

Intelligent Control System for Autonomous Underwater Robots with Vision Capabilities

Jebelli A*, Yagoub MCE and Dhillon BS

Faculty of Engineering, University of Ottawa, Canada

Abstract

In this work, an autonomous underwater robot prototype capable of detecting obstacles for safe navigation had been successfully designed and tested. It uses an efficient fuzzy-based system for controlling its movement and a robust image processing algorithm to process data collected by two embedded cameras.

Keywords: Autonomous Underwater Vehicles (AUV); Proportional-Integral-Derivative (PID) controllers; Fuzzy logic; Cameras

Introduction

Autonomous Underwater Vehicles (AUVs) can manage their operation by reliable programming of high-level controllers that make the device able to maneuver even under unpredicted conditions. However, decision-making issues should be adequately addressed for consistent operation, among them safe navigation through potential obstacles.

During the last few years, number of approaches have been proposed to tackle this problem but they mainly suffer from relatively slow response when there is a real need for quick decision-making [1-4].

In the designed prototype (Figure 1), two cameras have been implemented, one on the top and the one on the bottom, to collect data from the environment in order to detect obstacles. Furthermore, knowing that sensing in aquatic environments may result in image distortion occurring by refraction of light, a robust image processing technique was developed to process data through Proportional-Integral-Derivative (PID) controllers.

A technique combining color and Local Binary Pattern (LBP) characteristics was developed to efficiently describe the target and a tracking algorithm built by using these features. Tests under real conditions demonstrated the proposed approach.

Robot Design

To safely navigate, an autonomous vehicle should be able to accurately detect obstacles, make quick and appropriate decisions and select the suitable route to circumvent them.

To this aim, two thrusters were integrated to rapidly modify a pre-defined itinerary. To assure a fast response, the thruster angles can be easily changed to quickly provide both vertical and horizontal forces

needed to move the prototype once a safer direction is selected. Note that any change in terms of angle has been made possible by the instant movement of an engine stopper, which consumes much less energy than the constant movement of a thruster.

Next, a mass shifter was included, first because of the thruster movement and, second, to make the maneuver possible in the direction of the pitch. The whole set can have two movement modes. In the first mode, thanks to the mass shifter, the body should have a constant horizontal movement and its movement towards vertical and horizontal directions should be done through a change in the thruster angle. In the second mode, the thrusters should be kept fixed in a horizontal direction and the vertical movement should be made possible through a change in the body angle in the pitch direction. The first mode is used wherever there is little room of maneuver or to pass obstacles while the second mode is dedicated to preserve energy and/or change in depths. Note that this structure is innovative and energy-saving.

Once the issue of quick moving response solved, different sensors have been implemented to collect data from the device environment. A MPU-9250 Inertial Measurement Unit (IMU) sensor has been used [5]. It is composed of three sensors, i.e., an acceleration counter to measure the acceleration of the device, its balance and its deviation, a gyroscope to measure the circulation rate of the device and finally, a magnetometer to determine the position of the device relatively to the North Pole.

A HMC6343 compass sensor has been also incorporated to determine the heading point of the robot [6]. Because the magnetometer sensor is highly sensitive to electromagnetic noise and earth magnetic field intensity, its operation in submarine conditions might be not as accurate as expected.

Therefore, a separate digital compass sensor was integrated to increase accuracy. The selected compass sensor is tilt compensated and calibrated to handle magnetic distortions. It combines 3-axis magnetoresistive sensor and 3-axis MEMS accelerometer and compute a heading direction every 200 ms [6].

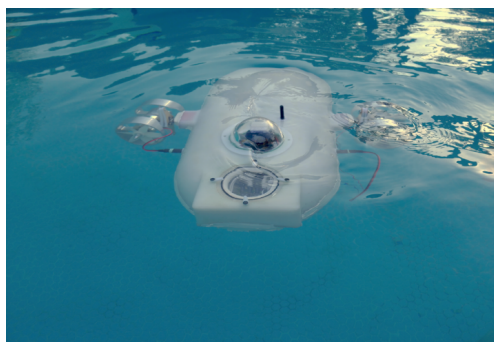


Figure 1: The robot prototype in operation.

*Corresponding author: Jebelli A, Faculty of Engineering, University of Ottawa, Canada, Tel: +1 613-562-5700; E-mail: ajebelli@uottawa.ca

Received April 29, 2017; Accepted June 13, 2017; Published June 16, 2017

Citation: Jebelli A, Yagoub MCE, Dhillon BS (2017) Intelligent Control System for Autonomous Underwater Robots with Vision Capabilities. J Appl Mech Eng 6: 270. doi: 10.4172/2168-9873.1000270

Copyright: © 2017 Jebelli A, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Because it is possible to define a linear relation between water pressure and water depth, the MS5803-14BA pressure sensor was used to measure the device depth. It is a high-resolution pressure sensor optimized for depth measurement systems with a water depth resolution of around 1 cm [7].

Image Processing

Color-texture histogram

Quick tracking of objects is considered as an essential issue in image processing applications and machine vision. We can be facing with many issues such as noise, image blocking, flicker or drastic changes in the background or foreground for tracking of objects in video images. Many algorithms have been proposed to address them.

Mean shift is one of the most popular because of its simplicity and efficiency [8]. Also, the most common way to display and describe an object or intended target in an image is to use histograms, considered as a density function of the target area. This approach is relatively strong to describe the appearance of objects. However, using only histogram in mean shift can raise some issues: one of them being that the target location information can disappear or, if the target has an appearance close to background, it cannot be well recognized. Therefore, additional features such as gradients or edge information can be used in alongside the histogram to resolve this problem [8].

Texture patterns represent the spatial structure of an object in the image. They are a proper feature to identify and display a target. Image texture features provide more and better data against image color histogram. Hence, the simultaneous use of color and texture histogram of image can increase the capacity of tracking confidence, particularly in complex images. But effective use of both texture and color features together is still a challenge [8]. In this work, the Local Binary Patterns (LBP) approach was thus used mainly because it is an effective and efficient technique to describe the texture characteristics of image with high computational features and unchangeable speed with rotation. It has been applied in a variety of areas including image texture analysis, facial recognition, image segmentation, etc. More recently, LBP pattern has been also used to track objects [8].

Local binary patterns

Through thresholding, the neighboring pixels with central pixel, LBP operator allocates a number to each pixel; this operator can be described as [8]: where g_c is the color intensity of the central pixel and g_p the color intensity of the P neighboring pixels located within a circle of center g_c and radius R. The function $s(x)$ is defined as:

$$LBP_{PR}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (1)$$

where g_c is the color intensity of the central pixel and g_p the color intensity of the P neighboring pixels located within a circle of center g_c and radius R. The function $s(x)$ is defined as:

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (2)$$

Note that a different resolution can be obtained by changing the parameters P and R. Another form of LBP operator is the immutable rotation defined as follows where the index 'riu2' refers to "immutable with rotation":

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{if } U(LBP_{P,R}) \leq 2 \\ P+1 & \text{otherwise} \end{cases} \quad (3)$$

$$U(LBP_{P,R}^{riu2}) = \left| \sum_{p=0}^{P-1} s(g_p - g_c) - s(g_0 - g_c) \right| \quad (4)$$

Target description by color-texture histogram

In describing the target, the micro-textons at the end of lines and corners, as flat patterns of the majority, represent the main characteristics of the target while, on the other hand, flat points and areas, as uniform patterns of the minority, are quantitative textures. In general, majority patterns are more important than the minority ones in describing the target, so we need to extract the majority patterns using the following formula [8].

$$LBP_{8,1}^{riu2} = \begin{cases} \sum_{p=0}^7 s(g_p - g_c + a) & U(LBP_{8,1}) \leq 2 \text{ and} \\ 0 & \sum_{p=0}^7 s(g_p - g_c + a) \in \{2, 3, 4, 5, 6\} \\ & \text{otherwise} \end{cases} \quad (5)$$

Patterns No. 0, 1, 7, and 8 are related to uniform patterns of the minority, No. 9 is related to non-uniform pattern, and No. 2 to 6 are related to original uniform patterns. Equation (5) classifies the minority patterns in the category of non-uniform patterns. Then, these patterns are used to describe the target. Originally, it can be said that we initially used the (5) as a mask and then we modeled the target by using the color features and LBP inside this mask. Using the model provided, only pixels of the desired object which have been extracted by Equation (5) is used to describe it. In this report, these points combine the color histogram of the image with LBP histogram and model the target. This method eliminates the flat backgrounds very well and also remove the noise interference effect; on the whole, this method extracts the main features of the image.

Figure 2 shows a test example. Figure 2.3 (Left) shows the target area in consecutive frames and Figure 2.3 (Right) shows the result of equation (5) after applying the mask in which the non-black pixels represent key features of the image.

PID Controllers

The next step was to implement the controllers. PID controllers are commonly used to monitor the variations of physical quantities such as engine velocity, pressure, or temperature.

A PID controller calculates the "error" value between the process output and a predefined desired set point. The purpose of the controller is to minimize this error by accurately setting a process control adjustment. A PID is mainly composed of three parts namely, proportional, integral, and derivative; each of them considers the error signal $e(t)$ as input while their output is summed up as:

$$Output(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de}{dt} \right) \quad (6)$$

and compared to the output of the system under consideration. The controller parameters are proportional gain K_p , integral time T_i , and

derivative time T_d . Thus, the transfer function of a PID controller can be expressed as:

$$G_c = K_p + \frac{K_i}{s} + K_d s \quad (7)$$

with s the complex number frequency, K_i the integral term, and K_d the derivative term. The behavior of a PID controller can be stated as follows: the K_p coefficient increases the system velocity and reduces the steady state error (but does not make it equal to zero). Adding the integral term K_i makes the steady state error zero but adds high overshooting to the transient response. The derivative term K_d weakens the transient response and encloses the step response to the ideal step form [9-11].

Fuzzy Logic Controller

Fuzzy logic controllers (FLC) involve using logical rules (IF-THEN rules) to either accept control system parameters, design, or change device conditions. Decision rules are combined by evaluating the factors considering the level at which each rule's constraints are met (membership function). Therefore, the controller output will be a weighted combination of all rules, which are to be effective at any given time. Kanakakis et al. [12] applied this approach to diver-less submarine vehicles while their application to AUVs has been widely

Left/Right	$\mu_L - \mu_M$
FLC Inputs	$\mu_R - \mu_M$
Up/Down	$\mu_t - \mu_m$
FLC Inputs	$\mu_b - \mu_m$

Table 1: Input fuzzy decision system.

		$\mu_R - \mu_m$				
Parameters		PB	PS	Z	NS	NB
$\mu_L - \mu_M$	PB	Z	Z	Z	Z	P
	PS	Z	Z	Z	P	P
	Z	Z	Z	Z	P	P
	NS	Z	N	N	P	P
	NB	N	N	N	N	P

Table 2: Fuzzy rules to decide dc motors performance decision to move the robot in yaw degrees of freedom (p: forward Rotation of the motors, z: off, n: backward rotation of the motors).

utilized [13-15]. As shown in Figure 3, a fuzzy controller is composed of four main parts: fuzzifier, rules base, decision-making part and defuzzifier.

In summary, the input is presented to the fuzzifier, which relates a normalized value between 0 and 1 to the input, based on the defined membership functions. This value, determined according to the linguistic rule as if-then term, is finally defuzzified and sent to the next level.

In the proposed fuzzy logic approach, the processed image is divided into three horizontal sections, each of them getting a value between 0 and 1 depending on the probability of existence of an obstacle. Then the image is divided into three vertical sections and the above scoring method is applied to these three sections as well.

The number related to each section is equal to the ratio of the number of white pixels in that band to its total pixels. To prevent noise error in image processing, a threshold was defined as follow: if the above ratio is less than 2%, the related band is assumed free of obstacles. A sample scoring is illustrated in Figure 4.

Then to prepare the fuzzy system inputs, let us state the band scores as μ_L (left vertical band), μ_M (middle vertical band), μ_R (right vertical band), μ_t (top horizontal band), μ_m (middle horizontal band), and μ_b (bottom horizontal band). Table 1 summarizes the fuzzy controller inputs, knowing that each of the above scores are reduced from the middle part score.

The fuzzy system decision is based on the score each section obtained compared with its corresponding middle band. Let us consider first the vertical bands and assume, for instance, that the term $\mu_L - \mu_M$ is negative; the vertical middle part obtains both a higher score in detecting an obstacle and a higher share than the left band in terms of location. Therefore, the left vertical band is a better choice for the robot's next movement. At the same time, the term $\mu_R - \mu_M$ is calculated and by comparing it to the previous one, the fuzzy system makes its decision about the updated Yaw direction of the robot. The decision output will then determine the differential velocity between the left and right engines. The number provided by the fuzzy system is reduced from the average velocity of the left DC engine and added to the right DC engine.

The fuzzy rules for this system are shown in Table 2. This decision table was set to favor direct tracks. In fact, if the vertical side bands are not significantly different from the middle band in terms of scoring, the direct track is the robot's next decision to move. Accordingly, sudden and large movements in a given direction are avoided and wasting

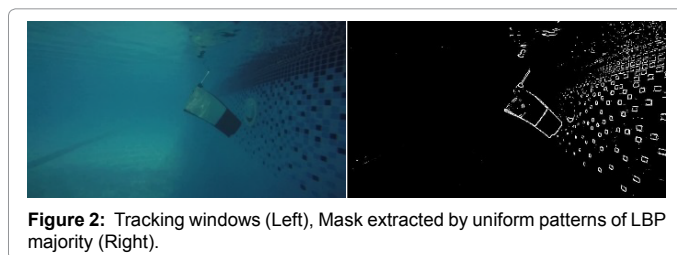


Figure 2: Tracking windows (Left), Mask extracted by uniform patterns of LBP majority (Right).

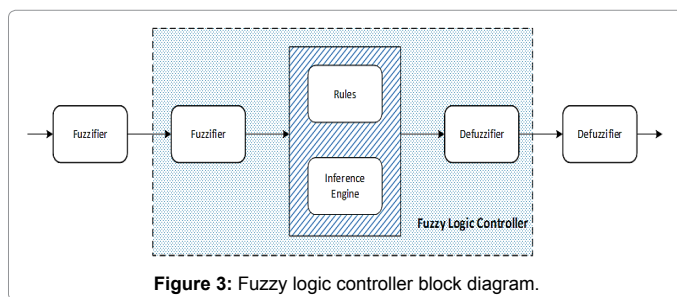


Figure 3: Fuzzy logic controller block diagram.

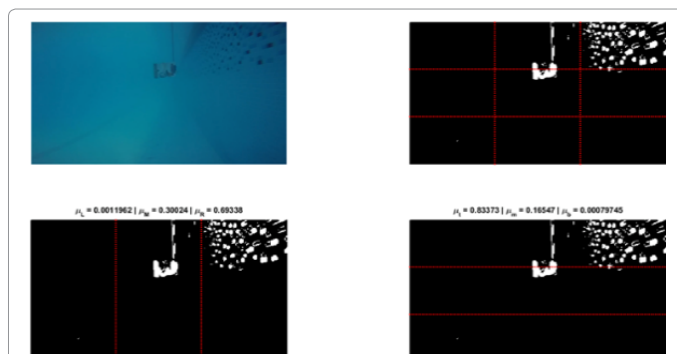


Figure 4: An example of image scoring system. The original image (Top left) is subdivided into six sections (Top right), i.e., three vertical bands (Bottom left) and three horizontal bands (Bottom right).

control commands reduced. The input and output membership functions of the fuzzy decision system are shown in Figures 5 and 6 respectively.

Note that the above membership functions were set after performing numerous tests and analyzing their performance on different images.

Similarly, there is another fuzzy controller to consider the difference between horizontal upper and lower sections and the middle one; the logic function is similar to FLC in DC engines. After performing the fuzzy operation, the fuzzy system output represents the angle in which the servomotors should be positioned (Table 3).

Like the vertical fuzzy system, the rules of the fuzzy controller were designed to orientate the robot. Figure 7 shows the system input membership functions. It should be mentioned that multiplying the scoring with a fixed number (90) give a range between -90 and +90, and then, output angles between -60° and 60°. The output membership

		$\mu_{b-\mu m}$				
Parameters		PB	PS	Z	NS	NB
$\mu_{t-\mu m}$	PB	0°	0°	0°	-20°	- 60
	PS	0°	0°	0°	-40°	-60
	Z	0°	0°	0°	-40°	-60
	NS	+20°	+40°	+40°	+40°	-60
	NB	+60°	+60°	+60°	+60°	+60

Table 3: Input fuzzy decision system.

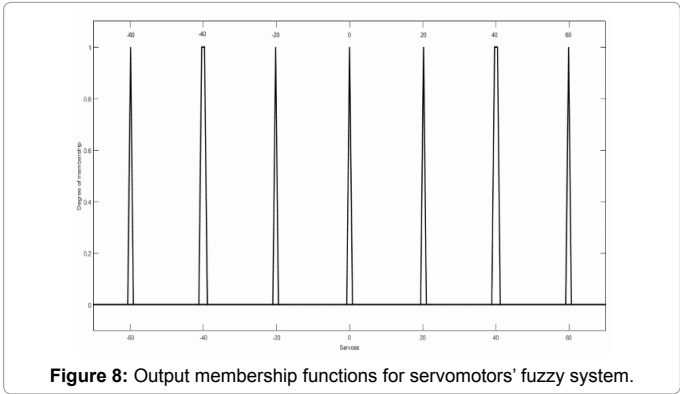


Figure 8: Output membership functions for servomotors' fuzzy system.

Type	Mamdani
And Method	Minimum
Implication Method	Minimum
Aggregation Method	Maximum
Defuzzification Method	Centroid

Table 4: Input fuzzy decision system.

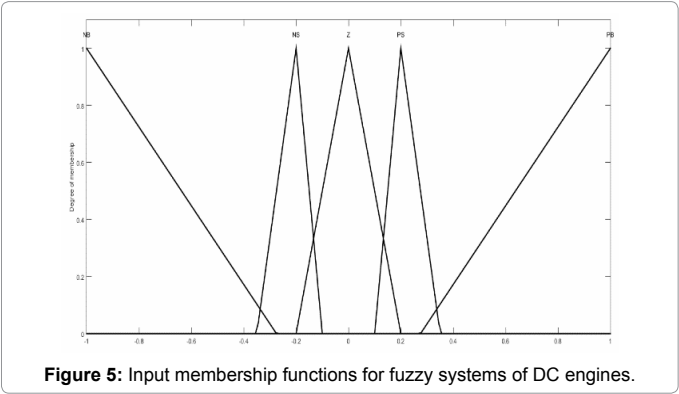


Figure 5: Input membership functions for fuzzy systems of DC engines.

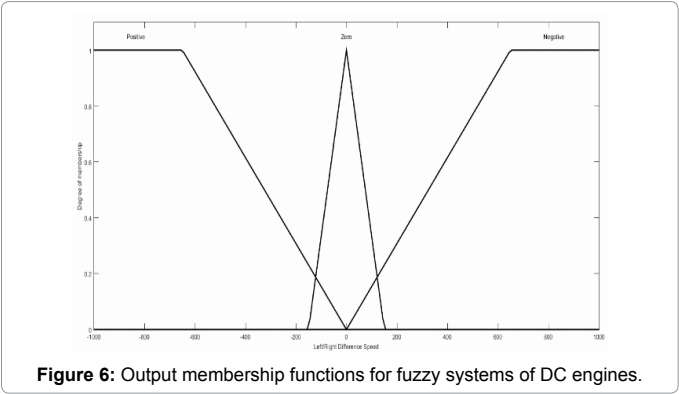


Figure 6: Output membership functions for fuzzy systems of DC engines.

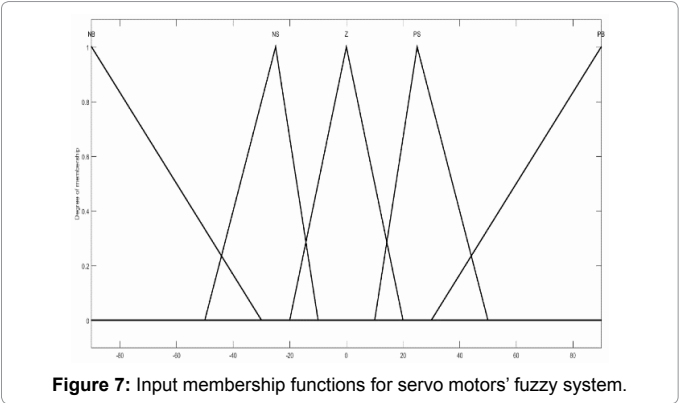


Figure 7: Input membership functions for servo motors' fuzzy system.

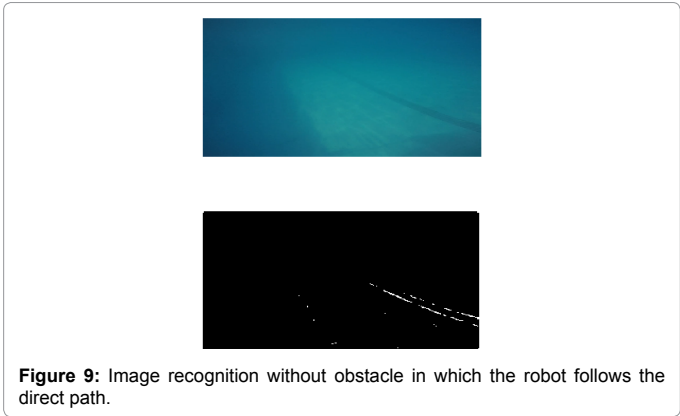


Figure 9: Image recognition without obstacle in which the robot follows the direct path.

functions are shown in Figure 8. Note that the fuzzy system output for servomotors is Singleton. Table 4 summarizes the methods used for the FLC.

Note also that in performing the fuzzy decisions, we added an extra constraint to limit the robot maximum depth to 8 m.

Experimental Tests

Test 1

We first validated the robot's ability to efficiently identify potential obstacles through the threshold we fixed. In fact, as shown in Figure 9, once the robot detected a shape, it processed the image and, as expected, did not recognize it as an obstacle. Therefore, the robot continues its optimal track determined by the compass sensor.

At $t = 1$ s, after passing the track the robot is diverted from its track by a -45° disturbance. At this moment the servo motors maintain their previous condition (zero degree angle) the left DC motor is turned off and the right side DC motor leads the robot to left by bow moment to compensate the robot angle difference with the track.

At $t = 5$ s, an external undesirable force was applied to force the robot body to roll $+30^\circ$, which means that the robot anchored left by 30° . In this situation, the servo motors took 90° angle and the right DC motor with a speed determined that the coefficient of roll angle with robot balanced mode, a force in the vertical direction applies to the right side of the robot so that the robot gains its balance.

After being stabilized, the robot returned to its initial track: the servo motors were rotated back to the “zero angle” position and the DC motors started to roll with their predefined default velocity. It is important to note that all the above commands were automatically managed by the proposed PID controller (Figure 10).

Test 2

In the second test, the robot successfully detected the obstacle depicted in Figure 11. It, therefore, took the decision to roll into right and went down with a gentle angle.

To this aim, the normal engine velocity (2000 rpm) was reduced and added to the right motor normal velocity, leading to the following quantities:

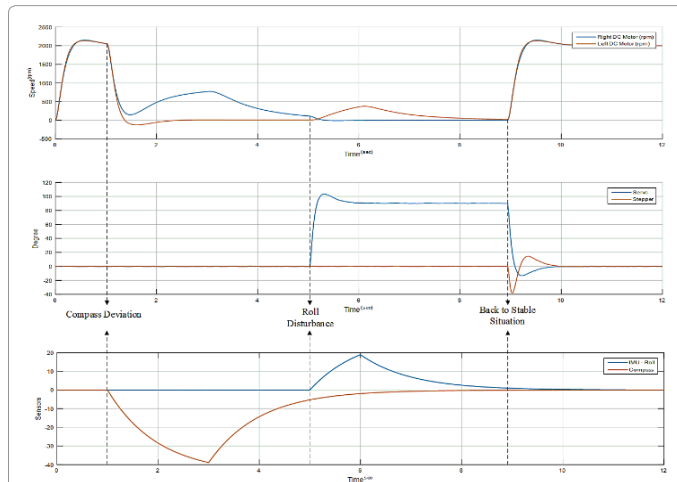


Figure 10: The engine performance in the first test.

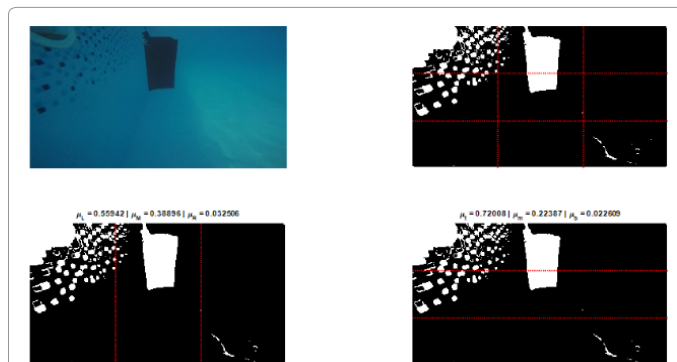


Figure 11: Obstacle detection in the second test. The original image (Top left) is subdivided into six sections (Top right), i.e., three vertical bands (Bottom left) and three horizontal bands (Bottom right).

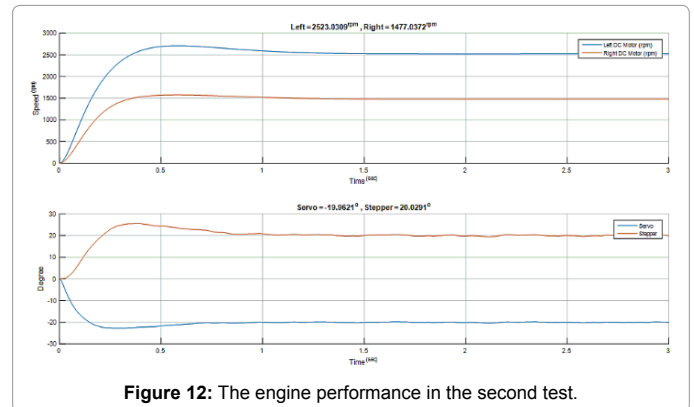


Figure 12: The engine performance in the second test.

- Left DC motor command velocity: $2000 - (-523) = 2523$ rpm.
- Right DC motor command velocity: $2000 + (-523) = 1477$ rpm.

Thus, because the moment created at the left side of the robot is higher than the right side, the robot bended to the right side. As for the first test, the entire sequence loop was controlled by the PID algorithm.

Note that after scoring, an optimal angle was determined and sent as command to the servomotors. In this case, a -20° correction was found to be adequate to balance the robot and thus, the angle of the stepper motor, following the opposite of the servo angle, was set to $+20^\circ$. From that, the robot's top went down and right to avoid the obstacle. The above process is described in Figures 12 and 13.

Test 3

In the third test, we detailed the robot process to bypass an obstacle. As shown in Figure 14, the robot detected an object, identified it as an obstacle, and took the appropriate decisions to successfully negotiate it.

After passing this obstacle the robot reaches a proper space for direct movement and performs its presupposed movement. After a few seconds, an external disturbance causes the robot to swing 30° degrees to its left side, we will see the control attempt to compensate disturbance. Then we will have a deviation from the path at 45° degrees to the left that will be compensated by the motors.

In this test, until the 11th frame, the fuzzy system will set commands to bypass the obstacle. These decisions were made based on the processes values of the image. Each row of the matrix below is the information of a frame.

μ_L	μ_M	μ_R	$\mu_L - \mu_M$	$\mu_R - \mu_M$
0.5812	0.4121	0.0067	0.1692	-0.4053
0.5809	0.4068	0.0123	0.1741	-0.3946
0.4368	0.5491	0.0140	-0.1123	-0.5351
0.3412	0.6495	0.0093	-0.3083	-0.6402
0.3552	0.6274	0.0174	-0.2722	-0.6099
0.6241	0.3534	0.0226	0.2707	-0.3308
0.8590	0.1319	0.0091	0.7271	-0.1228
0.9819	0.0041	0.0140	0.9778	0.0099
0.9817	0.0041	0.0142	0.9776	0.0102
0.9133	0.0114	0.0753	0.9663	0.0206
0.7436	0.0688	0.1876	0.9018	0.0639

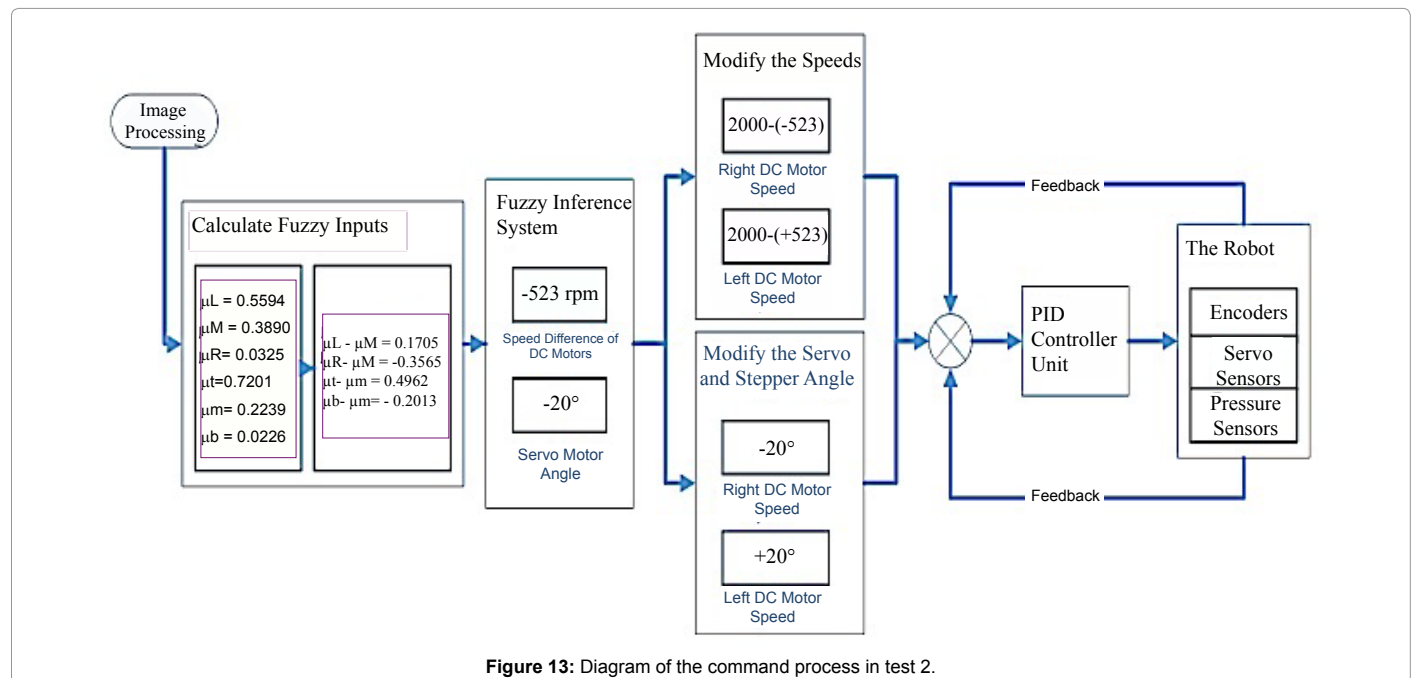


Figure 13: Diagram of the command process in test 2.

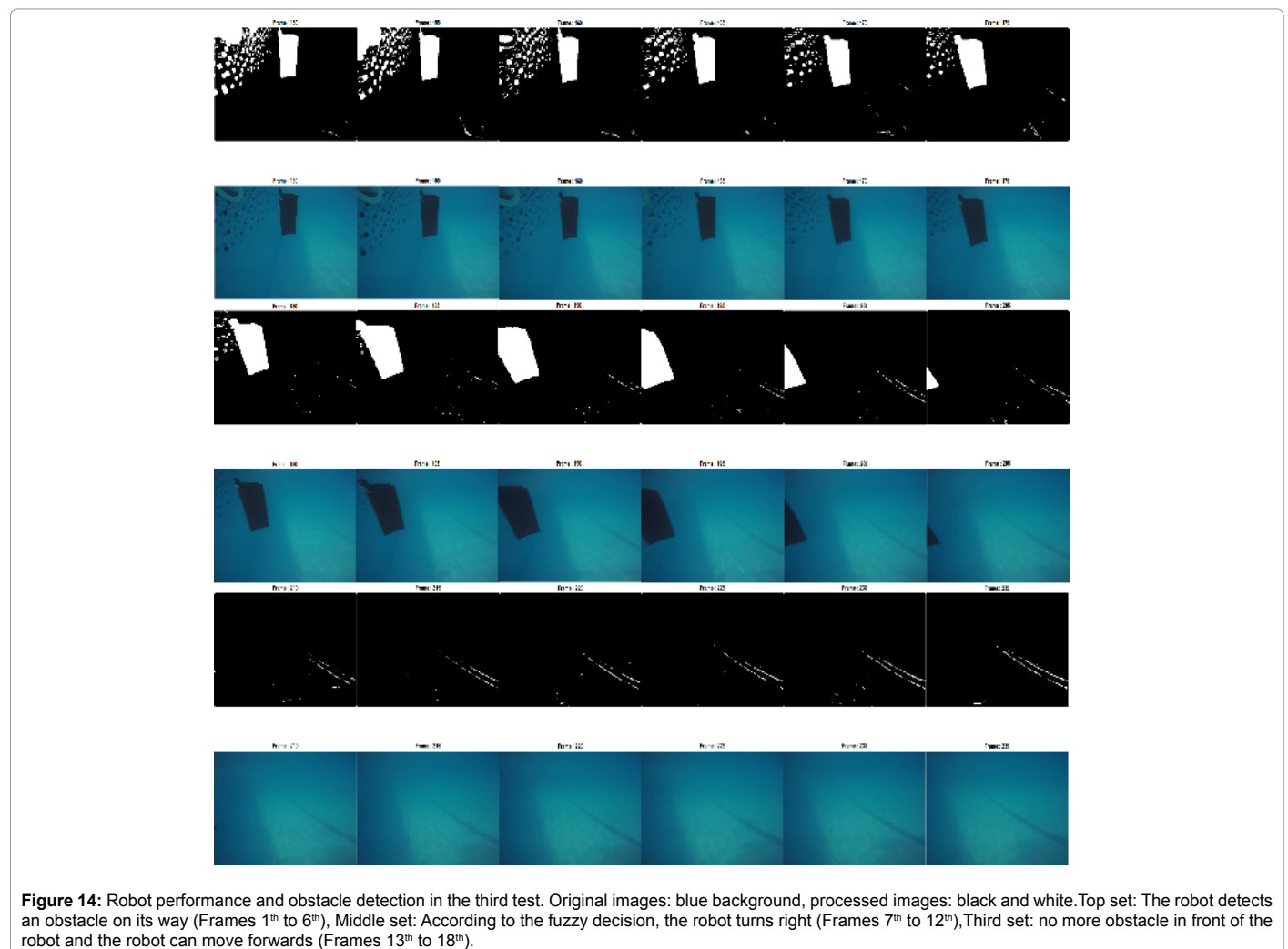


Figure 14: Robot performance and obstacle detection in the third test. Original images: blue background, processed images: black and white. Top set: The robot detects an obstacle on its way (Frames 1st to 6th), Middle set: According to the fuzzy decision, the robot turns right (Frames 7th to 12th), Third set: no more obstacle in front of the robot and the robot can move forwards (Frames 13th to 18th).

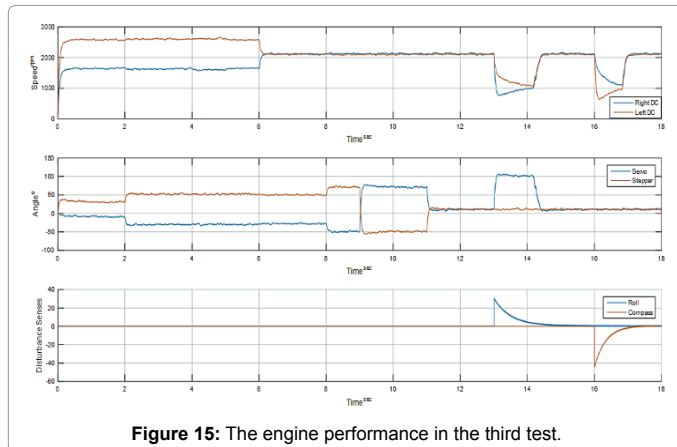


Figure 15: The engine performance in the third test.

$$\begin{bmatrix} \mu_t & \mu_m & \mu_b \\ 0.7754 & 0.2181 & 0.0065 \\ 0.7597 & 0.2284 & 0.0120 \\ 0.7094 & 0.2772 & 0.0134 \\ 0.6839 & 0.3072 & 0.0089 \\ 0.6366 & 0.3491 & 0.0144 \\ 0.6232 & 0.3595 & 0.0172 \\ 0.5745 & 0.4170 & 0.0085 \\ 0.5111 & 0.4807 & 0.0082 \\ 0.3713 & 0.6180 & 0.0107 \\ 0.2215 & 0.7394 & 0.0392 \\ 0.0203 & 0.9098 & 0.0699 \end{bmatrix} \rightarrow \begin{bmatrix} \mu_t - \mu_m & \mu_b - \mu_m \\ 0.5573 & -0.2116 \\ 0.5313 & -0.2164 \\ 0.4321 & -0.2638 \\ 0.3767 & -0.2983 \\ 0.2875 & -0.3347 \\ 0.2637 & -0.3423 \\ 0.1576 & -0.4085 \\ 0.0305 & -0.4724 \\ -0.2467 & -0.6073 \\ -0.5179 & -0.7002 \\ -0.8894 & -0.8398 \end{bmatrix}$$

Now the fuzzy decision inputs are ready and these vectors are presented to the fuzzy system. The following matrix is the commands presented to the engines (a row is formed for each frame).

$$\begin{bmatrix} \text{L/RDifferenceSpeed} & \text{ServoAngle} & \text{StepperAngle} \\ -533.7334 & -19.6000 & +19.6000 \\ -531.5049 & -19.6000 & +19.6000 \\ -560.8490 & -40.6000 & +40.6000 \\ -548.6814 & -40.6000 & +40.6000 \\ -575.7236 & -40.6000 & +40.6000 \\ -525.6532 & -40.6000 & +40.6000 \\ 0 & -39.2000 & +39.2000 \\ 0 & -39.2000 & +39.2000 \\ 0 & -60.2000 & +60.2000 \\ 0 & +60.2000 & -60.2000 \\ 0 & +60.2000 & -60.2000 \end{bmatrix}$$

The difference between the left and the right DC motors, servo angle and stepper angle (mass shifter) show the columns of this matrix. So, the robot behavior is designed based on the recent fuzzy decisions until the 11th second.

Until 13th seconds, the robot follows its track based on the

predetermined conditions. In the 13th second the IMU sensor reports the change of angle of 30° degrees in roll direction, so the robot decides to counteract this disturbance by changing the servo angles to 90° degrees with increasing the DC motors torque (The torque ratio of the difference between angle of the robot and the horizon).

Then in the 16th second a disturbance is created in Yaw direction and the robot is diverted to the right side. So, the robot decides to counteract this disturbance by changing the servo angles to zero level and the right DC motor has a relative higher torque according to the difference between angle of the robot and the horizon. Figure 15 shows the results of the above process.

Conclusion

One of the major benefits of the designed robot prototype is to save energy while being able to safely bypass obstacles. Indeed, with only two servo motors and a PID fuzzy-based controller, the system can provide an efficient and safe control of the robot's movements. In fact, even with six degrees of freedom, the robot can be proficiently managed with only two motors, avoiding complex designs.

References

1. Bingham BE, Wilson PR (2009) Design requirements for autonomous multivehicle surface-underwater operations. Marine Technol Society (MTS) J 43: 61-72.
2. Bingham B, Mindell D, Wilcox T, Bowen A (2006) Integrating precision relative positioning into JASON/MEDEA ROV operations. Marine Technol Society (MTS) J 40: 87-96.
3. Han KM, Choi HT (2011) Shape context based object recognition and tracking in underwater environment. IEEE Int. Geoscience and Remote Sensing Symp, Vancouver, Canada.
4. Choi HT (2010) New project and new underwater robot for new missions. International Conference on Advanced Mechatronics, Osaka, Japan.
5. MPU-9250 (2014) Product specification, Document Number: PS-MPU-9250A-01. InvenSense Inc.
6. HMC6343 (2014) Three-axis compass with algorithms, Honeywell's Magnetic Sensors, Honeywell Sensing, UK.
7. MS5803-14BA (2012) Miniature 14 bar Module, Measurement Specialties Inc.
8. Jebelli A, Yagoub MCE (2016) Efficient robot vision system for underwater object tracking. ICCSSE, Singapore.
9. Kemal Ari K, Asal FT, Cogun M (2012) PI, PD, and PID controllers: Project Report, Middle East Technical University.
10. Åström KJ (2002) Control system design, Lecture Notes, University of California.
11. Vervoort JHAM (2008) Modeling and control of an unmanned underwater vehicle, Master's Report, University of Canterbury, Christchurch, New Zealand.
12. Nag A, Patel SS, Akbar SA (2013) Fuzzy logic based depth control of an autonomous underwater vehicle. Automation, computing, communication, control and compressed sensing (iMac4s), International Multi-Conference p: 117-123.
13. Tamir DE, Rishe ND, Kandel A (2015) Fifty years of fuzzy logic and its applications. Springer, Berlin.
14. Zadeh LA (1996) Fuzzy logic computing with words. IEEE Trans. Fuzzy Systems 4: 103-111.
15. Banks W, Hayward G (2008) Fuzzy logic in embedded microcomputers and control systems. Byte Craft Ltd, Proof Theory for Fuzzy Logics, Springer, Berlin.