Osaka University Knowledge Archive

| | |
|---|---|
| Title | Development and operation of underwater robot for autonomous tracking and monitoring of subsea plumes after oil spill and gas leak from seabed, and analyses of measured data |
| Author(s) | Choyekh, Mahdi |
| Citation | 大阪大学, 2016, 博士論文 |
| Version Type | VoR |
| URL | https://doi.org/10.18910/59588 |
| rights | |
| Note | |

*Osaka University Knowledge Archive : OUKA*

https://ir.library.osaka-u.ac.jp/

Osaka University

Doctoral Dissertation

# Development and operation of underwater robot for autonomous tracking and monitoring of subsea plumes after oil spill and gas leak from seabed, and analyses of measured data

\

Mahdi CHOYEKH
June 2016

Graduate School of Engineering,
Osaka University

Department of Naval Architecture and Ocean Engineering



`

博士学位論文

# 海底から湧出する油やガスのプルームを自律的に追跡しモニタリングする海中ロボットの開発・運用と測定データの解析

シュワヤク マヘディ

2016 年 06 月

大阪大学大学院工学研究科

地球総合工学専攻船舶海洋工学コース

# Acknowledgements

I would like to express my gratitude to my supervisor Prof. Naomi Kato for giving me the golden opportunity to be a part of the project SOTAB which helped me in doing a lot of research and let me acquire much knowledge. I am also very grateful to Dr. Hiroyoshi Suzuki and Dr. Hidetaka Senga who have been supporting me the whole time and I am touched by their kindness. To all of you, I just wanted to say thank you for your guidance and for giving me time from your busy schedules. I am lucky to have met such wonderful supervisors as well as persons.

I wish to express my deep appreciation to Kimura, Ukita, Yamaguchi, Ryan, Kita, Ohnishi and Yamada for their valuable help and for their kindness during at-sea experiments. My sincere gratitude to Phyo, Swarn, Tar, Okubayashi, Nagai, Inoue, Tomobe and all other lab mates for all their help during my stay in Kato's lab and for the warm and friendly atmosphere.

Special thanks go to the Japanese Government for their involvement in the promotion of success in the lives of students. Thank you for putting your confidence in me and for trusting my abilities by giving me the honor to be a MEXT scholarship student. Your generous act is well appreciated and it allowed me to devote a greater amount of time to my studies and helped me to reach my goals.

For my beloved family and all my friends, I sincerely thank for their understanding, forgiveness, patience and support.

Suita, Osaka, Japan

August 1st, 2016

Mahdi CHOYEKH

# Contents


1 Introduction ..... 2
1.1 Background ..... 2
1.2 Objectives ..... 3
1.3 Overview of the Thesis ..... 3
1.4 Contributions ..... 4
2 SOTAB-I Overview and Software Development ..... 6
2.1 Outlines of SOTAB-I ..... 6
2.2 Hardware Description ..... 8
2.2.1 Power Supply ..... 8
2.2.2 Processing and Control Unit ..... 10
2.2.3 Actuators ..... 11
2.2.4 Tracking ..... 14
2.2.5 Orientation ..... 16
2.2.6 Communication ..... 17
2.2.7 Surveying Sensors ..... 18
2.2.8 Emergency ..... 20
2.3 Software Development ..... 21
2.3.1 Ship's Computer ..... 22
2.3.2 Robot's Computer ..... 24
2.3.3 Acoustic Communication ..... 31
2.4 Conclusions ..... 32
3 Navigation ..... 35
3.1 Introduction ..... 35
3.2 Navigation Overview ..... 37
3.2.1 Sensors Involved in the Navigation ..... 37
3.2.2 Shortcomings of the Employed Sensors ..... 38
3.2.3 Navigation Regions ..... 44
3.3 The Kalman Filter ..... 45
3.3.1 Overview ..... 45
3.3.2 Application of the Kalman Filter to Measured Experiments Results ..... 48
3.4 Conclusions ..... 56
4 SOTAB-I Guidance and Control ..... 58
4.1 Introduction ..... 58
4.2 Outlines of the Guidance and Control ..... 59
4.2.1 Operating Modes ..... 59
4.2.2 Operating Zones ..... 60
4.2.3 Control Priorities ..... 62
4.2.4 Control Program ..... 65
4.2.5 Equations of Motion ..... 66
4.3 Depth Control ..... 69
4.3.1 Introduction ..... 69
4.3.2 Establishment of the Buoyancy Model ..... 72
4.3.3 Control Algorithm ..... 79
4.3.4 Experiments Results on March 17th, 2016 ..... 84
4.3.5 Conclusions ..... 89
4.4 Altitude Control ..... 90
4.4.1 Introduction ..... 90
4.4.1 Altitude Calculation ..... 90
4.4.2 Control Algorithm ..... 95

4.4.3 Study of the effect of Altitude Control Parameters ..... 98
4.4.4 Experiments Results on March 17th, 2016 ..... 99
4.4.5 Conclusions ..... 103
4.5 Wings Control ..... 104
4.6 PID Collision Avoidance ..... 107
4.7 Energy Study ..... 110
4.7.1 Power Characteristics ..... 110
4.7.2 Depth Control: Cases Study ..... 112
4.7.3 Depth Stabilization ..... 115
4.7.4 Robot's Battery Autonomy Study ..... 116
4.7.5 Conclusions ..... 117
4.8 Conclusions ..... 118
5 Water Surveying ..... 121
5.1 Introduction ..... 121
5.2 Sensors Configuration and Calculation Process ..... 123
5.2.1 CTD ..... 123
5.2.2 ADCP ..... 125
5.2.3 UMS ..... 139
5.3 Vertical Water Column Survey in the Gulf of Mexico ..... 142
5.3.1 Temperature, Salinity, and Density ..... 143
5.3.2 Dissolution of Substances ..... 145
5.4 Vertical Water Column Survey in Toyama Bay on March 17th, 2016 ..... 147
5.4.1 Temperature, Salinity, Density and Sound Speed ..... 147
5.4.2 Vertical Profile of Water Currents ..... 148
5.5 Conclusions ..... 151
6 Conclusions and Future Work ..... 154
6.1 Conclusions ..... 154
6.2 Future Work ..... 158
Special Acknowledgements ..... 160
Appendix ..... 161
References ..... 168

# List of Figures

**Fig. 1** SOTAB-I robot ..... 6
**Fig. 2** Arrangement of devices and sensors installed on SOTAB-I ..... 7
**Fig. 3** Hardware classification ..... 8
**Fig. 4** Power supply conversion ..... 8
**Fig. 5** Example of power supply of actuators and sensors ..... 9
**Fig. 6** Computer communication interfaces ..... 10
**Fig. 7** Buoyancy device ..... 11
**Fig. 8** Relationship between the pressure and the volume in the oil reservoir ..... 12
**Fig. 9** Wings dimensions ..... 14
**Fig. 10** USBL system installed on the ship side ..... 15
**Fig. 11** Communication ..... 17
**Fig. 12** Diagram of operation during at-sea experiments ..... 21
**Fig. 13** Ship computer ..... 22
**Fig. 14** Overview of SOTAB-I GUI ..... 23
**Fig. 15** SOTAB-I computer ..... 24
**Fig. 16** Software architecture of SOTAB-I GUI ..... 25
**Fig. 17** Application layer, general structure of thread execution ..... 26
**Fig. 18** Serial data acquisition ..... 26
**Fig. 19** Example of UMS Data ..... 28
**Fig. 20** Effect of filtering using 1 € filter ..... 29
**Fig. 21** Buoyancy control driver flowchart ..... 30
**Fig. 22** ADCP/DVL data acquisition flowchart ..... 30
**Fig. 23** Acoustic communication data frame ..... 31
**Fig. 24** Robot's motion and orientation ..... 38
**Fig. 25** GPS position Vs Depth ..... 39
**Fig. 26** Robot's underwater positions near the seabed ..... 40
**Fig. 27** Position's accuracy Vs Vertical angle ..... 41
**Fig. 28** Effect of the sound speed on the depth measurement ..... 41
**Fig. 29** Acoustic positioning period ..... 42
**Fig. 30** Acoustic communication period ..... 43
**Fig. 31** Navigation regions ..... 44
**Fig. 32** The Kalman filtering algorithm ..... 45
**Fig. 33** Robot's positions and effect of the Kalman filtering ..... 49
**Fig. 34** Variations of the robot's positions near the seabed ..... 50
**Fig. 35** Sensor fusion using Kalman filter ..... 51
**Fig. 36** Robot's speeds near the sea surface ..... 54
**Fig. 37** Effect of ADCP aiding on position prediction ..... 54
**Fig. 38** Effect of DVL aiding on position prediction ..... 55
**Fig. 39** Water column regions ..... 60
**Fig. 40** Determination of water column region ..... 60
**Fig. 41** Control priorities ..... 62
**Fig. 42** Emergency flowchart ..... 63
**Fig. 43** Collision avoidance flowchart ..... 63
**Fig. 44** Manual control flowchart ..... 64
**Fig. 45** Normal operation zone flowchart ..... 65
**Fig. 46** Main program initialization ..... 65
**Fig. 47** Main program flowchart ..... 66
**Fig. 48** Relationship between the pressure and the buoyancy variation time ..... 74
**Fig. 49** Buoyancy variation with depth on 20th March 2015 experiment ..... 74
**Fig. 50** Buoyancy Variation Simulator ..... 77
**Fig. 51** Buoyancy simulator flowchart ..... 78

**Fig. 52** Depth Control Process ..... 79
**Fig. 53** Predictive depth control flowchart ..... 80
**Fig. 54** Predictive depth control input/output diagram ..... 81
**Fig. 55** Heuristic control flowchart ..... 83
**Fig. 56** Depth Control ..... 85
**Fig. 57** Predictive depth control ..... 86
**Fig. 58** Depth stabilization ..... 86
**Fig. 59** Robot's ascent ..... 87
**Fig. 60** Simulation Vs Experiment results ..... 87
**Fig. 61** Ideal control Vs Predictive control ..... 88
**Fig. 62** Altitude calculation process ..... 91
**Fig. 63** Altitude evaluation procedure ..... 94
**Fig. 64** Altitude control flowchart ..... 95
**Fig. 65** Altitude control ..... 100
**Fig. 66** Altitude control: Step 1 ..... 101
**Fig. 67** Altitude control: Step 2 ..... 101
**Fig. 68** Altitude control: Step 3 ..... 102
**Fig. 69** Altitude control: Step 4 ..... 103
**Fig. 70** Altitude control: Step 5 ..... 103
**Fig. 71** Comparison of lateral and vertical motion between case 1 and case 2 described in Table 22. ..... 104
**Fig. 72** Collision avoidance PID controller ..... 107
**Fig. 73** Vertical speed control through PID controller ..... 108
**Fig. 74** PID controller behavior ..... 109
**Fig. 75** Buoyancy device power characteristics ..... 110
**Fig. 76** Thruster power characteristics ..... 111
**Fig. 77** Comparison between five depth control scenarios ..... 113
**Fig. 78** Depth stabilization: Buoyancy actuator activation ..... 115
**Fig. 79** Altitude stabilization: Buoyancy actuator activation ..... 115
**Fig. 80** CTDs measurements of SOTAB-I and Toyama institute's sampling carousel ..... 125
**Fig. 81** Water current measurement process ..... 127
**Fig. 82** Depth interval averaging program inputs and outputs ..... 129
**Fig. 83** Effect of depth interval averaging ..... 130
**Fig. 84** Effect of moving average filtering ..... 131
**Fig. 85** SOTAB-I Deployment in Komatsu-shima ..... 132
**Fig. 86** Experiments place ..... 132
**Fig. 87** Single ping standard deviation of 1200 kHz systems ..... 133
**Fig. 88** Comparison between SOTAB-I and Fukae-maru water currents profiles near the sea surface ..... 135
**Fig. 89** Comparison between SOTAB-I and Fukae-maru water currents profiles near the seabed ..... 136
**Fig. 90** Comparison between SOTAB-I and Fukae-maru ADCPs in the middle zone ..... 137
**Fig. 91** Comparison between SOTAB-I and Fukae-maru water currents profiles using AINS ..... 139
**Fig. 92** Gulf of Mexico experimental zone (Google Map) ..... 142
**Fig. 93** Temperature and wind speed reported by the PSTL1 station on 13th December 2013 ..... 143
**Fig. 94** Vertical distributions of temperature, salinity and density ..... 144
**Fig. 95** Dissolution of substances in the water column ..... 145
**Fig. 96** Vertical distributions of temperature, salinity and density ..... 147
**Fig. 97** Water currents near the sea surface ..... 149
**Fig. 98** Water currents in the middle zone ..... 150
**Fig. 99** Water currents near the seabed ..... 151



`

# List of Tables

**Table 1** Principal particulars of SOTAB-I ..... 6
**Table 2** Variation of robot buoyancy against voltage on motor ..... 12
**Table 3** Wings dimensions ..... 14
**Table 4** Acoustic positioning performance ..... 42
**Table 5** Downlink acoustic communication performance ..... 43
**Table 6** The Kalman filter parameters' definition ..... 46
**Table 7** Configuration of the R matrix ..... 49
**Table 8** Calculation of the average robot's speed on the sea surface ..... 53
**Table 9** Definition of parameters of the equation of motion ..... 68
**Table 10** Time of variation of robot buoyancy with pressure ..... 73
**Table 11** Buoyancy variation model of 20th March 2015 experiment ..... 75
**Table 12** Comparison between the buoyancy variation model and experimental results ..... 76
**Table 13** Definition of the buoyancy control parameters ..... 81
**Table 14** Parameter configuration ..... 84
**Table 15** Comparison between simulation and experiment results ..... 88
**Table 16** Comparison between predictive control and ideal control ..... 88
**Table 17** Example of undesirable obstacle detection ..... 92
**Table 18** Statistics of undesired obstacle detections in the second dive on 2014/07/24 ..... 93
**Table 19** Effect of certain depth variation on altitude control performance ..... 98
**Table 20** Effect of target altitude variation on altitude control performance ..... 98
**Table 21** Altitude control parameters configuration ..... 99
**Table 22**: Summary of PID controller response. ..... 105
**Table 23** PID Collision avoidance parameters ..... 108
**Table 24** Power consumption of SOTAB-I devices ..... 110
**Table 25** Comparison between depth control scenarios ..... 112
**Table 26** Buoyancy device power consumption for 1000m dive ..... 116
**Table 27** Power consumption per dive ..... 116
**Table 28** Battery autonomy for 1000m dive ..... 116
**Table 29** CTD related oceanographic data ..... 123
**Table 30** SOTAB-I ADCP configuration ..... 126
**Table 31** Characteristics and Settings of Fukae-maru ADCP ..... 133
**Table 32** SOTAB-I drifting speed calculation on the sea surface ..... 134
**Table 33** Standard gas mixtures used for equilibration (in volume %) ..... 140
**Table 34** Characteristics of one pack Lithium-Ion battery ..... 161
**Table 35** Processing Unit Specification ..... 161
**Table 36** Thruster Specification ..... 161
**Table 37** Main characteristics of USBL system of SOTAB-I ..... 161
**Table 38** Main characteristics of GPS Compass on the ship ..... 161
**Table 39** Main characteristics of GPS receiver of SOTAB-I ..... 162
**Table 40** Main characteristics of the CTD sensor of SOTAB-I ..... 162
**Table 41** Main characteristics of the ADCP/DVL of SOTAB-I ..... 162
**Table 42** UMS Specifications. ..... 162
**Table 43** Compass Specifications. ..... 163
**Table 44** IMU specification ..... 163
**Table 45** Robot's GPS data format ..... 164
**Table 46** IMU configuration register ..... 164
**Table 47** IMU Binary Output Data Fields ..... 164
**Table 48** IMU binary output message configuration ..... 165
**Table 49** CTD data format ..... 165
**Table 50** ADCP/DVL PD0 Format ..... 165
**Table 51** UMS data format ..... 166



`

`

NOMENCLATURE

| | |
|---|---|
| ADC: | Analogic to digital converter |
| ADCP: | Acoustic Doppler current profiler |
| AUV: | Autonomous underwater vehicle |
| BTX: | Benzene-Toluene-Xylenes |
| CTD: | Conductivity-Temperature-Depth |
| CDOM: | Colored dissolved organic matter |
| DAC: | Digital to analogic converter |
| DIO: | Digital input output |
| DOC: | Dissolved organic carbon |
| DVL: | Doppler velocity logger |
| GPS: | Global positioning system |
| $I^2C$: | Inter-Integrated circuit |
| IDE: | Integrated development environment |
| KM: | Kalman filter |
| MIMS: | Membrane introduction mass spectrometry |
| PID: | Proportional-Integral-Derivative |
| PSU: | Practical salinity units |
| RMSE: | Root-mean-square deviation |
| RTOS: | Real time operating system |
| UMS: | Underwater mass spectrometer |
| USBL: | Ultra-short base line |
| UV–VIS: | Ultraviolet–Visible |
| VOC: | Volatile organic compound |
| VRU: | Vertical reference unit |
| WLAN: | Wireless local area network |



`

# Abstract

Oil spills produced by accidents from oil tankers and blowouts of oil and gas from offshore platforms cause tremendous damage to the environment as well as to marine and human life. To prevent oil and gas that are accidentally released from deep water from spreading and causing further damage to the environment over time, early detection and monitoring systems can be deployed to the area where underwater releases of the oil and gas first occurred. Monitoring systems can provide a rapid inspection of the area by detecting chemical substances and collecting oceanographic data necessary for enhancing the accuracy of simulation of behavior of oil and gas. An autonomous underwater vehicle (AUV) called the spilled oil and gas tracking autonomous buoy system (SOTAB-I) has been developed to perform on-site measurements of oceanographic data as well as dissolved chemical substances using underwater mass spectrometry. In this thesis, the outlines of SOTAB-I and a description of its hardware architecture and software development are presented at first. On the ship side, we designed and developed a GUI, which proved to be a valuable tool to keep the user informed about the status of the robot and the changes around its environment and allowed to send commands when necessary. On the robot side, the multilayered architecture of the SOTAB-I software enabled distribution of responsibilities. Software drivers were designed to assure the easy control of the actuators and the acquisition of sensors data. To ensure the transmission of critical information to the user on the mother ship, we established an acoustic communication data frame and implemented a verification method to secure the control of the robot through downlink. Next, the robot's navigation regions were defined. The use of the Kalman filter showed its effectiveness in filtering the position and absorbing spike noise. The filter also demonstrated its ability to fuse sensors' data. The predictive performance



`

of the filter was studied. The results showed that the use of the ADCP aiding contributed to the improvement of the position estimation accuracy. In the next section, the guidance and control is explained. A new method for depth control using the buoyancy control device was developed based on a model of the buoyancy variation with time. The model was established based on the results obtained in high pressure tank experiment and several at-sea experiments. The method was demonstrated at-sea experiments in Toyama Bay in Japan in March 2016. It showed the ability of the control algorithm to smoothly bring the robot to the target depth without a significant overshoot. The method could be further adapted to perform an altitude control through a progressive depth control algorithm based on 4 steps. The experiment results showed that it worked properly. In Suruga Bay experiments in Japan, the effect of wings of SOTAB-I was tested and showed the ability of the robot to move on the lateral plan with an acceptable ratio of the horizontal movement to the diving depth. The collision avoidance concept was defined with its regions. The PID speed control succeeded to smoothly freeze the robot submergence to the dangerous zone. The energy study enabled to estimate the robot's battery autonomy under different scenarios. The comparative study between the use of buoyancy device and thrusters proved the power efficiency of the depth control algorithm. The power efficiency of the depth stabilizer algorithm was studied and proved its ability to reduce the energy consumption of the robot. Finally, we present experimental results obtained during the early deployments of SOTAB-I in the shallow water of the Gulf of Mexico in the U.S., demonstrating the ability of SOTAB-I to collect substances' dissolutions in seawater such as hydrocarbons, followed by the results of the deep water experiments conducted in Toyama bay in Japan, enabling the demonstration of the ability of SOTAB-I to establish the vertical water column distribution of oceanographic data, such as temperature, salinity, density and water currents.

# Chapter 1: Introduction



`

# 1 Introduction

## 1.1 Background

The world economy depends to a large extent on the use of energy. Particularly, consumption has increased for oil by 0.8% and by 0.4% for natural gas from the end of 2013 to the end of 2014 (BP, 2015). To meet the increasing need for energy, both in industry and daily life, petroleum activities, such as drilling and shipping, are on the rise. For instance, oil production has increased by 2.3% for oil and by 1.6% for natural gas from the end of 2013 to the end of 2014 (BP, 2015). Thus, additional attention is required to avoid accidents that can happen due to such activities. Oil spills and blowouts of oil and gas from the seabed cause serious damage to the environment as well as to the economy, not to mention the damage to marine and human life (Ober, 2010). For the case in which methane gas is blown out from a seabed, it is partly dissolved in seawater and then partly consumed by methanotrophs (Kessler et al., 2011), which leads to the creation of local hypoxia zones caused by oxygen depletion (Shaffer et al., 2009). The rest of the gas is released to the atmosphere, contributing to global warming, as methane is a highly potent greenhouse gas (Solomon et al., 2009). Recently, several oil spill accidents have happened (ITOPF, 2015). Deepwater Horizon in the Gulf of Mexico in 2010 and the Elgin gas platform in the North Sea in 2012 are examples of these accidents. To prevent oil and gas spills from spreading and causing further damage to the environment over time, early detection and monitoring systems can be deployed around the offshore oil and gas production system. In addition, oceanographic data should be collected to comprehend the environmental changes around the accident. Based on the collected information, oil and

gas drifting simulations must be performed to predict where the spilled oil will wash ashore and to adequately deploy oil recovery machines before this occurs. In this context comes SOTAB project (Choyekh, et. al., 2013), whose objectives are as follows: (1) Autonomous tracking and monitoring of spilled plumes of oil and gas from subsea production facilities by an underwater buoy robot called SOTAB-I, (2) Autonomous tracking of spilled oil on the sea surface and transmission of useful data to a land station through satellites in real time by multiple floating buoy robots called SOTAB-II (Senga, et. al., 2012), (3) Improvement of the accuracy of simulations for predicting diffusion and drifting of spilled oil and gas by incorporating real-time data (Takagi, et. al., 2012) (Tsutsukawa, et. al., 2012).

## 1.2 Objectives

The development of a new type of AUV requires an evaluation process from two aspects. One is the guidance and control of the robot, and the other is the data sampling. The research focuses on 4 axis:

- Software development of the robot
- Navigation
- Guidance and control of the robot
- Water survey of oceanographic data and dissolution of substances

## 1.3 Overview of the Thesis

In the first part of this chapter, we present and overview of the underwater robot SOTAB-I and a description of its hardware, followed by its software development. The second part deals with the robot navigation. In the third part, the maneuverability of the SOTAB-I and

its guidance and control algorithms are explained. The fourth part shows the survey results obtained in shallow water in the Gulf of Mexico in the U.S. and in deep water in Toyama Bay in Japan. Finally, the conclusions and future work are presented.

## 1.4 Contributions

The contributions of this doctoral dissertation can be listed as

- Software development of the system
    - Robot GUI
    - Ship GUI
    - Acoustic communication protocol
    - Data storage and acquisition
- Navigation
    - Data fusion and filtering
    - Robot's motion's prediction
- Guidance and control of the robot
    - Control programs' priority management
    - Enhancement of the robot's simulator program by implementing the buoyancy device simulator and actuators' power consumption models
    - New method for depth and altitude control using buoyancy device
    - Study of the effect of wings control
    - Emergency and fail-safe management
    - Study of the energetic performance of the control programs and robot's battery autonomy.
- Water Surveying
    - Survey of the oceanographic data
    - Survey of the dissolution of substances

# Chapter 2: SOTAB-I Overview and Software Development



`

# 2 SOTAB-I Overview and Software Development

## 2.1 Outlines of SOTAB-I

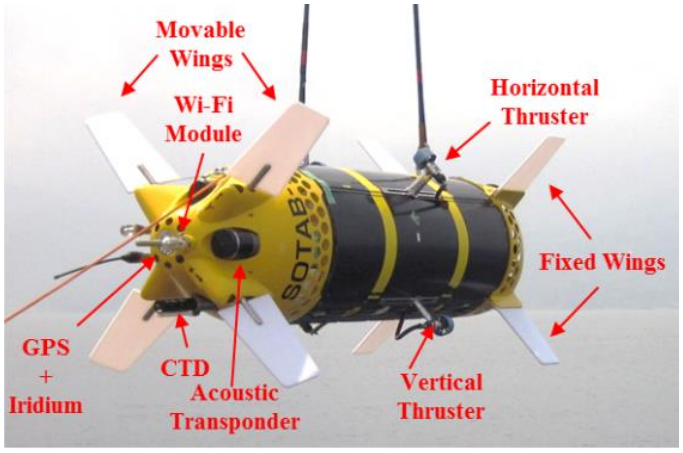



**Fig. 1** SOTAB-I robot

The SOTAB-I is 2.5 m long and weighs 325 kg. It can be submerged in water as deep as 2,000 m. It is able to descend and ascend by adjusting its buoyancy using a buoyancy control device while changing its orientation through two pairs of movable wings. The SOTAB-I can also move in horizontal and vertical directions using two pairs of horizontal and vertical thrusters. A visual overview of SOTAB-I is illustrated in Fig. 1, and its main characteristics are summarized in Table 1. The arrangement of devices and sensors installed on SOTAB-I is shown in Fig. 2.

**Table 1** Principal particulars of SOTAB-I

| | |
|---|---|
| Total Length [mm] | 2503 |
| Diameter [mm] | 667 |
| Weight in Air [kg] | 311.7 |
| Weight in Water [kg] | ±3.8 |

When the robot floats on the sea surface, a wireless local area network (WLAN) and an iridium satellite communication transceiver module are used for data transmission. When the robot is underwater, the user on the mothership and the SOTAB-I can communicate through the acoustic modem.

The robot tracking on the sea surface is ensured by a global positioning system (GPS) receiver that serves to determine the robot's absolute position. In the case where the robot is submerged, tracking is ensured by the ultra-short baseline (USBL) system. The vertical position of the robot in the water column is given by depth data from the CTD sensor. When the robot is within the bottom tracking altitude from the seabed, the Doppler velocity logger (DVL) is able to measure robot's velocities. The robot motion and orientation are given by the compass and the Inertial Measurement Unit (IMU). An ADCP is employed to measure the magnitude and orientation of water current layers. SOTAB-I is also fitted with a UMS to determine the characteristics and physical properties of the dissolved gas and oil. To obtain a visual representation of blowouts of plumes of gas on the seabed, the robot is equipped with a camera.

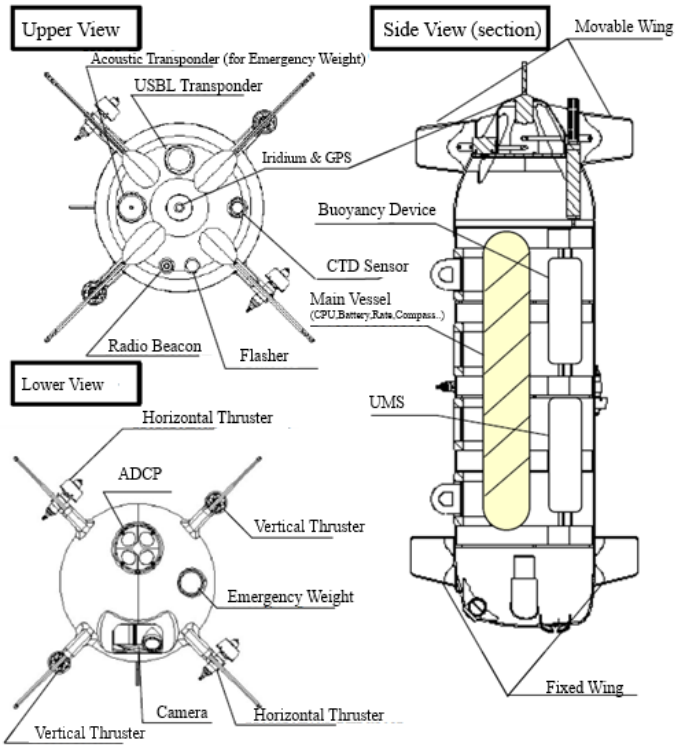



**Fig. 2** Arrangement of devices and sensors installed on SOTAB-I

## 2.2 Hardware Description

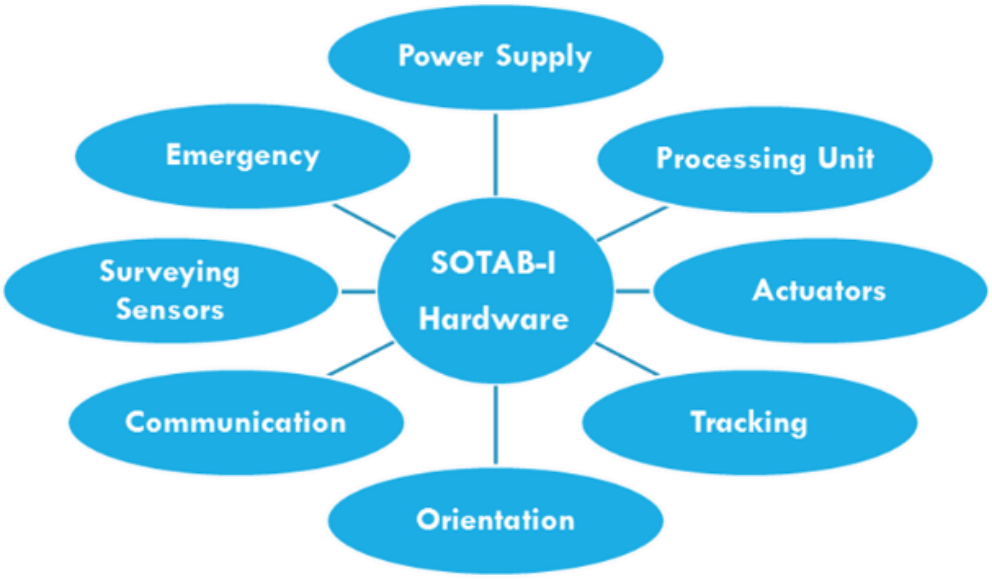



**Fig. 3** Hardware classification

SOTAB-I devices can be classified based on their function, as shown in Fig. 3. Following are further details about SOTAB-I hardware.

### 2.2.1 Power Supply

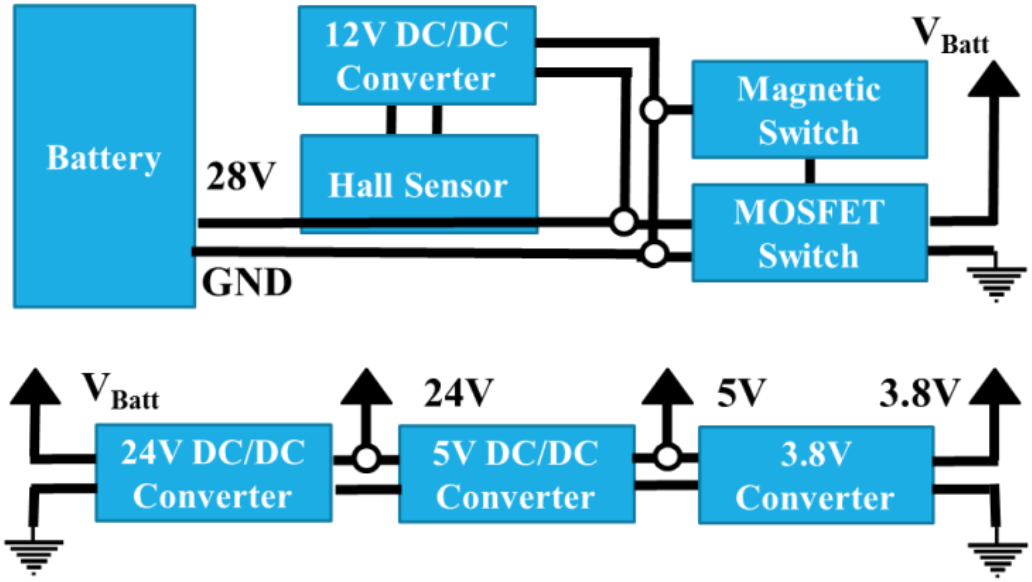



**Fig. 4** Power supply conversion

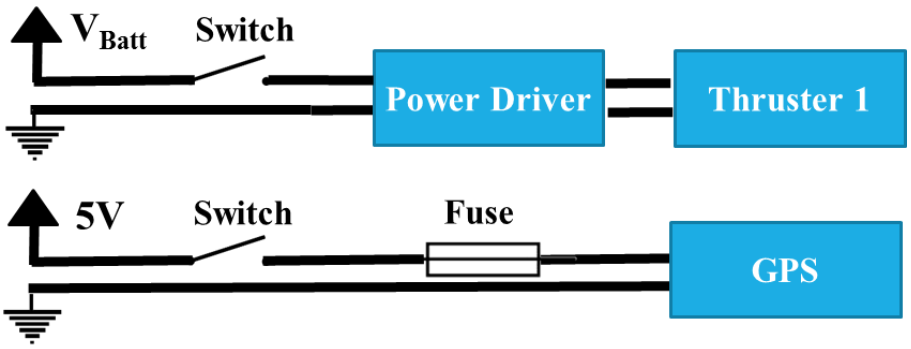



**Fig. 5** Example of power supply of actuators and sensors

SOTAB-I is powered by 32 packs of lithium-ion batteries divided into pairs and mounted in a serial arrangement. The 16 pairs are then mounted in parallel, which gives a voltage of 28.8 V because every single pack voltage is 14.4 V. The total capacity of the batteries is equal to 4608 Wh. Table 34 presents the characteristics of the batteries used. As shown in Fig. 4, the power is activated via an external magnetic switch that controls the gate of the power MOSFET (metal-oxide-semiconductor field-effect transistor), used here as a power switch. DC/DC converters are employed to convert the battery voltage output to the different levels of power supply required for the operation of the internal devices. A hall sensor is used to measure the current. Two power switches control boards serve to control the power supply of internal devices. They also include necessary fuses used for protection against surcharges. Power activation of each sensor/actuator is controlled by the processing unit board I/O port. Control boards and power boards are optically separated. Fig. 5 shows a typical electronic circuit used for powering the GPS sensor. The same concept is used for all other sensors, with the only difference in the input voltage. An example of a circuit used to power the thruster 1 is also shown in Fig. 5. Similarly, all actuators such as thrusters, wings, and buoyancy device have the same power supplying concept.

### 2.2.2 Processing and Control Unit

The processing unit of the SOTAB-I is made up of two processors. The volatile memory capacity is 1 GB. The characteristics of the processing unit are detailed in Table 35. The board has an Ethernet port connected to a Wi-Fi router that serves to connect to a remote computer through a wireless Local Area Network (LAN). The processing unit is also fitted with several interfaces such as USB, RS-232, and inter-integrated circuit (I2C). Three of the USB ports are used to connect external flash memories. The SOTAB-I computer is composed of three storage drives. The C drive is used for the operating system, the E drive for the programs' setup files, and the D drive for logging and program storage.

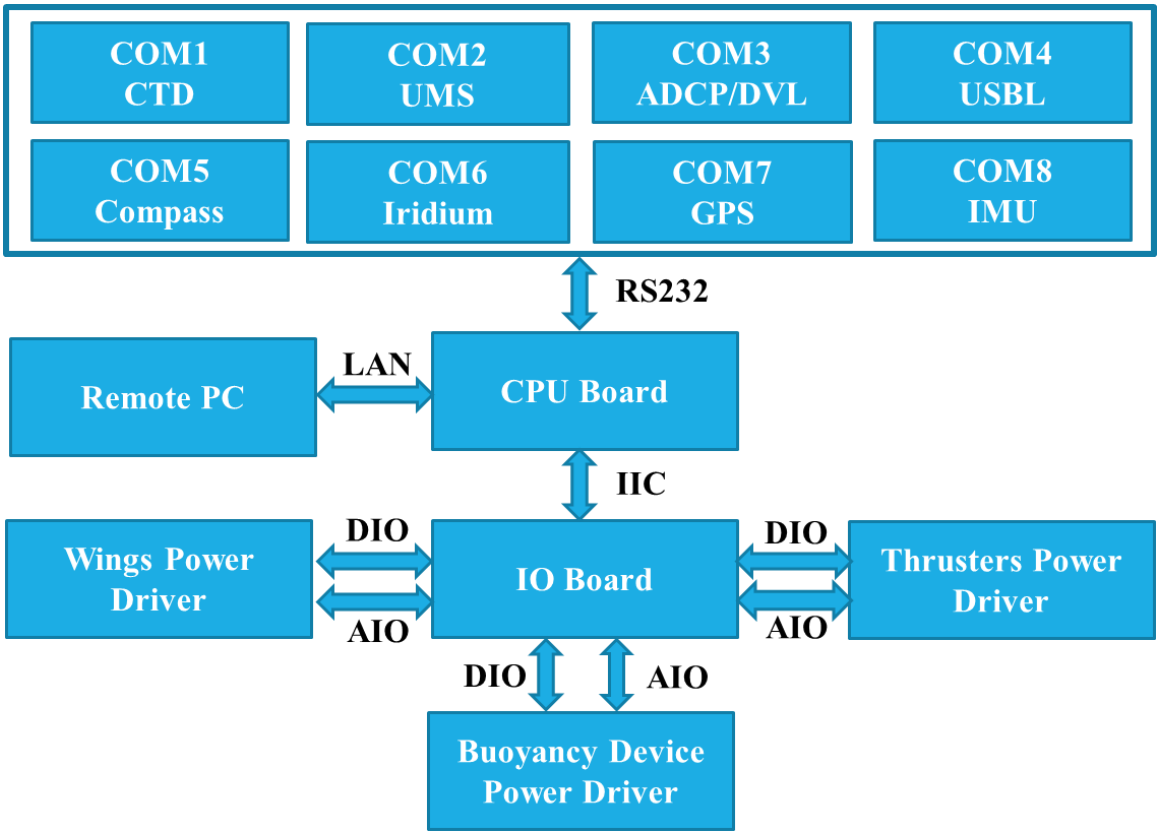



**Fig. 6** Computer communication interfaces

All sensors are directly interfaced through their respective RS-232 serial ports. Internally, the CPU board connects to an input/output board (IO board) through the inter-integrated

circuit ($I^2C$) serial interface. The IO board is composed of circuits that are able to interface with 32 digital input outputs (DIO), 8 analog outputs using digital-to-analog converters (DAC), and 8 analog inputs (AI) using analog-to-digital converters (ADC). The IO board ensures isolation between the CPU board and the power drivers. CPU board interfaces are summarized in Fig. 6

### 2.2.3 Actuators

#### 2.2.3.1 Buoyancy Device

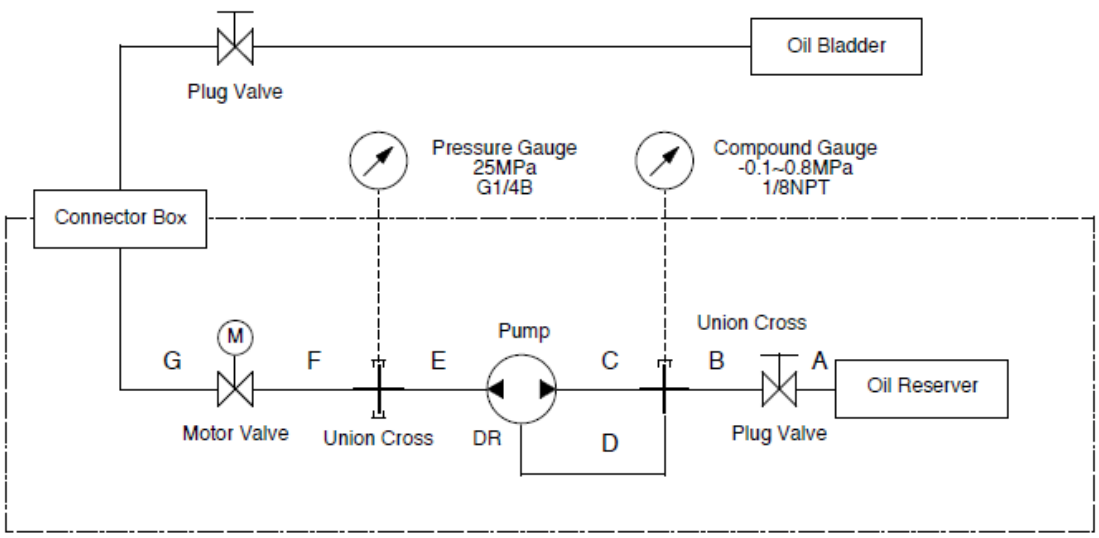



**Fig. 7** Buoyancy device

In the buoyancy control device, an oil hydraulic pump injects and extracts oil between the external oil bladder and the internal oil reservoir. A motor valve serves to automate opening and closing cycles, and a brake is used to lock the pump (Fig. 7). The flow rate during the injection of oil into the bladder at the external pressure of 20 MPa is 243 mL/min, and during the extraction of oil from the bladder at the same external pressure condition, it is 349 mL/min. In total, six digital inputs are employed to control the

buoyancy device. A digital input serves to control the power supply relay. One input is used to open the valve and another to close it. To control the motor pump actuator, one input is used to run/disable it and another one specifies the rotation direction. One more serves to activate/deactivate the brake. The feedback is provided by two digital outputs that report the valve position, and one analog output that provides the oil level.

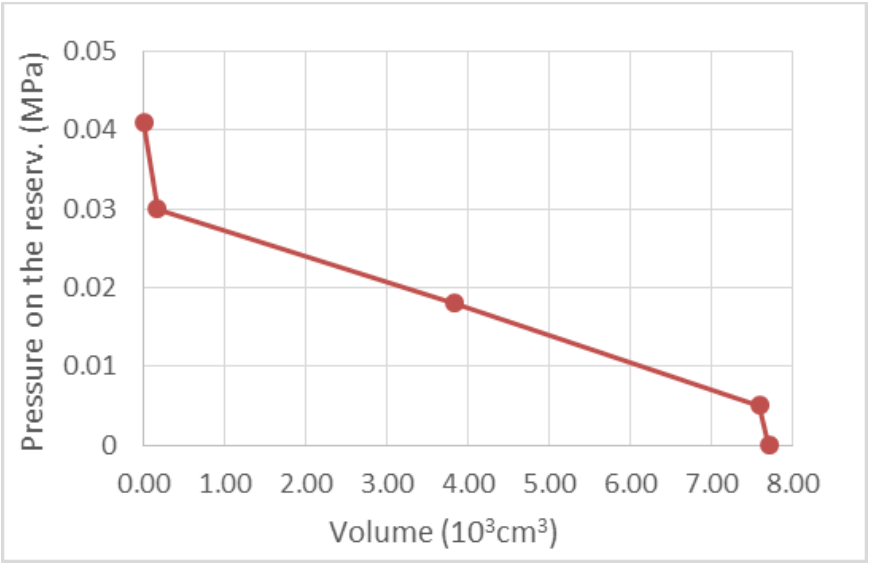



**Fig. 8** Relationship between the pressure and the volume in the oil reservoir

**Table 2** Variation of robot buoyancy against voltage on motor

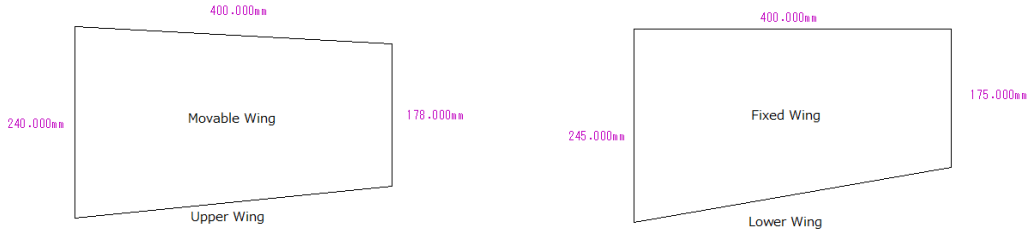
| Mass change (Sea water) (g) | Volume change ($cm^3$) | Potentiometer output voltage (V) | Pressure (MPa) |
|---|---|---|---|
| 7,884.8 | 7,700.0 | 1.621 | 0 |
| 7,746.6 | 7,565.0 | 1.599 | 0 |
| 7,763.2 | 7,581.2 | 1.602 | 0.005 |
| 3,925.7 | 3,833.7 | 1.012 | 0.018 |
| 171.8 | 167.8 | 0.443 | 0.030 |
| 0 | 0 | 0.389 | 0.041 |

Fig. 8 depicts the relationship between the pressure in the reservoir and the volume of the oil in the reservoir. The “pressure in the reservoir” is the pressure applied by the hydraulic pump to the oil reservoir cylinder. The maximum allowable drain pressure of the pump is 0.03 MPa. The oil reservoir is fitted with a linear potentiometer whose analog output

voltage is an image of the piston displacement under the pump drain pressure. The voltage is proportional to the oil volume of the reservoir, which leads to the variation of robot buoyancy, as shown in Table 2. When the oil room is full, the output voltage is 0.3892 V. When the oil room is empty, the output voltage is 1.6208 V. Because the reservoir has a cylindrical shape and the surface of the base is constant, the output voltage of the potentiometer can be used to determine the volume of the oil in the reservoir, which explains the linear volume/output voltage relationship. The mass change can be obtained after multiplying the volume change with the density of seawater, which is typically equal to 1024 kg/m3.

#### 2.2.3.2 Wings

The robot is equipped with two pairs of rotational wings situated at the top side of the robot and four fixed wings attached at the bottom side. Each pair of movable wings is controlled around a rotational axis with the same direction and angle by a servomotor, and the rotational axis of one pair of movable wings is perpendicular to that of the other. DC motor drives connected to a gear head with a 53:1 ratio enable each pair of wings to move from -90 to 90°. A rotational potentiometer with a resolution better than 0.1° provides information about the current orientation of the wings. A separate microcontroller control board integrating a proportional-integral-derivative (PID) controller program is used to control the wings through a serial port. The typical power consumption of the wing drive is 7 W. The dimension of the wings are specified in Table 3. The movable wing and the fixed wing are respectively illustrated in Fig. 9 (a) and Fig. 9 (b).




(a) Movable Wing **(b)** Fixed Wing

**Fig. 9** Wings dimensions

**Table 3** Wings dimensions

| | | |
|---|---|---|
| Movable Wing | Chord [mm] | 200 |
| | Span [mm] | 400 |
| Fixed Wing | Chord [mm] | 200 |
| | Span [mm] | 400 |

#### 2.2.3.3 Thrusters

Thrusters are controlled by an analog input varying from 0 to ±5 V. A relay is used for turning the power on and off. A tachometer measures the rotational speed and converts it to an analog output. Each thruster has a maximum thrust force equal to 35.28 N forward and 26.46 N backward. Table 36 gives more details about thruster specifications.

### 2.2.4 Tracking

SOTAB-I is fitted with a GPS receiver that can output the robot's absolute position when it is on the sea surface. DGPS accuracy is better than 3 m with 95% confidence. The characteristics of the GPS receiver are described in Table 39.

When the robot is underwater, an acoustic positioning system (APS) based on ultra-short baseline technology (USBL) is employed for tracking. It is composed of two parts. The first part consists of an acoustic transponder mounted on the robot. The second part consists of a pole on which all necessary devices needed on the ship side are assembled (Fig. 10). On the top of the pole, a GPS-Compass module and a vertical reference unit (VRU) are installed. At the bottom, an acoustic transceiver and a hydrophone are

submerged in seawater.

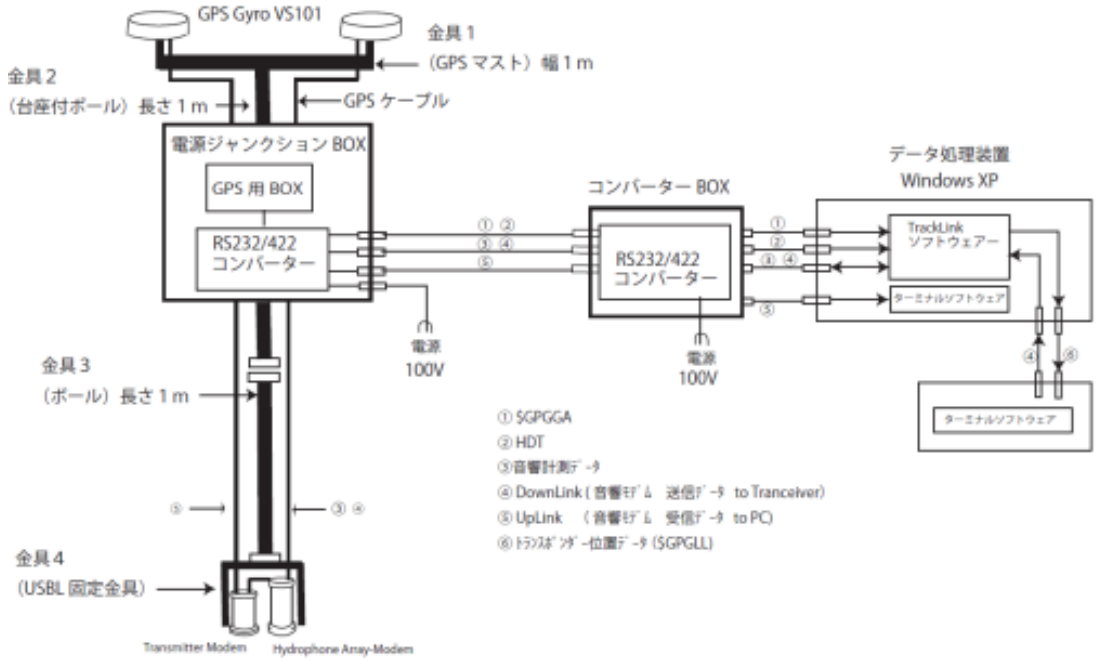



**Fig. 10** USBL system installed on the ship side

The relative underwater position of SOTAB-I to the mother ship is combined with the ship absolute position to determine the absolute position of the robot in the earth fixed coordinate. USBL determines the relative position by calculating the range and the angle between the transceiver and the transponder. At first, an acoustic pulse is transmitted by the transceiver. When the robot's transponder detects the pulse, the transponder replies by sending another pulse. The time difference between the transmission of the transceiver pulse and the arrival of the transponder pulse is converted to range, while the phase difference within the transceiver's transducer array determines the angle.

The USBL system deployed with the SOTAB-I can track up to eight targets at a range of up to 5000 m with ship noise. Its accuracy is equal to 0.15° for the bearing angle accuracy, and 0.3 m for the slant range. On board the ship, a GPS pole is deployed with a vertical

reference unit (VRU) system to determine the absolute position of the ship in addition to the tilting of the acoustic transceiver. The GPS-Compass has a DGPS horizontal accuracy better than 0.6 m with 95% confidence. Heading accuracy is better than 0.15° rms. Further details about the GPS-Compass are described in Table 38.

### 2.2.5 Orientation

SOTAB-I is fitted with a tilt-compensated compass module that provides attitude information. The accuracy of the compass module is 0.3° rms for the heading angle and 0.2° rms for the tilting angles. The maximum sampling frequency is 20 Hz, and its typical power consumption at that rate is 0.1 W. The module is accompanied with hard and soft iron calibration algorithms that contribute to the improvement of the reliability and consistency of heading measurement. Detailed features of the IMU sensor are detailed in Table 44.

SOTAB-I is also equipped with an Inertial Measurement Unit (IMU) and Attitude Heading Reference System (AHRS) that combine three-axis accelerometer, three-axis gyro, three-axis magnetic sensor, a barometric pressure sensor, and a 32-bit built-in microcontroller. The IMU microcontroller runs a quaternion based Extended Kalman Filter (EKF), which provides estimates of the attitude of the sensor as well as the real-time gyro biases. It can output acceleration, angular rate, and magnetic measurements along the body coordinate frame as well as in the North-East-Down (NED) coordinate frame. The device can also measure the temperature and the barometric pressure. The sensor has already been subject to a factory calibration. Additionally, it provides the user with the ability to apply a separate user calibration to remove additional bias, scale factor, and axis

misalignments. The sensor includes a separate EKF that provides real-time estimation of the local magnetic hard and soft iron distortions. The maximum power consumption of the module is 0.22 W. Data measurement update rate, including Kalman filter data processing, can go as high as 300 Hz. The communication with the IMU sensor is performed through an RS-232 serial port, but it is also possible using the SPI interface. The system was configured to output binary message data, which contains the maximum amount of information.

### 2.2.6 Communication

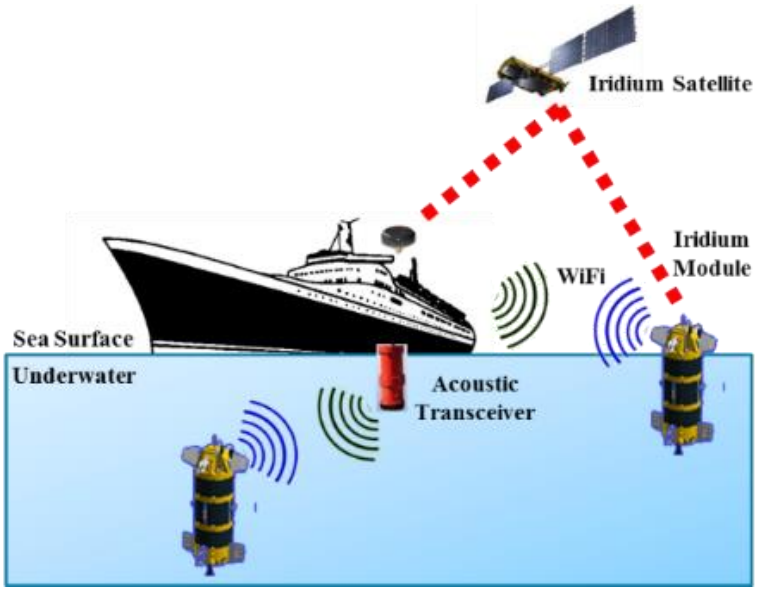



**Fig. 11** Communication

The communication between the SOTAB-I and the mother ship depends on the distance between them and whether the robot is on the sea surface or underwater (Fig. 11). When the SOTAB-I is on the sea surface and within a 200 m range of the mothership, it is possible to remotely connect to the robot computer and take full control of the robot. The UDP port is used to exchange data, such as the state of the joystick control button, between the ship GUI and the robot GUI. SOTAB-I is also fitted with an Iridium satellite

communication transceiver module. It has global coverage and can send and receive messages within 6 to 22 seconds. Transmission speed is up to 340 bytes per message, and reception speed is up to 270 bytes per message. When the robot is underwater, real-time communication with the ship using radio waves becomes impossible, and acoustic communication is used instead. In the current configuration, the acoustic communication speed is equal to 1320 bytes per 8 s for uplink and 80 bytes per 8 s for downlink. To be able to establish acoustic communication, it is important that the robot be within the operating beam width of the transceiver, which is equal to 120°.

### 2.2.7 Surveying Sensors

#### 2.2.7.1 CTD

The CTD sensor can measure the temperature, conductivity, and pressure of seawater. Based on those measurements, it is possible to calculate additional oceanographic data such as salinity and density as well as sound speed. The vertical position of the robot in the water column can also be obtained based on the pressure value. The CTD sensor employed has a sampling frequency up to 16 Hz, enabling a very high spatial resolution. It has a high accuracy and a typical power consumption of 3.4 W. A serial RS-232 communication port interfaces with the sensor. The characteristics of the sensor are described in Table 40.

#### 2.2.7.2 ADCP/DVL

An ADCP/DVL device was used for water profiling and bottom tracking. It integrates heading and attitude sensors that are necessary for coordinate transformation. The accuracy is within ±2° for the compass and ± 0.5° for the attitude. An integrated thermistor measures water temperature and improves the accuracy of calculation of sound speed as

well as enhances the accuracy of the acoustic measurements. The device is mounted looking downward at the bottom of the robot, as shown in Fig. 2. The device has four piston transducers with standard acoustic frequency equal to 1228.8 kHz, which enables high-resolution measurements of water currents up to 13 m range at 0.5 m layer resolution. The number of layers is selectable, and the layer thickness can be customized from 0.25 m to 5 m. When the robot is within the bottom tracking altitude from the seabed (23 m when the device is powered at 24 V), the Doppler Velocity Log (DVL) uses the sound waves bouncing off the sea floor to determine the robot velocities as well as its altitude from the seabed. The average power drawn by the ADCP/DVL pinging at its maximum rate at maximum altitude is 3 W, with a constant background power consumption equal to 2.2 W. The robot's processor connects to the ADCP/DVL device through an RS-232 serial port. Table 41 summarizes the main characteristics of SOTAB-I ADCP.

#### 2.2.7.3 UMS

SOTAB-I is equipped with an Underwater Mass Spectrometer (UMS). The UMS instrument used for the SOTAB-1 deployments contained a 200 amu linear quadrupole mass analyzer (E3000, Inficon, Inc., Syracuse, New York). Table **42** provides the specifications of SRI International membrane introduction mass spectrometry (MIMS) system.

Introduction of analytes into the mass spectrometer occurs through a hydrophobic and nonporous high-pressure polydimethlyl siloxane (PDMS) membrane introduction system, pressure tested to a depth of 2,000 m. Water samples are placed in contact with the semi-permeable membrane, usually at a constant flow rate. The transport of dissolved gases and

relatively non-polar volatile organic compounds (VOCs) through these membranes is compound-specific and temperature-dependent, but typically requires that the solute dissolves into the membrane, diffuses through it, and finally evaporates into the mass spectrometer. Once in the mass spectrometer vacuum chamber, the neutral gas-phase analytes are (1) ionized by electron impact, (2) sorted by their mass-to-charge (m/z) ratios (typically z = 1), and (3) detected to create a mass spectrum. The membrane interface used in this system provides parts-per-billion level detection of many VOCs and sub parts-per-million detection limits for many dissolved light stable gases.

The membrane probe assembly consists of a hollow fiber PDMS membrane stretched and mounted on a sintered Hastelloy C rod. One end of the supported membrane is capped with a polyetheretherketone (PEEK) rod; the other end is connected to the vacuum chamber via stainless steel tubing. The membrane assembly is inserted into a steel heater block that houses a thermocouple and heater cartridges for controlling sample and membrane temperature (± 0.1°C). A magnetic piston pump draws ambient water into the sample tubing, through the membrane probe assembly, and back to the environment.

### 2.2.8 Emergency

Since the communication with the robot may be lost for various reasons, such as a software crash or a hardware problem, commanding the robot may be no longer possible. In this particular case, a second method based on the use of an acoustic weight cutoff unit, which is completely independent from the circuit of the system, can be utilized for emergency surfacing. An additional radio beacon and a blinker flasher light are installed on the robot to assist in finding it in such circumstances.

## 2.3 Software Development

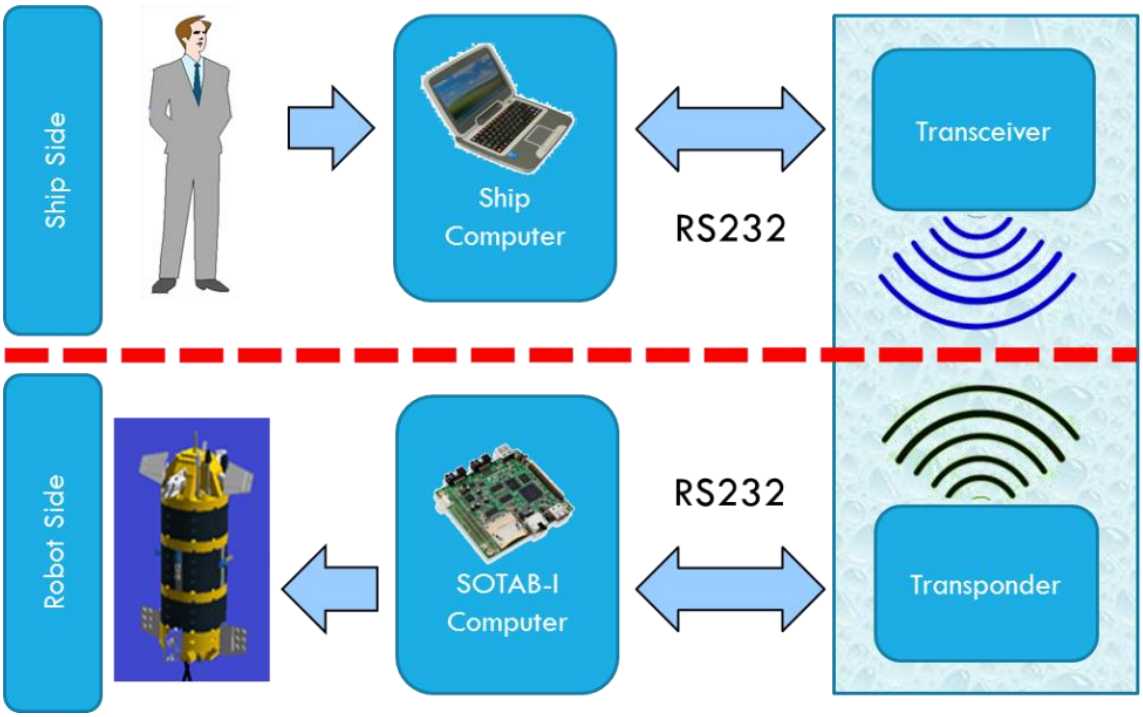



**Fig. 12** Diagram of operation during at-sea experiments

The SOTAB-I performs its surveying tasks autonomously. The operating mode is selected by the user on board the mother ship at the beginning of the operation by means of a graphical user interface (GUI). Orders are transmitted to the underwater robot through the acoustic modem. On the robot side, oceanographic and dissolved substance data collected by the SOTAB-I are sent to the mother ship in real time through an acoustic modem. On the mother ship side, the GUI receives the data collected from SOTAB-I, stores them, and displays the most essential data to the user (Fig. 12). Data related to the spilled plume and underwater currents are processed not only for guidance and control of the SOTAB-I, but also for the simulation and the prediction of plume behavior contributing to the decision-making process for the best deployment strategy of collecting and cleaning machines.

### 2.3.1 Ship's Computer

On board the ship, a portable computer running on Windows 7 operating system is employed. The integrated development environment (IDE) is Borland C++ 6. The GUI and the Tracklink will be executed. Tracklink is the software that comes with the USBL product.

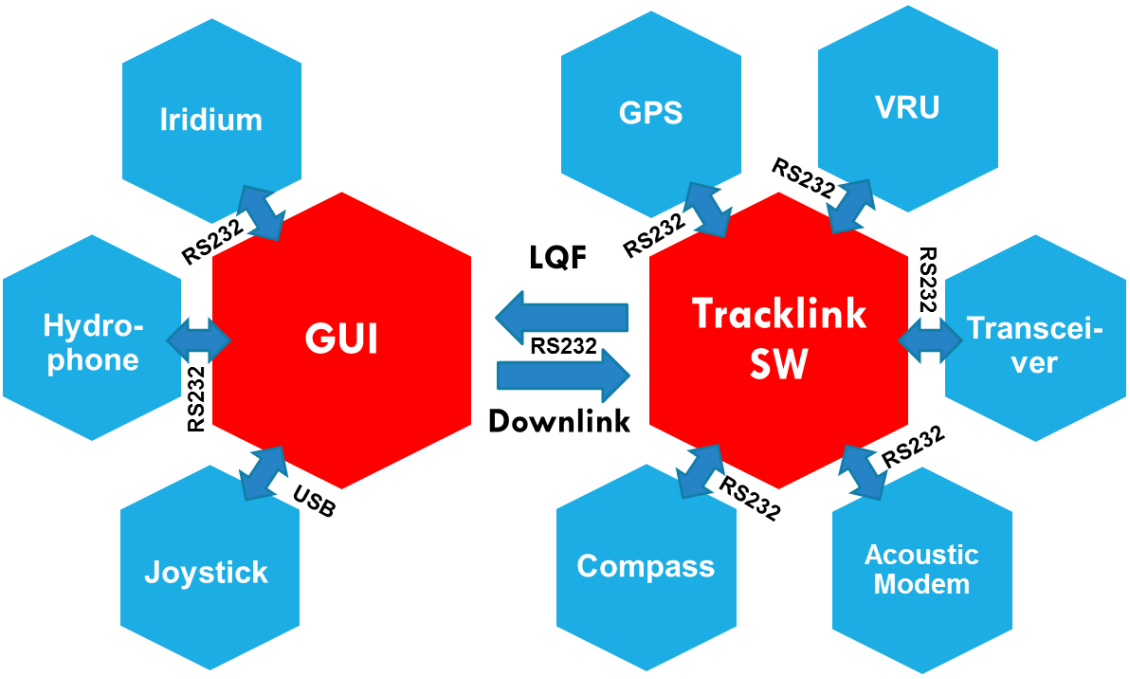



**Fig. 13** Ship computer

Tracklink is used to determine the robot position and to manage the underwater communication. It connects to the external devices through an RS-232 serial interface (Fig. 13). Tracklink output data follows the LQF format. The data include information about the ship position and heading as well as the robot position and its depth. Tracklink and the ship GUI exchange information through virtual RS-232 serial ports. The ship GUI sends downlink data to the robot via the Tracklink software. From the GUI side, the Iridium module and the hydrophone are interfaced through a serial RS-232 interface. A

joystick is interfaced through the USB port. The GUI (Fig. 14) has several roles that can be classified to three categories: display, guidance, and communication. The GUI shows the most important information about the robot status and its environment. It displays the robot position, orientation, and speed in addition to its actuators' status. Environmental data such as oceanographic data and substances' dissolutions are also displayed. Additionally, the GUI enables the selection of the operating mode and the manual control of the actuators. It gives the possibility to send a software emergency ascending order. The GUI combines interfaces for managing the Iridium satellite communication and the acoustic communication.

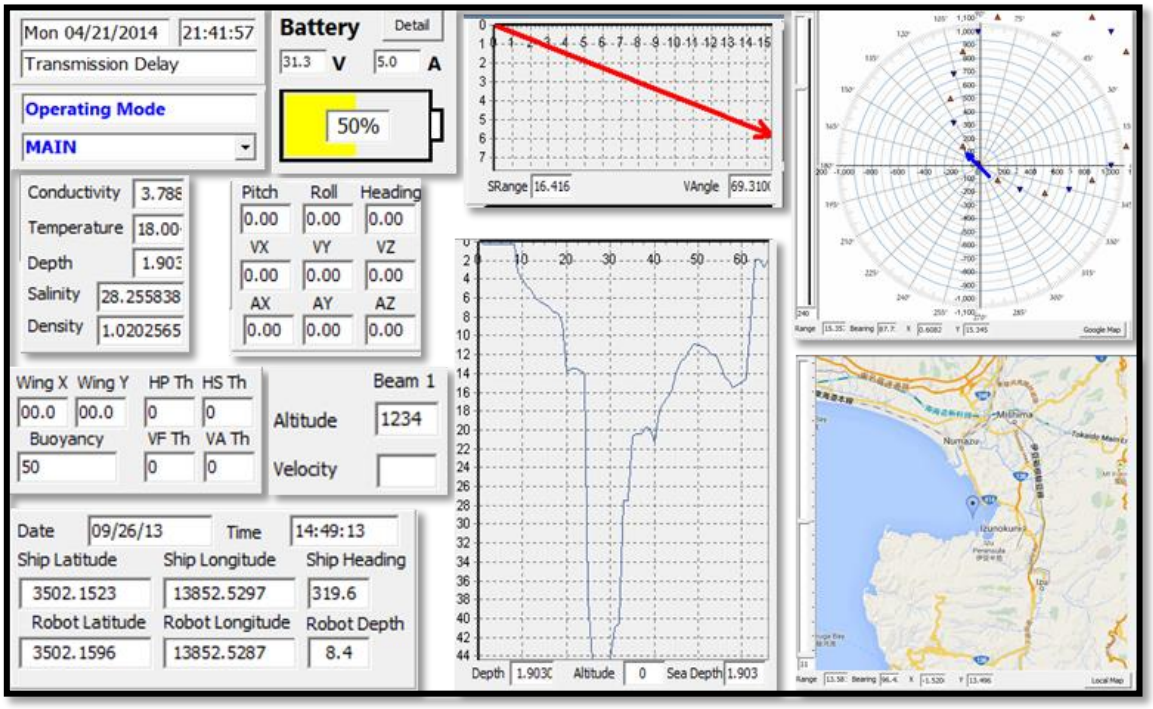



**Fig. 14** Overview of SOTAB-I GUI

### 2.3.2 Robot's Computer

The SOTAB-I program is running on the Windows XP embedded operating system. Borland C++ V6.0 was selected as the integrated development environment (IDE) to develop the GUI. At the beginning of the operation, the magnet that is put on the magnetic switch is removed and the system is powered on. After the operating system is started, the executable program of the GUI is launched automatically. The average CPU usage when the program is fully executing is 3%.

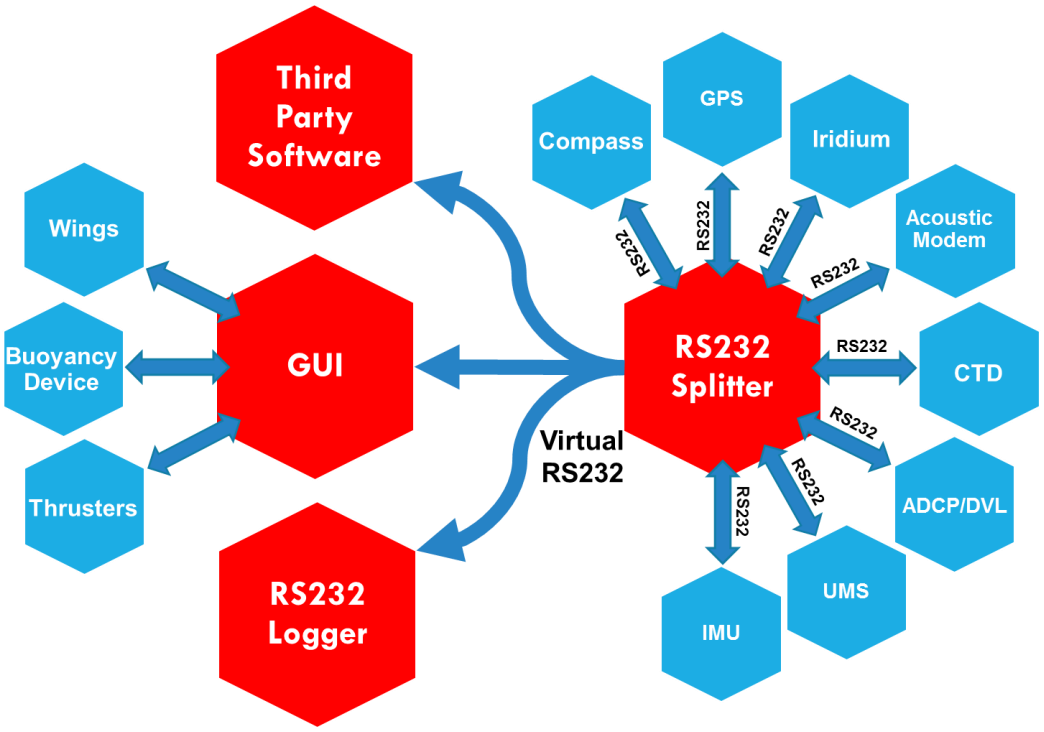



**Fig. 15** SOTAB-I computer

Fig. 15 shows an overview of SOTAB-I computer. The cells in red represent the softwares that are executed during operation and the blue cells corresponds to the peripherals connected to each software. All sensors are connected through RS232 serial communication. The serial data acquisition is detailed in section 2.3.2.2. The actuators control and their sensors' feedbacks are managed by the IO boards as shown in Fig. 6.

#### 2.3.2.1 Robot's GUI

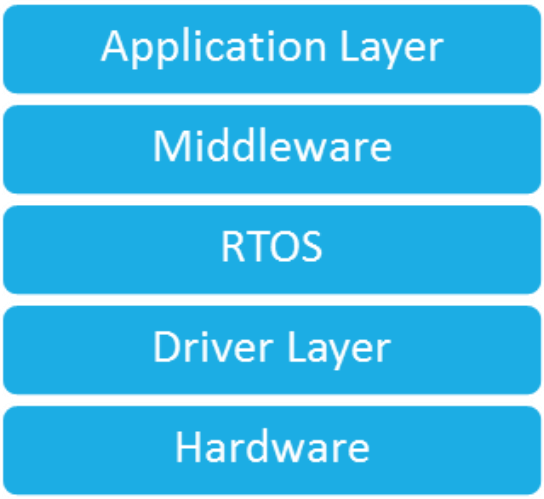



**Fig. 16** Software architecture of SOTAB-I GUI

The GUI software was organized into a hierarchical order as shown in Fig. 16. In this multilayered architecture, responsibilities are distributed. Drivers are the software libraries that directly interface with the hardware and control it. They initialize the device and manage it. The driver layer contains drivers for peripherals such as DAC, ADC, I2C interface, RS-232 interface … etc. The real time operating system (RTOS) can run multiple tasks simultaneously. The task can exchange information through the global variables (Fig. 17). All tasks are executed periodically. A sleep command is used to determine the frequency of the execution of the thread. At the beginning of the execution of the program, every task is initialized. The current thread runs sequentially through the instruction till the end. Then, the execution of the current thread is stopped until at least Tsleep has elapsed. The middleware layer provides services to the upper application to facilitate the communication, input/output, and data management. It is responsible for managing serial communication and LAN and ensures process-to-process communication. The application layer contains all personalized programs of the robot.

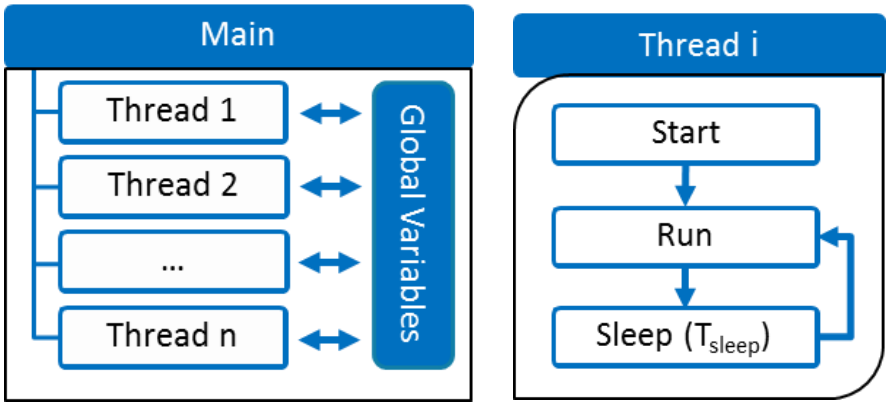



**Fig. 17** Application layer, general structure of thread execution

The program is composed of several files that can be classified into three categories: the header files that include all the function definitions and pre-defined variables; the source files that include the program source of function; and the configuration files, which are the only files that need to be modified by the user.

### 2.3.2.2 Data Acquisition

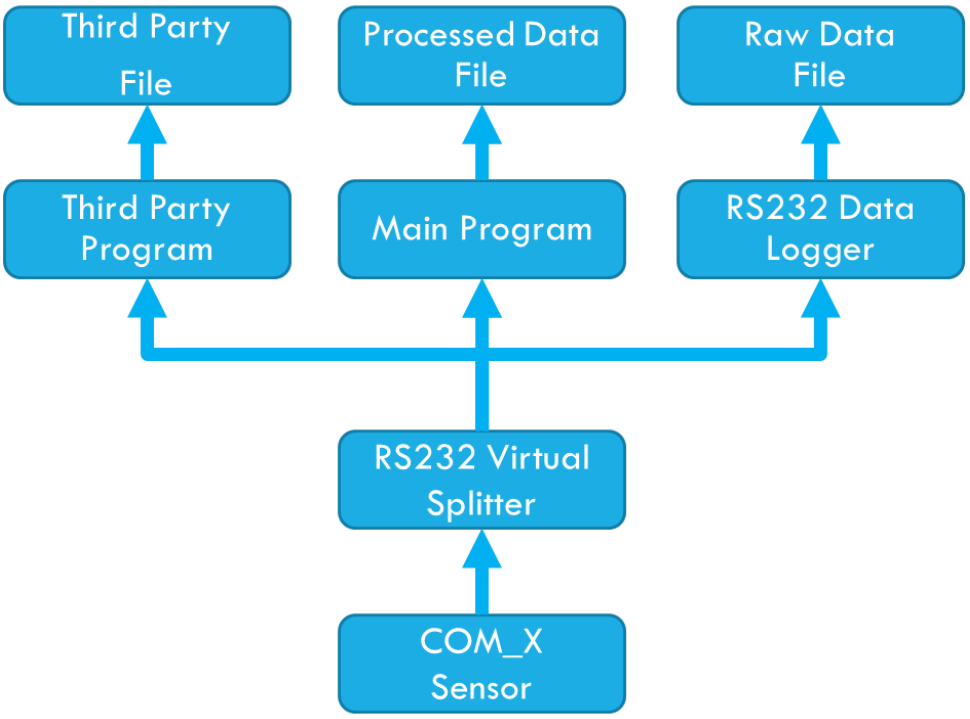



**Fig. 18** Serial data acquisition

The robot processor connects to the sensors through RS-232 serial ports. Because a physical serial port can only be used by a single application at one time, a virtual serial splitter was employed to duplicate the serial data input (Fig. 18). One of the duplicated inputs is directed to a serial logger software to save raw data in a file for ulterior detailed analysis. Another one is sent to the main program for real-time processing. All sensors' data are saved by the GUI every 1 s in their corresponding files with their associated time, which makes it easier to synchronize at post-processing. A third clone of the serial input is directed to a third-party software that is specific to each sensor.

The output data of each sensor was configured as explained next.

##### 2.3.2.2.1 GPS

The GPS receiver of the robot can output data in binary or text formats. The Global Positioning System Fix Data (GGA) text format was selected. Table 45 enumerates the signification of each data field.

##### 2.3.2.2.2 IMU

The IMU sensor can output data in both binary and text formats. Since the binary data can be configured to output more detailed information than the text format, it was selected. To configure the binary output, the configuration register number 75 was set accordingly. Table 46 shows the meaning of each field and

**Table 47** shows the list of data fields than can be outputted. Binary output message configuration command is made in the order listed, with a comma followed by a space between each parameter (Table 48).

#### 2.3.2.2.3 CTD

The CTD provides several possible configurations of the output formats. The "OutputFormat 3 (engineering units in decimal) was selected. In its current configuration, the CTD can output temperature, conductivity and pressure respectively with a comma followed by a space between each data field (Table 49). The device can also be further configured to output salinity and sound velocity.

#### 2.3.2.2.4 ADCP/DVL

The ADCP/DVL device offers several data output configurations in text and binary formats. The selected output format is PD0 (Table 50), which is a binary format that provides the most possible information.

#### 2.3.2.2.5 UMS

Data packet sent from UMS/Data-extraction stage to Robot is formatted as shown in Table **51**. It is a mix of binary and text data. Substances ion intensities are coded in binary format while other data are coded in text format. An example of UMS message is shown in Fig. 19.

```
$002UD, 12/11/05 18:21:45, pwr 00.0 0.0 00.0 0.0, rp 043, tp 10 0025 100, ms 1.13E-05 128
000, sh 14.11 80, vh 000 00, sp 000, th 00.0, ld 0, sv 0, uC 13.7 009, store 000.0, MASS(2)
MASS(4) MASS( 5 )MASS( 12 ) MASS( 14 ) MASS( 15 ) MASS( 16 )MASS( 17 ) MASS(
18 ) MASS(19 ) MASS( 20 )MASS( 27 ) MASS( 28 ) MASS( 29 ) MASS( 30 )MASS( 32 )
MASS( 34 ) MASS(35 ) MASS(37 ) MASS(39) MASS( 40 ) MASS( 41 ) MASS( 42 ) MASS(
43 ) MASS( 44 ) MASS( 45 ) MASS( 46 )MASS( 55 ) MASS( 56 ) MASS( 57 ) MASS( 58
) MASS( 65 )MASS( 67 ) MASS( 69 ) MASS( 70 )MASS( 71 ) MASS(72 ) MASS( 78 )
MASS( 83 ) MASS( 84 ) MASS( 85 ) MASS( 91 ) MASS( 92 )MASS( 106 ) MASS( 128 )
MASS( -1 ) MASS( -8 )END<CR> <LF>
```

**Fig. 19** Example of UMS Data

### 2.3.2.3 Data Filtering

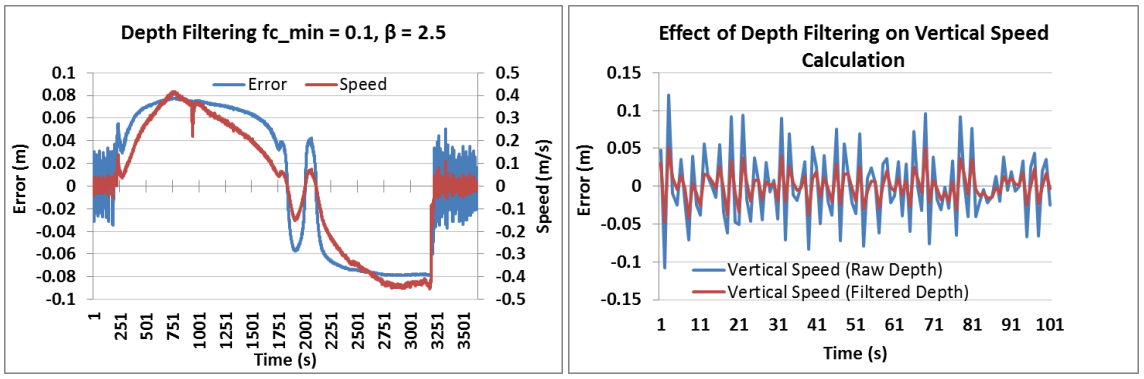



**Fig. 20** Effect of filtering using 1 € filter

One Euro filter (Casiez et al, 2012) was implemented for filtering the depth and speed. It is based on a first order low pass fill whose cutoff frequency is adaptive to the signal speed. At high speed, the cutoff frequency is increased in a way to reduce the lag. At low speed, the cutoff frequency is decreased leading to the reduction of jitter. The algorithm is characterized by its easy implementation and it's a good compromise Jitter/Lag.

There are two main parameters that determine the performance of the filter: fc_min which is the minimum cutoff frequency and beta. At first stage, beta is set to 0 and fc_min is determined at low speed signal in a way to have reasonable compromise Jitter/Lag. Once fc_min determined, beta value is adjusted using high speed signal.

As the accuracy of the CTD depth is better than 8cm, we determined the filter parameters in a way that the maximum difference between the filtered signal and the measured signal is less than 8cm up to 0.4m/s speed.

### 2.3.2.4 Devices Drivers

Drivers are programs that enable to simplify the operation of devices. Following are examples of software drivers developed. The first is used to control the buoyancy device, the second is used to acquire data from the ADCP/DVL sensor.

#### 2.3.2.4.1 Buoyancy Device Driver

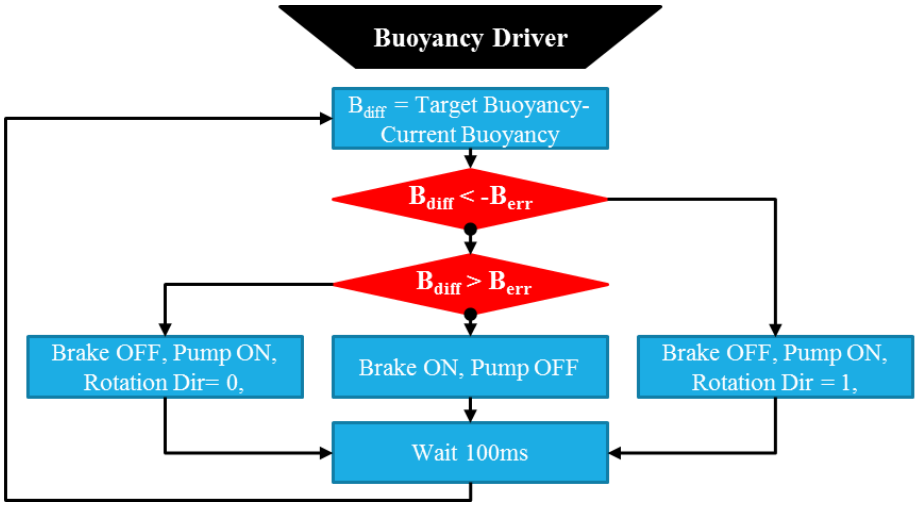



**Fig. 21** Buoyancy control driver flowchart

Fig. 21 shows the driver program flowchart used to control the buoyancy device where Berr represents the maximum value of random error on the oil level sensor. The role of each component of the buoyancy device are explained in section 2.2.3.1.

#### 2.3.2.4.2 ADCP/DVL

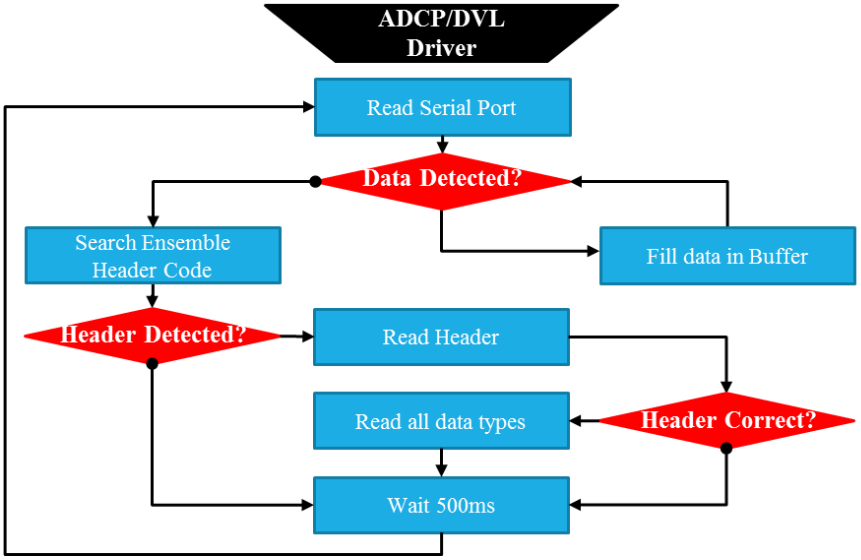



**Fig. 22** ADCP/DVL data acquisition flowchart

Fig. 22 depicts the flowchart of the acquisition of the ADCP/DVL data under the PD0 format (Table **50**).

### 2.3.3 Acoustic Communication

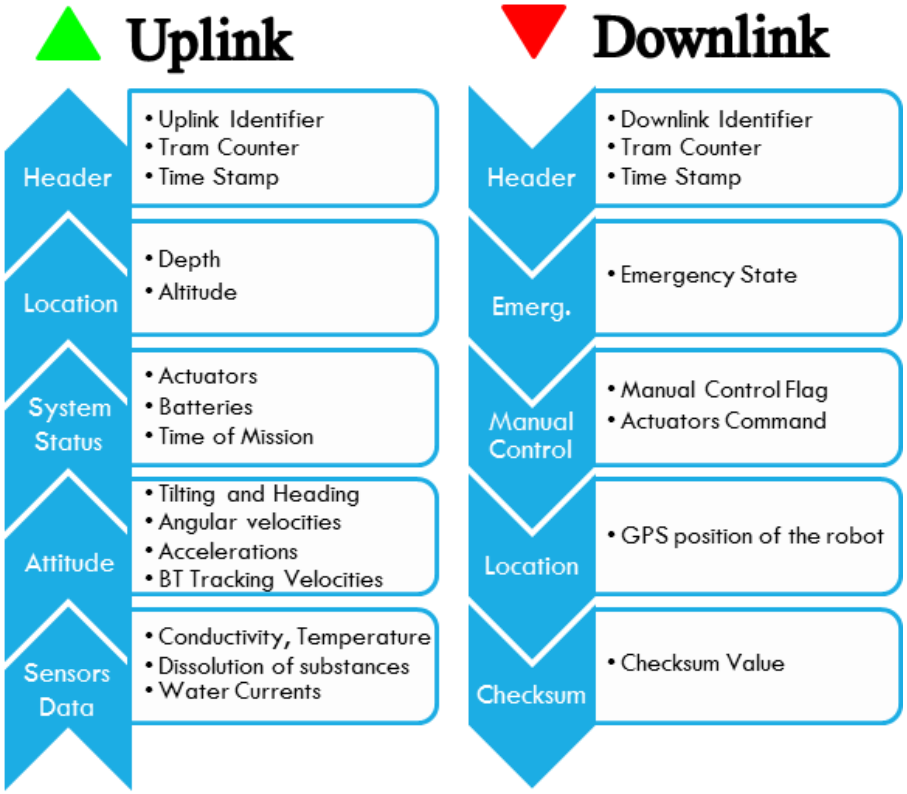



**Fig. 23** Acoustic communication data frame

The acoustic data exchange format between the ship and the SOTAB-I is defined as shown in Fig. 23. Downlink data are used to command the SOTAB-I. In addition, it is used for sending the robot's absolute underwater position. For that reason, downlink data are sent continuously. At the beginning of the experiment, SOTAB-I and the ship computer's times are synchronized. Using the time stamp that is associated with each downlink data, it is possible to determine the shifting between the time in which the robot position was determined on the ship, and the time when it was received by the robot. Uplink data include sensors data and actuators states to keep the user informed about the robot situation and the changes in its environment. Downlink data are also feedback on the uplink data to verify the authenticity of the communication and whether noise occurred

during data submission. The ASCII format was selected for coding data. This gives the possibility to read and send the data simply by using a serial terminal. Additionally, data can be interpreted without a specific parser. The binary format gives the advantage of sending a bigger amount of data. However, uplink and downlink data exchange rates are sufficient to send all important data under the text format.

During the experiments, it was noticed that additional bytes are sometimes injected into the downlink data. Since the downlink data are used to control the actuators of the robot, noise injected into the data may lead to a misinterpretation of the orders and cause serious damage. For instance, if the robot receives an order to move the vertical thruster in the backward direction, the robot will start diving very fast to the seabed, and it may be impossible to recover it. To enhance the immunity of the system against mistaken orders caused by noise, a checksum algorithm was used for downlink data. The integer addition checksum was employed. It is a simple algorithm that reduces considerably the probability of undetected errors. It can detect all single bit errors and all error bursts of length 16 bits or less. The percentage of undetected 2-bit errors over the total number of 2-bit errors is less than 3% for a message length equal to 80 bytes (Maxino et al., 2009).

## 2.4 Conclusions

In the first part of this chapter, the outline of SOTAB-I and its general characteristics were described and the main features of its internal devices were presented. SOTAB-I, is an autonomous profiling buoy for water surveying and oil spill sampling. Motion is achieved through the use of rotational fins in the horizontal plane, and through the buoyancy control device in the vertical plane. Additionally, the system possesses thrusters that ensure the

same functions with the benefit of faster reaction, but at the cost of higher power consumption. SOTAB-I is equipped with a sensor suite for measurements of oceanographic data as well as physical and chemical characteristics of an oil plume. The robot navigation is ensured through the GPS receiver and the USBL allows tracking of the platform respectively on the sea-surface and underwater.

In the second part of this chapter, the SOTAB-I software organization and it is developed algorithms were presented. On the ship side, we designed and developed a GUI, which proved to be a valuable tool to keep the user informed about the status of the robot and the changes around its environment. Additionally, it allowed to send commands when necessary. On the robot side, the multilayered architecture of the SOTAB-I software enabled distribution of responsibilities and enhanced the clarity of the program and its flexibility to changes. The program code optimization helped to reduce the processor usage. This contributed to the reduction in power consumption and prevented the processor from overheating inside the pressure cell. Software drivers were designed to assure the easy control of the actuators and the acquisition of sensors data. Furthermore, we configured each sensor in a way to guarantee the fastest sampling period to enable to get the maximum amount possible of important data. To enhance the immunity of the acquired data against fluctuations, we implemented software filters, which had a decisive impact in smoothing the measured signals while keeping a good tradeoff between jitter and lag. To ensure the transmission of critical information to the user on the mother ship, we established an acoustic communication data frame. In addition, we implemented a checksum verification method to secure the control of the robot through downlink, giving it an immunity against noise.

# Chapter 3: Navigation



`

# 3 Navigation

## 3.1 Introduction

The enhancement of the robot's navigation has considerable impact on expanding its capabilities and its value to oceanography. For instance, the robot absolute position is needed for several purposes, notably for robot guidance and water survey. Robot's control relies on underwater position to determine the direction and distance in which the robot should go. The position of the robot will also be needed for mapping purposes: The concentration of oil substances will be assigned to their absolute position. Then, it is possible to map the distribution of oil in the area and use that to develop a strategy to track the oil slick or find the blowout gas source. Besides, the ADCP sensor is only able to measure the relative water currents when the bottom tracking is active, which represents only a narrow region of the whole surveyed water column. Hence, the determination of the speed of the robot is needed to accurately determine the absolute vertical column distributions of water currents.

The way of tracking the robot differs depending on the water region. When the robot is on the sea-surface, SOTAB-I's GPS receiver can be used to determine the robot position. However, the high-frequency waves of the global positioning system (GPS) have very low penetrability into water. Instead, acoustic-based positioning systems (APS) are used widely for underwater tracking. Several APS techniques can be used (Vickery, 1998). Long baseline systems (LBL) have very good accuracy, but they require the deployment of at least 3 transponders on the seafloor. Short baseline systems (SBL) don't require the deployment of underwater transponders, but at least 3 ships mounted transceivers instead which make them easier to implement. However, they are less accurate than LBL systems

and their accuracy is dependent on the length of the baseline. Ultra-short baseline (USBL) system consists of a tightly integrated transducer array with a baseline distance less 10 centimeters that can be used in a small ship. The way of determining the underwater position in the USBL system is different from the LBL and SBL systems: It calculates the range and bearing of the robot with regard to the ship rather than trilateration. USBL systems, even though they require much calibration and are less accurate than LBL systems, has an acceptable range and accuracy. In not only as back up, but also to boost the accuracy and estimate instantaneous addition, they have the advantage of being very easy to deploy because they only require a single transceiver on the ship side and one transponder installed on the robot side. For the latter mentioned reasons, USBL system was selected as the APS for tracking the robot underwater. Some difficulties to track the robot underwater are faced, such as when the robot goes out of the range of the acoustic transceiver or when the ship motion introduces inaccuracies in the positioning signal. For the latter mentioned reasons, additional alternatives should be used to estimate the robot's positions between APS samples. There is no perfect technique that can surmount all challenges for underwater position in GPS-denied systems, but it is more about selecting the suitable techniques based on range, accuracy and time necessary to set up the whole system.

In the first part of this chapter, the performance of the navigation sensors at-sea experiments are analyzed. Next, we define the navigation regions of SOTAB-I and enumerate the sensors involved in each region as well as the tracking method that will be used. The last part deals with the Kalman filter that was implemented for data fusion. The predicting and filtering performances of the filter are studied. In this chapter, we

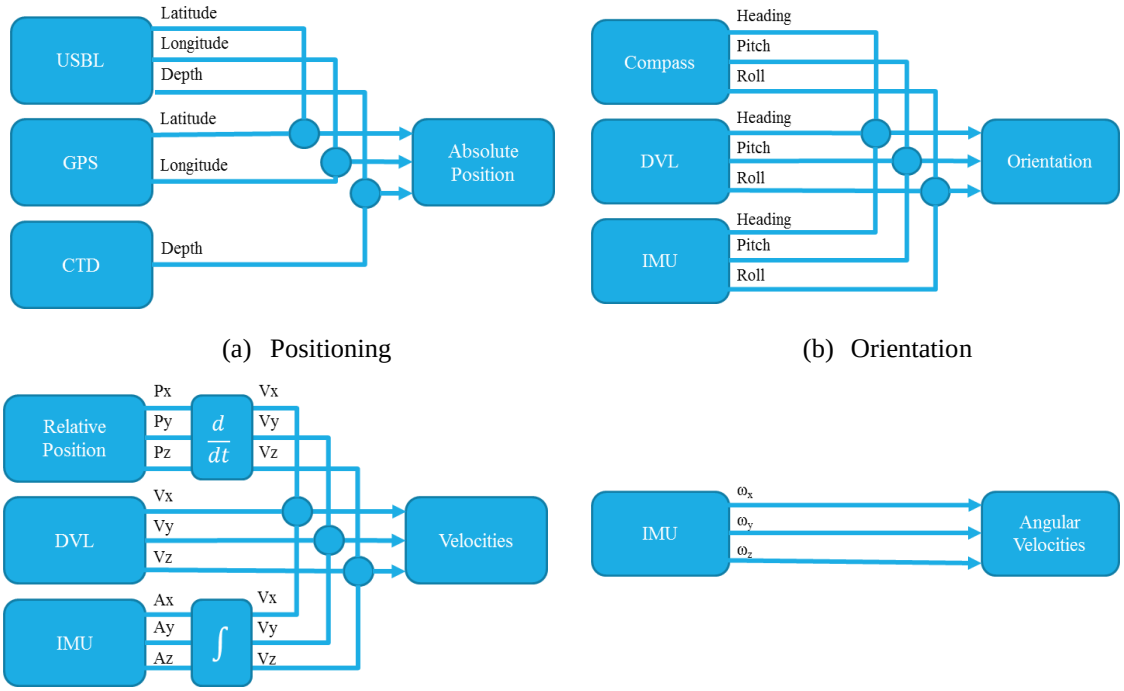
exclusively focused on the navigation data collected in the second experiment in Toyama Bay on 17/03/2016, in which the maximum depth reached by the robot at the time when this thesis was written. In the latter experiment, the robot could get close up to 8m altitude from the seabed, enabling to collect additional information of the bottom tracking velocities.

## 3.2 Navigation Overview

### 3.2.1 Sensors Involved in the Navigation

The robot's position can be determined based on the GPS, USBL and the CTD sensors (Fig. 24 (a)). On the sea surface, the robot's position in the horizontal axis can be determined using the GPS receiver. It happens also that the USBL system determine the robot position when it is on the sea surface. In such case, the GPS position will be used as a reference since it is more accurate and it is updating rate is faster. When the robot is underwater, the GPS is not operational anymore, and then only the USBL system can be used to determine the robot's horizontal position. In the vertical plane, the depth data calculated based on the pressure measurement of the CTD sensor will be employed since it is much more accurate than the depth data measured by the USBL system and it is sampling time is shorter. The robot's orientation can be determined either by the IMU sensor or by the DVL device (Fig. 24 (b)). In terms of accuracy and sampling time, the orientation data of the IMU are better than DVL and hence will be utilized in the first place. For robot's motion, velocities toward the 3 axis can be determined by integrating accelerometer data of the IMU (Fig. 24 (c)). When the altitude from the seabed is within bottom tracking range, the DVL becomes able to determine robot's velocities in the 3 axis. A third way is to convert the absolute position obtained from USBL or GPS to local

coordinate, then calculate the motion speed referring to the previous location. Same as for robot orientation, the IMU sensor's data will be used in the first place. Angular velocities around the X and Y axis can be obtained directly from the gyro of the IMU sensor (**Fig. 24** (d)).




(a) Positioning (b) Orientation

(c) Velocities (d) Angular velocities

**Fig. 24** Robot's motion and orientation

### 3.2.2 Shortcomings of the Employed Sensors

To study the shortcomings of the navigation sensors employed, we used data from the second experiment that was conducted in Toyama Bay on 17/03/2016.

#### 3.2.2.1 GPS Sensor

The GPS receiver will be used for tracking the robot on the sea surface. However, the GPS antenna needs to be above the seawater level to be able to work properly.

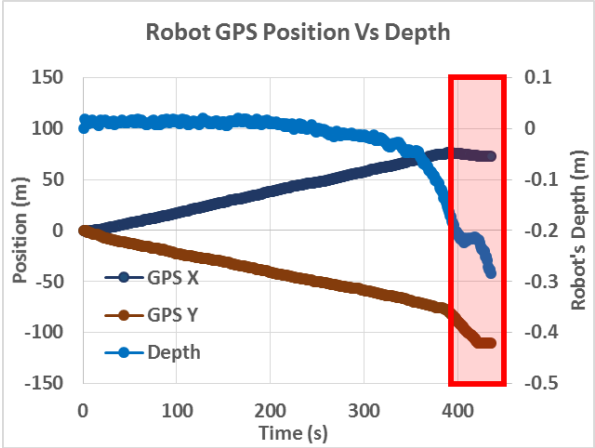



**Fig. 25** GPS position Vs Depth

**Fig. 25** shows that the rate of variation in the X and Y directions changes starting from 0.2 water depth. This corresponds to the point where the GPS antenna is fully submerged in sea water. Hence, the GPS data is not reliable starting from that point and a dead reckoning method should be used to predict the position of the robot.

### 3.2.2.2 USBL Positioning

#### 3.2.2.2.1 Spikes and Ambient Noise

Experimental data show the existence of spikes and ambient noise in the bearing, range and depth signals necessary to determine the local position of the robot

There are 3 main origins of this noise:

- Ambient noise (NA): This is generated by external factors such as waves, wind, rain etc.
- Self-noise (NS): This includes noise, which is entirely generated as part of the offshore operation and includes propulsion, machinery, hydrodynamic (flow), and circuit noise
- Reverberation noise (NR): Reverberation arises as a direct consequence of using an acoustic positioning system

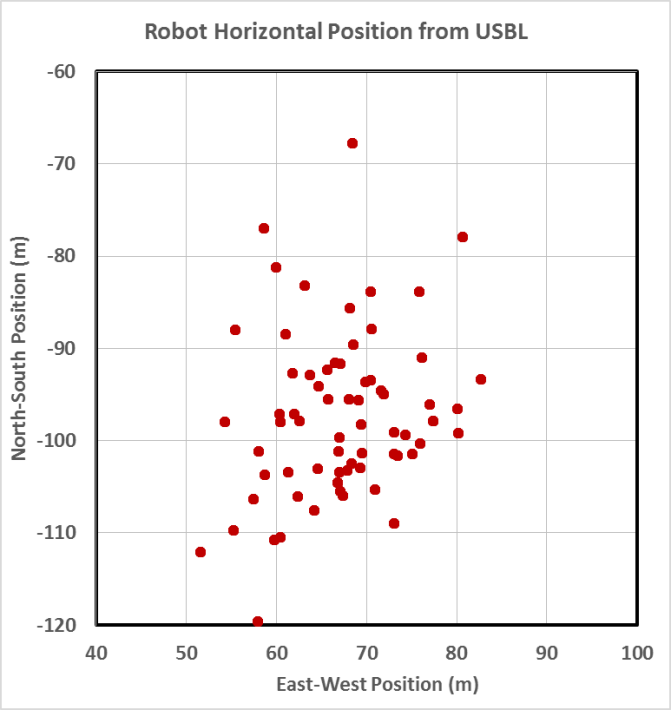



**Fig. 26** Robot's underwater positions near the seabed

Fig. 26 shows the underwater position of SOTAB-I near the seabed in the second experiment that was conducted in Toyama Bay on 17/03/2016. In that experiment, the robot dived up to 756m, and stopped at around 8m altitude from the seabed. At such low altitude, we can assume that the water currents are negligible. Hence, the robot horizontal position can be assumed as fixed. However, as it can be observed in Fig. 26, the raw horizontal position varied considerably up to 50m between extremities positions. Therefore, there is a necessity to implement a filter to absorb spikes and ambient noise.

#### 3.2.2.2.1 Out of Range

When the vertical angle is higher than the beam angle, the measurements become not accurate as if can be shown in the red zone in Fig. 27. The vertical angle corresponds to the angle that between the vertical axis of the ship and the robot. The beam angle of the transceiver mentioned in the datasheet is 60 degrees.

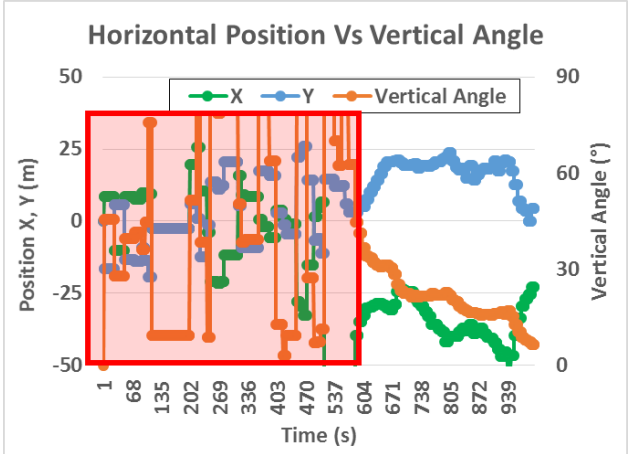



**Fig. 27** Position's accuracy Vs Vertical angle

This case always happens when the robot is still near the sea surface. The zone where the robot is underwater, but out of range is called "Signal dead zone" as shown in Fig. 31. Since the GPS and the USBL are not operational in that zone, a dead reckoning method based on ADCP aiding is used to predict the robot position.

#### 3.2.2.2.2 Sound Speed Variation

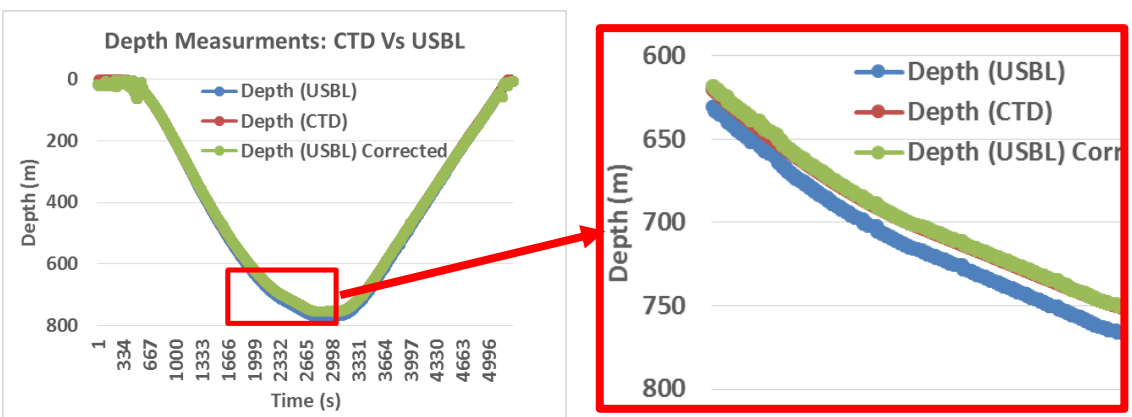



**Fig. 28** Effect of the sound speed on the depth measurement

In the tracklink software, the sound speed is fixed to 1500m/s. However, in real experiments the sound speed value varies from the sea surface to the seabed, which automatically leads to errors in the position measurement. Fig. 28 shows that the effect of the sound speed variation gradually influences the position measurement. The effect

becomes more noticeable at a water depth beyond 600m. The maximum shifting reported at 755m depth was 15m from the CTD measured depth data. To solve this problem, an offline sound speed correction was performed and showed its efficiency by reducing the shifting up to only 0.5m.

#### 3.2.2.2.3 Acoustic Positioning Rate

Table 4 summarizes the acoustic positioning performance.

**Table 4** Acoustic positioning performance

| Period (s) | Occurrence | Percentage |
|---|---|---|
| 8 | 532 | 93.66% |
| 16 | 31 | 5.46% |
| 24 | 4 | 0.70% |
| 32 | 0 | 0.00% |
| 40 | 1 | 0.18% |
| Total | 568 | 100.00% |
| | | |
| Success | 532 | 92.52% |
| Failure | 43 | 7.48% |
| Total | 575 | 100.00% |

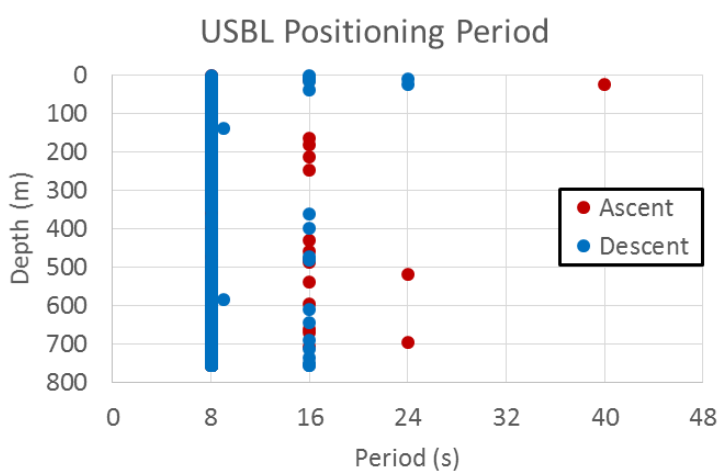



**Fig. 29** Acoustic positioning period

The USBL system positioning sampling time is 8s. However, and as it can be observed in Fig. 29, the positioning failed sometimes.

### 3.2.2.3 Acoustic Communication

Table **5** displays the performance of the acoustic communication in the downlink direction.

**Table 5** Downlink acoustic communication performance

| Period (s) | Occurrence | Percentage |
|---|---|---|
| 8 | 485 | 86.92% |
| 16 | 64 | 11.47% |
| 24 | 8 | 1.43% |
| 32 | 1 | 0.18% |
| **Total** | 558 | 100.00% |
| | | |
| Success | 485 | 85.39% |
| Failure | 83 | 14.61% |
| **Total** | 568 | 100.00% |

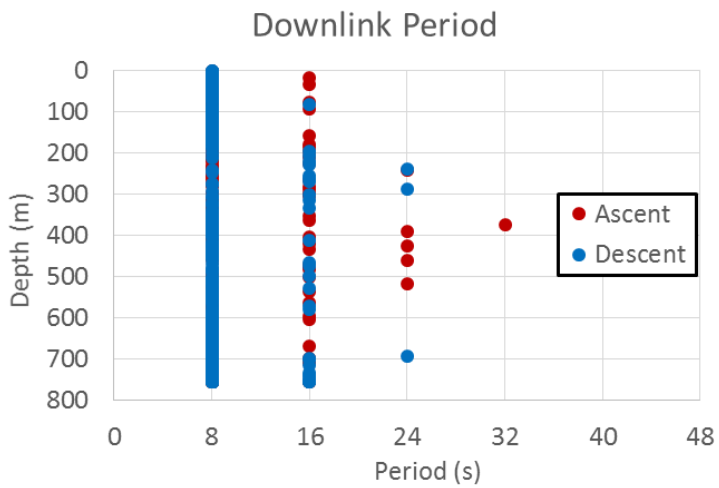



**Fig. 30** Acoustic communication period

The downlink data are sent periodically from the ship side to the robot side to update the robot with its current absolute position. Similar to the positioning, the minimum downlink rate is equal to 8s. Fig. 30 shows that there are some cases where the downlink communication wasn't successful. Hence it is important to develop an algorithm that is able to predict the robot position when such failures happen.

### 3.2.3 Navigation Regions

The sensors employed in the navigation depend on the vertical position of the robot and its distance from the ship. Fig. 31 shows the navigation regions of SOTAB-I.

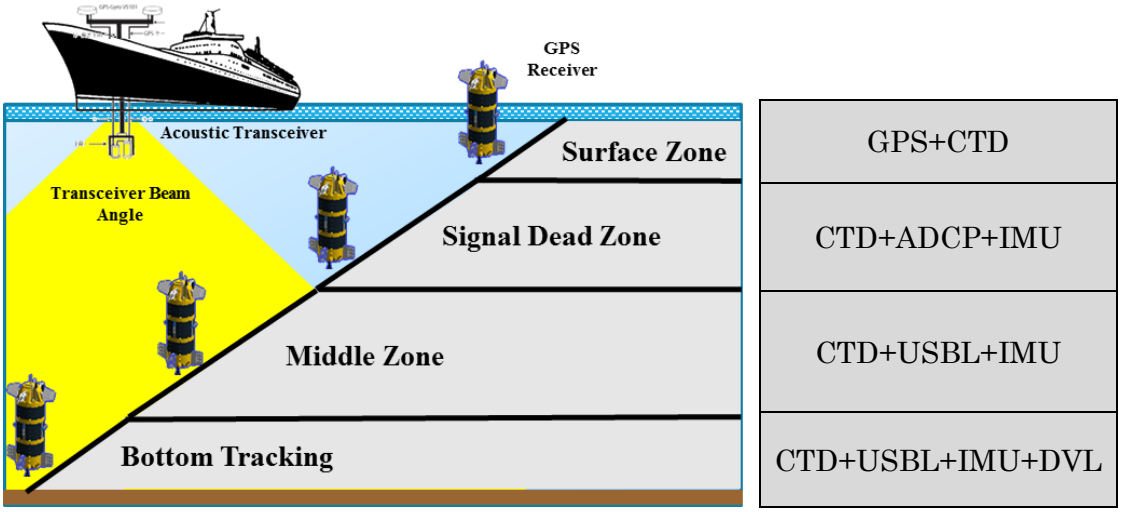



**Fig. 31** Navigation regions

#### 3.2.3.1 Surface Zone

On the sear surface, the robot localization is entirely performed through the GPS receiver until the antenna is submerged in the seawater. On the sea surface, it is possible to measure the average of the absolute water current from 3m, which corresponds to the depth of the first layer, up to the maximum range of the ADCP sensor.

#### 3.2.3.2 Signal Dead Zone

When the robot start diving and the GPS receiver antenna is underwater, the robot enters what we call “Signal dead zone” in which the robot is neither able to get the GPS position nor to perform acoustic positioning. The dead-reckoning method can be used to predict the next position based on current available position, speed and orientation using integrated inertial navigation system (INS). However, it has long term gradual error accumulation over time. To reduce the margin of error accumulation, the dead reckoning

algorithm was enhanced by ADCP aided navigation (Medagoda, et al., 2011).

#### 3.2.3.3 Middle Zone

Once the robot is within the beam angle range, acoustic positioning becomes operational and the robot's position is updated periodically through downlink data. In this zone the dead reckoning method is still used to estimate the robot's position between samples.

#### 3.2.3.4 Bottom Tracking Zone

When the robot is within bottom tracking range, the robot velocities as well as its altitude become known. Combining a USBL position fix with DVL velocities enables to enhance the accuracy of the predictions of the robot's position.

## 3.3 The Kalman Filter

### 3.3.1 Overview

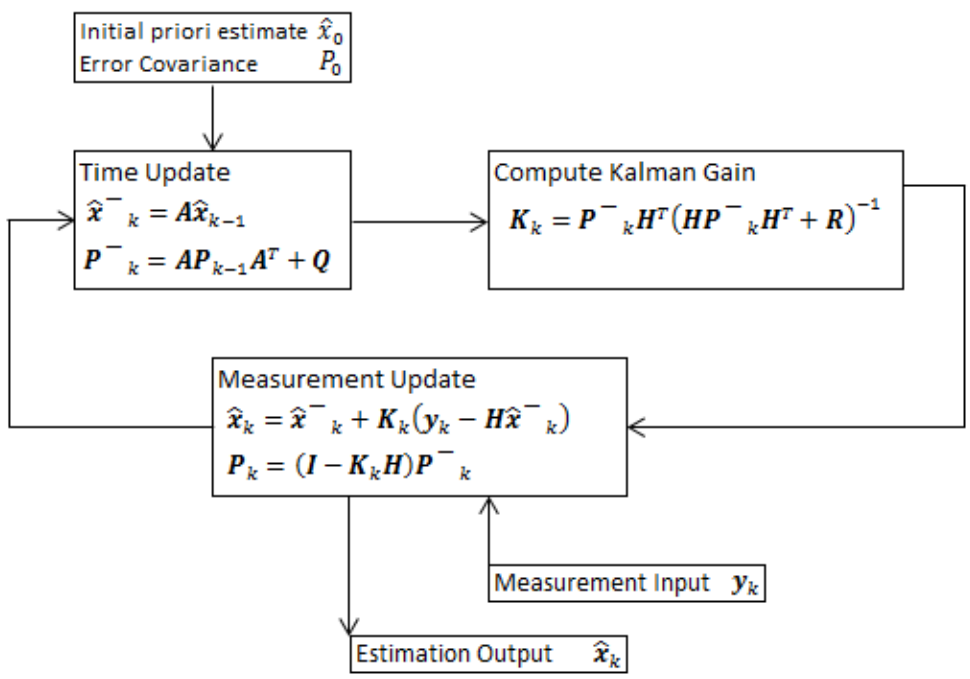



**Fig. 32** The Kalman filtering algorithm

**Table 6** The Kalman filter parameters' definition

| Symbol | Definition |
|---|---|
| $\boldsymbol{x}_k, \boldsymbol{x}_{k-1}$ | The state vector of the process respectively at time k and k-1 |
| $\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{x}}_{k-1}$ | A posteriori state estimate at step k |
| $\hat{\boldsymbol{x}}^-_k$ | A priori state estimate at step k |
| $\boldsymbol{y}_k$ | Observation vector at time k |
| $\boldsymbol{A}$ | The state at time step k to the state at step k+1 |
| $\boldsymbol{P}_k, \boldsymbol{P}_{k-1}$ | A posteriori estimate error covariance |
| $\boldsymbol{P}^-_k$ | A priori estimate error covariance |
| $\boldsymbol{H}$ | Design matrix |
| $\boldsymbol{K}$ | Kalman gain matrix |
| $\boldsymbol{Q}$ | The process noise covariance matrix |
| **R** | The measurement noise covariance matrix |

The Kalman filter consists of two steps of the prediction step and the filtering step. The KF at predicts the new state and its uncertainty, then works on correcting with the new measurement. Fig. 32 shows the Kalman filtering algorithm with its parameters defined in Table 6. More details on Kalman filter can be found in (Kalman, 1960).

Next, we describe the KF parameters. State vector x represents the position, velocity and the acceleration following the 3 axis directions X, Y and Z, with nine elements in total. X refers to the east-west, Y to the north-south and z to the down-up directions

$$\boldsymbol{x} = [P_X \quad P_Y \quad P_Z \quad V_X \quad V_Y \quad V_Z \quad A_X \quad A_Y \quad A_Z\,]^T \tag{1}$$

The state transition matrix A is a matrix of 9 × 9, and it is expressed by the following equations.

In the case where the IMU data are used:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{I_3} & \boldsymbol{I_3}.\boldsymbol{dt} & \frac{\boldsymbol{1}}{\boldsymbol{2}}.\boldsymbol{I_3}.\boldsymbol{dt^2} \\ \boldsymbol{0_3} & \boldsymbol{I_3} & \boldsymbol{I_3}.\boldsymbol{dt} \\ \boldsymbol{0_3} & \boldsymbol{0_3} & \boldsymbol{I_3} \end{bmatrix} \tag{2}$$

If the IMU data are not used:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{I_3} & \boldsymbol{I_3}.\boldsymbol{dt} & \boldsymbol{0_3} \\ \boldsymbol{0_3} & \boldsymbol{I_3} & \boldsymbol{0_3} \\ \boldsymbol{0_3} & \boldsymbol{0_3} & \boldsymbol{I_3} \end{bmatrix} \tag{3}$$

Where

$$\boldsymbol{0_3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \boldsymbol{I_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$



`

The process noise matrix is defined as follows:

$$\boldsymbol{Q} = \begin{bmatrix} q_i.\boldsymbol{I_3} & \boldsymbol{0_3} & \boldsymbol{0_3} \\ \boldsymbol{0_3} & q_{i.}.\boldsymbol{I_3} & \boldsymbol{0_3} \\ \boldsymbol{0_3} & \boldsymbol{0_3} & q_i.\boldsymbol{I_3} \end{bmatrix} \tag{5}$$

Where $q_i$ $(i = x, y, z)$ is the tuning parameter.

The noise covariance matrix ($\boldsymbol{Q}$) tells which to trust more: the measurements or the system dynamics. When Q is large, the Kalman filter trusts more the measurements and thus, tracks substantial changes in the data more tightly which may make the estimates noisy, but with no time lag. On the other hand, when Q is low, the time lag is high, but the result is less sensitive to fluctuations. So we can state that the choice if q is a tradeoff between the lag and the jitter.

R determines to which extent the measurements are accurate. When the value of R is high, this means that the measurements as not very accurate. If R is small, the Kalman filter will follow the measurements more tightly since they are accurate. The measurement noise covariance ($\boldsymbol{R}$) can be written as follows:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R_P} & \boldsymbol{0_3} & \boldsymbol{0_3} \\ \boldsymbol{0_3} & \boldsymbol{R_V} & \boldsymbol{0_3} \\ \boldsymbol{0_3} & \boldsymbol{0_3} & \boldsymbol{R_A} \end{bmatrix} \tag{6}$$

Where $\boldsymbol{R_P}$, $\boldsymbol{R_V}$ and $\boldsymbol{R_A}$ correspond respectively to the observation error covariance of the position, the speed and the acceleration following the 3 axis X, Y and Z.

$$\boldsymbol{R_P} = \begin{bmatrix} r_{P_X} & 0 & 0 \\ 0 & r_{P_Y} & 0 \\ 0 & 0 & r_{P_Z} \end{bmatrix}, \boldsymbol{R_V} = \begin{bmatrix} r_{V_X} & 0 & 0 \\ 0 & r_{V_Y} & 0 \\ 0 & 0 & r_{V_Z} \end{bmatrix}, \boldsymbol{R_A} = \begin{bmatrix} r_{A_X} & 0 & 0 \\ 0 & r_{A_Y} & 0 \\ 0 & 0 & r_{A_Z} \end{bmatrix} \tag{7}$$

$r_{\mathrm{i}}$ corresponds to the observation error covariance of the parameter $i$.

Q and R values are very critical to determine the overall performance of the Kalman filter. Their values can be determined using the trial and error method.

The initial value $\boldsymbol{P_0}$ of the covariance matrix $\boldsymbol{P}$ is assigned as follows:

$$\boldsymbol{P_0} = \boldsymbol{I_9} \tag{8}$$

The observation vector y represents the sensors measured values of position, speed and acceleration following the 3 directions. It can be written as follows:

$$\boldsymbol{y} = [p_x \quad p_y \quad p_z \quad v_x \quad v_y \quad v_z \quad a_x \quad a_y \quad a_z\,]^T \tag{9}$$

The design matrix H is given by the following formula:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H_P} & \boldsymbol{0_3} & \boldsymbol{0_3} \\ \boldsymbol{0_3} & \boldsymbol{H_V} & \boldsymbol{0_3} \\ \boldsymbol{0_3} & \boldsymbol{0_3} & \boldsymbol{H_A} \end{bmatrix} \tag{10}$$

Where $\boldsymbol{H_P}$, $\boldsymbol{H_V}$ and $\boldsymbol{H_A}$ correspond respectively to the design matrices of position, speed and acceleration following the 3 axis X, Y and Z.

$$\boldsymbol{H_P} = \begin{bmatrix} h_{P_X} & 0 & 0 \\ 0 & h_{P_Y} & 0 \\ 0 & 0 & h_{P_Z} \end{bmatrix}, \boldsymbol{H_V} = \begin{bmatrix} h_{V_X} & 0 & 0 \\ 0 & h_{V_Y} & 0 \\ 0 & 0 & h_{V_Z} \end{bmatrix}, \boldsymbol{H_A} = \begin{bmatrix} h_{A_X} & 0 & 0 \\ 0 & h_{A_Y} & 0 \\ 0 & 0 & h_{A_Z} \end{bmatrix} \tag{11}$$

$h_{\mathrm{i}}$ corresponds to the state of observation of the parameter $i$.

For the positioning, since we can determine the robot's depth from the CTD sensor every time, then $h_{P_Z} = 1$. When the GPS or the USBL system output the robot's location, then $h_{P_X} = h_{P_Y} = 1$. Otherwise, $h_{P_X} = h_{P_Y} = 0$.

Concerning the speed, if bottom tracking is active then, $\boldsymbol{H_V} = \boldsymbol{I_3}$. Otherwise, $\boldsymbol{H_V} = \boldsymbol{0_3}$.

Finally, since the IMU sensor's acceleration data is always available, then $\boldsymbol{H_A} = \boldsymbol{I_3}$.

### 3.3.2 Application of the Kalman Filter to Measured Experiments Results

In this experiment, the IMU data collected didn't reflect the robot motion, that's why acceleration data weren't used. USBL data were considered starting from 5m. GPS positions when the antenna was submerged in seawater were not counted. In the dead signal zone, ADCP aided INS was used.

Following is the Q matrix configuration:

$$\boldsymbol{Q} = \boldsymbol{0.0005} * \boldsymbol{I_9} \tag{12}$$

**Table 7** Configuration of the R matrix

| | |
|---|---|
| $r_{P_X}$ | If GPS is used $r_{P_X}$ = 1.84, if USBL is used $r_{P_X}$= 625 |
| $r_{P_Y}$ | If GPS is used $r_{P_Y}$ = 1.84, if USBL is used $r_{P_Y}$= 625 |
| $r_{P_Z}$ | 0.007 |
| $r_{V_X}$ | 0.006 |
| $r_{V_Y}$ | 0.006 |
| $r_{V_Z}$ | 0.003 |

In the Kalman filter, there are two matrices Q and R whose coefficients should be adjusted at the beginning. At the beginning, the $r$ values of the R matrix are initiated to the value of the variance of their associated sensor, then they are adjusted using trial and error method. Q is determined exclusively by tuning the q parameter in a way to ensure a good compromise between jitter and lag.

### 3.3.2.1 Filtering Performance

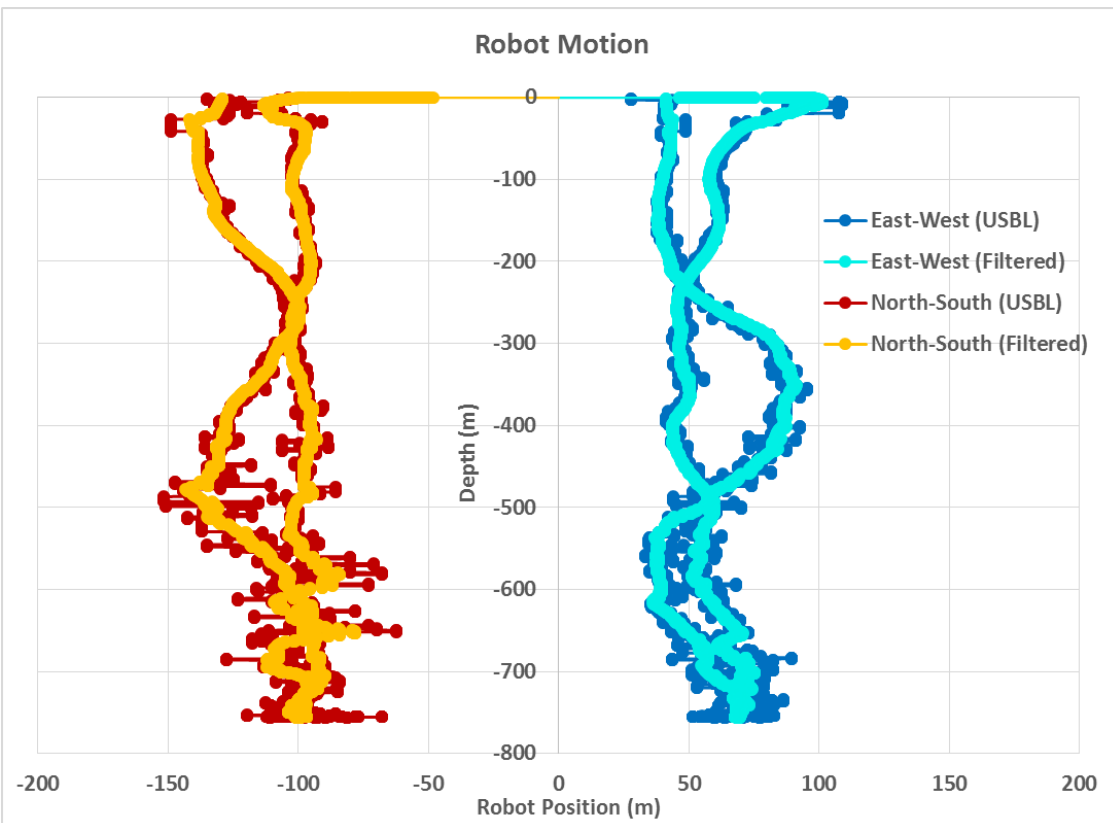



**Fig. 33** Robot's positions and effect of the Kalman filtering

Fig. 33 compares the raw robot underwater position measured by the USBL system with the filtered position using the Kalman filter along the water column in the East-West and North-South directions. From the obtained result, we can conclude that the Kalman filter demonstrated its ability to filter spike noise on the USBL data without significant lagging while keeping the same trend.

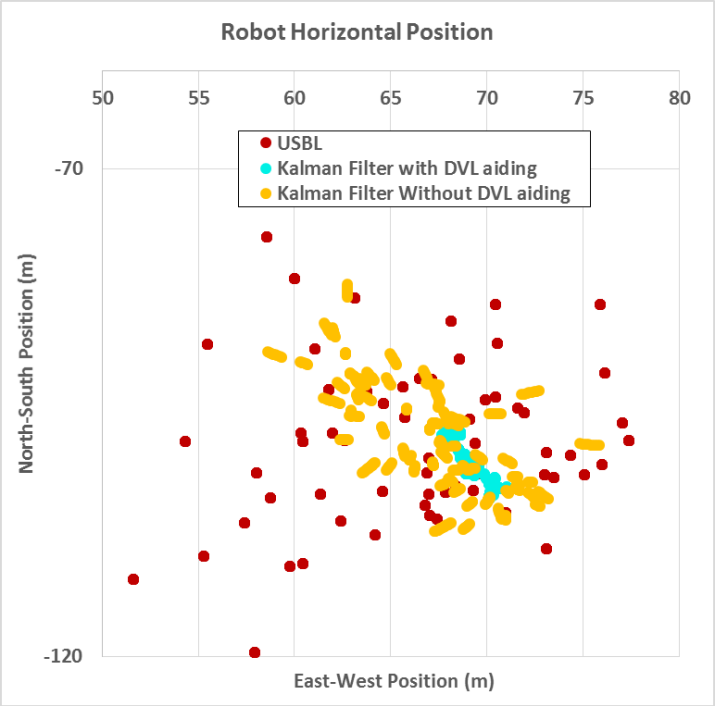



**Fig. 34** Variations of the robot's positions near the seabed

In this experiment, the robot went up to 755m, and stopped at around 8m altitude from the seabed. At such low altitude, we can assume that the water currents are negligible. Hence, the robot horizontal position can be assumed as fixed. Fig. 34 depicts the horizontal position of the robot and the effect of the filtering. It shows that the filter reduced the fluctuation of the raw USBL data from 25m to 15m without DVL aided navigation, and to 5m when USBL position and DVL bottom tracking velocities are combined.

### 3.3.2.2 Sensor's Data Fusion

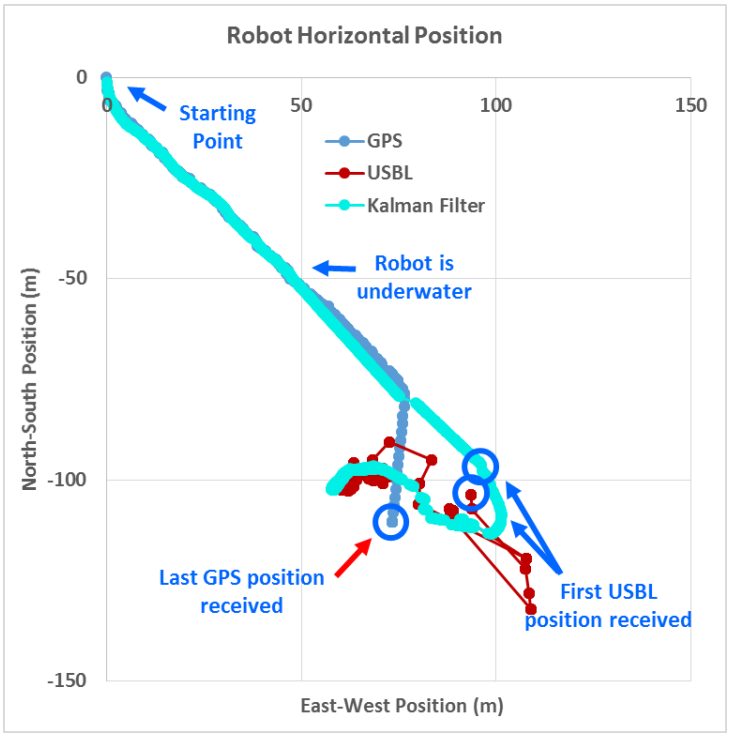



**Fig. 35** Sensor fusion using Kalman filter

When the robot is on the sea surface, the robot position is determined by the GPS receiver. Once the GPS antenna starts to be submerged in the water, the GPS position becomes not reliable and the robot enters an intermediary zone in which neither the GPS nor the USBL systems can be used to track the robot. In this particular zone, that we called "the signal dead zone", the robot uses dead reckoning method to predict the robot position until the robot goes deep enough to enter the beam range of the USBL system. Fig. 35 depicts the role of the Kalman filter in sensor data fusion. It shows that the algorithm succeeded to merge the dead reckoning predictions with the GPS positions at the first stage, the merged dead reckoning predicted position with the USBL position at the second stage. At the end, a combined robot position graph between GPS, USBL and dead reckoning positions was obtained.

#### 3.3.2.3 Position Prediction

The sea surface zone and the zone near the sea bed provide the best locations to evaluate the performance of the predictions. When the robot start diving, the dead reckoning method can be used to predict the motion of the robot till the USBL tracking becomes operational. Hence, if we compare the distance of the dead reckoning predicted solution with the first reliable USBL position we can judge the performance of the predicting algorithm.

Same when the robot is near the seabed and within bottom tracking range, the DVL, characterized by its high accuracy of speed measurement, can be used to evaluate predicting methods.

##### 3.3.2.3.1 Near the Sea Surface

To evaluate the quality of the position predictions, we focused on the transition between the sea surface zone, the signal dead zone and the middle zone. It is in the signal dead zone where the position predictions are most needed since there is no tracking device that is able to operate in that zone.

In the case of the second experiment on the 17/03/2016, APS could detect the robot's position starting from 5m water depth. However, the USBL positions obtained were very scattered up to 10m. For that reason, we used the dead reckoning method with ADCP aided navigation till the robot reached 10m water depth.

At first stage, the average robot speed on the sea surface is calculated based on the drifting of the robot from starting position till the point the robot started diving (Table 8).

**Table 8** Calculation of the average robot's speed on the sea surface

| Time | Latitude | Longitude | X(m) | Y(m) | VX (m/s) | VY(m/s) |
|---|---|---|---|---|---|---|
| 10:53:56 | 36.865793 | 137.182232 | 4.636 | -10.209 | 0.205888 | -0.18688 |
| 10:56:28 | 36.865537 | 137.182583 | 35.931 | -38.615 | | |

At second stage, the robot's speed in the signal dead zone is calculated. It is needed to enhance the accuracy of the estimations with the Kalman filter. In this calculation, we assumed that the absolute water current velocity of the top 10m water layer is constant. Following is the algorithm:

1. Calculate the absolute water currents near the sea surface (look at section 5.4.2.1 for more details)
2. When the robot starts diving, for every ADCP depth cell
    a. Calculate the average of relative water at every depth cell
    b. Calculate the robot speed by subtracting the averaged relative water current from the absolute water current
3. Align the robot's speed profile of all bins and take their average
4. Finally, the robot speed near the sea-surface is obtained.

Since the ADCP sensor can only measure the water currents starting from 3m water depth, the values of the robot's speed at depths between 0 and 3m are estimated using a polynomial interpolation between the speed of the robot at the sea surface determined by the GPS sensor, and the value of the robot's speed at 3m calculated from ADCP data. The results of the calculation are shown in Fig. 36.

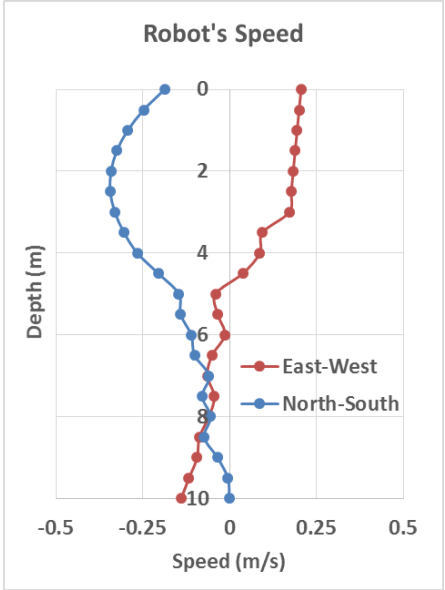
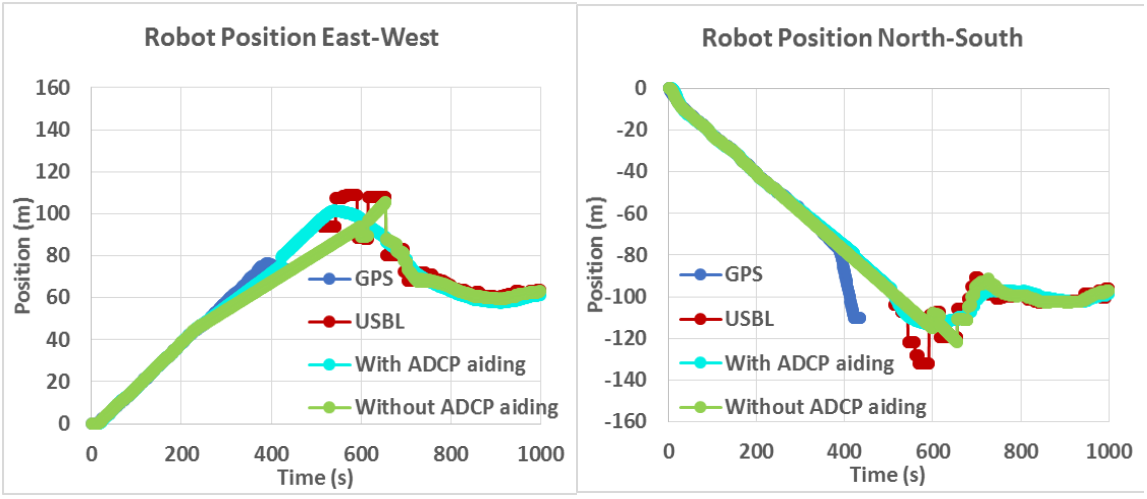



**Fig. 36** Robot's speeds near the sea surface



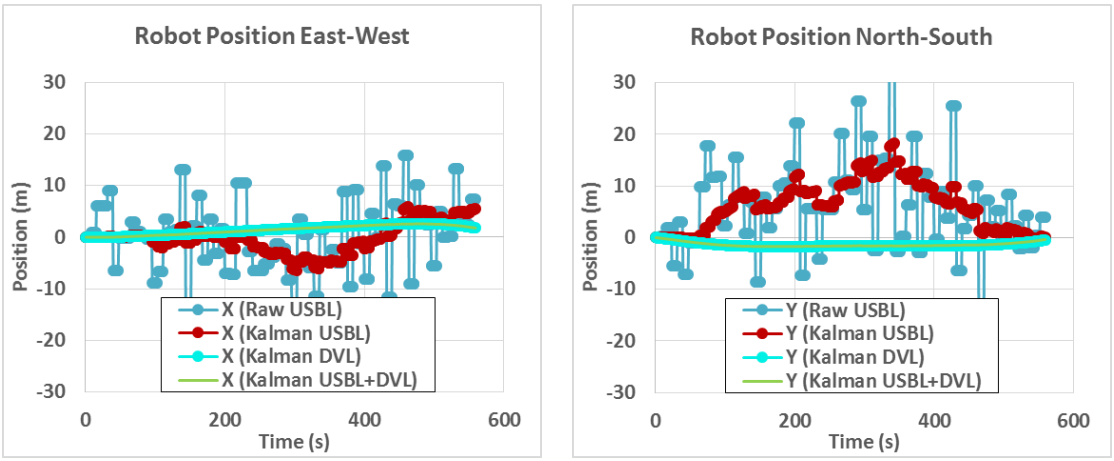

(a) East-West (b) North-South

**Fig. 37** Effect of ADCP aiding on position prediction

Fig. 37 shows the role of the ADCP aided navigation. The accuracy of the predictions was enhanced in the case were the ADCP aiding was used. The predictions of the dead reckoning method, starting from the last point in which the last reliable GPS position was obtained, enhanced with the ADCP aiding matches well the first USBL positions that were obtained 2 minutes later.

#### 3.3.2.3.2 Near the Seabed

When the robot approaches the seabed, the water current magnitude reduces significantly and the robot motion in the horizontal plane becomes slow. We consider studying the robot's the effect of the Kalman filter in the prediction of the robot's position.




(a) East-West (b) North-South

**Fig. 38** Effect of DVL aiding on position prediction

Fig. 38 compares 4 scenarios of navigation. In the first one, we use the raw data obtained from the USBL. It can be noticed that the USBL positions are scattered. The use of the Kalman filter helped to reduce the scattering, but if we compare the robot's motion direction determined by the speed value of the DVL sensor with the motion direction of the filtered USBL data, we notice that they are not matching. The DVL bottom tracking data are reliable in determining the direction and the speed where the robot moved. The use of dead reckoning algorithm associated with DVL aiding helped reflects well the robot motion direction and speed. In the last case, the association of DVL and USBL in the Kalman filter shows that the motion is almost the same as with the DVL only. This due to the inaccuracy of the USBL data translated by high values of $r_{P_X}$ and $r_{P_Y}$ in the covariance matrix R compared to the low values of $r_{V_X}$ and $r_{V_Y}$ of the DVL sensor.

## 3.4 Conclusions

In this chapter the robot's navigation was presented. In the first part, an overview of the sensors involved in the navigation was presented, followed by a study of their performances and shortcomings. Next, the navigation regions and the tracking method and the sensors involved in each region were defined. The use of the Kalman filter showed its effectiveness in filtering the position and absorbing spike noise, particularly of the USBL system. The filter also demonstrated its ability to fuse the sensors' data. The predictive performance of the filter was studied. The results showed that the use of the ADCP aiding contributed to the improvement of the position estimation accuracy.

# Chapter 4: SOTAB-I Guidance and Control



`

# 4 SOTAB-I Guidance and Control

## 4.1 Introduction

Due to their compactness, the use of AUVs for full-water surveying is being adopted increasingly (Jakuba et al., 2011) (Harvey et al., 2012). Among the existing types of underwater robots used to autonomously monitor marine environments in 3-D space from sea surface to seabed over the long term is the Argo Float (Roemmich et al., 2009) that floats vertically and repeats descending and ascending in the vertical direction by using a buoyancy control device. However, it does not have a function of active movement in the horizontal direction. Another method is the underwater glider (Eriksen et al., 2001), which has a streamlined body with fixed wings. It can descend and ascend also by using a buoyancy control device, while it moves in the horizontal plane like a glider for long distance. However, the ratio of vertical movement distance to horizontal movement distance is small. SOTAB-I provides functionality that lies midway between profiling buoys and gliders. It was designed not only to move in the vertical direction by a buoyancy control device, but also in the horizontal direction by two pairs of rotational fins.

In this chapter, we define the outlines of the guidance and control of the robot, notably its operating modes and zones as well as control program priorities. Next, we consider developing depth and altitude control algorithms as well as studying the effect of the wings' control. During operation, software or hardware deficiency may occur, leading the robot dangerously approaches the seabed. We present the developed collision avoidance algorithm preventing the robot from colliding with the seabed. The development of control programs should be adapted not only for fast intervention, but also to ensure a longer

operating time. The last part of this chapter deals with the power consumption of the robot and an energy study was done to estimate the robot's battery autonomy under different conditions.

## 4.2 Outlines of the Guidance and Control

### 4.2.1 Operating Modes

#### 4.2.1.1 Manual Mode

In this mode, the control of the robot will be performed manually through the GUI. Commands can range from simple orders to following a whole control scenario. This mode primarily enables us to test the robot's basic functions and to verify that all sensors and actuators as well as tracking and data transmission devices are working correctly.

#### 4.2.1.2 Survey Mode

SOTAB-I has three main surveying modes. At the first stage, SOTAB-I performs the water column survey by adjusting its buoyancy. The rough mode is used to collect rough data on physical and chemical characteristics of plumes by repeating descending and ascending on an imaginary circular cylinder centered at the blowout position of oil and gas through the variation of buoyancy and movable wings' angles. Finally, in case the UMS detects a high concentration of any particular substance, a precise guidance mode will be conducted to track and survey its detailed characteristics by repeating descending and ascending within the plume.

#### 4.2.1.3 Photograph Mode

This mode enables us to have a large visual overview of the area around the blowout position of oil and gas by taking pictures of the seabed and making image mosaicking. SOTAB-I moves laterally using horizontal thrusters along diagonal lines of a polygon

with a radius of 5 m centered on the blowout position of oil and gas.

### 4.2.2 Operating Zones

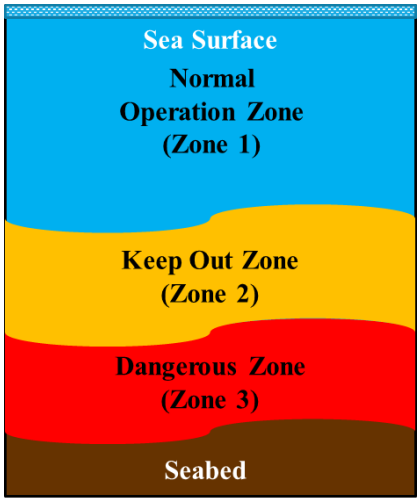



**Fig. 39** Water column regions

Based on the robot altitude from the seabed determined by the DVL, the water column is divided into three zones: The normal operation zone, the keep out zone and the dangerous zone (Fig. 39). The altitude of each zone is configurable by the user at the beginning of the experiment. In the case where the photograph mode is executed, the user should define the altitude of the dangerous and the keep out zones lower than the altitude from which the photographs will be taken. Fig. 40 illustrates the flow chart used for determining the zone in which the robot is situated.

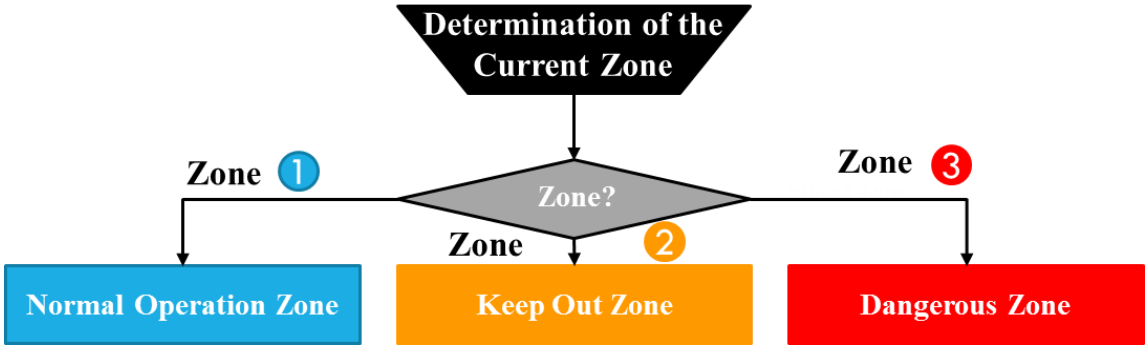



**Fig. 40** Determination of water column region

#### 4.2.2.1 Normal Operation Zone

In this zone, the robot performs surveying in operational modes as well as the photograph mode. When the user on the mother ship decides to launch a specific operational mode, the GUI reads its associated ID and includes it in the downlink data that will be sent through the acoustic modem. The robot receives the data, identifies the requested mode, and executes it. At the end of the execution, the robot waits for the next downlink order.

#### 4.2.2.2 Keep Out Zone

This is the zone from which operation of the robot is judged to be somewhat dangerous because the robot is getting closer to the seabed. A collision avoidance maneuver based on PID control of vertical thrusters is used to smoothly stabilize the robot above a predefined critical altitude (check section 4.6 for more details).

#### 4.2.2.3 Dangerous Zone

Operation of the robot in this zone is very dangerous and presents a risk that the robot can hit the seabed, which may lead to heavy damage to the robot and especially to the ceramic transducer of the DVL. The altitude of the dangerous zone is adaptive and is determined based on the vertical speed of the robot and its altitude. The control program calculates the time needed to reach the seabed based on the vertical speed value. In the case in which the time to reach the seabed becomes less than a specified time limit, the thrusters are activated. This enables the robot to have enough time to decelerate the robot. The thrusters are also automatically activated in the case where the robot altitude goes below a specified critical value.

### 4.2.3 Control Priorities

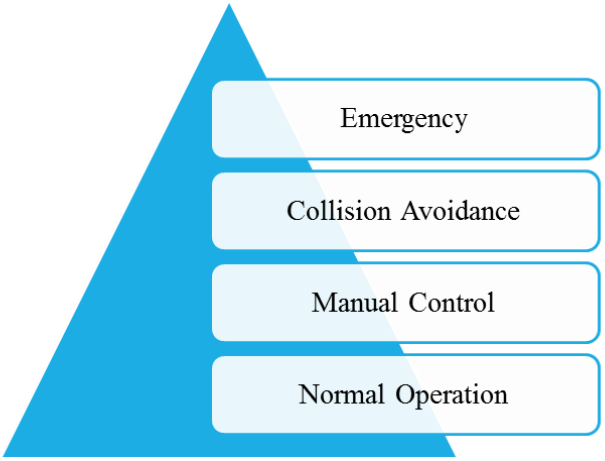



**Fig. 41** Control priorities

The execution priority of the control programs has a pyramidal hierarchy as shown in Fig. 41.

#### 4.2.3.1 Emergency

The emergency order has the highest priority. A state of emergency is flagged when the robot battery runs below a pre-defined threshold. At the beginning of each mission, the mission timer is reset. When the timer reaches the maximum mission time, the emergency is activated. The emergency state can also be sent through downlink data when an abnormality is detected in the control program or in the uplink data received. Overheating, humidity, and high currents can also be added, but they are not currently implemented in the software. When the emergency is activated, the program interrupts the control algorithm and sets the neutral buoyancy to its maximum value. The data logging resumes normally and sensors are kept powered on, except in the case when the power of the system becomes too low. It is important to mention here that this emergency is a software emergency, and it is different from the emergency encountered by dropping the ballast weight.

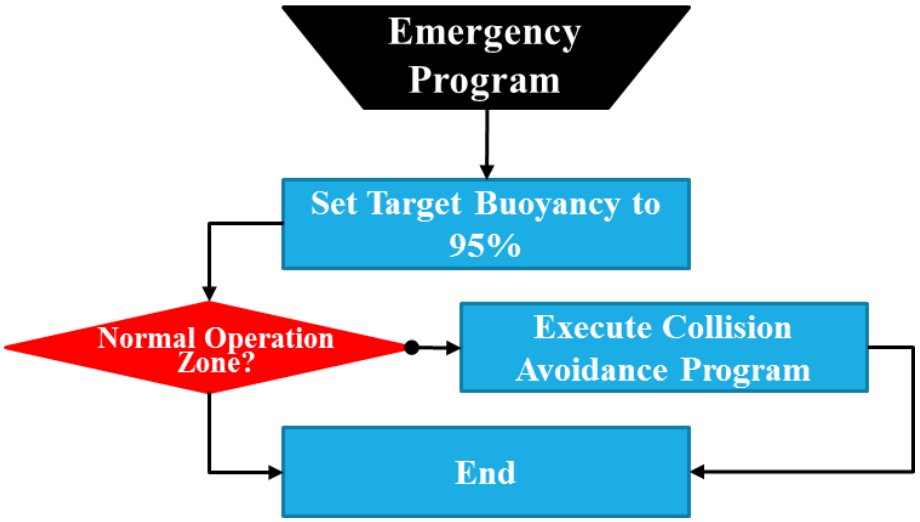



**Fig. 42** Emergency flowchart

#### 4.2.3.2 Collision Avoidance

The second-place priority is the collision avoidance program. The collision avoidance uses only the thrusters. Thus, if the control program does not involve the control of the thrusters, it will keep executing. Otherwise, the thruster command order of the control program will be ignored, and only the collision avoidance command will be considered. Fig. 43 depicts the flowchart of the collision avoidance program.

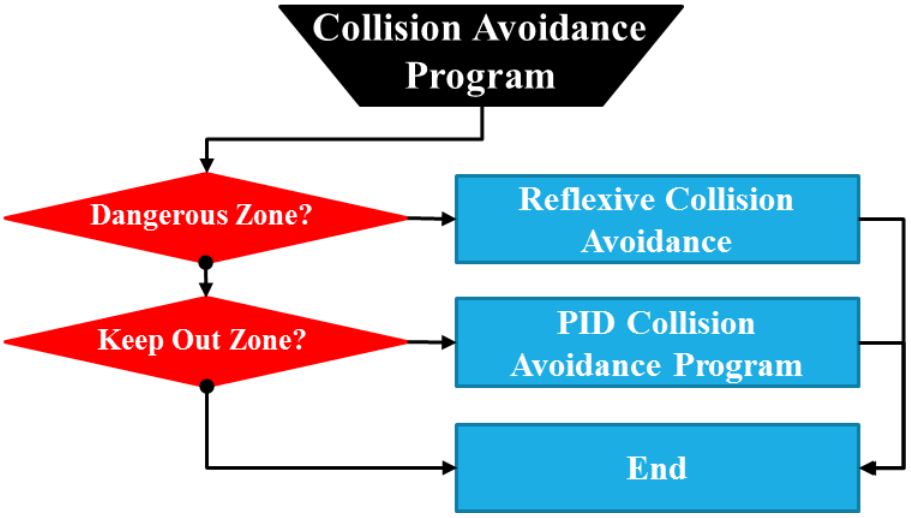



**Fig. 43** Collision avoidance flowchart

#### 4.2.3.3 Manual Control

Manual control comes next. On the sea-surface, only the manual control through Wi-Fi is counted, while the acoustic control is ignored. When the robot is underwater, the control through radio waves becomes impossible, and the acoustic communication is used instead. The robot is able to know its position regarding water level through the GPS status (GPS.STS) flag. When the robot is connected to Wi-Fi, the manual control can be performed either through GUI directly or through a joystick connected to the UDP port. The control of the robot through downlink data is explained in section 2.3.3. Fig. 44 illustrates the flowchart of the manual control program.

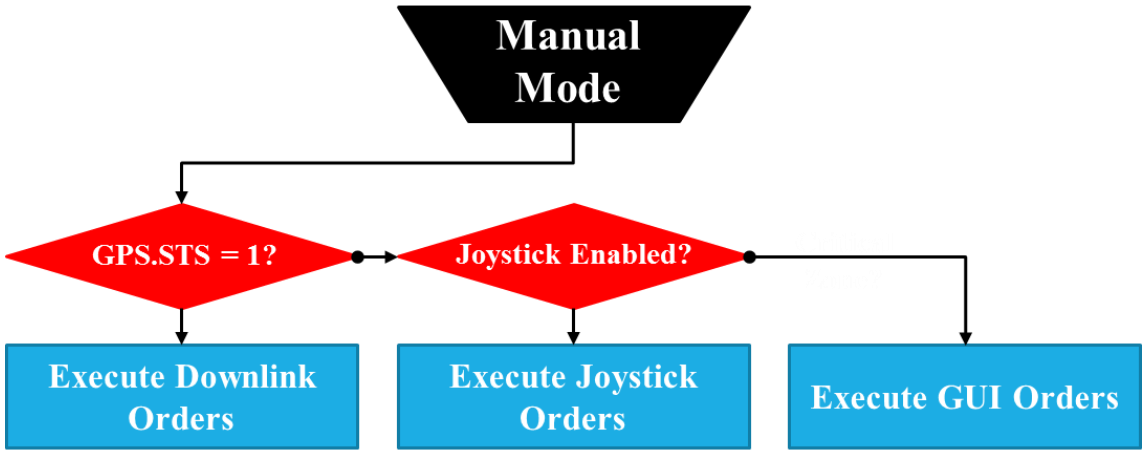



**Fig. 44** Manual control flowchart

#### 4.2.3.4 Normal Operation

Finally, if none of the higher-priority algorithms are flagged, the robot performs its designated operating mode described in section 4.2.1. The flow graph in Fig. 45 summarizes the normal operation of SOTAB-I.

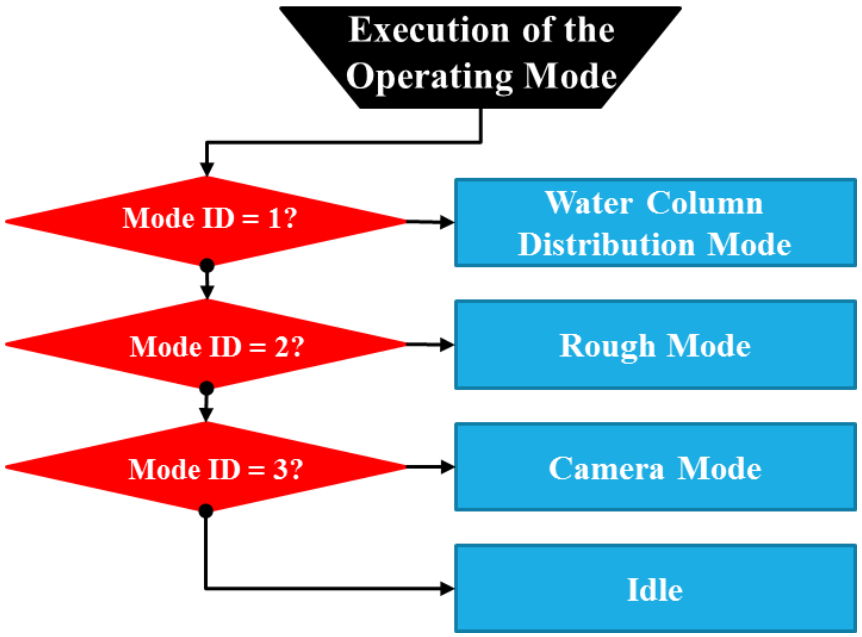



**Fig. 45** Normal operation zone flowchart

### 4.2.4 Control Program

Fig. 46 and Fig. 47 shows respectively the initialization steps performed at the time the GUI is launched. Fig. 47 depicts the main program loop.

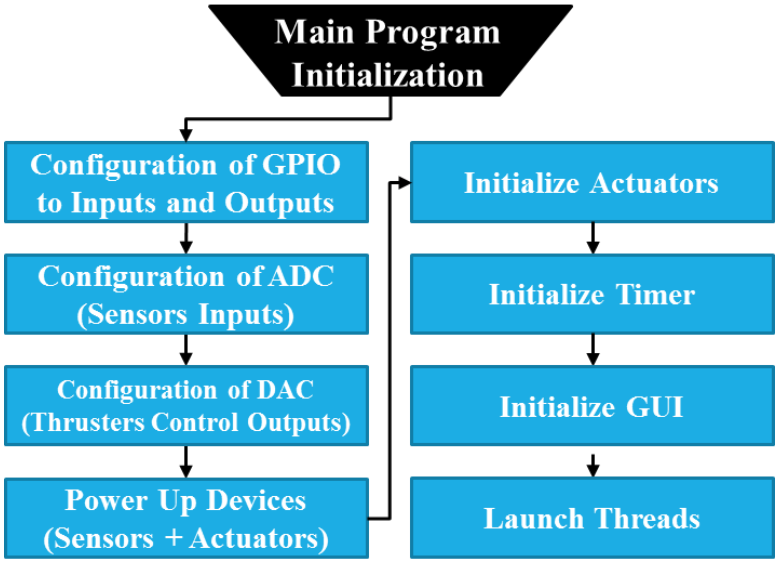



**Fig. 46** Main program initialization

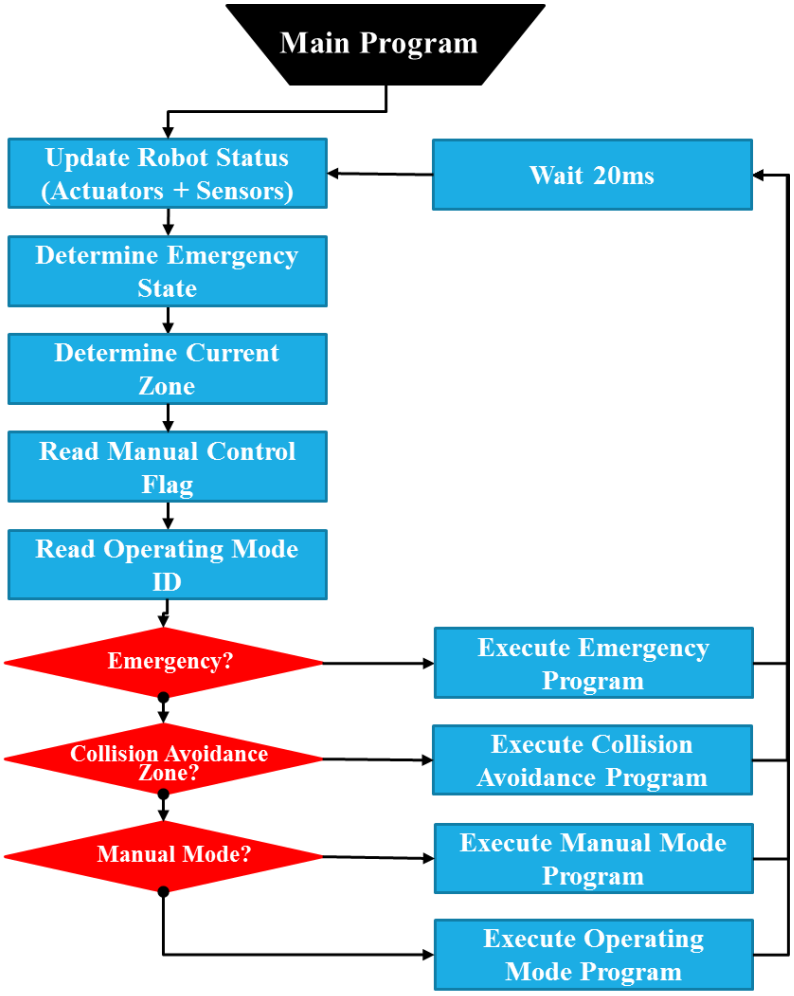



**Fig. 47** Main program flowchart

### 4.2.5 Equations of Motion

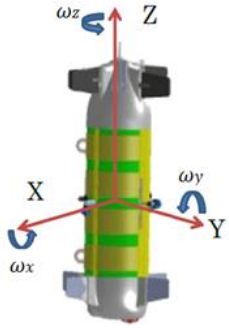



**Fig. 19** Body fixed coordinate system

Using the coordinate system illustrated in Fig.3.19, where the origin of the coordinates is set at the center of gravity, the following equation of motion in the body fixed coordinates is expressed, where the symbols $A_{11}, A_{21}$, ... and $I_{XX}, I_{XY}$, ... are defined as the added mass and the moment of inertia (Azuma and Nasu, 1977). Added mass is computed by the Hess-Smith method (Hess and Smith, 1964) and the moment of inertia is calculated from the equipment layout. Other parameters are defined in Table 9.

$$\begin{pmatrix} \mathrm{M}+\mathrm{A}_{11} & 0 & 0 & 0 & \mathrm{A}_{51} & 0 \\ 0 & \mathrm{M}+\mathrm{A}_{22} & 0 & \mathrm{A}_{42} & 0 & 0 \\ 0 & 0 & \mathrm{M}+\mathrm{A}_{33} & 0 & 0 & 0 \\ 0 & \mathrm{A}_{24} & 0 & I_{XX}+A_{44} & -\mathrm{I}_{\mathrm{YX}} & -\mathrm{I}_{\mathrm{ZX}} \\ \mathrm{A}_{15} & 0 & 0 & -\mathrm{I}_{\mathrm{XY}} & \mathrm{I}_{\mathrm{YY}}+\mathrm{A}_{55} & -\mathrm{I}_{\mathrm{ZY}} \\ 0 & 0 & 0 & -I_{XZ} & -\mathrm{I}_{\mathrm{YZ}} & \mathrm{I}_{\mathrm{ZZ}}+\mathrm{A}_{66} \end{pmatrix} \begin{pmatrix} \dot{U}_X \\ \dot{U}_Y \\ \dot{U}_Z \\ \dot{\omega}_X \\ \dot{\omega}_Y \\ \dot{\omega}_Z \end{pmatrix}$$

$$= \begin{pmatrix} \begin{Bmatrix} -F_1 \sin\Theta + F_{HX} + (M+A_{22})U_Y\omega_Z - A_{22}W_Y\omega_Z + A_{33}W_Z\omega_Y \\ +A_{24}\omega_X\omega_Z - (M+A_{33})\omega_Y\omega_Z \end{Bmatrix} \\ \begin{Bmatrix} F_1 \cos\Theta \sin\Phi + F_{HY} - (M+A_{11})U_X\omega_Z + (M+A_{33})U_Z\omega_X \\ +A_{11}W_X\omega_Z - A_{33}W_Z\omega_X - A_{15}\omega_Y\omega_Z \end{Bmatrix} \\ \begin{Bmatrix} F_1 \cos\Theta \cos\Phi + F_{HZ} + (M+A_{11})U_X\omega_Y - (M+A_{22})U_Y\omega_X \\ -A_{11}W_X\omega_Y + A_{22}W_Y\omega_X - A_{24}\omega_X^2 + A_{15}\omega_Y^2 \end{Bmatrix} \\ \begin{Bmatrix} F_2(y_B \cos\Theta \cos\Phi - z_B \cos\Theta \sin\Phi) + M_{HX} + (A_{22}-A_{33})U_Y U_Z \\ +A_{33}U_Y W_Z - A_{22}U_Z W_Y + A_{15}U_X\omega_Z + A_{24}U_Z\omega_X \\ -A_{15}W_X\omega_Z + (I_{YY}-I_{ZZ}+A_{55}-A_{66})\omega_Y\omega_Z \end{Bmatrix} \\ \begin{Bmatrix} F_2(-z_B \sin\Theta - x_B \cos\Theta \cos\Phi) + M_{HY} - (A_{11}-A_{33})U_X U_Z \\ -A_{33}U_X W_Z + A_{11}U_Z W_X - A_{24}U_Y\omega_Z - A_{15}U_Z\omega_Y \\ +A_{24}W_Y\omega_Z - (I_{XX}-I_{ZZ}+A_{44}-A_{66})\omega_X\omega_Z \end{Bmatrix} \\ \begin{Bmatrix} F_2(x_B \cos\Theta \sin\Phi + y_B \sin\Theta) + M_{HZ} + (A_{11}-A_{22})U_X U_Y \\ -(A_{15}+A_{24})(U_X\omega_X - U_Y\omega_Y) + A_{22}U_X W_Y - A_{11}U_Y W_X \\ +A_{15}W_X\omega_X - A_{24}W_Y\omega_Y + (I_{XX}-I_{YY}+A_{44}-A_{55})\omega_X\omega_Y \end{Bmatrix} \end{pmatrix} \tag{13}$$

$F_1$ and $F_2$ are given by the following equations

$$\boldsymbol{F_1} = \boldsymbol{\rho g V_B} - \boldsymbol{M g} + \boldsymbol{F_B} \tag{14}$$

$$\boldsymbol{F_2 = \rho g V_B + F_B} \tag{15}$$

**Table 9** Definition of parameters of the equation of motion

| Symbols | Definition |
|---|---|
| (X, Y, Z) | Body fixed coordinates |
| $(U_X, U_Y, U_Z)$ | Robot velocity |
| $(W_X, W_Y, W_Z)$ | Water current velocity |
| $(\omega_X, \omega_Y, \omega_Z)$ | Angular velocities |
| $A_{i,j}(i, j = 1\sim6)$ | Added mass |
| M | Robot mass |
| $(x_B, y_B, z_B)$ | Buoyancy center |
| $F_B$ | Buoyancy |
| $(F_{HX}, F_{HY}, F_{HZ})$ | Hydrodynamic forces |
| $(M_{HX}, M_{HY}, M_{HZ})$ | Hydrodynamic moments |
| ρ | Water density |
| $g$ | Gravity acceleration |
| $V_B$ | SOTAB-I's volume |
| $I_{i,j}$ $(i, j = X, Y, Z)$ | Moment of inertia |
| ϴ | Pitch angle |
| ϕ | Roll angle |
| ψ | Azimuth angle |
| $(\dot{\ })$ | Derivative with respect to time |

The motion of the robot can be simulated by solving equation (13) using the Newmark-$\beta$ method. Using this simulation, programs of the guidance and control were constructed and the required operating times are estimated.

For the rough guidance mode, a set of target points along the circles at the top and the bottom of a circular cylinder are given to perform line tracking between a point at the top of the circular cylinder and a point at the bottom of the circular cylinder. Let us define $(X_{E,i} \quad Y_{E,i} \quad Z_{E,i})$and $(X_{E,SOTAB} \quad Y_{E,SOTAB} \quad Z_{E,SOTAB})$ as the target point and the position of SOTAB-I, respectively, in the Earth fixed coordinate. If we define ($X_p$, $Y_p$, $Z_p$) as the offset of the present position from the target point in the body fixed coordinate, ($X_p$, $Y_p$, $Z_p$) can be obtained as follows

$$\begin{pmatrix} X_p \\ Y_p \\ Z_p \end{pmatrix} = T_B^{-1} \begin{pmatrix} X_{E,i} - X_{E,SOTAB} \\ Y_{E,i} - Y_{E,SOTAB} \\ Z_{E,i} - Z_{E,SOTAB} \end{pmatrix} \tag{16}$$

$$T_B = \begin{pmatrix} \cos\Theta\cos\Psi & \cos\Theta\sin\Psi & -\sin\Theta \\ \cos\Psi\sin\Phi\sin\Theta - \cos\Phi\sin\Psi & \cos\Phi\sin\Theta\sin\Psi + \cos\Phi\sin\Psi & \cos\Theta\sin\Phi \\ \cos\Phi\cos\Psi\sin\Theta + \sin\Phi\sin\Psi & \cos\Phi\sin\Theta\sin\Psi - \cos\Psi\sin\Phi & \cos\Phi\cos\Theta \end{pmatrix} \tag{17}$$

The equations of motions were incorporated in a program used to simulate the behavior of the robot. The simulator will help to evaluate the control programs before their implementation on the robot computer.

## 4.3 Depth Control

### 4.3.1 Introduction

There are several methods used to control the depth of AUVs. The majority relies on adjusting the buoyancy control device to control their depth. There exist a variety of mechanisms to adjust the buoyancy. In the submarines for example, the amount of the air/seawater of trim or ballast tanks is controlled to adjust the buoyancy. When the submarine is on the surface, air is filled in the ballast and the submarine's becomes positively buoyant. To start diving, water is introduces into the ballast tanks while the air is vented outside until it becomes negatively buoyant leading the submarine to sink. Compressed air is stored in flasks to adjust the amount of water inside the ballast tank during operation. Another widely employed mechanism in AUVs is to adjust the volume of the robot through a device that compresses and expands the air contained in a cylinder. This mechanism is characterized by its reliability and its relative fast response. However, the motor pump, used to ensure the compression and the expansion of the air, generates noise. Additionally, when the robot is decreasing its buoyancy, an amount of the ballast

water (seawater) is pulled from one region and then, may be dumped in another region to increase buoyancy. This represent a risk of dumping living organisms in an environment different from their original inhabiting region, which may harm their new environment. This problem is referred as the ballast water problem (Hallegraeff, 1998) (Zhang & Dickman, 1999). Among other existing technologies, there is the metal bellow mechanism, which imitates the change of state of the spermaceti oil from liquid to solid and vice versa, leading to change of density, in the sperm whale (Clarke, 1978). Similarly, AUV using metal bellow mechanism relies on the change of state of a low melting point liquid, such as wax (Mcfarland et al., 2003) or oil (Shibuya et al., 2013), by adjusting its temperature. This mechanism doesn't make noise and presents an ecological advantage over other systems since it doesn't involve any discharging of ballast materials, eliminating the ballast water problem. However, results show that their response time is slow and is energetically costly since the temperature of the oil should be maintained. A third mechanism imitates ray-finned fish, which adjust the volume of their bladders to adjust their buoyancy (Bond, 1996). They employ polymer buoyancy control device (Um et al, 2011). Electrolysis is used to generate pure hydrogen, which is a clean gas, in order to expand the volume of an artificial bladder leading to a displacement of water and an increase of buoyancy. To reduce the robot buoyancy, extra amount of gases are simply released outside via a valve. These system are characterized by their silent operation. However, they are more oriented for small devices operating near the sea surface where the water pressure is not significant. Due to the reliability and the fast response as well as their low power consumption, the buoyancy variation through the adjustment of the air volume in a cylinder mechanism was employed. The ballast water problem doesn't apply

for SOTAB-I since it is designed to operate around the same region of the blow out gas. The noise caused by the motor pump may be reduced by choosing a high quality actuator. For the same control mechanism, there exist several control strategies. An implementation of a cascaded velocity-position PID controller was used in a coastal profiling float by (Barker, 2014). The method consists of adjusting the velocity set point according to depth error between the current and target depths. The vertical velocity is controlled through a PID controller to achieve the desired depth. The algorithm succeeded to achieve the desired depth near the sea surface, but at a high energy cost. Another control strategy is employed in the underwater gliders where the buoyancy control device is performed simultaneously with a mechanism of gravity center movement in horizontal plane. On the other hand, Argo float uses only buoyancy device to adjust their depth. To do so, the float relies on the establishment of a highly accurate ballasting curve (Izawa et al, 2002). This requires a high precision ballasting experiment to adjust the robot's density in a way to become equal to the density of the seawater, which will be measured by a highly accurate CTD sensor, at the designated parking depth. This will lead the robot to reach its neutral buoyancy point.

There are many challenges and constraints associated with depth control of underwater robot. For instance, at-sea experiments require enormous financial and logistic resources, limiting the experiments time. Hence, it is important that the program should be easy to implement and repeatedly verified by simulating programs before its real deployment. On the other hand, environmental constraints like a considerable variation of the density of water between the sea surface and the seabed bring complications in the control because they lead to the variation of the neutral buoyancy value of the robot. Even if the neutral

buoyancy of the robot is determined accurately at a certain spatial condition, there is no guarantee that the robot will keep its vertical position due to the up-welling and down-welling water currents. Other constraints are represented by the hardware limitations. In fact, the buoyancy device employed has three controlling states: it can be controlled to increase the buoyancy, decrease it or stay idle. However, it is not possible to change the rate of variation directly. In addition, the rate of change of buoyancy is relatively slow, not symmetric in both directions and vary with depth. Moreover, the change of the buoyancy variation orientation is not instantaneous, there is a lag time of 2s between each change of state. The oil level sensor has also an inaccuracy within (+/-) 0.05% of the total oil volume. Previously, a PID controller was developed for depth control (Kato et al., 2015). It gave good performances and small overshoot, but only for a depth range up to 100m. Beyond that limit, significant overshoot was reported. The previous controller relied on a very accurate determination of the neutral buoyancy. In addition, the PID control parameters were not adaptive. Besides, it doesn't enable to freeze the robot at the target depth. For the lacks mentioned before, it is necessary to develop a new controller that overcomes the mentioned shortenings and take in consideration the environmental and hardware constraints.

A new method for depth control was developed. It is aimed to work for any target depth. The method relies mainly on the buoyancy variation model with depth, established based on tank and at-sea experiments data.

### 4.3.2 Establishment of the Buoyancy Model

The objective in this section is to establish a time model and a buoyancy model. The time

model is needed for the depth control of the robot. It enables to estimate the time needed for SOTAB-I to change its buoyancy from its current value to a target value. The buoyancy model is also needed for the simulating program. It enables to estimate the variation of the buoyancy value from its initial value every sampling period. We consider establishing a model for the buoyancy variation from 20 to 85% up to 1000m water depth based on the experiments results obtained at pressure tank and at-sea.

#### 4.3.2.1 Experiments Results of Buoyancy Variation

##### 4.3.2.1.1 Pressure Tank Experiments

Pressure tank experiments were performed to calculate the time necessary for changing 7500cc of oil in the reservoir in both directions. OUT->IN direction is when the oil hydraulic pump injects and extracts oil from the external oil bladder and injects it to the internal oil reservoir. IN->OUT is the opposite direction.

**Table 10** Time of variation of robot buoyancy with pressure

| | External pressure | Motor rotation speed | Flow rate | Time |
|---|---|---|---|---|
| | MPa | rpm | cc/min | min/7,500cc |
| IN>OUT | 0 | 7865 | 331 | 22.6 |
| | 10 | 6700 | 282 | 26.6 |
| | 20 | 5760 | 243 | 30.9 |
| OUT>IN | 0 | 7865 | 331 | 22.6 |
| | 10 | 7865 | 331 | 22.6 |
| | 20 | 8300 | 349 | 21.5 |

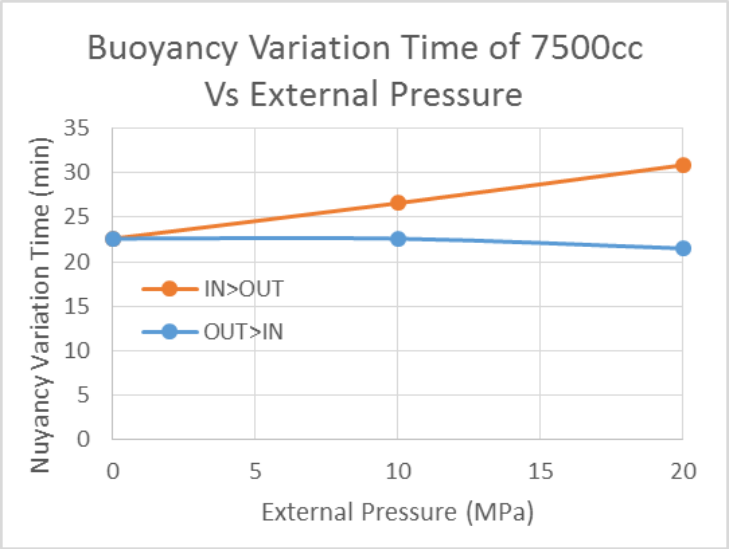
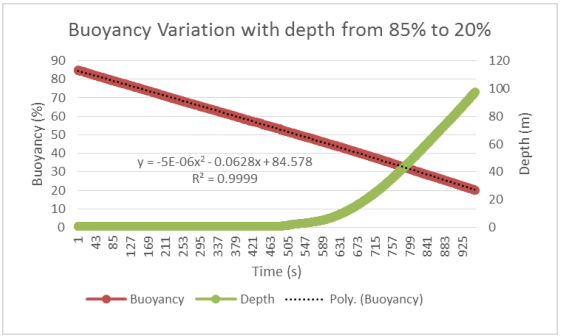


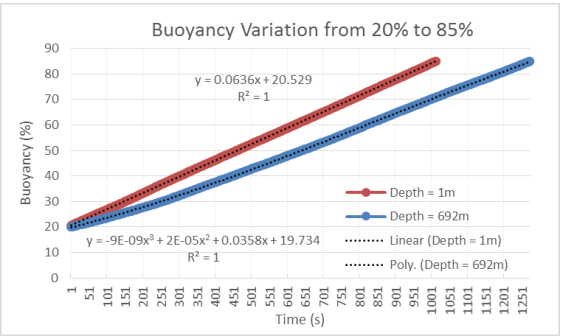

**Fig. 48** Relationship between the pressure and the buoyancy variation time

Based on Fig. 48 and Table 10, the time needed to change the buoyancy corresponding to 7500cc in the OUT>IN direction is constant till 10MPa (~1000m water depth). From 10MPa to 20Mpa, it can be modeled as a linear function. The time difference between the full scale variation of buoyancy at 0MPa and 20MPa is less than 9 minutes. For the IN>OUT direction, The Buoyancy variation time is almost same from 0 up to 10MPa. Beyond that limit, it becomes slightly faster.

#### 4.3.2.1.2 At-sea Experiments




(a) OUT>IN




(b) IN>OUT

**Fig. 49** Buoyancy variation with depth on 20th March 2015 experiment

Fig. 49 (a) confirms the results obtained in the pressure tank. The buoyancy variation rate is almost same from 0 to 100m. Hence, the time and buoyancy models in the OUT>IN direction can be modeled as a linear function. Fig. 49 (b) dates from an experiment on the 20th of March 2015 in Toyama Bay. It shows the buoyancy variation in the IN>OUT direction from 20 to 85% at 1m and 700m water depths. It shows that buoyancy variation can be represented under the form of a 3rd degree polynomial function as shown in Table 11.

**Table 11** Buoyancy variation model of 20th March 2015 experiment

| **Direction** | **Depth** | **Buoyancy variation** |
|---|---|---|
| OUT>IN | 1..700m | $Tc = -15.122 * (B_t – B)$ |
| IN>OUT | 1m | $Tc = 15.122 * (B_t – B)$ |
| | 700m | $Tc = 0.0015 * (B_t^3 - B^3) - 0.2688 * (B_t^2 – B^2) + 34.065 * (B_t – B)$ |

#### 4.3.2.2 Model of the Buoyancy Variation with Depth

Based on the experimental data, the buoyancy variation in the OUT>IN direction can be modeled as a linear function.

$$Tc\ (OUT\text{->}IN) = -15.122 * (B_t – B) \quad (18)$$

In the IN>OUT direction, the main parameter that contributes considerably in the change of the buoyancy variation speed is the depth. The model can be generalized and written under the form of 3rd degree polynomial function that depends on the depth D.

$$\mathbf{Tc\ (IN \text{->} OUT) = C_3\ (D) * (B_t^3 - B^3) – C_2\ (D) * (B_t^2 – B^2) + C_1\ (D) * (B_t – B)} \quad (19)$$

Linear interpolation and extrapolation of the coefficients a, b and c are used to determine the buoyancy model at a certain depth based on the models established for depths equal to 1m and 700m. The following equation shows the formula used to calculate Ci (D).



`

$$C_i(D) = (C_i(1) - C_i(700)) * D / (700 - 1),\ i = \{1, 2, 3\} \quad (20)$$

Knowing that

- At 700m $C_3 = 0.0015$; $C_2 = 0.2688$; $C_1 = 34.065$
- At 1m $C_3 = 0$; $C_2 = 0$; $C_1 = 15.122$

#### 4.3.2.3 Comparison between Buoyancy Variation Model and Experimental Data

Data of buoyancy variation time in several ranges were collected from previous at-sea experiments of SOTAB-I and compared to the values obtained by the model. To estimate the accuracy of the model, we defined the ratio Time/Range to estimate the time deviation of the model in seconds for every 1% of buoyancy variation. Table 12 summarizes the obtained results.

**Table 12** Comparison between the buoyancy variation model and experimental results

| Experiment | Depth | Orientation | Range of variation | Tc (Experiment) | Tc (Model) | Time/Range (s/1%) |
|---|---|---|---|---|---|---|
| 5/25/2015 2:28:43 | Air | OUT>IN | 95%→21 % | 1121 s | 1119 s | -0.03 |
| | | IN>OUT | 20%→94 % | 1166 s | 1164s | -0.03 |
| 11/27/2014 9:20:56 | 0~95m | OUT>IN | 79%→20 % | 859 s | 892s (50m) | +0.56 |
| | 700m | IN>OUT | 20%→85 % | 1274 s | 1340s (700m) | +1.01 |
| 11/28/2014 9:48:23 | 0~42m | OUT>IN | 78%→31 % | 703 s | 710s | +0.15 |
| | 0~100m | IN>OUT | 31%→79 % | 854 s | 823 s (50m) | -0.65 |
| 3/20/2015 14:12:33 | 95~155m | IN>OUT | 20%→30 % | 178 s | 199 s (155m)<br>184 s (95m) | +2.1<br>+0.6 |
| | 155~198 m | IN>OUT | 30%→40 % | 176 s | 192 s (155m)<br>193 s (198m) | +1.6<br>+1.5 |
| | 198~210 m | IN>OUT | 40%→49 % | 170 s | 169 s (198m) | -0.11 |

Table 12 shows the estimated time of buoyancy variation is close to the values obtained from experiments. The maximum deviation was obtained when the buoyancy variation is between 20% and 30%. The time estimation in the OUT>IN direction doesn't exceed 0.5s per 1% of buoyancy variation at full range, suggesting a high accuracy of the model in

that direction. On the other hand, in the IN->OUT direction, the maximum values of deviations were obtained in the 20 to 30% range, but didn't exceed 2.1s per 1% variation of buoyancy. So we can consider the overall model as reliable. It is important here to mention that the model accuracy can be improved by feeding the model with additional experiment data at various depths which will reduce the model interpolation and extrapolation error.

#### 4.3.2.4 Buoyancy Simulation Program

It is important also to accurately assimilate the behavior of the buoyancy device to get closer results to the real at-sea experiments case. We suggest establishing a simulator of the buoyancy device that will be integrated in the robot simulator program. The simulator input/output diagram is shown in Fig. 50.

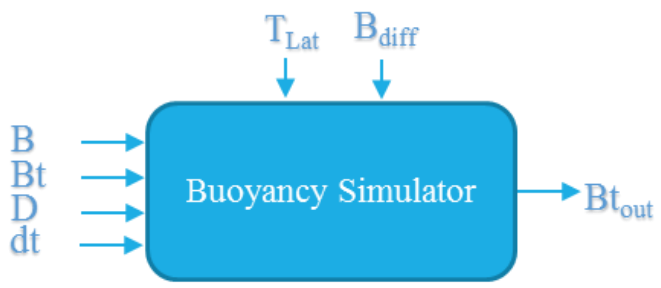



**Fig. 50** Buoyancy Variation Simulator

The simulator main function is to estimate the buoyancy variation (%) from the current buoyancy value (B) within a time lapse (dt). The time latency of the buoyancy variation from one direction to another (Tlat) is considered. In addition, the effect of depth (D) on the speed of buoyancy variation is taken into consideration in both directions based on the model established in section 4.3.2. The buoyancy target (Bt) serves to know in which direction the buoyancy is varying either increasing or decreasing. The tolerance margin

around the final target buoyancy is defined by Bdiff. Fig. 51 illustrates the flowchart of the buoyancy simulator.

At the beginning, the program determines the buoyancy variation orientation by comparing the current buoyancy (B) to the target buoyancy (Bt). In the case where the current orientation is different from the previous one, then this mean that the buoyancy variation should be stopped the time (Tlat) needed to change the orientation, exactly as what happens in the real case. The model of the buoyancy variation per time will be then selected. F corresponds to the model of buoyancy variation per time in the OUT->IN direction and G in the IN->OUT direction.

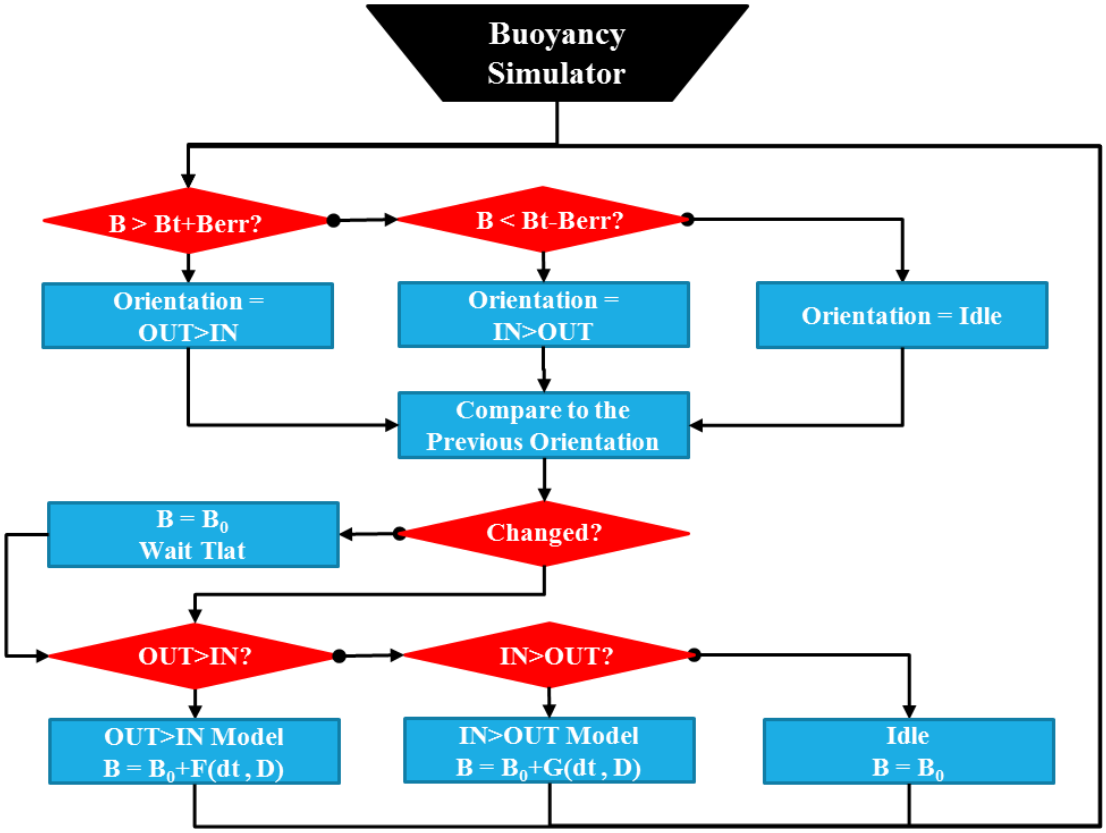



**Fig. 51** Buoyancy simulator flowchart

### 4.3.3 Control Algorithm

Depending on the operating mode, there are two main scenarios to be considered. The first is that the robot reaches the target depth and subsequently starts ascending, like in the rough mode. The second is to bring the robot to the target depth and freeze it there until an ascending order is received through acoustic communication or ascending timer overflow. Therefore, we can decompose the control program into two main steps. As shown in Fig. 52, the first step is to bring the robot from its current depth (D) to a set target depth (Dt) using a predictive depth controller. Once the target depth is reached, the second step is executed to stabilize it around the target depth.

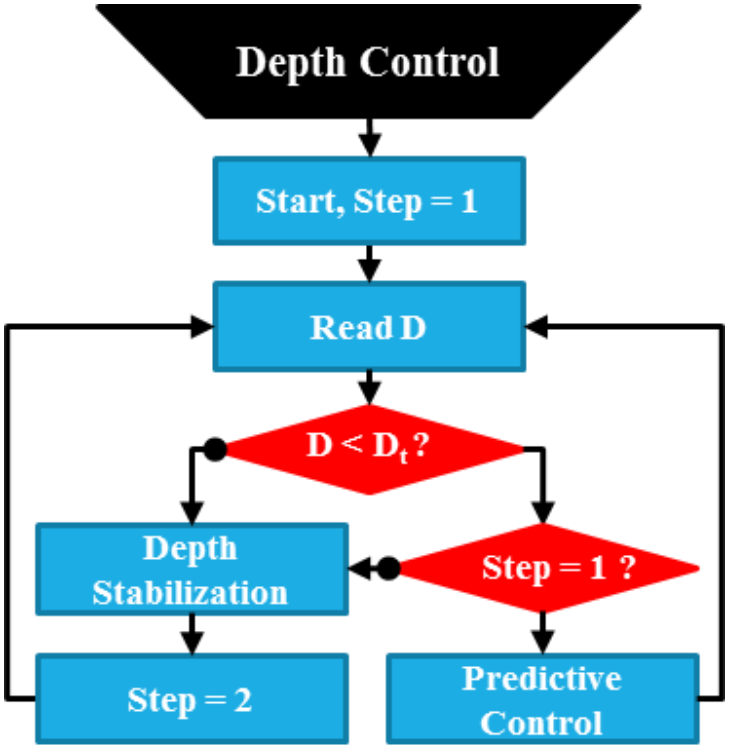



**Fig. 52** Depth Control Process

In the next part we explain in details the predictive controller and the depth stabilization algorithm.

#### 4.3.3.1 Predictive Control

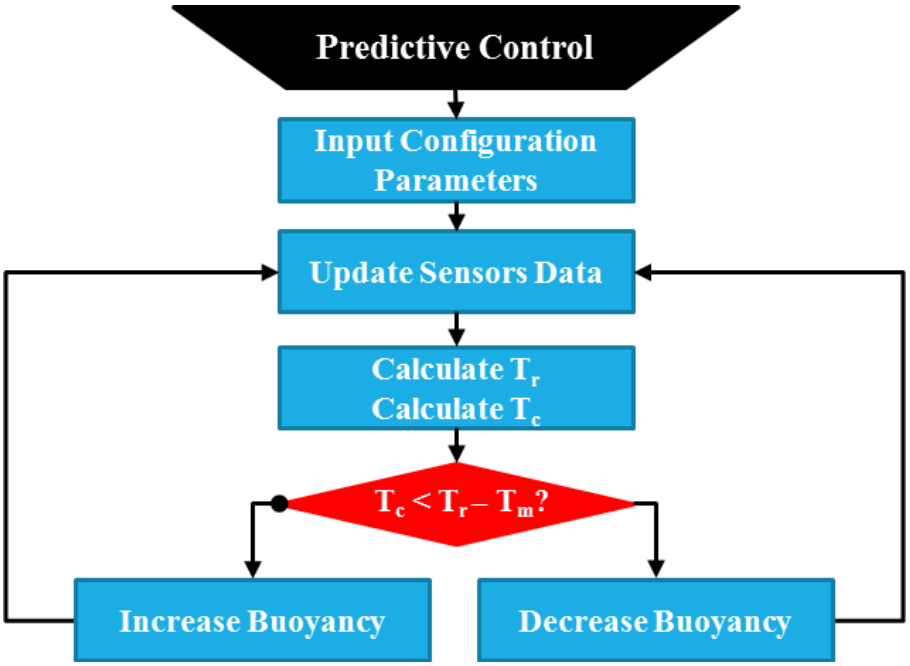



**Fig. 53** Predictive depth control flowchart

As shown in Fig. 53, we introduce the configuration parameters of the program at the beginning of the experiment. For instance, the estimated value of the neutral buoyancy ($B_n$) with the margin of uncertainty around it ($B_m$). We also input the value of sensors random error of the oil sensor ($B_{err}$) and of the depth data ($D_{err}$) of the CTD sensor, based on the sensors' data collected in the previous experiments.

Once the program is executed and its configuration is over, the predictive controller is executed every second. At first the robot updates all the sensors' data, such as the depth (D), provided by the CTD sensor, and the value of the current buoyancy (B), measured by a linear potentiometer image of the oil level. Other data can be derived based on the raw data, such as the vertical speed (S).

Fig. 54 shows the predictive control inputs and outputs diagram and Table 13 defines all the parameters related to the predictive controller.

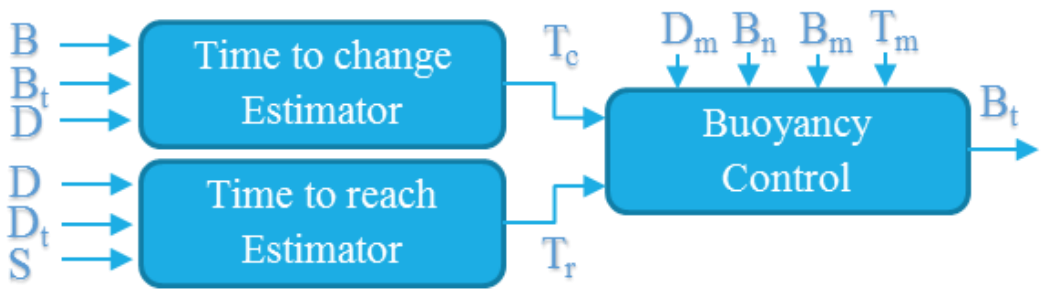



**Fig. 54** Predictive depth control input/output diagram

**Table 13** Definition of the buoyancy control parameters

| **Symbol** | **Definition** |
|---|---|
| D | Current depth (m). D > 0 |
| $D_t$ | Target depth (m) |
| $D_m$ | Margin of tolerance around $D_t$ |
| S | Current speed (m/s). S > 0 ➔ Robot descending |
| $S_m$ | Speed margin |
| B | Current buoyancy (%). Range: 20->95% |
| $B_t$ | Output target buoyancy |
| $B_n$ | Neutral buoyancy |
| $B_m$ | Margin of tolerance around the neutral buoyancy $B_n$ |
| $T_c$ | Time needed to change the buoyancy from B to $B_t$ in (s) |
| $T_r$ | Time needed to reach the target depth based on the current speed of the robot. |
| $T_m$ | Time margin used for security purpose. It compensates eventual inaccuracy in the buoyancy model |

At every second, it is possible to have an estimation of the time needed to reach the target depth (Tr) using equation 9.

$$T_r = ((D_t - D) / S) \tag{21}$$

The buoyancy variation model established in section 4.3.2.2 of this paper enables to estimate the time ($T_c$) needed for changing the robot buoyancy from its current value to the neutral buoyancy. The first step is based on the continuous estimation of the time to

reach ($T_r$) and the time to change ($T_c$) while decreasing the buoyancy value, till the stop condition is reached. We introduce Tm, which corresponds to the time error margin used to compensate eventual inaccuracies in the buoyancy device model.

- If the estimation of $T_r > T_c+T_m$ it means that it is possible to increase the vertical speed of the robot since we have enough time to change the buoyancy to its neutral level. Hence, we decrease the target buoyancy value.
- In the case where $T_r <= T_c+T_m$, then it means we have just enough time to change the buoyancy to the neutral level before the robot reaches its target depth. Hence, we start to increase the buoyancy of the robot progressively

#### 4.3.3.2 Depth Stabilization

Several algorithms can be used for depth stabilization. Among the most used are the PID controllers. However, one of the drawbacks of these controllers is that they require the actuators to operate at full time, which increases the power consumption. In addition, in our robot's case, the buoyancy variation speed is not constant and vary with depth. Furthermore, its variation is not symmetric in both IN->OUT and OUT->IN directions. For that reason, a conventional PID controller is not suitable, which requires the development of an asymmetric PID controller that adapts its parameters with the robot's depth. This will add a lot of complexity to the program and requires a longer time to implement it and to validate its performance. For that reason, we chose to use a heuristic controller for depth stabilization. The latter provides a simple way to control the depth. It is based on heuristic rules that enable to adjust the buoyancy based on the current depth and vertical speed of the robot. If we take the case where the robot depth (D) is below the

target depth ($D_t$), we can establish the following rules, to be executed by priority order:

1) If the robot is below the maximum tolerated depth ($D_t$+$D_m$), then increase the buoyancy.
2) If the robot is below $D_t$ and it is descending, then increase buoyancy
3) If the robot is below $D_t$, but it is ascending fast above a speed margin ($S_m$), then decrease buoyancy.
4) If the robot is below the target depth, and it's ascending slowly below ($S_m$), then the buoyancy actuator is idle.

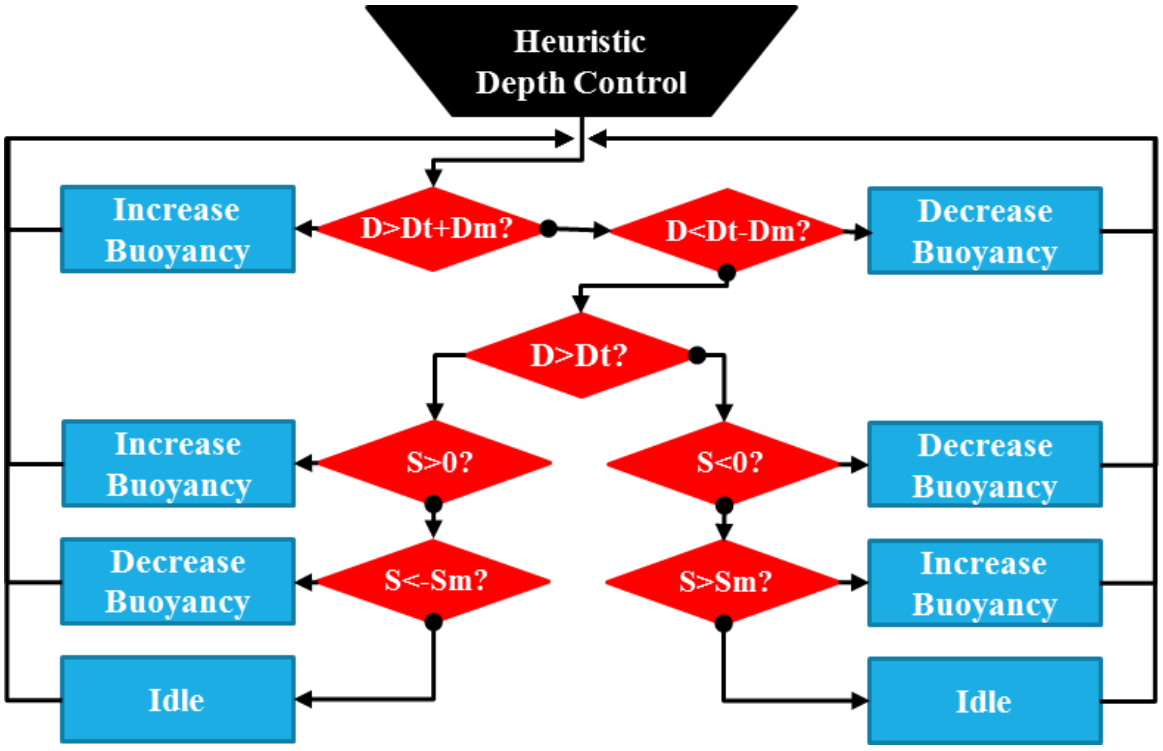



**Fig. 55** Heuristic control flowchart

Fig. **55** shows the flowchart of all the algorithm. It is important to mention here that the buoyancy variation values are limited between $B_n$+$B_m$ and $B_n$-$B_m$.

### 4.3.4 Experiments Results on March 17th, 2016

The depth control experiment was conducted in Toyama Bay on 17th March 2016. The robot was ordered to reach a set target depth equal to 500m and stay there for 15minutes before ascending to the sea surface. Hence, the control program can be divided into 3 main steps: In the first step, the robot uses the predictive depth control to reach the target depth (Step 1). Then the depth stabilization algorithm using the heuristic control is executed to freeze the robot depth for 15 minutes (Step 2). Finally, the target buoyancy is set to its maximum value to bring the robot to the sea surface (Step3). For that purpose, the depth control parameters were configured as shown in Table 14.

**Table 14** Parameter configuration

| **General parameters** | **Value** |
|---|---|
| Target depth (Dt) | 500m |
| Neutral buoyancy (Bn) | 62.5% |
| Buoyancy margin (Bm) | 2.0% |
| Buoyancy control threshold (Bdiff) | 0.2% |
| Buoyancy device accuracy (Berr) | 0.05% |
| **Predictive control parameters** | **Value** |
| Time margin (Tm) | 20s |
| Depth margin (Dm) | 0.5m |
| **Depth stabilization parameters** | **Value** |
| Vertical speed threshold (Sm) | 0.02m/s |
| Depth threshold | 5m |
| Stabilization period | 900s |

Fig. 56 shows the result of the experiment. As it can be observed, the robot managed to reach the target depth and freeze there for 15 minutes before starting the ascent.

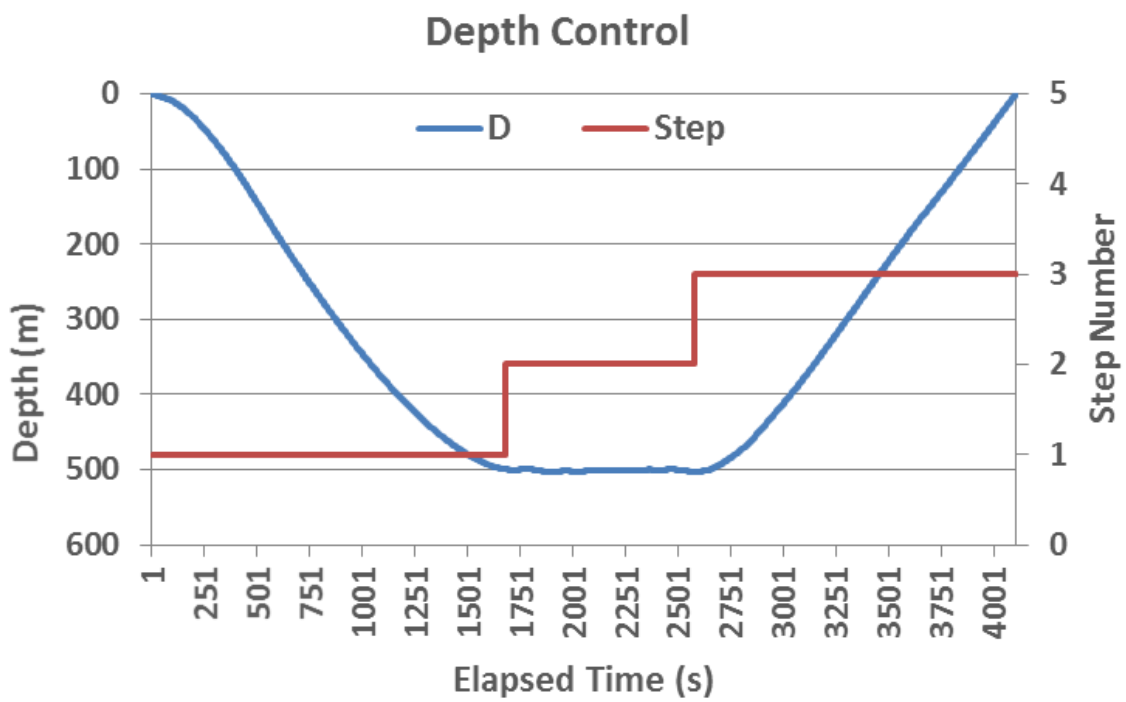
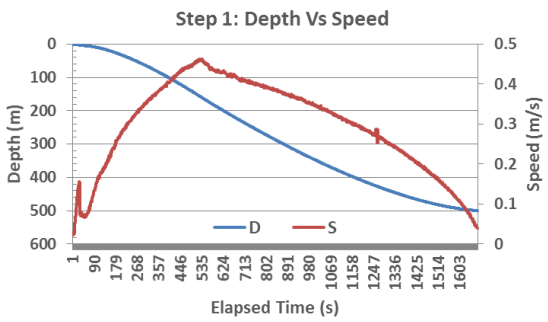
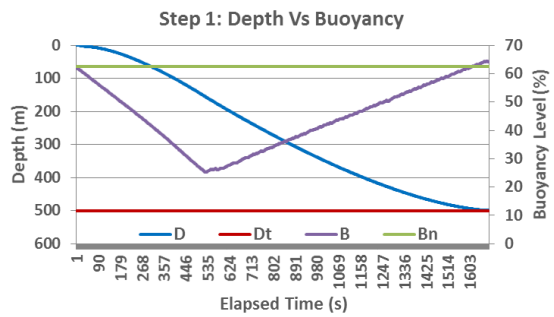



**Fig. 56** Depth Control

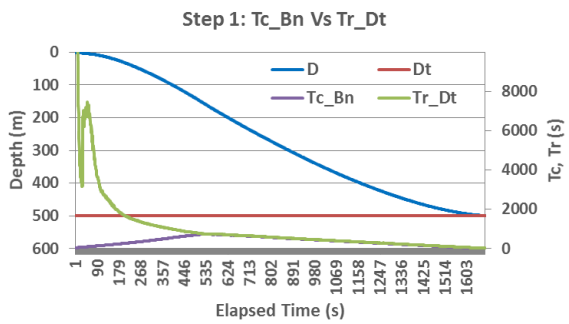
Fig. 57 (a) shows the variation of the robot's vertical speed with depth under the effect of the buoyancy control shown in Fig. 57 (b). The time to reach the target depth ($Tr_{Dt}$) and the time needed to change to the neutral buoyancy ($Tc_{Bn}$) are shown in Fig. 57 (c). The latter parameters define whether the buoyancy should be increased or decreased as explained in the flowchart in Fig. 52. It can be observed that robot managed to reach the target depth with a vertical speed near 0m/s at a buoyancy value near the neutral. The control algorithm succeeded to balance the robot's vertical speed based on the compromise between $Tc_{Bn}$ and $Tr_{Dt}$.




(a) Depth Vs Speed



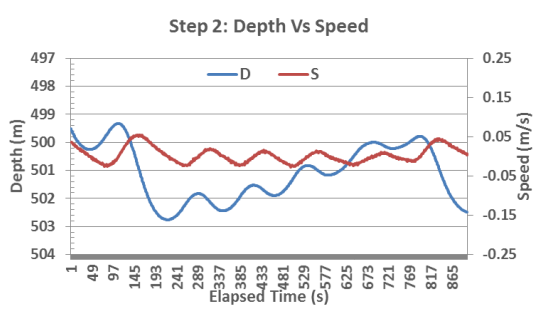

(b) Depth Vs Buoyancy



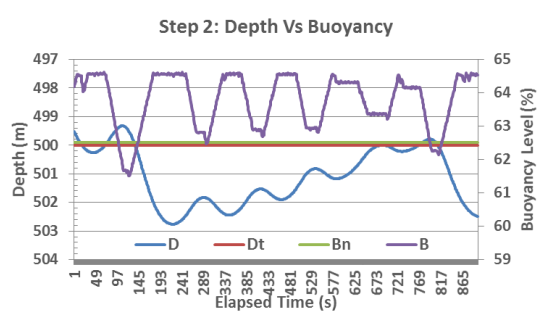

(c) Tc Vs Tr

**Fig. 57** Predictive depth control

Fig. 58 shows the result of the depth stabilization algorithm. The control program managed to maintain the robot's depth within the interval of tolerance around the target depth equal to 5m. In addition, the control program succeeded to limit the robot's vertical speed to less than 5cm/s (check Sm threshold). Furthermore, though the real neutral buoyancy was 64% and not 62.5% as set in the program, the control program succeeded in controlling the robot at the set depth.

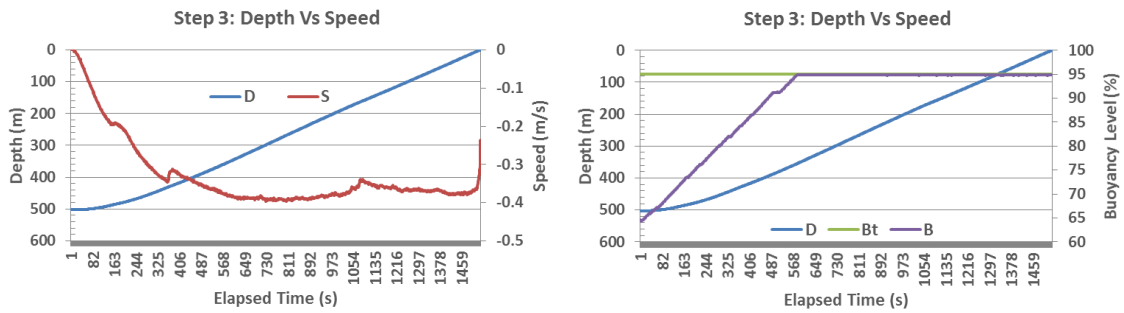



(a) Depth Vs Speed




(b) Depth Vs Buoyancy

**Fig. 58** Depth stabilization




(a) Depth Vs Speed (b) Depth Vs Buoyancy

**Fig. 59** Robot's ascent

Fig. 59 shows the robot ascent. The maximum speed was equal to 0.4m/s. The sudden variation of the vertical speed is due to the change of the wings' angles.

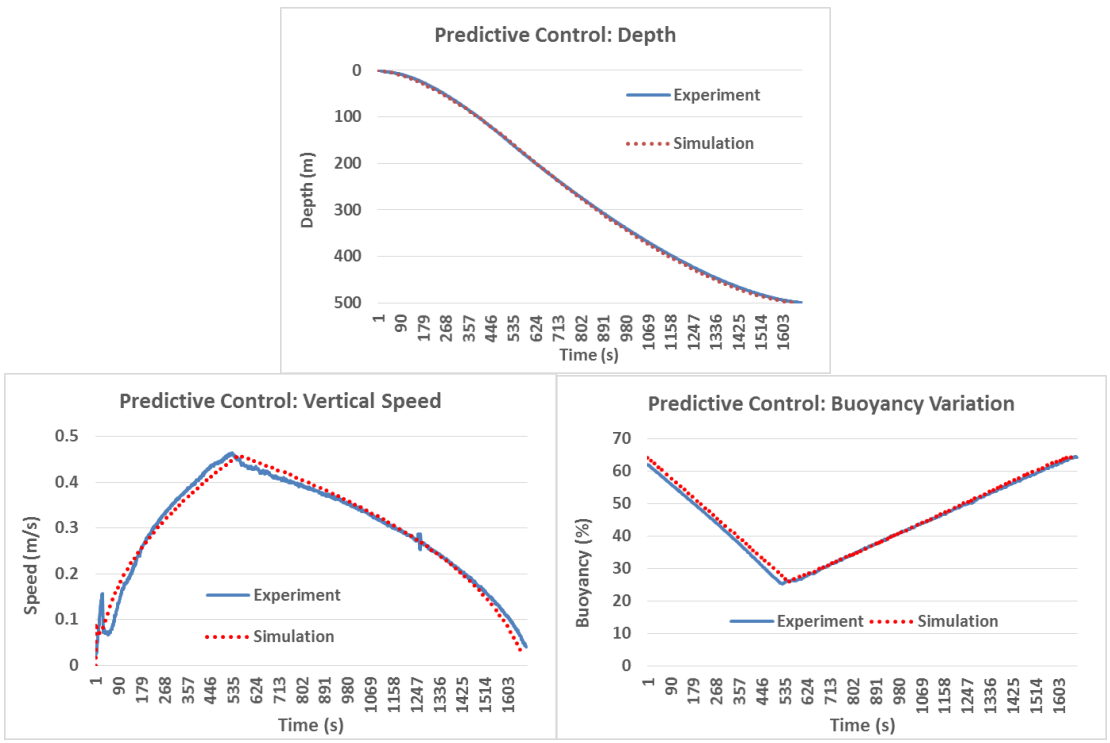



**Fig. 60** Simulation Vs Experiment results

Fig. 60 compares the results of the predictive control obtained in the experiment with simulation results. In the simulation program, we implemented the same predictive control algorithm with the same configuration of parameters that was used in the experiment. The results show a good correlation with the depth, vertical speed and buoyancy variation

profile. This confirms the reliability of not only the robot's model as well as its parameters, but also the buoyancy variation model integrated in the simulation program.

By comparing the experiment and the simulation, we found a good similarity in the obtained results translated with a slight shifting in the time spent to reach the target depth which was less than 1% as shown in Table 15.

**Table 15** Comparison between simulation and experiment results

| | Experiment | Simulation |
|---|---|---|
| Time to reach depth target | 1677s | 1661s |
| Maximum speed | 0.46m/s | 0.46m/s |
| Shifting of time from experiment (%) | | +0.96% |

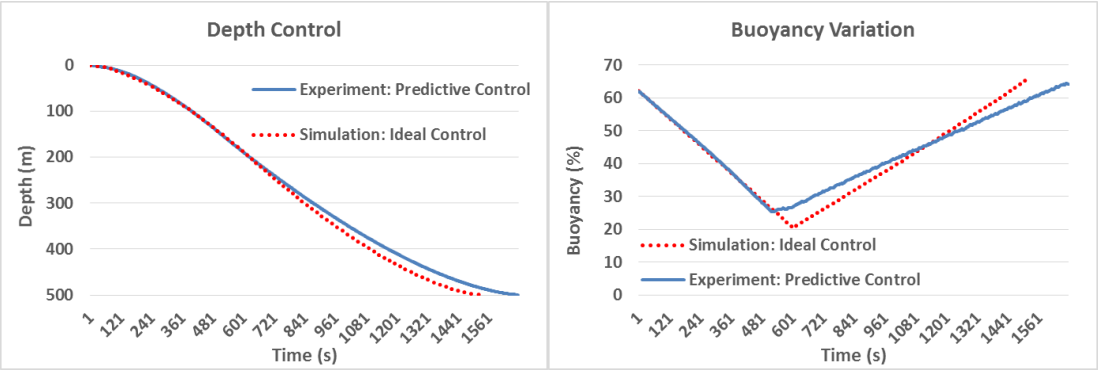



**Fig. 61** Ideal control Vs Predictive control

**Table 16** Comparison between predictive control and ideal control

| | **Predictive Control** | **Ideal Control** |
|---|---|---|
| Time to reach depth target | 1677s | 1540s |
| Maximum speed | 0.46m/s | 0.49m/s |
| Shifting from experiment (%) | | +8.90% |

Fig. 61 compares the ideal depth with the predictive depth control. The ideal depth control corresponds to the case in which we decrease continuously the buoyancy to a certain depth, then we increase continuously the buoyancy in a way that robot reaches the exactly the target depth. To obtain the ideal depth control, we used the simulation program to adjust the depth from which we start increasing the buoyancy through trial and error

method till we the robot stops exactly at the target depth, assuming that the simulation program reflects well the real motion of the robot as demonstrated in Fig. 60.

Table 16 shows a comparative study between ideal control and predictive one. The results show that the time to reach the target depth for the predictive controller is just 9% larger than for the ideal one. In the first part of the predictive control, where the time needed to change the buoyancy from its current value to the neutral buoyancy is less than the time expected to reach the target depth ($Tc_{Bn} < Tr_{Dt} + Tm$), the results are identical. The part where the $Tc_{Bn}$ and $Tr_{Dt}$ values are close, the results start having a shifting between the two curves.

### 4.3.5 Conclusions

The predictive control program succeeded to bring the robot to the target depth without overshoot at a buoyancy value equal to the neutral and a vertical speed close to 0. The predictive depth control algorithm performance is close to the ideal depth control in term of total time to reach the target depth. In addition, the result of the predictive control experiments matches well the result of the simulation. This shows that the simulation program can be used as a reliable tool to mitigate the real robot behavior.

On the other hand, the depth stabilization algorithm managed to keep the robot near the target depth at a limited overshoot and a very small vertical speed. Furthermore, the control program proved its robustness: Though the estimated neutral buoyancy was slightly different from the real neutral buoyancy, the robot managed to reach the exact target depth without problem.

## 4.4 Altitude Control

### 4.4.1 Introduction

There are several operating modes that requires SOTAB-I to get close to the seabed. For example in the photograph mode, SOTAB-I needs to approach the seabed to be able to take neat pictures of the blow out position. Same in the water column measurement mode, the robot is required to dive from the sea surface to near the sea bed to obtain a full water column profile. Hence, it is important to develop a program that controls the altitude of the robot. One way is to use the vertical thrusters to control the altitude. However, there is a risk that they mix up the sediments on the seabed which influences the transparency of the water. In addition, they will disturb the water flow, causing some inaccuracies in the water current measurement. To overcome the previously mentioned weaknesses, we suggest a second method that only uses the buoyancy device as an actuator to control the altitude of the robot.

### 4.4.1 Altitude Calculation

Fig. 62 shows the altitude from the seabed calculation process. At first stage, the raw altitude values are measured. Then, raw values will be subject to tilt and scale corrections followed by low pass filtering aiming to smooth the values. Finally, an altitude evaluation algorithm will be applied to assess the measured values and output the final value of the altitude.

**Fig. 62** Altitude calculation process

#### 4.4.1.1 Raw Data Acquisition

The DVL sampling time is set to 1s. Every second, every beam, of an ensemble of 4 beams that compose the DVL, gives a measurement value of the range. The raw data measured are neither tilt nor scale corrected.

#### 4.4.1.2 Tilt Correction:

It is important to take into consideration not only the pitch p and the roll r angles of the ADCP but also the interactions between them. The following equation gives the exact solution of corrected vertical ranges (Woodgate, 2011):

$$VRange_i = \cos(\alpha_i) * RDI\_Range_i \ / \cos\beta \tag{22}$$

Where β is beam angle equal to 30°, cos β is the factor that compensates for the scaling applied to the data by the RDI software, and $\alpha_i$ is the angle of the beam i = {1, 2, 3, 4} to the vertical plane. The expression of cos ($\alpha_i$) of each beam is given by the following formulas:

$$\cos(\alpha_1) = -\sin r \sin\beta + \cos\beta\sqrt{1-\sin^2 r-\sin^2 p} \tag{23}$$
$$\cos(\alpha_2) = +\sin r \sin\beta + \cos\beta\sqrt{1-\sin^2 r-\sin^2 p} \tag{24}$$
$$\cos(\alpha_3) = +\sin p \sin\beta + \cos\beta\sqrt{1-\sin^2 r-\sin^2 p} \tag{25}$$
$$\cos(\alpha_4) = -\sin p \sin\beta + \cos\beta\sqrt{1-\sin^2 r-\sin^2 p} \tag{26}$$

Where r and p are respectively the roll and the pitch angles measured by the ADCP/DVL sensor.

#### 4.4.1.3 Filtering

The 1€ filter described in section 2.3.2.3 was applied. It is characterized by its easy implementation and fast processing time.

#### 4.4.1.4 Altitude Evaluation

During previous experiments, it was noticed that in some rare cases the reflexive collision avoidance maneuver was activated when the robot was still far from the seabed. Inappropriate ascends causes troubles to the surveying efforts, notably water current measurements since vertical thrusters' activation can disturb the water flow. In addition, it makes the water column survey time longer. Investigation of bottom track data showed that some beams of DVL flagged the existence of obstacles when the seabed is still out of range. Some practical examples of these obstacle detection are given in Table 17.

**Table 17** Example of undesirable obstacle detection

| Case | Beam Altitude Value (cm) | | | | Depth (m) |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | |
| 1 | 0 | 0 | 91 | 91 | 1.55 |
| 2 | 43 | 0 | 0 | 69 | 1.28 |
| 3 | 43 | 0 | 69 | 0 | 0.95 |
| 4 | 69 | 47 | 0 | 0 | 0.48 |

(*) 0 corresponds to the case where the seabed is out of range.

In order to understand the origin of the problem and find the most suitable solution, some statistics were done. A classification of the number of undesired obstacle detections according to the beam number, the water depth and the number of simultaneous beams that had undesirable obstacle detections are shown in

Table **18**.

**Table 18** Statistics of undesired obstacle detections in the second dive on 2014/07/24

| Distribution per beam | | | | |
|---|---|---|---|---|
| Beam | B1 | B2 | B3 | B4 |
| Iterations | 13 | 17 | 17 | 21 |

| Distribution per depth | | | |
|---|---|---|---|
| Depth | D < 5m | 10m > D > 5m | D > 10m |
| Iterations | 61 | 5 | 2 |

| Distribution per number of beams simultaneously activated | | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| Iterations | 56 | 6 | 0 | 0 |

Table **18** shows that more than 90% of the undesirable obstacle detections were near the sea surface within 10m water depth. The faulty detections of the seabed can be caused by the diver when he is working on untying the robot from the ship crane. The sea surface is also where sea water mixes with fresh water originated from rivers that are accompanied with leaves and branches subject to detection by DVL. Obstacle detections may be also due marine life.

It can be noted also that most of the values corresponding to undesirable obstacle detections are within 1.5m range from the DVL. In addition, in almost all cases, the number of simultaneous beams that made undesirable detection is less than two.

The existing filter used in Komatsu-shima treated the case of singular wrong beam detection, but didn't take in consideration the case where more than one beam gives a wrong value. For that reason, further improvements must be done. In order to get the correct altitude measurement and enhance the decision of collision avoidance maneuver activation, a new algorithm for bottom tracking data acquisition was established.

At first stage, the four ranges measured by the 4 beams of the DVL will be filtered and tilt

corrected as previously explained. Following, the term “Out of range” is used to refer to the case where a beam couldn’t detect the seabed. In this case, a default value of 40m will affected to that beam as the maximum DVL range is always less than 40m. Afterward, the 4 beams range values will be sorted in the ascending order in a table. Thus, the “Minimum Altitude” value will correspond to the first element of the sorted table. Next, Fig. 63 illustrates the evaluation algorithm.

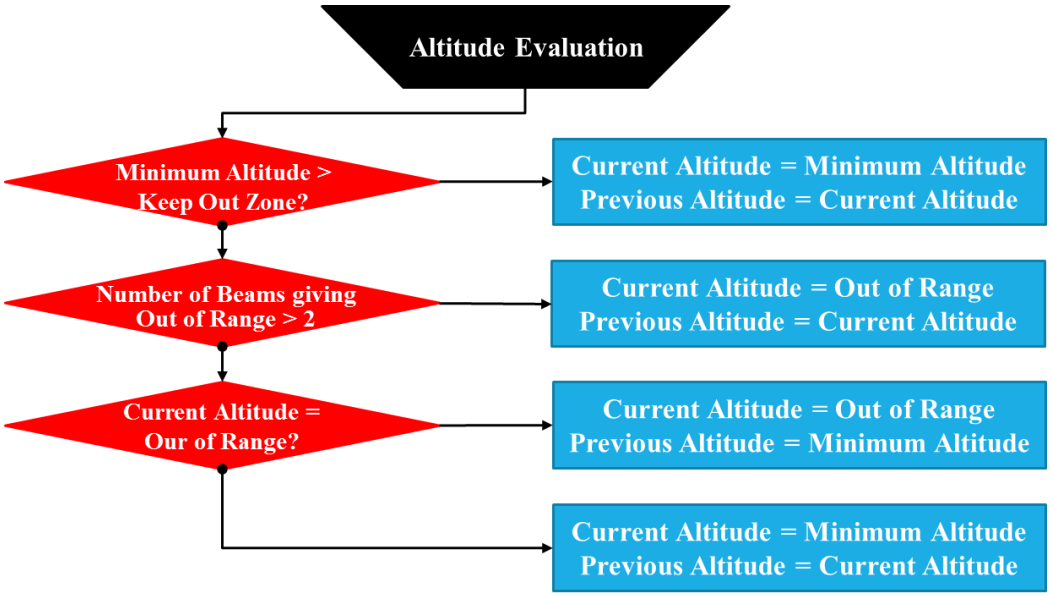



**Fig. 63** Altitude evaluation procedure

The algorithm starts by comparing the minimum altitude to the altitude that corresponds to the beginning of the collision avoidance zone. If the minimum altitude is higher than the collision avoidance zone altitude, the program takes it as final output value “Current Altitude” since it will not activate the collision avoidance maneuver. In the case where three beams or more detect an altitude within the collision avoidance zone, the program checks the previous altitude value. In the case where the previous value was out of range, it means that there is a sudden change of the altitude which strong probably correspond to an undesirable obstacle detection. In that case, the algorithm will set the current altitude

value to out of range. In the next iteration, if three beams or more detect again the seabed, the algorithm outputs their minimum value.

### 4.4.2 Control Algorithm

The method consists of the adaptation of the depth control algorithm detailed in section 4.3 to altitude control. It is important to remind here that the robot is only able to measure the altitude from the seabed when the bottom tracking is active. Hence, the set target altitude should be within bottom tracking range.

Fig. 64 illustrates the flow chart of the altitude control algorithm. It consists of 3 stages of predictive depth control followed by an altitude stabilization control executed to keep the robot at the set target altitude.

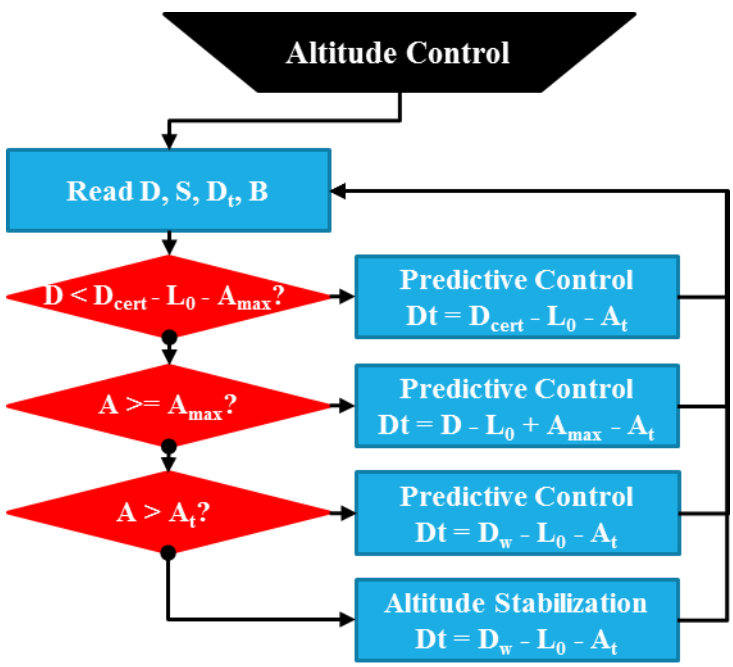



**Fig. 64** Altitude control flowchart

If we consider the case where the water depth is unknown, then there is a possibility that the robot gets close to the seabed at any moment. Hence, to ensure that the robot can stop

descending the buoyancy should be set near the neutral buoyancy value in a way to guarantee that the robot will be able to reach the neutral buoyancy on time. This has a direct impact on the robot speed which becomes very slow and then extend the experiment time and its cost. However, in real experiments, it is possible to get a rough estimate of the water depth from the GPS position. The determination of a safe approximation of the water depth contributes considerably to the reduction of the time needed to reach the target altitude.

Following are the altitude control steps:

**Step 1**:

The robot dives with a fast speed until the depth of the robot reaches the depth limit ($D_{cert}$) of the "Certain depth". The "Certain depth" is the certain minimum water depth value. $D_{cert}$ is input by the user on board before starting the descent. In this first step, the depth control with time estimation scheme is used. At first, the buoyancy control device will decrease the buoyancy. After the buoyancy level becomes lower than the neutral buoyancy of the robot, SOTAB-I will start diving. The buoyancy control device will continue to reduce its buoyancy level down up to 20%, which is set as the minimum buoyancy level of the buoyancy control device, with maximum speed. Then it will increase again its buoyancy level close to the neutral buoyancy level. The purpose of this strategy is that the robot should have enough time to change its buoyancy level to its neutral buoyancy when reaching the target depth. In this step, the target depth ($D_t$) is set at a fixed value as shown in the following equation:

$$\mathbf{Dt = D_{cert} - L_0 - A_t} \tag{27}$$

Where $L_0$ = 2.04m is the distance between the CTD and the DVL sensors, $A_t$ is the target

altitude.

**Step 2**: When the robot is near the certain depth limit, the variable target depth control is started. $D_t$ is calculated as follows:

$$\mathbf{D_t = D_{cert} - L_0 - A_{max} - A_t} \quad (28)$$

After passing the certain zone limit, there is a chance that the DVL will detect the seabed and output its current altitude (A). Therefore, the buoyancy change is limited up to the time (Tr) needed to reach the target depth. In this step, the depth control with time estimation is still being used. However, the target depth ($D_t$) is set equal to the current depth (D) plus the DVL range Amax minus the target altitude ($A_t$). The target depth will continuously change as the depth D of the robot decreases. Hence, it is a depth control with variable target depth. At this point, the buoyancy level of SOTAB-I is already close to the neutral buoyancy. Therefore, there will not be much change in the buoyancy level to ensure that the robot is able to stop when reaching the target depth, as shown in step 2. As a result, the robot will dive at a steady speed.

**Step 3**: When the DVL detects the seabed, the water depth ($D_w$) can be calculated as the sum of the depth D measured by CTD and the altitude (A) measured by DVL taking in consideration the distance ($L_0$) between the two sensors as shown next.

$$\mathbf{D_w = D + L_0 + A} \quad (29)$$

Once the water depth is known, it becomes possible to transform the altitude control to an equivalent depth control using the following equation:

$$\mathbf{D_t = D_w - L_0 - A_t} \quad (30)$$

The water depth $D_w$ is defined as the sum of the depth D measured by CTD and the altitude

A measured by DVL. This step is also carried out by using the depth control with time estimation scheme.

**Step 4**: When the robot is within the range of the target depth plus or minus the depth margin $D_m$, the depth control method is switched from the depth control with time estimation to depth stabilization control program. The depth margin $D_m$ is usually set around 1m as a compensation in the control mechanism of the buoyancy device. SOTAB-I will stay within the target depth for a certain period of time, which has been set on the timer. When the timer reaches zero, the robot will start ascending.

### 4.4.3 Study of the effect of Altitude Control Parameters

We implemented the altitude control program in the robot's simulator program to study the effect of the certain depth and target altitude on its time performance. Results are shown in Table 19 and Table 20.

**Table 19** Effect of certain depth variation on altitude control performance

| Water depth (m) | Target altitude (m) | Certain depth (m) | Time to reach Dt (s) | Average speed (m/s) | Maximum speed (m/s) |
|---|---|---|---|---|---|
| 800 | 3 | 0 | 4722 | 0.17 | 0.18 |
| | | 400 | 3701 | 0.22 | 0.44 |
| | | 600 | 2979 | 0.27 | 0.49 |
| | | 700 | 2605 | 0.31 | 0.5 |
| | | 750 | 2416 | 0.33 | 0.5 |

Table 19 shows that the closer is the certain depth is to the water depth, the faster the robot reaches its target altitude. The time difference is significant if we take the case of a certain depth equal to 0m, which takes twice longer time to reach the target altitude than the case where the certain depth is equal to 750m when the target altitude is equal to 3m.

**Table 20** Effect of target altitude variation on altitude control performance

| Water depth (m) | Certain depth (m) | Target altitude (m) | Time to reach Dt (s) | Average speed (m/s) | Maximum speed (m/s) |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 800 | 0 | 2 | 4646 | 0.17 | 0.18 |
| | | 4 | 4803 | 0.17 | 0.18 |
| | | 8 | 5202 | 0.15 | 0.16 |
| | | 16 | 6872 | 0.11 | 0.12 |

Table 20 proves also that the target altitude choice has its impact on the time response. Target altitudes that are closer to the seabed give faster response.

### 4.4.4 Experiments Results on March 17th, 2016

In this experiment, the robot was ordered to go to a target altitude equal to 9m then freeze there for 5 minutes before ascending to the sea-surface. The exact water depth at the place where the robot was launched was unknown, but the water depth was estimated to be at least equal to 724m. The parameters of the altitude control were set as shown in Table 21.

**Table 21** Altitude control parameters configuration

| | |
|---|---|
| Certain depth | 724m |
| Target Alt | 9m |
| Ascending Timer | 300s |
| BT Min Range | 24 |
| $D_m$ | 0.5m |
| $D_{err}$ | 0.007m |
| $S_m$ | 0.02m/s |
| $B_n$ | 62.5% |
| $B_m$ | 2% |
| $B_{diff}$ | 0.2% |
| $B_{err}$ | 0.05% |
| $T_m$ | 20s |

Fig. 65 shows the experiment result of the altitude control. It can be observed that the robot managed to reach near the seabed and freeze there for 5 minutes before ascending.

From the DVL data, we could measure the water depth which was equal to 766m. The control algorithm was composed of 5 steps. The 4 first steps were explained in section 4.4.2. The fifth step corresponds to the robot ascent.

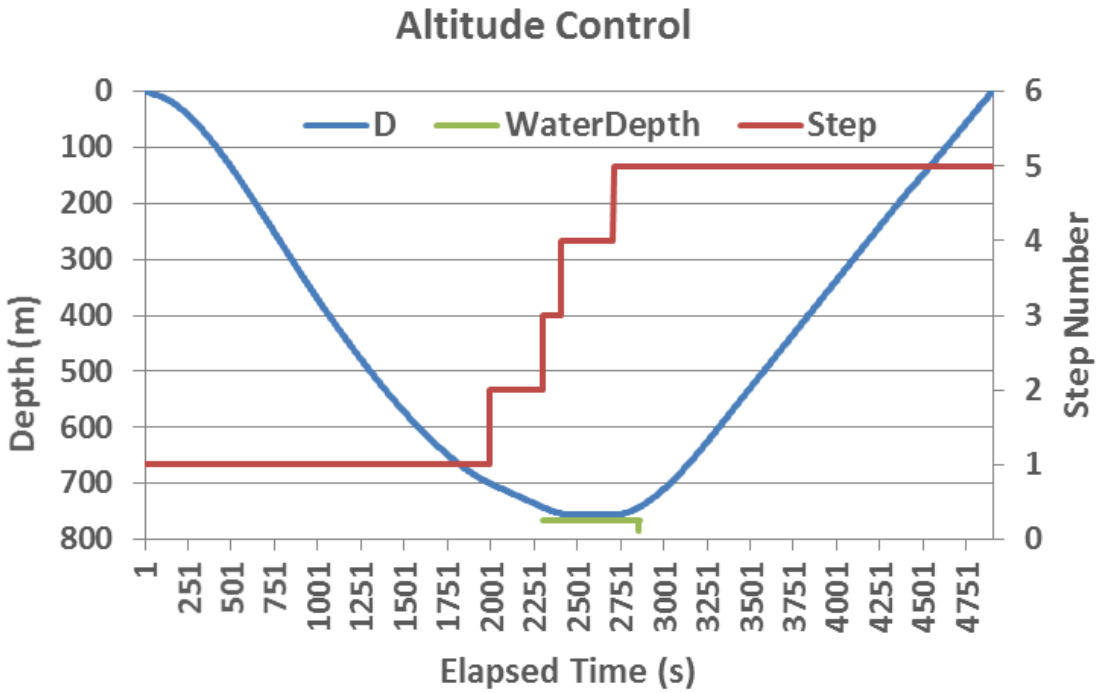
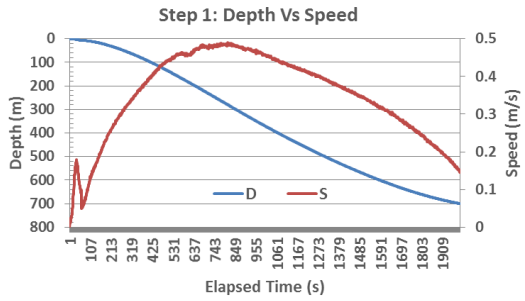


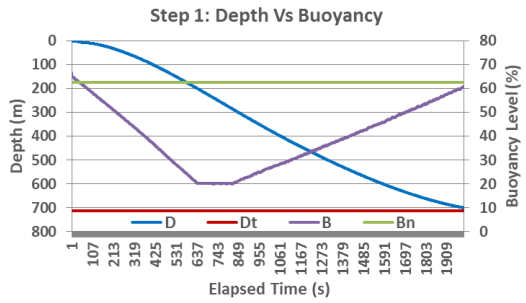

**Fig. 65** Altitude control

Fig. 66 illustrates the details of the predictive depth control applied in step 1 with a set target depth equal to 713m, which can be calculated using equation 10 knowing that the certain depth is defined as equal to 724m and the target altitude is equal to 9m. It can be observed that the robot reached the minimum buoyancy equal to 20% at a maximum vertical speed equal to 0.49m/s and maintained it for 175s, which enabled to reduce the time of the experiment. At the end of step 1, the robot's vertical speed was reduced to less than 0.15m/s and the buoyancy was equal to 61%, which is 3.5% below the maximum value of the estimated neutral buoyancy ($B_{n\_Max} = B_n + B_m = 62.5\% + 2\% = 64.5\%$). At that buoyancy value, the buoyancy device is able to change to reach the neutral buoyancy

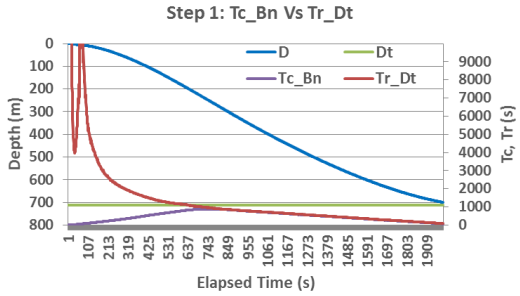
on time in case the robot detects the seabed.




(a) Depth Vs Speed

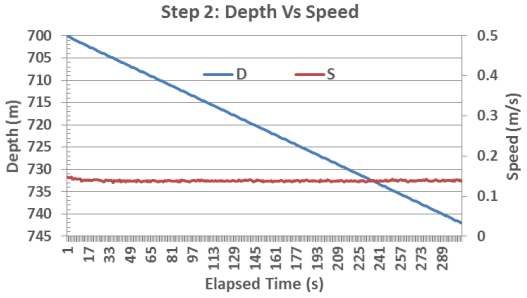



(b) Depth Vs Buoyancy

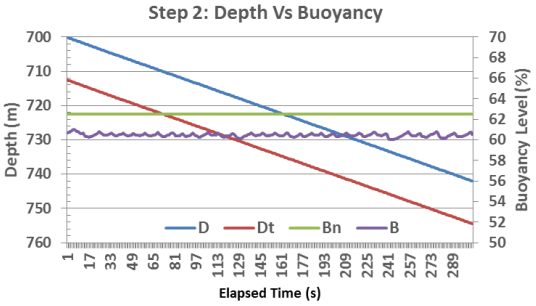



(c) Tc Vs Tr

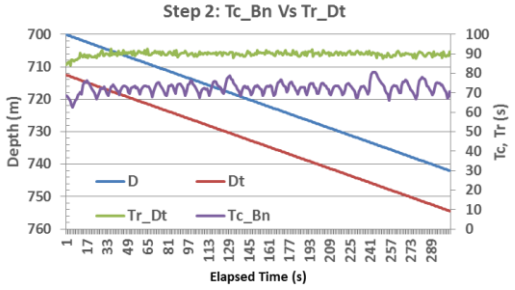
**Fig. 66** Altitude control: Step 1




(a) Depth Vs Speed




(b) Depth Vs Buoyancy



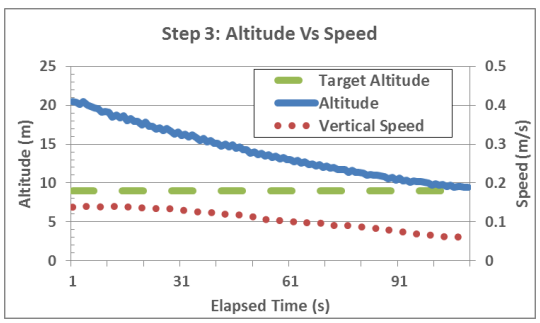

(c) Tc Vs Tr

**Fig. 67** Altitude control: Step 2

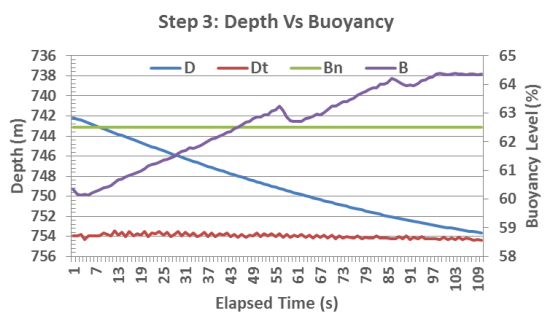
Fig. 67 illustrates the results of the predictive control with variable target depth. At this

step, since the robot didn't detect the seabed, it descends slowly in a way to be sure that the robot will be able to stop at the target altitude. In this step, the robot buoyancy is almost constant. As a consequence, the robot speed is also constant and equal to 0.14m/s.

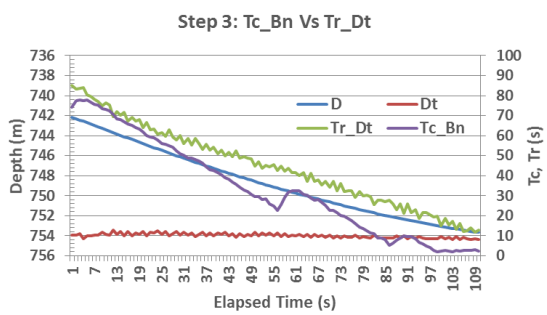
At 742m water depth, the robot detected the seabed and the 3rd step was activated. The target altitude was transformed to an equivalent target depth $D_t$ = 755m. As shown in Fig. 68, the predictive depth control program succeeded to smoothly reach the robot at the target depth with a vertical speed almost equal to 0 and a value of buoyancy very close to the neutral buoyancy.




(a) Depth Vs Speed



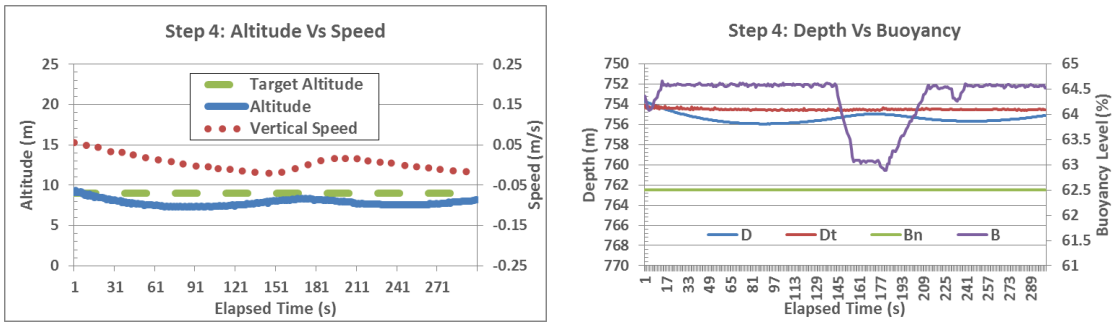

(b) Depth Vs Buoyancy




(c) Tc Vs Tr

**Fig. 68** Altitude control: Step 3

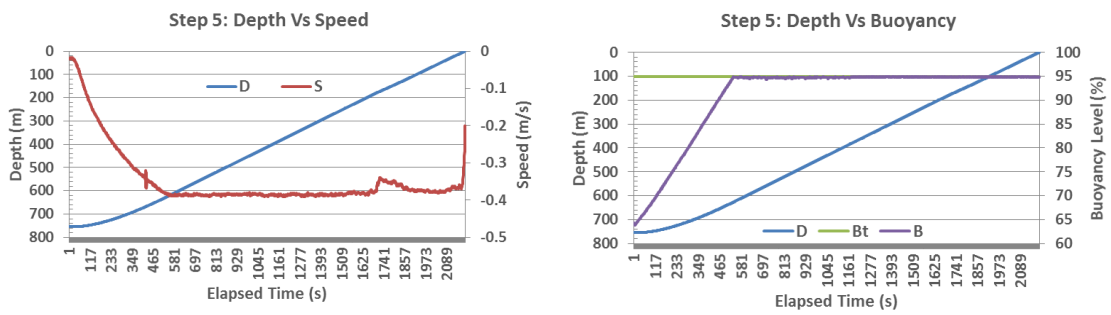
To freeze the robot at the target altitude, a depth stabilizer algorithm is used. The result of its implementation is shown in Fig. 69. It shows that the robot succeeded to keep its depth within (+/-) 1m from set target depth.




(a) Depth Vs Speed (b) Depth Vs Buoyancy

**Fig. 69** Altitude control: Step 4

Fig. 70 illustrates the robot ascent to the sea surface. The maximum speed reached was 0.39m/s




(a) Depth Vs Speed (b) Depth Vs Buoyancy

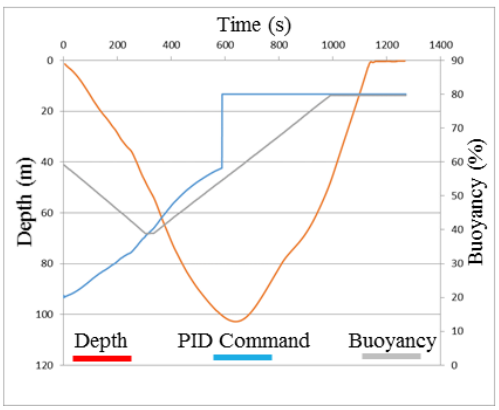
**Fig. 70** Altitude control: Step 5

### 4.4.5 Conclusions

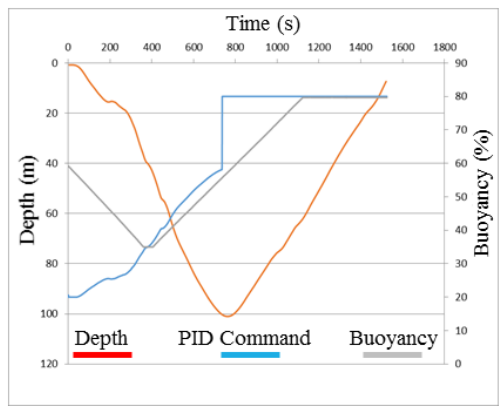
The 3 stages of predictive depth control succeeded to bring the robot to the target altitude with a buoyancy value close to the neutral and with vertical speed almost equal to 0. Besides, the altitude stabilizer succeeded to maintain the robot within 1m from the target altitude. The definition of the certain depth helped to reduce the time to reach the target altitude.

## 4.5 Wings Control

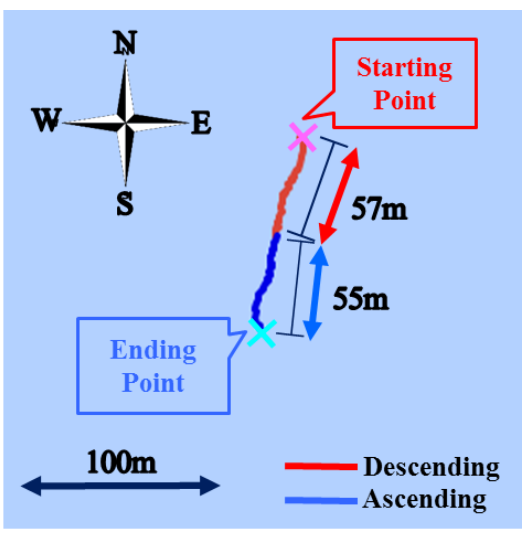
The effect of wings' control was tested. Fig. 71 illustrates time variations of the diving depth of the robot, the command of percentage of buoyancy, and the actual percentage of buoyancy for two cases: case 1 where the wings' angles are set to 0° (Fig. 71 (a)) and case 2 where the wings' angles are set to 0° during descending and 30° during ascending (Fig. 71 (b)). Here, percentage of buoyancy is defined as the ratio of the present buoyancy force to the maximum change of buoyancy of 74.5 N. The initial buoyancy was set at 60% when the robot was floating on the water surface.



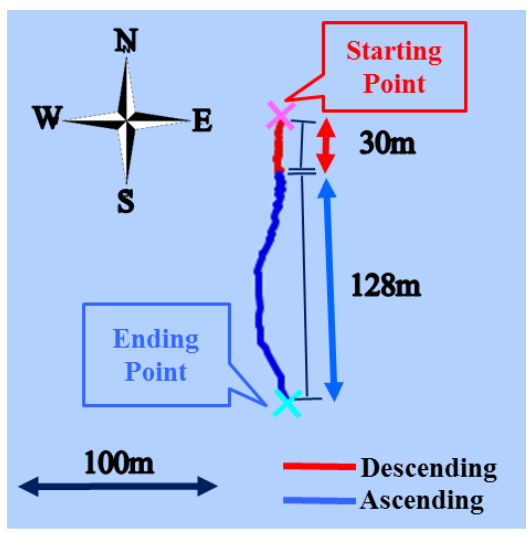

(a) Vertical motion Case 1.




(b) Vertical motion case 2.




(c) Lateral motion case 1.




(d) Lateral motion case 2.

**Fig. 71** Comparison of lateral and vertical motion between case 1 and case 2 described in Table 22.

**Table 22**: Summary of PID controller response.

| | **Case 1** | **Case 2** |
|---|---|---|
| Wing angle | 0° | 0° during descent<br>30° during ascent |
| Target depth | 100m | |
| Reached depth | 102.82m | 101.06m |
| Descending time | 639s | 732s |
| Ascending time | 500s | 798s |

Table 22 summarizes the results of dives in case 1 and 2. It can be noticed that it took more time during descent in case 2 than in case 1, although the wings angles' were set to zero in both cases. The reason was that in case 2, there is a region of depth where the descent speed was decreased to around 200s as a result of the sudden activation of the vertical thrusters due to the detection of an obstacle by the DVL sensor. Vertical thruster activation is a precaution to avoid colliding with the seabed, but in this case the obstacle detection was erroneous. On the other hand, it took more time for the robot in case 2 to ascend than for the robot in the case of wing angle of 0° due to the increase of the drag force on the wings. For a negative buoyancy force of 29.81 N, the descent velocity was equal to 0.381 m/s. For a positive buoyancy force equal to 14.91 N, the ascent velocity was equal to 0.312 m/s.

Fig. 71 (c) and (d) display the comparison of horizontal movement in descent and ascent between cases 1 and 2 in Table 22. Since the USBL system cannot accurately measure the position of the robot when it is near the surface, as the robot's transponder is not within the beam angle of the transceiver, the SOTAB-I's lateral motion shown in Fig. 71 (c) and Fig. 71 (d) correspond to the robot's positions when the water depth is more than 27 m.

In case 1, the drifting distance is 57 m when descending and 55 m when ascending. Knowing that the descent time is 444 s and that the ascent time from 27 m to 100 m is 385 s, the average lateral drifting speed can be approximated to 0.12 m/s for descent and 0.14 m/s for ascent. In case 2, there were no USBL data available while descending from the sea surface down to 75 m water depth. However, starting from 75 m USBL records could be obtained for the remainder of the descent and the entire ascent. Drifting distance when descending from 75 m to 100 m water depth is 30 m, while the drifting distance when ascending from 100 m to 27 m water depth was 128 m. As the descent time was 224 s and the ascent time was 630 s, the average drifting speed can be estimated as 0.13 m/s for descent and 0.2 m/s for ascent. Under the assumption that the water current profile during descent are almost the same as that during the ascent, we can estimate that the robot was exposed to almost the same drifting caused by water currents during the descent as during the ascent. As a consequence, the drifting distance under the effect of water currents can be approximated based on the average lateral speed of the robot in descending as 82m from which the drifting distance under the effect of the wings is determined as 46m. Therefore, the ratio of the horizontal movement to the diving depth reaches about 0.63 and the drifting vertical angle is around 32°.

In the following sections, the wings' angles will be set to 0°, which means that the lateral motion will not be very significant. It will be mainly dominated by the drifting of the robot due to water currents and therefore will have little influence on the measured values. Vertical speed depends mainly on the value of the buoyancy.

## 4.6 PID Collision Avoidance

The collision avoidance maneuver is activated when the position of the robot in the water column (Fig. 39) is beyond the normal operation zone. The DVL device has the ability to measure the altitude from the seabed as well as the absolute velocities of the robot when the seabed is within the bottom tracking range. However, the raw altitude value measured need to be processed since it may be subject to many environmental conditions that may affect its reliability. Hence, the application of an effective collision avoidance control algorithm should be proceeded by a reliable measurement of the altitude value as explained in section 4.4.1. There are two types of collision avoidance algorithms as explained in Fig. 43. In this section, we consider studying the PID collision avoidance, which consists of adjusting the vertical submergence speed of the robot using a PID control of the vertical thrusters. In Fig. 72, the error (e) is the difference between the target speed $S_0$ and the actual speed S of the robot. This error value will be input to PID controller, which includes proportional, integral, and derivative terms. Then the vertical thrusters will receive a manipulated value MV from the output of PID controller and produce an ascending thrust force T.

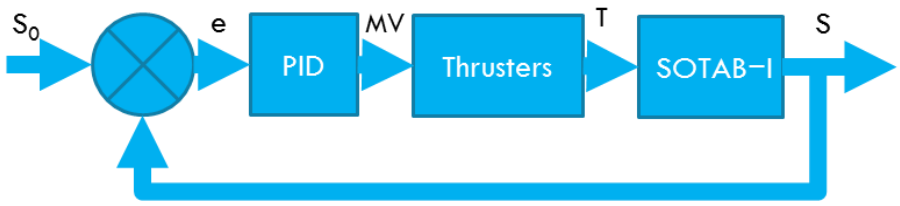



**Fig. 72** Collision avoidance PID controller

The PID coefficients were determined by simulation. In the simulation program, in order to find suitable coefficients, level of buoyancy of SOTAB-I was varied from maximum buoyancy to the neutral point. In addition, collision avoidance zone height was changed

to better evaluate the performance of the regulator. PID parameters were established by tuning. Table 23 shows the corresponding parameters.

**Table 23** PID Collision avoidance parameters

| Definition | Value |
|---|---|
| Keep out zone altitude | 10m |
| Dangerous zone altitude | 5m |
| Proportional coefficient Kp | 1.8 |
| Integral coefficient Ki | 0.25 |
| Derivative coefficient Kd | 2.2 |

The PID controller of the collision avoidance maneuver was tried in Komatsu-shima. The primary objective of the experiment was to test the PID controller of depth based on the buoyancy control device. However, the target depth, set to 50m, was close to the seabed and exceeded the normal operation zone defined in Fig. 39. For that reason, the collision avoidance maneuver was triggered.

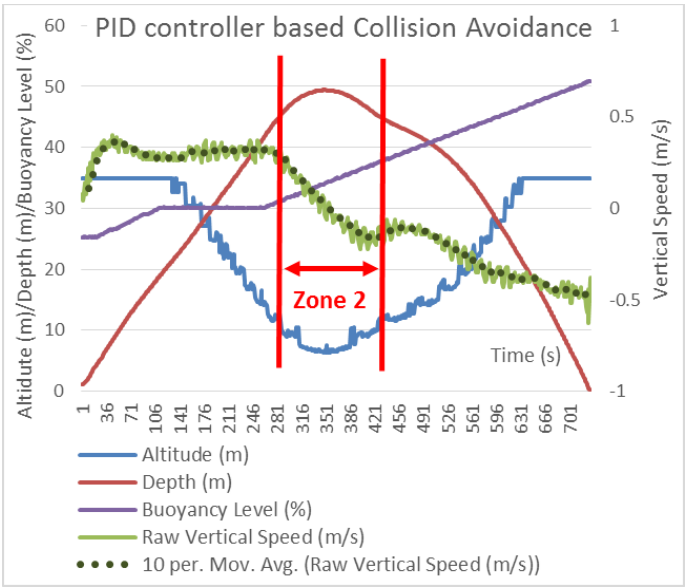



**Fig. 73** Vertical speed control through PID controller

As shown in Fig. 73, in the zone 2 which corresponds to the collision avoidance zone, the PID regulator of speed succeeded to decelerate the submergence of the robot and to freeze

robot's speed near 0 till the buoyancy level of the robot become above neutral buoyancy and the robot started ascending. The variation of the input and the output of the PID speed controller are shown in Fig. 74.

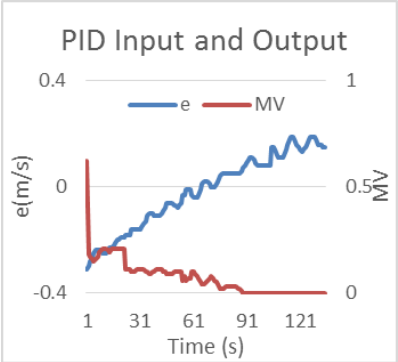



**Fig. 74** PID controller behavior

The graph shows an overshoot of 0.15 m/s shifting from the set speed. It may be noticed in Fig. 73 that the altitude value was not smooth and varied considerably, which lead to the altitude to jump suddenly between the normal operation zone and the collision avoidance zone. This influenced the quality of the result obtained. The new method of altitude data processing suggested in the previous section, which takes into consideration the pitch and roll of the robot may solve this issue in the future experiments.

In case of uneven seafloor, the fast sampling frequency of the DVL and the relative slow motion of the robot in the XY plane make the robot able to react against sudden changes of altitude from the seabed. In the extreme case where the change rate is too fast, the robot will find itself in the keep out zone and then will set the thrusters ascending speed to its maximum value preventing the robot from getting closer to seabed.

## 4.7 Energy Study

### 4.7.1 Power Characteristics

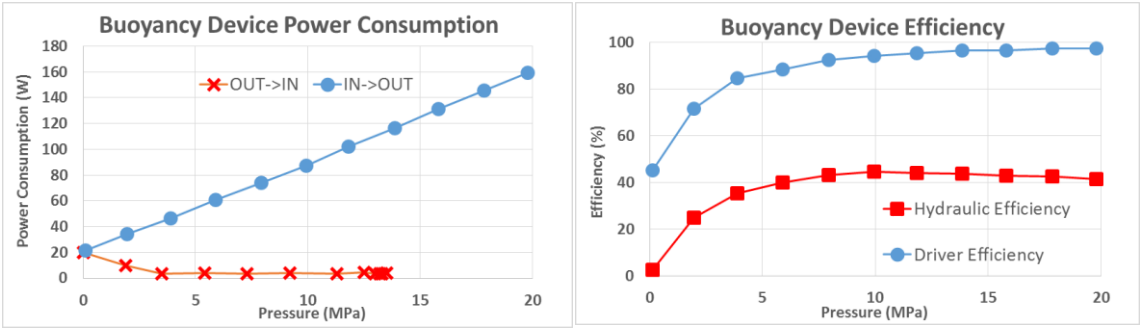
#### 4.7.1.1 Sensors and Control Unit

Table 24 shows the power consumption of sensors installed on SOTAB-I. The battery's total capacity is 4608 kWh.

**Table 24** Power consumption of SOTAB-I devices

| Device | Power Consumption |
|---|---|
| CTD | 3.4 W (Typical) |
| UMS | 60~80 W |
| DVL | 3 W (Typical) |
| USBL | 100 W (Maximum) (during transmission) |
| Iridium | 1 W |
| GPS | 0.4 W (Typical) |
| IMU | 0.22 W (Maximum) |
| Compass | 0.1 (Maximum) |
| LAN | 2.3 W (Maximum) |
| CPU | 16.27 W (Typical), 25.3 W (Maximum) |
| Internal Circuits | 36.56W |

#### 4.7.1.2 Buoyancy Device




(a) Power Consumption (b) Efficiency (OUT->IN)

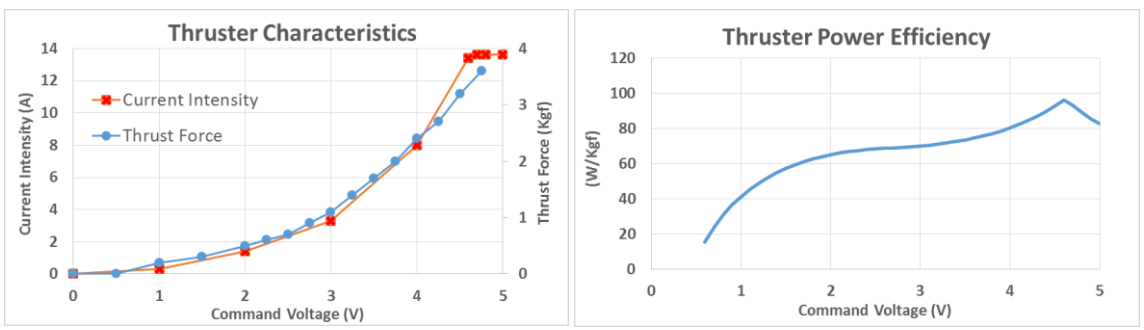
**Fig. 75** Buoyancy device power characteristics

It can be observed in Fig. 75 (a) that the power consumption in the OUT->IN direction increases proportionally to the external pressure. This is explained by the mechanical compressing energy needed to reduce the volume of the air. In the IN->OUT direction,

the power consumption decreases with the increase of external pressure up to 3.6MPa. Beyond the latter value, it remains almost constant. The decrease of power can be explained by the effect of the relaxation of the air compressed that helps to push the plunger. Fig. 75 (b) shows that the buoyancy power driver efficiency increases as higher external pressure is applied. For the hydraulic efficiency, the best efficiency is reached around 10MPa.

Based on the graphs in Fig. 75 (a), a model of power consumption variation with depth and direction was established to evaluate the energetic performances of our developed depth control program. The model was incorporated in the robot simulation program.

#### 4.7.1.3 Thrusters




(a) Power Consumption and Thrust (b) Efficiency

**Fig. 76** Thruster power characteristics

Fig. 76 (a) shows the power consumption and thrust force variation against command voltage. It can be observed that the thrust and the current intensity have similar trends. Fig. 76 (b) is deduced from Fig. 76 (a), it shows the power consumed per one thrust force unit. We can note that in the overall, the increase of thrust force comes at the cost of reducing the power efficiency.

A model of power consumption against the thrust force of the thruster was established and integrated in the robot simulation program.

### 4.7.2 Depth Control: Cases Study

By incorporating the power consumption models of the buoyancy device and the thrusters in the robot simulator program, the power performance of our depth control algorithm could be studied.

**Table 25** Comparison between depth control scenarios

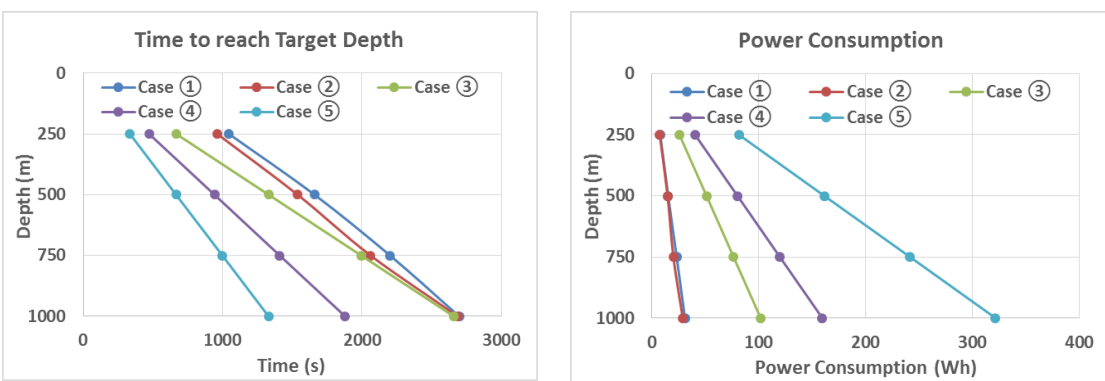
| **Parameter** | **Case ①** | **Case ②** | **Case ③** | **Case ④** | **Case ⑤** |
|---|---|---|---|---|---|
| Actuator Used | Buoyancy Device | | Vertical Thrusters | | |
| Control Method | Predictive Control | Ideal Control | Total Thrust = 2Kgf | Total Thrust = 4Kgf | Total Thrust = 8Kgf |
| Target Depth (m) | 250 | | | | |
| Time to reach (s) | 1047 | 963 | 669 | 474 | 336 |
| (%) of Case ① | | -8.02 | -36.1 | -54.73 | -67.91 |
| Average power (W) | 25.9 | 26.7 | 137.7 | 305.4 | 871.0 |
| Energy (Wh) | 7.6 | 7.1 | 25.6 | 40.2 | 81.3 |
| (%) of Case ① | | -6.58 | 236.84 | 428.95 | 969.74 |
| Target Depth (m) | 500 | | | | |
| Time to reach (s) | 1661 | 1540 | 1332 | 943 | 667 |
| (%) of Case ① | | -7.28 | -19.81 | -43.23 | -59.84 |
| Average power (W) | 32.6 | 32.6 | 137.7 | 305.4 | 871.0 |
| Energy (Wh) | 15.0 | 14.5 | 51.0 | 80.0 | 161.4 |
| (%) of Case ① | | -3.33 | 240.00 | 433.33 | 976.00 |
| Target Depth (m) | 750 | | | | |
| Time to reach (s) | 2203 | 2063 | 1996 | 1411 | 999 |
| (%) of Case ① | | -6.35 | -9.40 | -35.95 | -54.65 |
| Average power (W) | 37.7 | 34.8 | 137.7 | 305.4 | 871.0 |
| Energy (Wh) | 23.0 | 19.9 | 76.3 | 119.7 | 241.7 |
| (%) of Case ① | | -13.48 | 231.74 | 420.43 | 950.87 |
| Target Depth (m) | 1000 | | | | |
| Time to reach (s) | 2698 | 2688 | 2659 | 1880 | 1330 |
| (%) of Case ① | | -0.37 | -1.45 | -30.32 | -50.70 |
| Average power (W) | 41.0 | 38.4 | 137.7 | 305.4 | 871.0 |
| Energy (Wh) | 30.7 | 28.7 | 101.7 | 159.5 | 321.8 |
| (%) of Case ① | | -6.51 | 231.27 | 419.54 | 948.21 |

In these cases, we consider studying the power consumption of actuators in 5 scenarios of depth control. The power consumption of other devices, such as the processing unit and the senses, is not taken into account. Cases ① and ② employed the buoyancy device to control the depth. Case ① uses the Predictive control described in section 4.3.3.1. Case

② corresponds to the ideal control where we assume that neutral buoyancy is accurately determined, water density is uniform along the whole water column and that the water currents in the vertical and the horizontal direction are negligible. Cases ③, ④ and ⑤ use the vertical thrusters for depth control and the buoyancy level is set to its neutral value. Three values of thrust force were studied.

In Fig. 77 (b), we can observe that the increase of the power consumption is almost linear for both buoyancy and thruster. The rate of energy against target depth is higher when using thrusters and increases sharply by doubling the thrust force. The time to reach the target depth is also almost linear for both cases.

Table 25 and Fig. 77 show that the performance of the predictive depth controller is very close to the ideal case in both temporal and energetic aspects. For instance, the power consumption in the ideal case has been just 15%, or less, smaller than the predictive controller. In addition, the time to reach the target depth in the ideal case is just 10%, or less, faster.




(a) Time to reach depth target (b) Energy

**Fig. 77** Comparison between five depth control scenarios

If we take particularly the cases where the target depth is low, the speed performance of the robot when using the thrusters is much higher than the buoyancy control. For example, for the case ④ where the total thrust force was set to 4Kgf, the robot spends half of the time spent by robot using the predictive control in the case ②, though the power consumption was 4 times higher. Nevertheless, if we consider the case of deep-water like for 1000m target depth, the time to reach using the predictive algorithm on the buoyancy device takes just 1.5% longer than in the case where the thrusters are used with a thrust force set to 2Kgf. On the other hand, use of thrusters doubles the amount of energy for use of the buoyancy device. Hence, we can deduce that starting from 1000m water depth, the predictive controller becomes more efficient in both speed and energy than the case where the thruster force is set as 2Kgf. It can also be observed that for depth control using the buoyancy device consumes almost 2.3 times less than the case using thrusters at a total thrust force equal to 2Kgf for a target for almost the same speed performance.

In this study, we only dealt with the power consumption of the actuator. However, in real experiments, actuators and sensors are activated simultaneously. The study of the total power consumption per dive was studied in section 4.7.4

### 4.7.3 Depth Stabilization

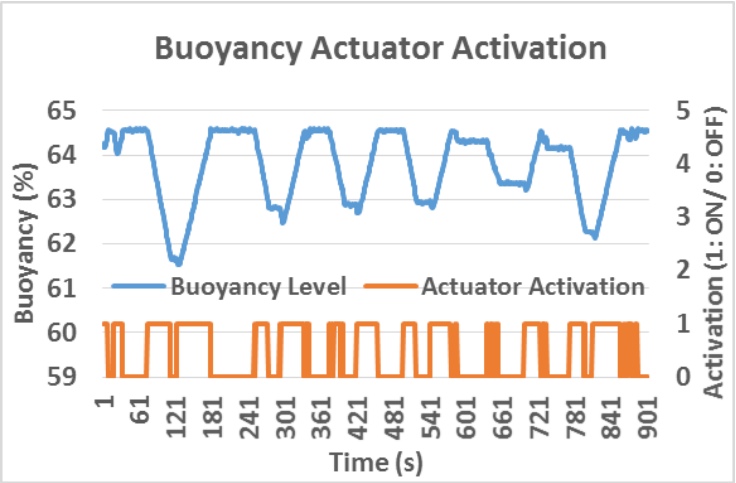



**Fig. 78** Depth stabilization: Buoyancy actuator activation

From an energetic aspect, Fig. 78 shows the periods in which the buoyancy actuator was activated using depth stabilization explained in section 4.3.3.2. The sum of the periods is equal to 424s from a total of 900s. So, in more than 50% of the depth stabilization period the actuator was idle, which contributes to the reduction of the power consumption.

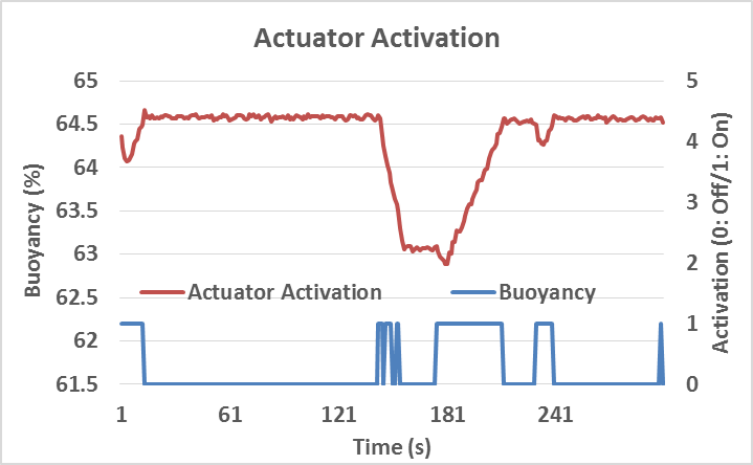



**Fig. 79** Altitude stabilization: Buoyancy actuator activation

In the altitude control experiment for step 4 explained in section 4.4.4, the robot activated the actuator for 70s during the 300s of execution of the algorithm, which means that was idle 76% of the total period (**Fig. 79**).

### 4.7.4 Robot's Battery Autonomy Study

**Table 26** Buoyancy device power consumption for 1000m dive

| | Time (s) | Time (%) | Average power (W) | Energy (Wh) | Energy (%) |
|---|---|---|---|---|---|
| Descent | 2800 | 47.20 | 41.99 | 32.66 | 65.21 |
| Ascent | 3132 | 52.80 | 20.03 | 17.42 | 34.79 |
| Total | 5932 | 100.00 | 30.39 | 50.08 | 100.00 |

Table 26 shows the results of simulation of a dive up to 1000m using the predictive controller and the power consumption of the buoyancy device. The results will be used to estimate the total energy of the robot as shown in Table 27.

**Table 27** Power consumption per dive

| Device | Average power (W) | Energy (Wh) | % of the total Energy |
|---|---|---|---|
| CTD | 3.4 | 5.60 | 1.99 |
| UMS | 70 | 115.34 | 40.90 |
| DVL | 3 | 4.94 | 1.75 |
| USBL | 7.5 | 12.36 | 4.38 |
| Iridium | 1 | 1.65 | 0.58 |
| GPS | 0.4 | 0.66 | 0.23 |
| IMU | 0.22 | 0.36 | 0.13 |
| Compass | 0.1 | 0.16 | 0.06 |
| LAN | 2.3 | 3.79 | 1.34 |
| CPU | 16.27 | 26.81 | 9.51 |
| Internal Circuits | 36.56 | 60.24 | 21.36 |
| Buoyancy | 30.39 | 50.08 | 17.76 |
| Total | 171.14 | 282.00 | 100.00 |

As it can be observed in Table 27, the UMS counts for more than 40% of the total power consumption. The buoyancy device and the internal circuits related to the power drives consumes an amount that is around 38%.

**Table 28** Battery autonomy for 1000m dive

| | |
|---|---|
| Total battery capacity | 4608Wh |
| Time per dive | 5932s (01:38:52) |
| Energy/Dive | 282Wh |
| Number of dives | 16.34 |
| Battery autonomy | 96931s (1 day 02:55:31) |

Table 28 shows that SOTAB-I can perform around 16 dives and last 1 day and 2 hours at

full power. The autonomy of the batteries can be extended by reducing the acoustic uplink frequency and the sensors (mainly the UMS) activation time. It is also possible to deactivate some devices based on the robot position. On the sea surface, acoustic communication doesn't work properly. Thus, it can be switched off. When the robot is underwater, magnetic waves cannot get through to the robot. This means that the GPS receiver, the Iridium communication and the WiFi modules can be switched off. Furthermore, the internal circuit, particularly the power boards, drains significant energy even when the actuators are in idle mode. When the control of the robot's actuators is over, the power supply of the power board should be turned off, allowing to save 36.56W.

### 4.7.5 Conclusions

The comparative study between the ideal controller and the predictive controller shows that the latter's energy performance is close to the ideal. The comparative study shows that the buoyancy device is a lot more efficient in terms of energy than thrusters even though the robot takes longer time to reach the target depth. Thrusters have better speed performance, but results shows that at deep water, the buoyancy device can perform better in some cases where the thruster is employed.

The study of robot autonomy showed that SOTAB-I is able to operate at its full performance for the a whole day without stop and can conduct up to 16 dives up to 1000m water depth while surveying oceanographic data and the dissolution of substances.

## 4.8 Conclusions

In this chapter, we detailed the water column regions and robot operating modes. A pyramidal hierarchy was established to define each control program priority. This architecture enabled to safely interrupt the control program and automatically perform software emergency ascent to avoid dangerous situations when a problem is detected.

Among the contributions in the guidance and control of the robot is the enhancement of the robot's simulator by integrating a buoyancy device simulator. Comparison between simulation and experiments results shows that the robot's simulator can be reliably used to estimate the robot's motion in the vertical axis, which can be very helpful to develop better depth and evaluate their performance before implementing them in real experiments. The implementation of the actuators power consumption models were also very critical to estimate the energy cost of the depth control algorithm and estimate the battery autonomy.

A new method for depth control using the buoyancy control device was developed. A model of the buoyancy variation with time was established. It was built based on the results obtained in high pressure tank experiment and several at-sea experiments. The depth control algorithm is based on the comparison between the time estimated for the robot to change its buoyancy from its current value to the neutral value, and the time expected for the robot to reach the target depth. The method was demonstrated at-sea experiments in Toyama Bay in Japan in March 2016. It showed the ability of the control algorithm to smoothly bring the robot to the target depth without a significant overshoot. The algorithm is characterized by its flexibility and doesn't require a strict determination neutral buoyancy value. A margin of inaccuracy can be customized before performing the

dive. The method could be further adapted to perform an altitude control through a progressive depth control algorithm based on 4 steps. The experiment results showed that it worked properly.

In Suruga Bay experiments in Japan, the effect of wings of SOTAB-I was tested and showed the ability of the robot to move on the lateral plan with an acceptable ratio of the horizontal movement to the diving depth. From this point of view, we can say that the use of the rotational wings to move simultaneously in the vertical and the lateral planes was effective.

The collision avoidance concept was defined with its regions. The PID speed control succeeded to smoothly freeze the robot submergence to the dangerous zone.

The energy study enabled to estimate the robot's battery autonomy under different scenarios. The comparative study between the use of buoyancy device and thrusters proved the power efficiency of the depth control algorithm based on the control of the buoyancy device. The power efficiency of the depth stabilizer algorithm was studied and proved its ability to reduce the energy of the robot. The depth stabilization algorithm managed to reduce the power consumption by setting the actuator idle for almost half of the total period. This is an apparent advantage when compared to the PID and hysteresis controllers where the actuator is always active.

# Chapter 5: Water Survey



`

# 5 Water Surveying

## 5.1 Introduction

A wide variety of methods that deal with underwater oil spills survey and investigation exists, and each presents strengths and weaknesses according to the circumstances and the purposes for which it is deployed. For substances' dissolution measurement, among the most commonly used techniques is the extraction of discrete samples for subsequent analyses (Joye et al., 2011). However, this method has limited temporal and spatial resolution. Additionally, it requires much effort and is time consuming. Furthermore, a risk is that the characteristics of the original collected samples could change during the collecting and handling processes. Other techniques are utilized to track particular substances. For instance, optical sensors, such as colored dissolved organic matter (CDOM) and ultraviolet–visible (UV–VIS) sensors, can be used to continuously measure the concentration dissolved organic carbon (DOC) in a water sample (Lee et al., 2015). These methods can provide continuous information regarding the dissolution of substances, but only for a particular and limited variety of substances. The spilled oil and gas tracking autonomous buoy system (SOTAB-I) integrates an underwater mass spectrometer (UMS) that overcomes the previously mentioned weaknesses. The UMS enables real-time on-site measurements. It is distinguished by its good flexibility and sensitivity as well as its high reliability. It can detect multiple substances' dissolutions simultaneously (Short et al., 2006).

The challenge in water surveying is not only to detect oil and substances dissolved in seawater, but also to obtain other related oceanographic data, as many research programs have demonstrated that temperature (Servio et al., 2002), pressure (Handa, 1990), and

salinity (Yang et al., 2007) are critical factors that considerably affect the formation and dissociation of gas hydrate. In addition, measurement of underwater currents is important for detecting and tracking dissolved gases and for predicting the evolution of the blowout gas in simulation models. Few existing compact systems are able to conduct a complete survey that can measure salinity, temperature, and depth as well as underwater currents and dissolved gases simultaneously. In Deep Spill experiments in Norway, for example, substances' dissolutions were collected using a rosette with sampling flasks towed from a ship. Underwater current data were collected from two separate acoustic Doppler current profiler (ADCP) devices, one on board a second ship and the other connected to an acoustic transmitter and moored to the seabed (Johansen et al., 2003). Using such a technique is helpful to obtain a full survey of the area. However, it requires good synchronization while operating because sensors are mounted in different places. In addition, it requires further resources for deployment, which increases the cost of the survey. The SOTAB-I combines necessary sensors for a full and complete real-time and on-site survey by integrating a UMS, an acoustic Doppler current profiler (ADCP), a conductivity-temperature-depth sensor (CTD), and a camera.

Another challenge related to underwater surveying efforts is the range of the survey. For example, for water profiling, few systems that can provide a deep water profile exist. Existing acoustic water profiling sensors based on ADCP can barely perform measurements of water current distribution beyond the limit of 1000 m depth from the sea surface. Furthermore, in most ADCP devices, resolution of the water layer decreases when higher ranges are applied. Some investigations could lead to a longer range by using multiple ADCP devices. For instance, a solution that implements a dual-meter system

based on a surface- and a bottom-mounted ADCP was suggested (Vogel et al., 2001). The system could measure the water currents' profile up to 1200 m. However, the system lacks the flexibility needed for prompt intervention missions. The SOTAB-I has the capability to perform a high-resolution survey with extended range from sea surface up to a water depth of 2000 m.

In the first part of this chapter, we explain the surveying sensors' configuration and the process employed to obtain oceanographic data and the substances dissolution. The experiments' results in the at-sea experiments in shallow water in Komatsu-shima, the Gulf of Mexico and in deep water in Toyama bay are presented.

## 5.2 Sensors Configuration and Calculation Process

### 5.2.1 CTD

#### 5.2.1.1 CTD data processing

Based on CTD measurements, it is possible to calculate the depth, salinity, density, and speed of sound. Table 29 summarizes oceanographic data that can be obtained with the CTD sensor with their associated symbols and scales.

**Table 29** CTD related oceanographic data

| | Symbol | Unit | Comment |
|---|---|---|---|
| Temperature | $T_{90}$ | [°C] | Given in ITS- 90 scale |
| Conductivity | C | [S/m] | |
| Pressure | P | [dcb] | |
| Depth | D | [m] | |
| Salinity | S | [ ] | Given in practical salinity scale PSS-78 |
| Potential density | $\rho_\theta$ | [kg/m3] | Based on the equation of state for seawater - EOS80 |
| Sound Speed | V | [m/s] | Sound velocity is calculated based on Chen-Millero equation |

Table 29 shows the results for an example of data calculated based on CTD sensor measurements. The depth was calculated using the following formula:

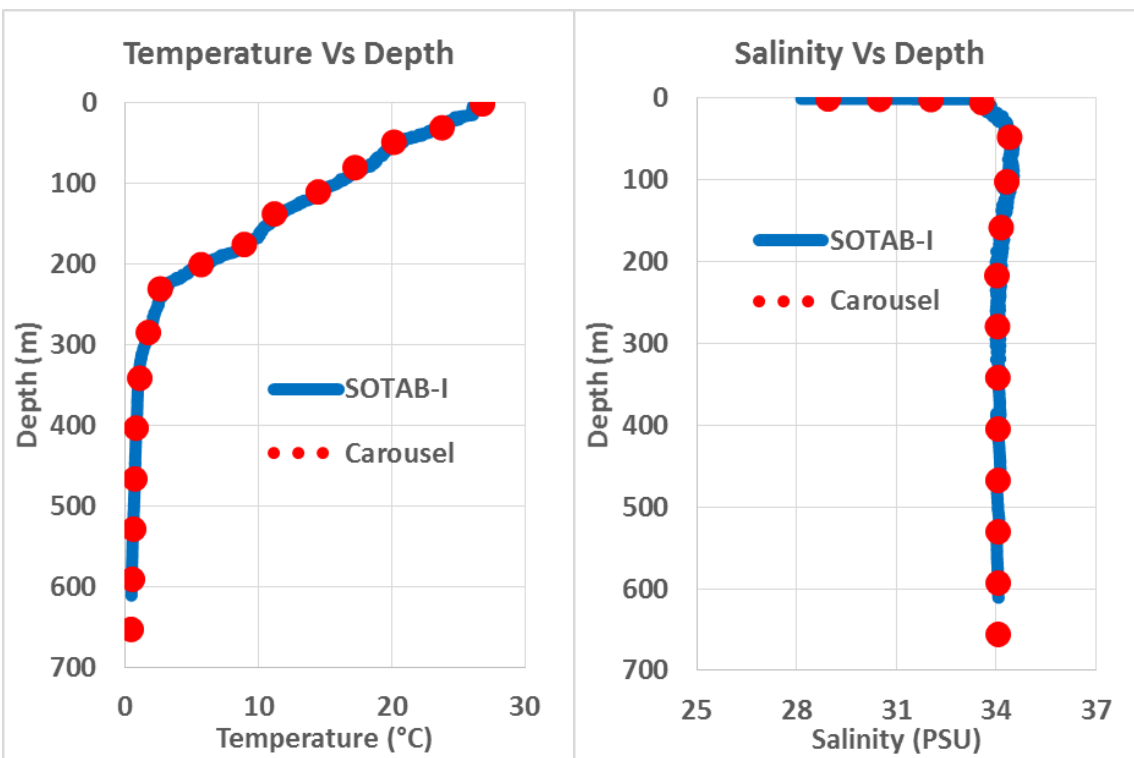
$$\text{depth(m)} = ((((-1.82P * 10^{-15} + 2.279 * 10^{-10})P - 2.2512 * 10^{-5})P + 9.72659)P)/g \quad (31)$$

Where P is the pressure in decibar and g is the local gravity value in m/s$^2$.

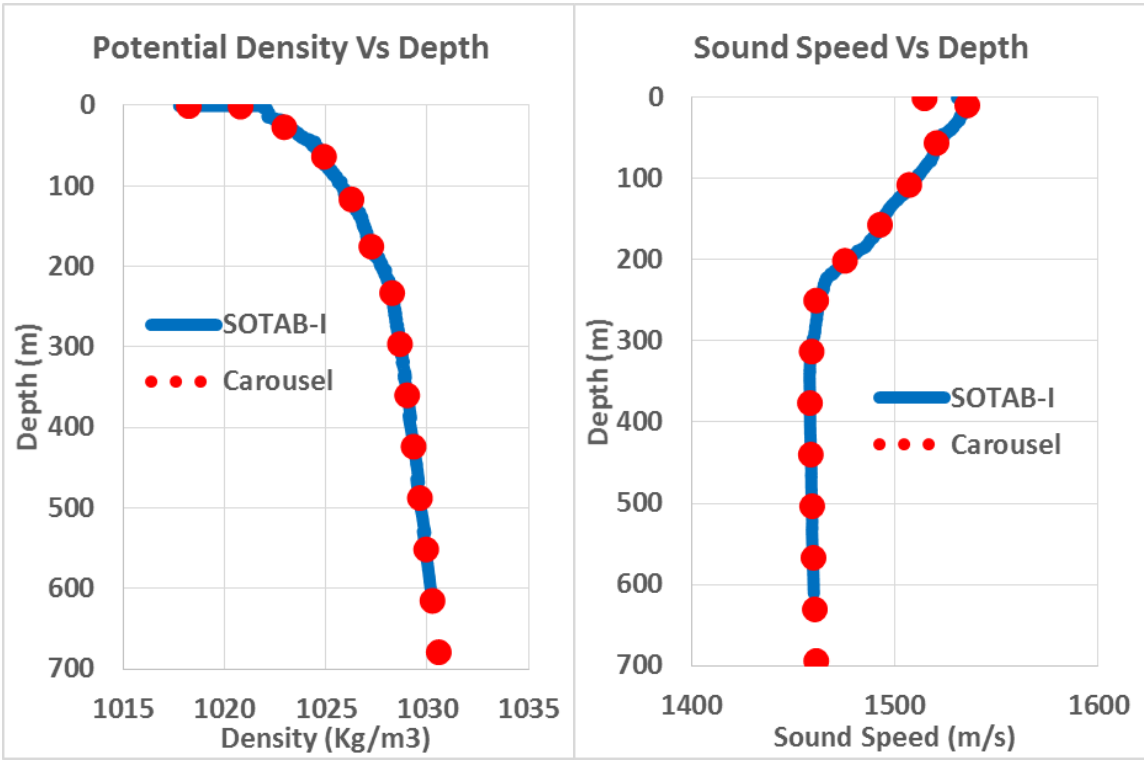
Formulas for the computation of salinity, density, and sound velocity were obtained from (Fofonoff and Millard, 1983) after conversion of temperature from the international temperature scale ITS-90 to IPTS-68.

#### 5.2.1.2 Evaluation of CTD data

The evaluation of the CTD data was conducted in Toyama Bay on 29th July. 2014. Toyama institute deployed their sampling carousel equipped with a CTD sensor at 11:30AM down to 700m water depth. 2 hours later, SOTAB-I was deployed down to 610m. Surveyed data by both CTD sensors were compared.




(a) Temperature (b) Salinity




(c) Potential density (d) Sound speed

**Fig. 80** CTDs measurements of SOTAB-I and Toyama institute's sampling carousel

The graphs in Fig. 80 show that the temperature, the salinity, the potential density and the sound speed water column profiles are very similar. This confirms the reliability of the measurements of the CTD sensor of SOTAB-I and the calculation process of the derived salinity, density and sound speed.

### 5.2.2 ADCP

#### 5.2.2.1 ADCP configuration and characteristics

SOTAB-I configuration was set as water profiling is done every second for 10 water layers referred also as bins with 0.5m thickness. Measurements are configured to be given in the Earth coordinates taking in consideration tilting and bin mapping. Most important characteristics and configuration are summarized in Table 30.

**Table 30** SOTAB-I ADCP configuration

| ADCP configuration | Symbol | Value |
|---|---|---|
| Sampling time | TE | 1s |
| Pings/Ensemble | WP | 1 |
| Nb. of depth cells | WN | 10 |
| Layer thickness | WS | 0.5 m |
| Water profiling mode | WM | 1 |
| Blank after transmit | WF | 0.44 m |
| Salinity | ES | 35 |
| Depth of transducer | ED | 0 m |
| 1st bin distance | | 0.99 m |
| Coordinate transformation | EX | 0x1F (Earth coordinates, use tilts, 3-beam solutions, bin mapping) |

The ADCP is installed in the top bottom of the body. Data of water current are collected when the robot is descending in order to reduce the turbulences that are induced by robot body motion.

The ADCP presents some limitations. Near the sea surface, there are 3 distinct zones where the water column profile of the water currents cannot be measured. Since the ADCP is mounted on the bottom of the robot, the part of the water column located above the transducers, called “Draft”, cannot be measured. Additionally, there is a zone, called “Blanking distance,” situated below the transducers where the transceivers cannot receive the echoes due to their physical properties. The last zone is called “Lag”, which represents the distance between successive portions of the pings transmitted by an ADCP. The sum of the distances of the previously mentioned zones represents the portion of the water column that cannot be profiled by the ADCP near the sea surface. Near the seabed, the side-lobe interference, caused by the strong reflections of the side-lobe energy with the seabed, dominates the echoes from scatterers (Teledyne RD Instruments, 2006).

#### 5.2.2.2 Water Current Profiling Process

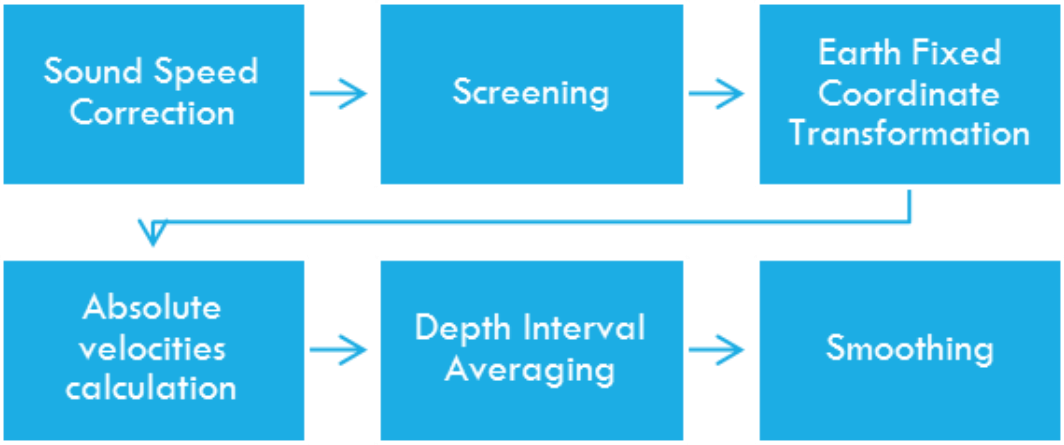



**Fig. 81** Water current measurement process

Fig. 81 shows the steps needed for establishing water current profile.

##### 5.2.2.2.1 Adjustment of incorrect sound speed

The accuracy of velocities in any coordinate system is directly connected to sound speed: an error of 1% in sound speed will result in a 1% error in velocity measurement. The sound speed in the sea water depends on the pressure, the temperature and the salinity. The WHN1200 integrates a thermistor able to measure temperature but it is not equipped with any pressure or salinity sensors. The ADCP calculates sound speed based on the measured temperature and pre-set salinity. However, salinity of seawater is variable, especially near the sea surface. In order to obtain accurate velocity data, the ADCP needs to know the real speed of sound in water. For that reason, sound speed near the transducer is calculated based on the CTD sensor measurements.

It is possible to correct velocity data in post processing by using the following equation:

$$V_{CORRECTED} = V_{UNCORRECTED}\ (C_{REAL} / C_{ADCP}) \qquad (32)$$

Where $C_{REAL}$ is the real sound speed at the transducer, and CADCP is the speed of sound used by the ADCP.

Ranges of cells, to a smaller extent, are also affected by sound speed variations and then are subject to correction. The range may be corrected by using the following equation:

$$L_{CORRECTED} = L_{UNCORRECTED}\ (C_{REAL} / C_{ADCP}) \tag{33}$$

Where

$L_{CORRECTED}$: Corrected range cell location

$L_{UNCORRECTED}$: Uncorrected range cell location

#### 5.2.2.2.2 Screening

This step is performed automatically by the ADCP. Velocity data are subject to four kinds of screening: the correlation test, the fish rejection algorithm, the error velocity test, and the percent good test. At this stage, the ADCP checks the reasonableness of the velocity components for each depth cell and flags bad data.

#### 5.2.2.2.3 Transformation to Earth fixed coordinates

At first stage, the ADCP transforms vector of beam velocities to the vector of velocity components in the instrument fixed coordinate system. The ADCP was configured to convert the data to Earth coordinates (East, North, Up) based on tilt and heading data.

#### 5.2.2.2.4 Calculation of absolute velocity

The robot speed VSOTAB-I should be added to the measured relative water current velocity VADCP in order to obtain the absolute velocity V of water currents. V can be obtained using the following equation:

$$V = V_{ADCP} + V_{SOTAB-I} \tag{34}$$

SOTAB-I can provide robot velocities from the GPS sensor when the robot is on the sea surface, from the USBL positioning system when the robot is in the middle zone, or from the DVL when bottom tracking is active or. In this section, water current measurement

will be treated in the case where bottom tracking is active.

#### 5.2.2.2.5 Depth interval averaging

When the robot is descending along the water column, water profiles of ADCP depth cells overlap, giving multiple measurements for each water depth. Having a high density of measurements helps to reduce random errors. Following, we will refer to ADCP depth cells by "bins" in order to differentiate it from the depth cells of water column. Each ADCP bin measures water current at its corresponding depth. At first step, it is important to calculate the corrected depth associated with each bin ($Bin_{iDepth}$) given by

$$Bin_{iDepth} = D_{CTD} + \cos\Theta * \cos\Phi * (D_0 + Bin_{1Dist} + WS * (Bin_i - 1)) \quad (35)$$

Where $D_{CTD}$ is the depth value calculated based on CTD sensor pressure data, $D_0$ is the distance between the CTD sensor and the ADCP, WS is the bin thickness defined in Table 30 and $Bin_{1Dist}$ is the distance to the middle of the first bin.

At second stage, after depths are corrected, depth and its associated velocity of each bin will be input to a depth interval velocity averaging program as shown in Fig. 82.

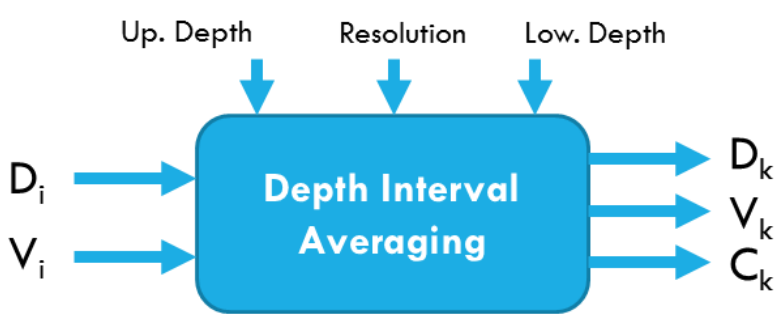



**Fig. 82** Depth interval averaging program inputs and outputs

The water column will be divided into a number of depth layers $D_N$ with R resolution between a certain lower and upper depth. $D_N$ is calculated using the following formula:

$$D_N = |\text{Upper Depth} - \text{Lower Depth}| * \text{Resolution} \quad (36)$$

Water velocities $V_i$ at depth $D_i$ will be averaged within DN discrete depth intervals. For

each depth interval $D_k$, average velocity value $V_k$ with a certain coefficient $C_k$ corresponding to the number of samples measured. For each depth layer, the program makes the sum of the water currents and then divide it by the number of samples measured within its range.

Fig. 83 shows a comparison water current distribution given by the first bin and the water profile obtained after considering all velocities values of the other bins overlapping in the same depth cell. In this graph, the depth cell resolution was set to 0.25m.

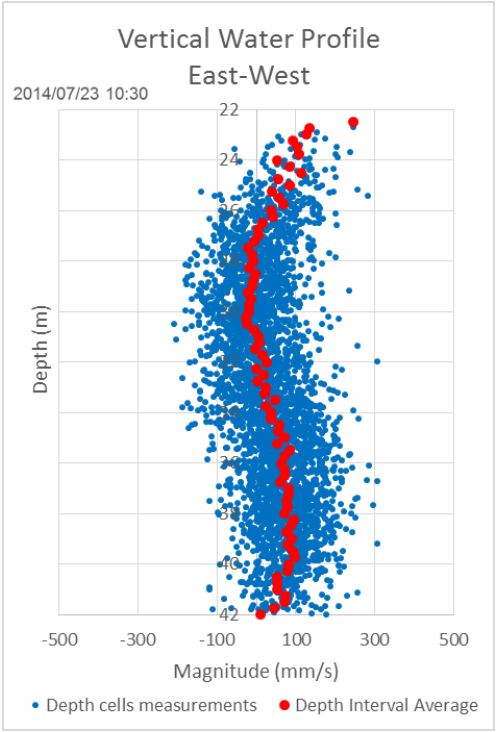



**Fig. 83** Effect of depth interval averaging

#### 5.2.2.2.6 Smoothing

In the previous step, we associated with each depth cell a coefficient that reflects the density of measurements at this depth. The number of samples will be the coefficient that will be associated with each depth layer when calculating the moving average

$$V_{iF} = \frac{\sum_{i+n/2}^{i-n/2}(V_i * C_i)}{\sum_{i+n/2}^{i-n/2} C_i} \quad (37)$$

Where n is the number of depth cell to be averaged

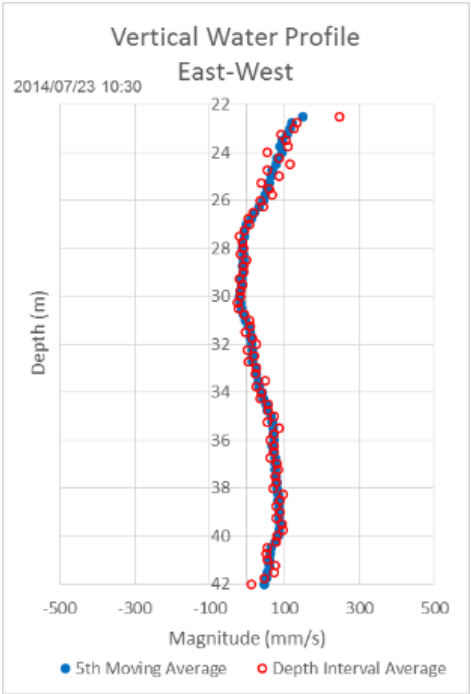



**Fig. 84** Effect of moving average filtering

Fig. 84 shows the effect of five points moving average. It shows that water profile is smoothed particularly in the areas where the density of measurements is the lowest notably in the upper at the beginning of bottom tracking and lower layers at the deepest water depth reached by the robot where not all bins were able to measure water profile.

### 5.2.2.3 Evaluation of the ADCP Data

At-sea experiments were conducted in Komatsu-shima in Japan on the 23rd and the 24th of July 2014 (Fig. 85 and Fig. 86). The mother ship employed in the experiments is the Fukae-maru of Kobe University. The ship length is 49.97m and its breadth is 10m. It is equipped with a crane used to lift the robot from/into the water. The ship has an ADCP for measuring water currents (Sugii et al., 2014).

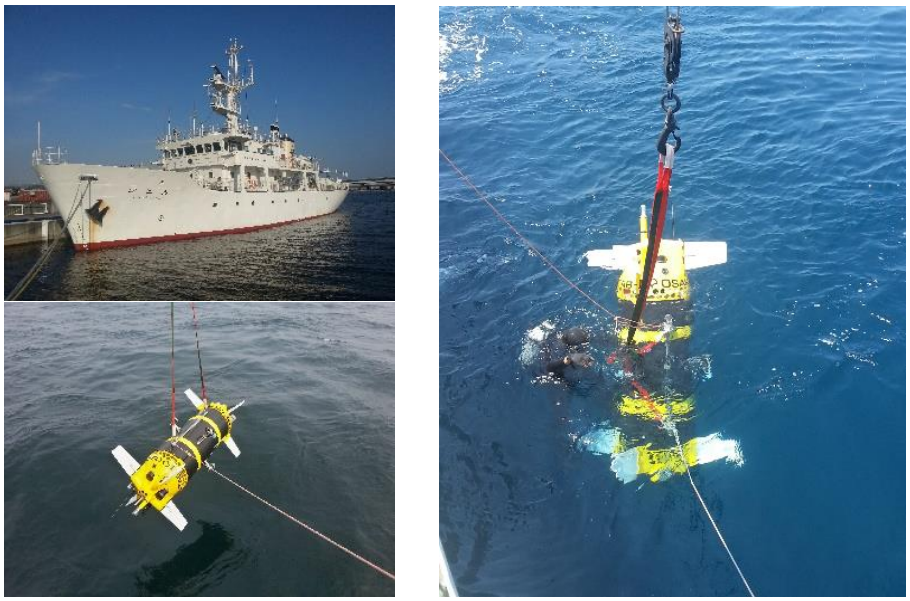


**Fig. 85** SOTAB-I Deployment in Komatsu-shima

Five dives in total were carried out in shallow water at a water depth around 56m. Weather conditions were favorable as air temperature was between 24 and 32 °C and wind speed didn't exceed 18km/h.

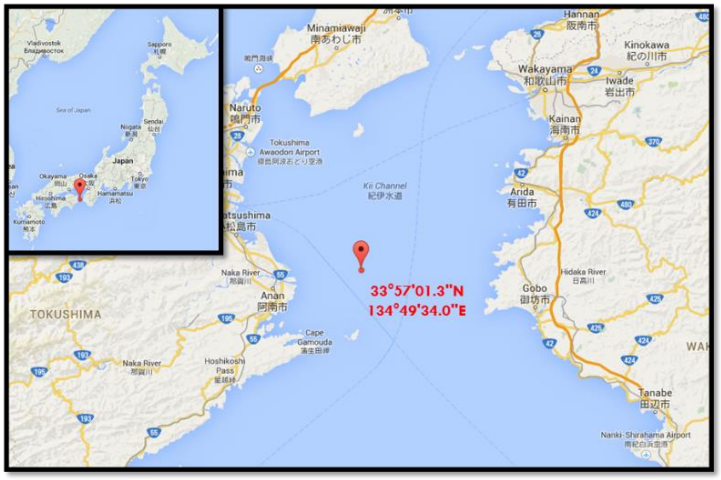



**Fig. 86** Experiments place

Fukae-maru was equipped with an ADCP Broadband 307.2 kHz configured to perform water profiling every minute for 40 water layers with 2m thickness. Main characteristics and configuration are summarized in Table 31.

**Table 31** Characteristics and Settings of Fukae-maru ADCP

| ADCP | FUKAE-MARU |
|---|---|
| Reference | RDI Broadband |
| Frequency | 307.2kHz |
| Sampling Time | 60s |
| Pings/Ensemble | 23 |
| Nb. Of Layers | 40 |
| Layer thickness | 2 m |
| 1st bin range | 3.94m |
| Standard deviation | 6.6cm/s |
| Range | 110m |

SOTAB-I configuration was set as water profiling is done every second for 10 water layers (bins) with 0.5m thickness, which covers a range equal to 5m. As it can be observed in Fig. 87, the standard deviation up to 5m range is lowest and it is equal to 3 cm/s, so we can assume the first 10 layers gives the most accurate results.

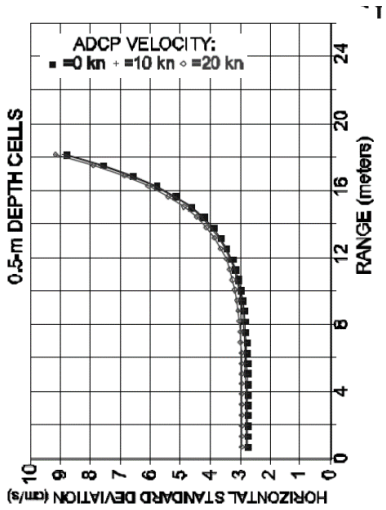



**Fig. 87** Single ping standard deviation of 1200 kHz systems

The ADCP device measures the relative water currents. So, the robot speed $V_{SOTAB\text{-}I}$ should be added to the measured relative water current velocity $V_{ADCP}$ in order to obtain the absolute velocity V of water currents. V can be obtained using equation (34).

Depending on the robot's navigation region (check section 3.2.3 for more details), the speed measurement of the robot changes as explained next.

- **Sea surface zone**: Absolute water current profile V obtained by summing $V_{ADCP}$ and $V_{SOTAB\text{-}I}$ obtained by time derivation of the GPS position. (ADCP+GPS) refers to the absolute water current profile V calculated in this region.
- **Middle zone**: ADCP+USBL: Absolute water current profile V obtained by summing $V_{ADCP}$ and $V_{SOTAB\text{-}I}$ obtained by time derivation of the USBL position. (ADCP+USBL) refers to the absolute water current profile V calculated in this region.
- **Bottom tracking zone**: ADCP+DVL: Absolute water current profile V obtained by summing $V_{ADCP}$ and $V_{SOTAB\text{-}I}$ measured by the DVL bottom tracking. (ADCP+DVL) refers to the absolute water current profile V calculated in this region.

The aim of this section is to assess the accuracy of the water currents measurement process used in SOTAB-I in each region by comparing the water currents profiles measured by Fukae-maru to the ones obtained by SOTAB-I using various methods.

For the case of Fukae-maru ship, the experiment was conducted in shallow water at a water depth around 56m. The ship absolute water current profile was calculated based on the ship GPS.

Fig. 88 compares the water currents profile of Fukae-maru with SOTAB-I profile near the sea surface. SOTAB-I speed was measured based on the time derivation of the drifting distance measured by the GPS on the sea surface (Table 32).

**Table 32** SOTAB-I drifting speed calculation on the sea surface

| Time | Relative position (East-West) | Relative position (North-South) | Robot's speed (East-West) | Robot's speed (North-South) |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 9:41:06 | 33.272 | -143.629 | 0.112 m/s | -0.250 m/s |
| 9:43:32 | 49.63 | -180.119 | | |

During the same period, the relative water currents measured the robot's ADCP at each water layer will be averaged. At the sea surface, SOTAB-I can measure the water current from 3m, which represents the depth of the first bin, down to 7.5m corresponding to the depth of the last beam, which is bin 10.

The first water layer depth that can be measured by the Fukae-maru ADCP is situated at 5m water depth. It is the sum of the first bin range (3.94m) and the depth at which the ADCP sensor was setup on the ship (1m).

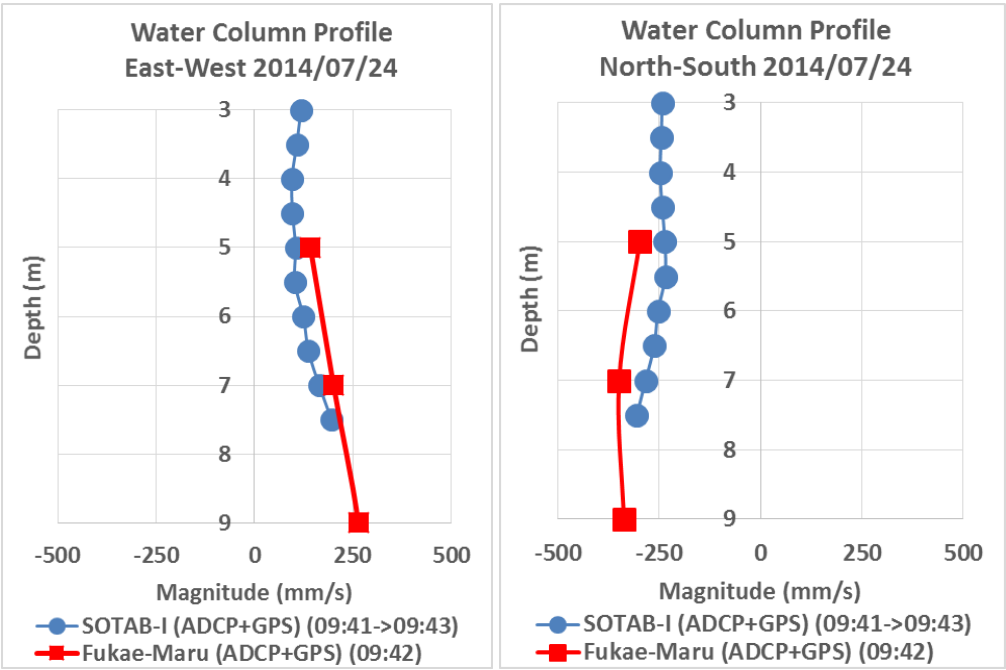



**Fig. 88** Comparison between SOTAB-I and Fukae-maru water currents profiles near the sea surface

Fig. 88 shows strong correlation between SOTAB-I and Fukae-maru profiles. We can conclude that the calculation method of the water profile using SOTAB-I near the sea surface is reliable and accurate.

Next, we consider studying the performance of the water currents measurements near the seabed. In this experiment, the robot's DVL bottom tracking started when the robot reached 22m water depth down to 43m. Hence, if we consider the distance of the first bin from the CTD sensor (3m), the water profile can be measured starting from 25m depth. The maximum depth where the robot can measure the water current is determined by the maximum depth that the robot reached (43m) added to the distance of the last bin (bin 10) from the CTD sensor (7.5m) giving 50.5m.

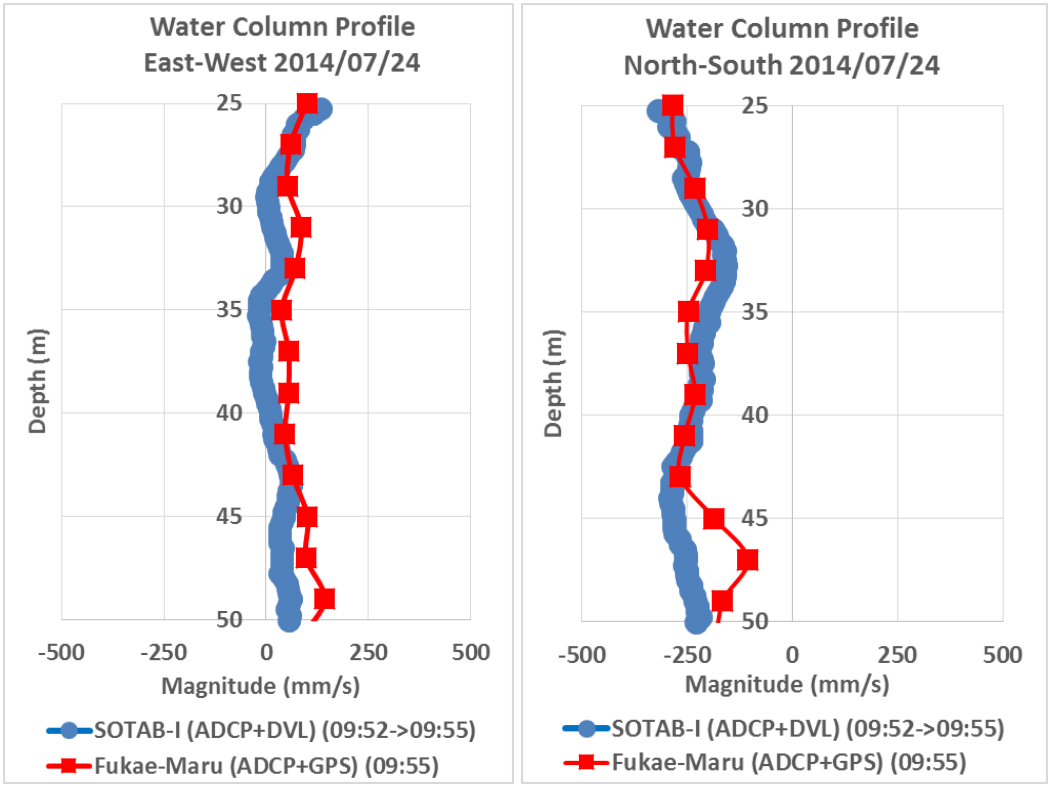



**Fig. 89** Comparison between SOTAB-I and Fukae-maru water currents profiles near the seabed

Fig. 89 shows that water profile measured by SOTAB-I is in good agreement with the Fukae-maru profile. Water current direction as well as its curve trend are very similar, in both directions. The slight difference may be explained by the temporal and spatial

variation of SOTAB-I and Fukae-maru positions. In addition, water currents are varying over time. Finally, the resolution of the two compared ADCPs is different as SOTAB-I has better resolution enabling it to get high resolution profiling and higher density of measurements which contribute to the decrease of random errors.

Following, the water profile in the middle zone was calculated. The USBL data has poor accuracy on the sea surface and in the dead signal zone.

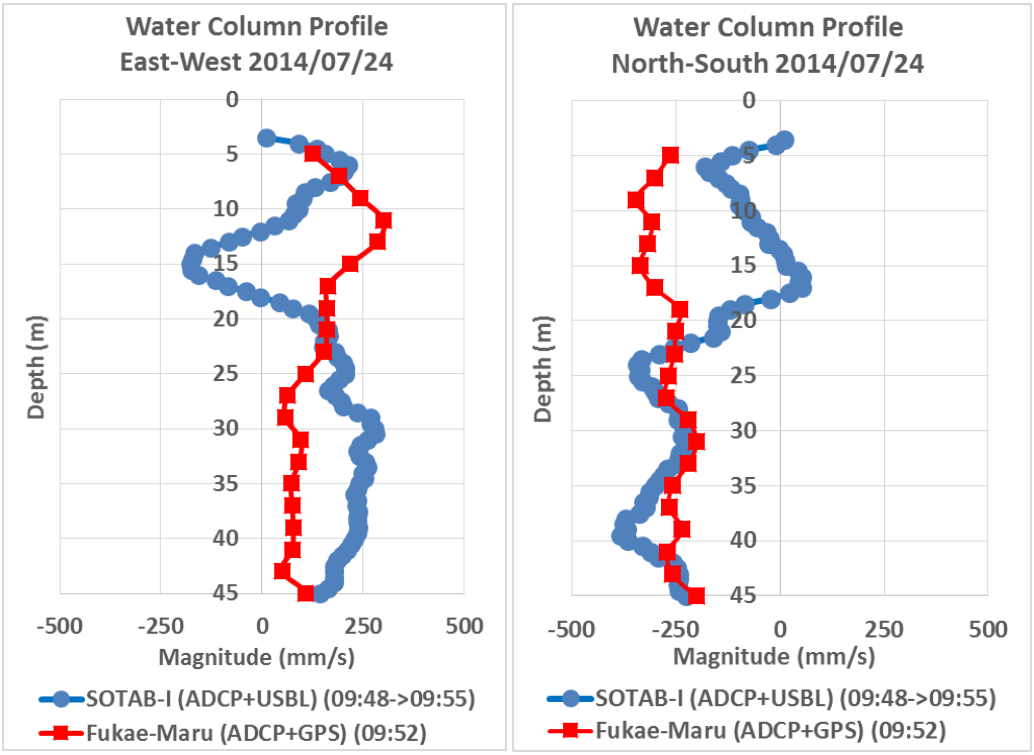



**Fig. 90** Comparison between SOTAB-I and Fukae-maru ADCPs in the middle zone

Fig. 90 shows the water current profile measured using robot's speed calculated based on the USBL position data. It shows that near sea surface down to 20m, there is a small correlation in the curve trend and the shifting is large. This may be explained by the fact that the robot is situated not enough close to the beam angle range of the USBL transceiver. When the robot is deeper that 20m, the trend of the Fukae-maru and SOTAB-

I water currents measurements become similar and the shifting value becomes much lower. So we can say that in the middle zone, the USBL data can be used help to estimate the magnitude and the direction of the water currents.

To overcome the shortcoming of the USBL system near the sea surface and the dead signal zone, we consider using AINS to measure the water currents profiles in these zones. The ADCP aiding method sequence starts from the initial water current profile established in Fig. 88 based on the GPS data. When SOTAB-I starts diving, the water currents at the same depth layers are measured again by the previous ADCP bins. Assuming that the average water currents along the water column are constant during the robot dive, the robot's velocity can be calculated, followed of by the determination of the absolute water current at the depth layer that coincides with the next bin. The method is explained in details in (Medagoda et al., 2011).

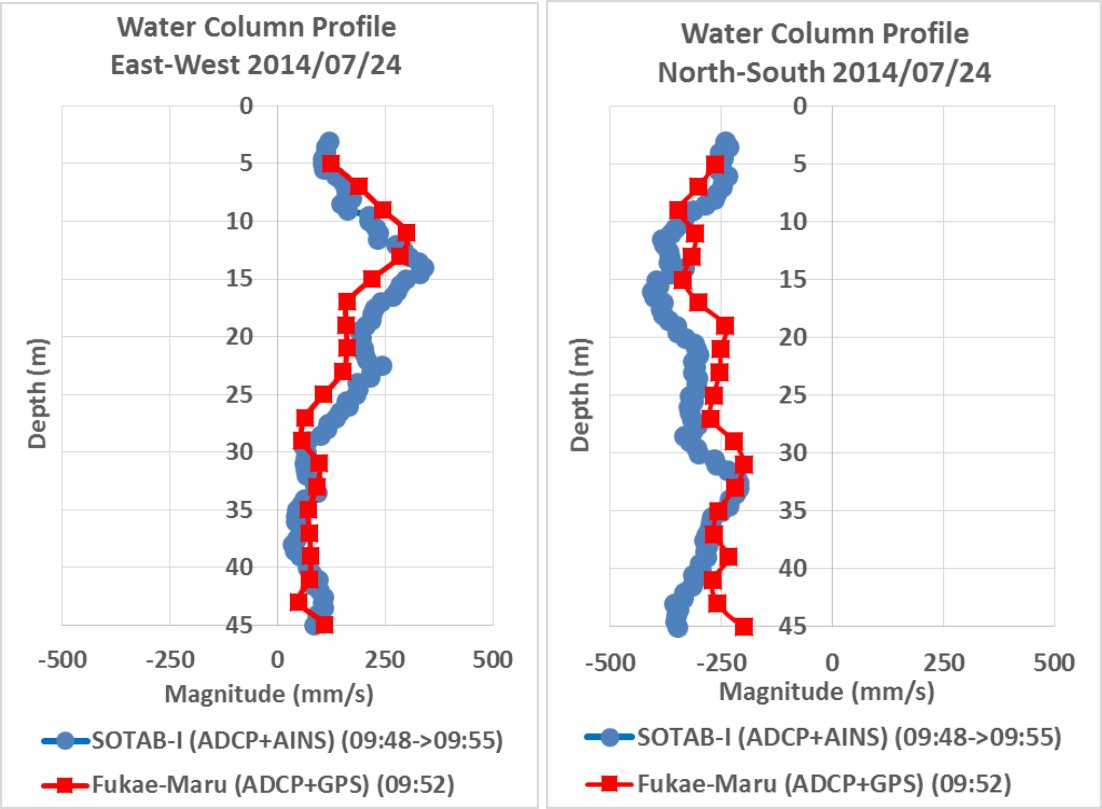



**Fig. 91** Comparison between SOTAB-I and Fukae-maru water currents profiles using AINS

Fig. 91 shows the result of water currents calculation using AINS. Both water currents profiles are very similar in terms of direction and magnitude. This proves that the method is reliable and can be used in the dead signal zone.

### 5.2.3 UMS

The UMS was calibrated for dissolved gases (methane, nitrogen, oxygen, argon, and carbon dioxide) by equilibrating acidified artificial seawater for more than one hour with gas mixtures that contained certified mole fractions of the gases. Salinity and temperature, measured during sample analysis, allowed calculation of dissolved gas concentrations. Gas volume percentages are shown in Table 33. The UMS was calibrated for ethane, propane, and butane by equilibrating seawater with gas mixtures that contained a certified

mole fraction of ethane, propane, or butane for two point calibrations of these gases (background and one concentration). The UMS was also calibrated for VOCs by analysis of VOC standards created by serial dilution of stock solutions of benzene, toluene, and xylenes. Calibration was not performed for hydrogen sulfide or naphthalene. Each sample was analyzed until a stable signal was achieved. Blank samples (i.e., UMS residual gas backgrounds) were measured by leaving deionized water in the MIMS assembly with the sample pump inactivated overnight to allow complete degassing of the sample in contact with the membrane. The UMS assembly temperature was controlled at 25°C during calibration to mimic deployment conditions. The UMS cast data were subsequently converted to concentrations for the dissolved gases (μmol/kg) and VOCs (ppb) from the calibration parameters and concurrently collected physical (CTD) data using algorithms and software developed by the Stanford Research Institute (SRI).

**Table 33** Standard gas mixtures used for equilibration (in volume %)

| **Gas** | **Mixture 1** | **Mixture 2** | **Mixture 3** | **Mixture 4** | **Mixture 5** | **Mixture 6** | **Mixture 7** |
|---|---|---|---|---|---|---|---|
| Methane | 0.0995 | 0.2500 | 2.5000 | 3.351 | | | |
| Nitrogen | Balance | Balance | Balance | Balance | Balance | Balance | Balance |
| Oxygen | 20.85 | 21.0000 | 17.0100 | 9.9600 | | | |
| Argon | 1.009 | 1.3010 | 1.0040 | 0.6990 | | | |
| Carbon Dioxide | 0.0990 | 0.7510 | 0.1500 | 0.0400 | | | |
| Ethane | | | | | 0.1000 | | |
| Propane | | | | | | 0.1000 | |
| Butane | | | | | | | 0.1000 |

Linear least squares regressions provided UMS calibration coefficients for methane, nitrogen, ethane, oxygen, propane, argon, carbon dioxide, and butane concentrations using measured UMS ion currents, at m/z of 15, 28, 30, 32, 39, 40, 44, and

58. The ion current at m/z 44 (called $I_{44}$), which is the mass spectrometer ion signal intensity for m/z 44 corresponding to the diagnostic ion for carbon dioxide, was also used in the nitrogen regression to account for contributions from carbon dioxide fragmentation.

Additionally, all signal intensities were background corrected by subtracting the signal intensity at m/z 5 (electronic background); this subtraction accounts for changes in electronic noise resulting from UMS temperature variability. The signal intensity at m/z 5 is used as the electronic background because there is no chemical that will give a peak in the mass spectrum at m/z 5. The "argon" or "water" correction is then used, as described in (Bell et al., 2007; Bell, 2009), to account for temperature variations in the field. The UMS calibration parameters and deployment parameters were identical. The calibration parameters that were identical were the sample flow rate and temperature of the membrane introduction heater block. A time delay was applied to the UMS cast data to adjust for the sample travel time through the tubing and membrane permeation.

The argon and water vertical profiles are the measured ion intensities at m/z 40 (argon) and m/z 18 (water vapor) as a function of depth. These are used to normalize the concentration profiles of the other analytes to account for changes in permeation through the membrane interface with increased pressure, as well as other changing environmental conditions that affect the signal intensities (Bell et al. 2007; Bell, 2009), therefore, high frequency noise in these data sets was removed using a Butterworth filter prior to normalization of the other profiles.

The typical measurement accuracy at best is 2%, but this varies for different chemicals. The response time is at best 5-10 s for the light compounds and worse for the high molecular weight compounds. A typically reasonable spatial resolution can be

obtained with an ascent and descent rate of 0.5 m/s. As mentioned in the robot maneuverability section, the maximum vertical and lateral speed of the SOTAB-I are below that rate.

### 5.3 Vertical Water Column Survey in the Gulf of Mexico

At-sea experiments were performed from the 6th to the 15th of December 2013 in the Gulf of Mexico in the U.S. (Fig. 92), near where the Deepwater Horizon oil spill accident in 2010 and the Hercules 265 oil rig blowout in 2013 occurred that led to the release of methane gas. The aim of the exploration was the evaluation of the performance of the SOTAB-I's surveying abilities. Due to the strong wind and severe weather conditions, experiments were carried out in shallow water and in particular, at the mouth of the Mississippi River, where the UMS data were measured. The area is characterized by its prevalent abandoned oil rigs and natural seepage of hydrocarbons (Mitchell et al., 1999).

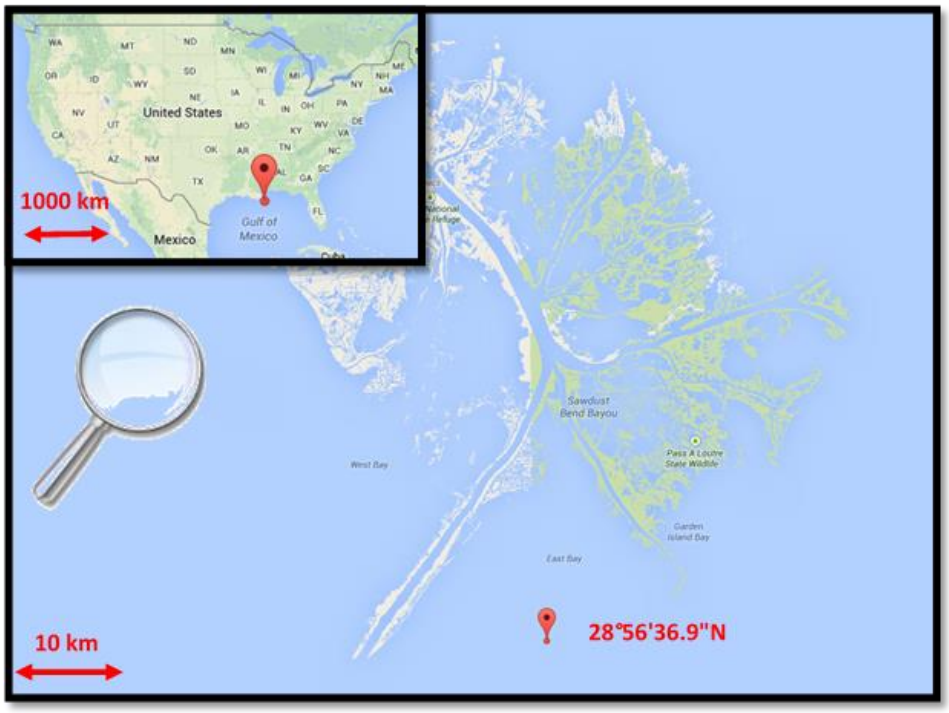



**Fig. 92** Gulf of Mexico experimental zone (Google Map)

This section is mainly focuses on the experimental results obtained on the 13th of December 2013 from 13:30 to 14:30 dive.

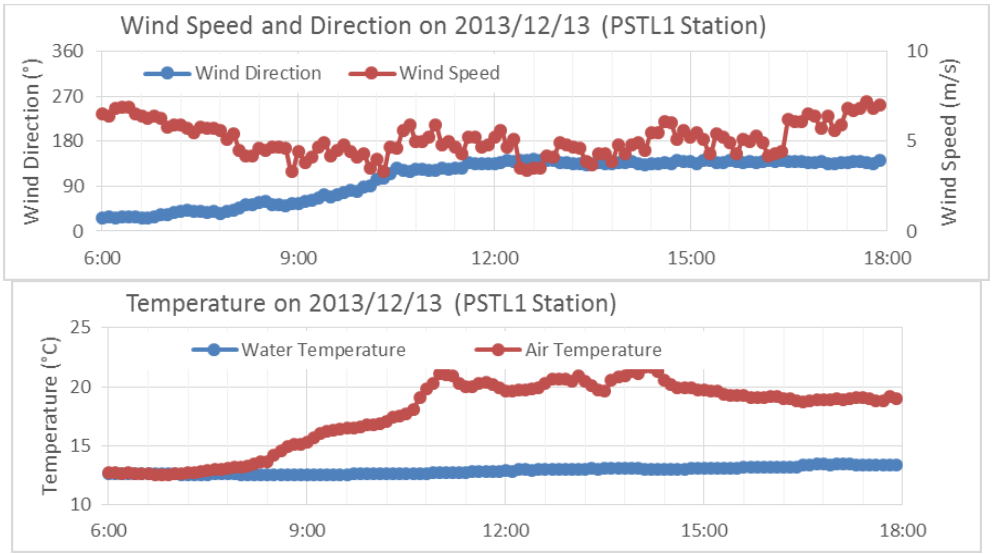


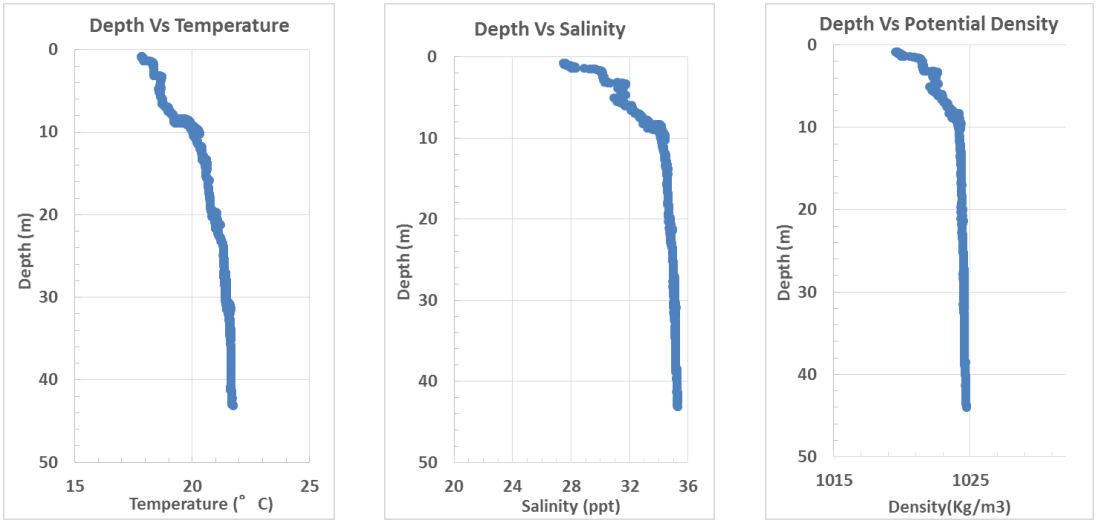

**Fig. 93** Temperature and wind speed reported by the PSTL1 station on 13th December 2013

Fig. 93 displays the meteorological data reported by the PSTL1 station on the 13th of December 2013. The station is located in the mouth of the Mississippi River at the position (28°55'56" N 89°24'25" W). The distance between the PSTL1 station and the place where the experiments were carried out on the 13th of December 2013 is around 20 km. The station reported a southeastern wind direction with a gradually decreasing speed from 5 m/s at the time of the experiment. Water temperature was 13°C, and atmospheric temperature was 20°C.

### 5.3.1 Temperature, Salinity, and Density

Fig. 94 displays the vertical distributions of temperature and salinity measured by the CTD. It can be observed that the sea temperature in the upper layer is colder than in the bottom layer (Fig. 94 (a)), which can be explained by the cold atmospheric

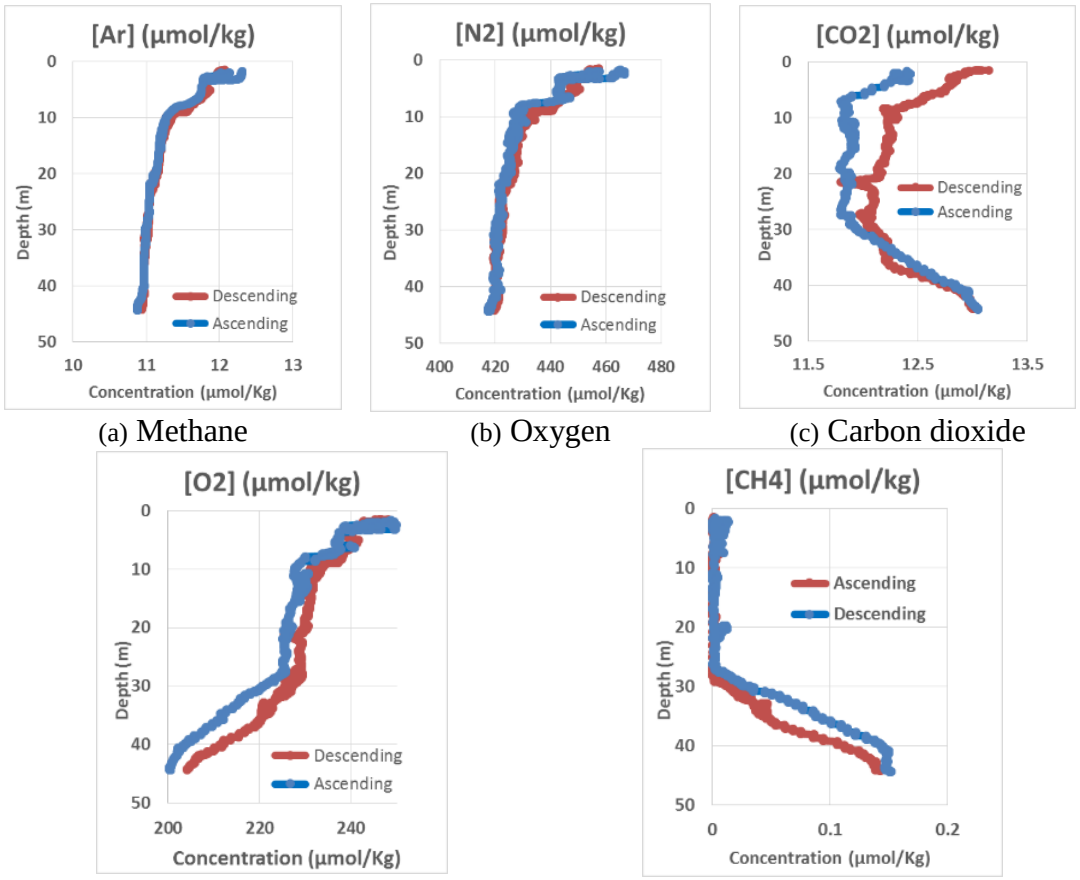
temperatures that dropped under 12°C on the day that preceded the experiments. The atmospheric temperature reported by the PSTL1 buoy database was between 13°C and 19°C at the time of the experiments on December 13th (Fig. 93). The salinity level near the sea-surface changed considerably from 28 psu at 2 m water depth to 34 psu at 8 m (Fig. 94 (b)),. This can be explained by the location of the site, which is in the middle of the Mississippi mouth where fresh water flows out to the Gulf of Mexico. The fresh water layer was breached through an adjustment of the buoyancy device. The density depends on temperature, salinity, and pressure. In Fig. 94 (c), it can be observed that the increase of density was primarily dominated by the large variation of salinity.




(a) Temperature (b) Salinity (c) Density

**Fig. 94** Vertical distributions of temperature, salinity and density

### 5.3.2 Dissolution of Substances




(a) Methane (b) Oxygen (c) Carbon dioxide

(d) Nitrogen (e) Argon

**Fig. 95** Dissolution of substances in the water column

Fig. 95 illustrates the change of concentration of some substances along the water column. Fig. 95 (a) and (b) show the respective vertical concentration profiles for nitrogen and argon needed for the calculation of the other substances' dissolution profiles mentioned previously. Fig. 95 (c) demonstrates that the concentration of methane in the upper water layers is negligible down to a depth of 30 m, and it starts to increase steadily down to a water depth of 44.6 m. In Fig. 95 (d), it can be observed that the oxygen concentration moderately decreased from a water depth of 0 m to that of 10 m, followed by a slower rate of decline from 10 m to 27 m water depth. Then, oxygen concentrations declined

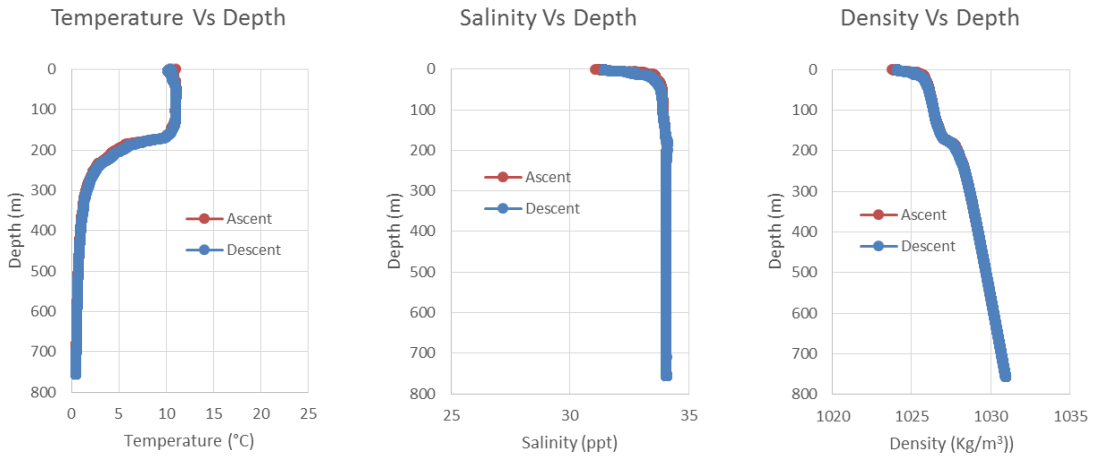
considerably from a water depth of 27 m to that of 44 m. It can be seen that the oxygen concentration decreased with increasing depth. In Fig. 95 (e), three zones can be distinguished based on the change in carbon dioxide concentrations: in water depths between 0 m and 10 m, carbon dioxide concentrations decreased gradually. From 10 m to 27 m, it kept decreasing, but at a slower rate. Below 30 m, carbon dioxide concentrations increased down to a water depth of 44 m.

From this perspective, we can say that the SOTAB-I succeeded in measuring dissolved substance variations along the vertical water column. Conversely, other alkanes and benzene-toluene-xylene (BTX) were below the sensory threshold and had no significant concentrations.

Very few methods will verify or corroborate the UMS measurements. We have used dissolved oxygen (DO) sensors in the past to compare the UMS oxygen measurements (m/z 32), and the comparison was generally very good (Bell et al., 2007; Bell, 2009). The SOTAB-1 deployments were not in a location where we would expect to see alkanes and BTX. We believe that the methane that we detected was biogenic methane and not associated with an oil reservoir. We have verified the UMS ability to detect these compounds in the lab and in other deployments (see Wenner et al., 2004 for BTX using an earlier version).

## 5.4 Vertical Water Column Survey in Toyama Bay on March 17th, 2016

### 5.4.1 Temperature, Salinity, Density and Sound Speed




(a) Temperature (b) Salinity (c) Density

**Fig. 96** Vertical distributions of temperature, salinity and density

Fig. 96 (a) shows the seawater temperature distribution along the water column. Three zones can be distinguished. The first, which is called "mixed layer," is very dependent on the atmospheric temperature. The seawater temperature on the sea surface was equal to 10.5°C. It varied between 10°C and 11°C on the first 30 m, then remained steady up to 125m where it started to decrease gradually up to 170m. In the second zone, called the thermocline zone, a sharp decrease of seawater temperature from 10° to 3° can be observed between 170m and 230m water depth. The last zone—the deep layer—is the coldest layer and features a slow rate temperature change. The temperature reached 0.45° at 764 m water depth. The salinity profile (Fig. 96 (b)) can also be divided into three distinct zones. In the surface zone, a sharp variation of salinity from 31 to 33.5 psu in the upper 20m layer can be observed. It is due to the precipitations that preceded the experiments and to the nature of the bay, which is surrounded by the Northern Japanese Alps characterized by high mountain ranges and multiple rivers, a source of fresh water.

The water layer between 20m and 170m is characterized by an increase, at a slowing rate, in the salinity level up to 34.1 psu at 170m. The second zone is the halocline zone. It is characterized by a slight gradual decrease in the salinity level up to 34.05 psu from 180 to 270 m depth. Finally, in the deep zone, a slight increase from 34.05 to 34.07 psu occurred between 270m to 480 m, followed by a steady salinity value up equal to 34.07 psu to 764m. For the density profile in Fig. 96 (c), it was calculated based on the pressure, temperature and salinity, the vertical column density distribution. It can be divided into two zones. The first zone is called the mixed layer, it is characterized by a sharp increase in the water density from the sea surface where it is equal to 1023.5 Kg/m3, to 1025.8 Kg/m3 x observed. Density measured at 764m water depth was equal to 1030.9 kg/m3. It can be noticed that the vertical water distributions of the temperature, the salinity, and the density are very similar in ascending and descending conditions. Slight differences are due to the temporal and spatial conditions in addition to the slight variance of the accuracy of the CTD sensor with its environmental temperature.

### 5.4.2 Vertical Profile of Water Currents

#### 5.4.2.1 Near the Sea Surface

The average water current in the zone near the sea surface can be calculated by combining the robot's GPS speed and the relative water currents measured by the ADCP. Following are the calculation steps:

1. Calculate the average robot drifting speed on the sea surface (Table 8)
2. Calculate the average of the relative water currents at each depth layer when the robot is on the sea surface in the same period used in step 1.

3. Calculate the absolute water currents near the sea surface using formula (9)

Since the ADCP sensor is mounted on the bottom of the robot, situated at 2m water depth, and that the blank distance of the ADCP sensor is equal to 1m, then the measurement of the water currents starts from 3m water depth.

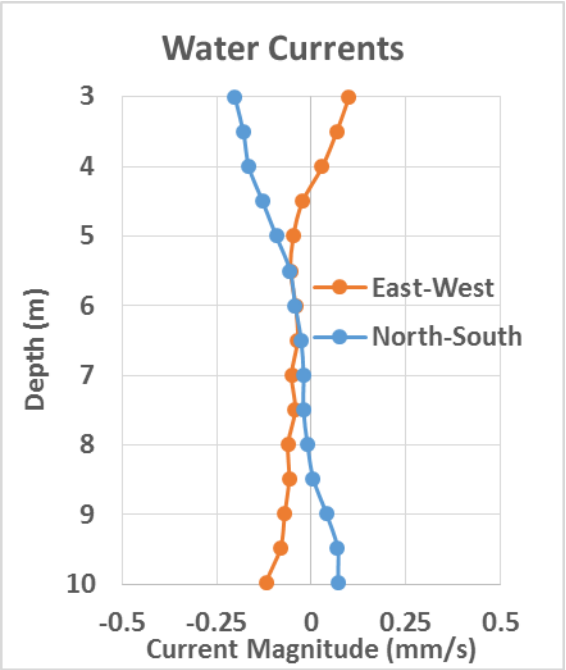



**Fig. 97** Water currents near the sea surface

Fig. 97 shows the water current near the sea surface. On the sea surface, the water current is flowing in the south-east direction which is in conformity with the robot motion direction.

### 5.4.2.2 In the Middle Zone

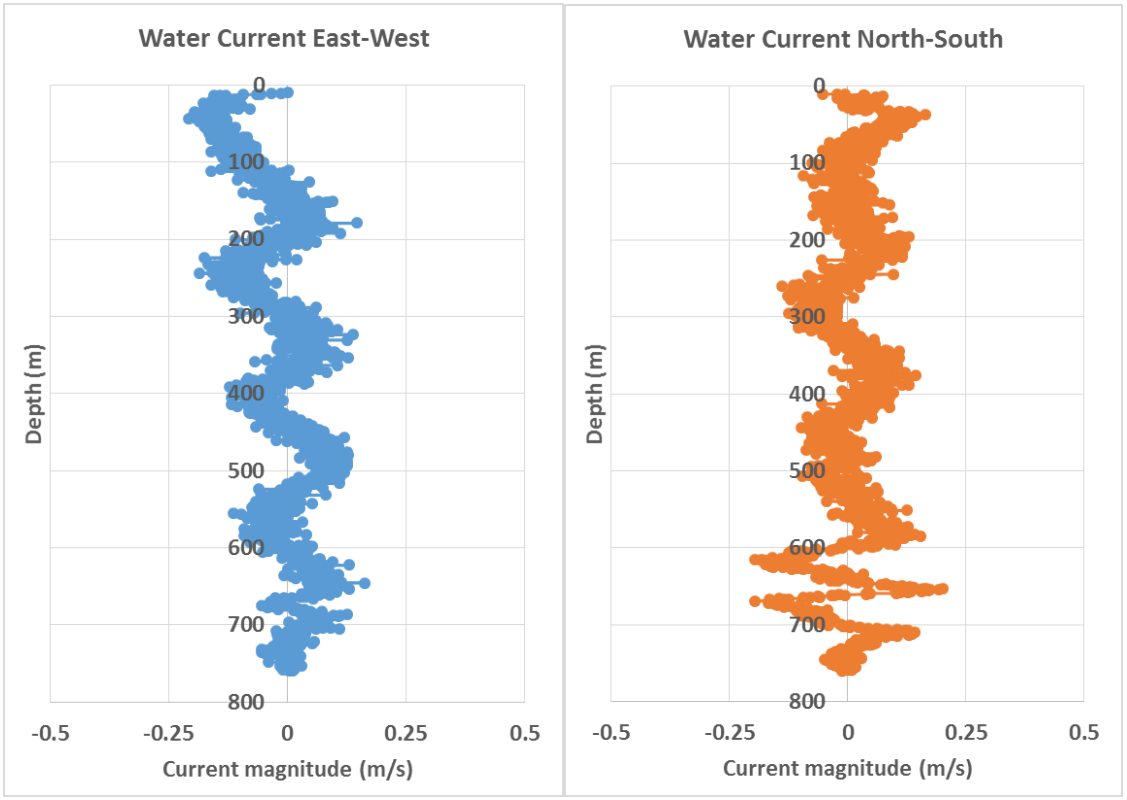



**Fig. 98** Water currents in the middle zone

Fig. 98 shows the water column currents distribution in the middle zone. The robot's speed used to calculate the absolute water current was obtained from the Kalman filter. The process used to calculate the final absolute water current was described in section 5.2.2.2

#### 5.4.2.3 Near the Seabed

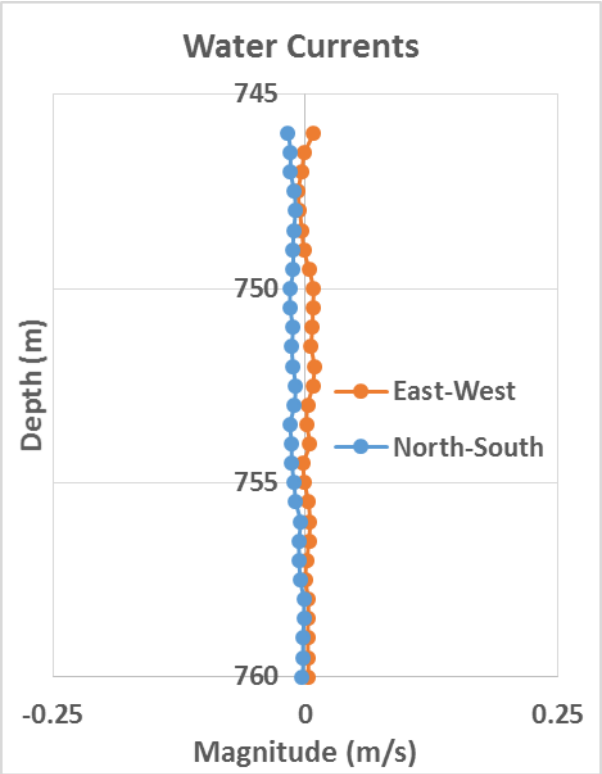



**Fig. 99** Water currents near the seabed

Fig. 99 shows the water current near the seabed. The water currents were determined based on the bottom tracking velocities of the robot. It can be observed that the water currents are very weak near the seabed.

## 5.5 Conclusions

SOTAB-I is being developed in order to perform on site measurements of oceanographic data as well as dissolved chemical substances using underwater mass spectrometry and those transmissions in real time with their corresponding position for a rapid inspection of the area where underwater releases of oil and gas first occurred. Collected data will help to comprehend the environmental changes due to accidents and boost the accuracy of simulation of behavior of oil and gas, which contributes to the efforts to avoid further

damage.

The oceanographic data calculation was explained and complemented by the development of software tools to automatize the repetitive work that comes with the processing of huge amounts of data surveyed. A calculation method of water current velocity using SOTAB-I was proposed and resulted in a smooth curve. The data of the water current profile are essential for the prediction of spilled oil behavior. At Komatsu-shima experiments, water currents measurement process was evaluated. Water current profile measured by the SOTAB-I showed a good agreement with the one measured by the Fukae-maru ADCP.

From the Gulf of Mexico experiments in the U.S., the surveying abilities of SOTAB-I in shallow water were demonstrated. The water column distributions of dissolution of substances as well as the temperature and salinity were measured. The UMS could measure the variation in the concentration of various substances simultaneously, such as methane, oxygen, and carbon dioxide. In these experiments, a vertical water column survey was conducted within 50 m water depth. In the Toyama Bay experiment, these efforts were continued to extend the range up to 700 m water depth.

# Chapter 6: Conclusions and Future Work



`

# 6 Conclusions and Future Work

## 6.1 Conclusions

To prevent further damage caused by oil spills and gas blowout accidents, a spilled oil and gas tracking autonomous buoy system (SOTAB-I) was developed. It has the advantages of being a compact system with an extended surveying range. The robot can perform on-site measurements of several chemical substances dis-solved simultaneously and is able to collect oceanographic data. The SOTAB-I can transmit data in real time with their corresponding position, making it very suitable for rapid inspection. Data collected will be processed by simulating and predicting programs that will help to explain the environmental changes due to the accident and boost the accuracy of oil drifting simulation. Consequently, establishing a better deployment strategy of collecting data becomes possible, and will contribute to the efforts to avoid further damage that can be caused by oil spill disasters.

SOTAB-I software organization and it is developed algorithms are presented. On the ship side, we designed and developed a GUI, which was a valuable tool to keep the user informed about the status of the robot and the changes around its environment, and to interact with the robot by controlling its actuators when necessary. On the robot side, the multilayered architecture of the SOTAB-I software enabled distribution of responsibilities and enhanced the clarity of the program and its flexibility. The program code optimization helped to reduce the processor usage. This contributed to the reduction in power consumption and prevented the processor from overheating inside the pressure cell. Software drivers were developed to assure the easy control of the actuators and the acquisition of sensors' data. Furthermore, we configured each sensor in a way to ensure

the fastest sampling period to enable to get the maximum amount possible of important data. To enhance the immunity of the acquired data against fluctuations, we implemented software filters, which had a positive impact in smoothing while keeping a good tradeoff between jitter and lag. To ensure the transmission of critical information to the user on the mother ship, we established an acoustic communication data frame. In addition, we implemented a checksum verification method to secure the control of the robot through downlink, giving it an immunity against noise.

Water column regions and robot operating modes were explained. A pyramidal hierarchy was established to define each control program priority. This architecture enabled the robot to avoid dangerous situations and interrupt the control program to automatically perform software emergency ascent when a problem is detected.

A new method for depth control using the buoyancy control device was developed. A model of the buoyancy variation with time was established. It was built based on the results obtained in high pressure tank experiment and several at-sea experiments. The depth control algorithm is based on the comparison between the time estimated for the robot to change its buoyancy from its current value to the neutral value, and the time expected for the robot to reach the target depth. The method was demonstrated at-sea experiments in Toyama Bay in Japan in March 2016. It showed the ability of the control algorithm to smoothly bring the robot to the target depth without a significant overshoot. The algorithm is characterized by its flexibility and doesn’t require a strict determination neutral buoyancy value. A margin of inaccuracy can be customized before performing the dive. The method could be further adapted to perform an altitude control through a progressive depth control algorithm based on 4 steps. The experiment results showed that

it worked properly.

In Suruga Bay experiments in Japan, the effect of wings of SOTAB-I was tested and showed the ability of the robot to move on the lateral plan with an acceptable ratio of the horizontal movement to the diving depth. From this point of view, we can say that the use of the rotational wings to move simultaneously in the vertical and the lateral planes was effective.

The collision avoidance concept was defined with its regions. The PID speed control succeeded to smoothly freeze the robot submergence to the dangerous zone.

The energy study enabled to estimate the robot’s battery's autonomy under different scenarios. The comparative study between the use of buoyancy device and thrusters proved the power efficiency of the depth control algorithm based on the control of the buoyancy device. The power efficiency of the depth stabilizer algorithm was studied and proved its ability to reduce the power consumption of the robot. The depth stabilization algorithm managed to reduce the power consumption by setting the actuator idle for almost half of the total period. This is an apparent advantage when compared to the PID and hysteresis controllers where the actuator is always active.

The oceanographic data calculation was explained. A calculation method of water current velocity using SOTAB-I was proposed and resulted in a smooth curve. The data of the water current profile are essential for the prediction of spilled oil behavior. In Komatsu-shima experiments, water currents measurement process was evaluated. Water current profile measured by the SOTAB-I showed a good agreement with the one measured by the Fukae-maru ADCP.

From the Gulf of Mexico experiments in the U.S., the surveying abilities of SOTAB-I in

shallow water were demonstrated. The water column distributions of dissolution of substances as well as the temperature and salinity were measured. The UMS could measure the variation in the concentration of various substances simultaneously, such as methane, oxygen, and carbon dioxide. In these experiments, a vertical water column survey was conducted within 50 m water depth. In the Toyama Bay experiment, these efforts were continued to extend the range up to 760 m water depth.

Even though the robot succeeded to realize the objectives for which it was developed, the robot presents some hardware and operational limitations.

- In the case where some oil penetrates to the robot, the performance of robot's sensors and actuators may be altered. In fact, the robot is equipped with a UMS based on membrane introduction technology. This means that if oil droplets may cause the sensor to stop working. This same problem may happen when oil droplets are introduced to the CTD pump. For that reason, the use of special filters may be required in such circumstances.
- The preparations needed to charge the robot's battery take time and effort because it requires to disassemble some parts of the robot to access to the pressure cell. It will be more helpful to simplify the process by using a water proof external plug cable to charge the robot.
- The ballast is activated using an acoustic signal. There is a possibility that the acoustic communication fails. For that reason, it will be safer to add another method to cut automatically the ballast system if a pre-set time of mission. It will

be also more convenient to add some leaking detection sensors inside the pressure cell to ascend the robot before the situation get worse.

- The camera used to take pictures of the seabed is not connected to the robot's computer. In addition, it is not possible to change to video mode. Interfacing the camera to the robot's PC will be very useful to collect more detailed information about the blow out gas area.
- The heavy weight of the robot requires the use of a crane to lift it, which increases the logistic resources needed for its deployment. In addition, a diver is needed to release and recover the robot from the crane one the robot is deployed. It will be more helpful to find a simpler method to deploy it without the need of the diver intervention.

## 6.2 Future Work

- Software development
    - Develop survey data transmission through Iridium satellite and robot control.
    - Improve the ship GUI to show the graphs of the dissolved gas concentration in real time.
- Navigation
    - Employ the Kalman filter in real time
    - The ADCP aiding was employed in the signal dead zone and proved its ability to improve the accuracy of the robot position estimation. Efforts

should be extended to apply it in the middle zone, which may help to enhance the robot navigation in that large zone.

  - The IMU sensor employed didn't reflect accurately the motion of the robot. A better calibration and different setting parameters need to be configured. Once done, the IMU data should be integrated in the Kalman data fusing algorithm and its impact on improving the robot's navigation should be studied.
  - The Kalman filter parameters should be further investigated to look for a better performance. Using adaptive parameters may contribute to the improvement of the robot navigation

- Guidance and control
  - Enhance the accuracy of the buoyancy variation model by feeding it with more experimental data.
  - Integrate the robot simulation model in the robot program to help to predict the robot behavior.
  - Improve the time efficiency of the predictive depth controller.
  - Integrate a neutral buoyancy calculator program that enables to determine the value of the neutral buoyancy taking in consideration the density measured by the CTD sensor. In this case the robot depth can be adjusted the same way as with the floats
  - Improve the power efficiency of the depth stabilizer.
  - Implement the power saving strategies established and tested in the simulation program.

- Water survey
    - Synchronization between SOTAB-I and SOTAB-II in field operation.
    - In these experiments, a vertical water column survey was conducted up to 760 m water depth. Efforts must be continued in order to extend the range not only in the vertical plane, but also to cover a cylindrical area with a diameter of 5 km. In the near future, in order to demonstrate the abilities of the SOTAB-I in deep water near methane gas seepage.
    - Evaluation of the data by deploying simultaneously a sample carousel and SOTAB-I. This will help to evaluate the CTD data and the water current measurements.
    - Study the effect of the robot attitude variation of the quality of the measured water currents.

## Special Acknowledgements

This research project was funded for 2011FY-2015FY by Grant-in-Aid for Scientific Research (S) of Japan Society for the Promotion of Science (No. 23226017).

# Appendix

**Table 34** Characteristics of one pack Lithium-Ion battery

| | |
|---|---|
| Ref | Paco KV-100 Li-Ion battery pack |
| Electric Charge | 10.0Ah |
| Voltage | 14.4V |
| Capacity | 144Wh |
| Weight | 1.4Kg |

**Table 35** Processing Unit Specification

| | |
|---|---|
| Ref | Advantech PCI-104, PCM-3363 |
| Processors | Intel Atom N455 Single Core, 1.66 GHz, Cache 512 KB<br>Intel Atom D525 Dual Core, 1.8 GHz, Cache 1 MB |
| Memory | 1GB DDR3 800 MHz |
| Interfaces | USB 2.0, RS-232, RS-422/485, SMBus (configurable to I2C), Ethernet, PS/2, GPIO |
| Power Supply Voltage | 5V |
| Power Consumption | N455: Typical 7.02 W / Maximum 11.825W<br>D525: Typical 9.25 W / Maximum 13.475W |

**Table 36** Thruster Specification

| | |
|---|---|
| Ref | Mitsui Thruster Model 260 |
| Voltage | 24V |
| Nominal Current | 12A |
| Thrust Force | 3.6 Kgf forward, 2.7Kgf Backward |
| Maximum Depth | 2000m |
| Weight | 2 Kg |

**Table 37** Main characteristics of USBL system of SOTAB-I

| | |
|---|---|
| Ref | LinkQuest Inc. TrackLink 5000HA |
| Accuracy | Positioning: 0.15 degree<br>Slant Range: 0.3 meter |
| Working Range | up to 5000 m |
| Operating Beam width | 120 degrees |
| Operating Frequency | 14.2 to 19.8 kHz |
| Maximum Depth | Transponder up to 7000 m<br>Transceiver: up to 20 m |
| Power Consumption | Transmit Mode: 40 Watts<br>Receiver Mode: 1 Watt |

**Table 38** Main characteristics of GPS Compass on the ship

| | |
|---|---|
| Ref | Hemisphere GPS Compass VS101 |
| Update rate Standard | 10 Hz; optional 20 Hz (position and heading) |
| Horizontal accuracy | < 0.6 m 95% confidence (DGPS1) |
| Heading accuracy | < 0.15º rms @ 1.0 m antenna separation |
| Pitch/roll accuracy | < 1º rms |
| Typical Acquisition times | Cold start < 60 s,Warm start < 20 s, Hot start < 1 s |
| Power consumption | ~ 5 W nominal |



`

**Table 39** Main characteristics of GPS receiver of SOTAB-I

| | |
|---|---|
| Ref | Garmin GPS 15L |
| Update rate | 1s to 900s |
| Acquisition times | Reacquisition: Less than 2 seconds<br>Warm: Approximately 15 seconds<br>Cold: Approximately 45 seconds |
| GPS accuracy: | Position: < 15 meters, 95% typical** |
| DGPS (WAAS) accuracy | Position: < 3 meters, 95% typical |

**Table 40** Main characteristics of the CTD sensor of SOTAB-I

| | |
|---|---|
| Ref | CTD Sensor SBE-49 FastCAT |
| Constructor | Sea-bird Electronics |
| Sampling Rate | 16 samples/second |
| Range | Temperature: -5 to +35 °C<br>Conductivity: 0 to 9 S/m<br>Pressure: 0 to 7000 meters |
| Resolution | Temperature: 0.0001 °C<br>Conductivity: 0.00005 S/m in oceanic waters<br>Pressure: 0.002% of full scale range |
| Weight | In air 2.7 kg, In water 1.4 kg |

**Table 41** Main characteristics of the ADCP/DVL of SOTAB-I

| | |
|---|---|
| Ref | Navigator DVL WHN1200 |
| Constructor | Teledyne RD Instruments |
| Operating frequency | 1200kHz |
| Maximum depth | 3000m |
| Built-in sensors | water temperature gauge, inclinometer, compass |
| Beam angle | 30 degrees |
| Water track velocity (ADCP) | Layer size selectable from 0.25m to 5m<br>Number of layers is selectable from 1 to 128 |
| Bottom track velocity (DVL) | Altitude from 4 individual measurements<br>Minimum altitude 0.5m<br>Maximum detectable altitude 30m<br>Velocity Range: -/+10m/s<br>Long Term Accuracy: ±0.2% ±0.1cm/s |

**Table 42** UMS Specifications.

| | |
|---|---|
| Ref | SRI International |
| Mass Analyzer Type | Linear Quadrupole Mass Filter |
| Mass Range | 1-200 amu |
| Inlet System | Membrane Introduction (PDMS) |
| Power Consumption | 60 - 80 Watts |
| Operation Voltage | 24 VDC |
| Maximum Deployment Time | 10 -14 Days (exhaust limited) |
| Dimensions | Diameter 24 cm, Length 64 cm |
| Weight | In air: 35 kg, In water: 5 kg neg. |
| Depth Capability | 2000 m |

**Table 43** Compass Specifications.

| | |
|---|---|
| Ref | PNI's FieldForce TCM-XB |
| Accuracy | Heading 0.3º RMS<br>Pitch 0.2º RMS<br>Roll 0.2º RMS |
| Resolution | Heading <0.1º RMS<br>Tilt <0.01º |
| Range | Heading: 0-360º<br>Pitch: ±90º<br>Roll: ±180º |
| Maximum Sample Rate | 30 samples/sec |
| Supply Voltage | 3.6 – 5 V |
| Calibration | Hard and Soft Iron |

**Table 44** IMU specification

| **Attitude & Heading** | |
|---|---|
| Range | Heading, Roll:±180°<br>Pitch : ±90° |
| Static Accuracy | Heading: 2.0° RMS<br>Pitch/Roll:0.5° RMS |
| Angular Resolution | < 0.05° |
| Output Rate | 400 Hz |
| **Gyro Specifications** | |
| Range: | ±2000 °/s |
| In-Run Bias Stability: | < 10 °/hr |
| Linearity: | < 0.1% FS |
| Noise Density: | 0.0035 °/s √Hz |
| Bandwidth: | 256 Hz |
| Alignment Error: | ±0.05° |
| Resolution | < 0.02 °/s |
| **Accelerometer Specifications** | |
| Range: | ±16 g |
| In-Run Bias Stability | < 0.04 mg |
| Linearity: | < 0.5 ° FS |
| Noise Density: | < 0.14 mg/√Hz |
| Bandwidth: | 260 Hz |
| Alignment Error: | ±0.05 ° |
| Resolution | < 0.5 mg |
| **Pressure Sensor Specifications** | |
| Range | 10 to 1200 mbar |
| Resolution | 0.042 mbar |
| Accuracy | ±1.5 mbar |
| **Electrical** | |
| Input Voltage: | 4.5 V to 5.5 V |
| Max Power Consumption: | 220 mW |
| Digital Interface: | Serial TTL, RS-232 |

**Table 45** Robot's GPS data format

$GPGGA,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,M,<10>,M,<11>,<12>*hh<CR><LF>

| | |
|---|---|
| <1> | UTC time of position fix, hhmmss format |
| <2> | Latitude, ddmm.mmmm format (leading zeros are transmitted) |
| <3> | Latitude hemisphere, N or S |
| <4> | Longitude, dddmm.mmmm format (leading zeros are transmitted) |
| <5> | Longitude hemisphere, E or W |
| <6> | GPS quality indication, 0 = fix not available, 1 = Non-differential GPS fix available, 2 = differential GPS fix available, 6 = estimated (only if NMEA 0183 version 2.30 mode is enabled in field 7 of the PGRMC1 sentence |
| <7> | Number of satellites in use, 00 to 12 (leading zeros are transmitted) |
| <8> | Horizontal dilution of precision, 0.5 to 99.9 |
| <9> | Antenna height above/below mean sea level, -9999.9 to 99999.9 meters |
| <10> | Geoidal height, -999.9 to 9999.9 meters |
| <11> | Differential GPS (RTCM SC-104) data age, number of seconds since last valid RTCM transmission (null if not an RTCM DGPS fix) |
| <12> | Differential Reference Station ID, 0000 to 1023 (leading zeros are transmitted, null if not an RTCM DGPS fix) |

**Table 46** IMU configuration register

| Offset | Name | Format | Unit | Description |
|---|---|---|---|---|
| 0 | AsyncMode | uint16 | - | Selects whether the output message should be sent out on the serial port(s) at a fixed rate.<br>0 = None. User message is not automatically sent out either serial port.<br>1 = Message is sent out serial port 1 at a fixed rate.<br>2 = Message is sent out serial port 2 at a fixed rate.<br>3 = Message is sent out both serial ports at a fixed rate. |
| 2 | RateDivisor | uint16 | - | Sets the fixed rate at which the message is sent out the selected serial port(s). The number given is a divisor of the ***ImuRate*** which is nominally 800Hz. For example to have the sensor output at 50Hz you would set the Divisor equal to 16. |
| 4 | OutputGroup | uint16 | - | Selects which output groups are active in the message. The number of OutputFields in this message should equal the number of active bits in the OutputGroup. |
| 6 | OutputField(1) | uint16 | - | Active output fields for the first active group. |
| 4+2*N | OutputField(N) | uint16 | - | Active output fields for the Nth active group. |

**Table 47** IMU Binary Output Data Fields

| Bit Offset | Group 1 Common | Group 3 IMU | Group 5 Attitude |
|---|---|---|---|
| 0 | TimeStartup | ImuStatus | VpeStatus |
| 1 | Reserved | UncompMag | YawPitchRoll |
| 2 | TimeSyncIn | UncompAccel | Quaternion |
| 3 | YawPitchRoll | UncompGyro | DCM |
| 4 | Quaternion | Temp | MagNed |
| 5 | AngularRate | Pres | AccelNed |
| 6 | Reserved | DeltaTheta | LinearAccelBody |
| 7 | Reserved | DeltaVel | LinearAccelNed |
| 8 | Accel | Mag | YprU |
| 9 | Imu | Accel | |
| 10 | MagPres | Gyro | |
| 11 | DeltaTheta | SensSat | |
| 12 | VpeStatus | Raw | |
| 13 | SyncInCnt | | |
| 14 | | | |
| 15 | | | |

**Table 48** IMU binary output message configuration

| **Configuration Line Format** | | "$①, ②, ③, ④, ⑤, ⑥, ⑦, ⑧*XX\r\n" | |
|---|---|---|---|
| **Configuration Command** | | "$VNWRG,75,1,16,15,1,0FFF,01F7*XX\r\n" | |
| **Configuration Explanation** | | | |
| **N°** | **Cfg** | **Definition** | **Role** |
| ① | VNWRG | Register Command | Write Register |
| ② | 75 | Register ID | Binary Message Configuration Register |
| ③ | 1 | Serial Port Cfg. | Output data on Serial Port 1 |
| ④ | 16 | Divisor Cfg. | Set sampling frequency to 50Hz |
| ⑤ | 15 | Output Groups Cfg. | Output Group 1 + Group 3 + Group 5 |
| ⑥ | 1 | Group 1 Output Fields Cfg. | Group 1: Output Time Start Up |
| ⑦ | 0FFF | Group 3 Output Fields Cfg. | Group 3: Output All data |
| ⑧ | 01F7 | Group 5 Output Fields Cfg. | Group 5: Output All data except DCM |

**Table 49** CTD data format

*Example*: FastCAT with **OutputSal=N**, **OutputSV=N**
example scan = ttt.tttt, cc.ccccc, pppp.ppp = 23.7658, 0.00019, 0.062

- Temperature = ttt.tttt = 23.7658;
  temperature (°C, ITS-90) = 23.7658
- Conductivity = cc.ccccc = 0.00019;
  conductivity (S/m) = 0.00019
- Pressure = pppp.ppp = 0.062;
  pressure (decibars) = 0.062

**Table 50** ADCP/DVL PD0 Format

| | |
|---|---|
| ALWAYS OUTPUT | **HEADER**<br>(6 BYTES + [2 x No. OF DATA TYPES]) |
| | **FIXED LEADER DATA**<br>(59 BYTES) |
| | **VARIABLE LEADER DATA**<br>(65 BYTES) |
| **WATER PROFILING DATA**<br>WD-command<br>WP-command | **VELOCITY**<br>(2 BYTES + 8 BYTES PER DEPTH CELL) |
| | **CORRELATION MAGNITUDE**<br>(2 BYTES + 4 BYTES PER DEPTH CELL) |
| | **ECHO INTENSITY**<br>(2 BYTES + 4 BYTES PER DEPTH CELL) |
| | **PERCENT GOOD**<br>(2 BYTES + 4 BYTES PER DEPTH CELL) |
| BP-command | **BOTTOM TRACK DATA**<br>(85 BYTES) |
| ALWAYS OUTPUT | **RESERVED**<br>(2 BYTES) |
| | **CHECKSUM**<br>(2 BYTES) |

**Table 51** UMS data format

*$*AAA*UD*,YY/MM/DD<space>hh:mm:ss,<space>*pwr*<space>BB.B<space>C.C<space>DD.D<space>E.E,<space>*rp*<space>FFF,<space>*tp*<space>HH<space>JJJJ<space>KKK,<space>*ms*<space>L.LL*E*MMM<space>NNN<space>PPP,<space>*sh*<space>QQ.QQ<space>RR,<space>*vh*<space>000<space>00,<space>*sp*<space>000,<space>*th*<space>00.0,<space>*ld*<space>V,<space>*sv*<space>0,<space>*uC*<space>XX.X<space>YYY,<space>*store*<space>ZZZ.Z,<4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B><4B>*END*<CR> <LF>

Where

- A packet is sent out to the robot whenever a transpector scan is received (which is approximately once every 7 sec)
- items in blue color are data item delimiters
- Data items are ASCII formatted unless mentioned otherwise
- All data items are fixed format (fixed number of bytes per item) and zero-padded (e.g. if a data item is 3 bytes = XXX and a value is 21 then UMS will send it as '0''2''1')
- <space> = ASCII code 0x20, <CR> =0x0D, <LF> = 0x0A

0 – indicate currently unused items, they will be always reported as zero values

| | | |
|---|---|---|
| $ | AAA | 3-Byte ID for the UMS3000 system |
| UD | MM/DD/YY | Date Month/Day/Year |
| | hh:mm:ss | Local time Hour/Minute/Second |
| pwr | BB.B | Primary Input Voltage (V) |
| | C.C | Primary Current Consumption (A) |
| | DD.D | Secondary Input Voltage (V) |
| | E.E | Secondary Current Consumption (A) |
| rp | FFF | roughing pump speed in rps |
| tp | HH | Turbo pump bearing temperature– (deg C), |
| | JJJJ | Turbo pump current (mA), |
| | KKK | Turbo Pump speed (% of max speed = 90000 rpm) |

| | | |
|---|---|---|
| ms | LLEMMM | Transpector total pressure (in torr) as a floating point number (e.g. 12E-34), |
| | NNN | Hardware function byte in ASCII format (range is 000 to 256), |
| | PPP | CCC bytes in ASCII format (range is 000 to 256) |
| sh | QQ.QQ | Sample heater current temperature (deg C), |
| | RR | Duty cycle of the heater (%) |
| ld | V | If equals 0 – no leak detected, if 1 – water leak is detected in the system |
| uc | XX.X | System board (PCB) temperature (deg C), |
| | YYY | Relative humidity (%) |
| store | ZZZ.Z | Amount of total storage available used (%) |
| | <4B> | represents a scan value in Amps for a particular mass, which is the binary representation of a floating point number, i. e. 4 bytes, for example – first four bytes is for 2amu, second is -4amu, third is 5amu etc |

## References


Azuma A, Nasu, K. -I. (1977) The Flight Dynamics of an Ocean Space Surveying Vehicle. Institute of Space and Science, Univ. of Tokyo, Report No. 547, Vol. 42, No. 2, p 41-90

Barker, L. (2014). Closed-loop buoyancy control for a Coastal Profiling Float. MBARI Intern Report.

Bell, R. J (2009) Development and deployment of an underwater mass spectrometer for quantitative measurements of dissolved gases. Ph.D. Thesis, University of South Florida,St. Petersburg, Florida

Bell, R. J et al (2007) Calibration of an in situ membrane inlet mass spectrometer for measurements of dissolved gases and volatile organics in seawater environ, Sci. Technol, 41. 8123–8128

Bond, C. E. (1996). Swim bladder, Biology of Fishes (2nd edition), Orlando: Saunders College Publishing, 1996. C. E. Bond, "Swim bladder", Biology of Fishes (2nd edition), Orlando: Saunders College Publishing

BP (2015) BP Statistical Review of World Energy 2015. (2015). Retrieved December 9, 2015, from http://www.bp.com/content/dam/bp/pdf/energy-economics/statistical-review-2015/bp-statistical-review-of-world-energy-2015-full-report.pdf

Casiez, G., Roussel, N., Vogel, D. (2012) 1 Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems. CHI'12, the 30th Conference on Human Factors in Computing Systems, May 2012, Austin, United States. ACM, pp.2527-2530, 2012

Clarke, M. R. (1978). Physical Properties of Spermaceti Oil in the Sperm Whale. Journal of the Marine Biological Association of the United Kingdom J. Mar. Biol. Ass., 58(01), 19. doi:10.1017/s0025315400024383

Eriksen, CC et al (2001) Seaglider: a long-range autonomous underwater vehicle for oceanographic research. IEEE Journal of Oceanic Engineering 26(4)

Fofonoff, N, Millard, R (1983) Algorithms for computation of fundamental properties of seawater, Unesco technical papers in marine science, #44, p 1-53.

Handa, Y. P (1990) Effect of hydrostatic pressure and salinity on the stability of gas hydrates, J Phys Chem, 94: 2652-2657

Hallegraeff, G. (1998). Transport of toxic dinoflagellates via ships' ballast water: bio-economic risk assessment and efficacy of possible ballast water management strategies. Marine Ecology Progress Series Mar. Ecol. Prog. Ser., 168, 297-309. doi:10.3354/meps168297

Harvey, J et al (2012) AUVs for ecological studies of marine plankton communities, Sea Technology, Vol. 53 (9), p 51

Hess, J. L, Smith, A.M.O (1964) Calculation of Non-lifting Potential Flow about Three Dimensional Bodies. Journal of Ship Research, Vol. 8, No. 2, p 22-44

ITOPF (2015) Oil Tanker Spill Statistics 2014. http://www.itopf.com/knowledge-resources/data-statistics/statistics/. Sep 2015

Izawa, K., Mizuno, K., Miyazaki, M., Inoue, A., Ando, K., Takatsuki, Y., Kobayashi, T., Takeuchi, K. (2002) On the weight adjustment of profiling floats. ARGO Technical Report FY2001, pp.18 - 35

Jakuba, M. V et al (2011) Toward automatic classification of chemical sensor data from autonomous underwater vehicles, Intelligent Robots and Systems, IEEE/RSJ International Conference on Intelligent Robots and Systems, p 4722-4727

Johansen, Ø et al (2003) DeepSpill–Field study of a simulated oil and gas blowout in deep water, Spill Science & Technology Bulletin, Vol. 8( 5–6) p 433–443

Joye, S. B et al (2011) Magnitude and oxidation potential of hydrocarbon gases released from the BP oil well blowout, Nature Geoscience 4, p 160–164

Kalman, R. E. (1960) A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering J. Basic Engineering, 82(1), 35. doi:10.1115/1.3662552

Kato, N., Senga, H., Suzuki, H., Okano, Y., Takagi, Y., Yoshie, M., Tanaka, T., Kobayashi, E., Chiba, H. (2015). "Autonomous Spilled Oil and Gas Tracking Buoy System and Application to Marine Disaster Prevention System," Interspill 2015. Amsterdam, Netherland 24-26 March, 2015

Kessler, J. D. et al (2011) A persistent oxygen anomaly reveals the fate of spilled methane in the deep gulf of mexico, Science 21, Vol. 331 No. 6015 p 312-315.

Lee, E.-J., Yoo, G.-Y., Jeong, Y., Kim, K.-U., Park, J.-H., and Oh, N.-H. (2015) Comparison of UV–VIS and FDOM sensors for in situ monitoring of stream DOC concentrations, Biogeosciences, 12, 3109-3118, doi:10.5194/bg-12-3109-2015, 2015.

Maxino, T. C, Koopman, P. J (2009) The effectiveness of checksums for embedded control networks, IEEE Transactions on dependable and secure computing, VOL. 6, NO. 1

Mcfarland, D., Gilhespy, I., & Honary, E. (2003) DIVEBOT: A diving robot with a whale-like buoyancy mechanism. Robotica, 21(4), 385-398. doi:10.1017/s026357470300496x

Medagoda, L., Williams, S. B., Pizarro, O., & Jakuba, M. V (2011). Water column current

profile aided localisation combined with view-based SLAM for Autonomous Underwater Vehicle navigation. 2011 IEEE International Conference on Robotics and Automation. doi:10.1109/icra.2011.5980141

Mitchell, R et al (1999) Estimates of total hydrocarbon seepage into the Gulf of Mexico based on satellite remote sensing images, EOS Supplement 80, OS242

Ober, HK (2010) Effects of Oil Spills on Marine and Coastal Wildlife. In: EDIS New Publications RSS. http://edis.ifas.ufl.edu/uw330. Sep 2015

Um, T. I., Chen, Z., and Bart-Smith, H. (2011) A novel electroactive polymer buoyancy control device for bio-inspired underwater vehicles. 2011 IEEE International Conference on Robotics and Automation, 172-177. doi:10.1109/icra.2011.5980181

Roemmich, D et al (2009) The argo program observing the global ocean with profiling floats, Oceanography 22(2):34–43, doi:10.5670/oceanog.2009.36

Senga, H., Kato, N., Yu, L., Yoshie, M., and Tanaka, T. (2012) "Verification Experiments of Sail Control Effects on Tracking Oil Spill," OCEANS 2012, Yeosu, IEEE, May 21-24, Proc. (CD-ROM).

Servio, P, Englezons, P (2002) Measurement of dissolved methane in water in equilibrium with its hydrate, J Chem Engineer Data, 47:87-90

Shaffer, G et al (2009) Long-term ocean oxygen depletion in response to carbon dioxide emissions from fossil fuels, Nature Geoscience 2, p105–109

Shibuya, K., Kishimoto, Y. and Yoshi, S. (2013) Depth Control of Underwater Robot with Metal Bellows Mechanism for Buoyancy Control Device Utilizing Phase Transition. Journal of Robotics and Mechatronics JRM, 25(5), 795-803. doi:10.20965/jrm.2013.p0795

Short, R. T et al (2006) Detection and quantification of chemical plumes using a portable underwater membrane introduction mass spectrometer, Trends in Anal. Chem. 25 (7) p 637–646

Solomon, E. A et al (2009) Considerable methane fluxes to the atmosphere from hydrocarbon seeps in the Gulf of Mexico, Nature Geoscience 2, p561–565

Sugii, M., Hayashi, M., Yano, Y., and Wakabayashi, N. (2014). Outline of an ADCP newly mounted on Fukae-maru and results of trial on the sea, Review of Graduate School of Maritime Sciences, Kobe University, Vol. 11, p 29-38

Takagi, Y., Ban, T., Okano, Y., Kunikane, S., Kawahara, S., Kato, N., and Ohgaki, K. (2012). "Numerical tracking of methane gas/hydrate and oil droplet in deep water spill," Proc. of InterAcademia 2012.

Teledyne RD Instruments, (2006) Acoustic Doppler Current Profiler principles of

operation -- A practical primer: P/N 951-6069-00, 59 p.

Tsutsukawa, S., Suzuki, Y., and Kato, N. (2012) “Efficacy Evaluation of Data Assimilation for Simulation Method of Spilled Oil Drifting,” Proc. of 5th PAAMES and AMEC2012, Taiwan, Paper No. SEPAS-05

Vogel, M et al (2001) Real-Time deepwater current profiling system, Proc.OCEANS 2001(MTS/IEEE), Vol.1, p269 – 274

Vickery, K (1998) Acoustic positioning systems. A practical overview of current systems, In Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles. Fort Lauderdale, FL, USA, p 5–17

Wenner, P. G et al (2004) Environmental chemical mapping using an underwater mass spectrometer, TrAC, Trends Anal. Chem. 23 p 288-295. doi:10.1016/S0165-9936(04)00404-2

Woodgate, R, Holroyd, A. (2011). “Correction of Teledyne Acoustic Doppler Current Profiler (ADCP) Bottom-Track Range Measurements for Instrument Pitch and Roll,” S, eprint arXiv

Yang, D. H, Xu, W. Y. `(2007) Effects of salinity on methane gas hydrate system, Sci China Ser D-Earth Sci. Springer Vol. 50 No. 11. p 1733-1745

Zhang, F., & Dickman, M. (1999). Mid-ocean exchange of container vessel ballast water. 1:Seasonal factors affecting the transport of harmful diatoms and dinoflagellates. Marine Ecology Progress Series Mar. Ecol. Prog. Ser., 176, 243-251. doi:10.3354/meps176243



`

# Published papers

## Journal Papers

Choyekh, M., Kato, N., Short, T., Ukita, M., Yamaguchi, Y., Senga, H., Yoshie, M., Tanaka, T., Kobayashi, E., Chiba, H. (2014). "Vertical Water Column Survey in the Gulf of Mexico Using Autonomous Underwater Vehicle SOTAB-I," Marine Technology Society Journal, JFOE & JASNAOE, pp. 88-101.

## Reviewed Conference Papers

Choyekh, M., Kato, N., Short, M., Yamaguchi, Y., Dewantara, R., Senga, H., Yoshie, M., Tanaka, T., Kobayashi, E., Chiba, H. (2016). "Depth Control of AUV using a Buoyancy Control Device," ICSDRI2016, Jeju, Korea

Choyekh, M., Ukita, M., Yamaguchi, Y., Kato, N., Senga, Y., Yoshie, M., Tanaka, T. and Kobayashi, E., (2015). "Vertical Water Column Survey of Komatsu-shima in Japan Using Autonomous Underwater Vehicle SOTAB-I," Proceedings of the Twenty-fifth International Ocean and Polar Engineering Conference, Hawaii, U.S.A, ISOPE, Volume 2 pp.637-644.

Choyekh, M., Ukita, M., Kimura, R., Kato, N., Senga, H, Yoshie, M., Tanaka, T. and Sakagami, N., (2014). "Structure of Control System of Spilled Oil and Gas Tracking Autonomous Underwater Vehicle SOTAB-I," Proceedings of the Twenty-fourth International Ocean and Polar Engineering Conference, Busan, South Korea, ISOPE, pp.624-631.

Choyekh, M., Kimura, R., Akamatsu, T., Kato, N., Senga, H, Suzuki, H., Okano, Y., Ban, T., Takagi, Y., Yoshie, M., Tanaka, T. and Sakagami, N. (2013). "Development of Spilled Oil and Gas Tracking and Monitoring Autonomous Buoy System and its Application to Marine Disaster Prevention," International Society of Offshore and Polar Engineers, Anchorage, U.S.A, ISOPE, Vol.1, pp.695-702.

## Other Conference Papers

Choyekh, M., Kato, N., Short, M., Yamaguchi, Y., Dewantara, R., Yoshie, M., Tanaka, T., Kobayashi, E., Chiba, H. (2015). "Early detection and monitoring of large scale hazardous substances spill from offshore platforms using autonomous underwater vehicle," The 23rd Ocean Engineering Symposium (OES2014), Tokyo, Japan, August 6-7, pp. 237-244

Choyekh, M., Ukita, M., Kimura, R., Yamaguchi, Y., Kato, N., Senga, H., Yoshie, M., Tanaka, T., Kobayashi, E., Chiba, H. (2014). "Sea Deployment of Spilled Oil and Blowout Gas Tracking Autonomous Buoy SOTAB-I in Suruga Bay and Gulf of

Mexico," JASNAOE2014, Tokyo, Japan

Choyekh, M., Ukita, M., Yamaguchi, Y., Kimura, R., Kato, N., Senga, H., Yoshie, M., Tanaka, T., Short, T., (2014). "Water Column Survey using a Spilled Oil and Gas Tracking Autonomous Underwater Vehicle SOTAB-I in Gulf of Mexico," The 32nd Robots Society Japan Symposium (RSJ2014), Nagasaki, Japan

## Co-Authored Papers

**Reviewed Conference Papers**

Yamaguchi, Y., Ukita, M., Choyekh, M., Kato, N., Senga, H., Yoshie, M. Tanaka, T., Kobayasi, E., Chiba, H. (2015). "Estimation of Water Current profile in Deepwater using Autonomous Underwater Vehicle SOTAB-I," The 25th International Offshore and Polar Engineering Conference (ISOPE2015), 21st-26th June, Kona, Hawaii, USA

Kimura, R., Choyekh, M., Kato, N., Senga, H., Suzuki, H., Ukita, M., and Kamezuka, K., (2013). "Guidance and Control of an Autonomous Underwater robot for Tracking and Monitoring Spilled Plumes of Oil and Gas from Seabed", ISOPE 2013, June 30–July 5, Vol.2, pp.366-371

**Other Conference Papers**

Kimura, R., Ukita, M., Kamezuka, K., Kato, N., Senga, H., Suzuki, H., and Choyekh, M., (2013). "Construction of Guidance and Control System and GUI of an Autonomous Underwater Robot: SOTAB-I," CAMS 2013, USB, 2013.09, Papers

Ukita, M., Kimura, R., Kato, N., Senga, H., Suzuki, Y., Choyekh, M., Kamezuka, K., "Construction of Guidance and Control System and GUI of an Autonomous Underwater Robot: SOTAB-I," Proc. of 31th Annual Conference of the Robotics Society of Japan, 2013.09, Conference Report / Oral Presentation (In Japanese)

Kato, N., Senga, H., Suzuki, H., Choyekh, M., Kimura, R., Akamatsu, T., Y., Yoshie, M., Tanaka, T. and Sakagami, N. "Development of Spilled Oil and Gas Tracking and Monitoring Autonomous Buoy System and its Application to Marine Disaster Prevention, ", Proceedings of SPE Annual Technical Conference and Exhibition, SPE166586, 2013.