IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN (P): 2347-4599 ISSN (E): 2321-8843
Vol. 4, Issue 5, May 2016, 35-46

AUTONOMOUS WIRELESS UNDERWATER SURVEILLANCE VEHICLE CONTROL, HARDWARE-SOFTWARE CO-DESIGN FOR FISHERIES APPLICATIONS

ADENIYI, D.A., WEI, Z. & YANG, Y

Department of Computer Sc & Technology, College of Information Sc. & Engineering, Ocean University of China, Qingdao, Shandong, China

ABSTRACT

Navigation technology is considered as a fundamental problem in robotic technology. In this paper, we present hardware, software and algorithm design of a wireless control system for Autonomous Underwater Surveillance Vehicle (AUSV). We carefully designed and developed a wireless hardware control system for AUSV movement to assist fisheries operators in the collection and transmission of image data. We also developed software using C#(C-Sharp) programming language to control the movement of the AUSV. We, as well, proposed the adoption of a Hybrid Grid Map Based Genetic (HGMBG) path planning algorithm for finding an efficient optimal path in the grid map for the proposed AUSV. Extensive performance simulation results show that our control system is able to find optima path within a short time in similar environment with existing control system in almost all the simulation cases. Our system can, as well, provide an efficient and cost effective AUSV surveillance control operation consistently in a shallow water environment such as fisheries applications.

KEYWORDS: Wireless, Control System, Programming, Underwater, Surveillance, Fisheries Application

INTRODUCTION

Mobile underwater surveillance vehicle is a combined research of artificial intelligence and robotics. Autonomous underwater surveillance vehicles (AUSV) have attracted a rapidly growing interest from researcher during the last few years. This is due to the fact that they are easy to deploy, easy to manage with little infrastructures requirement. AUSV can be applied to a wide range of aspects, such as fisheries pool surveillance, observation of underwater climate, water pollution tracking, etc. [1].

The general problem with many existing AUSV is their ability to see obstacles and generate optimum path so as to avoid them in order to reach a particular position and orientation. This is called maneuvering planning [2]. In order to achieve performance efficiency of AUSV in term of time, distance, cost and complexity, several hardwares, softwares and path planning approaches have been introduced. The approaches are based on the environment, type of sensor, the AUSV capability, etc. [2].

In this paper, we propose the design, construction and implementation of a wireless control system for an AUSV for fisheries pools ecological system survey activities. We adopted two approaches namely: hardware and software. The proposed system will be able to observe fishery's pool temperature, capture aquatic objects' images data to be able to observe growth, or their bleaching state or to detect sign of spawning and transmit same to human operator for analysis.

The contribution of this paper is three-fold. Firstly, we design and construct wireless electronic circuit hardware to control the movement of the AUSV. The circuit is constructed using opto-coupler (solid state switch), transistors, diodes, relays resistors, 25 pins D-connector, LPT1 communication port to interface and establish communication between the computer system and the AUSV. The umbilical ethernet cable is used as transmission media between the control system and the AUSV. The AUSV unit is equipped with a USB camera for continuous underwater image recording and transmission, a USB temperature sensor for temperature measurement. The AUSV is also equipped with USB acceleration sensor for sensor unit posture detection. Secondly, we develop a program using C# (popularly called C-Sharp) programming language as a platform for controlling the AUSV movement underwater. Thirdly, we used a new hybrid Grid Map Based Genetic (GMBG) path planning algorithm to implement the designed control system. The GMBG algorithm is an improvement on the traditional genetic algorithm. This is done to find an efficient optima path in the grid map for the proposed AUSV.

The detailed description of the design, construction and implementation of the wireless electronic control hardware for the AUSV is presented. We also present the detailed description of the C# program code used as platform for controlling the AUSV movement and for the implementation of the proposed GMBG algorithm. Finally, a thorough presentation of the experimental result and implementation details will be carried out.

RELATED WORK

This section deals with review of related work pertinent to this study, pointing out similarities and differences between our work and other previous and related works. The review is organized as follow:

Overview of Related Underwater Vehicle Design

Survey and observation of deep sea can be dated back to the invention of Bathyscaphe by Professor Auguste Piccard in 1948. Since then various underwater equipments such as manned submersible, unmanned underwater vehicle, etc, has been developed [3]. Hyakudome [3], In his work designed an Autonomous underwater vehicle. He proposed a smart control system for the AUV. He as well, proposed the use acoustic telemetry for communication. However, his work did not explain the software required to drive the hardware and how the hardware works with the required software. Burkardt et al. [4], in the design and implementation of their Rangnarok AUV, designed an actuator control board which controls the servo used to turn the steering wheel. They also designed thruster control board that is powered through custom H-bridge configurations and their speed set by the computer over serial lines. The software on the vehicle is written in C/C++ and python run on GNU/Linux operating system. But in this study, the program is written in C# and run on the popular windows operating system to enhance the robustness of the application.

In the work of Wang et al [5], a low –cost unmanned underwater vehicle was designed for shallow water operation. An embedded central processing unit with microprocessors, FPGAs or small desktop PC is used for accessing sensor, processing data and setting control output such as motor speed. Their system features low-cost and wide potential use for normal shallow water task. Our work shares essentially the same goal of shallow water and low cost. But in our work, effort is concentrated on the design and implementation of hardware and software for the propulsion and steering of the vehicle, to provide multi directional movement for the vehicle, rather than the design of the entire AUV system.

Recently, underwater wireless sensor network and communication have attracted a rapidly growing interest from

researchers. Some algorithms, wireless sensors and other hardware have been proposed by [6, 7, 8, 9, 10], in order to support the operation of the underwater vehicle. Lv et al [11], proposed the use of acoustic channel access method for dense mobile underwater sensor networks. Lee and Kim [12] proposed two fast retransmit technique to overcome ACK indiscretion problems. In [1, 13], orthogonal regression based multihop localization algorithm and energy efficient routing algorithm was developed for underwater wireless sensor networks.

Designing an efficient path planning algorithm is essential issue in mobile robots navigation since path quality influences the efficiency of the entire application. To this effect several path planning algorithms are proposed by [2, 14, 15, 16, 17], these includes Genetic algorithm, Bug and point bug algorithms. In the literature, all these previous works limit their studies to either underwater vehicle design, path planning algorithm design or wireless sensor networking and communication in a large and deep sea environment. In this work we focus on the design and implementation of efficient and low cost hardware and software for the control of a model underwater surveillance vehicle. This is to be used in a shallow water fisheries operation with depth of less than 20 meters. In the implementation we will adopt an improved version of some of the proposed algorithm and sensor devices being understudied.

METHODOLOGY

This section describes the detailed design, construction and implementation of both the hardware, software and the adopted GMBG path planning algorithm for the control of the AUSV. It showcases how a wireless control system for an AUSV system is developed to assist in the collection and transmission of image data in fisheries applications. The system will provide efficient and cost effective AUSV operations for fisheries applications.

HARDWARE DESIGN

The Hardware consists of the improvised/model AUSV and the control circuit built up of transistors, diodes, relays, resistor opto – coupler (solid state switch) and 25 Pin D – connector for communicating between the AUSV and the Personal Computer via a socket outlet incorporated into the interfacing circuit. Wireless technology was employed using umbilical ethernet cable as a means of communication between the AUSV and the personal computer. The umbilical ethernet cable signal can cover a distance of more than 500 meters and can transmit data of 8 byte per second between the PC and the AUSV.

Whenever the relay is switch on from the PC, current flows into the socket which in turn causes the AUSV to take action based on which of the Pins is turned on.

Design of the Circuit and Circuit Diagram

In the construction of any system of this nature the first approach is to design and analyze the components to be used. This is achieved by grouping the design processes into three stages viz: 1.the coupling stage, 2. the switching stage, and 3.the rectifier stage [18].

- **The Coupling Stage**

This stage is made up of opto-coupler, photo diode or transistors to electrically isolate the interfacing circuit from the personal computer and resistor used to limit current flow into the opto-coupler.

- **The Switching Stage**

The stage is concerned with switching of the relay. Current flows through the resistor to turn on the transistor. The Diode is connected in parallel to the relay to protect it when the transistor is not conducting i.e. to prevent current flow in reverse direction.

- **The Rectification Stage**

This provides DC supply to the interfacing circuit, and as well rectifies the AC supply of 240 v to 12v. A 240-12V step down is used through a bridge diode to rectify the supply. Capacitor is connected to provide smoothing of the supply voltage. A 6v regulate is placed along the load so as to regulate the input voltage at 6v to the interfacing circuit. Figures 1 and 2 shows the interfacing sender and receiver circuit diagram respectively. Figure 3 shows the designed control circuit enclosed in a PVC box.

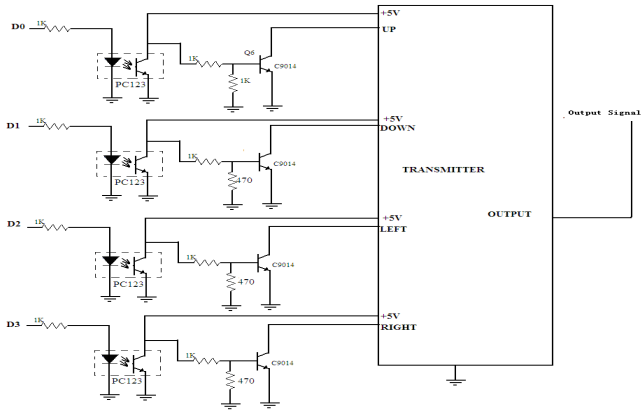

Figure 1: The Interfacing (Sender) Circuit Diagram

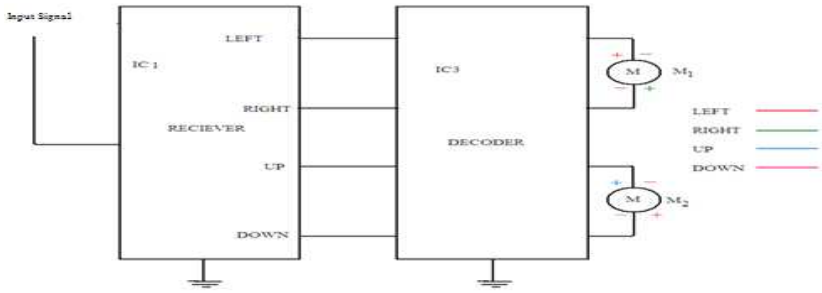

Figure 2: The Receiver Circuit Diagram

Safety Precautions

To achieve the workability of the circuit certain precautionary measures were taken which includes:

- Testing of the circuit to be sure it is free from short and opened circuits.
- Soldering of the components was done carefully so that the heat does not damage the components and the board.
- The program was carefully tested and debugged to reduce errors.

Source of Errors

Any errors found in the circuit might be due to aging of the component, improper reading from the multi-meter i.e. parallax error or improper analysis of the circuits [18].

SOFTWARE DESIGN

The software part of this research work is a program in high level language using C sharp (C#) programming language which handles the control of signal to the interfacing circuit. Options are presented to the operator in visual form i.e. graphical display of various working options depending on which action you want the AUSV to take at a particular time [19, 20].

In designing the program pseudo code and flowchart was drawn to serve as a guide to the coding of the program.

Pseudo –Code

- Initialize the LPT Port by sending the value, 0 to the ports so as to set all the registers to bit 0 i.e. switch off all devices.
- Checks for device selected, by looping in a timer, then send the appropriate value to the port to turn ON or OFF.
- If the device status is OK, set timer to switch signal ‘ON’ or ‘OFF’ using two text boxes for each device.
- Save and exit port address, appliance name, ON time and OFF time for each appliance by using file system object to save to a file and overwrite existing file.

In controlling the appliance values are sent from the software to the LPT port to turn ON or OFF the pins.

For value of 1, pin D1 is ON i.e. the AUSV move forward

- For value of 2, pin D2 is ON i.e. the AUSV move backward
- For value of 5, pins D1 and D3 are ON i.e. to turn left, forward
- For value of 9, pins D1 and D4 are On i.e. to turn right, forward
- For value of 6, pins D2 and D3 are On i.e. to turn left, backward
- For value of 10, pins D2 and D4 are On i.e. to turn right, backward
- For value of 0, pins D1, D2, D3, D4 are OFF i.e. to stop the AUSV.

Programming Fundamental

C – Sharp (C#) is an object oriented programming language, object actually help to make programming easier than ever before. Once one understands a few basic concepts, every other thing becomes easier. After creating the interface for the application using forms and controls, next is to write code that determine the applications behavior [19, 20].

Structure of C Sharp (C #) Application

As earlier stated, C# is an object oriented program and it is event driven. Developing a C# program can be approached by following three basic steps which includes: planning the interface, designing the user interface,

programming (Coding) the object controls in the form.

- **Planning The Interface: -** This involved planning how the interface will look like, the controls needed in the interface, the functions and behavior of each control. These have to be Pre-determined by the programmer before starting the work.
- **Designing The User Interface: -** Different forms and controls needed for the project are designed and their properties set for smooth running of the program.
- **Programming the Object Controls in the Form: -** This involves writing of codes for the objects in the form and even the form itself. Because C# application is based on objects, the structure of its code models its physical representation on the screen. The objects contain data and code. The form on the screen is a representation of the properties that define appearance and behavior of each form in an application. There is a related form module (with extension.frm) that contains its code.

The Code Modules

Code in C# is stored in modules and they are of three kinds, viz. the form, standard and class modules. Simple application can consist of just a single form and all the code resides in the form module. As the application gets large additional forms can be added.

- **The Form Modules:** The form module serves as foundation of most C# application. They usually have.frm file name extension. They contain procedures that handle events, general procedure and form- level declaration of variables, constants, types and external procedures.
- **The Standard Modules:** These serve as container for procedures and declarations commonly accessed by the modules within the application. They usually have.bas filename extension. Standard modules can be reused in many different applications.
- **The Class Modules: -** The class modules are the foundation of object oriented programming in C#.

Code is written in class modules to create new object which includes customized properties and methods. Actually forms are class modules that can have controls placed on them and can display form window, the class modules usually have.cls filename extension. Figure 3 shows the screen shot of the C# control system application, Figure 4 shows the part of C# code developed for the control system. The complete code is available on request.

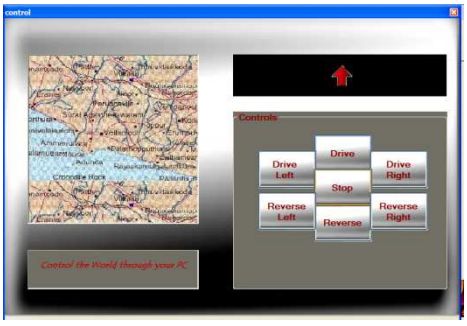

Figure 3: The Screen Shot of the C# Control System Application

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;
namespace Robotic
{  public partial class main : Form{
        public main(){
            InitializeComponent();
            axHwinterface1.OutPort(888,0);    }
        private void btnCar_Click(object sender, EventArgs e) {
       Form control = new control(); control.Show() ;} using System;
using System.Collections.Generic;
using System.Windows.Forms;
namespace Robotic
{   static class Program
    {     /// <summary>
        /// The main entry point for the application.
              [STAThread]
        static void Main()
        {  Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new main()); } } }
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;
namespace Robotic
{ public partial class control : Form
    {   public control()
        {   InitializeComponent();
            axHwinterface1.OutPort(888,  0); }
        private void IO(int val)
        { axHwinterface1.OutPort(888,(short)(val));    }
private void button1_Click(object sender, EventArgs e)
        { //drive
            IO(1);
            pictureBox1.Image =
global::Robotic.Properties.Resources.forwd;    }
        private void button2_Click(object sender, EventArgs e)
        {  //reverse
            IO(2);
            pictureBox1.Image =
global::Robotic.Properties.Resources.right;  }
        private void button4_Click(object sender, EventArgs e)
        {  //drive left
            IO(5);
            pictureBox1.Image = global::Robotic.Properties.Resources.left;  }
        private void button3_Click(object sender, EventArgs e)
        { //drive right
            IO(9);
            pictureBox1.Image =
global::Robotic.Properties.Resources.rightt;  }
        private void button6_Click(object sender, EventArgs e)
        {  //reverse right
            IO(10);
            pictureBox1.Image =
global::Robotic.Properties.Resources.RRight;}
        private void button7_Click(object sender, EventArgs e)
        {  //reverse left
            IO(6);
            pictureBox1.Image =
global::Robotic.Properties.Resources.Rleft;  }
        private void button5_Click(object sender, EventArgs e)
        { //stop  IO(0);
pictureBox1.Image = global::Robotic .Properties.Resources.forwd;  }  } }
{ public partial class main : Form
    {  public main(); { InitializeComponent();
            axHwinterface1.OutPort(888,0);  }
        private void btnCar_Click(object sender, EventArgs e)
        { Form control = new control(); control.Show();    }
   private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        { this.Close(); }
```

Figure 4: Part of C# Code Developed for the Control System

The GMBG Path Planning Algorithm

In the literature, the environment and the robots' knowledge about the environment are key factors in path planning problem. The environment can be static or dynamic [14]. The robot's knowledge about the environment can either be complete or incomplete. If the robot has complete knowledge about the environment, the problem is said to be global path planning, otherwise it is classified as local path planning [14]. In this paper, we adopted the global path planning in a dynamic grid environment due to the nature of many fisheries pool. In solving robotic path planning problems, a number of met heuristics algorithms have been proposed, such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), bees' algorithm, etc.

The Genetic Algorithm (GA)

The traditional Genetic algorithm has been used extensively in solving robotic path planning problem. In Genetic Algorithm (GA), the path population is initialized; series of dots are used to express the path of individuals. The path of individual coding is equivalent to a chromosome in biogenetic manipulation. The chromosome is selected, copied, crossover, mutation to genetic operations. After a number of operations, the evolution immediately stops. Finally, the current optimization of the individual is output [16]. Detailed description of the GA path planning algorithm is presented in [2, 14, 15 and 16].

The Grid Map Based Genetic (GMBG) Path Planning Algorithm

The GMBG is a hybrid algorithm. It is an improvement on the traditional genetic algorithm. In this work, we represent the fishery pool environment with grid map. The environment is partitioned into cells of equal size forming a two dimensional matrix. Each cell is numbered; starting with 0, for the top left cell, next cell to the right is numbered 1, next 2, 3, and so on. The grid based environment map is shown in figure 5. The shaded cells indicate the obstacle cells. The AUSV can move from one free cell to another, horizontally, vertically or diagonally. The AUSV has eight possible moves from each cell. In this study, we assume that all obstacles are dynamic and not known in advance. We represent free cell with "0" and obstacle cell with "144", in a 12 by 12 cell grid based environment. A feasible solution is defined by path transverse from the initial cell to the goal cell, through a number of free cells. The proposed adopted GMBG path planning algorithm is shown in figure 6

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
12	13	14	15	16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71
72	73	74	75	76	77	78	79	80	81	82	83
84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107
108	109	110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129	130	131
132	133	134	135	136	137	138	139	140	141	142	143

Figure 5: The Grid Based Environment

1. Perform parameters initialization
2. Generate randomly the initial population or set of feasible path(cells) on the map
3. Compute a route that goes through all the cells
4. While (Not goal point) do
5. Select the next cell to be visited
6. If (next cell contain obstacle) then go to step 5
7. Else
8. If robot rotation <= 360 then
9. Robot rotate right or left according to next cell to be visited
10. If sudden point
11. If(180 degree rotation) then
12. Ignore reading /* To prevent the robot from returning to previous point */
13. Else
14. Get distance from current sudden point to the next sudden point
15. Get angle of robot rotation
16. Travel to the approximate location to the selected cell according to distance and rotation angle
17. Reset rotation
18. End if
19. End if
20. End if
21. End while

Figure 6: The GMBG Path Planning Algorithm

EXPERIMENTAL DESIGN

Building an underwater vehicle is out of the scope of this work. To experiment our designed control system we adopted an improvised (prototype/model) underwater vehicle. The prototype has a complete set of components including vehicle hull, propulsion, depth control, sensors and electronic battery and communications. The experimental setting is similar to that of observation class remotely operated vehicles (ROVs). An umbilical ethernet cable is being used for remote link between the UV and the external control computer at the development phase. Figure 7 shows the model SUV and the control system. Note that in an actual final developed application, in-water test or autonomous full-mission testing, the SUV will be acting autonomously and this umbilical will no longer be required. It is expected that the developed circuit and software be incorporated in to the SUV control board for autonomous operation. The custom software is written in C#. The software stack is built upon the windows operating system, windows XP or higher specification. The software is powered by an Intel dual core CPU 1017UU@ 1.60GHZ, 1.60GHZ processor on mini-ITX motherboard. Figure 7 shows the complete architecture of the designed control system with the model underwater Vehicle.

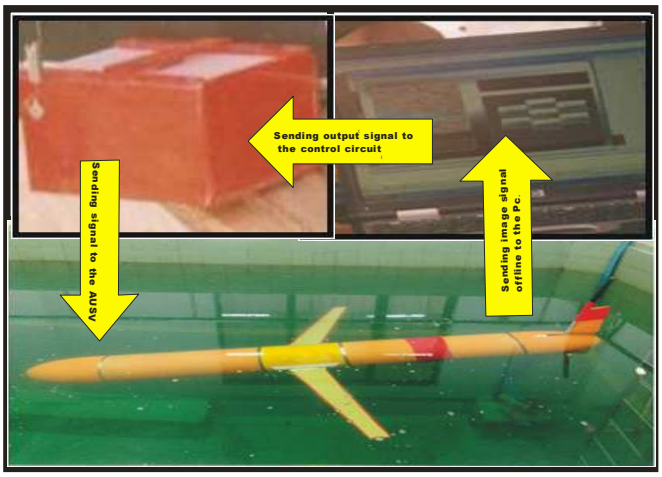

Figure 7: The Complete Architecture of the Designed Control System with the Model Underwater Vehicle

SUMMARY OF FINDINGS

In this study, we started the research and development of the demonstration of a control hardware and software for ASUV operation. For the ASUV, the working schedule is preset on the computer before starting observations. The schedule is made up of navigation course and procedure of observation devices. When the vehicle notice some obstacle along its programmed course, it takes avoidance action by either turning left, right or moves back based on availability of free cells on the grid map. This is made possible through our C# application and the constructed control hardware developed for the purpose.

CONCLUSIONS

A control system has been successfully designed for a model ASUV for shallow water operation such as fishery applications Low-Cost materials are used to make the system affordable for medium and large scale fishery operations. The designed system features a wide potential use for normal shallow water with a working depth of less than 20 meters. It also has the potential to control the model ASUV movement in eight different directions based on available cell on the grid map. Performance simulation of operation of our system shows that our control system is capable of finding optimal path within a short time. It can as well provide an efficient and cost effective ASUV surveillance operation consistently in a shallow water environment.

RECOMMENDATION FOR FUTURE WORK

We are of opinion that this study could be taken much further by improving many elemental technologies used for the control system, navigation system and so on. The software module could also be improved upon to accommodate more functions.

REFERENCES

1. Ren, Y., Zhong, J., Huang, J., Song, Y., Xin, X., Yu, N. and Feng, R. (2014). Orthogonal regression based multihop localization algorithm for large-scale underwater wireless sensor networks. International journal of distributed sensor networks. dx.doi.org/10.1155/2014/596082.

2. Buniyamin, N., Wan, N., Shariff, N. and Mohamad, Z. (2011). A simple local path planning algorithm for autonomous mobile robots. International journal of Systems Applications, Engineering & Development. Issue 2, Vol. 5, pp. 151-159.

3. Hyakudome, T. (2011). Design of autonomous underwater vehicle. International Journal of Advance Robotic System. Vol. 8, No. 1, pp. 122-130.

4. Burkardt, M., Barron, L., Brook, T., Bunney, M., Chang, C., Chayanupatkul, P., Dear, P., Du, X., Esslinger, K., Franklin, S., Fu, Z., Gong, E., Gu, P., Hamada, M., Heidel, J., Henderson, K., Jatusiripitak, N., Kambo, J., Kaplan, M., Kelly, D. Lee, M., Lee, K., Lei, K., Levy, N., Luo, Y., Mahoney, M., Malcoci, A., Ng, L., Porter, Z., Ross, M., Sideris, C., Sim, J., Soohoo, H., Spitzer, A., Szoka, E., Thiel, E., Ting, A., Tome, S. Tseng, P., Turkmen, B., Vazquez, A., Wijaya, A., Xing, C. and Zhang, R. (2013). Design and implementation of the Ragnarok AUV. Cornell University autonomous underwater vehicle. Cornell university Autonomous underwater vehicle. Downloaded on 15th November, 2015.

5. Wang, W.H., Chen, X. Q., Marbug, A. Chase, J.G. and Hann, C.E. (2009). Design of low-cost unmanned underwater vehicle for shallow waters. International Journal of Advanced Mechatronic System. doi.10.1504/IJAMECHS,2009.023202.

6. Li, Z., Yao, N. and Gao, Q. (2014). Relative distance based forwarding protocol for underwater wireless network. International journal of distributed sensor networks. dx.doi.org/10.1155/2014/173089.

7. Kim, S., Yoo, Y. (2014). SLSMP; Time synchronization and localization using sea water movement pattern in underwater wireless network. International journal of distributed sensor networks. dx.doi.org/10.1155/2014/172043.

8. Jung, J.W. and Kim, K.M.(2013). Optimizing of iterative turbo equalizer for underwater sensor communication. International journal of distributed sensor networks. dx.doi.org/10.1155/2013/ 129781.

9. Chun, S. and Kim, K. (2013). Passive acoustic source tracking using underwater distributed sensor. International journal of distributed sensor networks. dx.doi.org/10.1155/2013/723862.

10. Suzuki, T., Kato, K., Makihara, E., Kobayashi, T., Kono, H., Sawai, K., Kawabata, K., Tekemura, F., Isomura, N. and Yamashiro, H. (2014). Development of underwater monitoring wireless sensor network to support coral reef observation. International journal of distributed sensor networks. dx.doi.org/10.1155/2014/189643.

11. Lv, C., Wang, S., Tan, M. and Chen, L. (2014). UA-MAC: An underwater acoustic channel access method for dense mobile underwater sensor networks. International journal of distributed sensor networks. dx.doi.org/10.1155/2014/374028.

12. Lee, S. and Kim, D. (2014). Two fast retransmission techniques in UWSNs with ACK indiscretion problem. International journal of distributed sensor networks. dx.doi.org/10.1155/2014/160541.

13. Xu, M., Liu, G. and Wu, H. (2014). An Energy efficient routing algorithm for underwater wireless sensor networks inspired by ultrasonic frogs. International journal of distributed sensor networks. dx.doi.org/10.1155/2014/351520.

14. Chaari, I., Koubaa, A., Bennaceur, H. Ammar, A., Trigui, S., Tounsi, M., Shakshukif, E. and Youssef, H. (2014).On the Adequacy of Tabu Search for Global Robot Path Planning Problem in Grid Environments. Procedia Computer Science. "5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014)". doi: 10.1016/j.procs.2014.05.466.

15. Mathew, D., Peter, C., Iuliu, V. & Daniela, R. (2006). Data mulling over underwater wireless sensor networks using autonomous underwater vehicle. IEEE International Conference on Robotics and automation. pp. 20091-2098.

16. Xuan, Z., Bin, G., Peng, S. (2012). Improved genetic algorithm for dynamic path planning. International Journal of Information and Computer Science. Vol. 1. Issue 2, pp. 16-20. ISSN(online): 2161-5381.

17. Payeur, P. (2004). Improving robotic path planning efficiency with probabilistic virtual environment models. IEEE International Conference on Virtual environment, Human-Computer Interface and Measurement System. pp 13-18.

18. Boylested, R. L. and Nashelsky, L. (1996), Electronic Device and Circuit Theory, (6th ed.). New Jersey: Printice-Hall, Inc.

19. Bollinger, G. J. Duffie, A. N. (1988), Computer Control of Machine and Processes, Massachuselts: Adison – Wesley Publishing Company.

20. Findler N. V. and Meltzer B. (2004). Artificial Intelligence and Heuristic Programming. Edinburgh: Edinburgh University Press.