# An Autonomous Navigation System for Unmanned Underwater Vehicle

Seong-Gon Kim[1] and Yong-Gi Kim[2]
[1]University of Florida
[2]Gyeongsang National University
1USA
[2]Korea

## 1. Introduction

Autonomous underwater vehicles should possess intelligent control software that performs intellectual functions such as cognition, decision and action, which originally belong to the ability of domain expert, since the unmanned underwater robot is required to navigate in hazardous environments where humans do not have direct access to. In this paper, we suggest an intelligent system architecture called the RVC model, which can be applied to various kinds of unmanned vehicles. The architecture consists of the collision avoidance system, the navigation system, and the collision-risk computation system. The RVC architecture is devised to make use of artificial intelligence techniques, and to provide the subsystems structural and functional independency.

The collision avoidance system adopts a new heuristic search technique for the autonomous underwater vehicles equipped with obstacle avoidance sonar. The fuzzy relation product between the sonar sections and the properties of real-time environment is used to decide the direction for the vehicle to proceed. The simulation result leads to the conclusion that the heuristic search technique enables the AUV to navigate safely through obstacles and reach its destination goal with the optimal path. The navigation system executes the offline global path planning for the AUV to guarantee the safe and efficient navigation from its start point to the target destination. The system also does the duty of monitoring and controlling the vehicle to navigate following the directed path to destination goal. The collision-risk computation system produces a degree of collision risk for the underwater vehicle against surrounding obstacles using information from the circumstances, obstacles, and positions. The degree is provided to the collision avoidance system as one of the decision tools used for safe avoidance with the obstacles. A 3D simulator is developed to test the AUV navigation system based on the RVC model. The goal of the simulator is to serve as a testing ground for the new technologies and to facilitate the eventual transfer of these technologies to real world applications. The simulation system consists of an environment manager, objects and a 3D viewer. Objects model all physical elements such as the map, obstacles and the AUV. The environment manager plays the role of an intermediary, which allows created objects to interact with each other, and transmits information of the objects to the 3D viewer. The 3D viewer analyzes the received information and visualizes it with 3D graphics by using OpenGL primitives.

Source: Underwater Vehicles, Book edited by: Alexander V. Inzartsev,
ISBN 978-953-7619-49-7, pp. 582, December 2008, I-Tech, Vienna, Austria

## 2. Intelligent system architecture

The navigation system for autonomous underwater vehicles needs various techniques to be effectively implemented. The autonomous technique usually contains complicated and uncertain factors and thus makes use of some artificial intelligence methods to solve the problems. Artificial intelligence techniques are classified largely into two categories. One is the symbolic AI technique, such as knowledge-based system, which operates in ways similar to the human thought process, and the other is the behaviour-based AI technique such as neural network or fuzzy which behaves much like human sensorial responses. The former is considered a higher-level intelligence but it alone is not enough to make a system conduct intelligently in domains where very sophisticated behaviours are needed.

### 2.1 RVC intelligent system model

Research in autonomous navigation systems became very active with the rapid advancement of hardware technologies during the end of the 20th century. Researchers had tried to implement intelligent control for autonomous navigations using symbolic AI techniques but they could not succeed because of the difference in representation methods between the symbolic AI techniques they were attempting to use and the actual information needed to operate the navigation system. The symbolic AI technique is adequate for problems which are well-defined and easy to represent but not for real world problems which are usually ill-defined and in most cases have no limitation. These difficulties made researchers work on the development of AI techniques that were good for solving real world problems. Reactive planning (Agre et al., 1987), computational neuroethology (Cliff, 1991), and task-oriented subsumtion architecture (Brooks, 1986) are the results of the research, and are called behaviour-based AI (Turner et al., 1993). Many researches concluded that symbolic AI or behaviour-based AI techniques alone cannot reach the allowable goal for the navigation system of unmanned underwater vehicles (Arkin, 1989) and recent researches on autonomous navigations are focused on using both AI techniques and improving the performance of the system (Arkin, 1989; Turner, 1993; Scerri & Reed, 1999; Lee et al. 2004; Bui & Kim, 2006). The two AI techniques have different characteristics and thus is hard to combine the two techniques into a single system effectively. In this article, an intelligent system model, called the RVC (Reactive Layer-Virtual World-Considerate Layer), is introduced for the effective combination of symbolic and behaviour-based AI techniques into a system.

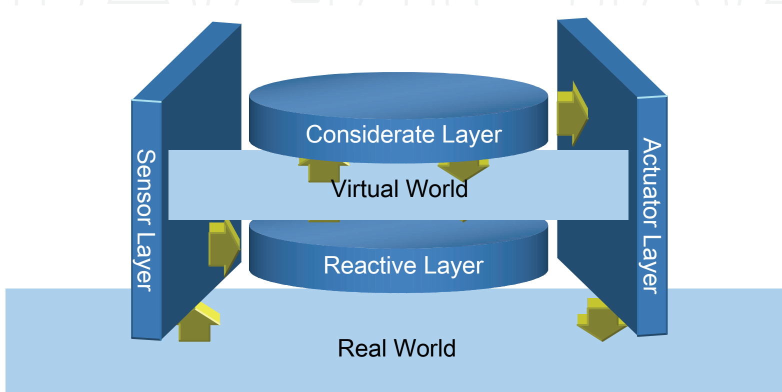



Fig. 1. RVC intelligent system

Fig.1 is the schematised RVC intelligent system model. The model is conceptualised for cordial combination of the two different AI techniques, and it also enhances the structural and functional independency of each subsystem, such as collision avoidance system, navigation system, or collision risk computation system. In this model, the reactive layer processes the uncertain problems in the real world and then passes the symbolized results to the considerate layer where the symbolic AI technique makes use of the information for the final decision. For this procedure, the model needs a common information storage space, where the information produced from the reactive layer is represented in real-time before it is consumed by the considerate layer. From the considerate layer's point of view, the information storage space resembles a subset of real world, and thus this storage space will be referred to as a 'Virtual world' henceforth.

### 2.2 Autonomous navigation architecture based on RVC intelligent system model

Autonomous navigation system based on the RVC intelligent system model uses the concept of information production/consumption and client/server for transferring the collected information from the real world to each module of the system in real-time. For this purpose, the intelligent navigation system contains functions such as memory management, data communication, and scheduling. Data communication in the system adopts the TCP/IP protocol, and this makes the system platform-independent and thus makes load balancing smooth. The scheduling function synchronizes the exchanging of real-time data among the modules, and it also processes possible errors in the system. The RVC intelligent system model guarantees independency among the modules in the system, and this enables the parallel development of each system module. Fig. 2 is the autonomous navigation architecture based on the RVC system model.

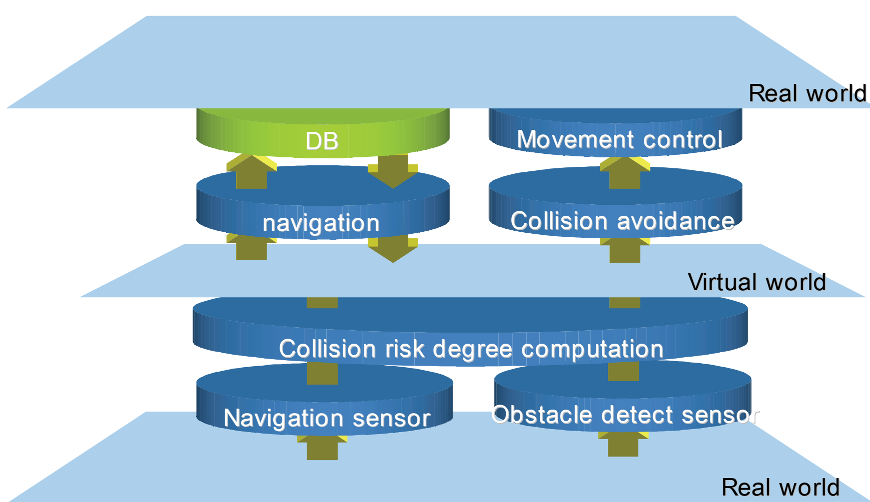



Fig. 2. Autonomous navigation architecture based on RVC system model

## 3. Subsystems for autonomous navigation system

### 3.1 Collision avoidance system

Relational representation of knowledge makes it possible to perform all the computations and decision making in a uniform relational way, by mean of special relational compositions called triangle and square products. These were first introduced by Bandler and Kohout and are referred to as the BK-products in the literature. Their theory and applications have made substantial progress since then (Bandler & Kohout, 1980a, 1980b; Kohout & Kim, 1998, 2002; Kohout et al., 1984).

There are different ways to define the composition of two fuzzy relations. The most popular extension of the classical circular composition to the fuzzy case is so called max-min composition (Kohout et al., 1984). Bandler and Kohout extended the classical circular products to BK-products as sub-triangle (◁, "included in"), super-triangle (▷, "includes"), and square (□, "are exactly the same"). Assume the relations $R$ and $S$ are fuzzy relations, then the *R-afterset* of $x$, $xR$ and the *S-foreset* of $z$, $Sz$, obviously are fuzzy sets in $Y$. The common definition of inclusion of the fuzzy set $xR$ in $Y$ in the fuzzy set $Sz$ in $Y$ is given by (1).

$$xR \subseteq Sz \Leftrightarrow (\forall y \in Y)(xR(y) \leq Sz(y)) \tag{1}$$

A fuzzy implication is modeled by means of a fuzzy implication operator. A wide variety of fuzzy implication operators have been proposed, and their properties have been analyzed in detail (Bandler & Kohout, 1980c; Lee et al., 2002). For this study, we make use only of operator 5 as shown in (2).

$$a \rightarrow_5 b = \min(1, 1 - \mathrm{a} + \mathrm{b}) \tag{2}$$

Using (2), with $n$ the cardinality of $Y$, we easily obtain the definitions for the sub-triangle and supper-triangle products in (3), (4) while the square product using the intersection and the minimum operator is shown in (5) and (6) respectively.

$$x_i(R \lhd S)z_j = \frac{1}{n}\sum_{y \in Y} \min(1, 1 - x_i R(y) + Sz_j(y)) \tag{3}$$

$$x_i(R \rhd S)z_j = \frac{1}{n}\sum_{y \in Y} \min(1, 1 + \mathrm{x_i} R(y) - Sz_j(y)) \tag{4}$$

$$x_i(R \,\square\, S)z_j = x_i(R \lhd S)z_j \cap x_i(R \rhd S)z_j \tag{5}$$

$$x_i(R \,\square\, S)z_j = \min(x_i(R \lhd S)z_j, x_i(R \rhd S)z_j) \tag{6}$$

Along with the above definitions, α-cut and Hasse diagram are also the two important features of this method. The *α-cut* transforms a fuzzy relation into a crisp relation, which is represented as a matrix (Kohout & Kim, 2002; Kohout et al., 1984). Let $R$ denotes a fuzzy relation on the $X \times Y$, the *α-cut* relation of $R$ is defined as the equation (7).

$$R_\alpha = \{(x, y) \mid R(x, y) \geq \alpha \text{ and } 0 \leq \alpha \leq 1 \tag{7}$$

The Hasse diagram is a useful tool, which completely describes the partial order among the elements of the crisp relational matrix by a Hasse diagram structure. To determine the Hasse diagram of a relation, the following three steps should be adopted (Lee & Kim, 2001).

Step 1. Delete all edges that have reflexive property.

Step 2. Eliminate all edges that are implied by the transitive property.

Step 3. Draw the diagraph of a partial order with all edges pointing upward, and then omit arrows from the edges.

In this study it is required that obstacle avoidance sonar range can be partitioned into several sub-ranges. One of these represents for the successive heading candidate for AUVs to go ahead. Whenever obstacle is detected, the sonar return is clustered and the sections in which obstacles present can be identified. The sonar model is illustrated as in Fig.3. Domain experts who have wide knowledge about ocean science could give the properties about the environmental effects to the of AUVs navigation.

A forward looking obstacle avoidance sonar whose coverage range can be divided into multi-sections is used to determine a heading candidate set *S*. Otherwise, a property set *P* describes the effects of AUVs toward the real time environment. The fuzzy rule base and membership function for the corresponding property can be estimated subjectively by the expert knowledge. With the set of the candidate $S = \{s_1, s_2, s_3, ..., s_i\}$ and the set of environmental properties $P = \{p_1, p_2, ..., p_j\}$, the relation *R* is built as (8). The elements $r_{ij}$ of this relation mean the possibility the section $s_i$ can be characterized by the property $p_j$. The value of $r_{ij}$ is calculated by means of the rule bases with the membership functions.

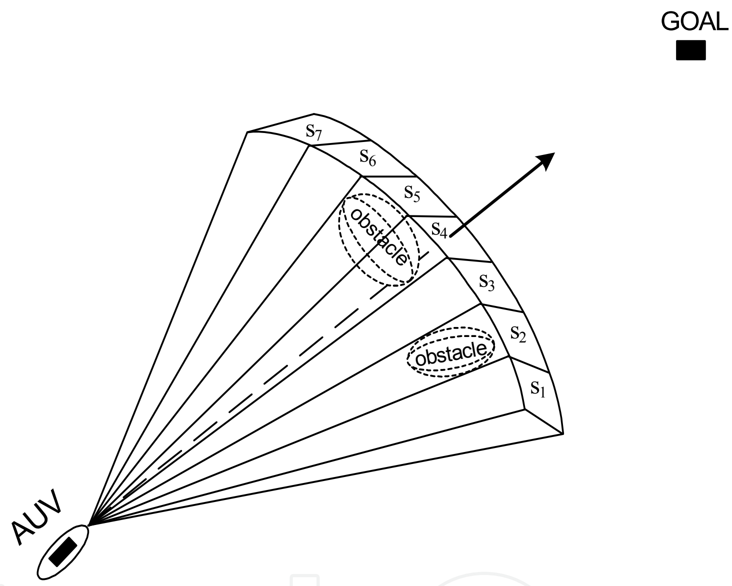



Fig. 3. A model of forward looking obstacle avoidance sonar

$$R = S \times P = \begin{array}{c c} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1j} \\ r_{21} & r_{22} & \cdots & r_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ r_{i1} & r_{i2} & \cdots & r_{ij} \end{bmatrix} & \begin{array}{c} s_1 \\ s_2 \\ \vdots \\ s_i \end{array} \\ \begin{array}{cccc} p_1 & p_2 & \cdots & p_j \end{array} & \end{array} \tag{8}$$

$$T = R \triangleleft R^T = \begin{array}{c c} \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1i} \\ t_{21} & t_{22} & \cdots & t_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ t_{i1} & t_{i2} & \cdots & t_{ii} \end{bmatrix} & \begin{array}{c} s_1 \\ s_2 \\ \vdots \\ s_i \end{array} \\ \begin{array}{cccc} s_1 & s_2 & \cdots & s_i \end{array} & \end{array} \tag{9}$$

$$R_\alpha = \alpha\_cut(T,\alpha) = \begin{array}{cc} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} \\ a_{21} & a_{22} & \cdots & a_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ii} \end{bmatrix} & \begin{matrix} s_1 \\ s_2 \\ \vdots \\ s_i \end{matrix} \\ \begin{matrix} s_1 & \;\; s_2 & \;\; \cdots & \;\; s_i \end{matrix} & \end{array} \tag{10}$$

In the next step a new fuzzy relation $T$ is computed by using sub-triangle product $\triangleleft$ to fuzzy relation $R$ and $R^T$, the transposed relation of $R$. The fuzzy relation $T$ as shown in (9) is the product relation between candidate set $S$ that means the degree of implication among elements of candidate set. Then, the *α-cut* is applied to fuzzy relation $T$ in order to transform into crisp relation as shown in (10). It is important to select a reasonable *α-cut* value because the hierarchical structure of candidate set depends on an applied *α-cut*. Finally, we draw the Hasse diagram, which completely describes a partial order among elements of candidate set, that is to say, a hierarchical structure among the elements of candidate set with respect to the optimality and efficiency. Select then the top node of the Hasse diagram as the successive heading direction of AUVs.

Because the energy consumption in vertical movement of AUVs is much greater than in the horizontal movement (1.2 times) (Ong, 1990), this technique focus strongly on the horizontal movement. In the case of obstacle occurrence, AUVs just turn left or turn right with the turning angle determined by degree from the current heading to the selected section. But in the exception case a very wide obstacle has completely filled up the sonar's coverage, AUVs must go to up one layer at a time and then apply the algorithm to find out the turning. Until obstacle clearance, AUVs are constrained to go back to the standard depth of the planned route.

The algorithm of the proposed technique can summarize into five below steps and is imitated briefly in control flow as shown in Fig. 4.

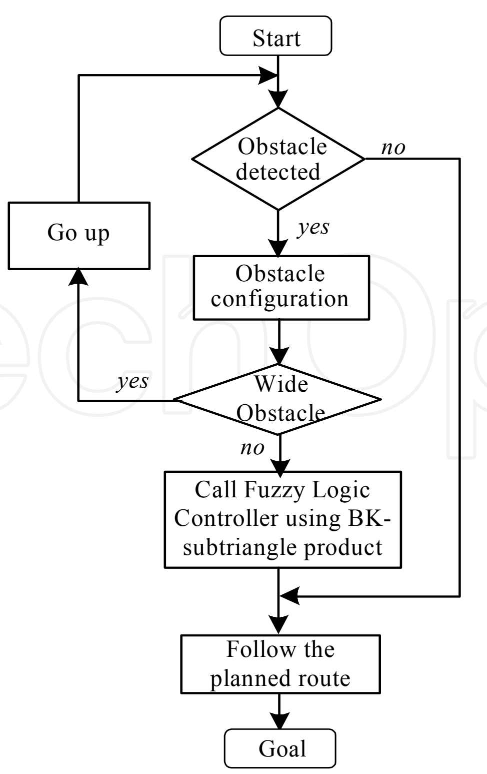



Fig. 4. A control flow of collision avoidance of AUV

Step 1. If AUVs detects obstacle then go to next step, else go to step 5
Step 2. Determine P and configure S
Step 3. If very wide obstacle is detected in all of S then go up and return step 1; else go to next step.
Step 4. Call the fuzzy logic controller using BK-subtriangle product to S and P to figure out the successive heading for obstacle avoidance
Step 5. Go on in the planned route

### 3.2 Navigation

Generally, the navigation system of unmanned underwater vehicles consists of two fucntions. One is path planning and the other is guidance and control (Vasudevan & Ganesan, 1996; Oommen et al., 1987). Path planing is the fuction of setting a path from a start point to a target destination using waypoints, and the function of guidance and controlling is to monitor and guide the vehicle to follow the designated path. The duty of the navigation system of the unmanned underwater vehicle in this article is transferring the following information into the autonomous navigation system's Virtual world: first, the results of an offline global path planning which allows the system a safe and optimal operation from start point to target destination, and secondly, monitoring and controlling the vehicle to stay on the set path to target destination. Fig. 5 shows the stucture of the navigation system

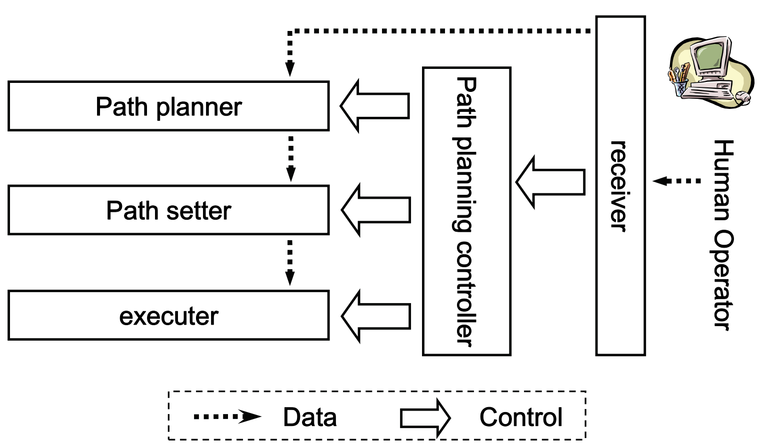



Fig. 5. Navigation system

Unmanned underwater vehilcles operate in a 3-dimensional environment and the vehicles do not have to consider static obstacles that are located below a certain depth. Global path planning for the autonomous navigation system adopts a new palnning algorithm (Kim, 2005) in which points of contact with the obstacles and waypoint trees are utilized to get the optimal path to the target destination. To get the global path, this algotithm computes the position of contact points between the start point and the static obstacle, and then connects the contact points to produce a waypoint tree. The waypoint tree is searched using a depth-first search algorithm to get the optimal path to the destination. The waypoints produced are delievered to the Virtual world, and will be used by other subsystems such as the collision avoidance system.
Fig. 6 shows an example of paths produced using the contact points when there is a static obstacle between the start point and the destination. First, it calculates the position of the left

contact point Ls and the right contact point Rs between the start point S and the obstacle, then it calculates the position of the left contact point Lg and the right contact point Rg between the destination G and the obstacle. Then, the contact points between Ls and Lg and the contact points between Rs and Rg are calculated recursively. The produced paths and contact points are stored using the data structure shown in Fig. 7 where the coordinates of the points are the data of the node, and the pointers are directed to next nodes.

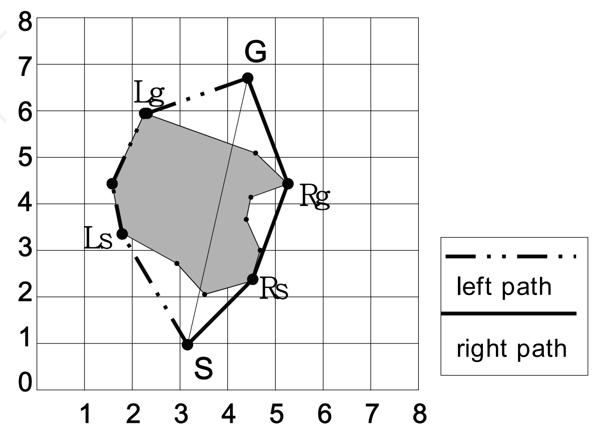



Fig. 6. Path planning

| | | | | | |
|---|---|---|---|---|---|
| $P_A$ | Left path | $P_B$ | $P_C$ | Right path | $P_D$ |

Fig. 7. The structure of node

When more than one obstacle exists between the start point and the target destination, the algorithm produces a waypoint tree for each contact point of the obstacle. Fig. 8 is a marine chart of such case, and the waypoint tree is shown in Fig. 9. With the waypoint tree, one can extract the obstacles that actually affect navigation of the vehicle from all the static obstacles existing between start point S to destination point G. Information of the left and right paths for avoiding the obstacles will be stored in the waypoint tree. The waypoint tree will have the minimum required information for producing all the paths from start point S to destination point G.

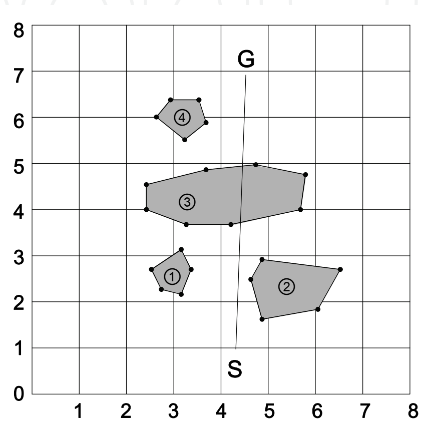



Fig. 8. A marine chart with multiple obstacles

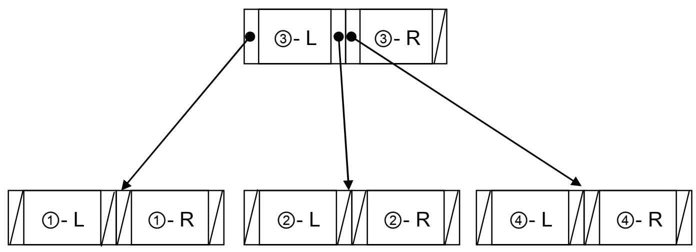



Fig. 9. Way-point tree

### 3.3 Collision risk computation system

The Collision Risk Computation System uses information from the surrounding environment as well as the obstacle and positioning information to compute the risk of the autonomous underwater vehicle colliding with various obstacles that exist in its environment (Kim, 2001; Hara & Hammer, 1993). The system provides a basis for the decisions it makes so that if the system finds the autonomous underwater vehicle at risk of colliding with an obstacle, it changes the navigation path so that it can safely avoid the obstacle.

The Collision Risk Computation System uses fuzzy inference which consists largely of 3 modules as seen in Fig. 10 to compute collision risks the autonomous underwater vehicle might face while navigating in its environment. The first module is the input module that reads in the vector information of the autonomous underwater vehicle and obstacle from the Virtual world, then computes the obstacle's DCPA(Distance of the Closest Point of Approach) and TCPA(Time of the Closest Point of Approach). The second Collision Risk Computation Module then uses fuzzy logic to calculate the risk of collision. It fuzzifies the DCPA and TCPA from the first module and performs a fuzzy-inference, then defuzzifies it to compute the risk of collision. In order to send the computed collision risk value to the Collision Risk Computation System, the third Output Module takes the computed collision risk and transfers it to the Virtual world.

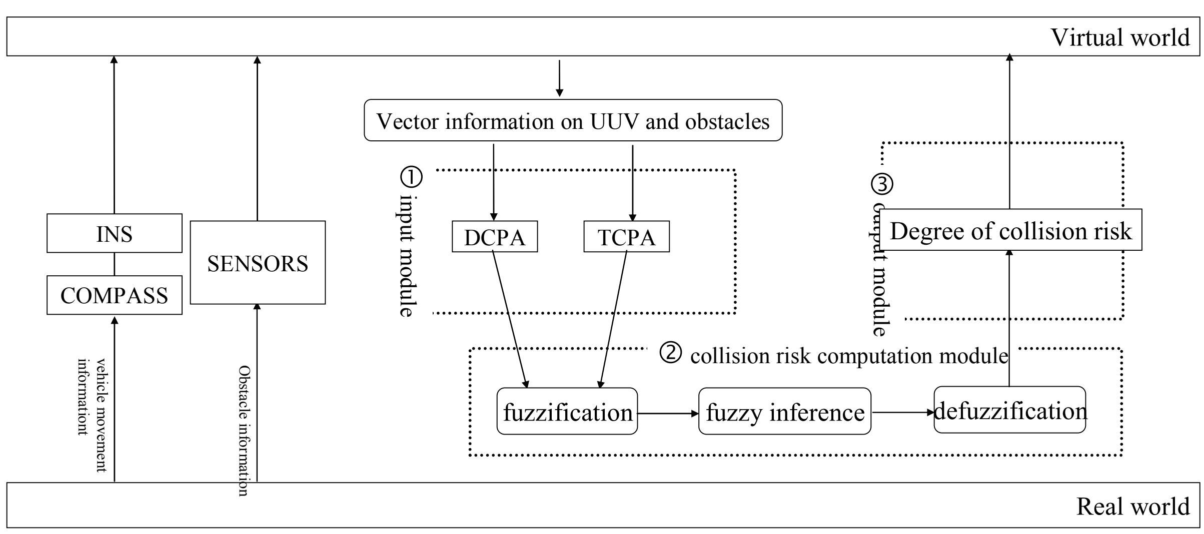



Fig. 10. Collision-risk computation system

The collision risk is computed by fuzzy-inference using DCPA and TCPA as its input. The inference rule uses the centroid method with the min operation as the antecedent and the product operation as the consequent. The membership functions of DCPA and TCPA, which are the input values, and the collision risk, which is the output value, are first defined. Fig. 11, Fig. 12, and Fig. 13 show the membership functions of the DCPA, TCPA and collision risk, respectively. The labels used for each membership function is as follows:

P : Positive, N : Negative, S : Small, M : Medium, B : Big

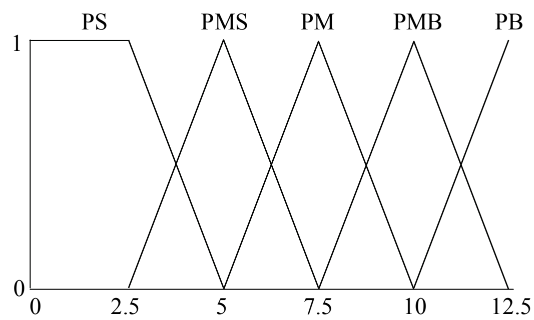



Fig. 11. Membership function of DCPA(meter)

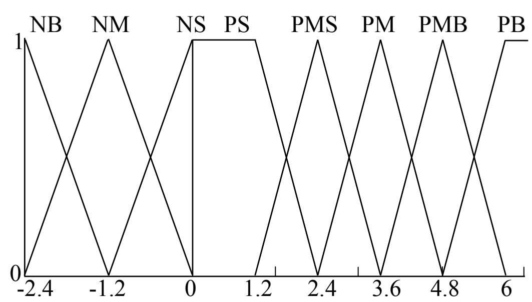



Fig. 12. Membership function of TCPA(second)

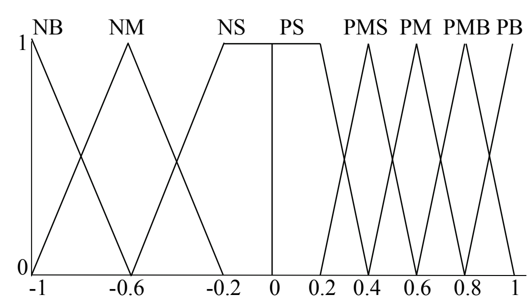



Fig. 13. Membership function of degree of collision risk

| TCPA / DCPA | NB | NM | NS | PS | PMS | PM | PMB | PB |
|---|---|---|---|---|---|---|---|---|
| PS | NS | NM | NB | PB | PMB | PM | PMS | PS |
| PMS | NS | NS | NM | PMB | PM | PMS | PS | PS |
| PM | NS | NS | NS | PM | PMS | PS | PS | PS |
| PMB | NS | NS | NS | PMS | PS | PS | PS | PS |
| PB | NS | NS | NS | PS | PS | PS | PS | PS |

Table 1. Inference rules for degree of collision risk

Table 1 is the inference rule table used to compute the collision risk. It was preset based on a navigation expert's knowledge and this can be changed depending on any particular navigator's experience or knowledge.

## 4. Simulation system

The simulation system is a software that lets users experiment through a computer the autonomous underwater vehicle's overall navigation process as well as the objects that appears during the navigation by modeling all the states, factors, objects, devices etc. associated with the autonomous underwater vehicle and equipping it with a virtual autonomous navigation system. The simulation system appropriately models the various sensors, the speedometer and other navigation devices that work as input, and the propulsion and steering device that work as output so that they can interact with the autonomous navigation system.
The autonomous underwater vehicle's navigation system receives information about its environment necessary for navigation as input from various sensors, then analyzes this information to send new action commands to the propulsion or steering device. The simulation system replaces navigation sensors, obstacle sensors, and movement controller in the RVC model that was presented in chapter 2, and displays the execution result on the screen. As seen in Fig. 14, the simulation system simulates the interaction between the autonomous navigation system and the physical devices, then displays the result in 3D. This system largely consists of an environment manager, objects and a 3D viewer.

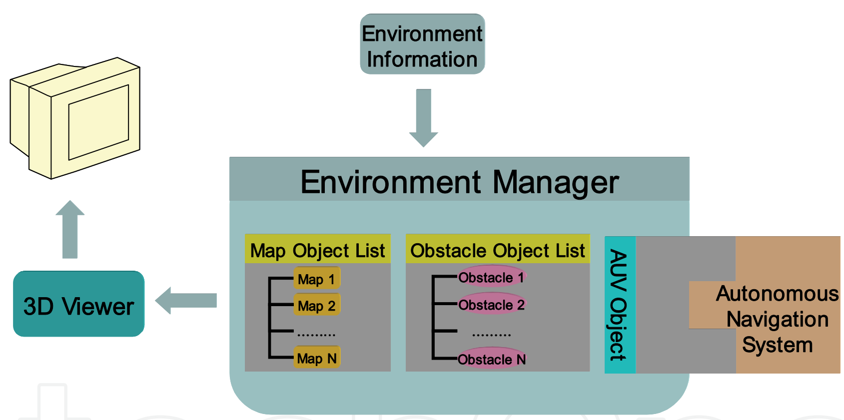



Fig. 14. Structure of simulation system

The environment manager takes environment information such as Table 2 and models the factors that influence the autonomous underwater vehicle's navigation while enabling interaction between the objects that are involved. The autonomous navigation system references the various information generated by the environment manager and carries out its tasks. The main function of the environment manager is to construct and manage the simulation environment by generating various objects and initializing its properties while tuning the interaction between these objects.
An object is the object oriented representation of the state of the environment information and the autonomous underwater vehicle. To represent the autonomous underwater vehicle's navigation environment, the simulator creates subjects of identical properties into classes, and these classes are the main frame of objects. The objects generated by the environment manager represents the navigation environment and has specific properties

and motions depending on its given function. The environment manager generates a map, obstacle and the autonomous underwater vehicle as an object. The map object represents the ocean floor geography that the autonomous underwater vehicle navigates through and has a hierarchical structure constituting of Areas and Spots. The obstacle object represents all obstacles that can appear during navigation and can be divided in to dynamic obstacles and static obstacles depending on their movement. The autonomous underwater vehicle object represents the autonomous underwater vehicle itself and is connected with the autonomous navigation system. Fig. 15 shows the overall environment information classes.

| Subject | Start symbol | Input content | Example |
|---|---|---|---|
| Area | E | Entire area range | E200 200 |
| UUV | A | UUVs position value | A-67.3 3.3 81.4 |
| Obstacle | O | Obstacles position value | O-7.8 0.0 -29.4 |
| Geography | M | Grid's relative position value<br>Grid's four corner altitude value<br>Spot altitude value<br>Grid's unitvector value | M00001317.2 8.6<br>12.9 17.2 17.2<br>-0.7 0.1 0.7 |

Table 2. Environment Information

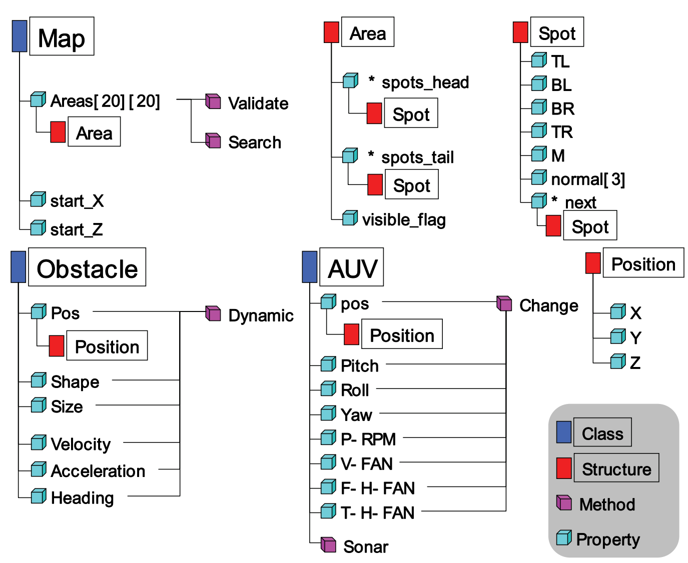



Fig. 15. Environment information class

The 3D Viewer is the component that displays the simulation process in 3D as seen in Fig. 16. The 3D viewer receives information from the environment manager as input and uses OpenGL to render the simulation process in 3D. OpenGL is the software interface to the graphical hardware allows the generation of objects or computations necessary in producing 3D applications. It can run on various hardware platforms but does not support commands with the ability to generate complex objects and can only generate primitives such as points, lines and polygons. For the representation of complex structures such as geography, obstacles and the autonomous underwater vehicle object, the primitives are combined to build the objects necessary for the simulation.

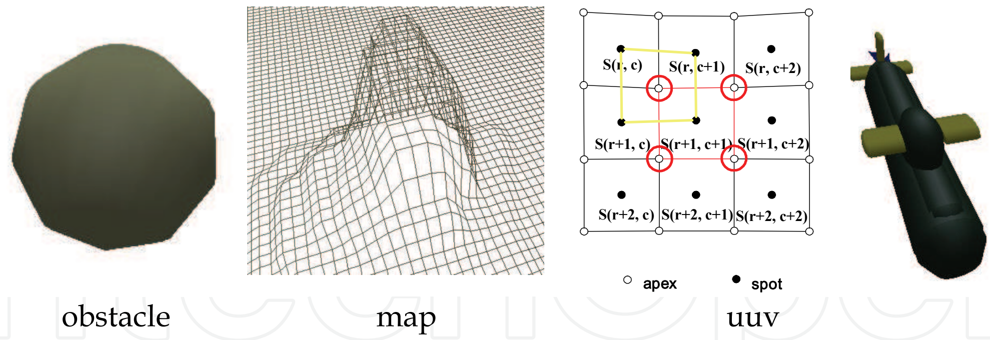



Fig. 16. 3-D viewer for object

The Collision Avoidance System, Collision Risk Computation System and the Simulation System were used together in the simulation to test the performance of the presented underwater vehicle's autonomous navigation. For the specifications necessary for the simulation, the autonomous underwater vehicle developed by the Korean Agency for Defense Development was used and is shown in Table 3.

| Spec. | Value |
|---|---|
| Vehicle length/diameter | 10 (ratio) |
| Max speed | 8.0kts |
| Max operation depth | 100m |
| Displacement tonnage | 1.380kg |

Table 3. Specification of UUV

The underwater vehicle's autonomous navigation system was tested using a scenario where three dynamic obstacles exist. The autonomous underwater vehicle's starting point and destination point were set to S(0,0,-10) and G(0,210,-10), respectively. The first obstacles starting point and destination point were set to O(-20),150,-10) and O(13,-60-10), respectively, where it approaches the autonomous underwater vehicle from the front left side. The second obstacle approaches the autonomous underwater vehicle from the rear with the starting point and destination point set to O(30,-10,-17) and O(-2,160,-17), and the last obstacle approaches the UUV directly from the front with the starting point and destination point set to O(0,200,-10) and O(0,-20,-10), respectively. Fig. 17 shows the actual

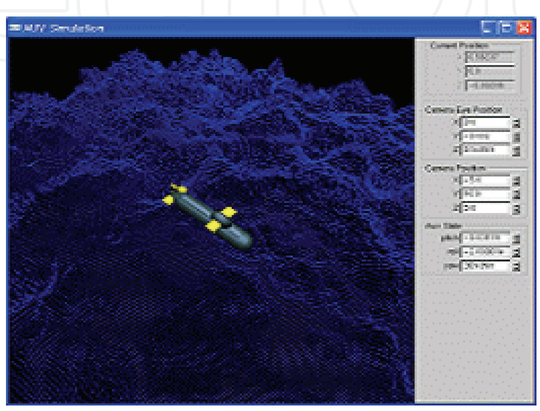
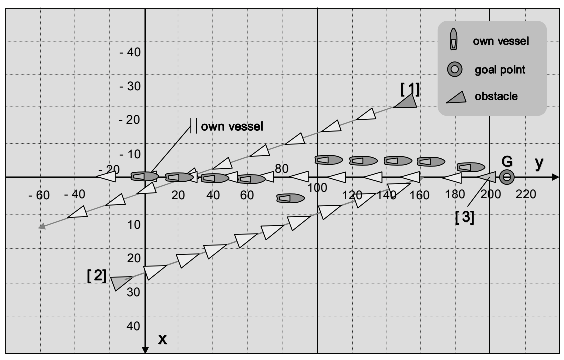
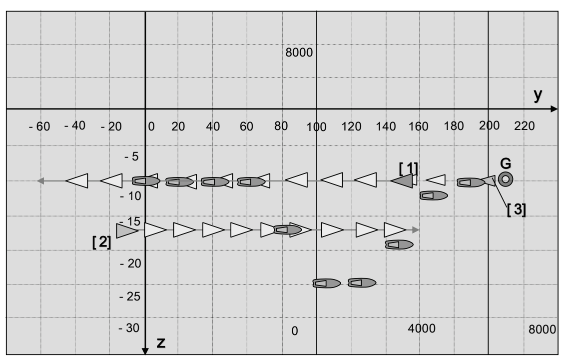


Fig. 17. Display of simulation

simulation in progress and Fig. 18 shows the results as a map to help understand the simulation results. As shown in the simulation results, the autonomous underwater vehicle detected the first approaching obstacle O(-11,87,-10) at point P(0,63,-10) and sends an avoidance command to point P(7,84,-17), then continues to avoid the second obstacle O(18, 58, -17) to point P(-7, 105, -24), and this confirmed that the collision avoidance performed reasonably and efficiently.




(a). Simulation result in view of [X-Y] axis




(b). Simulation result in view of [Y-Z] axis

Fig. 18. Simulation result with scenarios

## 5. Conclusion

This paper designed a RVC intelligent system model that can be applied to various unmanned vehicles and the underwater vehicle's intelligent autonomous navigation system was designed consisting of a collision avoidance system, a navigation system and a collision risk computation based on a Virtual world system. During the development of the Virtual world system, several points such as the fusion of different techniques, preservation of system consistency, real time system processing etc. were taken into consideration, and since it models a client/server structure, it also has the features of consistency, independence maximization, and load balancing. The RVC intelligent system can be applied not only to autonomous underwater vehicles, but to various autonomous robots such as unmanned aerial vehicles, mobile robots and autonomous submarines. To test the performance of the underwater vehicle's intelligent autonomous navigation system based on this RVC intelligent system model, a 3D simulator was developed, and through a scenario with dynamic obstacles existing in the navigational environment, the validity of the intelligent autonomous navigation system was verified.

## 6. References


Agre, E. & Chapman, D. (1987). An Implementation of a theory of activity, *Proceedings of the Sixth National Conference on Artificial Intelligence*

Arkin, R. (1989). Towards the Unification of Navigational Planning and Reactive Control, *Proceeding of the AAAI Spring Symposium on Robot Navigation*, Mar. 1989

Bandler, W. & Kohout, L (1980a). Fuzzy Relational Products as a Tool for Analysis and Synthesis of the Behavior of Complex natural and Artificial System, In: *Fuzzy Sets: Theory and Application to Analysis and Information Systems*, Wang, K, & Chang, P. (Ed.)., (341-367), Plenum Press, New York

Bandler, W. & Kohout, L. (1980b). Semantics of Implication Operators and Fuzzy Relational Products, *International Journal of Man-Machine Studies*

Bandler, W & Kohout L. (1980c). Fuzzy Power Sets and Fuzzy Implication Operator, *Fuzzy Set and System*, No. 4, (13-30)

Brooks, A. (1986), A robust layered control system for a mobile robot," IEEE Journal of Robotics and Automation

Bui, L. & Kim, Y. (2006). An Obstacle Avoidance Technique for Autonomous Underwater Vehicles Based on BK-Products of Fuzzy Relation, *Fuzzy Sets and Systems*, Vol. 157, (560-577)

Cliff, D. (1991), A provisional manifesto, *Proceedings of the First International Conference on Simulation of Adaptive Behavior*, The MIT Press

Hara, K. & Hammer, A. (1993). A safe Way of Collision Avoidance Maneuver based on Maneuvering Standard using Fuzzy Reasoning Model, *Proceedings of MARSIM 93*, 1993, St. John's

Kim, E. (2001). Collision-risk computation system for collision avoidance. *Transaction of Korean Fuzzy and Intelligent System Association*, Vol. 2, (524-527)

Kim, Y. (2005). Autonomous ship control system using intelligence techniques, Technical Report, TECD-417, Korean Agency for Defense Development

Kohout, L.; Keravnou, E. & Bandler, W. (1984). Automatic Documentary Information Retrieval by Means of Fuzzy Relational Products, In: *Fuzzy Sets in Decision Analysis*, Gaines, B. R.; Zadeh, L. A. & Zimmermann, H. J., (Ed.), (308-404), North-Holland, Amsterdam

Kohout, L. & Kim E. (1998). Semiotic descriptors in fuzzy relational computations, Proc IEEE Int Symp Intelligent Control, IEEE Int Symp Computational Intelligence in Robotics and Autonomous and Intelligent Systems and Semiotic (A Joint Conf Science and Technology of Intelligent Systems), pp. 828-833, 1998, Piscataway

Kohout L. & Kim E. (2002). The role of BK-products of Relations in Soft Computing, *Soft Computing*, Springer-Verlag, No. 6, 92-115

Lee, Y. & Kim, Y. (2001). An Intelligent Navigation System for AUVs using Fuzzy Relational Products, *IFSA World Congress and 20th NAFIPS International Conference*, *Joint 9th*, Vol.2 , 709-714

Lee, Y.; Kim, Y. & Kohout, L. (2004). An Intelligent Collision Avoidance System for AUVs using Fuzzy Relational Products, *Information Sciences*, Vol. 158, 209-232

Lee, Y.; Noe, C. & Kim, Y. (2002).Implication Operators in Fuzzy Relational Products for a Local Path-Planning of AUVs, *Proceedings of NAFIPS, Annual Meeting of the North American,* Fuzzy Information Processing Society, 22

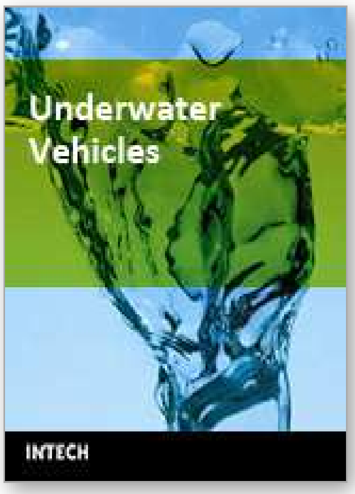
Ong, M. (1990). A Mission Planning knowledge-based system with Three-Dimensional Path Optimization for the NPS Model 2 Autonomous Underwater Vehicle, Master's Thesis, Naval Postgraduate School

Oommen, B. et al. (1987). Robot Navigation in Unknown Terrains Using Learned Visibility Graphs. Part I: The Disjoint Convex Obstacle case, *IEEE journal of Robotics and Automation*, Vol. RA-3, No. 6, (672-681)

Scerri, P. & Reed, N. (1999). Requirements for a General Agent Architecture for Agent-Based Simulation Environments, *Agents'99 Workshop on Autonomy Control Software*, May, 1999

Turner, R.; Blidberg, R.; Chappell, S. & Jalbert, J. (1993). Generic behaviors: An approach to modularity in intelligent systems control, *Proceedings of the 8th International Symposium on Unmanned Untethered Submersible Technology*, 1993

Vasudevan, C. & Ganesan, K. (1996). Case-Based path Planning for Autonomous Underwater Vehicles, In: *Underwater Robots*, Yuh, Ura and Bekey (Eds.), (1-15), Kluwer Academic Publishers, Boston




**Underwater Vehicles**
Edited by Alexander V. Inzartsev

ISBN 978-953-7619-49-7
Hard cover, 582 pages
**Publisher** InTech
**Published online** 01, January, 2009
**Published in print edition** January, 2009



**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821