Modelling and Simulation of the Autonomous Underwater Vehicle (AUV) Robot

Muhammad D. Muhaiyuddin[a], **Zulhasnor Hassan**[a], **Chen Jia**[a], **Huimei Zhou**[a], **Rong Pan**[a] **and Amir M. Anvar**[a]

[a] *School of Mechanical Engineering*
The University of Adelaide
SA 5005, AUSTRALIA
Emails: muhammad.muhaiyuddin@student.adelaide.edu.au; zulhasnor.benhassan@student.adelaide.edu.au; chen.jia@student.adelaide.edu.au; amir.m.anvar@gmail.com

Abstract: Autonomous underwater vehicles (AUV) have been used widely in the oceanic applications for many purposes, such as maritime fishery study, the installation of underwater pipelines in the oil and gas industry as well as to support offshore engineering, and mine hunting and defence applications.

Projects in the development, manufacturing and testing of AUVs are complex, requiring a large investment of time, money and effort. Therefore, the simulation of the AUVs has been identified as being cost effective and time saving, as simulation is a vital step in understanding the behaviour and operation of the system before its actual deployment in the field.

A number of subsystems must be considered for building a successful AUV simulator, these are the Inertial Navigation System (INS), Machine Vision, Sonar, Communication (underwater and on surface), and the Simultaneous Localization and Mapping (SLAM) module systems. This paper discusses the methods and the outcomes of the feasibility study of the AUV simulator under investigation.

Keywords: *Autonomous Underwater Vehicle (AUV), simulator, simultaneous localization and mapping, SLAM, robotics*

1. INTRODUCTION

Research and development of the AUVs plays a significant role in modern robotics. AUVs have been used in many industries such as maritime, offshore engineering, undersea mining, military and oceanography [2]. The AUV consists of five important subsystems; the Inertial Navigation System (INS), Machine Vision, Sonar, Communication (underwater and on surface) and the Simultaneous Localization and Mapping (SLAM) module. The main objective for this project is to develop an AUV simulator that integrates these key AUV subsystems. To achieve this, various knowledge of each AUV subsystems are required. Moreover, the AUV subsystems need to be connected to each other in order for the AUV simulator to perform task similar to real life application. This paper discusses the AUV simulator and the concepts behind building this simulator.

The Inertial Navigation System (INS) is one of the most important subsystems in AUV(s). This subsystem is responsible in computing the AUV's underwater position by estimating its latest position based on the data received from inertial sensors such as the accelerometer and gyroscope [3]. Understanding of AUV's physical model is essential in producing an efficient navigational system within the AUV. The physical model of the AUV such as its mass, volume and area obviously have a huge impact on the AUV's motion [6].

The next section describes the Machine vision subsystem which is used to observe and record the surrounding environment using visual tools [1]. Simulation modelling of the vision system mainly involves stereo image processing and obstacle avoidance systems.

The Sonar Imaging section provides full explanations of the Ultrasonic wave that is used in Sonar subsystem. Time-of-flight (TOF) of an ultrasonic wave is used as a method for distance measurement based on the pulse-echo method [7]. By using the TOF and propagation velocity of ultrasonic wave, the distance of the object to the autonomous underwater robot system can be estimated.

The Modelling and Simulation of the communication subsystem was built by using specific toolboxes as provided by the MATLAB-Simulink toolset, such as the RF Blockset and the Communication System Toolbox. RF communications consist of a transmitter and receiver, used to convert the signal between the baseband signal and assigned channel signal in transmitters and receivers, respectively [4].

SLAM is one of the fundamental aspects that need to be considered for any intelligent autonomous robot which allows the robot to move without any direct predefined information and building a picture of the surrounding environment using all its integrated sensors. With this scheme, the trajectory motion of the robot can be calculated.

I. AUV MODELLING

A. Hardware Background

The design of the AUV simulator is based on an existing AUV test-bed (see Figure 1) [1]. The AUV system consists of 4 separate modules which are:

- Inertial Navigation Sensors – Thrusters, GPS receiver, compass, accelerometer, pressure sensor, Doppler Velocity Log (DVL), and altimeter
- Detection, Tracking and Identification Sensors – Sonar, and stereo vision camera.
- Communication system – Radio Frequency (RF) air to air modems, and underwater communication modems.
- Command and Control system – PC104 stack.

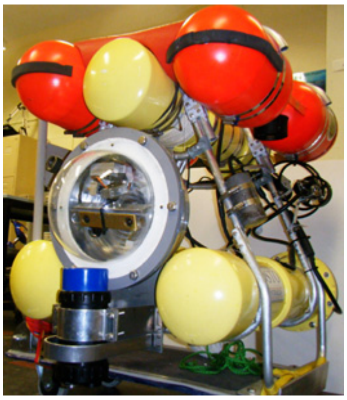


Figure 1. Real AUV model.

II. SIMULATION FRAMEWORK

The AUV simulation framework is constructed using the MATLAB-Simulink toolbox with the design architecture as shown in figure 2.

The design framework considers several subsystems which are which are discussed in more detail in the next sections.

III. SIMULATION RESULTS

1) Inertial Navigation System (INS)

The Inertial Navigation System (INS) subsystem plays a major role in the AUV's control system. This subsystem is responsible for computing the AUV's position underwater by estimating its latest position based on the data received from its inertial sensors such as accelerometer and gyroscope.

a) AUV physical model

Understanding of the AUV's physical model such as mass, volume and area of operation is essential in producing an efficient AUV navigational system. [6]. In this case, dynamic models that are related to the AUV's physical properties need to be generated in order to study the motion of the AUV [3]. From the figure 3, it can be seen that AUV's motion can be divided into two coordinate reference systems. The fixed central point is in x, y and z directions while the dynamic central point is in ξ, η and ς directions [6].

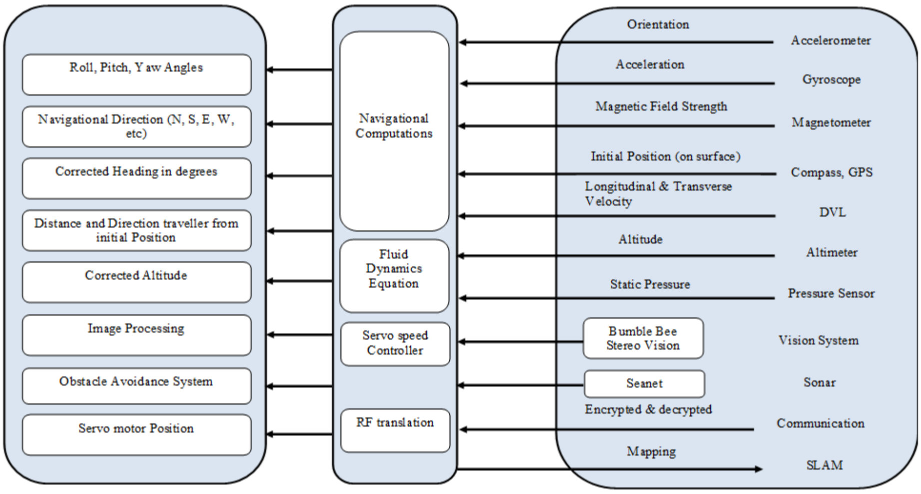


Figure 2. AUV simulation design framework.

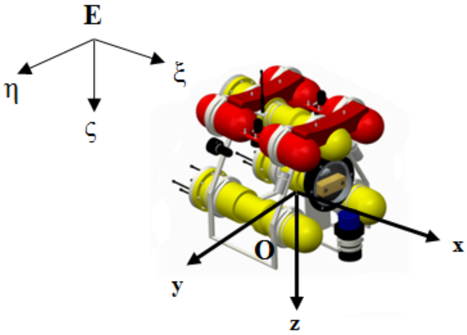
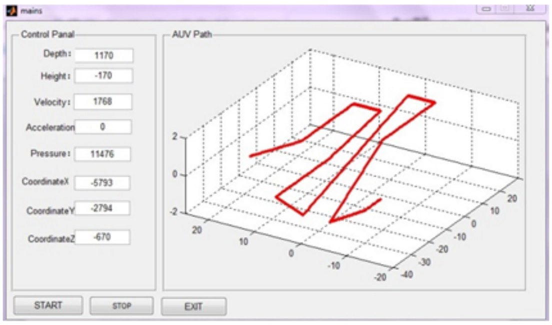


Figure 3: AUV Physical Model



Depth: 1170
Height: -170
Velocity: 1768
Acceleration 0
Pressure: 11476
CoordinateX -5793
CoordinateY -2794
CoordinateZ -670

Figure 4: INS Simulation result

In order to analyse the movement of the AUV underwater, there is a need to define two coordinate reference systems which is included with the earth fixed inertial and the AUV's body coordinate systems. As shown in figure 3 the inertial coordinate system is defined by ξ, η and ς, and the moving coordinate system is defined by the x, y and z components [6]. At the initial stage, the linearised to simplify motion of the AUV is easier to be modelled since the real AUV moves underwater with low speed. The centre of gravity of the AUV is located at the centre point of the AUV. Therefore, force, moment, speed and angular rate of the AUV are acting in three directions that can be expressed by the following formulas:

Force: $F = [X, Y, Z]^T$ (1)

Moment: $M = [K, M, N]^T$ (2)

Speed: $V = [u, v, w]^T$ (3)

Angular Rate: $\omega = [p, q, r]^T$ (4)

Based on these dynamic equations for the AUV, a simple mathematical model has been built using the MATLAB-Simulink toolbox thereby generating AUV simulation outputs.

From the figure 4, it can be seen that the AUV is able to produce output of the depth to the seafloor, height to the surface of water, moving velocity, pressure, acceleration, and current position in x, y, z coordinates.

2) Machine Vision

Machine vision is defined as the ability of the machine's sensor to observe and record the information of the surrounding environment using visual tools [1]. At the early development stages, the vision subsystem had been designed as a separate system and was used solely for capturing images and videos using a stereo vision system. This stereo vision system consists of two Fire-I digital cameras mounted in parallel. These cameras were calibrated for underwater applications and are capable of providing video at an uncompressed 640x480 resolution at 30 frames per second [7]. Currently, the vision system has been integrated with the all AUV systems. With the successful integration, several complex algorithms and schemes, such as image processing, obstacle avoidance and servo motor controller were able to be integrated with the vision system. Simulation of vision system involved stereo image processing and obstacle avoidance systems.

a) Stereo Image Processing System

The purpose of stereo image processing system is to create a three dimensional representation of the surrounding environment in the form of disparity images which can then be used to determine the location, distance and size of any obstacles that are potential threats to the AUV. In order to create a disparity image, the two offset images captured by the stereo camera were coordinated in order to calculate the disparity at each pixel within a neighbouring pixel of both images. There are several algorithms that have been developed to create an accurate disparity image as shown in figure 5.

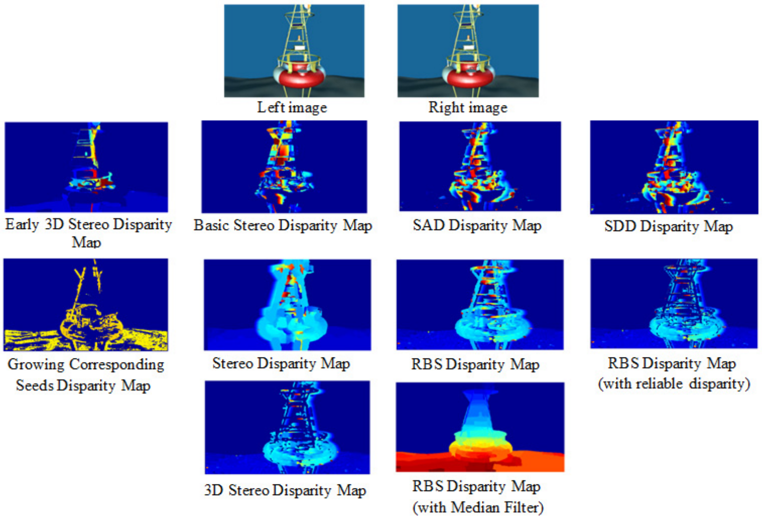

Left image
Right image
Early 3D Stereo Disparity Map
Basic Stereo Disparity Map
SAD Disparity Map
SDD Disparity Map
Growing Corresponding Seeds Disparity Map
Stereo Disparity Map
RBS Disparity Map
RBS Disparity Map (with reliable disparity)
3D Stereo Disparity Map
RBS Disparity Map (with Median Filter)

Figure 5. Stereo Disparity Images based on different algorithms.

b) *Obstacle Avoidance System*

The technique that was implemented for the object tracking is known as Optical flow estimation. With the optical flow estimation, the motion vectors in the video sequence can be estimated. The procedure involved in optical flow estimation is as follows: 1) Converting the video into binary format by using thresholding and morphological techniques, 2) Using Blob analysis block to locate the objects, 3) Creating bounding boxes around the target objects by using the Draw Shapes block. The following figure 6 shows the object tracking using the optical flow block model.

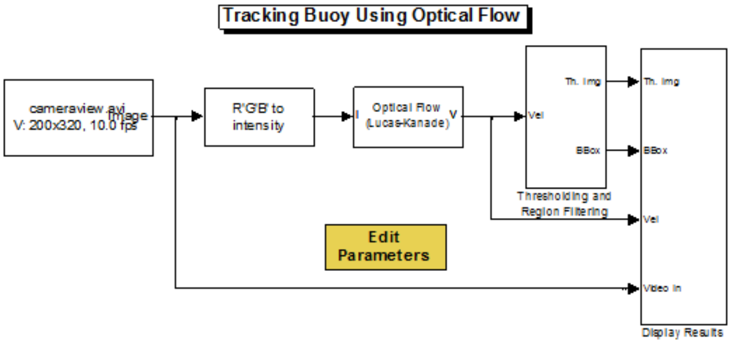


Figure 6. Object tracking Block Diagram.

3) Sonar Imaging

Recognizing the surrounding is important for all autonomous robotic systems. Sonar sensor is used in many autonomous mobile robots in order to measure the distance of objects for environmental recognition. Sound Navigation and Ranging (Sonar), is a technology that utilizes sound to navigate, communicate and detect visible objects within its operating range. Compared with light, sound has less attenuation and dispersion in water, thus making sonar sensor the best choice to be used underwater. Sonar operates by bouncing acoustic waves off matter. Using this principle, sonar is able to determine the distance of the obstacles in its surroundings by measuring the time for the reflected waves to return.

In this part, simulation of ultrasonic waves was demonstrated using the replicated signals from the active-sonar sensor to measure the distance of the surrounding objects to the AUV. The Sonar system improves the AUVs awareness of objects in its surrounding, thus enhancing the performance of the collision avoidance system.

a) Signal Processing

The distance of the surrounding objects to AUV can be calculated from cross-correlations function. The cross correlation function, C_1 (t) can be expressed as:

$$C_1(t) = \frac{\pi}{4} \cdot \frac{1}{N} \sum_{i=0}^{N-1} h_1(N-i) \cdot x_1(t-i) \qquad (5)$$

Where, $x_1(t)$ is a single-bit received signal, $h_1(i)$ is also single-bit reference signal, N is number of samples by digital comparator. The difference of cross-correlation can be defined as:

$$C_1(t) - C_1(t-1) = \frac{\pi}{4} \cdot \frac{1}{N} \sum_{i=0}^{N-1} h_1(N-i) \cdot x_1(t-i) - \frac{\pi}{4} \cdot \frac{1}{N} \sum_{i=0}^{N-1} h_1(N-i) \cdot x_1(t-i-1)$$

$$= \frac{\pi}{4} \cdot \frac{1}{N} [x_1(t) \cdot h_1(N) - x_1(t-N) \cdot h_1(N) + \sum_{i=1}^{N-1} \{h_1(N-i+1)\} \cdot x_1(t-1)] \qquad (6)$$

Value of $h_1(1)$ and $h_N(i)$ are 1 and -1. Therefore values of h_1 (N-i)-h_1(N-i+1) can be expressed as

$$h_1(N-i) - h_1(N-i+1) = \begin{cases} 2, & \cdots h_1(N-i) > h_1(N-i+1) \\ -2, & \cdots h_1(N-i) < h_1(N-i+1) \\ 0, & \cdots h_1(N-i) = h_1(N-i+1) \end{cases} \qquad (7)$$

Using these cross-correlation function, distance of the objects detect by the sonar can be estimated as

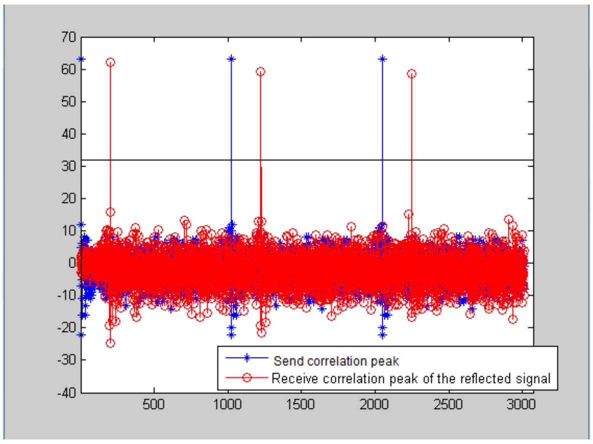
$$d = (TOF \cdot v)/(2 \cdot f_c) \qquad (8)$$

Where v is the velocity of the ultrasonic wave transmits by sonar, which is 1560 m/s (sound velocity underwater), fc, is the threshold frequency and TOF is Time-of-flight.

b) Simulation

In this section, the simulation involves running a simple ultrasonic wave that is reflected by an object at a specific distance. This simulation shows the distance of the object to the AUV and the alternative path (2D) that can be taken by the AUV to avoid a collision with the object. The simulation has again been build using the MATLAB-Simulink toolbox.

Figure 7 shows the output of the MATLAB-Simulink simulation. The simulator is intialised with the sonar time interval, TOF and threshold frequency. The peak of the blue line represents the signal has been transmit by sonar, whereas the red peak line shows the signal reflected by objects and received by the sonar system. The distance of the objects from the sonar has been calculated using equation (8), which the result can be seen in Figure 7.



Fugure 7: Signals Send and Receive by Sonar (MATLAB code).

During this process, the MATLAB simulation code has been built and tested using the MATLAB Simulink. The Simulink block diagram of the sonar image is presented in from Figure 8.

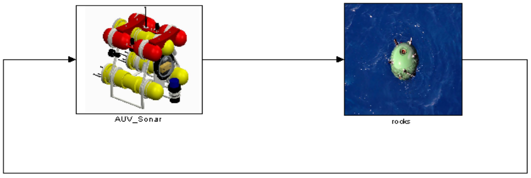


Figure 8. Simulink block diagram for sonar subsystem.

4) Simultaneous Localisation & Mapping (SLAM)

The Extend Karman Filter (EKF) SLAM is one of the most frequently used methods for intelligent robots. EKF-SLAM needs to combine sonar detection results to estimate the route. The motion sensor values can be updated in a timely manner reducing the error between actual trajectories and the simulated route.

The AUV state vector can be illustrated as, $X_v = [x\, y\, z\, \varphi\, u\, v\, w\, r]^T$ (9)

In this formula [*x y z*] show the AUV global position, and also [*u v w*] show the AUV motion in *x, y, z* direction separately. φ is the azimuth angle and *r* represents roll angle.

The AUV estimate vector can be illustrated as, $\hat{X}_v^- = [\hat{x}\, \hat{y}\, \hat{z}\, \hat{\varphi}\, \hat{u}\, \hat{v}\, \hat{w}\, \hat{r}]^T$ (10)

The AUV state covariance can be shown as, $P_v = \begin{bmatrix} \sigma^2_{xx} & \cdots & \sigma^2_{xr} \\ \vdots & \ddots & \vdots \\ \sigma^2_{rx} & \cdots & \sigma^2_{rr} \end{bmatrix}$ (11)

Sonar detecting data can be regarded as character points; all character points' play's an important role in mapping process. The covariance matrix of the map P$_m$ has characteristic information except the diagonal value. All values cover the map characteristic points more than once to improve the accuracy of the SLAM algorithm.

$$P_m = \begin{bmatrix} \sigma^2_{x_1 x_1} & \cdots & \sigma^2_{x_1 y_n} \\ \vdots & \ddots & \vdots \\ \sigma^2_{x_n y_n} & \cdots & \sigma^2_{y_n y_n} \end{bmatrix} \qquad (12)$$

The AUV state vector in absolute coordinate can be shown as,

$$\hat{X}_m^- = [\hat{x}_1\, \hat{y}_1 \,\cdots\, \hat{x}_n\, \hat{y}_n]^T \qquad (13)$$

In this case the Karman threshold value is,

$$K_k = P_k^- H_D^T (H_D P_k^- H_D^T + R_D)^{-1} \qquad (14)$$

Where, P_k^- is the variance of the next time step system; condition.

$$H_D = [H \quad 0_{3\times 2n}] \qquad (15)$$

$$R_D = \begin{bmatrix} \sigma^2_u & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_v & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_w & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_p & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_e \end{bmatrix} \qquad (16)$$

Therefore, the upgraded formula is presented as; $\hat{X}_k^+ - \hat{X}_k^- + K_k(Z_D(k) - \hat{X}_k^-)$
(17)

$$P_k = P_k^- (I - K_k H_D)^{-1} \qquad (18)$$

EKF-SLAM normally uses the motion sensor to measure the distance of each travelled segment. This figure would be compared with the preassigned segment that was supposed to travel to estimate error. Furthermore, the active-sonar system detects and measures the robot's position and its distances from any possible object(s) within the surrounding environment in real-time. In this case, through EKF-SLAM algorithm the optimal trajectory and map would be estimated [5].

The EKF-SLAM simulation (figure 9) models that the AUV following the existing scenario with the constant velocity. The AUV detects beacons which are set advance to estimate the trajectory. The AUV detects the beacons which are set in advance to estimate the trajectory.

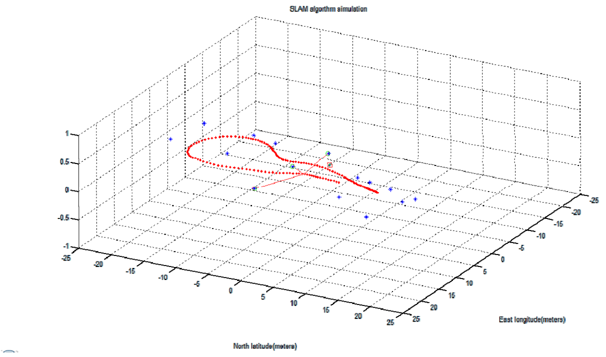


Figure 9. Simulating AUV route base on EKF-SLAM algorithm.

.

As can be seen in the figure 9, the AUV catches beacons while moving towards trajectory, so that the AUV can follow the mission plan autonomously.

Figure 10 and figure 11 shows the AUV's mission plan and simulating route respectively. It can be found that the deviation of two routes is minor which means that the EKF-SLAM decrease error and improves the simulation accuracy.

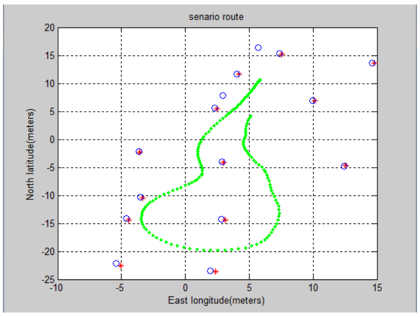


Figure 10. original AUV route

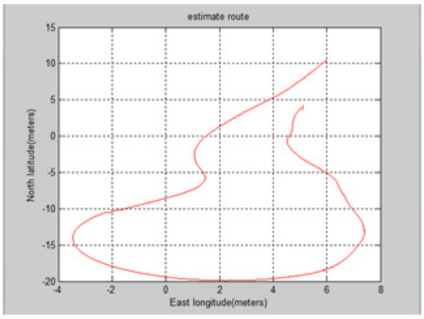


Figure 11: Simulating AUV route base on EKF-SLAM algorithm.

2. CONCLUSIONS & FUTURE WORKS

The simulation of each subsystems of the AUV developed in MATLAB-simulink is completed. By applying various mathematical models and algorithms, simulation results of each subsystem can be produced. A number of future improvements to the AUV simulation development have been identified. The first improvement would be to build a single simulation platform, including a Graphical User Interface (GUI) that provides a common setup and analysis of subsystem results. The second stage would involve the integration of each AUV subsystem, so that all subsystems are connected and communicate. The last stage involves the testing and development of new experiments to compare real world AUV performance with simulation results. Once the improved framework is completed, the simulator can be used to test new ideas, theories and the integration of new sensors into the AUV simulator.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the Defence Science and Technology Group (DSTG).

REFERENCES

[1] Ballard, C, Cui, JY, Jacobs, A & Tran, Q 2010, 'Intelligent AUV Robot Navigation and Tracking using Underwater Machine Vision and Sonar Imaging systems', Level 4 Design Project, School of Mechanical Engineering, The University of Adelaide

[2] B. Allen, R. Stokey, T. Austin, N. Forrester, R. Goldsborough, M.Purcell, C. von Alt, "REMUS: A small low cost AUV; System description, field trials, performance results," *Proceedings of IEEE Ocean 97*, pp. 994-1000, 1997.

[3] Chen, C. Bui, D.A., Rutter, J. & Wang, Q. 2011, 1073; Design, Build and Implementation of Intelligent Submersible Robot, Level 4 Design Project, School of Mechanical Engineering, The University of Adelaide

[4] J. Klahold, J. Rautenberg and U. Ruckert, "Continuous Sonar Sensing for Mobile Mini-Robots," Proceedings of the Robotics & Automation, 2002, Washington, pp. 323-328

[5] Newman, PM, Leonard, JJ & Rikoski, RJ 2003, 'Towards Constant-Time SLAM on an Autonomous Underwater Vehicle Using Synthetic Aperture Sonar', Department of Engineering Science, Oxford University, OX1 3PJ, UK

[6] Pang, YJ, Sun, YS, Gang, Y & Wan, L 2008, 'An integrated GPS/DR navigation system for AUV', *Journal of Marine Science and Application*, vol.5, pp. 8-13.

[7] Tan K.M., Liddy T., Anvar A., Tien-Fu L., "The advancement of an autonomous underwater vehicle (AUV) Technology," Proceedings of Industrial Electronics and Applications, 2008, Singapore, pp. 336-341