“Autonomous Navigation of Unmanned Vehicles: A Fuzzy Logic Perspective”

N. C. Tsourveloudis, L. Doitsidis

Intelligent Systems and Robotics Laboratory
Department of Production Engineering and Management
Technical University of Crete
University Campus, Chania 73100
GREECE

K. P. Valavanis

Center for Robot Assisted Search and Rescue
Department of Computer Science and Engineering
University of South Florida
Tampa, FL 69320-5399
U.S.A

Acknowledgement: L. Doitsidis is supported by “HERAKLITOS” fellowships for research from the Technical University of Crete, EPEAEK II – 88727. Kimon Valavanis has been partially supported by ONR Grants N00014-03-01-786 and N00014-04-1-0487.

1. Introduction

Unmanned robotic vehicles are capable of performing desired tasks in unstructured, uncertain and potentially hostile environments. They may be remotely-operated or function *(semi-) autonomously* without human intervention. However, it will long before unmanned robot vehicles function as completely autonomous entities in diverse environments. Current unmanned vehicles adhere to different levels of *autonomicity* as defined by existing technology limitations and used sensors. Important operational characteristics related to unmanned vehicle functionality (*aerial*, *aquatic* or *terrestrial*), include the following:

Perception: *Acquire and use knowledge about the environment and itself.* This is done by taking measurements using various sensing devices and then extracting meaningful information that should be used in all later tasks (such as *localization*, *planning*, *collision free motion control, recharging,* etc).

Intelligence: *Operate for a considerable time period without human intervention*. This is associated with the learning and inference capabilities, which of the vehicle should have to be able to adapt (its behavior or/and shape) to the environment.

Action: Travel from point A to point B. The vehicle should utilize predefined and acquired knowledge to move in dynamic environments without involving humans in the navigation loop.

In robotics, autonomy is mainly associated with navigation issues. From a conceptual point of view, autonomous navigation of robotic vehicles may be achieved via continuous interaction between *perception*, *intelligence* and *action*, as shown in Figure 1.

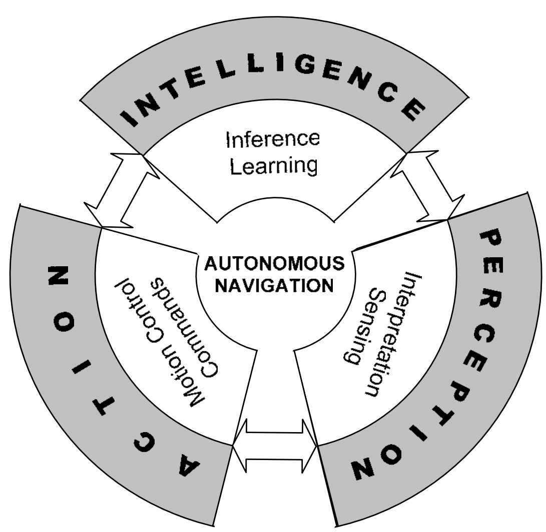


Fig. 1: Autonomous navigation conceptual loop.

Navigation of autonomous robotic vehicles in obstacle filled dynamic environments requires derivation and implementation of efficient real-time sensor based controllers. Effective control algorithms for autonomous navigation, should imitate the way humans are operating manned or similar vehicles. Considering the environment uncertainty that is difficult if not impossible to model, fuzzy logic is one of the most widely used mathematical tools for autonomous vehicle navigation (Driankov & Saffiotti, 2001). Fuzzy logic techniques have already been used and are being used currently for autonomous navigation of *ground* (indoors (Aguire & Gonzalez, 2000; Doitsidis, et al., 2002; Goodridge & Luo, 1994; Ishikawa, 1991; Li, 1994; Oriolo, et al., 1998; Pin & Watanabe, 1994; Tsourveloudis, et al., 2001; Tunstel, 1996) and outdoors (Hagras, et al., 2001; Seraji & Howard, 2002; Valavanis, et al., 2005), *aerial* (fixed (Doitsidis, et al., 2004; Nikolos, et al., 2003a) and rotary wings (Amaral, 2001; Hoffmann, et al., 1999; Kadmiry & Driankov, 2001; Kadmiry & Driankov, 2004; Sugeno, 1999) and *water* (surface (Vanick, 1997) or submersible (Kanakakis, et al., 2004; Kanakakis, et al., 2001)) robotic vehicles.

The wide applicability of fuzzy logic in autonomous navigation is mainly based on suitable knowledge representation of inherently vague notions achieved through fuzzy IF-THEN rules. These rules typically contain linguistic information, which describes the problem at hand very simple and fast. Further, in the majority of fuzzy logic application in navigation, a mathematical model of the dynamics of the vehicle is not needed in the design process of the motion controller. Only the problem-specific heuristic control knowledge is needed for the inference engine design. From a more practical point of view, fuzzy logic is the most appropriate modeling tool for representing imprecision and uncertainty of the sensor readings. Another reason that explains the popularity of fuzzy logic in autonomous navigation is the low computation time of the hardware implementations of fuzzy controllers which favors real-time applications.

This chapter presents implementations of fuzzy logic in the area of autonomous vehicles navigation. It discusses successful applications of collision free motion control of *ground*, *aerial* and *underwater* unmanned vehicles navigation. The common characteristic in all applications regardless of the type of vehicle is the navigation architecture used. This generic concept of fuzzy navigation architecture is discussed in the next section. Section 3 presents the implementation of the proposed generic architecture for ground, aerial and underwater robots. The chapter concludes with future research trends for unmanned vehicles.

2. Navigation Architecture

In the literature, the navigation problem is separated in two parts: *global* navigation concerned with generating a path leading to the goal point; and *local* navigation, which follows the global path avoiding collisions with obstacles. The solutions presented with the use of fuzzy logic fall more or less in the second category. (Saffiotti, 1993) discuss the problem of mixing the two essential navigation behaviors, that is, pure reactive and goal-oriented behaviors, by using fuzzy logic. Generally speaking, the approaches to fuzzy navigation in dynamic environments follow either a classical paradigm or a behavior-based paradigm. Fuzzy navigation schemes, which follow the classical paradigm, have one set of control rules that includes all situations that may arise. All rules operate at all times to generate the control law. Behavior based fuzzy navigation acknowledges that there are different types of behaviors which the autonomous vehicle must exhibit in different situations.

Each behavior is given a set of rules and an inference engine is used to determine which behavior (or combination of behaviors) needs to be invoked in the current situation. In both paradigms, the “reaction” is given by a set of rules, which describe the navigation priorities.

Fuzzy inference approaches tend to de-emphasize goal-directed navigation and focus more upon handling reactive and reflexive cases. The results of the fuzzy inference controllers generally do not tend towards optimal paths. However, surprise obstacles and rapidly moving obstacles are handled with more certainty compared to methodologies in which certain performance criteria should be optimized (Tsourveloudis, et al., 2001).

Regardless of the final navigation goal or the type of vehicles, some kind of sensor data management is needed. Sensor readings provide information about the environment and the vehicle itself. These readings are almost at all times erratic, incomplete or conflicting and should be further processed in order to provide meaningful information. This information is essential for the motion commands of the vehicle. The overall architecture of the proposed navigation schema is shown in Fig. 2.

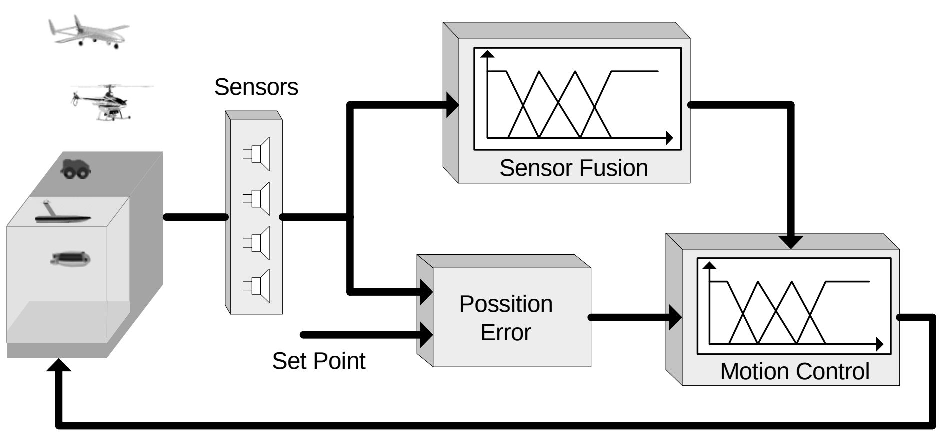


Fig. 2. Architecture of the fuzzy logic navigation scheme

The *sensor fusion* module is a fuzzy logic controller which takes as input the data provided by the various sensors and delivers information for eventual obstacles in respect to vehicle’s position and orientation. The interpreted obstacle information forms a collision possibility, which is send to the *motion control* module. The collision possibility together with position and/or orientation error are inputs of the motion fuzzy controller, which is responsible for the output commands to the driving devices. In further details, the first layer of the fuzzy logic inference engine performs sensor fusion from sensor readings, providing information about potential collisions in four directions, and the second layer guarantees collision avoidance with dynamic (moving) obstacles while following the desired trajectory. It has been shown (Tsourveloudis, et al., 2001), (Nikolos, et al., 2003b) that a path planning algorithm can be easily incorporated in the generic navigation architecture shown in Fig. 2

The architecture presented in Fig. 2 has been successfully applied to various unmanned vehicles as it is described in the following sections. In all these applications the basic idea of the layered fuzzy control is utilized in respect to the control demands of each robotic vehicle.

3. Physical Implementation

Some of the most difficult applications for robotics lie outdoors in dangerous and unknown environments. These include applications such as search and rescue (in land and sea), surveillance, humanitarian demining, underwater construction and mapping, environment monitoring, meteorology, agriculture and defense. Autonomous navigation of unmanned vehicles in unstructured environments is a multidiscipline and attractive challenge for researchers from academia and industry. The presentation that follows describes state-of-the-art applications of fuzzy logic that follow the architecture presented in the previous section. For most of the cases presented MATLAB have been used running either in Linux or Windows.

3.1 Ground Vehicles

The proposed navigation scheme was initially implemented on the *Nomad 200* mobile robot platform (Tsourveloudis, et al., 2001), for indoor navigation, and later on an ATRV skid steering mobile robot manufactured by *iRobot* (Doitsidis, et al., 2002).
The mobile robot ATRV-mini (shown in Fig. 3) has a ring of *24* sonar sensors that are placed around the vehicle and grouped in pairs A_i, $i=1,\ldots,12$, as shown in Fig. 3, each with an actual maximum measurement range of $2m$, returning data readings every $0.1s$. The data used for localization and calculation of the heading were produced from an odometer.

Fig. 3. DAMON: The ATRV - mini of the Intelligent Systems and Robotics Laboratory, Technical University of Crete.

For this robot system, a two-layer Mamdani-type controller has been designed and implemented (Doitsidis, et al., 2002). In the first layer, there are four fuzzy logic controllers responsible for obstacle detection and collision possibility calculation in the four main directions, ***front***, ***back***, ***left***, ***right***. The four controllers receive as inputs sonar sensor data and return as output the respective direction collision possibility.

Data from group sensors $A_1, A_2, \ldots, A_5$ (5 inputs) and group sensors $A_7, A_8, \ldots, A_{11}$ (5 inputs) serve as inputs to the individual controllers responsible for the calculation of the ***front*** and ***back*** collision possibilities, respectively. Data from group sensors A_5, A_6, A_7 (3 inputs) and group sensors A_{11}, A_{12}, A_1 (3 inputs) serve as inputs to calculate the ***left*** and ***right*** possibilities, respectively (Fig. 4).

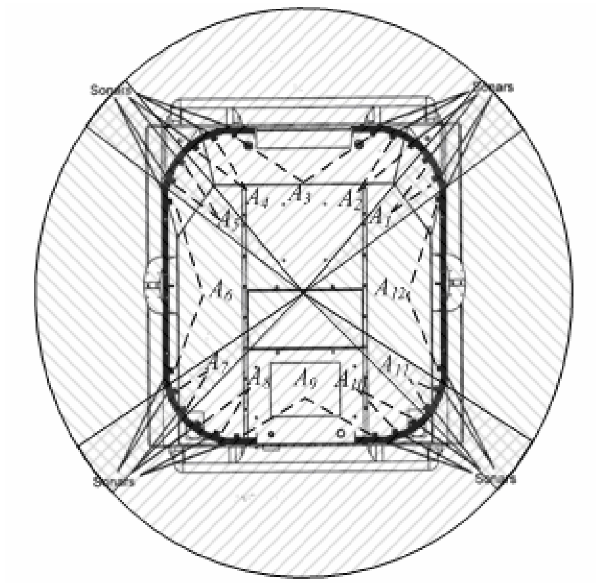


Fig. 4. Sonar grouping for the ATRV-mini.

The individual fuzzy controllers utilize the same membership functions to calculate the collision possibilities.

Collision possibilities are calculated using fuzzy rules of the type:

$$\text{R: IF } d_i \textit{ is } <LD^{(k)}> \text{ AND } d_{i+1} \textit{ is } <LD^{(k)}> \text{ THEN } c_j \text{ is } <LC^{(k)}>,$$

where k is the rule number, d_i represents sensors group i minimum readings, $LD^{(k)}$ is the linguistic variable of the term set D ={*near, medium_distance, away*}, c_j is the collision direction and $LC^{(k)}$ the variable with term set C={*not_possible, possible, high_possibility*}.

The overall output of the first layer is calculated using the *max-min* composition between the fuzzified readings:

$$\mu_C{}^*(c_j) = \max \min_{d_i} [\mu_D{}^*(d_i), \mu_{R^{(k)}}(d_i, c_j)] , \tag{1}$$

where $\mu_D{}^*(d_i)$ is the minimum of the fuzzified sonar readings and $\mu_{R^{(k)}}$ is the mathematical expression of the kth navigation rule.

The input variables to the second layer fuzzy controller are: a) the four *collision possibilities* with linguistic values {*not_possible, possible*, *high_possibility*; b) the *angle error* with linguistic values {*Backwards_1, Hard_Left, Left, Left2, Left1 Ahead, Right1, Right2, Right, Hard_Right, Backwards_2.*} The *angle_error* takes values from -180^o to 180^o and it is the difference between the desired and the actual heading of the vehicle.

The output variables are: a) *translational_velocity* with linguistic variables {*back_full, back_normal, back_slow, stop, front_slow, front_normal, front_full*} b) *rotational_velocity* with linguistic variables {*right_full, right, right1, no_rotation, left1, left, left_full*}. The *translational velocity* takes values that range from –1.5m/sec to 1.5 m/sec. The *rotational velocity* takes values ranging from –3rad/sec to 3 rad/sec. The number of linguistic values for the angle error, translational and rotational velocities is chosen after conducting several experiments to ensure smooth and accurate collision free navigation.

If the value of the translational velocity is positive the vehicle moves forward; if it is negative the vehicle moves backwards. A positive rotational velocity results in vehicle turn left; a negative value in vehicle turn right. Navigation and collision avoidance are performed using rules of the type:

$$\text{IF } c_j \text{ is } LC^{(k)} \text{ AND } \theta \text{ is } L\Theta^{(k)} \text{ THEN } tv \text{ is } LTV^{(k)} \text{ AND } rv \text{ is } LTV^{(k)},$$

where k is the rule number, c_j is collision of type j, i.e., the output of the obstacle detection module, θ is the angle error, tv is the translational velocity and rv is the rotational velocity. $LC^{(k)}$, $L\Theta^{(k)}$, $LTV^{(k)}$, $RTV^{(k)}$ are the linguistic variables of c_j, θ, tv, rv respectively. AND = min in all rules. The generic mathematical expression of the k^{th} navigation rule is:

$$\mu_{R^{(k)}}(c_j,\theta,tv,tr) = \min[(\mu_{LC^{(k)}}(c_j),\mu_{L\Theta^{(k)}}(\theta),\mu_{LTV^{(k)}}(tv),\mu_{LTR^{(k)}}(tr)]. \qquad (2)$$

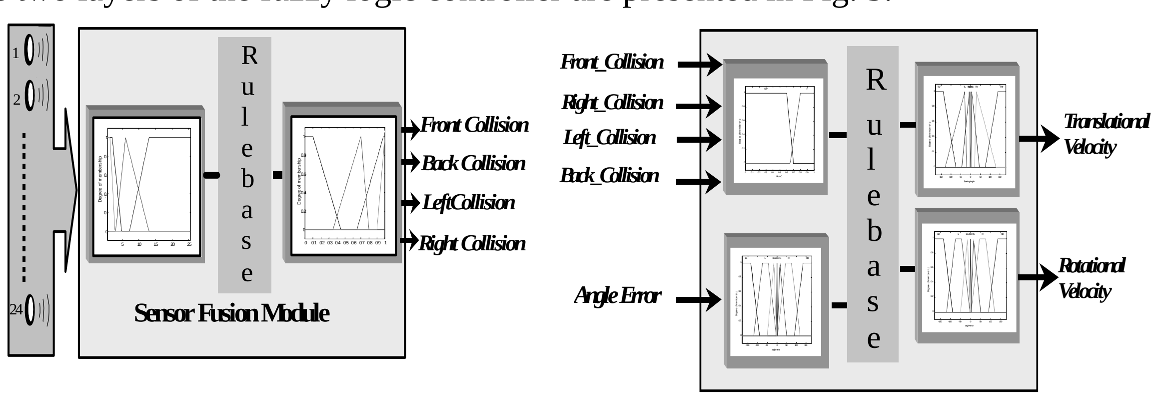
The overall navigation output is given by max-min composition:

$$\mu_N^*(tr,tv) = \max \min_{c_j,\theta}[\mu_{AND}^*(c_j,\theta),\mu_R(c_j,\theta,tr,tv)], \qquad (3)$$

where,

$$\mu_R(c_j,\theta,tv,tr) = \bigcup_{k=1}^{K} \mu_{R^{(k)}}(c_j,\theta,tv,tr). \qquad (4)$$

The two layers of the fuzzy logic controller are presented in Fig. 5.



a) b)

Fig. 5. a) The sensor fusion module, b) The motion control module.

a) b)

Fig. 6. Navigation in a cluttered environment a): Environment Map, b): Aggregation of sonar readings.

A modification of the proposed navigation scheme for outdoors environment was implemented in an ATRV-Jr. robot equipped with a different sensor suite, including a SICK LMS 200 scanning planar laser and a GPS system (Valavanis, et al., 2005). The implementation have been done in MATLAB running on-board the actual vehicles connected with the *JMatlink class* with Java which was responsible for handling all the processes which were running on the vehicles.

It consists of four modules: the laser range filter, position detection, heading error calculation and the actual fuzzy logic robot controller. The control system receives as inputs laser, odometer and GPS data, as well as a control reference input (next waypoint or goal point). It outputs actuator commands in terms of robot rotational and translational velocity.

The fuzzy logic controller is implemented as a Mamdani-type controller similar to previous work (Tsourveloudis, et al., 2001; Doitsidis, et al., 2002). The fuzzy logic controller rule base includes the fuzzy rules responsible for vehicle control. The inference engine activates and applies relevant rules to control the vehicle. The fuzzification module converts controller inputs into information used by the inference engine. The defuzzification module converts the output of the inference engine into actual outputs for the vehicle drive system.

The fuzzy controller input from the filtered laser range block consists of a three value vector with components related to the distance of the closest object in the left sector of the scan, in the center sector and in the right sector, respectively. The sectors are presented in Fig. 7.

This information is used to calculate three collision possibilities *left*, *center*, *right* reflecting potential static / dynamic obstacles in the robot field of view, similar to the approach followed in (Tsourveloudis, et al., 2001; Doitsidis, et al., 2002) but for outdoor environments. The fourth input to the fuzzy logic controller is the robot's heading error calculated from the robot's current heading and the desired heading.

Implementation wise, each of the three aggregate range inputs includes three trapezoidal membership functions namely, *close*, *medium* and *far*. The input linguistic variables are denoted as *left distance*, *right distance* and *center distance* corresponding the left area, right area and center area sectors. The *heading error* input uses four trapezoidal membership functions and one triangular membership function. They are empirically derived based on extensive tests and experiments.

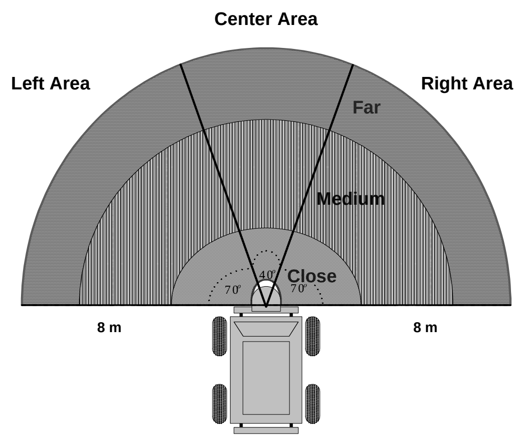


Fig. 7. Laser scanner sectors.

Each distance input variable d_i (corresponding to left area, center area, right area) is fuzzified and expressed by the fuzzy sets C_i, MD, A_i referring to *close*, *medium*, and *far*. The range of the membership functions for each d_i is between 0-8 meters. The input variable *heading error, he,* is fuzzified and expressed by the fuzzy sets *FL, L, AH, R, FR*, referring to *far left*, *left*, *ahead*, *right*, and *far right,* respectively. The range of the membership functions for the *heading error,* is between -180 and 180 degrees.

The fuzzy logic controller has two output variables, *translational velocity* (*tr*) implemented with two trapezoidal and one triangular membership functions, and *rotational velocity* (*rv*) implemented with four trapezoidal membership functions and one triangular membership function.

The output variable *tr* is expressed by the fuzzy sets *ST, SL, F* referring to *stop*, *slow*, and *fast*, while the output variable *rv* is expressed by the fuzzy sets *HRR, RR, AHR, LR, HLR* referring to *hard right*, *right*, *ahead*, *left*, *hard left*.

The output commands are normalized in a scale from 0 to 1 for the translational velocity, where 0 corresponds to complete stop and 1 to maximum speed. Rotational velocity output commands are normalized from -1 to 1, where -1 corresponds to a right turn with maximum angular velocity and 1 to a left turn with maximum angular velocity. Each fuzzy rule *j* is expressed as:

IF d_1 is D_{j1} AND d_2 is D_{j2} AND d_3 is D_{j3} AND *he* is HE_j THEN *tr* is TR_j AND *rv* is RV_j;

for j=1,…, number of rules. D_{ji}, is the fuzzy set for d_i in the *jth* rule which takes the linguistic value of C_i, *MD, A*. HE_j is the fuzzy set for the *he* which takes the linguistic values *FL, L, AH, R, FR*. TR_j and RV_j are the fuzzy sets for *tr* and *rv* respectively.

The generic mathematical expression of the *jth* navigation rule is given by:

$$\mu_{R^{(j)}}(d_i, he, tr, r) = \min[\mu_{D_{ji}}(d_i), \mu_{HE^{(j)}}(he), \mu_{TR^{(j)}}(tr), \mu_{RV^{(j)}}(rv)]. \tag{5}$$

The overall navigation output is given by the max-min composition and in particular:

$$\mu_N^*(tr, vr) = \max_{d_i, he} \min[\mu_{AND}^*(d_i, he), \mu_R(d_i, he, tr, rv)], \tag{6}$$

where $\mu_R(d_i, he, tr, rv) = \bigcup_{j=1}^{J} \mu_{R^{(i)}}(d_i, he, tr, rv)$. The navigation action dictates change in robot speed and/or steering correction and it results from the deffuzification formula, which calculates the center of the area covered by the membership function computed from (6).

In order to validate the proposed scheme experiments performed in an outdoor environment which had grass, trees and some vegetation. Different set of experiments included waypoint navigation based on static predefined points while avoiding static and dynamic obstacles, raster scan of a certain area with multiple robots navigating and in the same time avoiding each other. Sample pictures of the ATRV-Jr. moving around in an area with trees are presented in Fig. 8.

Fig. 9 shows one full path traveled through an initial point to the final point, in a tree covered area while periodic laser scans are shown over the course of the robot's path.

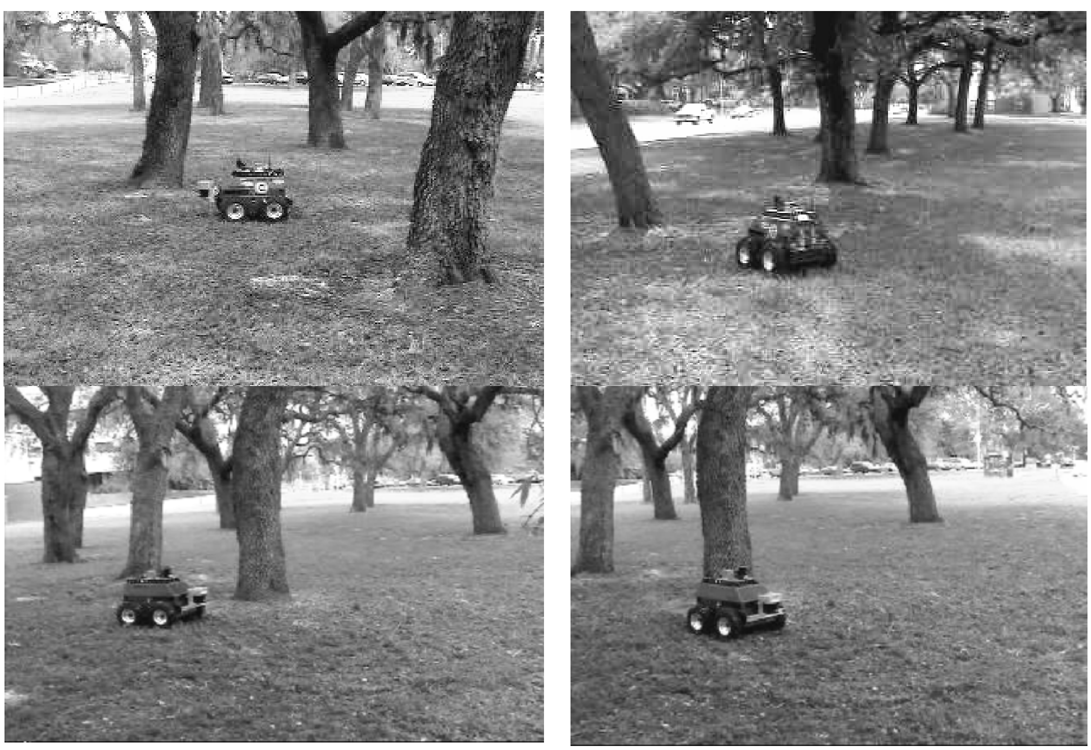


Fig. 8. Unmanned Ground Vehicle navigating in an area with trees.

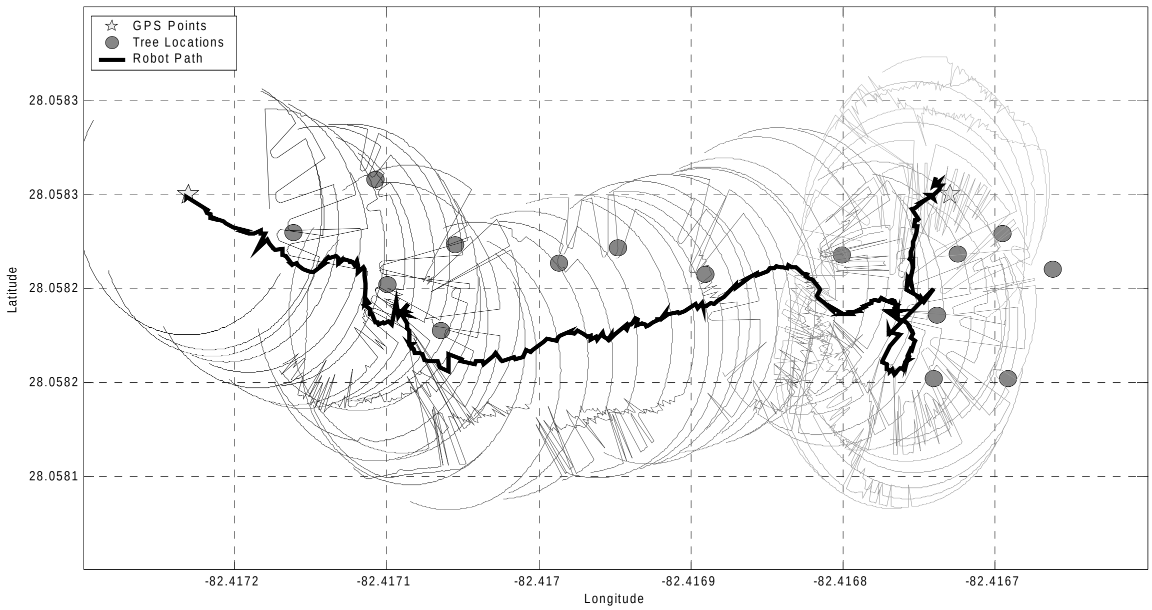


Fig. 9. Robot path in an outdoor environment.

3.2 Aerial Vehicles

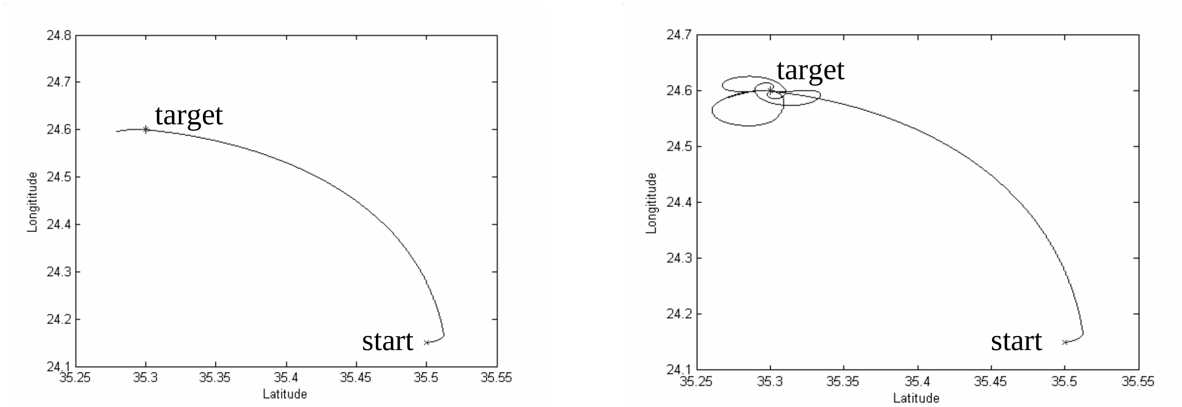
Implementations of the proposed navigation scheme for fixed wing unmanned aircrafts are presented in (Doitsidis, et al., 2004b) and in a similar approach in (Nikolos, et al., 2003a).

In (Doitsidis, et al., 2004b) a two module fuzzy logic based autonomous navigation system which is capable i) to fly through specified waypoints in a 3-D environment repeatedly, ii) to perform trajectory tracking, and, iii) to duplicate / follow another aerial vehicle. Two fuzzy logic control modules are responsible for altitude control and latitude-longitude control; when combined, they may adequately navigate the aerial vehicle. All input and output linguistic variables have a finite number of linguistic values with membership functions empirically defined.

The altitude fuzzy logic controller has three inputs, that is a) altitude error, b) change of altitude error, and, c) airspeed. The altitude error is the difference between the desired altitude and the current altitude of the airplane. The change of altitude error indicates whether the aerial vehicle is approaching the desired altitude or if it is going away from it. The airspeed is the current speed of the vehicle. Outputs are the elevators command and the throttle command, responsible for the decent and accent of the aerial vehicle. The latitude-longitude controller has as inputs the heading error and the change of heading error. The heading error is the difference between the desired and the actual heading of the airplane. The output is the roll angle of the airplane.

A simulation environment has been implemented in MATLAB. Vehicle’s motion dynamics were adopted from the *Aerosim Block Set* that can be integrated in SIMULINK. The simulation test bed consists of the following subsystems: 1) Aircraft model, 2) Altitude Fuzzy Logic Controller, 3) Latitude-Longitude Fuzzy Logic Controller, and 4) Error Calculating block.

Experimental results are presented in Fig. 10. In Fig. 10a the vehicle is passing through from a certain point.



a) b)

Fig. 10. Trajectories followed by the vehicle.

Further, the NEARCHOS UAV, presented in Fig. 12, has been used as a test bed. The flight behavior of this UAV has been modeled in terms of simple analytic relationships, which proved very helpful in representing its actual flight in the horizontal plane. A fuzzy controller for the autonomous navigation on the horizontal plane, has been developed in (Nikolos, et al., 2003a). The controller inputs are the heading error of the aircraft and its current roll angle, while the output is the change command of the roll angle. The basic purpose of the navigation system was, to make the vehicle able to follow a predefined trajectory.

Even though the current roll angle takes values ranging from $–90^0$ to 90^0, the flight control system of the tested vehicle functions safely in a range from 70^0 to 70^0. The linguistic variables that represent the current roll angle are: *Right_Big (rb), Right_Medium (rm), Right_Small (rs), Zero, Left_Big (lb), Left_Medium (lm), Left_ Small (ls).* The second input to the fuzzy controller is the heading error, which is defined as the difference between the desirable and the factual direction of the aircraft. The factual direction is the heading of the aircraft, which is provided from the GPS. The desirable direction is the heading of a vector, with a starting point the current aircraft's position and ending point the desirable position. The linguistic variables that represent the heading error are: *Negative_Big (nb), Negative_Medium (nm), Negative_Small (ns), Zero, Positive_Big (pb), Positive_Medium (pm), Positive_Small (ps).* The membership functions of the input variables are presented in Fig. 11.

The desired and the actual heading direction take values ranging from 0^0 to 360^0, whereas the heading error takes values ranging from -180^0 to 180^0. However, in this implementation the heading error takes values in the region $[-100^0, 100^0]$. Negative (positive) values of heading error correspond to desirable right (left) roll. The linguistic variables that represent the heading error are: *Negative_Big (nb),* *Negative_Medium (nm), Negative_Small (ns), Zero, Positive_Big (pb), Positive_Medium (pm), Positive_Small (ps).*

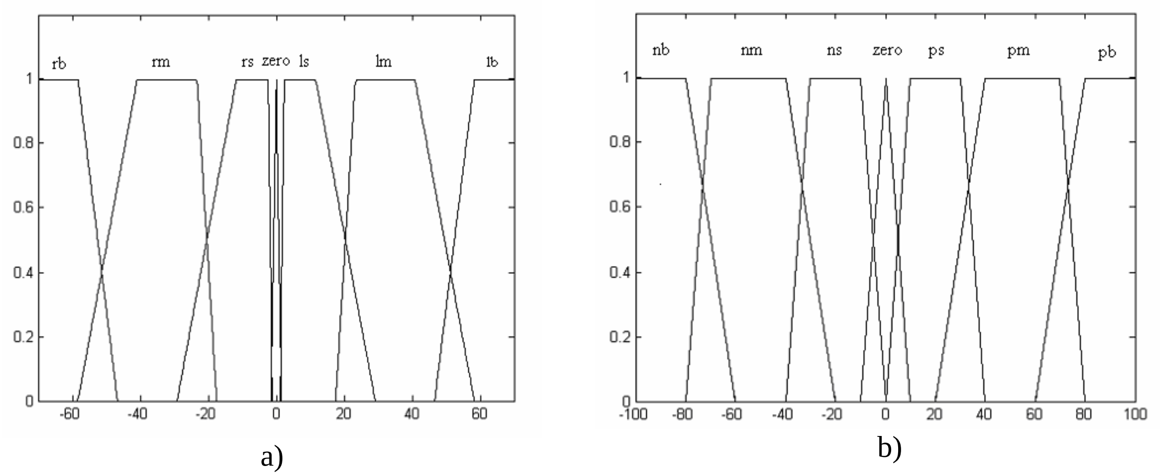


Fig. 11. Membership functions plot of input variables: a) Current Roll, b) Heading Error.

Fig. 12. The UAV NEARCHOS (Property of EADS – 3 SIGMA S.A)

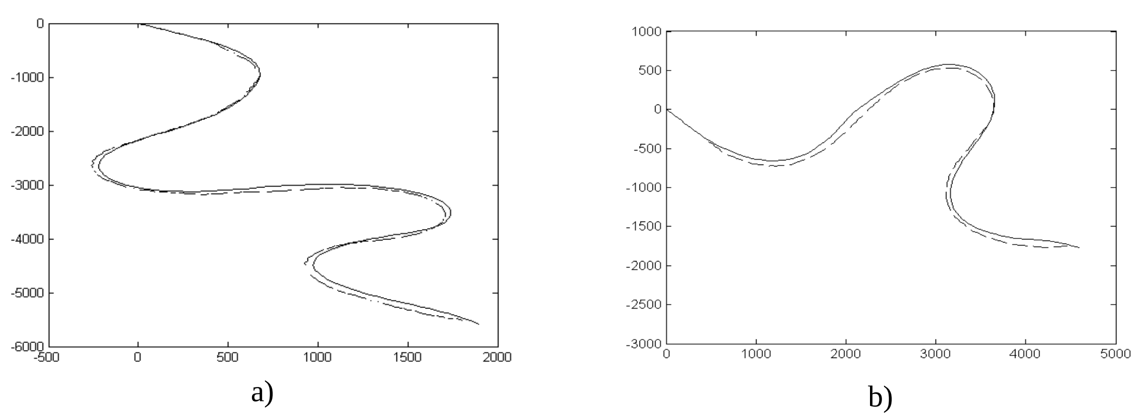


Fig. 13. Trajectories followed by the UAV. Dashed lines represent the observed path while solid lines are the desired trajectories.

Experimental results about how the fuzzy logic controller performed in terms of trajectory following are presented in Fig. 13, where the continuous and discontinuous lines represent the desired and the trajectory that the fuzzy logic controller forced the vehicle to follow respectively.

3.3 Underwater Vehicles

Most of the difficulties in navigation of underwater vehicles (see Fig. 14) are due to the inherently uncertain nature of these environments. Fuzzy logic offers features that significantly help in addressing such problems (Kato, 1995; Tsourveloudis, et al., 1998; Kanakakis, et al., 2004). Here, we present an overview of the fuzzy logic implementations for the navigation of *Autonomous Underwater Vehicles* (AUVs) introduced in (Tsourveloudis, et al., 1998; Kanakakis, et al., 2001; Kanakakis, et al., 2004).

Some of the already known generic problems in autonomous navigation remain in the area of AUVs. These problems are, the *sensor fusion problem*: how to fuse information from different sensor modalities, or different readings from the same sensor; *the coherence problem*: how to register representations used at different levels of abstraction, and *the coordination problem*: How to coordinate the simultaneous activity of several, possibly competing behaviors such as collision avoidance and goal reaching. Additional problems in the 3-D autonomous underwater navigation are: 1) Unknown environments: Poor map and perceptual information, currents with unpredictable behaviour, and 2) Very *limited communication* options with the vehicle.

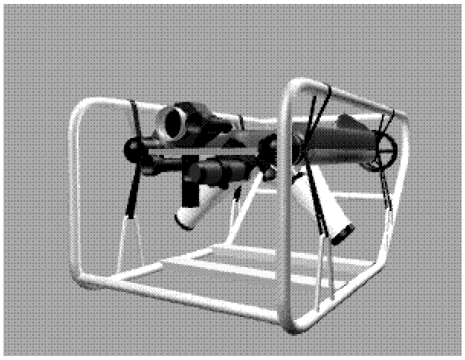


Fig 14**.** The underwater vehicle *Phantom S2*.

Fuzzy logic navigation solutions have shown a good degree of robustness, which is crucial in the area of underwater robotics, where: 1) sonar data is unreliable, 2) mathematical models about the environment and the vehicle are usually not available, and 3) the only available navigation expertise is due to vehicle operators.

The aim of underwater navigation is to guide the vehicle at a predefined point, by controlling various parameters such as pitch, yaw etc. The desired values of these parameters are the control objectives at any time instant. The fuzzy rules, which contain the navigation parameters, aim towards two goals: ideal trajectory tracking and collision avoidance. The generic expression of a fuzzy rule could be:

IF *trajectory-condition* AND *obstacle-condition* THEN *vehicle-action.*

The navigation architecture proposed for the underwater vehicles enables two layers of fuzzy controllers. These controllers can provide accurate and collision free navigation of an AUV in ocean environments, considering position accuracy with the presence of ocean currents and providing vertical stability to the vehicle. Similar to the generic architecture described in Section 2, the autonomous underwater scheme has the following modules:

- The **sensor fusion/collision avoidance module**, where the readings of the sensors of the vehicle are provided to estimate the position of the vehicle and the collision possibility in all surrounding directions. The sensor fusion module is responsible for position monitoring and obstacle detection. As AUVs operate in unknown or poorly mapped ocean environments, static or moving obstacles find themselves in the desired path of the vehicle. In these cases the vehicle should be able to use it's on board sensors to monitor its position and to detect moving or static obstacles. This implies the use of a number of different kinds of sensors, like vision cameras, laser sensors, magnetic compasses, gyroscopic mechanisms and sonar sensors. For most cases where vision is poor, sonar sensors are used to estimate an underwater environment.
- The **motion control module**, which performs low-level control of the vehicle's propellers, thrusters and fins in order to reach the determined goal point having the target surge velocity. The inputs are the goal point and the actual position and orientation, in earth-fixed coordinates, the target surge velocity and the vector of the actual vehicle velocities in body-fixed coordinates, and the sea current velocity.

Since the design of fuzzy controllers does mot require any strict modeling of the vehicle's behavior the above design is adopted for its simplicity, considering that it can be applied in all types of AUVs.

3.3.1 The Sensor Fusion/Collision Avoidance Module

The sensor fusion module outputs the collision possibility in front, right, left and back directions. The output linguistic variables are: *front_collision, right_collision, left_collision, back_collision*, taking the linguistic values *not possible, possible, high*. The collision possibilities in the four cardinal directions are computed from fuzzy rules of the type:

$$\text{IF } d_i \textit{ is } \langle LD^{(k)} \rangle \text{ THEN } c_j \text{ is } \langle LC^{(k)} \rangle,$$

and for example: IF sonar 1 *distance* is <*close*> THEN *front_collision* is <*high*>, where, k is the rule number, d_i represents the readings of the sensor i, $LD^{(k)}$ is the linguistic variable of the term set $D = \{close, near, far\}$, c_j is the collision of type j ($j \in \{$*not possible , possible, high*$\}$).

A second fuzzy controller in the same module is responsible for the collision avoidance. It takes as inputs: a) *collision possibilities* with linguistic values *not possible and high*, b) *head_error* with linguistic values*: left_big, left, left_small, zero,* *right_small, right,* and *right_big*, and c) *pitch_error* with linguistic values *down_big, down, down _small, zero, up_small, up,* and *up_big*. The output variables are: a) *head_change* with linguistic variations, such as, *left_fast, left, left_slow, zero, right_slow, right,* and *right_fast*, b) *pitch_change* with linguistic values *down_fast, down, down _slow, zero, up_slow, up* and, *up_fast*, c) *surge_speed* with linguistic values *slow, normal,* and *high*. The collision avoidance controller consists of rules of the following form:

IF c_j is $LC^{(k)}$ AND ψ is $L\Psi^{(k)}$ AND θ is $L\Theta^{(k)}$, THEN $d\psi$ is $LD\Psi^{(k)}$ AND $d\theta$ is $LD\Theta^{(k)}$ AND u is $LDU^{(k)}$,

where, k is the rule number, c_j is the collision of type j, ψ is the heading error, θ is the pitch error, u is the vehicle's surge speed, and $LC, L\Psi, L\Theta\ LD\Psi, LD\Theta, LDU$ are the linguistic variables of c_j, $\psi, \theta, d\psi, d\theta, u$ respectively.

3.3.2 The Motion Control Module

The overall motion controller consists of the following subsystems:

- The *speed control* subsystem is responsible for the vehicle's speed by controlling its propellers revolution rate.
- The *heading control* subsystem controls the steering in the horizontal plane by controlling vehicle's head angle.
- The *depth control* subsystem controls the motion of the AUV in the vertical plane by regulating vehicle's pitch angle and depth.
- The *roll control* subsystem controls the roll parameter of the motion of the AUV.
- The *ocean current* subsystem adjusts the position of vehicle in case of undersea currents. Under the presence of a sea current, the vehicle has a drift and a deviation from the originally planned course. Although this deviation can be considered in both the speed and steering controllers, this controller adds maneuverability by modifying the steering controls. Its aim is to overcome the lateral drag by modifying the desired head and pitch angle.

The overall architecture of the fuzzy logic based navigation is shown in Fig. 15.

A vehicle control action (a fin angle, a thruster voltage or a desired propeller revolution rate) may be commanded from more than one of the above subsystems; thus, for each commanded action and during each simulation step the outputs from all subsystems form a *control vector* that controls the actual vehicle. The values of this control vector are bounded within the operational limits of vehicle servomotors to reflect reality. It should be noted that the effect of ocean currents of different velocity is taken into consideration in all phases of design and testing.

3.3.3 Simulation Results

The proposed architecture is applied to *Phoenix* AUV of the Naval Postgraduate School at Monterey, California, USA. Its dimensions and hydrodynamic model are given with clarity in the (Brutzman, 1994). The NPS-Phoenix AUV is neutrally buoyant and has a hull length of 7.3 ft. It has four paired plane surfaces (eight fins total) and four paired thrusters built in cross-body tunnels. It has two screw bi-directional propellers. Its design depth is 20 ft (6.1 m) and the hull is made of press and welded aluminum. The vehicle endurance of 90-120 min is supported by a pair of lead-acid gel batteries at speeds up to 2 ft/sec (0.61 m/sec).

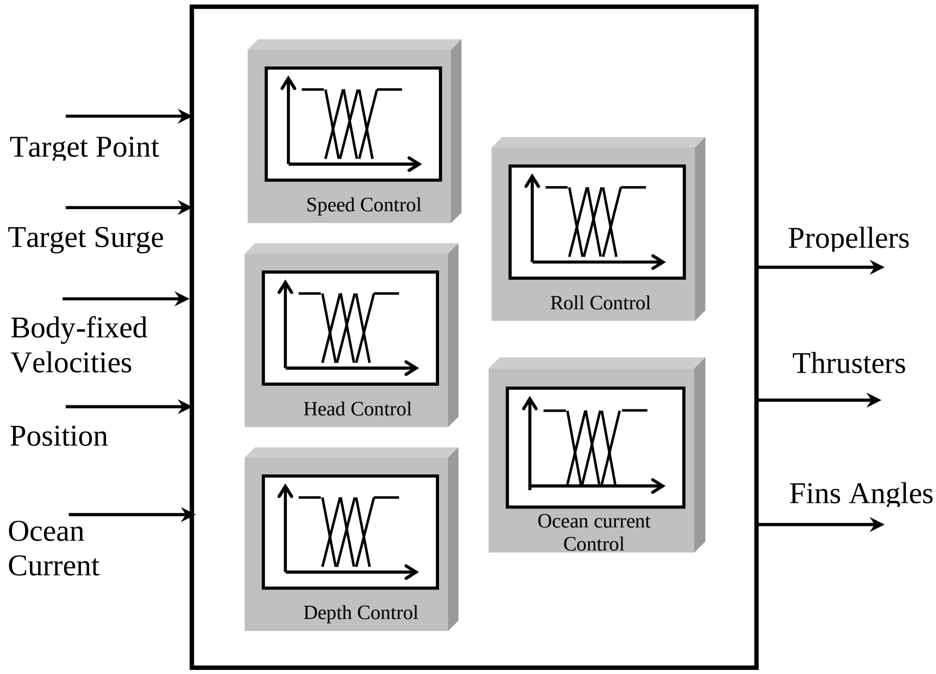


Fig. 15. The motion control module

The behavior of the vehicle was examined under various situations including step response, ocean current, and smooth curve path. The overall performance of the controller was found to be encouraging for further research and refinement. Fig. 16 show the AUV following a rectangle saw-tooth curve in the horizontal plane and gradually descents and ascents in the vertical plane. Ocean current is present. The vehicle is simulated under ocean current with Y (earth-fixed) velocity of 0.3, 0.6, 0.8 1.0, 1.2 ft/sec. Fig. 17a, b) show the trajectories presented in Fig. 16 in vertical and horizontal planes, respectively.

4. Conclusions

The technology of unmanned vehicles, in all its aspects, is an exciting one, especially since it holds the promise of saving human lives by letting machines do dull, dirty or dangerous missions into high-threat environments or just unknown environments

This chapter shows how core research on fuzzy logic affects the advances in unmanned vehicles navigation technology and the way it can be applied to a variety of robotic vehicles, operating on ground, air or underwater. Furthermore, three key attributes for a vehicle to be considered as capable for performing autonomous navigation have been identified. *Perception* which is the ability of the vehicle to acquire knowledge about the environment and itself; *intelligence* which is the ability of a vehicle to operate for a considerable amount of time without human intervention, and *action* which is the ability of the vehicle to travel from point A to point B. How capable is a vehicle to perform these different functions is the metric to evaluate the degree of its autonomy.

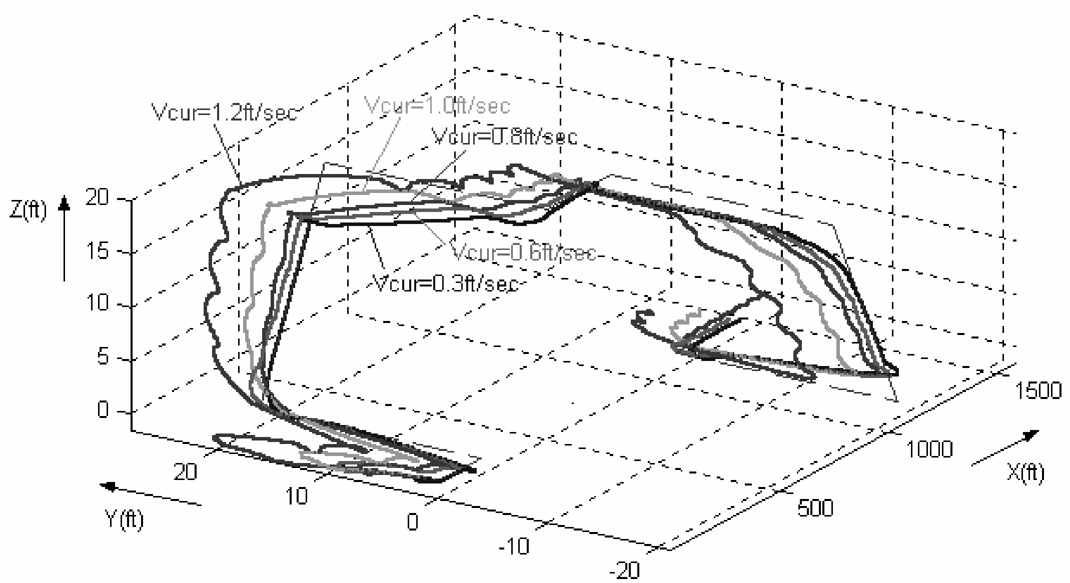
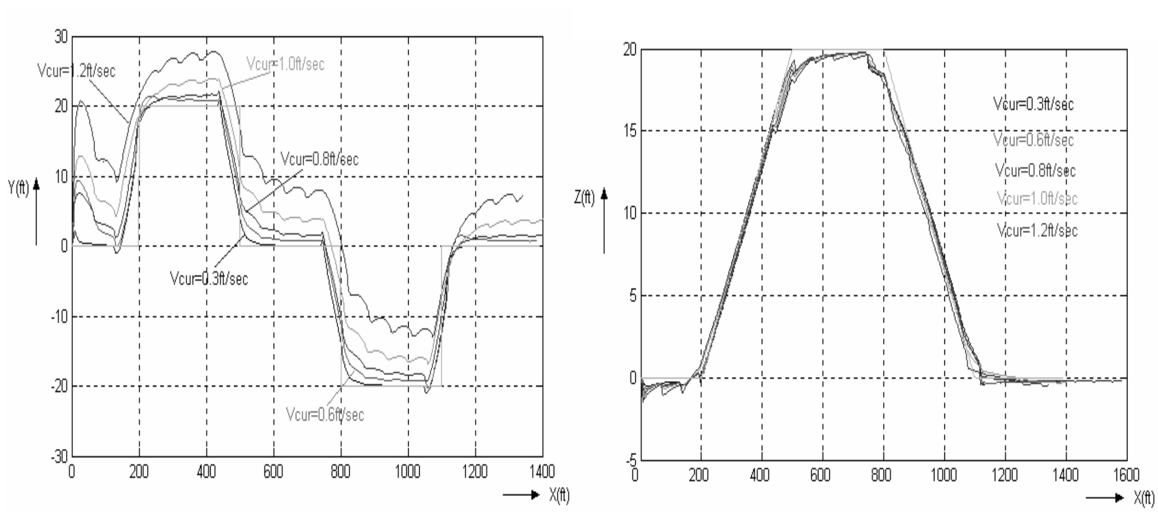


Fig. 16. Simulated AUV trajectory for various ocean current velocities.



a) b)

Fig. 17. AUV paths for various current velocities: a) in XY plane, b) in XZ plane.

Based on these attributes, fuzzy logic has been identified as a useful tool for developing controllers for vehicles so that they will be able to perform autonomous navigation. A two layer fuzzy logic controller architecture has been described. The first layer is the sensor fusion module in which the vehicle evaluates readings from various sensors and interacts with the environment.

In the second layer, which is the motion control module, the information derived from the previous layer is combined with other parameters i.e. heading, speed, altitude, position etc. This layer outputs the actual commands that will move the vehicle towards its mission.

This architecture has been proven effective in almost all types of robotic vehicles (see for example: Doitsidis, et al., 2002; Doitsidis, et al., 2004; Kanakakis, et al., 2001; Kanakakis, et al., 2004; Nikolos, et al., 2003a; Tsourveloudis, et al., 1998; Tsourveloudis, et al., 2001; Valavanis, et al., 2005).

In the next decade advances in technologies such as, Computers and Communications, Electronics, Computer-integrated manufacturing and Materials Design, will drastically support unmanned robotics to mature while dropping their costs. This will lead to a dramatic growth of unmanned robotic systems, a fact that will further affect the consumer, education, research, business and military markets.

5. References

Aguire, E. & Gonzalez, A. (2000). Fuzzy Behaviors for Mobile Robot Navigation: Design, Coordination and Fusion. *International Journal of Approximate Reasoning*, Vol. 25, 225-289, ISSN 0888-613X.

Amaral, T. G. B., & Crisostomo, M. M. (2001). Automatic Helicopter Control Motion using Fuzzy Logic. *Proceedings of the IEEE International Fuzzy Systems Conference*, Vol. 3, pp. 860-863, December 2001, Melbourne, Australia.

Brutzman, D, (1994).A virtual world for an autonomous underwater vehicle. Ph.D. Thesis, Naval Postgraduate School, Monterey, California, U.S.A.

Doitsidis, L.; Valavanis, K. P.; & Tsourveloudis, N. C. (2002a). Fuzzy Logic Based Autonomous Skid Steering Vehicle Navigation. *Proceedings of the IEEE International Conference on Robotics and Automation,* pp. 2171-2177, May 2002, Washington DC, U.S.A.

Doitsidis, L.; Valavanis, K. P.; Tsourveloudis, N. C. & Kontitsis, M. (2004b). A Framework for Fuzzy Logic Based UAV Navigation and Control. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4041-4046, April 2004, New Orleans, U.S.A.

Driankov, D. & Saffiotti, A. (2001). *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*. Physica – Verlag, ISBN 3790813419, Heidelberg.

Goodridge, S. C. & Luo, R. C. (1994). Fuzzy Behavior Fusion for Reactive Control of an Autonomous Mobile Robot: MARGE. *Proceedings of the IEEE International Conference on Robotics and Automation,* pp. 1622-1627, May 1994, San Diego, U.S.A.

Hagras, H.; Callaghan, V. & Colley, M. (2001). Outdoor Mobile Robot Learning and Adaptation. *IEEE Robotics and Automation Magazine*, pp. 53-69, ISSN 1070-9932.

Hoffmann, F.; Koo, T. J. & Shakernia, O. Evolutionary Design of a Helicopter Autopilot. *Advances in Soft Computing - Engineering Design and Manufacturing, Part 3: Intelligent Control*, Springer-Verlag.

Ishikawa, S. (1991). A Method of Indoor Mobile Robot Navigation by Using Fuzzy Control. *Proceeding of the IEEE/RSJ International Workshop on Intelligent Robots and Systems,* pp. 1013-1018, November 1991, Osaka, Japan.

Kadmiry, B. & Driankov, D. (2001). Fuzzy Control of an Autonomous Helicopter. *Proccedings of the IFSA World Congress and 20th NAFIS International Conference,* pp. 2797-2802, July 2001, Vancouver, Canada.

Kadmiry, B. & Driankov, D. (2004). A fuzzy gain-scheduler for the attitude control of an unmanned helicopter. *IEEE Transactions on Fuzzy Systems***,** Vol. 12, 502-515.

Kanakakis, V.; Tsourveloudis, N. C. & Valavanis, K. P. (2001). Design and testing of a fuzzy logic controller for underwater vehicles. *Proceedings of the IARP International Workshop on Underwater Robotics for Sea Exploration and Environmental Monitoring,* Rio de Janeiro, Brazil.

Kanakakis, V.; Valavanis, K. P. & Tsourveloudis, N. C. (2004). Fuzzy-Logic Based Navigation of Underwater Vehicles. *Journal of Intelligent and Robotic Systems*, Vol. 40, 45-88.

Kato, N.; (1995). Applications on fuzzy algorithm to guidance and control of underwater vehicles. Yuh. J. (ed), Underwater Robotic Vehicles: Design and Control, TSI Press.

Li, W. (1994). Fuzzy Logic-based ‘Perception-Action’ Behavior Control of a Mobile Robot in Uncertain Environment. *Proceedings of the 3rd IEEE Conference on Fuzzy Systems,* pp. 1626-1631, March 1994, San Francisco, U.S.A.

Nikolos, I. K.; Doitsidis, L.; Christopoulos, V. N. & Tsourveloudis, N. C. (2003a). Roll Control of Unmanned Aerial Vehicles using Fuzzy Logic. *WSEAS Transactions on Systems*, Vol. 4, 1039-1047.

Nikolos, I. K.; Valavanis, K. P.; Tsourveloudis, N. C. & Kostaras, A. N. (2003b). Evolutionary algorithm based offline/online path planner for UAV navigation. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Vol. 33, 898 – 912.

Oriolo, G.; Ulivi, G. & Vindittelli, M. (1998). Real-time Map Builting and Navigation for Autonomous Robots in Unknown Environments. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 28, 316-333.

Pin, F. G. & Watanabe, Y. (1994). Navigation of Mobile Robots Using Fuzzy Logic Behaviorist Approach and Custom Design Fuzzy Inference Boards. *Robotica,* Vol. 12.

Saffiotti, A.; Ruspini, E. H. & Konoligr, K. (1993). Blending Reactivity and Goal-Directedness in a Fuzzy Controller. *Proceedings of the Second IEEE Conference on Fuzzy Systems*, pp. 134-139, March 1993, San Francisco, U.S.A.

Seraji, H. & Howard, A. (2002), Behavior-Based Robot Navigation on Challenging Terrain: A Fuzzy Logic Approach. *IEEE Transactions on Robotics and Automation*, Vol. 18, 308-321.

Sugeno, M. (1999). Development of an Intelligent Unmanned Helicopter. *Fuzzy Modeling and Control: Selected Works of M. Sugeno*, H. T. Nguyen and N. R. Prasad (eds), CRC Press, Boca Raton.

Tsourveloudis, N. C., Gracanin, D., Valavanis, K. P. (1998). Design and testing of navigation algorithm for shallow water autonomous underwater vehicle. *Proceedings of the IEEE OCEANS 98*, Nice, France.

Tsourveloudis, N. C.; Valavanis, K. P. & Hebert, T. (2001). Autonomous Vehicle Navigation Utilizing Electrostatic Potential Fields and Fuzzy Logic. *IEEE Transactions on Robotics and Automation*, Vol. 17, 490-497.

Tunstel, E. (1996). Mobile Robots Autonomy Via Hierarchical Fuzzy Behavior. *Proceedings of the 6th International Symposium on Robotics and Manufacturing,* pp. 837-842, May 1996, Montpelier, France.

Valavanis. K. P.; Doitsidis. L.; Long. M. T. & Murphy. R. R (2005).Validation of a Distributed Field Robot Architecture Integrated with a MATLAB Based Control Theoretic Environment: A Case Study of Fuzzy Logic Based Robot Navigation", *IEEE Robotics and Automation Magazine,* (In press).

Vaneck, T.W. (1997). Fuzzy Guidance Controller for an Autonomous Boat. *IEEE Control Systems Magazine*, Vol. 17. 43-51.

White, B. A.; Blumel, A. L. & Hughes, E. J., (2000). A Robust Fuzzy Autopilot Design Using Multi-Criteria Optimization. *International Journal of Fuzzy Systems*, Vol. 2, pp. 129-138.