarine canyons, dense obstacles submarines areas, as well as contrived situations like labyrinths. In figure 8 is shown the vehicle path after carrying out a simply mission: the AUV has to go to the goal point located at coordinates O(56,486,310), starting from the position S(-250,394,0). The continuous red line represents the AUV path, while the blue scattered points are the sonar readings that shape a little isle contour. An other relevant experiment, shown in figure 9, concerns the execution of a mission in which the AUV, starting from the point S(-397,-132, 0), has to go out a closed canyon and reach the goal position O(-438,-464,350). The third simulation that we propose regards a mission in which the vehicle starts from the point S(-1154,-64,0) and reaches the goal point O(–400,–486,350) after crossing a dense obstacle sea area (see figure 10).



Posiz. origine X (m) 200
Posiz. origine Y (m) 400
Larg. campo (m) 1

Fig. 8 - AUV Sara path mission top view



Posiz. origine Y (m) 0
Larg. campo (m) 1000
Alt. campo (

Fig. 9 - AUV Sara go out from a closed canyon, the scattered blue points are the echo sounders readings, while the continuous red line is the vehicle path (top view)

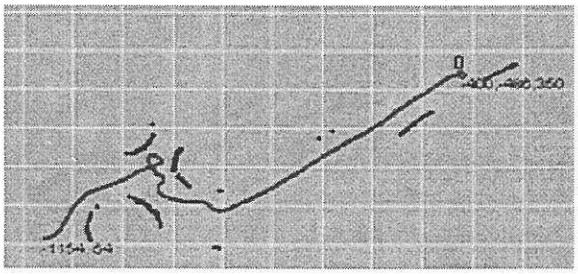


Fig. 10 - Dense obstacle sea area crossing top view

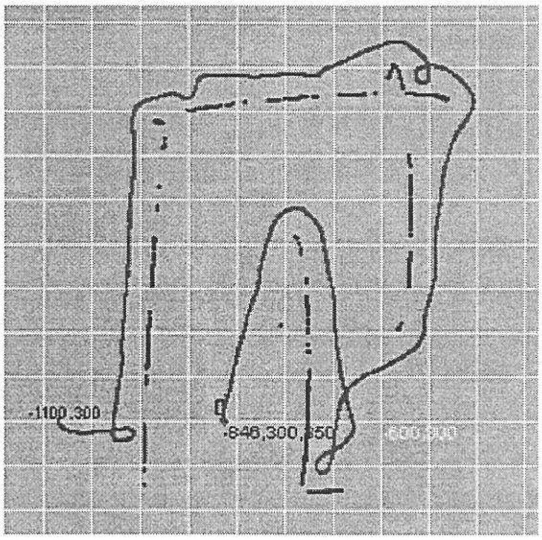


Fig. 11 - AUV Sara go into a labyrinth reaching its goal

The last test is the most difficult. The AUV starts from the position S(-1100,300,350) and has to go into a closed maze like structure in order to reach the goal point located at O(–846,300,350). Figure 11 shows the collision avoidance system performance.

7. CONCLUSIONS

This paper has described a collision avoidance system based on echo sounders range readings for AUV and has also approached the real world map building problem. The solution proposed is tested through several simulations performed simulating eterogeneous submarine scenarios. All the executed tests provided satisfactory results, showing that the vehicle is able to reach its mission goal avoiding the obstacles and finding a suitable path in all the situations. The main limitation of the proposed system is its sensitivity to the accuracy of the vehicle position valuations. Future works will regard in water tests programmation.

REFERENCES

[1] B. Papalia, W. Prendin, G. Veruggio,"SARA, an autonomous underwater vehicle for researches in Antarctica", in Proceedings of OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Vol.3 pp.617-620.

[2] Cooke R. A., "Microcomputer Control of Free Ranging Robots." Proc. of the 13th Int. Symp. on Industrial Robots and Robots, Chicago, Ill., April, 1983, pp. 13.109-13.120.

[3] Weisbin C. R., de Saussure G., and Kammer D., "SELF-CONTROLLED. A Real-Time Expert System for an Autonomous Mobile Robot." Computers in Mechanical Engineering, September, 1986, pp. 12-19.

[4] V. Lumelsky,T. Skewis, "Incorporating range sensing in the robot navigation function" IEEE Transactions on Systems Man and Cybernetics, vol. 20, pp. 1058 – 1068, 1990.

[5] V. Lumelsky, Stepanov, "Path-planning srategies for a point mobile automaton amidst unknown obstacles of arbitrary shape" in Autonomous Robots Vehicles, I.J. Cox, G.T. Wilfong (Eds), New York, Springer, pp. 1058 – 1068, 1990.

[6] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots" International Journal of Robotics Research, vol. 5, no. 1, pp. 90–98, 1995.

[7] J. Borenstein, Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots" IEEE Transaction on Robotics and Automation, vol. 7, no. 3, pp. 278 – 288, 1991.

[8] S. Quinlan, O. Khatib, "Elastic bands: Connecting path planning and control" in Proceedings of the IEEE Conference on Robotics and Automation, pp. 802 – 807, 1993.

[9] Ishii K. , Nishida S., Watanabe K., Ura T., "A collision avoidance system based on self-organizing map and its application to an underwater vehicle" in Proceedings of 7th International Conference on Control, Automation, Robotics and Vision, 2002. ICARCV 2002. Publication Date: 2-5 Dec. 2002 Volume: 2, On page(s): 602- 607 vol.2

[10] Demuth G., Springsteen S. "Obstacle avoidance using neural networks" in Proceedings of Symposium on Underwater Vehicles Technologies, 5-6 June 1990 Washington DC, pp. 213-222.

[11] T. Kohonen, "Self-organized formation of topologically correct feature maps", *Bioi. cyhem.,* vol.43, pp. 59-69., 1982.

[12] H. Ritter and T. Kohonen, "Self-Organizig Semantic Maps", *Bioi. Cybern.,* Vol. 61, pp. 241-254, 1989.

[13] V. Creuze, B. Joveuncel, "Avoidance of underwater cliffs for autonomous underwater vehicles" in Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and System, 2002. Volume 1, pp. 793- 798.

[14] Zapata R., Lepinay P., "Collision avoidance and bottom following of a torpedo-like AUV" in Proceedings of MTS/IEEE Conference OCEANS '96. 'Prospects for the 21st Century'. Pp. 571-575 vol.2

[15] Fodrea L. R.,"Obstacle Avoidance Control for the REMUS Autonomous Underwater Vehicle", MSME Thesis, Naval Postgraduate School, Dec. 2001

[16] Key-Pyo Rhee, Nam-Sun Son, Dong-Jin Yeo, Jong-Ik Lee, Jae-Moon Yoon, "Design of an Obstacle Avoidance System of Underwater Vehicle in Diving and Steering Plane" Proceeding of the 2nd International Workshop on Ship Hydrodynamics, Wuhan, China, Sep. 2001.

[17] P. Ridao, J. Batlle, M. Carreras "O2CA2, a new object oriented control architecture for autonomy: the reactive layer", Control Engineering Practice, Vol.10,n.8 August 2002, pp. 857-873.

[18] Borenstein, J. and Koren, Y., "Histogramic In-motion Mapping for Mobile Robot Obstacle Avoidance." *IEEE Journal of Robotics and Automation*, Vol. 7, No. 4, 1991, pp. 535-539.

[19] Fossen, T.I, "Guidance and Control of Ocean Vehicles", John Willey & Sons, 1995.

[20] P. Ridao, J. Battle, J. Amat, M. Carreras, " A distribuited environment for virtual and/or real experiments for underwater robots", in Proceedings of 2001 IEEE international conference on Robotics & Automation – Seoul, Korea – May 21-26 2001.

Edito dall' ENEA
Funzione Centrale Relazioni Esterne
Unità Comunicazione

Lungotevere Thaon di Revel, 76 - 00196 Roma

www.enea.it

Stampa: Laboratorio Tecnografico ENEA - CR Frascati

Finito di stampare nel mese di gennaio 2007