Journal of Intelligent and Robotic Systems **40:** 45–88, 2004.

Fuzzy-Logic Based Navigation of Underwater Vehicles

V. KANAKAKIS, K. P. VALAVANIS⋆ and N. C. TSOURVELOUDIS
Technical University of Crete, Intelligent Systems and Robotics Laboratory,
Department of Production Engineering and Management, 73100 Chania, Crete, Greece

(Received: 10 July 2003; in final form: 17 March 2004)

Abstract. A fuzzy logic based general purpose modular control architecture is presented for underwater vehicle autonomous navigation, control and collision avoidance. Three levels of fuzzy controllers comprising the *sensor fusion module*, the *collision avoidance module* and the *motion control module* are derived and implemented. No assumption is made on the specific underwater vehicle type, on the amount of *a priori* knowledge of the 3-D undersea environment or on static and dynamic obstacle size and velocity. The derived controllers account for vehicle position accuracy and vertical stability in the presence of ocean currents and constraints imposed by the roll motion. The main advantage of the proposed navigation control architecture is its simplicity, modularity, expandability and applicability to any type of autonomous or semi-autonomous underwater vehicles. Extensive simulation studies are performed on the *NPS Phoenix* vehicle whose dynamics have been modified to account for roll stability.

Key words: fuzzy logic navigation, sonar sensors, collision avoidance, motion control, ocean current, collision possibility, goal based navigation, reaction based navigation.

1. Introduction

This paper is the outgrowth and overall generalization of recently published research on fuzzy logic based autonomous mobile robot path planning and collision avoidance [15, 16, 37, 39] as well as on underwater vehicle path planning [29].

The central objective of this paper is to present the fundamentals of general modular control architecture for sonar sensor based autonomous underwater vehicle (AUV) navigation in 3-D unknown (undersea) environments. The derived control architecture is tested and validated on the *NPS Phoenix* underwater vehicle whose model has been modified accordingly to account for roll motion control (although the vehicle is self-stabilized in roll [8]). The main reason for using the *NPS Phoenix* AUV for extensive simulation studies is due to the fact that its dimensions, specifications and hydrodynamic model and coefficients have been accurately defined [7, 8].

⋆ Corresponding author. Department of CSEE, Univeristy of South Florida, Tampa, FL, USA; e-mail: kvalavan@csee.usf.edu

When deriving the overall control architecture, no assumption is made on the AUV type, on the amount of *a priori* knowledge of the 3-D undersea environment (from potentially existing undersea environment maps), or on static and dynamic obstacle size and velocity. The derived controllers account for AUV position accuracy and vertical stability in the presence of ocean currents and constraints imposed by the AUV roll motion. The main advantage of the proposed navigation control architecture is its simplicity, modularity, expandability and applicability to any type of autonomous or semi-autonomous underwater vehicle (ROV or AUV).

A fuzzy logic framework is used for navigation and control. Three levels of fuzzy controllers comprising the *sensor fusion module*, the *collision avoidance module* and the *motion control module* are derived and implemented.

The rationale behind using fuzzy logic is that fuzzy logic has already been proven to be a very useful modeling tool when dealing with problems characterized by the presence of uncertainty [14, 17, 33, 34, 45]; in this case the vehicle's movement and sensing actions depend on a number of environment conditions that are impossible to model. As such, no realistic assumptions can be made about trajectory generation, path planning and collision avoidance. Therefore, sensor-based navigation controllers with *reactive* and *reflective* capabilities [4, 32] must be derived that generate control commands based on sensor data. Such controllers behave as *goal-based controllers* when no obstacles are considered or as *reaction-based controllers* when obstacle avoidance is necessary.

The *sensor fusion module* is responsible for position monitoring and obstacle detection. It receives data (inputs) from the AUV sonar sensor ring and through the linguistic variable *collision* it provides information about potential collisions in four main motion directions *front, back, left, right*. The *collision avoidance module* receives as inputs the calculated collision possibilities together with the vehicle heading pitch error from a target point (or the desired target, if known) and generates the new target point (or the next way point). The *motion control module* is responsible to control the vehicle's propellers, thrusters and fins in order to reach the goal point with the target surge velocity; it is composed of five subsystem controllers, the *speed control*, the *heading control*, the *depth control*, the *roll control* and the *ocean current control* subsystems.

It is emphasized that the *sensor fusion module* may receive data from several sets of sensors (not only sonar sensors) and may calculate collision possibilities in any number of primary and/or secondary directions. The *collision avoidance* module may also receive velocity, acceleration (if necessary) and angle (pitch, roll) data to adjust to new (desired) target point velocities, accelerations and angles. These two modules are totally independent of the vehicle type. The *motion control module* uses information from the (specific) vehicle to be tested, even though an analytic detailed model may not be required.

Considering that in the most general case the number of generated fuzzy rules (in the rule base) is $O(K^n)$, where K is the number of fuzzy predicates and n is the number of input variables [33], the simplicity of the proposed navigation architecture is justified by the small number of inputs in each module (as opposed to a single controller with several more inputs). The modularity is justified by the overall controller decomposition into a small number of modules (in this case, three). Its expandability is obvious given the fact that the sensor fusion module may be further decomposed into several sensor subsystems to account for several sets of sensors, each having ni inputs, $i = 1, 2, \ldots, N$, with corresponding complexities $O(K^{ni})$ and (upper bound of) total complexity of $O(K^{n1}) + O(K^{n2}) + \cdots + O(K^{nN})$, as opposed to a single module with complexity $O(K^{n1+n2+n3+\cdots+nN})$.

A rather extensive summary of existing ROV/AUV types and control architectures may be found in [6, 10, 35, 38]. Representative ROV and AUV adaptive and sliding mode control techniques are reported in [3, 2, 9, 11, 19, 24, 47]. Fuzzy logic based AUV navigation and control techniques are the topic of [13, 22, 30, 36, 40, 48, 44], while neuro-fuzzy based techniques may be found in [21, 41, 42]. Comprehensive studies related to underwater vehicle modeling, guidance and control are discussed in [46] and [18], while research on the *NPS Phoenix* AUV is reported in [7, 8, 22, 23, 25, 27, 26, 31]. Additional information may be found in [1, 12, 20, 23, 43].

The rest of the paper is organized as follows. Section 2 discusses the proposed navigation control architecture and corresponding controllers. Section 3 presents simulation results based on realistic scenarios using the *NPS Phoenix* vehicle. Section 4 concludes the paper. Appendices A and B summarize vehicle kinematics and dynamics and derive the *NPS Phoenix* modified equations of motion used in the presented simulation studies.

2. The Control Architecture

The overall control architecture configuration with the *sensor fusion module*, the *collision avoidance module* and the *motion control module* clearly identified is shown in Figure 1. Figure 2 shows the detailed structure of the *motion control* module that is composed of five individual control subsystems. The roll controller subsystem is separated from the rest since several existing vehicles are self-stabilized in roll motion, hence the roll controller may be inactive. Figures 1 and 2 do clarify the modularity and generality of the control architecture, illustrating at the same time that only the *motion control* module may be vehicle (model) dependent. Details for each module and their functionality follow.

Further, it is stated that all derived fuzzy controllers are of the Mamdani type.

2.1. THE SENSOR FUSION MODULE

The *sensor fusion* module is responsible for vehicle position monitoring and obstacle detection. Several types of sensors may be used and modeled, however, only a ring of sonar sensors is considered here (as in previously reported research [15, 16, 37, 39]). This assumption is reasonably valid because in cases where vi-

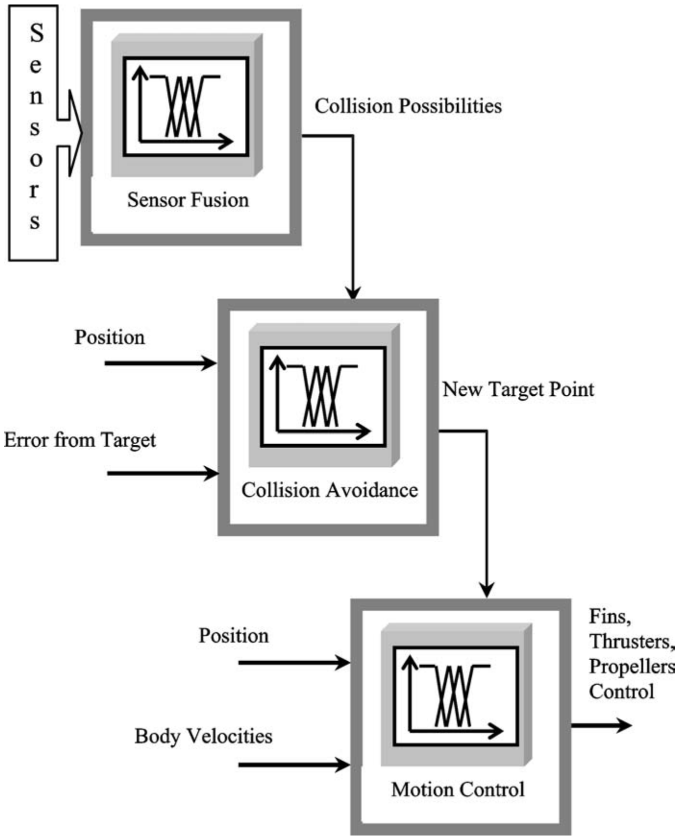


Figure 1. The overall navigation control architecture.

sion is poor (and given the fast attenuation of acoustic signals in an underwater environment), sonar sensors are widely used for navigation.

Without loss of generality, it is considered that a vehicle is equipped with a ring of 14 sonar sensors that cover 360° in the horizontal plane and a slice of 15° (effective area – pitch) in the vertical plane as shown in Figure 3. Table I shows the region covered by each sonar in terms of vehicle *head* and *pitch* angles. No distinction is made between side scan and forward sonar sensors; however, sonar effective distance within which they return reliable readings is very conservative to reflect reality.

Sonar sensors are located at the vehicle's center of gravity as shown in Figure 4, so that the proposed controller may be easily applied to all ROV and AUV types without explicit consideration of each vehicle's specific geometry and exact sonar positioning. This is a valid approximation given the vehicle's size, speed and the effective radius of marine sonar sensors. For a specific vehicle, when greater accuracy is essential (for navigation very close to obstacles, during docking from/to a greater marine vehicle), each sonar's exact position with respect to the vehicle's

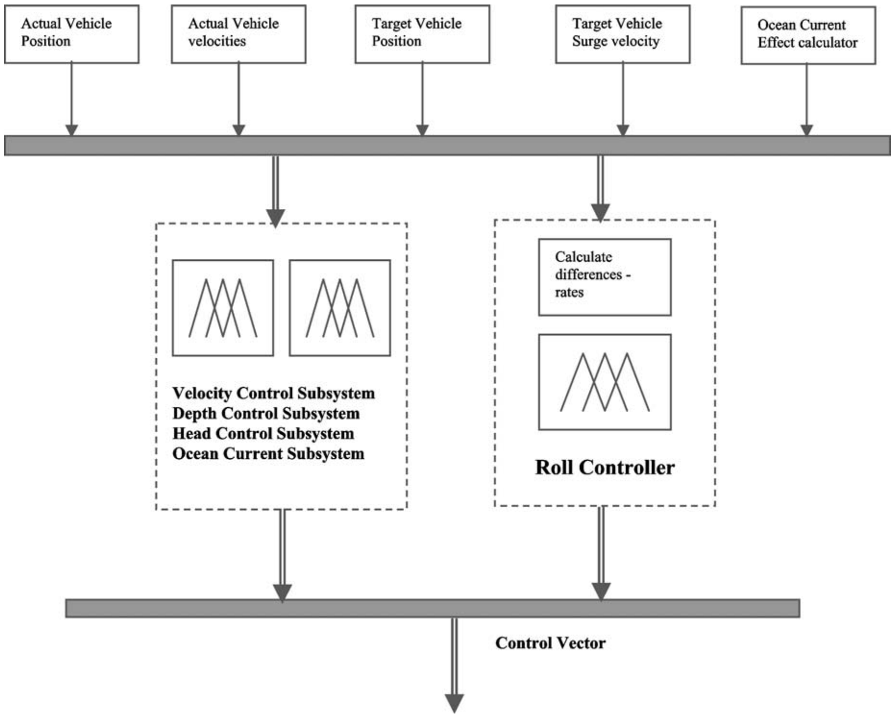


Figure 2. The motion controller configuration.

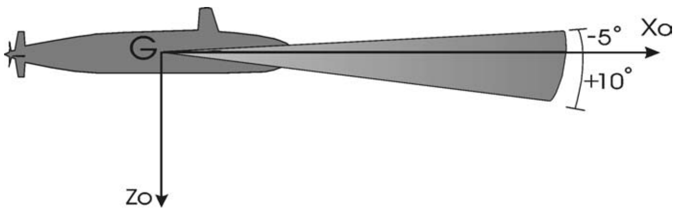


Figure 3. Sonar's effective area (pitch angle) in the vertical plane.

center of gravity should be taken into consideration and the readings be modified accordingly.

Given the nature of (a mostly slender body) vehicle motion in the water – it basically moves in the forward direction, movement in reverse and lateral directions is seldom – sonar sensors may be grouped closer to each other in the heading direction to provide accurate environment information in the vehicle's front area. The vehicle's turn radius is limited by its shape in cases of a nonzero *surge* velocity. The environment in front of the vehicle is of major importance for collision avoidance, as well as the fact that potential static or dynamic obstacles are mostly expected from "front and below" the vehicle (from the ocean floor). Thus, the sonar realistic effective *pitch* angle is defined to be between $[-10°, +5°]$ as shown in Figure 3 (assumption reached after reviewing existing sonar sensor specifications).

Table I. Sonar grouping in the horizontal plane

Sonar	Head (deg)		Pitch (deg)	
	From	To	From	To
1	–7.5	7.5	–5	10
2	–22.5	–7.5	–5	10
3	–45	–22.5	–5	10
4	–75	–45	–5	10
5	–105	–75	–5	10
6	–135	–105	–5	10
7	–165	–135	–5	10
8	–180	–165	–5	10
	165	180		
9	135	165	–5	10
10	105	135	–5	10
11	75	105	–5	10
12	45	75	–5	10
13	22.5	45	–5	10
14	7.5	22.5	–5	10

Sonar readings (data) are mapped into the linguistic variable "*distance*" with linguistic values *close, near* and *far*.⋆ The distance of the closest possible obstacle (reflected though sonar readings) is fed to a fuzzy inference engine that implements data readings to calculate collision possibilities in four cardinal directions *front*, *right*, *left* and *back*. The outputs are collision possibilities with linguistic variables *front_collision, right_collision, left_collision, back_collision* taking linguistic values *not possible, possible*, and *high*.⋆⋆

Since within each collision area correspond more than one sonar sensors, each rule has a relative *weight* indicating the sensor's contribution towards calculating the collision possibility in a given direction [15, 16, 37, 39], as also shown in Figure 4. For example, sonar 1 has importance weighted 1.0 in the *front_collision*, sonar 2 has importance weighted 0.8 in the *front_collision* and 0.5 in the *right_collision*, sonar 3 has importance weighted 0.2 in the *front_collision* and 0.8 in the *right_collision* [15, 16, 37, 39], etc.

Collision possibilities are calculated using 57 fuzzy rules of the type:

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

⋆ In case of no obstacle, the sonar's effective radius is used as the "obstacle-free" distance until the next reading.

⋆⋆ There is no limitation on the number of linguistic values. Three are sufficient in an underwater environment, while 5–7 have been considered in a mobile robot environment [15, 16, 37, 39].

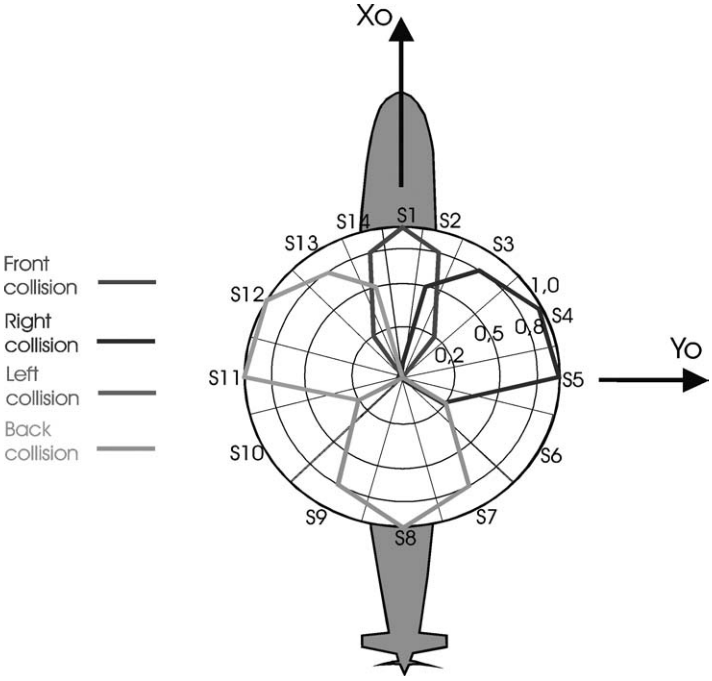


Figure 4. Sonar arrangement and relative importance.

where k is the rule number, d_i represents sensor i readings, $LD^{(k)}$ is the linguistic variable of the term set $D = \{close, near, far\}$, c_j is the collision direction and $LC^{(k)}$ the variable with term set $C = \{not\ possible, possible, high\}$.

The obstacle distance membership function is calculated by the *max–min* composition between the fuzzified readings and the previous fuzzy relation as:

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

2.2. THE COLLISION AVOIDANCE MODULE

The *collision avoidance* module is responsible for motion path changes when obstacles are within the vehicle path. Obstacle avoidance behavior occurs even when sonar sensors detect no obstacle within their measurement range (effective range), resulting in collision free navigation (until an obstacle is detected). In this case, the controller does not alter the vehicle target point and navigation is *goal directed*. When an obstacle is detected, the controller outputs a *head change* angle and a *pitch change* angle indicating the new vehicle direction to avoid the obstacle; in this case, navigation is *reaction directed*. Obviously, after obstacle avoidance, navigation behavior switches back to goal directed until a new obstacle is detected.

Goal directed navigation requires strategies (rules) that reduce *head* and *pitch* error to zero. *Reaction directed* navigation involves possible collision states and avoidance strategies. In extreme cases where the vehicle is closer than a predefined threshold distance to an obstacle, its speed is reduced to avoid collision and a *goal surge speed* is generated by the *collision avoidance* module controller.

The controller inputs are the calculated collision possibilities in the four cardinal directions, together with the vehicle heading and pitch error (from the desired target point). The input variables are:

(a) *collision possibilities* with linguistic values *not possible, 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 *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 rule base of the *collision avoidance* module consists of 56 rules of type:

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. The mathematical meaning of the kth rule is given as a fuzzy relation $R(k)$ on $C \times \Psi \times \Theta$, which in the membership function domain is

$$\mu_{R^{(k)}}(c_j, \psi, \theta) = \min\left[\mu_{LC^{(k)}}(c_j), \mu_{L\Psi^{(k)}}(\psi), \mu_{L\Theta^{(k)}}(\theta)\right]. \tag{2}$$

The whole rule base meaning may be described as the union of all individual rule meanings:

$$\mu_R(c_j, \psi, \theta, d\psi, d\theta, u) = \bigcup_{k=1}^{K} \mu_{R^{(k)}}(c_j, \psi, \theta). \tag{3}$$

The generic mathematical expression of the navigation output on $C \times \Psi \times \Theta$ is

$$\mu_N^*(d\psi, d\theta, u) = \max \min_{c_j,\psi,\theta}\left[\mu_{\mathrm{AND}}^*(c_j, \psi, \theta), \mu_R(c_j, \psi, \theta, d\psi, d\theta, u)\right] \tag{4}$$

where $\mu_{\mathrm{AND}}^*(c_j, \psi, \theta)$ is the combined input effect and $\mu_R(c_j, \psi, \theta, d\psi, d\theta, u)$ is the union of all individual rule meanings.

2.3. THE MOTION CONTROL MODULE

The *motion control* module, shown in Figure 5, controls the vehicle's propellers, thrusters and fins to reach a goal (or way) point with a *target surge* velocity. This is

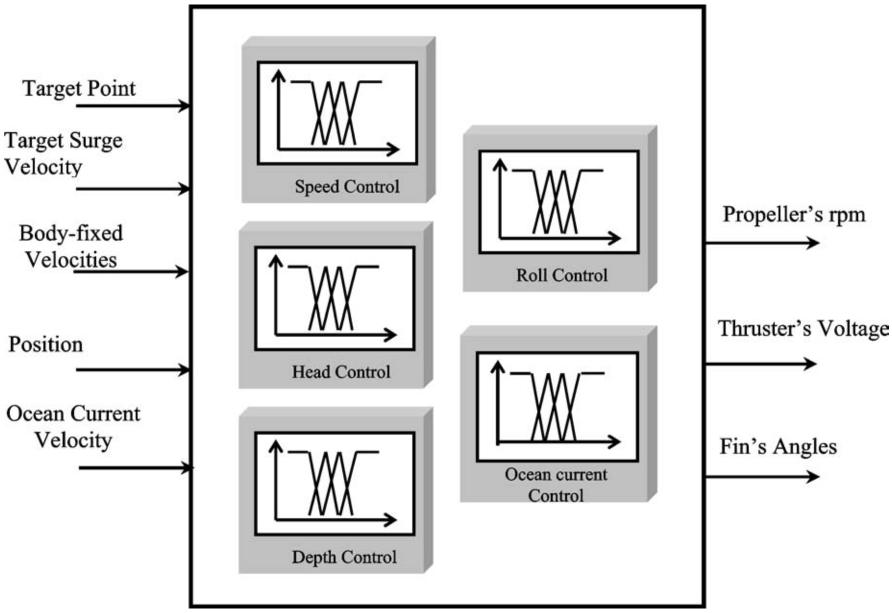


Figure 5. The motion control module configuration.

accomplished by regulating the voltage applied to the vehicle's motors and also the fin angle. Only minor modifications are necessary to apply the presented motion control module to any vehicle. As stated in [25] and [24], although equations of motion are coupled, steering, velocity and depth states are decoupled and lightly interacting resulting in a simpler controller structure.

The *motion control* inputs are the goal point (or the next way point), the vehicle actual position and orientation in earth-fixed coordinates, the *target surge* velocity, the actual vehicle velocities in body-fixed coordinates, and the *ocean current* velocity. Five subsystems control the vehicle:

- The *speed control* subsystem is responsible for the vehicle's speed by controlling the propellers revolution rate.
- The *heading control* subsystem controls steering in the horizontal plane by controlling the vehicle's head angle.
- The *depth control* subsystem controls the vehicle motion in the vertical plane by regulating the pitch angle and depth.
- The *roll motion control* subsystem controls the roll motion parameters, and,
- The *ocean current control* subsystem adjusts the vehicle position due to undersea currents. A sea current creates vehicle drift and deviation from a planned course. Although this deviation may be compensated through the speed and steering controllers, this subsystem controller is responsible for additional maneuverability by further adjusting steering control, overcoming lateral drag by modifying the desired head and pitch angle.

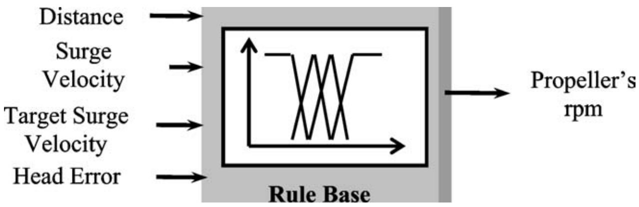


Figure 6. Speed control subsystem.

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. While describing each individual controller, although not explicitly stated, the effect of an ocean current is taken into consideration.

2.3.1. *Speed Control Subsystem*

Figure 6 shows the block diagram of the speed control subsystem. This controller calculates the distance from the target point. In the presence of an ocean current, a *modified surge velocity* is defined (by subtracting geometrically from the vehicle *surge velocity* the sea current velocity projection to the body-fixed x-axis) to account for the presence of ocean currents. The *modified surge velocity*, the *distance*, the *target surge velocity* and the *heading error* are the inputs to the fuzzy controller. The input linguistic variables and their values are: *distance*: {*zero*, *near*, *far*, *very-far*}, *surge_velocity*: {*slow*, *normal*, *fast*}, *target_surge_velocity*: {*slow*, *normal*, *fast*}, *heading_error*: {*negative*, *normal*, *positive*}. The linguistic values of the only output variable *propeller_rpm* are: {*fast-astern*, *slow-astern*, *zero, dead-slow-ahead*, *slow-ahead*, *fast-ahead*}.

The speed controller rule base consists of 22 rules of the form:

IF ⟨*distance* is *far*⟩ AND ⟨*surge_velocity* is *normal*⟩ AND
⟨*target_surge_velocity* is *normal*⟩ AND ⟨*heading_error* is *normal*⟩
THEN ⟨*propellers_rpm* is *slow-ahead*⟩.

The ordered propellers rpm (same for port and starboard propellers) are given by the max–min *composition* of the 22 rules, after defuzzification using the centroid method.

The value of *propeller_rpm* is *zero* if the vehicle's *heading_error* is not *normal*. In this situation the vehicle will first adjust its heading by the *heading control* subsystem (described below) and then it will move towards the target point.

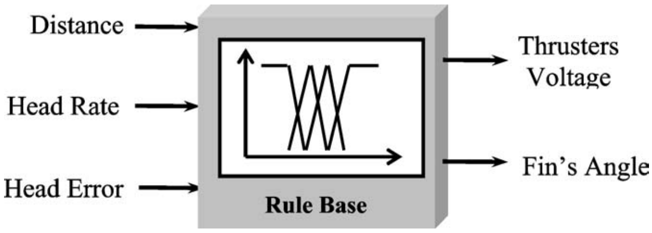


Figure 7. Head control subsystem.

2.3.2. *The Heading Control Subsystem*

Figure 7 shows the block diagram of the head control subsystem. The controller calculates the difference between the desired heading and the actual yaw angle ψ, called *head error angle* with values ranging between $[-180°, 180°]$. The ocean current effect is then accounted for by modifying the *head error angle* in such a way so that the vehicle will change its heading in the direction of the ocean current equalizing the current drift. The *modified head error angle*, the *head angle rate change*, and the *distance* from the target point are the three inputs to the fuzzy controller. Their linguistic values are: *head_error_angle*: {*big-negative*, *negative*, *zero*, *positive*, *big-positive*}, *head_rate*: {*negative*, *normal*, *positive*}, *distance*: {*zero*, *near*, *far*}.

The linguistic values of the output variables *lateral thruster voltage* and *ruder_fin_angle* are: {*big-negative*, *negative*, *zero*, *positive*, *big-positive*}. The rule base consists of 16 rules of the type:

IF ⟨*head_error_angle* is *positive*⟩ AND ⟨*head_rate* is *normal*⟩
AND ⟨*distance* is *far*⟩
THEN ⟨*lateral thruster voltage* is *positive*⟩
AND ⟨*ruder_fin_angle* is *positive*⟩.

The ordered bow and stern lateral thruster voltage, bow and stern rudder fin angle are equal and have opposite signs. The variable *distance* is taken into consideration to avoid abrupt changes in the vehicle's steering and heading angle as it comes closer to a target point with poor accuracy. This situation may occur when there is strong ocean current. Thus, the vehicle will be guided to the next target point and will not circle around the last point.

2.3.3. *The Depth Control Subsystem*

This subsystem consists of two controllers responsible for monitoring *pitch angle* and *depth control*, respectively, as shown in Figure 8. The two controllers are essential due to the fact that the Euler angle representation has two singularities at *pitch angle* $\theta = \pm 90°$; thus, by using an additional controller it is ensured that the vehicle will not operate close to those singularities.

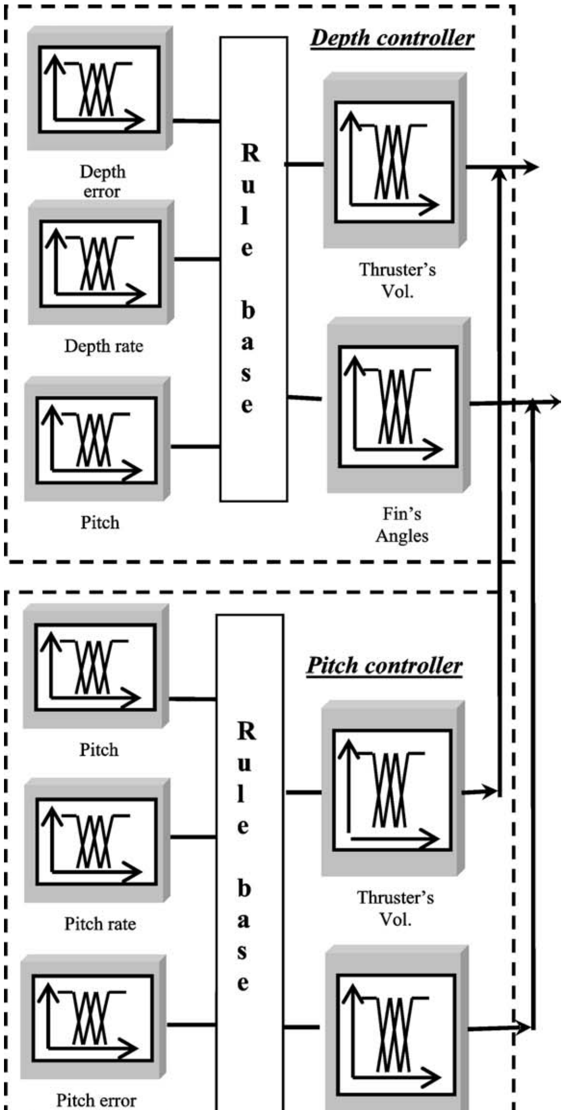


Figure 8. Depth control subsystem.

The *pitch controller* has inputs with respective linguistic values *pitch_error*: {*big-negative*, *small-negative*, *zero*, *small-positive*, *big-positive*}; *pitch_rate*: {*negative*, *normal*, *positive*}; *distance*: {*zero*, *near*, *far*}; *pitch*: {*out-of-limits-negative*, *normal*, *out-of-limits-positive*}. Outputs are: *vertical thruster voltage* and *plane_fin_angle* with linguistic values {*big-negative*, *negative*, *zero*, *positive*, *big-positive*}. The rule base consists of 18 rules of the form:

IF ⟨*pitch_error* is *small-positive*⟩ AND ⟨*pitch_rate* is *positive*⟩
AND ⟨*distance* is *far*⟩ AND ⟨*pitch* is *normal*⟩
THEN ⟨*vertical thruster voltage* is *big-positive*⟩
AND ⟨*plane_fin_angle* is *big-positive*⟩.

The variable *distance* functions in the same way as in the head controller. The variable *pitch* ensures that the actual vehicle pitch is not close to the singular values. In cases of large pitch angles θ, the vehicle changes its depth to the desired one by the depth controller. The ordered bow and stern vertical thruster voltage and bow and stern plane fin angles are equal and have opposite signs, to create a moment to change the pitch angle.

The input variables of the *depth controller* are: *depth_error*: {*negative*, *zero*, *positive*}, *depth_error_rate*: {*negative*, *normal*, *positive*} and *pitch*: {*out-of-limits-negative*, *normal*, *out-of-limits-positive*}. The output variable *vertical_thruster_voltage* takes the values from the set {*negative*, *zero*, *positive*} and the output variable *vertical_fin's_angle* takes the values from the set {*negative*, *zero*, *positive*}. The rule base has 11 rules. An example of a depth controller rule is

IF ⟨*depth_error* is *negative*⟩ AND ⟨*depth_error_rate* is *normal*⟩
AND ⟨*pitch* is *normal*⟩
THEN ⟨*vertical_thruster_voltage* is *negative*⟩
AND ⟨*vertical_fin's_angle* is *negative*⟩.

The vertical thruster's voltages and the vertical fin's angles from the depth controller are summed (with the same signs) to the outputs of the pitch controller. The sum is bounded within the servomotor limits (for example, 24 or 12 Volts for a thruster voltage).

2.3.4. *Roll Motion Subsystem*

Figure 9 shows the block diagram of the roll motion subsystem. This subsystem is examined separately, as it is optional since many ROV and AUV types are self-stabilized in roll motion. The controller inputs are the *roll angle* and the rate of the φ Euler angle (rate of change of rotation angle about the x-axis). The output is a fin angle α, which is added to the port plane fins (bow and stern) and subtracted from the bow and stern plane fins. The plane bow and stern plane fins have an angle difference, which equals 2α, and so a moment about the x-axis is created, used to overcome and control the vehicle roll motion.

The roll control subsystem consists of a fuzzy controller with two inputs and one output, with linguistic variables and values as follows: *roll_angle*: {*big-negative*,

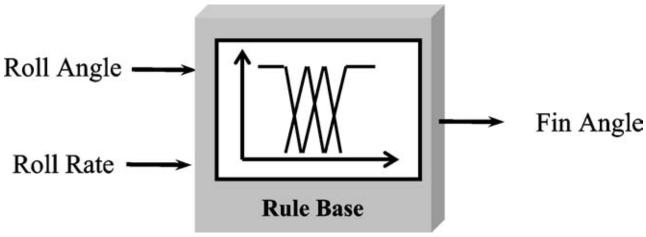


Figure 9. Roll control subsystem.

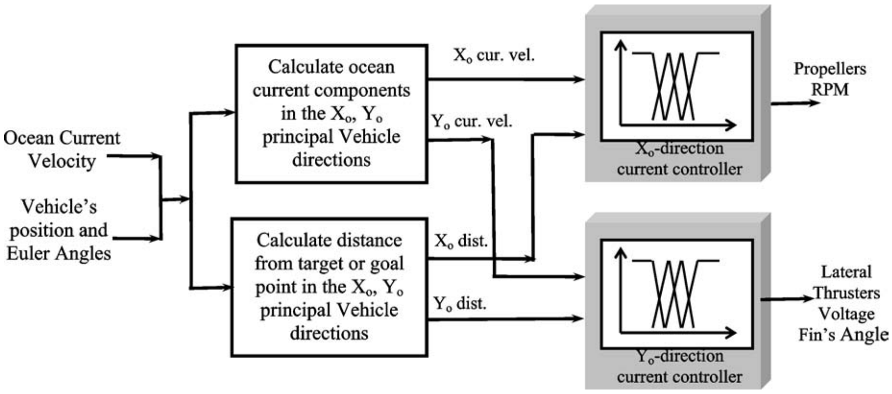


Figure 10. Ocean current control subsystem.

negative, *zero, positive, big-positive*}, *roll_angle_rate*: {*negative, zero, positive*}, *fin_angle*: {*big-negative*, *negative*, *zero, positive, big-positive*}. The rule base consists of 8 rules of the type:

IF ⟨*roll_angle* is *negative*⟩ AND ⟨*roll_angle_rate* is *negative*⟩
THEN ⟨*fin_angle*⟩ is ⟨*big* − *negative*⟩.

2.3.5. *Ocean Current Subsystem*

This controller, shown in Figure 10, is designed to help the vehicle overcome drag produced by a lateral ocean current with known mean velocity in the horizontal X–Y plane. It consists of two fuzzy controllers (one for each X_o and Y_o direction). The controller inputs are the vehicle position, Euler angles, the target or goal point position and the ocean current velocity. The projections to the X_o and Y_o vehicle axes of the distance from the goal point and the ocean current velocity are calculated. The X_o projection of the current is expected to affect the forward motion of the vehicle and slow or accelerate the vehicle, while the Y_o projection of the current is expected to have a side drag effect. The output of the X_o-direction current controller results in forward motion propellers rpm, and the output of the Y_o-direction current controller are lateral motion thrusters voltages and fins angles.

The X_o-direction current controller deals with the longitudinal current velocity that is the projection of the current velocity to the body-fixed X_o-axis, and the delta-x that is the projection of the distance arrow on the X_o body-fixed axis. Its aim is to accelerate or slow down the vehicle by modifying the ordered revolution rate of its propellers. The output variable is the *propellers_rpm* that is added to all propellers of the vehicle that control the forward motion. Linguistic values for the input variables x_o*_current_velocity* and *delta_x* are {*negative, zero, positive*}, while the values of the output variable *propellers_rpm* are: {*big-negative*, *negative*, *zero*, *positive*, *big-positive*}. A typical rule (out of 9 rules) for the ocean current controller is

IF ⟨x_o*_current_velocity* is *zero*⟩ AND ⟨*delta_x* is *zero*⟩
THEN ⟨*propellers_rpm* is *zero*⟩.

The Y_o-direction current controller deals with the lateral current velocity that is the projection of the current velocity to the body-fixed Y_o-axis, and the delta-y, that is the projection of the distance arrow on the Y_o body-fixed axis. Its aim is to overcome the lateral drag by adding voltage to the lateral thrusters and adding fins angles to lateral fins. Consequently, the output variable is the *lateral_thruster_voltage* that is added to bow and stern lateral thrusters. Linguistic values for the input variables y_o*_current_velocity* and *delta_y* are {*negative, zero, positive*}, while the values of the output variables are: *lateral_thruster_voltage* {*big-negative*, *negative*, *zero*, *positive*, *big-positive*} and *lateral_fin's_angle* {*big-negative*, *negative*, *zero*, *positive*, *big-positive*}. A typical rule (out of 9 rules) for the ocean current controller is

IF ⟨y_o*_current_velocity* is *zero*⟩ AND ⟨*delta_y* is *zero*⟩
THEN ⟨*lateral_thruster_voltage* is *zero*⟩
AND ⟨*lateral_fin's_angle* is *negative*⟩.

Simulation results are presented next.

3. Simulation Results

The proposed navigation control architecture has been applied and tested on the *NPS Phoenix* shown in Figure 11. The vehicle's dimensions, specifications, hydrodynamic model and coefficients have been accurately defined in [7, 8]. The *NPS Phoenix* is neutrally buoyant with a design depth of 20 ft (6.1 m), endurance of 90–120 min supported by a pair of lead-acid gel batteries at speeds up to 2 ft/sec (0.61 m/sec). It has a slender body with a hull length of 7.3 ft, four paired plane surfaces (eight fins total) and four paired thrusters built in cross-body tunnels. It has two screw bidirectional propellers and the hull is made of press and welded aluminum.

The *NPS Phoenix* is considered to be self stabilized in roll as explained in [8] – the bow and stern plane fins and rudder fins have the same angle values. Therefore,

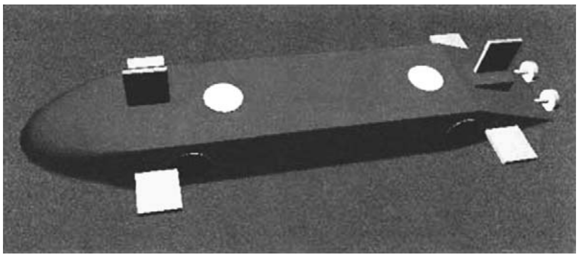


Figure 11. The *NPS Phoenix.*

to produce roll moments and control the roll motion, the equations derived in [8] have been modified by considering different angles for each of the eight control surfaces. The *control vector* (formed by the motion control module) is a 14×1 vector calculated during each simulation time step. Appendix B presents the complete set of modified equations and 14 vehicle control actions, derived to account for roll motion control.

The effect of Reynolds number on the hydrodynamic coefficients has not been considered and all hydrodynamic derivatives and coefficients are constant and not varying with the vehicle's velocity. The added mass matrix is symmetrical (valid assumption since the vehicle moves at very low speed) and the off-diagonal elements are much smaller than their diagonal counterparts as justified in [18]. As explicitly derived and justified in [29], the *mass inertia added mass* matrix used in simulation studies (the vehicle is symmetrical in the $x_0 - z_0$ plane, Appendix A) is:

$$[M] = \begin{bmatrix} 14.6124 & 0 & 0 & 0 & 2.4066 & 0 \\ 0 & 26.8075 & 0 & -2.4066 & 0 & 0.2704 \\ 0 & 0 & 50.0509 & 0 & 6.8863 & 0 \\ 0 & -1.2033 & 0 & 7.6573 & 0 & 0 \\ 1.2033 & 0 & 7.0215 & 0 & 171.0958 & 0 \\ 0 & 0.1352 & 0 & 0 & 0 & 54.7080 \end{bmatrix}.$$

Actuator dynamics are not modeled and the ordered control actions are fed directly into the dynamics model during simulation. The vehicle dynamics model and controllers have been programmed in *Matlab*® and *Simulink*®. Simulation is discrete with fixed steps of 0.25 sec. The Euler angles representation has been utilized since the vehicle is not commanded to operate close to the singularity of the *pitch* angle at $\pm 90°$.

Simulation tests study vehicle functionality under the presence of a horizontal ocean current and in the presence of obstacles (ocean floor configuration) found in the vehicle path (that need be avoided). Therefore, results reflect: (i) navigation accuracy tests, and, (ii) collision-free navigation tests. The overall simulation procedure is illustrated in Figure 12.

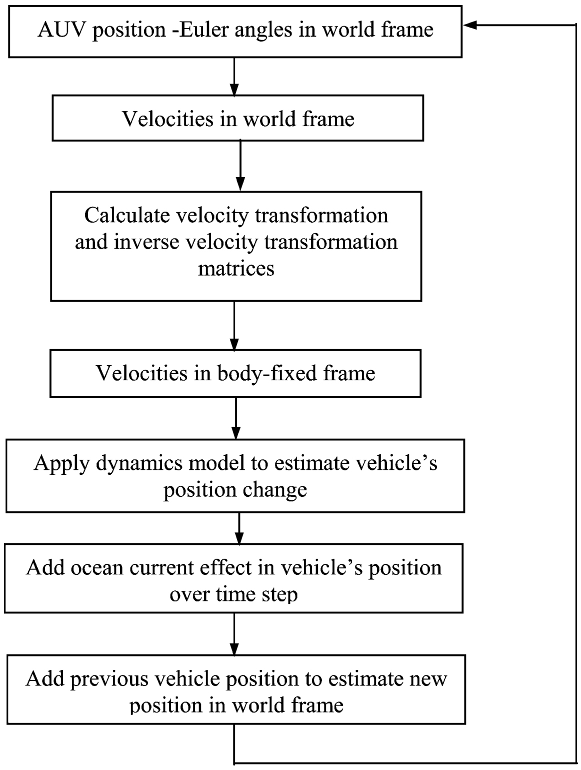


Figure 12. The simulation procedure.

The ocean floor has been represented by a meshed 3-D surface, produced using mathematical functions of the form:

$$\begin{aligned} z(x, y) &= \sin(y + a) + b \sin(x) + c \cos\left(d\sqrt{y^2 + x^2}\right) \\ &\quad + e \cos(y) + f \sin\left(f\sqrt{y^2 + x^2}\right) + g \cos(y), \end{aligned}$$

where a, b, c, d, e, f, g are constants experimentally defined, in order to produce a surface simulating an ocean floor.

Effective sonar radius within which sonars return reliable readings is assumed to be 60 ft (worst case scenario). During each simulation step, the vehicle position and the Euler angles are fed to a subroutine along with the effective sonar radius. A rectangular grid is generated with center the vehicle position and size twice the sonar effective radius (120 ft). The grid discretization step is 5 ft, chosen to keep computational complexity manageable. For each grid (node), the value of the ocean floor z component is calculated, and the generated (x, y, z) represent the geometry of the ocean floor in the vehicle territory. For each (x, y, z) that represents an ocean floor point, the distance between the vehicle and that point is calculated. If this distance exceeds the effective sonar radius, the reading is neglected. If the distance between the ocean floor point and the vehicle is found to be smaller than

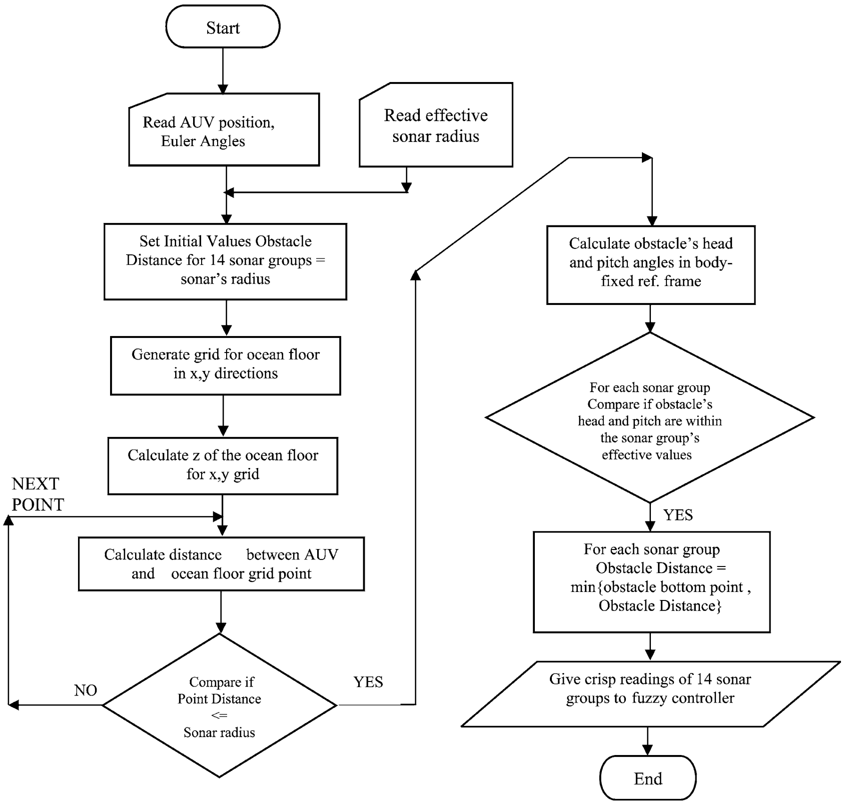


Figure 13. Subroutine to simulate sonar group readings in each simulation circle.

the effective sonar radius, then the reading is considered as a possible obstacle if it is within the sonar effective area in the vertical plane and has a *pitch* angle of $-5°$ to $+10°$. The direction of the point in response to the vehicle reference frame is calculated and the *pitch* angle of the point in response to the body reference frame is also calculated. These angles are compared with the effective angles of the sonar groups given in Table I. The sonar group reflecting the reading for the point is identified. For this group the distance of the obstacle is taken to be the minimum between the distance of the point and the distance calculated in previous steps for other ocean floor points. Thus by examining all the ocean floor point of the mesh, the function returns for each sonar group, the minimum distance in which this group will sensor an obstacle. If no obstacles are found, the effective sonar radius will be returned as an obstacle distance. The logical diagram of the above subroutine is shown in Figure 13.

3.1. NAVIGATION ACCURACY TESTS

Two test cases are presented. In the first test case, the vehicle follows a rectangle saw-tooth curve in the horizontal plane and gradually descents and ascents in the

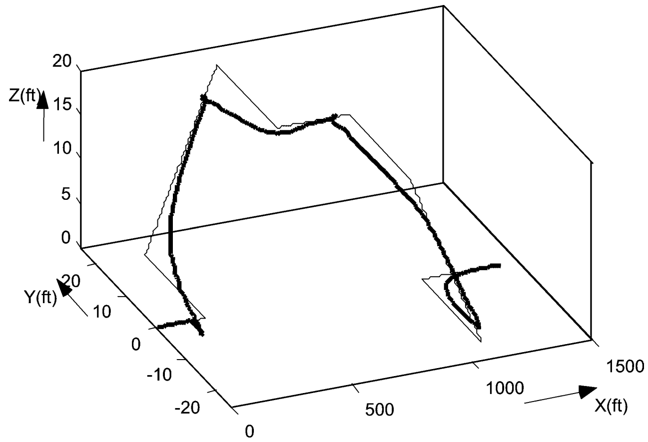


Figure 14. Test case 1: ordered and simulated path.

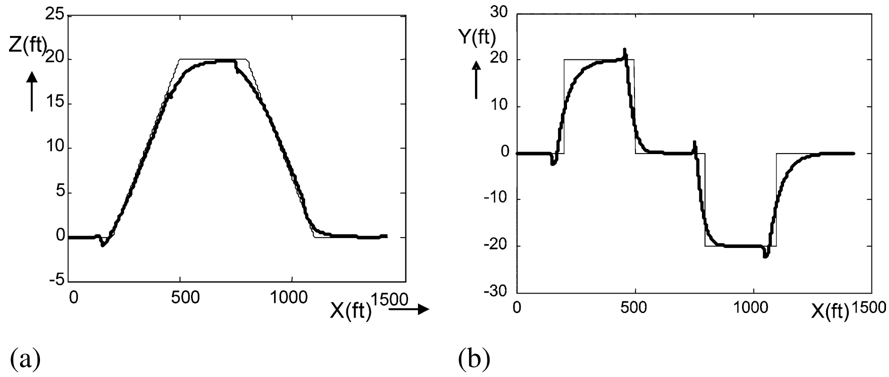
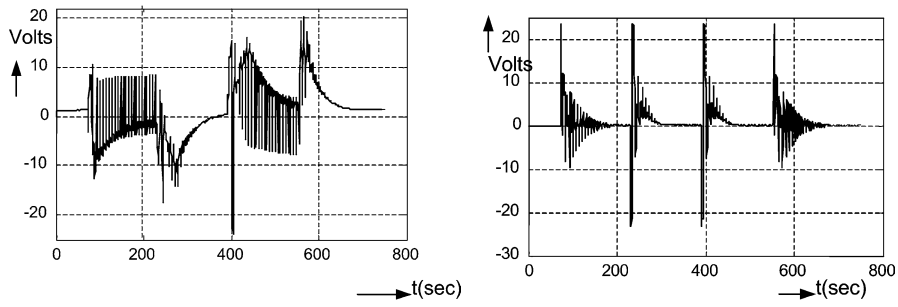


Figure 15. Test case 1: ordered and simulated path in (a) the vertical and (b) the horizontal plane.

vertical plane. No ocean current is assumed. The overall motion is shown in Figures 14–16, including the decoupled motion in the X–Y and X–Z planes. Figure 15 shows that the vehicle's dynamics are not completely decoupled. A change in the vehicle's steering control results in changes on depth control and vise versa. Peaks in the actual path in the X–Y plane given a command for a step change in Y occur because the controller sees a big head error angle, the propellers are shut down by the velocity controller and the head angle is modified to the desired one by the thrusters and fins as shown in Figure 16(b). This action drifts the vehicle to the opposite Y-direction and causes the head controller outputs to oscillate from positive to negative values.

In the second test case, the vehicle follows the same curve but in the presence of a horizontal ocean current (coordinates in X and Y axes). Figure 17 demonstrates the vehicle's deviation from the desired trajectory for ocean currents with earth-fixed velocities of 0.3, 0.6, 0.8 1.0 and 1.2 ft/sec, respectively. Figures 18–20



(a) (b)

Figure 16. Test case 1: (a) vertical bow thruster voltage and (b) lateral bow thruster voltage versus time.

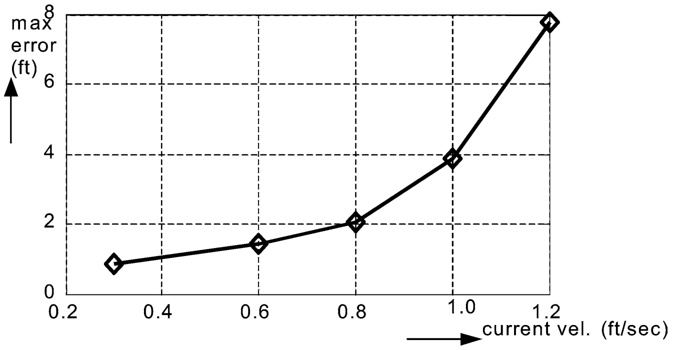


Figure 17. Test case 2: maximum deviation from ordered path versus lateral current velocity.

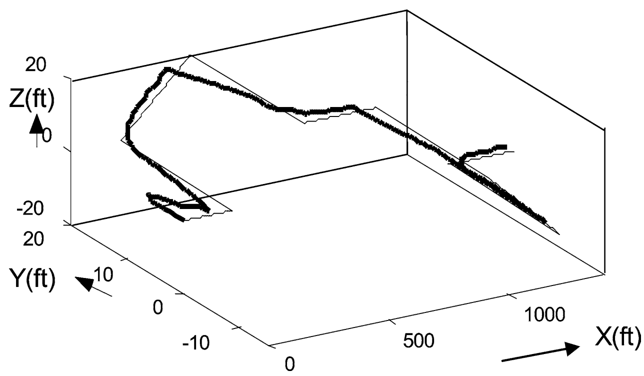
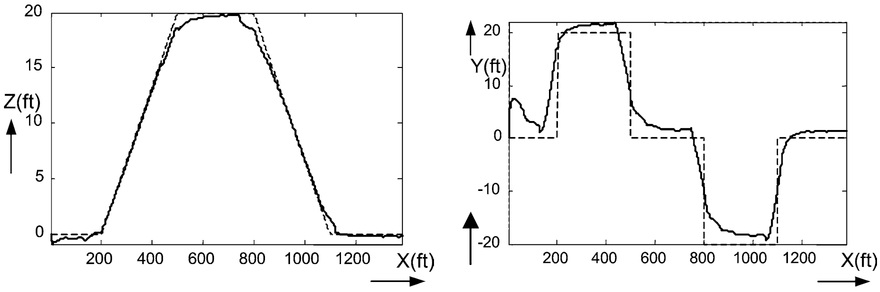


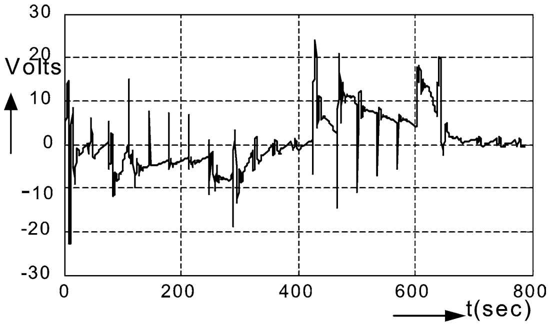
Figure 18. Test case 2: ordered and actual path with lateral Y current velocity 0.8 ft/sec.

illustrate the test case for lateral sea current velocity 0.8 ft/sec. Figure 19 shows that the vehicle reaches the desired trajectory in the vertical plane, while constantly overshoots in the horizontal plane pushed by the lateral current. In Figure 20, the vertical bow thruster and the lateral bow thruster voltage versus time are given for lateral current velocity 0.8 ft/sec.

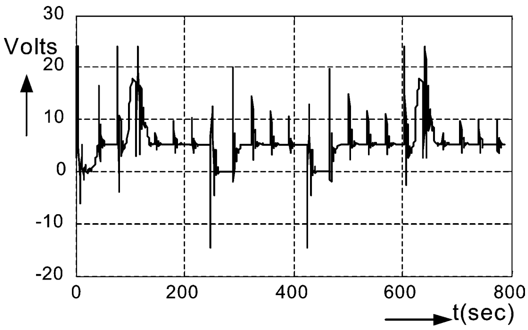


(a) (b)

Figure 19. Test case 2: ordered and actual decomposed path with lateral *Y* current velocity of 0.8 ft/sec.



(a)



(b)

Figure 20. Test case 2: (a) vertical bow thruster and (b) lateral bow thruster voltage versus time with lateral *Y* current velocity of 0.8 ft/sec.

3.2. COLLISION-FREE NAVIGATION TESTS

In order to test the vehicle movement in an ocean environment, the ocean floor is represented through the previously defined mathematical expression; the desired vehicle path is fed directly to the controller and the vehicle is commanded to follow

Table II. Start and end points for test cases 3, 4 and 5

	Starting point	Ending point	Ocean current velocity
Case 3	$X = 0$ ft	$X = 1500$ ft	$V_x = 0.0$ ft/sec
	$Y = 450$ ft	$Y = 1500$ ft	$V_Y = 0.0$ ft/sec
	$Z = 50$ ft	$Z = 60$ ft	
Case 4	$X = -250$ ft	$X = 1400$ ft	$V_x = 0.0$ ft/sec
	$Y = -250$ ft	$Y = 1100$ ft	$V_Y = 0.0$ ft/sec
	$Z = 80$ ft	$Z = 80$ ft	
Case 5	$X = -250$ ft	$X = 1400$ ft	$V_x = -0.31655$ ft/sec
	$Y = -250$ ft	$Y = 1100$ ft	$V_Y = 0.38703$ ft/sec
	$Z = 80$ ft	$Z = 80$ ft	$V = 0.50$ ft/sec

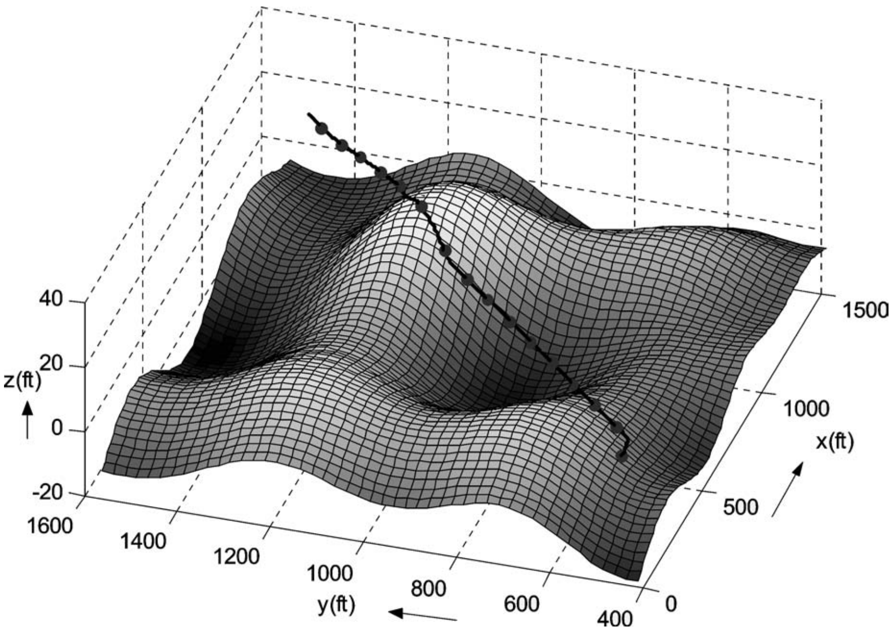
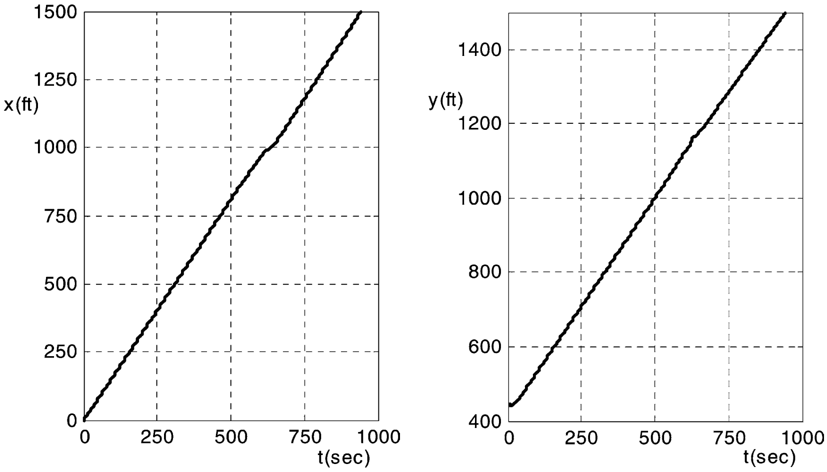


Figure 21. Test case 3: simulated vehicle path.

it. The vehicle identifies static obstacles (ocean floor configuration) as it reaches its target, avoiding collision with them. Three simulated test cases are presented for two different ocean floor topographies (one for case 3, one for cases 4 and 5) where the vehicle is commanded to pass through ocean floor hills from a starting point to a goal target point. The starting and ending goal points and the ocean current velocity for the three test cases are given in Table II. The ocean floor topography is the same for cases 4 and 5 for comparison purposes.



(a) (b)

Figure 22. Test case 3: simulated vehicle path versus time.

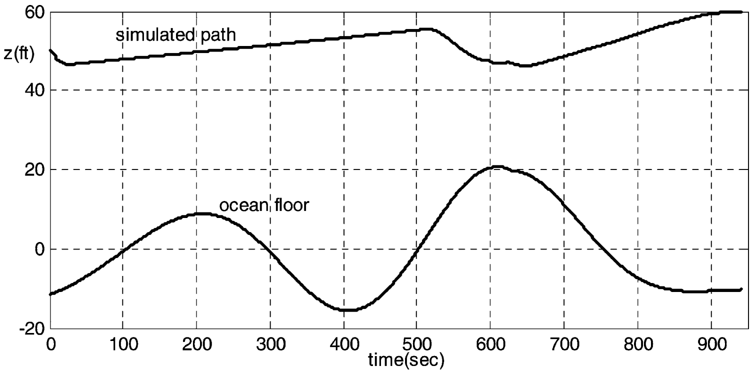
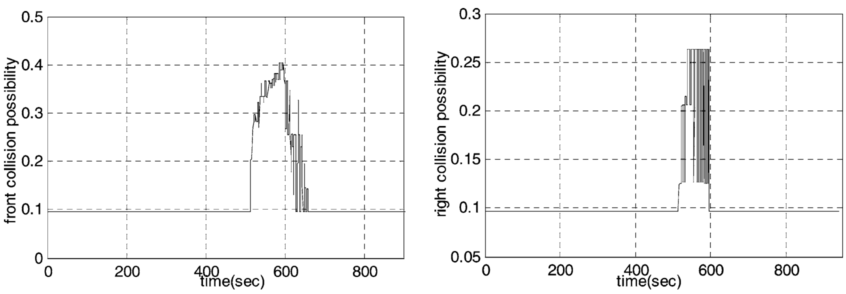
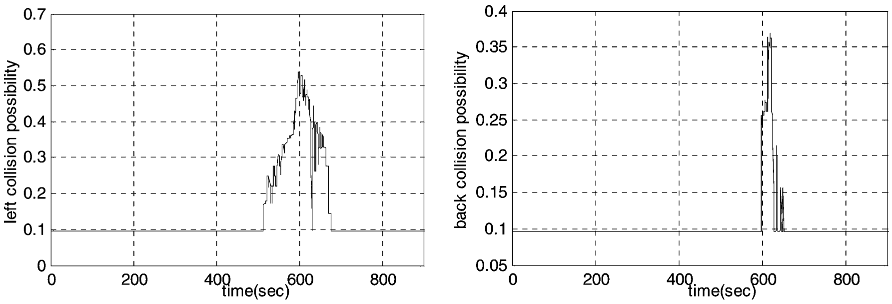


Figure 23. Test case 3: simulated path in $z(t)$ and ocean floor elevation through the vehicle's path.



(a) (b)

Figure 24. Test case 3: (a) front collision possibility, (b) right collision possibility versus time.



(a) (b)

Figure 25. Test case 3: (a) left collision possibility, (b) back collision possibility versus time.

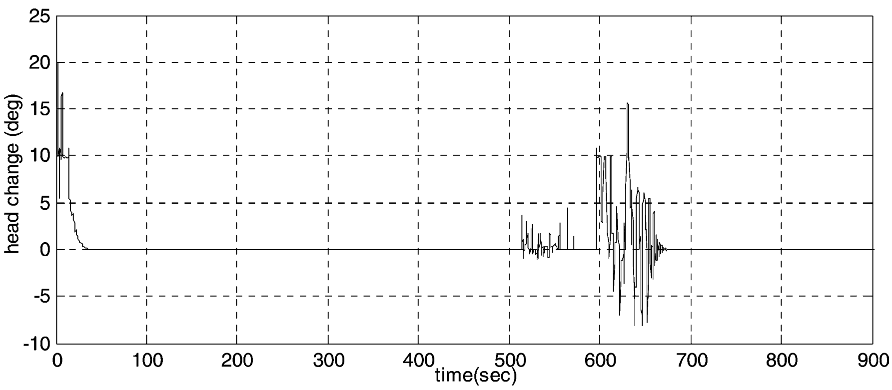


Figure 26. Test case 3: head change versus time.

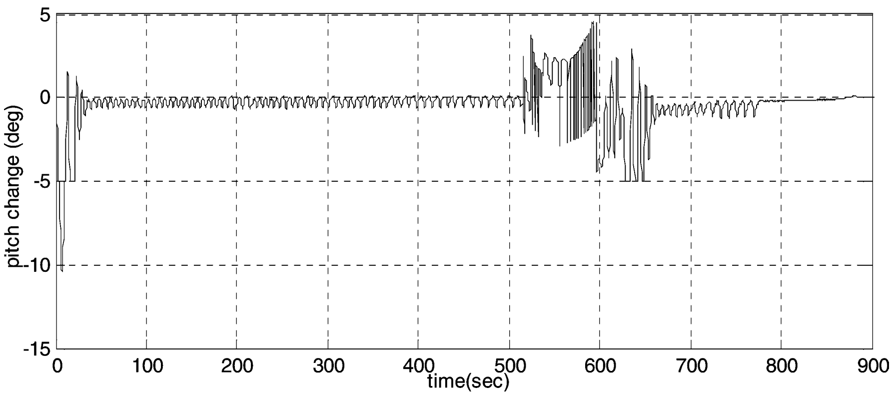


Figure 27. Test case 3: pitch change versus time.

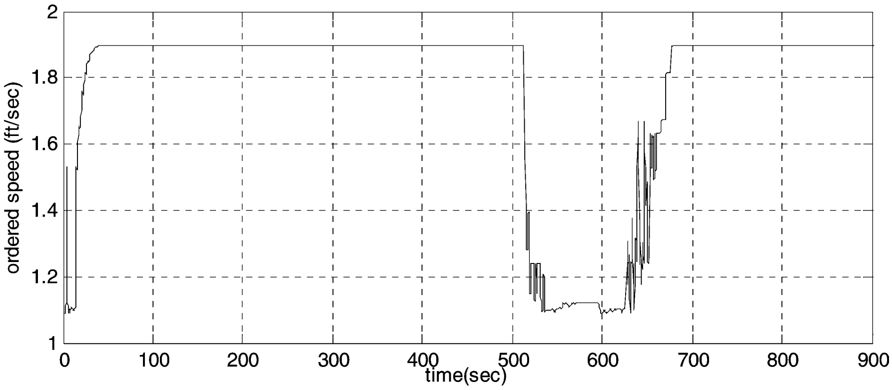


Figure 28. Test case 3: ordered speed versus time.

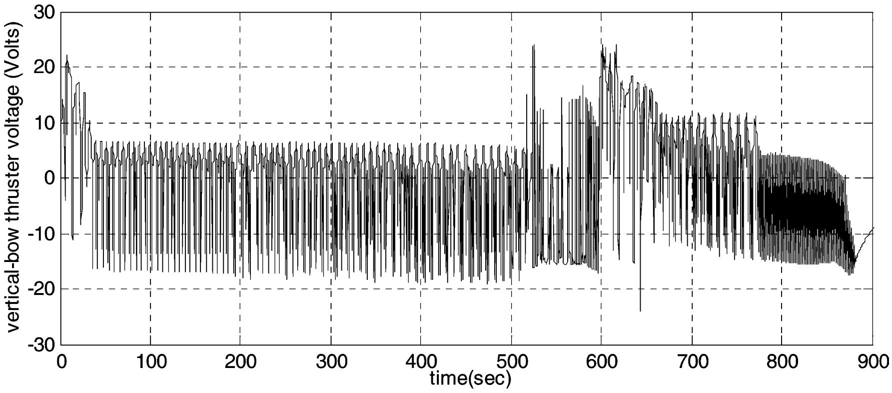


Figure 29. Test case 3: vertical-bow thruster voltage versus time.

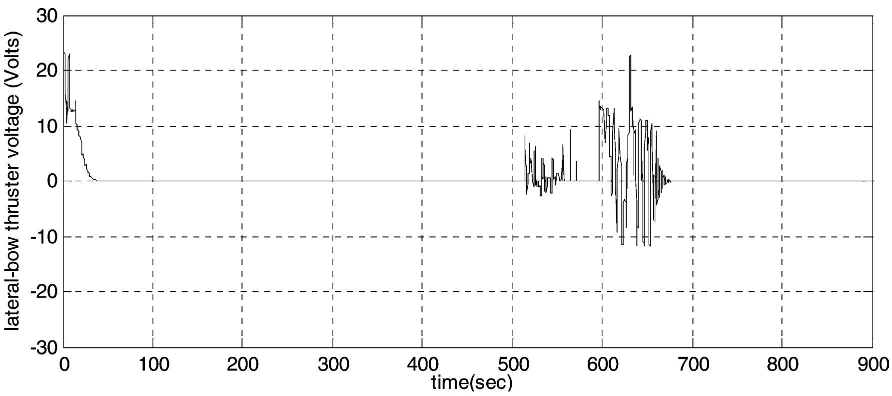


Figure 30. Test case 3: lateral-bow thruster voltage versus time.

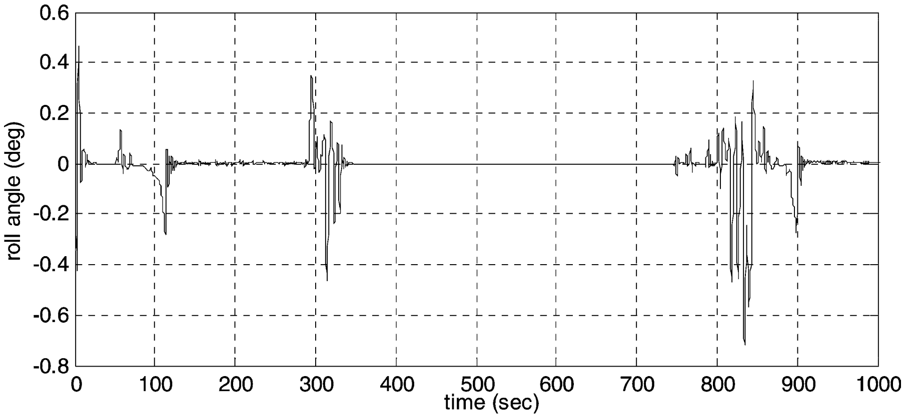


Figure 31. Test case 3: roll angle versus time.

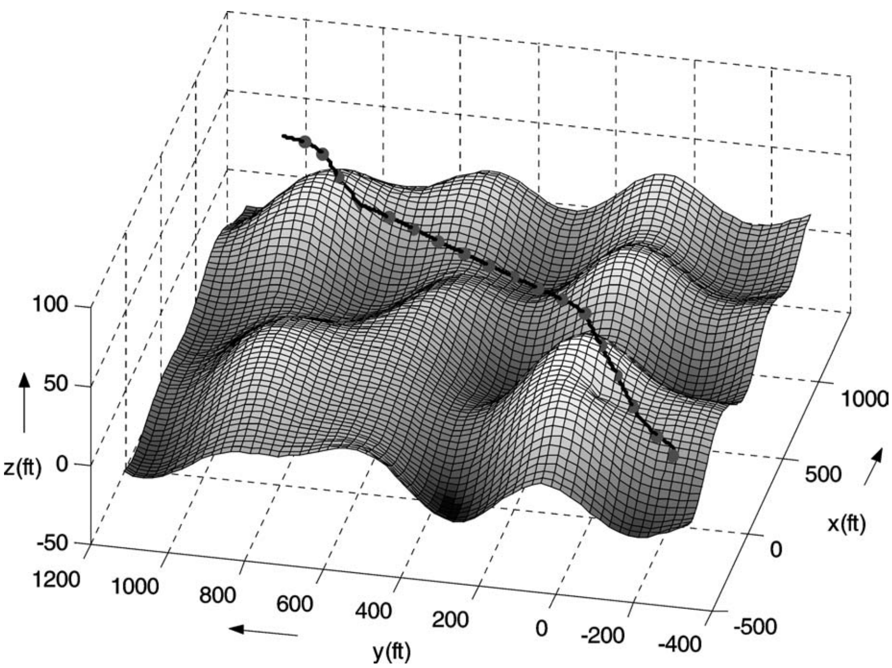
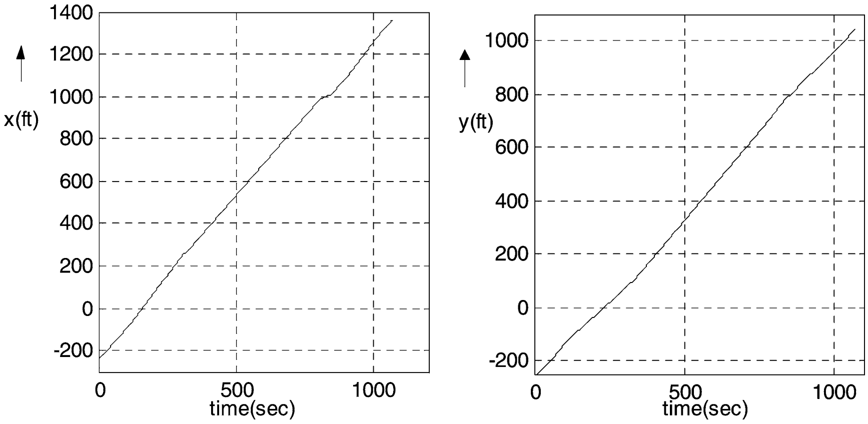


Figure 32. Test case 4: simulated vehicle path.

Figures 21–55 present results for each of the three test cases. For each case, results include the vehicle's path with respect to the ocean floor, the vehicle elevation in the z-axis with respect to the elevation of the ocean floor in the x–y plane; the vehicle's path in the presence of ocean current (if present) compared to the same path with no ocean current present; the front collision, right, left and back collision



(a) (b)

Figure 33. Test case 4: simulated vehicle path versus time.

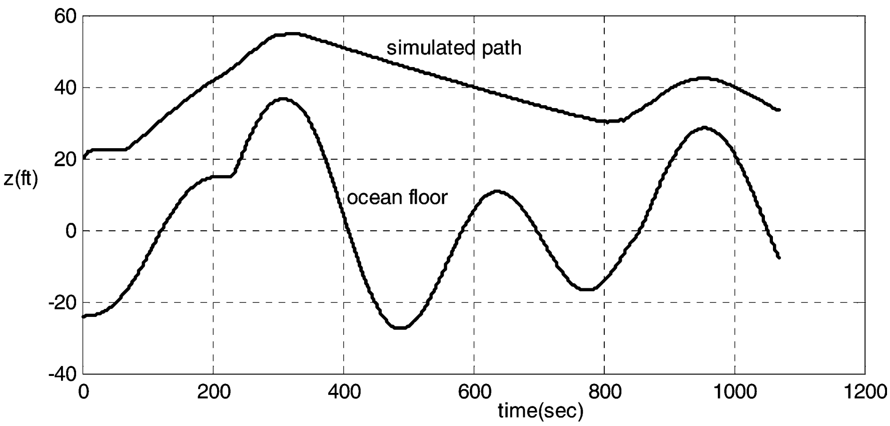
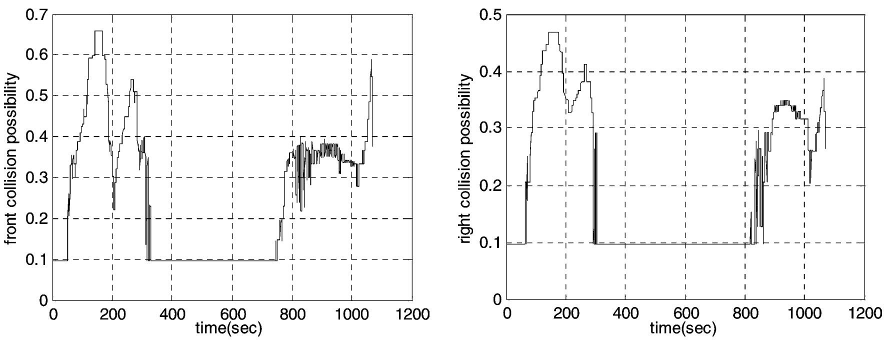
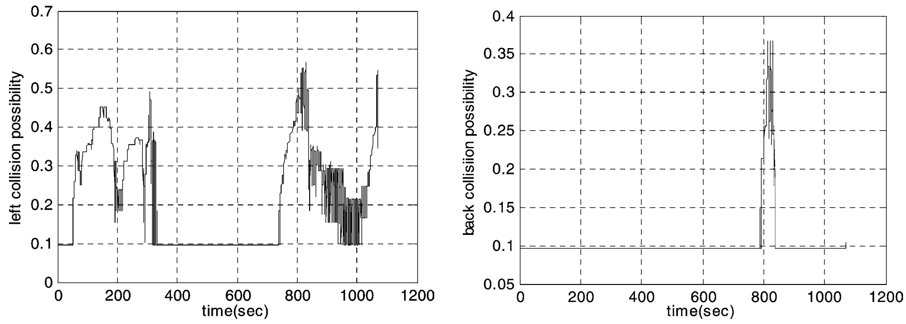


Figure 34. Test case 4: simulated path in $z(t)$ and ocean floor elevation through the vehicle's path.



(a) (b)

Figure 35. Test case 4: (a) front collision possibility, (b) right collision possibility.



(a) (b)

Figure 36. Test case 4: (a) left collision possibility, (b) back collision possibility.

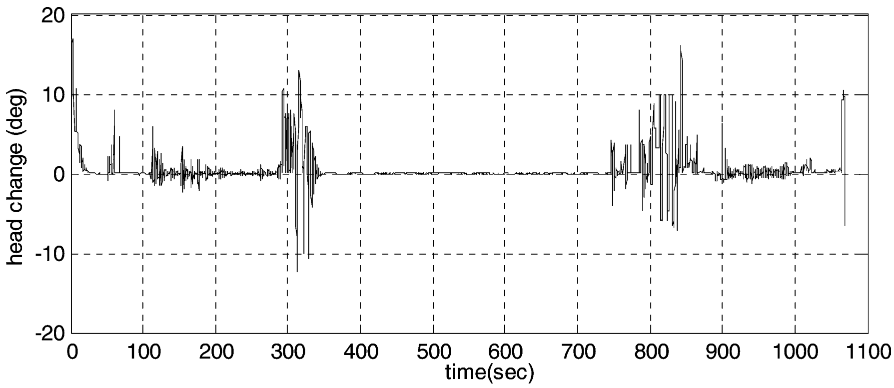


Figure 37. Test case 4: head change versus time.

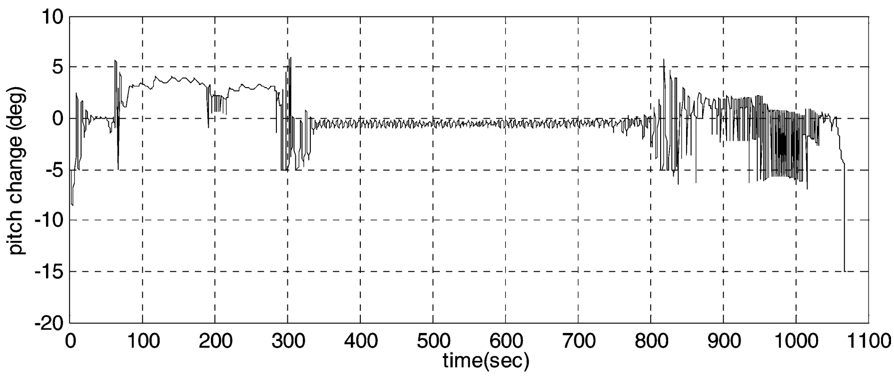


Figure 38. Test case 4: pitch change versus time.

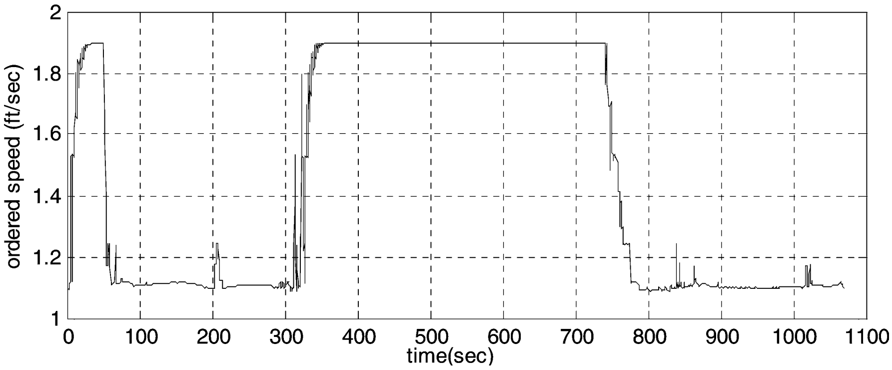


Figure 39. Test case 4: ordered speed versus time.

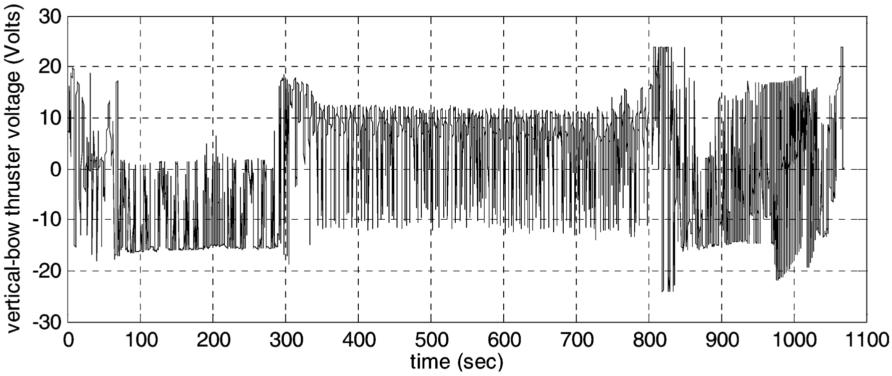


Figure 40. Test case 4: vertical-bow thruster voltage versus time.

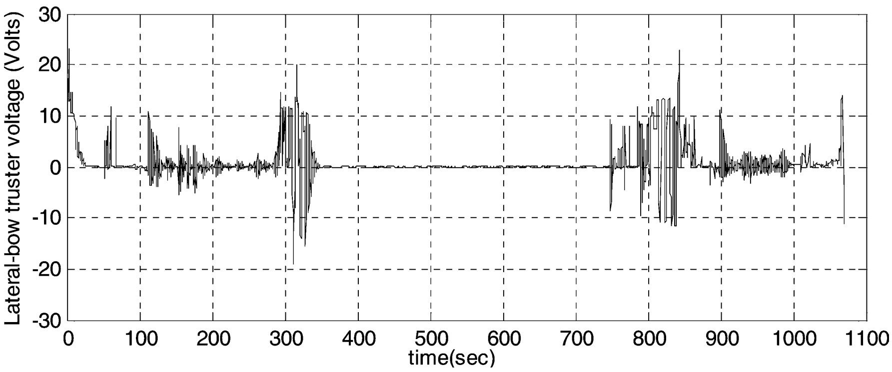


Figure 41. Test case 4: lateral-bow thruster voltage versus time.

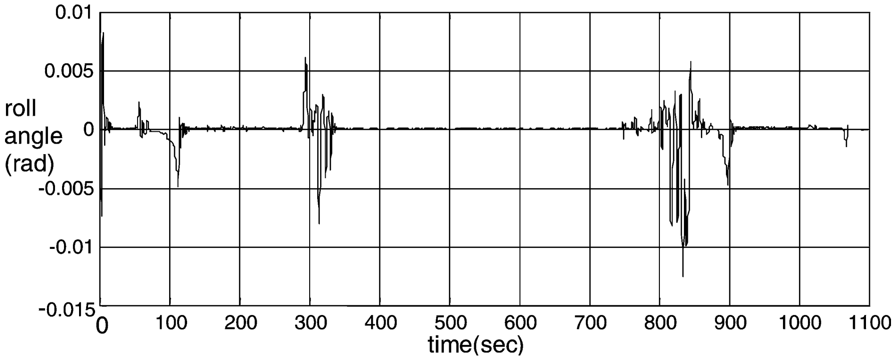


Figure 42. Test case 4: roll angle versus time.

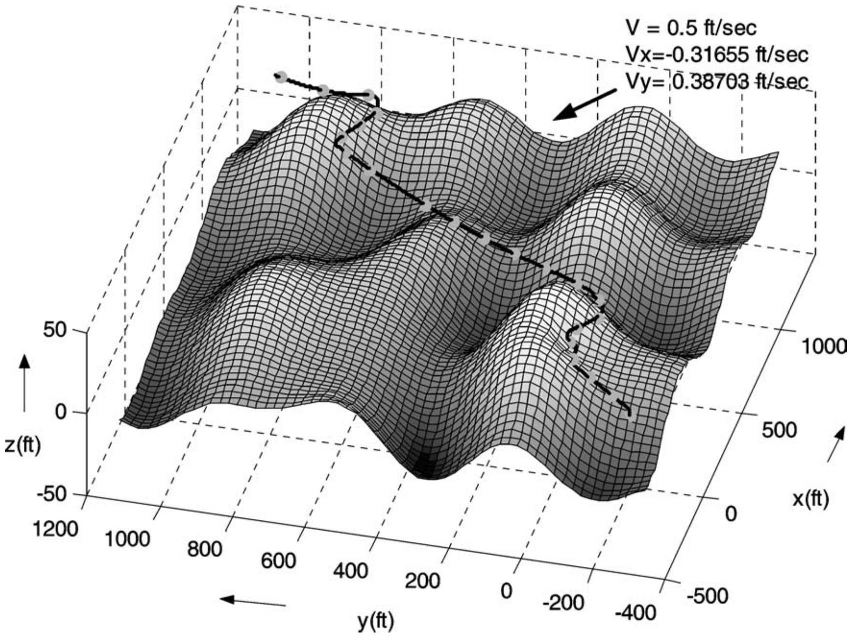


Figure 43. Test case 5: simulated vehicle path.

possibilities; head change, pitch change and ordered surge speed; ordered thruster's voltage and roll control.

From the above presented results it may be concluded that the vehicle is capable of avoiding obstacles while navigating autonomously in an ocean environment even under the presence of ocean currents and with no a-priori knowledge of the environment. The vehicle navigates through the ocean floor as it moves towards its final target point maintaining a safety distance from it where no collision is likely to happen.

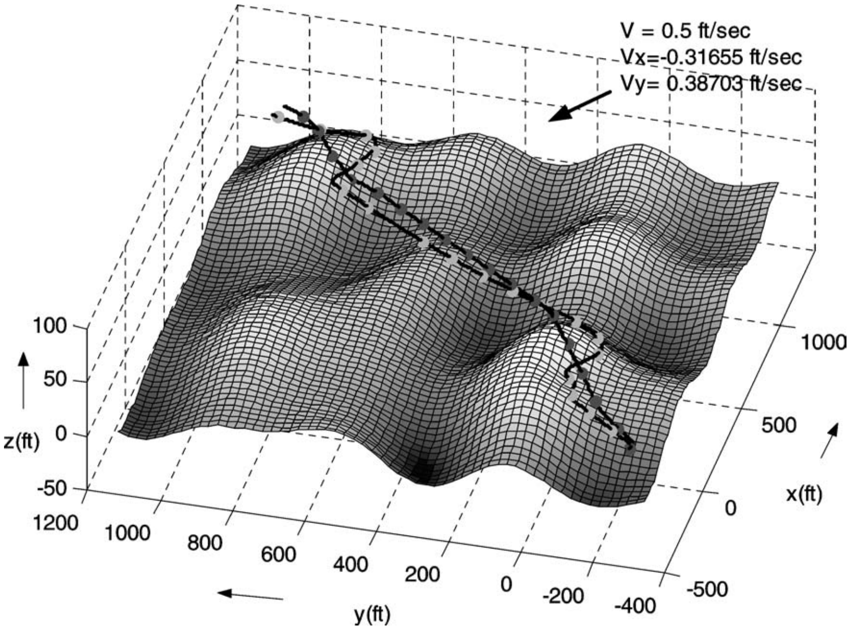


Figure 44. Test case 5: simulated vehicle path; blue dot path: with the presence of 0.5 ft/sec lateral ocean current; red dot path: with no ocean current present.

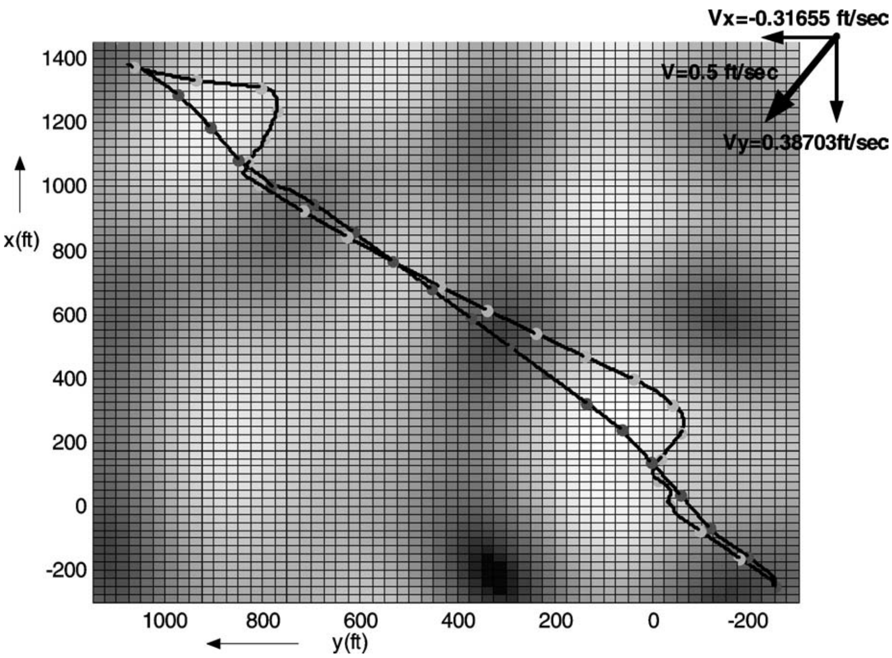
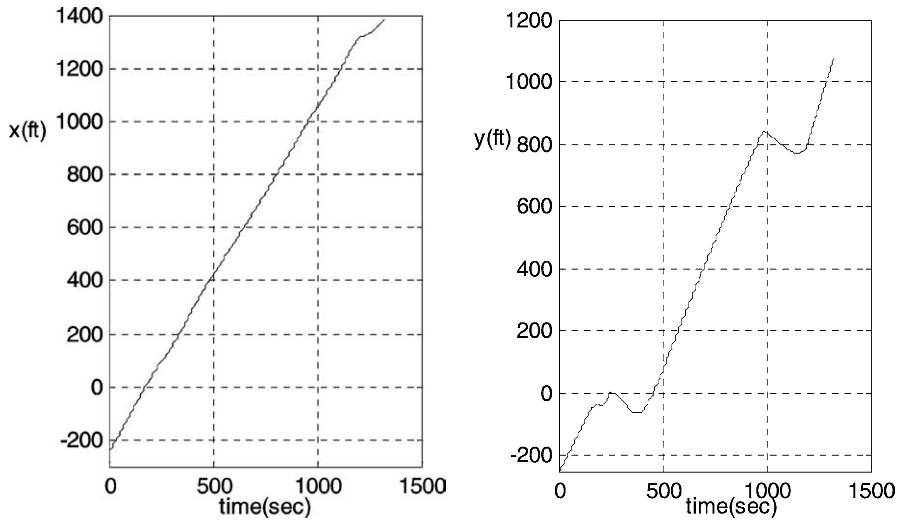


Figure 45. Test case 5: simulated vehicle path in x–y plane; blue dot path: with the presence of 0.5 ft/sec lateral ocean current; red dot path: with no ocean current present.



(a) (b)

Figure 46. Test case 5: simulated vehicle path versus time with the presence of 0.5 ft/sec lateral ocean current.

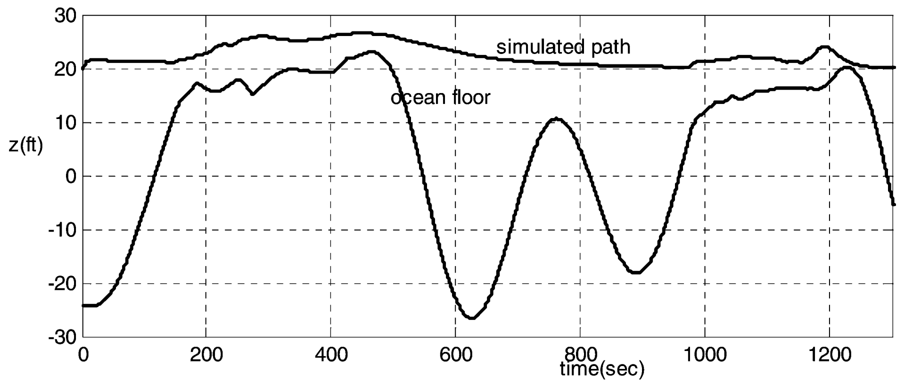
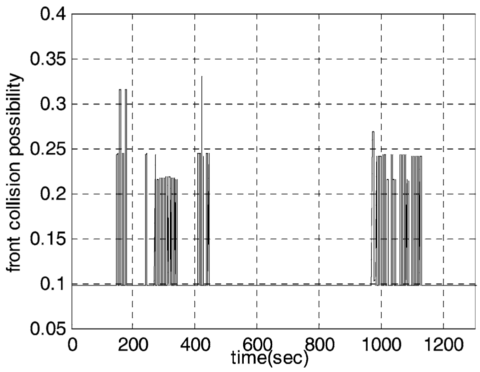


Figure 47. Test case 5: simulated path with the presence of 0.5 ft/sec lateral ocean current versus time $z(t)$ and ocean floor elevation through the vehicle's path.

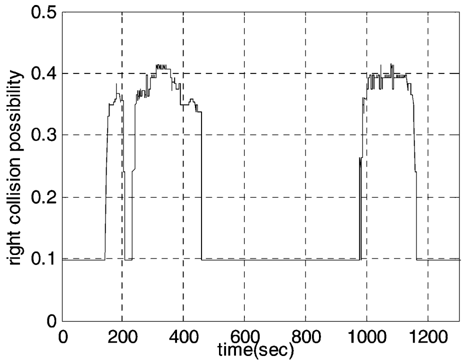
4. Conclusions

The presented architecture has been derived as a general framework for navigation and control of underwater vehicles. It is a generalization of mobile robot specific architectures, with advantages related to modularity, expandability and simplicity. Only the *motion control module* is vehicle dependent, while the other two modules are general. It may account for multiple and diverse sensor fusion, while collision possibilities may be calculated in more primary/secondary directions.

The architecture was tested on the *NPS Phoenix*, a widely known underwater vehicle with accurately derived model. Results have demonstrated the suitability

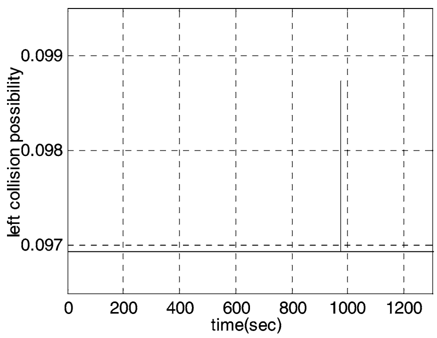


(a)

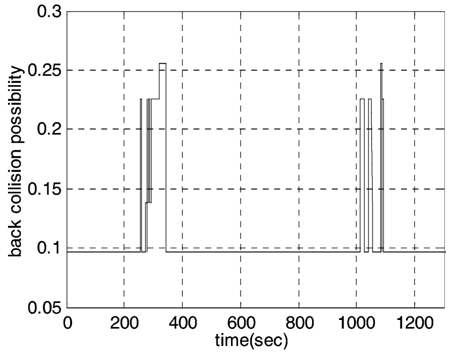


(b)

Figure 48. Test case 5: (a) front collision possibility, (b) right collision possibility with the presence of 0.5 ft/sec lateral ocean current versus time.



(a)



(b)

Figure 49. Test case 5: (a) left collision possibility, (b) back collision possibility with the presence of 0.5 ft/sec lateral ocean current versus time.

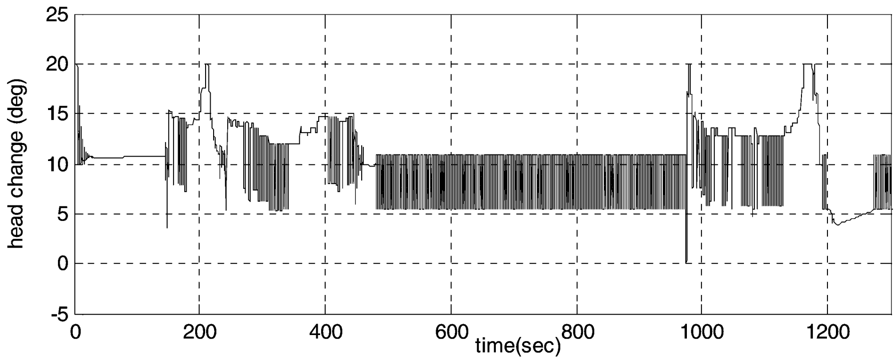


Figure 50. Test case 5: head change with the presence of 0.5 ft/sec lateral ocean current versus time.

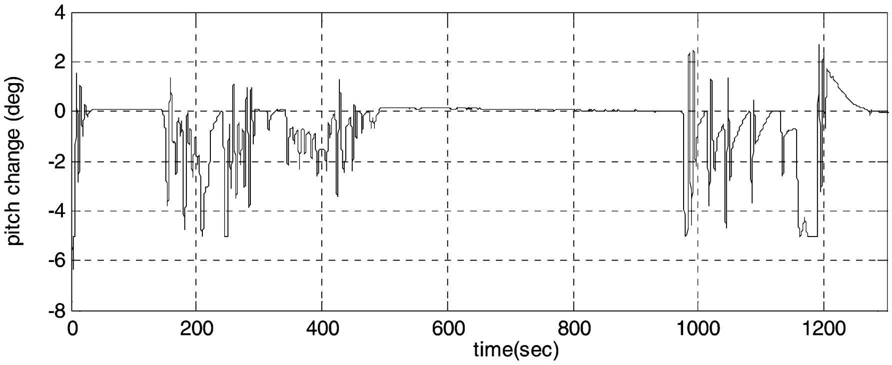


Figure 51. Test case 5: pitch change with the presence of 0.5 ft/sec lateral ocean current versus time.

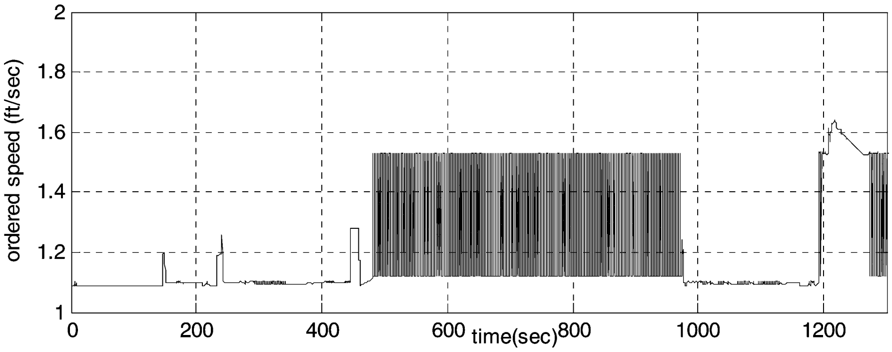


Figure 52. Test case 5: ordered speed with the presence of 0.5 ft/sec lateral ocean current versus time.

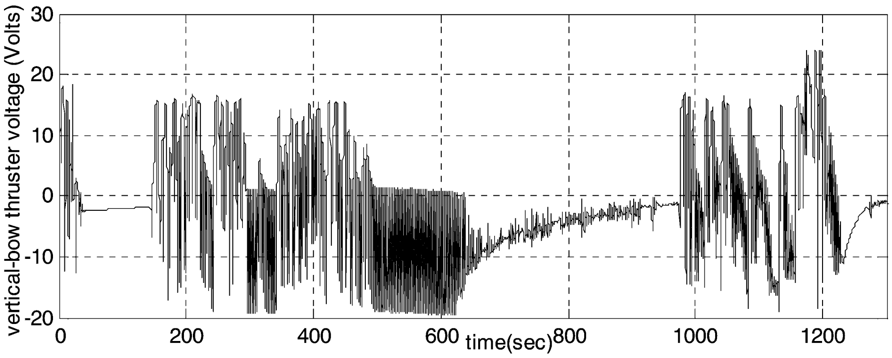


Figure 53. Test case 5: vertical-bow thruster voltage with the presence of 0.5 ft/sec lateral ocean current versus time. Simulation fixed-step size of 0.10 sec.

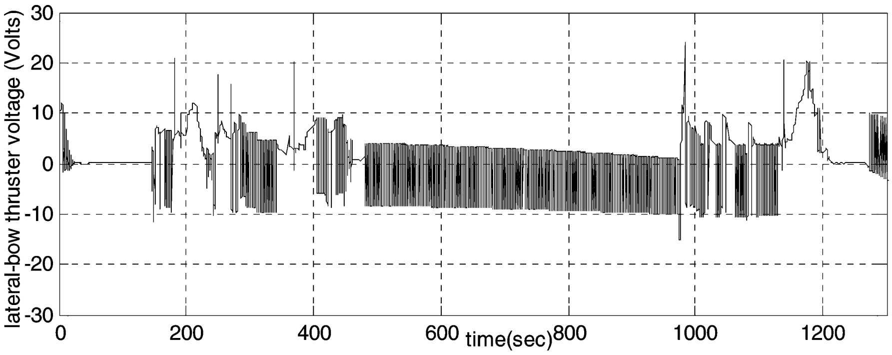


Figure 54. Test case 5: lateral-bow thruster voltage with the presence of 0.5 ft/sec lateral ocean current versus time. Simulation fixed-step size of 0.10 sec.

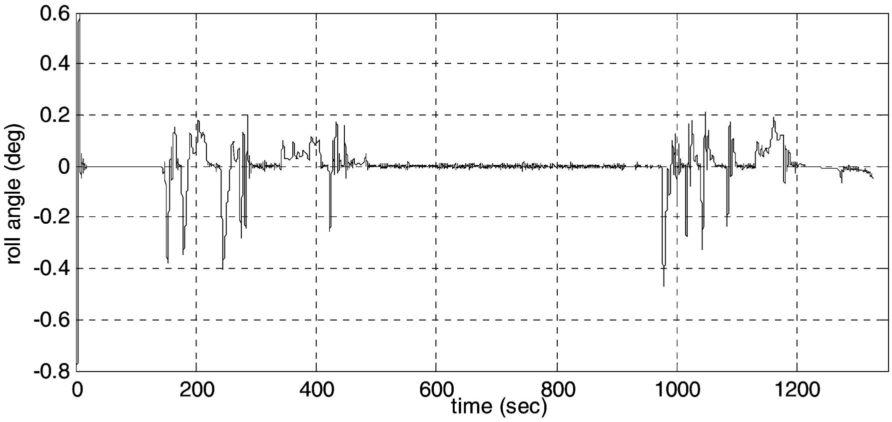


Figure 55. Test case 5: roll angle with the presence of 0.5 ft/sec lateral ocean current versus time.

of the proposed framework and that the fuzzy controllers perform well, even for simultaneous roll control and ocean current presence. The simulation time step time of 0.25 seconds is justifiable given the vehicle maximum speed. Additional tests for larger sonar effective distances do not produce worse results; however, they are not included in this paper.

Appendix A. AUV Kinematics and Dynamics

Consider the marine vehicle shown in Figure 56 with the body-fixed reference frame (X_o, Y_o, Z_o), origin at the vehicle's center of gravity G and with X_o, Y_o, Z_o the vehicle's principal axes of inertia; define the earth-fixed (world) coordinate frame (X, Y, Z) [18]. The vehicle's 3-D position and orientation is determined as a function of 6 DOF with parameters shown in Table III. The first three coordinates and their time derivatives determine the vehicle's position and translational motion

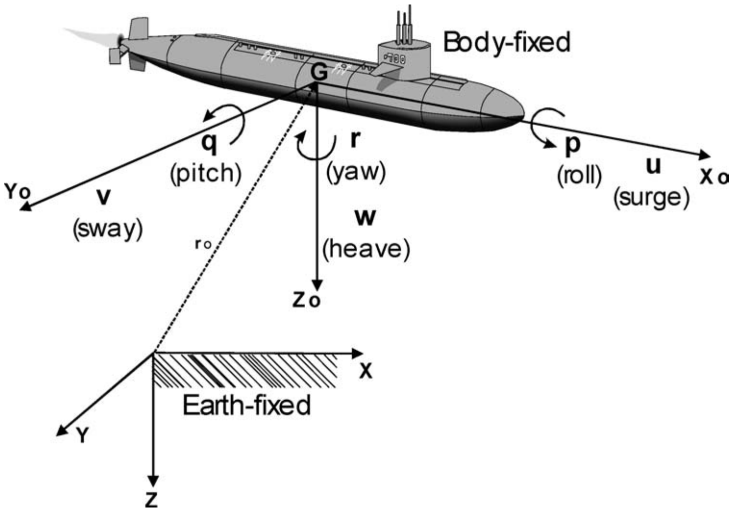


Figure 56. Body-fixed and earth-fixed reference frames and principal motions.

Table III. Notation and coordinates for underwater vehicles

	Name	Forces/ moments	Linear/angular velocities	Position and Euler angles
1	*Surge* (x-axis motion)	X	u	x
2	*Sway* (y-axis motion)	Y	v	y
3	*Heave* (z-axis motion)	Z	w	z
4	*Roll* (rotation about x)	K	p	φ
5	*Pitch* (rotation about y)	M	q	θ
6	*Yaw* (rotation about z	N	r	ψ

along the x-, y-, z-axes, while the last three and their time derivatives determine the vehicle's orientation and rotational motion.

For any marine vehicle, the following vectors are defined:

$$\eta_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \eta_2 = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad \eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}, \quad v_1 = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad v_2 = \begin{bmatrix} p \\ q \\ r \end{bmatrix},$$

$$v_3 = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \qquad \tau_1 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \qquad \tau_2 = \begin{bmatrix} K \\ M \\ N \end{bmatrix}, \qquad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix},$$

η is the vehicle's position and orientation with respect to the earth-fixed frame; v the vehicle's linear and angular velocity with respect to the body-fixed frame; τ the forces and torques with respect to the body-fixed frame. After defining the three principal rotation matrices about the x-, y-, z-axes $C_{x,\varphi}$, $C_{y,\theta}$, $C_{z,\psi}$, respectively, the linear velocity transformation determining the vehicle's path relative to the earth-fixed reference frame is $\dot{\eta}_1 = J_1(\eta_2) \cdot v_1$, $v_1 = J_1^{-1}(\eta_2) \cdot \dot{\eta}_1$, $J_1(\eta_2) = C_{z,\psi}^{\mathrm{T}} \cdot C_{y,\theta}^{\mathrm{T}} \cdot C_{x,\phi}^{\mathrm{T}}$ and $J_1^{-1}(\eta_2) = J_1^{\mathrm{T}}(\eta_2)$. The orientation of the body-fixed reference frame with respect to the earth-fixed frame is given by:

$$v_2 = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + C_{x,\phi} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + C_{x,\phi} C_{y,\theta} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix},$$

or $v_2 = J_2^{-1}(\eta_2)\dot{\eta}_2$. $J_2(\eta_2)$ is undefined for $\theta = \pm 90°$ and $J_2^{-1}(\eta_2) \neq J_2^{\mathrm{T}}(\eta_2)$. Since v_2 cannot be integrated to obtain angular coordinates, η_2 is used instead. Therefore, the vehicle's kinematic equations may be expressed in the following form:

$$\begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} J_1(\eta_2) & 0 \\ 0 & J_2(\eta_2) \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad \Leftrightarrow \quad \dot{\eta} = J(\eta) \cdot v,$$

$$J_1(\eta_2) = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix},$$

$$J_2^{-1}(\eta_2) = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\sin\phi \end{bmatrix},$$

$$J_2(\eta_2) = \begin{bmatrix} 1 & \sin\theta\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix}.$$

The body-fixed 6-DOF nonlinear dynamic equations of motion are represented in compact form (including vehicle thruster forces, hydrodynamics damping, and lift and restoring forces) with the equation $M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau$, $\dot{\eta} = J(\eta)v$, where M is the inertia matrix (including added mass), $C(v)$ the Coriolis and centripetal terms (including added mass), $D(v)$ is the damping matrix, $g(\eta)$ is the gravitational forces and moments vector, τ is the control forces and moments vector. The earth-fixed 6-DOF nonlinear dynamic equations of motion, assuming that $J(\eta)$ is bounded away from $\theta = \pm 90°$ are:

$$\begin{aligned}
&M_\eta(\eta) = J^{-\mathrm{T}}(\eta) \cdot \mathrm{M} \cdot J^{-1}(\eta), \\
&C_\eta(v, \eta) = J^{-\mathrm{T}}(\eta) \cdot \big(C(v) - M \cdot J^{-1}(\eta) \cdot \dot{J}(\eta)\big) \cdot J^{-1}(\eta), \\
&D_\eta(v, \eta) = J^{-\mathrm{T}}(\eta) \cdot D(v) \cdot J^{-1}(\eta),
\end{aligned}$$

Table IV. Control actions of the NPS *Phoenix* AUV

1.	δ_{rbt}	Angle of rudder – bow – top fin
2.	δ_{rbb}	Angle of rudder – bow – bottom fin
3.	δ_{rst}	Angle of rudder – stern – top fin
4.	δ_{rsb}	Angle of rudder – stern – bottom fin
5.	δ_{pbr}	Angle of plane – bow – right fin
6.	δ_{pbl}	Angle of plane – bow – left fin
7.	δ_{psr}	Angle of plane – stern – right fin
8.	δ_{psl}	Angle of plane – stern – left fin
9.	n_{port}	Ordered rpm for port propeller
10.	n_{stbd}	Ordered rpm for stbd propeller
11.	$V_{\mathrm{bow\text{-}vertical}}$	Vertical bow thruster voltage
12.	$V_{\mathrm{stern\text{-}vertical}}$	Vertical stern thruster voltage
13.	$V_{\mathrm{bow\text{-}lateral}}$	Lateral bow thruster voltage
14.	$V_{\mathrm{stern\text{-}lateral}}$	Lateral stern thruster voltage

$$g_\eta(\eta) = J^{-\mathrm{T}}(\eta) \cdot g(\eta),$$
$$\tau_\eta(\eta) = J^{-\mathrm{T}}(\eta) \cdot \tau.$$

So the earth-fixed representation is

$$M_\eta(\eta) \cdot \ddot{\eta} + C_\eta(v, \eta)\dot{\eta} + D_\eta(v, \eta)\dot{\eta} + g_\eta(\eta) = \tau_\eta.$$

Appendix B. The NPS PHOENIX Modified Equations of Motion

The original Phoenix equations of motion have been properly modified to enhance different fin port and starboard fin angles, in order to achieve roll control. For the body-fixed reference frame, the nomenclature for the control actions is given in Table IV.

Surge Equation of Motion.

$$\begin{aligned}
&\left(m - \frac{\rho}{2}L^3 X_{\dot{u}}\right)\dot{u} + mz_G\dot{q} - my_G\dot{r} \\
&\quad = m\big[vr - wq + x_G(q^2 + r^2) - y_G pq - z_G qr\big] \\
&\qquad + \frac{\rho}{2}L^4\big[X_{pp}p^2 + X_{qq}q^2 + X_{rr}r^2 + X_{pr}pr\big] \\
&\qquad + \frac{\rho}{2}L^3\Bigg[X_{wq}wq + X_{vp}vp + X_{vr}vr \\
&\qquad + uq\left(\frac{X_{uq\delta b}}{2}(\delta_{pbR} + \delta_{pbL}) + \frac{X_{uq\delta s}}{2}(\delta_{psR} + \delta_{psL})\right)
\end{aligned}$$

$$
\begin{aligned}
&+ ur\left(\frac{X_{ur\delta r}}{2}(\delta_{rbT}+\delta_{rbB}) + \frac{X_{ur\delta r}}{2}(\delta_{rsT}+\delta_{rsB})\right)\Bigg]\\
&+ \frac{\rho}{2}L^2\Bigg[X_{vv}v^2 + X_{ww}w^2 + uv\left(\frac{X_{uv\delta r}}{2}\right)(\delta_{rsT}+\delta_{rsB})\\
&+ uw\left(\frac{X_{uw\delta b}}{2}(\delta_{pbR}+\delta_{pbL}) + \frac{X_{uw\delta s}}{2}(\delta_{psR}+\delta_{psL})\right)\\
&+ u|u|\left(X_{u|u|\delta b\delta b}\left(\frac{\delta_{pbR}+\delta_{pbL}}{2}\right)^2 + X_{u|u|\delta s\delta s}\left(\frac{\delta_{psR}+\delta_{psL}}{2}\right)^2\right.\\
&\left.+ X_{u|u|\delta r\delta r}\left(\frac{\delta_{rbT}+\delta_{rbB}}{2}\right)^2 + X_{u|u|\delta r\delta r}\left(\frac{\delta_{rsT}+\delta_{rsB}}{2}\right)^2\right)\Bigg]\\
&- (W-B)\sin(\theta)\\
&+ \frac{\rho}{2}L^2C_{d0}\left[(2\,\mathrm{ft/sec}/700\,\mathrm{rpm})^2\frac{1}{2}(n_{\mathrm{port}}|n_{\mathrm{port}}| + n_{\mathrm{stbd}}|n_{\mathrm{stbd}}| - u|u|\right].
\end{aligned}
$$

Sway Equation of Motion.

$$
\begin{aligned}
&\left(m - \frac{\rho}{2}L^3Y_{\dot v}\right)\dot v + \left(-mz_G - \frac{\rho}{2}L^4Y_{\dot p}\right)\dot p + \left(mx_G - \frac{\rho}{2}L^4Y_{\dot r}\right)\dot r\\
&\quad= m\big[-ur + wp - x_Gpq + y_G(p^2+r^2) - z_Gqr\big]\\
&\quad+ \frac{\rho}{2}L^4[Y_{pq}pq + Y_{qr}qr]\\
&\quad+ \frac{\rho}{2}L^3[Y_{uq}uq + Y_{ur}ur + Y_{vq}vq + Y_{wp}wp + Y_{wr}wr]\\
&\quad+ \frac{\rho}{2}L^2\Bigg[Y_{uv}uv + Y_{vw}vw\\
&\quad+ u|u|\left(\frac{Y_{u|u|\delta rb}}{2}(\delta_{rbT}+\delta_{rbB}) + \frac{Y_{u|u|\delta rs}}{2}(\delta_{rsT}+\delta_{rsB})\right)\Bigg]\\
&\quad- \frac{\rho}{2}\int_{\mathrm{tail}}^{\mathrm{nose}}\big[(c_{dy}h(x)(v+xr)^2 + c_{dz}b(x)(w-xq)^2\big]\frac{v+xr}{U_{cf}(x)}\,\mathrm{d}x\\
&\quad+ (W-B)\cos(\theta)\sin(\varphi)\\
&\quad+ \left(\frac{2lb}{24^2}\,\mathrm{Volts}\right)\big[V_{\text{bow-lateral}}|V_{\text{bow-lateral}}| + V_{\text{stern-lateral}}|V_{\text{stern-lateral}}|\big].
\end{aligned}
$$

Heave Equation of Motion.

$$
\begin{aligned}
&\left(m - \frac{\rho}{2}L^3Z_{\dot w}\right)\dot w + my_G\dot p + \left(-mx_G - \frac{\rho}{2}L^4Z_{\dot q}\right)\dot q\\
&\quad= m\big[uq - vp - x_Gpr - y_Gqr + z_G(p^2+q^2)\big]
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{\rho}{2}L^4[Z_{pp}p^2 + Z_{pr}pr + Z_{rr}r^2]\\
&+\frac{\rho}{2}L^3[Z_{uq}uq + Z_{vp}vp + Z_{vr}vr]\\
&+\frac{\rho}{2}L^2\bigg[Z_{uw}uw + Z_{vv}v^2 + u|u|\bigg(Z_{u|u|\delta b}\bigg(\frac{\delta_{pbR}+\delta_{pbL}}{2}\bigg)\\
&+Z_{u|u|\delta s}\bigg(\frac{\delta_{psR}+\delta_{psL}}{2}\bigg)\bigg)\bigg]\\
&-\frac{\rho}{2}\int_{\mathrm{tail}}^{\mathrm{nose}}\big[(c_{dy}h(x)(v+xr)^2 + c_{dz}b(x)(w-xq)^2\big]\frac{w-xq}{U_{cf}(x)}\,\mathrm{d}x\\
&+(W-B)\cos(\theta)\cos(\varphi)\\
&-\bigg(\frac{2lb}{24^2}\,\mathrm{Volts}\bigg)[V_{\text{bow-vertical}}|V_{\text{bow-vertical}}| + V_{\text{stern-vertical}}|V_{\text{stern-vertical}}|].
\end{aligned}
$$

Roll Equation of Motion.

$$
\begin{aligned}
&\bigg(mz_G - \frac{\rho}{2}L^4K_{\dot v}\bigg)\dot v + my_G\dot w + \bigg(I_x - \frac{\rho}{2}L^5K_{\dot p}\bigg)\dot p - I_{xy}\dot q + \bigg(-I_{xz} - \frac{\rho}{2}L^5K_{\dot r}\bigg)\dot r\\
&\quad = \big[-(I_z - I_y)qr - I_{xy}pr + I_{yz}(q^2 - r^2) + I_{xz}pq\big]\\
&\qquad - m\big[y_G(-uq + vp) - z_G(ur - wp)\big]\\
&\qquad + \frac{\rho}{2}L^5[K_{pq}pq + K_{qr}qr + K_{p|p|}p|p| + K_p p]\\
&\qquad + \frac{\rho}{2}L^4[K_{|u|p}|u|p| + K_{ur}ur + K_{vq}vq + K_{wp}wp + K_{wr}wr]\\
&\qquad + \frac{\rho}{2}L^3\bigg[K_{uv}uv + K_{vw}vw - u|u|\bigg(K_{u|u|\delta p}\frac{\delta_{pbR}+\delta_{pbL}}{2}\\
&\qquad + K_{u|u|\delta s}\frac{\delta_{psR}+\delta_{psL}}{2}\bigg)\bigg]\\
&\qquad + (y_GW - y_BB)\cos(\theta)\cos(\varphi) - (z_GW - z_BB)\cos(\theta)\sin(\varphi)\\
&\qquad + \frac{\rho}{2}|y_{\text{fin-pres}}|L^2\bigg[u|u|\bigg(Z_{u|u|\delta b}\frac{\delta_{pbR}-\delta_{pbL}}{2} + Z_{u|u|\delta s}\frac{\delta_{psR}-\delta_{psL}}{2}\bigg)\bigg]\\
&\qquad + \frac{\rho}{2}|z_{\text{fin-pres}}|L^2\bigg[u|u|\bigg(Y_{u|u|\delta rb}\frac{(\delta_{rbT}-\delta_{rbB}}{2} + Y_{u|u|\delta rs}\frac{\delta_{rsT}-\delta_{rsB}}{2}\bigg)\bigg].
\end{aligned}
$$

Pitch Equation of Motion.

$$
\begin{aligned}
&mz_G\dot u + \bigg(-mx_G - \frac{\rho}{2}L^4M_{\dot w}\bigg)\dot w - I_{xy}\dot p + \bigg(I_y - \frac{\rho}{2}L^5M_{\dot q}\bigg)\dot q - I_{yz}\dot r\\
&\quad = \big[-(I_x - I_z)pr + I_{yz}qr - I_{yz}pq - I_{xz}(p^2 - r^2)\big]
\end{aligned}
$$

$$
\begin{aligned}
&+ m\big[x_G(-uq + vp) - z_G(-vr + wq)\big] \\
&+ \frac{\rho}{2} L^5\big[M_{pp}p^2 + M_{pr}pr + M_{r|r|}r|r| + M_{q|q|}q|q| + M_q q\big] \\
&+ \frac{\rho}{2} L^4[M_{uq}uq + M_{vp}vp + M_{vr}vr] \\
&+ \frac{\rho}{2} L^3\bigg[M_{uw}uw + K_{vv}v^2 \\
&+ u|u|\bigg(M_{u|u|\delta b}\frac{\delta_{pbR} + \delta_{pbL}}{2} + M_{u|u|\delta s}\frac{\delta_{psR} + \delta_{psL}}{2}\bigg)\bigg] \\
&+ \frac{\rho}{2}\int_{\text{tail}}^{\text{nose}} \big[(c_{dy}h(x)(v + xr)^2 + c_{dz}b(x)(w - xq)^2\big]\frac{(w - xq)x}{U_{cf}(x)}\,\mathrm{d}x \\
&- (x_G W - x_B B)\cos(\theta)\cos(\varphi) - (z_G W - z_B B)\sin(\theta) \\
&- \bigg(\frac{2lb}{24^2}\ \text{Volts}\bigg)\big[V_{\text{bow-vertical}}|V_{\text{bow-vertical}}|x_{\text{bow-vertical}} \\
&\qquad\qquad + V_{\text{stern-vertical}}|V_{\text{stern-vertical}}|x_{\text{stern-vertical}}\big].
\end{aligned}
$$

Yaw Equation of Motion.

$$
\begin{aligned}
&m y_G \dot{u} + \bigg(m x_G - \frac{\rho}{2}L^4 N_{\dot{v}}\bigg)\dot{v} + \bigg(-I_{xz} - \frac{\rho}{2}L^5 N_{\dot{p}}\bigg)\dot{p} - I_{yz}\dot{q} + \bigg(Iz - \frac{\rho}{2}L^5 N_{\dot{r}}\bigg)\dot{r} \\
&\quad = \big[-(I_y - I_x)pq + I_{xy}(p^2 - q^2) + I_{yz}pr - I_{xz}qr\big] \\
&\qquad - m\big[x_G(ur - wp) - y_G(-vr + wq)\big] \\
&\qquad + \frac{\rho}{2}L^5\big[N_{pq}pq + N_{qr}qr + N_{r|r|}r|r| + N_r r\big] \\
&\qquad + \frac{\rho}{2}L^4\big[N_{up}up + N_{ur}ur + N_{vq}vq + N_{wp}wp + N_{wr}wr\big].
\end{aligned}
$$

References

1. Amat, J., Monferrer, A., Batlle, J., and Cufi, X.: GARBI: A low-cost underwater vehicle, *Microprocessors Microsystems* **23** (1999), 61–67.
2. Antonelli, G., Cassavale, F., Chiaverini, S., and Fusco, G.: A novel adaptive control law for underwater vehicles, *IEEE Trans. Control Systems Technol.* **11**(2) (2003), 221–232.
3. Antonelli, G., Chiaverini, S., Sarkar, N., and West, M.: Adaptive control of an autonomous underwater vehicle: Experimental results on ODIN, *IEEE Trans. Control Systems Technol.* **9** (July 2001), 756–765.
4. Arkin, R. C.: *Behavior-Based Robotics*, MIT Press, 1998.
5. Balch, T. and Parker, L. E. (eds): *Autonomous Robots*, A. K. Press, 2002.
6. Brooks, R. A.: A robust layered control system for a mobile robot, *IEEE J. Robotics Automat.* **2** (1986), 14–23.

7. Brutzman, D. P.: A virtual world for an autonomous underwater vehicle, PhD Thesis, Naval Postgraduate School, Monterey, CA, 1994.
8. Brutzman, D. P.: Virtual world for an autonomous underwater vehicle, Tutorial notes, in: *MTS/IEEE OCEANS '96*, 1996.
9. Corradini, M. L. and Orlando, G.: A discrete adaptive variable structure controller for MIMO systems and its application to an underwater ROV, *IEEE Trans. Control Systems Technol.* **5** (1997), 349–359.
10. Coste-Maniere, E., Wang, H. H., and Peuch, A.: Control architectures: What's going on? in: *Proc. of Internat. Program Development in Undersea Robotics and Intelligent Control (URIC): A Joint U.S./Portugal Workshop*, Lisbon, Portugal, 1995, pp. 54–60.
11. Cristi, R., Papoulias, F. A., and Healey, A.: Adaptive sliding mode control of autonomous underwater vehicles in the dive plane, *IE* **15** (1990), 152–160.
12. Cui, Y. and Sarkar, N.: A unified force control approach to autonomous underwater manipulation, in: *Proc. of IEEE Conf. on Robotics and Automation*, San Francisco, CA, 2000, pp. 1263–1268.
13. DeBitetto, P.: Fuzzy logic for depth control of unmanned undersea vehicles, *IEEE J. Oceanic Engrg.* **20**(3) (1995), 242–248.
14. Driankov, D., Hellendoorn, H., and Reinfrank, M.: *An Introduction to Fuzzy Control*, Springer, Berlin, 1996.
15. Doitsidis, L., Valavanis, K. P., and Tsourveloudis, N. C.: Fuzzy logic based autonomous skid steering vehicle navigation, in: *CD-ROM Proc. of the IEEE Internat. Conf. on Robotics and Automation*, Washington, DC, May 2002.
16. Doitsidis, L., Valavanis, K. P., and Tsourveloudis, N. C.: Sonar sensor based autonomous navigation and collision avoidance of skid-skeering mobile robots, *J. Autonom. Robots* (submitted).
17. Farinwata, S. S., Filev, D. E., and Langari, R. (eds): *Fuzzy Control: Synthesis and Analysis*, Wiley, New York, 2000.
18. Fossen, T.: *Guidance and Control of Ocean Vehicles*, Wiley, New York, 1994.
19. Fossen, T. I. and Sagatun, S. I.: Adaptive control of nonlinear systems: A case study of underwater robotic systems, *J. Robotic Systems* **8** (1991), 393–412.
20. Ganesan, K. and Smith, S. M.: A pragmatic software architecture for UUVs, in: *Proc. of Symposium on Autonomous Underwater Vehicle Technology*, 1996, pp. 209–215.
21. Guo, J. and Huang, S. H.: Control of an autonomous underwater vehicle testbed using fuzzy logic and genetic algorithms, Department of Naval Architecture and Ocean Engineering, National Taiwan University, Taipei.
22. Healey, A. J.: Analytical redundancy and fuzzy inference in AUV fault detection and compensation, in: *Proc. of Oceanology-1998*, Brighton, 1998, pp. 45–50.
23. Healey, A. J. and Brutzman, D. P.: *Underwater Robotics Workshop*, *Proc. of the 8th Internat. Conf. on Advanced Robotics*, July 1997.
24. Healey, A. J. and Lienard, D.: Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles, *IEEE J. Ocean Engrg.* **18**(3) (1993), 327–339.
25. Healey, A. J. and Marco, D. B.: Slow speed flight control of autonomous underwater vehicles: Experimental results with the NPS AUV II, in: *Proc. of the 2nd Internat. Offshore and Polar Engineering Conf.*, San Francisco, CA, 1992, pp. 523–532.
26. Healey, A. J., Marco, D. B., McGhee, R. B., Brutzman, D. P., and Cristi, R.: Evaluation of the NPS Phoenix autonomous underwater vehicle hybrid control system, in: *Proc. of American Control Conference*, Seattle, 1995, pp. 2954–2963.
27. Healey, A. J., Marco, D. B., McGhee, R. B., Brutzman, D. P., Cristi, R., and Papoulias, F. A.: Coordinating the hovering behaviors of the NPS AUVII using onboard sonar servoing, in: *Proc. of IARP 2nd Workshop on Mobile Robots for Subsea Environments*, Monterey, CA, 1994, pp. 53–62.

28. Janabi-Sharifi, F. and Wilson, W. J.: An intelligent assembly robotics system based on relative pose measurements, *J. Intelligent Robotic Systems* **12**(1) (1995), 49–86.
29. Kanakakis, V., Tsourveloudis, N. C., and Valavanis, K. P.: Design and testing of a fuzzy logic controller for an autonomous underwater vehicle, in: *CD-ROM Proc. of the IARP Internat. Workshop on Underwater Robotics for Sea Exploration and Environmental Monitoring*, Rio de Janeiro, Brazil, October 2001.
30. Kato, N.: Applications of fuzzy algorithm to guidance and control of underwater vehicles, underwater robotic vehicles, in: J. Yuh (ed.), *Underwater Robotic Vehicles: Design and Control*, TSI Press, 1995.
31. McClarin, D.: Discrete asynchronous Kalman filtering of navigation data for the Phoenix AUV, Tutorial, March 1996.
32. Rosenblatt, J., Williams, S., and Durrant-Whyte, H.: Behavior-based control for autonomous underwater exploration, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, San Francisco, CA, 2000.
33. Saffiotti, A.: The uses of fuzzy logic for autonomous robot navigation, *Soft Computing* **1**(4) (1997), 180–197.
34. Saffiotti, A.: Handling uncertainty in control of autonomous robots, in: A. Hunter and S. Parsons (eds), *Uncertainty in Information Systems*, Lecture Notes in Artificial Intelligence 1455, Springer, Berlin, 1998, pp. 198–224.
35. Staff, T. S.: *Remotely Operated Vehicles of the World*, Ocean News and Technology Press, 1998.
36. Tsourveloudis, N., Gracanin, D., and Valavanis, K. P.: Design and testing of navigation algorithm for shallow water autonomous underwater vehicle, in: *Proc. of OCEANS '98*, 1998.
37. Tsourveloudis, N., Valavanis, K. P., and Hebert, T.: Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic, *IEEE Trans. Robotics Automat.* **17**(4) (2001), 490–497.
38. Valavanis, K. P., Gracanin, D., Matijasevic, M., Kolluru, R., and Demetriou, G. A.: Control architectures for autonomous Uuderwater vehicles, *IEEE Control Systems Mag.* **17** (December 1997), 48–64.
39. Valavanis, K. P., Hebert, T., Kolluru, R., and Tsourveloudis, N.: Mobile robot navigation in 2-dynamic environments using electrostatic potential fields, *IEEE Trans. Systems Man Cybernet. Part A* **30**(2) (2000), 187–196.
40. Vukic, Z., Omerdic, E., and Kuljaca, L.: Fuzzy autopilot for ships experiencing shallow water effect in manouvering, in: *Proc. of the 4th IFAC Conf. on Manouvering and Control of Marine Craft*, Brijuni, HR, 1997, pp. 69–74.
41. Wang, J. S., Lee, C. S. G., and Yuh, J.: An on-line self-organizing neuro-fuzzy control for autonomous underwater vehicles, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, Detroit, MI, 1999, pp. 2416–2421.
42. Wang, J. S., Lee, C. S. G., and Yuh, J.: Self-adaptive neuro-fuzzy systems with fast parameter learning for autonomous underwater vehicles, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, San Francisco, CA, 2000, pp. 3861–3865.
43. Whitcomb, L. L.: Underwater robotics: Out of the research laboratory and into to the field, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, San Francisco, CA, 2000, pp. 709–716.
44. White, K. A., Smith, S. M., Ganesan, K., Kronen, D., Rae, G. J. S. and Langenbach, R. M.: Performance results of a fuzzy behavioral altitude flight controller and rendezvous and docking of an autonomous underwater vehicles with fuzzy control, in: *Proc. of Symposium on AUV Technology*, 1996, pp. 117–124.
45. Yen, J. and Langari, R.: *Fuzzy Logic*, Prentice-Hall, Englewood Cliffs, NJ, 1999.
46. Yuh, J. (ed.): *Underwater Robotic Vehicles: Design and Control*, TSI Press, 1995.

47. Yuh, J., Nie, J., and Lee, C. S. G.: Experimental study on adaptive control of underwater robots, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, Detroit, MI, 1999, pp. 393–398.
48. Zaslavsy, G. R. and Kandel, A.: Longitudinal fuzzy control of a submerged vehicle, *Fuzzy Sets Systems* **115**(2) (2000), 305–319.