



TECHNICAL UNIVERSITY OF CRETE  
SCHOOL OF PRODUCTION ENGINEERING AND MANAGEMENT

# **Autonomous Underwater Robots' Cooperative Behaviour**

Savas Piperidis

Dissertation submitted in partial fulfillment of the  
requirements for the degree of Doctor of Philosophy

December 2013



Dedicated to Alexandra for her patience.







All thesis text, figures and photographs, unless otherwise stated, are copyrighted by Savas Piperidis and licensed under a *Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License*



## **Advisory Committee**

Professor Nikolaos Tsourveloudis

School of Production Engineering and Management, Technical University of Crete.

Professor Vasilios Kouikoglou

School of Production Engineering and Management, Technical University of Crete.

Associate Professor Ioannis Nikolos

School of Production Engineering and Management, Technical University of Crete.

## **Dissertation Committee**

Professor Kimon Valavanis

Department of Electrical and Computer Engineering, University of Denver.

Associate Professor Michail Lagoudakis

School of Electronic and Computer Engineering, Technical University of Crete.

Assistant Professor Yiorgos Demetriou

School of Engineering and Applied Sciences, Department of Computer Science and Engineering, Frederick University.

Assistant Professor Lefteris Doitsidis

Assistant Professor, Department of Electronic Engineering, Technological Educational Institute of Crete.



## Abstract

A novel cooperative controller design for autonomous underwater vehicles is proposed, implemented and tested. The design procedure follows the basic principles of behaviour-based systems to create two different biomimetic roles, encapsulating the characteristics of an extremely rare cooperative underwater predator behaviour found in the wild: bottlenose dolphins group-hunting with division of labour and role specialization. The behaviour-based model is comprised of an hierarchy with different levels of competence, interaction and intercommunication between several behaviour modules. Two distinct, cooperating roles mimicking the *driver* and the *non-driving* or *barrier* members of a hunting bottlenose dolphins group emerged from the behaviour-based subsumption architecture controller designation. This behaviour-based cooperation exploits the *driver's* individual capability, in fact a physical gift, of initializing, coordinating a hunting bout, detecting and herding the school of fish along with the contribution of *non-driving* dolphins ability to comprehend and follow the *driver's* master plan and thus act as barriers for the fish trying to escape. Both roles are programmed to act upon sensory information regarding the robot's environment, provided by a vision sensor module and an inertial measurement unit. The vision sensor detects colour light targets, classifies them according to the experimentation scenario and triggers behaviours' interaction. The inertial measurement unit supports behaviours' navigation and guidance necessities. Controller testing, under several simulated scenarios, proved the reliability and modular functionality of the cooperative behaviour-based model and its potential for supporting the autonomy of underwater robotic vessels commissioned in oceanography, environmental monitoring and exploration, serving tasks that can easily be seen as an extent of dolphins' group hunting. Apart from the simulation, the same behaviour-based controller was tested in real world, with a prototype autonomous vessel. In this case, due to certain limitations of the experimental apparatus, the behaviour-based controller implemented an individual biomimetic role inspired by the real life routines of wandering, hunting, feeding, hiding and nesting exercised by underwater creatures. The individual role, behaviour-based controller was an effort to approach the sea creatures' autonomy, observed macroscopically as an inevitable iteration and commutation of distinct behaviour modules, guided by the overall objective of staying *alive*. The prototype robot was commissioned as an experimentation platform for testing the framework of autonomous behaviour-based controllers and materialize the biomimetic metaphor between the maritime environment and a laboratory experimentation area of  $1\text{m}^3$ .

x

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Water World . . . . .	1
1.2	Submarines . . . . .	1
1.3	Unmanned Underwater Vehicle . . . . .	3
1.4	Autonomy versus Remote Operation . . . . .	6
1.5	Mimicking Nature . . . . .	8
1.6	Undersea Cooperative Robotic Behaviour . . . . .	9
<b>2</b>	<b>Literature and Contribution</b>	<b>11</b>
2.1	AUV control strategies . . . . .	11
2.1.1	Atomic control . . . . .	11
2.1.2	Cooperative control . . . . .	13
2.2	Proposed cooperative AUV control strategy . . . . .	15
<b>3</b>	<b>Construction</b>	<b>17</b>
3.1	Low logistics AUVs . . . . .	17
3.2	Components of Ale III . . . . .	18
3.2.1	Ale III autonomy . . . . .	23
3.3	Log Book . . . . .	25
3.4	Software Architecture . . . . .	25
<b>4</b>	<b>Testing Controllers on Ale III</b>	<b>29</b>
4.1	Experimentation Area . . . . .	29
4.2	PD Controller Development and Testing . . . . .	30
4.3	Target Following . . . . .	33
4.3.1	Target following PD controller . . . . .	34
4.3.2	Fuzzy Logic target following controller . . . . .	37
4.4	Vessel's Stability and Seaworthiness . . . . .	40
4.5	Conclusions . . . . .	40
<b>5</b>	<b>Behaviour Based Biomimetic Control</b>	<b>43</b>
5.1	Behaviour Based Control . . . . .	43
5.1.1	Ale III autonomous behaviour showcase . . . . .	44
5.2	Behaviours Subsumption Architecture . . . . .	47
5.2.1	Level 0 . . . . .	48
5.2.2	Level 1 . . . . .	51
5.2.3	Level 2 . . . . .	53
5.2.4	Behaviours Interaction Example . . . . .	56
5.3	Experimental Results . . . . .	59

<b>6</b>	<b>Cooperative Behaviour</b>	<b>67</b>
6.1	Behaviour Based Cooperative Control . . . . .	67
6.1.1	Animal Group Hunting . . . . .	67
6.1.2	Mimicking Dolphins Hunting . . . . .	68
6.2	Ale III Dynamic Mode . . . . .	69
6.2.1	Simulation software architecture . . . . .	71
6.3	Cooperative Behaviour Architecture . . . . .	72
6.3.1	Driver role . . . . .	73
6.3.2	nonDriver role . . . . .	87
6.4	Experimental Results . . . . .	90
6.4.1	Initialization Phase . . . . .	90
6.4.2	Feeding Bout Phase . . . . .	91
6.5	Simulation Issues . . . . .	92
<b>7</b>	<b>Conclusions</b>	<b>95</b>
7.1	Ale III AUV . . . . .	95
7.2	Behaviour Based Biomimetic Control . . . . .	95
7.3	Behaviour Based Cooperative Control . . . . .	96
7.4	Further Research and Future Plans . . . . .	96
<b>A</b>	<b>Thruster Blueprints</b>	<b>99</b>
<b>B</b>	<b>Electronics Schematics</b>	<b>101</b>
B.1	Power Supply Circuit . . . . .	101
B.2	COM and Sensors Circuit . . . . .	103
B.3	COM and Motor Drivers Circuit . . . . .	105



# List of Figures

1.1	Water World. . . . .	1
1.2	Alexander the Great inside the first submarine . . . . .	3
1.3	<i>Turtle</i> submersible . . . . .	3
1.4	British submarine aircraft carrier <i>HMS M2</i> . . . . .	4
1.5	<i>Type XXI</i> submarine . . . . .	4
1.6	Deepsea Submarine <i>Idabel</i> . . . . .	6
1.7	Remotely Operated Vehicle . . . . .	6
1.8	Autonomous Underwater Vehicle . . . . .	7
1.9	Tether Management System . . . . .	7
1.10	<i>Iver2</i> AUV . . . . .	9
1.11	<i>Salamander Robotica II</i> amphibious AUV . . . . .	9
2.1	AUV Control Software by <i>OceanServer Technology</i> . . . . .	12
3.1	<i>Ale III</i> prototype AUV . . . . .	18
3.2	<i>Ale III</i> main parts . . . . .	19
3.3	<i>Ale III</i> main parts inside hull's enclosure . . . . .	20
3.4	<i>Ale III</i> first version . . . . .	21
3.5	<i>Ale III</i> custom made thruster . . . . .	21
3.6	<i>Ale III</i> roll and pitch passive stability . . . . .	23
3.7	<i>Ale III</i> surging and yawing propelling model . . . . .	25
3.8	<i>Ale III</i> near the water surface. . . . .	26
3.9	ACoS Software Architecture. . . . .	27
4.1	<i>Ale III</i> operating at the experimentation area inside the laboratory. . . . .	30
4.2	<i>Ale III</i> inside the experimentation area . . . . .	31
4.3	Yaw PD Controller . . . . .	32
4.4	Target Following Controller . . . . .	33
4.5	CMUcam3 snapshot during target following . . . . .	34
4.6	PD Target Following Controller experimental results . . . . .	36
4.7	FL Target Following Controller linguistic variables membership functions. . . . .	38
4.8	FL Target Following Controller experimental results . . . . .	41
5.1	Side view of a three dimensional representation of <i>Ale III</i> experimentation scenario . . . . .	46
5.2	Top view of a three dimensional representation of <i>Ale III</i> experimentation scenario . . . . .	47
5.3	Zero level behaviour based control subsumption architecture. . . . .	49
5.4	Zeroth level, 01- <i>proximity adjuster</i> behaviour's FSM diagram. . . . .	49
5.5	Zeroth level, 02- <i>yaw control</i> behaviour's FSM diagram. . . . .	50
5.6	First level behaviour based control subsumption architecture. . . . .	51
5.7	Level 1, 11- <i>search- wander</i> behaviour's FSM diagram. . . . .	51

5.8	Level 1, 12- <i>nesting</i> behaviour's FSM diagram. . . . .	52
5.9	Level 2 behaviour based control subsumption architecture. . . . .	54
5.10	Level 2, 21- <i>prey</i> behaviour's FSM diagram. . . . .	55
5.11	Level 2, 22- <i>predator</i> behaviour's FSM diagram. . . . .	56
5.12	Example of ACoS behaviours' interaction, phase 1. . . . .	58
5.13	Example of ACoS behaviours' interaction, phase 2. . . . .	58
5.14	Example of ACoS behaviours' interaction, phase 3. . . . .	59
5.15	Example of ACoS behaviours' interaction, phase 4. . . . .	59
5.16	Example of ACoS behaviours' interaction, phase 5. . . . .	60
5.17	Example of ACoS behaviours' interaction, phase 6. . . . .	60
5.18	Example of ACoS behaviours' interaction, phase 7 . . . . .	61
5.19	Biomimetic behaviour based control experimentation results, test A. . . . .	63
5.20	Biomimetic behaviour based control experimentation results, test B. . . . .	64
5.21	Biomimetic behaviour based control experimentation results, test C. . . . .	65
5.22	Biomimetic behaviour based control experimentation results, test D. . . . .	66
6.1	Vehicle velocity vector $\nu = [u, v, w, p, q, r]^T$ in the Body Fixed coordinate system. . . . .	69
6.2	Simulated <i>Ale III</i> . . . . .	71
6.3	<i>Ale III</i> AUV team inside Webots simulation environment . . . . .	72
6.4	Zero level <i>D0</i> control architecture. . . . .	73
6.5	<i>D01- yaw control</i> behaviour's FSM diagram. . . . .	73
6.6	<i>D01- yaw control</i> experimental results, test A . . . . .	75
6.7	<i>D01- yaw control</i> experimental results, test B . . . . .	76
6.8	<i>D01- yaw control</i> experimental results, test C . . . . .	77
6.9	<i>D01- yaw control</i> experiment. . . . .	78
6.10	<i>D02- visual search</i> behaviour's FSM diagram. . . . .	78
6.11	<i>D02- visual search</i> experimental results, test A . . . . .	79
6.12	<i>D02- visual search</i> experimental results, test B . . . . .	80
6.13	<i>D02- visual search</i> experimental results, test C . . . . .	81
6.14	<i>D02- visual search</i> experiment. . . . .	82
6.15	Level <i>D1</i> control architecture. . . . .	82
6.16	<i>D11- wait</i> behaviour's FSM diagram. . . . .	83
6.17	<i>D12- check for mates</i> behaviour's FSM diagram. . . . .	83
6.18	Level <i>D2</i> control architecture. . . . .	85
6.19	<i>D21- seek</i> behaviour's FSM diagram. . . . .	85
6.22	<i>D22- hunt/ feed</i> behaviour's FSM diagram. . . . .	86
6.20	<i>D21- seek</i> behaviour's meandrous shaped course. . . . .	87
6.21	<i>D21- seek</i> behaviour's spiral pattern course. . . . .	88
6.23	<i>Ale III</i> AUV team foraging by a school of fish inside Webots simulation environment . . . . .	89
6.25	<i>D22- hunt/ feed</i> behaviour's FSM diagram. . . . .	89
6.24	<i>nD</i> control architecture. . . . .	90
6.26	<i>Ale III</i> AUV team and a school of fish inside Webots simulation environment . . . . .	91
6.27	<i>D</i> role behaviours' interaction, during bout's initialization phase. . . . .	92
6.28	<i>nD1</i> role behaviours' interaction, during bout's initialization phase. . . . .	92
6.29	<i>nD2</i> role behaviours' interaction, during bout's initialization phase. . . . .	93
6.30	<i>D</i> role behaviours' interaction, during feeding bout phase. . . . .	93
6.31	<i>nD1</i> role behaviours' interaction, during feeding bout phase . . . . .	94
6.32	<i>nD2</i> role behaviours' interaction, during feeding bout phase. . . . .	94
A.1	<i>Ale III</i> thruster top view . . . . .	99
A.2	<i>Ale III</i> thruster back view . . . . .	100
A.3	<i>Ale III</i> thruster side view . . . . .	100

B.1	Power Supply Circuit. . . . .	102
B.2	COM and Sensors Circuit. . . . .	104
B.3	COM and Electric Motors Circuit. . . . .	106



# List of Tables

3.1	<i>Ale III</i> technical specifications . . . . .	24
4.1	Rules associating $CentroidX_{fz}$ and $dCentroidX_{fz}$ . . . . .	39
4.2	Rules associating $CentroidY_{fz}$ and $dCentroidY_{fz}$ . . . . .	39
4.3	$pixelsNumber_{fz}$ rules . . . . .	39
5.1	Behaviours Index . . . . .	48
5.2	ACoS reflexes in accordance with the detected light colour signs . . . . .	57



# Nomenclature

ACoS	Ale III Autonomous Controller Software
API	Application Protocol Interface
AUV	Autonomous Underwater Vehicles
COM	Computer On Module
D	Driver
FL	Fuzzy Logic
FSM	Finite State Machine
GPIO	General Purpose Input Output
GPS	Global Positioning System
GUI	Graphical User Interface
I/O	Input- Output
IMP	IMU Module Program
IMU	Inertial Measurement Unit
nD	non Driver
PD	Proportional Derivative
PWM	Pulse Width Modulation
ROV	Remotely Operated Vehicles
TTL	Transistor- Transistor Logic
UAN	Underwater Acoustic Network
UAV	Unmanned Aerial Vehicle
UUV	Unmanned Underwater Vehicles
VSMP	Vision Sensor Module Program





# Chapter 1

## Introduction

### 1.1 Water World

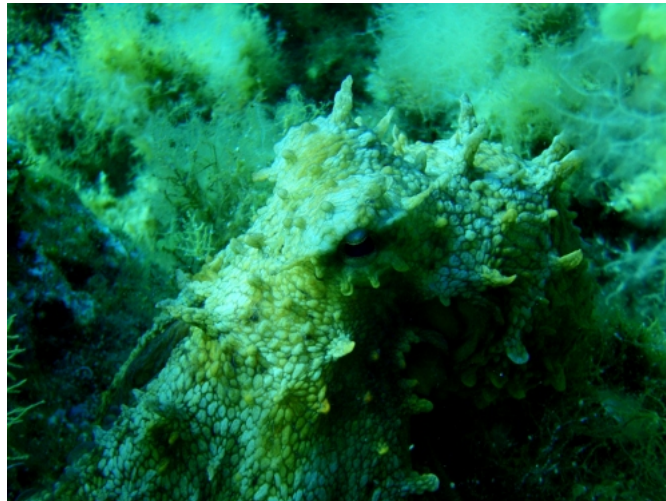


Figure 1.1: Water World.

More than 70% of earth's surface is covered by water. The existence of underwater food reserves was the first reason for human activities expansion to the water world. Great civilization progress was triggered by this transgression of human, terrestrially bounded specifications. Ship construction and over seas navigation changed forever the human interaction with nature. Underwater natural resources, such as fossil fuels along with the exploration efforts for the vast ocean biodiversity, were the basic reasons to promote marine activities and research during last century [44].

### 1.2 Submarines

Combining empirical design solutions with the salutary buoyancy force, pioneer marines conquered offshore water surface a long time ago, at the prehistoric millenniums. The world underneath water's surface, though, was a bleak, hostile place hunted by sea imaginary monsters, until the first decades of twentieth century when

the first world war unleashed a deadly weapon capable of independent operation underwater: the submarine. It was a *monster* itself, not at the imagination of Jules Vern but constructed by that era's state of the art naval architecture.

The first submarine is ascribed to Alexander the Great by a Hellenic legend. According to the legend, the fearless warrior's desire for exploration led him to dive to the Indian Ocean, inside a bell shaped, crystal construction, suspended from one of its ships, during his military campaign to Persia at the fourth century B.C. Figure 1.2 is a legend depiction on a Hellenic stamp printed in 1977. The first published study for a submarine is dated back in 1580 when William Bourne, an English dilettante scientist, prescribed a boat with expanding and contracting structures altering its buoyancy [81]. This innovative technique that would, theoretically, enable the whole boat to dive underwater and emerge to the surface, was never implemented. After 40 years, in 1620 Cornelius Drebbel, a Dutchman in the service of James I of England, constructed the first submarine and tested it in the Thames [51].

Several attempts took place during next century, for building underwater crafts, by European or American scientists and inventors. Figure 1.3 is a design of the *Turtle*, a military submarine designed in 1775 by David Bushnell, a Yale College undergraduate, was the first verified submersible capable of independent underwater operation and movement [141]. *Turtle* embodied the basic operational requirements of a submarine: the ability to submerge, the ability to maneuver under water and the ability to maintain an adequate air supply to support the operator of the craft. To achieve these requirements, Bushnell devised a number of important innovations:

- *Turtle* was the first submersible to use water as ballast for submerging and raising the submarine.
- It was the first craft to use a screw propeller for maneuvering under water.
- Bushnell invented a breathing device that was part of *Turtle's* equipment.

*Turtle* participated in the operations of the American Revolutionary War. Although it did not achieve military success, it demonstrated submarine's operational potentials and triggered a lot of new inventions and innovations that led, in 1863, to the first mechanically powered boat not to rely on human power for propulsion and to use compressed air to empty its ballast tanks: the French *Le Plongeur* [144]. Powered by a compressed air engine, with a length of 46 meters, width of 7 meters and displacing 400 tons, *Le Plongeur* was by far the largest submarine to appear before the twentieth century. At the start of 20th century important technological evolutions appeared in submarine development. Diesel electric propulsion became the dominant power system and equipment such as the periscope became standardized. In 1896, the Irish inventor John Philip Holland designed submarines that, for the first time, made use of internal combustion engines on the surface and electric battery power submerged. Holland's underwater crafts were put into service by navies, which included Great Britain, Japan, Russia, and the United States [65]. World War I established the use of submarine as the new naval force of attack, reconnaissance and espionage, operating effectively with range beyond 5.000 miles and speed of eight knots [32].

During the interwar years, the most notable design developed was the submarine aircraft carrier, acting as a reconnaissance unit at a time when radar still did not exist, Figure 1.4. A catapult was adapted to submarine's deck along with a waterproof hangar for launching and recovering one or more small seaplanes.

World War II had a great impact at the submarine technology evolution, best represented by the German *Type XXI* class, the first submarines designed to operate primarily submerged, Figure 1.5. A *Type XII* class submarine could dive at a depth of 260 meters, had a range of 15.500 nautical miles surfaced and 385 miles submerged and could achieve an operational speed of 17 knots surfaced or 16 knots submerged [134].

After 1950 two new technologies were implemented at submarines' operation: the nuclear powered propulsion and the development of equipment able to extract oxygen from sea water. After these progresses a submarine's operational range and the submersion periods were limited only by factors like food reserves and crew morale inside vessel's limited space, claustrophobic hull. Great achievements like the voyage of *USS Nautilus*' crossing of the North pole beneath the Arctic ice cap in 1958 [13] and the *USS Triton's* submerged circumnavigation of the world in 1960 [24] proved post war submarines' operational potentials and reliability. By the year 2000, 47 nations operate more than 700 submarines, almost three hundred of them nuclear powered while new designs are being pursued in the United States, Germany, Italy, Denmark, Norway, Sweden, and Japan [81].



Figure 1.2: Alexander the Great inside the first submarine. This picture is in the public domain.

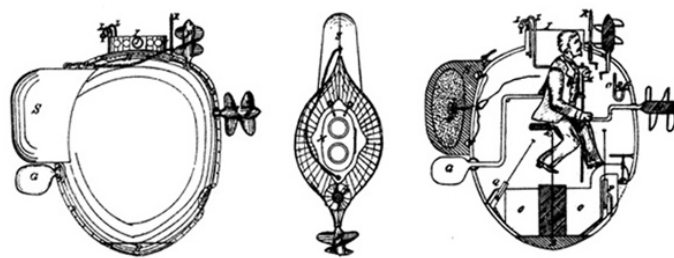


Figure 1.3: *Turtle* drawing made by F. M. Barber in 1875 from description left by its designer David Bushnell. Picture by *Naval History & Heritage Command, US Navy*/This picture is in the public domain.

Although the overwhelming majority of submarines were constructed to serve military operations, several vessels were designed as civil submarines and served tasks like tourism, sea bottom survey, exploration and biological research, underwater pipeline and cable inspection, sea platforms maintenance and monitoring. Tourist activities funded the construction of 45 civil submarines during the period from 1985 to 1997, operating around the world [174]. Recent designs like the deep sea submarine *Idabel*, Figure 1.6, can transport three people at a depth of 900 meters, achieving a range of 10 nautical miles, serving tourism, exploration or biological research [140].

## 1.3 Unmanned Underwater Vehicle

Nowadays submarines are able to offer high quality military, industry and scientific research services underwater. On the other hand, their enormous operating cost along with the physical and psychological limitations of their crew, servicing in such a hostile environment, set hard limits for submarines' missions. Unmanned Underwater Vehicles (UUV) were designed to overcome these drawbacks in submarine underwater operations. Since UUV do not confront the necessity for human life support their dimensions and cost are significantly lower and moreover, their operation does not suffer from limitations concerning food reserves or crew physical and psychological fatigue. There are two types of UUV:

**Remotely Operated Vehicles (ROV)** tethered via umbilical cables to a *master* facility, usually hosted in a surface vessel or platform. The cables offer unlimited support of its operation by the means of power supply and bidirectional, high bandwidth, data transfer. An ROV may be equipped with one or more manipulators with grabbers or several other tools. It requires continuous human help for all of its operations such as navigation, positioning and manipulation. ROVs may carry out complicated

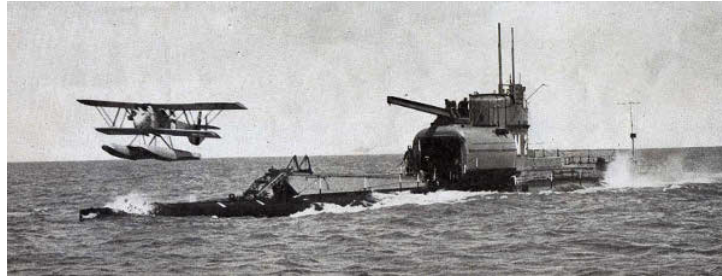


Figure 1.4: British submarine aircraft carrier *HMS M2* launching a specially designed *Parnall Peto* seaplane. This picture is in the public domain.



Figure 1.5: A *Type XXI* submarine taken over in April, 1948, by the US Navy, as *U-3008* and used as a test vessel. Picture by *Naval History & Heritage Command, US Navy*/This picture is in the public domain.

military, industry, scientific and salvage tasks free of submarine's limitations and safety issues. Still, their operational cost remains high, outreaching the daily price of 7.500€[188], because of the necessary surface crew support. Moreover ROVs' umbilical cables introduce the need for a tether management system increasing the overall cost and complexity of vehicle's operation. Also an ROV may be lost due to entangled or snapped tether [188]. ROVs have proved their operational capabilities as commercial products covering a great variety in pricing, equipment and performance. There are agile and pilot friendly *mini* class, low cost models, weighting no more than 10kg with operational depth up to 300m [57], [68], [60], [167], [92], [93] used at:

- recreational and scientific underwater explorations [124],
- documentary film making since capturing footage in deep, dangerous, and confined areas unattainable by divers, for submersion periods without time limitation [122],
- ship, propeller, pipeline, cable and oil platform inspections [153],
- search and rescue inside shipwrecks and other potentially hazardous environments [85],
- nuclear power plants and reactors , tank and hydroelectric structures inspection [88], [76],
- aquaculture fish farms cages inspection [76],
- military and law enforcement [57].

*Observation* class ROVs have a powertrain up to 40kW, weight up to 500kg and may dive to depths up to 3000m [164], [91], [94], [128], [95], [123], [97], [57], [68]. *Observation* class overwhelms *mini* class

capabilities with an inevitable cost in agility and pricing. Thus, *observation* class ROVs are commissioned to more demanding, open and deep sea tasks like:

- oil and gas pipeline engineering intervention in petroleum industry,
- cleaning debris, sand and mud from deep sea facilities,
- mine countermeasures and torpedo recovery in the military,
- deep sea salvage and accident investigation,
- oceanic scientific research covering aspects of chemical oceanography, rock coring and the study of thermal vents.

*Work* and *heavy duty* is the most powerful classes of ROVs. Featuring a thrust power up to 300kW and weighting up to 5000kg, these classes' vehicles are able to undertake the most demanding commissions of underwater cable servicing and deep sea engineering intervention, even at the outrageous depths of 6000m [94], [97], [90], [151]. At 1995, Kaiko, a 10000kg ROV, constructed by the Japan Agency for Marine- Earth Science and Technology dived at 10911.4m, inside Mariana Trench, the deepest ocean bottom on Earth [115]. Eight years after Kaiko was lost due to accidental breaking of its tether during a research off Shikoku Island [183].

**Autonomous Underwater Vehicles (AUV)** able to undertake underwater tasks without any sort of external pilot or guidance support. AUVs are not tethered and so there is no umbilical connection to a master vessel or any kind of host platform during their undersea commissions. Although the torpedoes may be considered as the first submersibles presenting autonomous behavior, yet there was no AUV development until the 1970s. Before that, a few AUVs were built, the first one being the Self Propelled Underwater Research Vehicle known as SPURV, developed at the University of Washington in 1957 [29]. SPURV had an operating depth of 3000m, an energy autonomy of 4 hours with the speed of 5 knots and serviced, until 1979, in scientific commissions like Arctic [131], diffusion, acoustic transmission, and submarine wakes research. In 1970's the University of New Hampshire's Marine Systems Engineering Laboratory developed the open space frame vehicle EAVE [100]. SKAT was the first AUV built in Russia, at the Shirshov Institute of Oceanology, *OSR-V* was the first Japanese and *EPAULARD* the first French vehicle in 1970s [177]. The next three decades, until the arrival of the new millennium, was a time of experimentation with some successes and many failures. AUVs designed and developed until 2000 were either large, inefficient, expensive, or a combination of all three [177].

Underwater autonomy had to wait for the delivery of innovative technological solutions confronting aspects of energy storage, computational efficiency and underwater sensing. Last fifteen years' underwater technology evolution triggered the development of 231 unique AUV configurations of 133 vehicle platforms [21]. Although most of the contemporary AUVs is the result of ongoing research and development in academia [127], [19] and government funded, military [137] or civil [20], [55], [129], [101] services, underwater technology is nowadays mature enough to present a range of fully functional commercial products, ready to undertake commission 'out of the box' [2], [1], [3], [169], [56], [97], [151] and service the following tasks:

- coastal and sea bed mapping, beach survey, rapid environmental assessment and monitoring, [124],
- oil and gas fuel industry [14],
- cable deployment and route survey [2],
- military uses including mine detection and countermeasures, explosive ordnance disposal, anti-submarine warfare, covert intelligence, surveillance and reconnaissance [2], [1],
- geological, geographical and hydro- graphical surveys [102], [82], [7],
- hull inspection [83].



Figure 1.6: Deepsea submarine *Idabel*, property of Roatan Institute of Deepsea Exploration. Picture by *Stanley Submarines*/Used under permission granted by *Karl Stanley*.

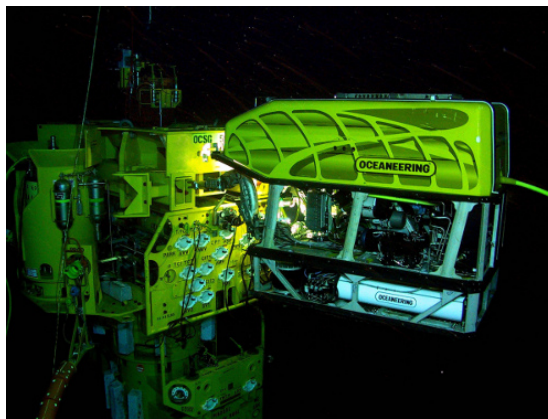


Figure 1.7: An *Oceaneering* Remotely Operated Vehicle in an underwater oil and gas field, operating a wrench torque tool on a valve. This picture is in the public domain.

## 1.4 Autonomy versus Remote Operation

The absence of umbilical cables grants unparalleled maneuverability and agility to AUVs, compared to tethered ROVs. The untethered vessel may achieve steep course changes without cable bend radius limitations. It may also follow rotating paths, enter inside cavities and holes, maneuver through contiguous obstacles, take over commissions under ice covered sea, lakes and rivers without the risk of entangling or breaking the cable. There is no danger to lose control or worst, to lose the vessel due to a broken tether or some kind of cable's failure. AUV's operation does not require the support of a tether management system, as does the Figure 1.9. Thus, no matter how far or deep AUVs are operating, their commission is free of the complexity, the weight and the cost of such a system. This is a significant advantage, particularly in the case of long range and deep sea commissions. Tether's total length may be 5000m or more, resulting to a gigantic, energy consuming umbilical system weighting more than 15000kg for both the tether management and the launch- recovery system [151].

AUVs' potential for free movement is exploited by the use of specifically designed *gliders*. A glider AUV, instead of using electrically driven propeller, exploits small changes in its buoyancy in conjunction with the hydrodynamic lift forces exerted on a pair of wings attached to the glider hull to convert its, emerging or diving, vertical motion to horizontal and thereby propel itself forward. As a result, the vehicle moves through the water in a saw-tooth like pattern consuming extremely low energy amounts. Pitch and roll are controlled using



adjustable ballast. Usually, the vessel's battery plays the role of ballast [187]. Gliders often emerge to sea surface not only to determine and correct its position estimation using the Global Positioning System (GPS), but also for land-station contact through a mobile phone link or a dedicated radio link for data transmission and possible on-line modification of glider mission's requirements [7]. Navigation is accomplished using a combination of GPS readings while on the surface and internal sensors' measurements regarding vehicle's orientation, depth and attitude during dives. Gliders are usually commissioned in oceanography [136], [8] where, due to their minimized energy consumption, may undertake ultra long range offshore missions. On November 2013 *SeaExplorer* glider [155] breaks AUVs missions' duration and distance records, reaching the mythic milestone of 60 days and 1183km of continuous offshore autonomous operation on a single battery. During its two months mission, the glider averaged 0.5 knots and provided over 1168 profiles of the water column from near surface to 500 meters deep [6].



Figure 1.8: *Autosub 1a* Autonomous Underwater Vehicle on surface, ready for recovery. Picture by StifynTonna licensed under Creative Commons Attribution-Share Alike 3.0 Unported CC-BY-SA-3.0 license, via Wikimedia Commons.



Figure 1.9: A *Nord Stream* ROV inside tether management system's *garage*, ready for launching. The *garage* is tethered to mothership's launch- recovery system. Picture by *Nord Stream* AG/Permission granted.

Figure 1.10 shows *Iver2*, the evolution of *Workhorse* AUV [10], a commercially available model indicating contemporary global market trends for designing and constructing low logistics AUVs. It is developed by

OceanServer Technology, Inc. [139] as an 100m depth rating, *affordable next generation* vessel. Its basic configuration pricing starts at 57200\$ and includes a digital 3 Axis Compass, depth sensor, altimeter inside the standard 80cm torpedo tube.

The absence of the umbilical cable connection, supporting the autonomous vessel's operation and navigation, introduces certain technological as well as theoretical challenges [14]. Autonomy in an unknown and unstructured underwater environment implies a wide range of technical and research topics:

- on board intelligence to pilot the vessel, since there is no remote operator,
- machine reasoning and learning for the ability to react in a reliable way to unexpected situations [102],
- robust and heterogeneous sensory system able to support intelligent techniques inside the noisy underwater world [38], where vision based systems are not fully reliable due to the generally poor visibility and color absorption,
- localization and navigation with the lack of GPS support, since electromagnetic transmission of such signals is impossible underwater [106], [104], [84],
- limited online communication, usually acoustic via a low bandwidth channel, since radio frequencies do not propagate well in seawater and information transfer, between vehicles as part of a team or between the vehicle and the surface, become both critical and difficult [38],
- non-linear dynamics, strongly affected by hydrodynamic and added mass effects [22],
- design of energy efficient, low consumption electronic devices, thrusters and actuators since on board energy resources are limited,
- development of manipulators adapted to AUVs' specific needs [62],
- high density energy storage, battery recharging using wireless power transfer [16], fuel cell [87] and solar power [185] technologies.

## 1.5 Mimicking Nature

Biomimetism and bioinspiration have proved to be innovative tools in AUV design and control. Mimicking nature in biorobotic projects is like exploiting the results of a worldwide research effort lasting several hundreds of thousands years: *the physical evolution*. Often AUVs encapsulate design solutions that lie far from how nature has evolved in oceans. They use, for example, propeller thrusters for moving instead of fins. Also, some of them are constructed with open space frames instead of the common eye drop or torpedo shaped fishes' silhouette. Still there are several AUVs that inherit, some or most, of their characteristics from the vast biodiversity of aquatic or amphibious species.

The anatomy of sea creatures has evolved through the ages improving their hydrodynamics and efficiency. Thus, the design and the mechanical structure of certain AUVs is mimicking the anatomy found in some sea creatures [79], [133], [142], trying to exploit evolution's fittest solutions. The torpedo shaped design, adopted by the majority of contemporary AUVs is mimicking open sea fishes' streamlined shape. Apart from the structural design, the physical evolution process has resulted in several undersea [46], [190] or amphibious [47], [184] locomotion techniques proved to suitable for adequate AUV propelling or swimming.

Biomimetism has affected robot control theory as well. Based on certain observations on animal specific behaviours, emergent biomimetic control methodologies may be developed to take advantage of the dynamics of interaction among the layered behaviours themselves and between the behaviours and the environment [126]. In [12] a fundamental assumption is presented: in order for a robot to act autonomously over a wide range of tasks and environments, it must be capable of exhibiting a variety of different behaviours. This assumption is supported by ethological observations on the behaviour of animals and differs from traditional approaches to robot control which have concentrated on the development of a single behaviour, often to the exclusion of others, featuring intensive use of accurate measurements, numerical models and control theory. Biomimetic



control proposals like in [28] favour the, necessary for AUV applications, concepts of perception, situation, skill and *behaviour*. The same concepts are used to describe human and animal *behaviour* as well.

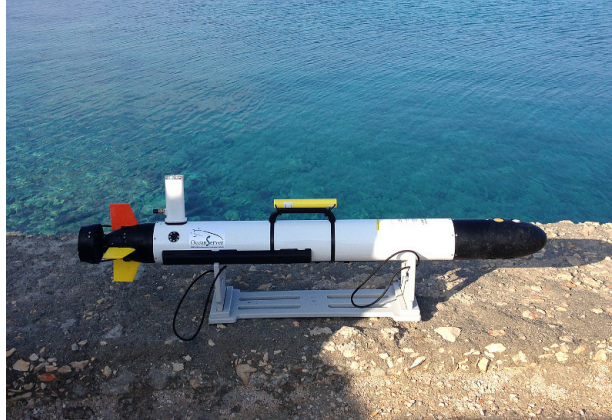


Figure 1.10: Iver2 AUV by OceanServer Technology, Inc. Picture by *OceanServer Technology, Inc.*/Permission granted.



Figure 1.11: *Salamander Robotica II*, a biomimetic amphibious AUV. Picture by Kostas Karakasiliotis, *Biorobotics Laboratory*/Permission granted.

## 1.6 Undersea Cooperative Robotic Behaviour

AUVs not only operate alone but also in teams, using low bandwidth acoustic or other kind of intercommunication, as electromagnetic waves do not propagate in seawater. Since autonomous movement eliminates the danger of entangling tethers, a team of cooperating AUVs may undertake more redundantly and effectively missions like oceanography and sea bed mapping. Such undersea missions typically have large areas to cover while AUV are not able to carry heavy payload individually, because of spatial and energy sources limitations. Therefore, research efforts are being done on multi AUV approaches using strategies like leader follower [48], artificial potentials and virtual structures [63], mainly in AUV commissions of area coverage. In [39] a generic multi- agent based layered architecture for designing and specifying AUV teams at a high level of abstraction, regardless of AUVs characteristics and skills, is proposed following the multi agent systems trend of research, to compete the drawbacks of previous approaches that:

- lack robustness, in the case of unstructured environment without any prior knowledge or maps for it.
- are not scalable because they depend on reliable communication. Undersea intercommunication channels are acoustic suffering from high noise levels and low bandwidth. Thus, the communication transfer rate is limited, introducing *bottleneck* phenomena as the number of the AUV team members increases.
- do not cope with a potential team members heterogeneity in the flotilla meaning that different individual AUV skills and capabilities can not be exploited inside the team and
- and vehicles with lower performance penalize the whole group.

A new computational simple and scalable approach that can very straightforwardly embed any type of physical or other constraints and limitations, like obstacle avoidance, non linear sensor noise models and localization fading environments is presented in [108]. This approach claims that navigates a team of AUVs exploring unstructured dynamic environments, providing accurate static and dynamic maps during oceanographic missions.

Due to the known limitations of underwater communication, the fundamental idea behind the cooperative approach in [37], based on the formalism of potential game theory is that each vehicle must explicitly include communication requirements within its overall mission objectives. This way, each AUV team member is able to balance its actions to maintain a desired communication performance and guarantee fulfillment of its assigned task.

A common oceanography task performed by AUVs is undersea sampling to produce a map of a given environmental quantity, such as water temperature, sound speed and sea bed morphology, accurately and in the minimum amount of time. If the sampling is performed by a team of AUVs, then a sampling strategy must be chosen to exploit the availability of multiple vessels through some coordination strategy, possibly implemented in a distributed fashion. In [7] an original algorithm is presented that solves the distributed coordinated adaptive sampling problem with the additional constraint of limiting the maximum distance between two adjacent AUVs and preserve underwater communication potentials.

In many offshore, underwater oil and gas field missions, a team of collaborating AUVs is preferred than a single vehicle, since it provides greater flexibility and redundancy. One significant application of cooperating AUVs is to inspect underwater oil pipelines where a number of AUVs are required to hover above the pipeline at identical or different depths along parallel paths, and map the pipeline from different viewpoints using multiple copies of the same suite of vision sensors. In [98] a specific AUVs' formation is considered to visually cover and monitor throughout the pipeline using a new navigating and tracking control scheme based on an adaptive region boundary approach. The control formulation utilizes a boundary as desired target rather than a region or a point and in addition, an edge-based segmentation approach is used to allow the team members to navigate into a desired location on the boundary lines or surfaces, whilst the target itself is moving.

AUVs may also join heterogeneous teams with Unmanned Aerial Vehicles (UAVs), forming a combination easily adapted to possible changes in both mission objectives and environmental parameters. The sea surface spatial coverage ability of the UAV combined with AUVs capability to measure various oceanography environmental parameters with high accuracy can be used in open sea missions such as tracking an oil spill, measuring temperature gradients and harmful algal blooms [166].

# Chapter 2

## Literature and Contribution

### 2.1 AUV control strategies

During last decade an outgrowing number of AUV models and control strategies, available either commercially or as academia research prototypes, were developed. The extremely noisy and unstructured undersea environment along with the highly non linear and coupled underwater vessels' dynamics pose a great challenge for the design and implementation of efficient and robust robotic controllers. Electromagnetic undersea transmission of GPS signals is impossible and thus AUV navigational tasks are far more difficult than the ones concerning autonomous robotic ground or aerial vehicles. Limitations and impediments grow bigger when the control strategy implements a cooperation task among an AUV team. Instead of several reliable online, indoors or outdoors available communication choices when experimenting with ground or aerial robots, there is usually only one undersea choice of communication: acoustic, via a low bandwidth channel. Thus underwater cooperative control schemes have to explore novel methodologies to confront with this hostile environment.

As a result of AUVs operational potentials expansion during last decade, several control schemes have been developed and tested in simulation, in harbors, across coastal sea line or offshore in the vast oceanic environment. These schemes refer to the low level atomic control like, among others, hardware and actuators manipulation, sensor fusion and image processing. Higher level atomic control strategies, command AUV's devices, process sensory input and undertake the decision making task, navigating and guiding the vessel according to mission's targets. Moreover, during last years the potential of swarming AUVs under a common task led to the development of novel cooperative control methods or properly adapted versions of already existing ones.

#### 2.1.1 Atomic control

Modular design is one of the basic characteristics of AUV controllers. Distinct controller modules are differentiated according to vessel's specifications and payload. *Stingray* AUV, a compact lightweight vessel used in academia research and international competitions, uses a software architecture that implements a deliberative control scheme and comprises of individual modules responsible for individual undersea tasks, intercommunicating via a custom proprietary on board messaging system [23]:

1. A *navigational* module connected to the compass and the pressure sensor that estimates the orientation and depth.
2. A *vision* module implementing image processing algorithms of edge detection and Hough probabilistic transform, programmed and tuned a priori to accomplish a specific colour or object detection and recognition tasks.
3. A Graphical User Interface (GUI), module for communicating with the vessel's controller while tethered. Using this module the user may tune the controller parameters, monitor the data logger and program

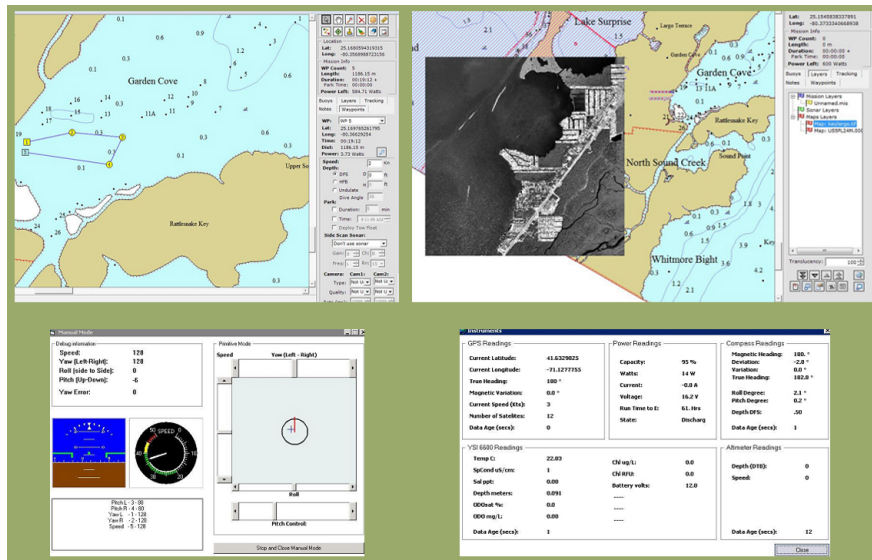


Figure 2.1: AUV Control Software developed by *OceanServer Technology, Inc* as a commercially available product for *Iver2* autonomous vessel. Pictures by *OceanServer Technology, Inc.*/Permission granted.

the vision module into various object search modes.

4. A *planner* module implements a deliberative control taking into account missions parameters. It communicates with the navigational and vision modules to gather the necessary sensory inputs and produces thruster commands as outputs of proportional integral derivative, *PID*, control loops.

*PID* deliberative control, along with sliding mode control, is a robust and reliable solution used in several AUV projects [34], [110], where the precise estimation of vessel's dynamics is necessary. In [116] a dynamic diving controller based on Lyapunov theory and back stepping techniques was used, assuming a perfect knowledge of AUV's dynamic parameters. Then, an adaptive scheme was designed to obtain the desired robustness. The same technique was used for the problem of trajectory tracking of an under actuated, glider AUV moving in the vertical plane with the aid of an internal moving mass [119].

A different approach is chosen in [106] for the implementation of a general modular fuzzy logic control architecture for sonar sensor based AUV navigation in a three dimension unknown undersea environments. Its advantage is that no assumption is made on the AUV type, the amount of a priori knowledge of the 3-D undersea environment, or the static and dynamic obstacle size and velocity. The proposed technique comprises three levels of fuzzy controllers with *goal based* characteristics when no obstacles are considered or *reaction based* characteristics when obstacle avoidance is necessary:

1. The *sensor fusion* module, responsible for position monitoring and obstacle detection, receives sensory input and provides information about potential collisions in four main motion directions: front, back, left, right.
2. The *collision avoidance* module receives as inputs the calculated collision possibilities together with the vehicle heading pitch error from a target point and generates the new target point, or the next way point.
3. The *motion control* module determines the thrusters' and fins' commands in order to reach the goal point with the target surge velocity. It is composed of five subsystem controllers, the speed control, the heading control, the depth control, the roll control and the ocean current control subsystems.

Simulation results proved that the proposed fuzzy logic controller performed well, even for simultaneous roll control and ocean current presence. In the specific case that the AUV navigational task is undertaken inside a structured, or known undersea area, the path planning strategy may be accomplished via genetic algorithms and B-Splines [105].

AUV navigation and localization may, also be accomplished, using triangulation techniques upon laser pointers or stripe visual traces [103], [49], [112]. These techniques fuse data deriving from laser traces projection on the image plane and computer vision object tracking algorithms to estimate the position vector of the vessel and navigate it using a closed loop steering control [109].

*Inertial navigation* systems and *dead reckoning* methods provide a self contained passive means for three dimensional positioning in the open ocean with excellent short term accuracy. Over time, drifting gyroscope problems introduce errors and degrade navigation accuracy. Such errors may be minimized using *simultaneous localization and mapping* methods [179], [178], [18], [61] in conjunction with *long baseline systems*, which entail cumbersome and time consuming installation and calibration procedures, hull-mounted *short baseline systems*, which have to be rigidly mounted to a vessel hull and are affected by the natural bending of the hull and finally *ultra short baseline systems*, which provide factory calibrated and fast deployable systems that are suited for low cost navigation systems [31], [172], [135], [82].

In the category of commercially available products there has been a great progress on the control software. Contemporary AUV products provide user friendly GUI software that enables AUV operation without the use of programming languages for mission planning, hardware and sensor manipulation. Iver2 AUV [10], designed and constructed by OceanServer Technology, is equipped with an *Underwater Vehicle Console Program* and the mission planning *VectorMAP* software, Figure 2.1. Underwater Vehicle Console Program is a remote desktop utility that enables user to log on and operate the vessel as long as it is on surface and inside wireless communication range. OceanServer's Console Program operates in compliance with AUV's safety rules and facilitates manual operation, sensors' monitoring and data logging engine deployment. Navigation tasks are accomplished via a dead reckoning on board system and an integrated active navigation Doppler velocity log system. VectorMAP is a higher level planning software presenting an intuitive GUI that allows AUV's operator to design oceanographic missions easily, alter mission's parameters on the fly while on surface and extract completed data files for post processing. It is important to mention that Iver2 is a compact, cost sensible AUV, its basic configuration pricing starts at 57200\$ and includes the licences of the above mentioned control software modules. It is a characteristic case of a new family of user friendly and affordable AUV control solutions for research and oceanography commissions.

## 2.1.2 Cooperative control

Cooperative control assumes that there is, in atomic level, a reliable and robust low level on board controller for every cooperating AUV team member. Each AUV, following the methodologies described in the previous subsection, obeys the higher level- cooperative commands and materializes them according to its native low level atomic controller. Due to the impediments in undersea communication data transfer rates and bandwidths, multi AUV controllers are developed in a distributed and adaptive fashion [36], [159]. A distributed cooperation algorithm for a team of prototype AUVs, commissioned in oceanographic sampling applications is described in [7]. In such missions, the technique of adaptive sampling is used when full coverage is not strictly required or it is not possible because sensors can only make a point- wise measurement. The adaptivity of sampling increases the range of spatial sampling wherever the environmental map is rapidly changing, while decreases it when the environmental map is almost constant. Several approaches make use of a dynamic programming algorithm which solves the coordinated adaptive sampling problem with the additional constraint of limiting the maximum distance between two adjacent vehicles in a distributed fashion, taking into account the range constraint of vessels intercommunication. The cooperation algorithm was tested with a test case scenario inside a simulated environment and showed an emergent team behaviour in terms of team formation, indicating that constrained cooperation naturally leads to a coordinated motion.

Multi agent technology principles may also create an architecture to control and coordinate multiple AUVs in undersea communication deficient environments. The multi agent architecture in [163] exploits a simple

broadcast communication system in conjunction with real time vehicle prediction to handle the limitations inherent in multi AUV operations. Another approach to the multi AUV control problem, with capabilities like multi agent exploration, is described in [35] that proposes a cooperative distributed search algorithm based on a minimum entropy method.

A cooperative control scheme is implemented in [38] by AUV teams commissioned in surveillance tasks. The integration of AUVs as mobile nodes of a communication Underwater Acoustic Network (UAN) [86] and as surveillance assets, acoustically controlled by a command and control centre, serves as a response against possible intrusions. Each AUV is a node at a UAN comprised of four more fixed nodes and one attached to a support ship. This scheme's goal is the implementation of a mission *supervisor* capable of interpreting and generating messages to the other network nodes, and to give commands to each vehicle's native *Guidance, Navigation* and *Control* system, following a *back seat driver* paradigm [26] and the *Mission Oriented Operating Suite* open source software project. The Mission Oriented Operating Suite provides core middleware capabilities in a publish- subscribe architecture, as well as several intercommunication implementations. The key idea in the back seat driver paradigm is the separation of vehicle *autonomy* from vehicle *control*. The autonomy system provides heading, speed, and depth commands to the vehicle control system. The vehicle control system executes the control and passes position, heading, and speed navigation information to the autonomy system. Thus, each AUV's mission supervisor, the *autonomy* part of the controller, must be able to make decisions and give high level commands to the native *control* vehicle system, which is solely responsible for the low level execution of the commands. The mission supervisor is divided into different independent modules each of which is assigned to a specific task, like centralized decision making or communications. The decision bot is implemented as an event- driven Mealy finite state machine, which generates an output based on its current state and input. During a five days testing, the implemented controller solution showed robustness in preserving network connectivity and group functionality at a basic level without any major malfunctioning.

The *Noptilus* project presents a new approach that is capable to efficiently and autonomously navigate a team of AUVs when deployed in exploration of unknown static and dynamic environments towards providing accurate static and dynamic maps of the environment [108]. Instead of using simultaneous localization and mapping and target tracking algorithms, a cognitive based adaptive optimization approach [113] is used to develop a scalable and computationally efficient controller. The controller confronts the problem as if optimizing the active exploration criterion, that is a function of AUVs' positions and the environment conditions inside the area where the AUVs live, like relative positioning of landmarks and targets. This optimization is constrained by several parameters like the inevitable fading of localization estimations based on IMU, bottom sensing and depth sensor readings, the non linear noise affecting the sensory inputs, the limited visibility, the obstacle avoidance and the scalability. Realistic simulation experiments exhibited the efficiency of the proposed approach presenting certain advantages such as its computational simplicity and scalability, and the fact that it can very straightforwardly embed any type of physical or other constraints and limitations.

Another cooperative task is the formation control of an AUV team accomplished by the artificial potential field method based on virtual bodies [64]. This methodology allows for adaptable formation control and can be used for missions such as gradient climbing and feature tracking in an uncertain ocean environment. A control strategy for multi AUV formation control is the *leader- follower* controller implemented in [189] by introducing the notion of a virtual structure: AUVs with underwater communication range constraints are divided into some clusters according to their relative positions and leaders in all clusters can track their desired trajectories based on a virtual structure. Alternatively an  $l - \phi$  controller may implement the *leader- follower* formation strategy, manipulating both the distance and the angle between the leader and followers to achieve certain given values. In [84] and [152] a virtual AUV fuzzy logic controller is used to confront the cooperative  $l - \phi$  controller based on the assumption that the navigation and obstacle avoidance of the multi AUV formation is converted to the path planning of the virtual AUV. Another fuzzy cooperative control scheme, described in [107], uses a behaviour fusion method for the multi AUV system, creating a hybrid architecture of a navigation supervisor for the leader and follower AUV respectively. Behaviour fusions for the leader and follower AUV employ fuzzy logic to weight the behaviours using local sensor information and fuse the fuzzy command actions. The fuzzy actions are weighted by the navigation supervisor and fused through fuzzy reasoning to produce a command action, from which the final crisp actions for the leader and follower AUV are determined by defuzzification.

## 2.2 Proposed cooperative AUV control strategy

The main subject of this dissertation is to design, implement and test a novel behaviour based multi AUV control model, by taking into account the methodologies, control issues and solutions discussed in the previous section. The novelty lies in the fact that the proposed model is inspired by and mimics an extremely rare behaviour found in the wild: the bottlenose dolphins, *Tursiops truncatus*, group-hunting with a division of labour and role specialization [75]. Moreover the proposed control model is distributed and its architecture serves firm simplicity and scalability constraints. A three level hierarchy of behaviour modules was designed to form an equal number of levels of competence for a cooperative AUV behaviour based model, according to the behaviour based control principles [126], [33]. It is important to mention that the cooperative controller's design and development were made without making any assumptions or computations regarding vessel's dynamics or its environment specifications, thus emphasizing on the controller's potential for modularity, expandability and applicability to different types of AUVs and missions. The distinct behaviour modules were specifically designed to encapsulate the two distinct roles in dolphins group hunting: the role of the single *driver* that initializes and leads the hunting bout to a detected target of school of fish and the role of two or more *non driving* or *barrier* dolphins that follow their leader. The proposed controller was implemented and tested extensively with a homogeneous AUV team, comprised of vessels equipped with vision and inertial measurement unit sensors. The intercommunication between driver and non driving AUV is accomplished via visual light signs produced by the omni directional colour light source, positioned on top of each vehicle.

The proposed strategy's special characteristics differentiate it from the control methodologies depicted in the previous section. Although there have been published works [114], [162], [40], [27], [41] concerning behaviour based cooperative AUV control, still there is nothing among them being analogous to the biomimetic bottlenose dolphins cooperative strategy suggested here. Contemporary multi-AUV cooperative hunting algorithms cover the issues of target searching, formation of dynamic alliance and pursuing to capturing using complex and demanding methodologies [181]. The dolphin predator biomimetic approach is extremely simple, uses a finite state machine intuitive representation and may be expanded to cover different underwater tasks like oceanography and environmental monitoring, security commissions for surveilling territories of interest and interventions in underwater industry for fault detection. Several experimentation scenarios were conducted, successfully tested and verified the controller's functionality and robustness, including:

- coordinating the AUV team members,
- searching of a target,
- pursuing a target after it is detected and verified,
- the final phase of intervention or, using biomimetic terms, *feeding*.

Contemporary commercially available AUV models are designed to dive offshore [96], [21], a fact that leads to high experimentation cost in terms of money, time and human resources. Thus, to achieve low logistics underwater experimentation, *Ale III*, a compact and agile prototype AUV, was designed and constructed as an experimentation platform for the suggested control model. The design process was aided by simulation experimentation as a proof of concept for vessel's seaworthiness and agility [145]. It uses two lateral thrusters for surging and yawing. Its third thruster is suited at the vessel's bottom and adds heaving capability. This *differential* way of propelling, along with vessel's small dimensions, facilitated indoors underwater experimentation inside the premises of *Intelligent Systems and Robotics Laboratory* of the Technical University of Crete, during the last three years. The prototype AUV is equipped with an embedded vision sensor module capable of programming and implementing image processing algorithms. An inertial measurement unit complements vessel's perception of reality by estimating its three dimensional orientation. An on board computer provides adequate computational power for the compiling and execution of autonomous controller programs. PD and fuzzy PD underwater target following controllers were developed and tested [146]. *Ale III* was also used for the testing of an atomic controller version of the proposed behaviour model. The cooperative behaviour model was tested solely in simulation due to the experimentation area's dimensional limitations.





# Chapter 3

## Construction

This chapter describes the design solutions and the main characteristics of *Ale III*, a prototype AUV compact and agile enough for experimentation inside a small tank of 1m<sup>3</sup> [146]. Apart from the small dimensions and agility, the vehicle's design strategy aims at two additional main objectives: low construction and maintenance cost and use of open source software and hardware components. *Ale III* was designed and constructed, as part of the present dissertation, during the period 2011- 2012 at the *Intelligent Systems and Robotics Laboratory* of the Production Engineering and Management School of Technical University of Crete. The prototype was commissioned successfully inside a specially designed laboratory area for three years of experimentation and testing, with results described at Chapters 4 and 5.

### 3.1 Low logistics AUVs

As described in Section 1.3, fully functional AUVs are available as commercial products. In addition, several prototype AUVs have been developed around the world as a result of completed or ongoing academic research. Contemporary AUVs are covering a wide range of different designs and operational capabilities in oceanography, industry and military. One common characteristic among them is the fact that all the available vessels are designed to dive in the open sea, navigable lakes and rivers or at Olympic sized swimming pools [96], [21]. The dimensions magnitude of contemporary AUVs may be demonstrated by the size and the power demands of one of the smallest thrusters, available as 'off the shelf' component: 23×8cm and 125W respectively [5]. The product range of today's AUVs is useless for experimentation indoors, at a tank inside the premises of a typical research laboratory's facilities. Only some fish-like biomimetic AUVs [80], [121], [182], [186], [191] offer the opportunity of indoors operation, yet the limitations concerning their embedded computational power and energy storing, restrict their operational capabilities.

Apart from the experimentation area size, another critical issue is AUVs' logistics. The deployment of an AUV is quite simpler and its operation is more inexpensive than that of an ROV. Yet, the acquirement cost for a commercially available AUV model, as well as its operational cost, especially when the last includes the support of crewed surface vessels, remains high. In recent years there have been efforts to limit AUVs logistics to more affordable levels leading to a significant increase in the capability of small class AUVs with advances in navigation, propulsion, control and compact sonar payloads [9]. A range of oceanographic and geophysical survey instruments can be found on such AUVs, including bathymetric mapping sonars, side-scans and optical cameras, offering a satisfactory quality, covering current engineering and geophysics requirements in a wide range of survey tasks [82].

The overall development cost of a prototype or commercially available AUV may also be monitored by using open source software and hardware [34]. Lately a continuously growing family of hardware and software products are available under open source licenses with a considerably lower cost than the proprietary ones. Also, open source philosophy offers, apart from cost reduction, several other benefits to the research and development

procedure such as:

- fully customizable options,
- modularity along with the ability to use already available resources,
- in situ maintenance opportunity,
- capability for system's *diaphanous* modules integration.

Drawbacks like the lack of support and reliability are effectively controlled by the heterogeneous expertise and the extroversion found at the internet resources of the open source community.

## 3.2 Components of Ale III

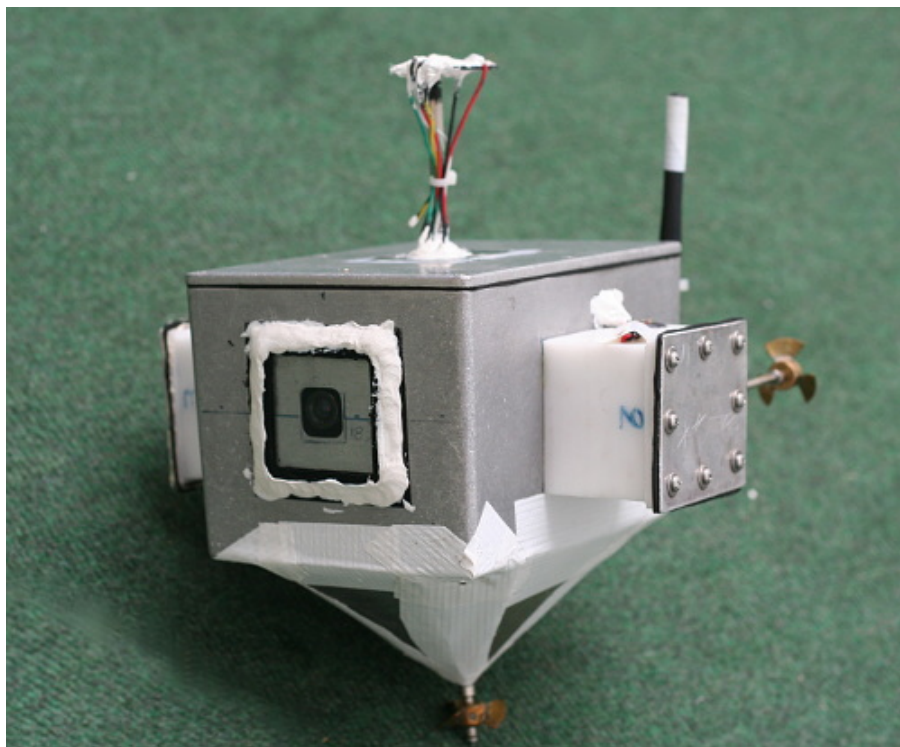


Figure 3.1: *Ale III* AUV, designed and constructed at the *Intelligent Systems and Robotics Laboratory* of the Technical University of Crete.

Ale III, as shown at the photos of Figures 3.2, 3.3, comprises the following parts:

1. water proof hull made of a *Deltron 480060* [52] die cast aluminium box rated at IP68 BS EN 60529:1992 [170] ingress protection standard with dimensions  $171.5 \times 120.6 \times 106.7$ cm. This solution was chosen for the following reasons:
  - Simplicity and 'off the shelf' availability. The only interventions made to the box were drilling one opening at hull's front face for the camera lens' vision, one at hull's bottom for the heave thruster propeller shaft and one at the lid for IMU's input- output cables, Figure 3.3. A piece of glass was attached to the lens' opening, Figure 3.2. All openings were sealed using *Sikaflex 291* [160] elastomeric sealant.

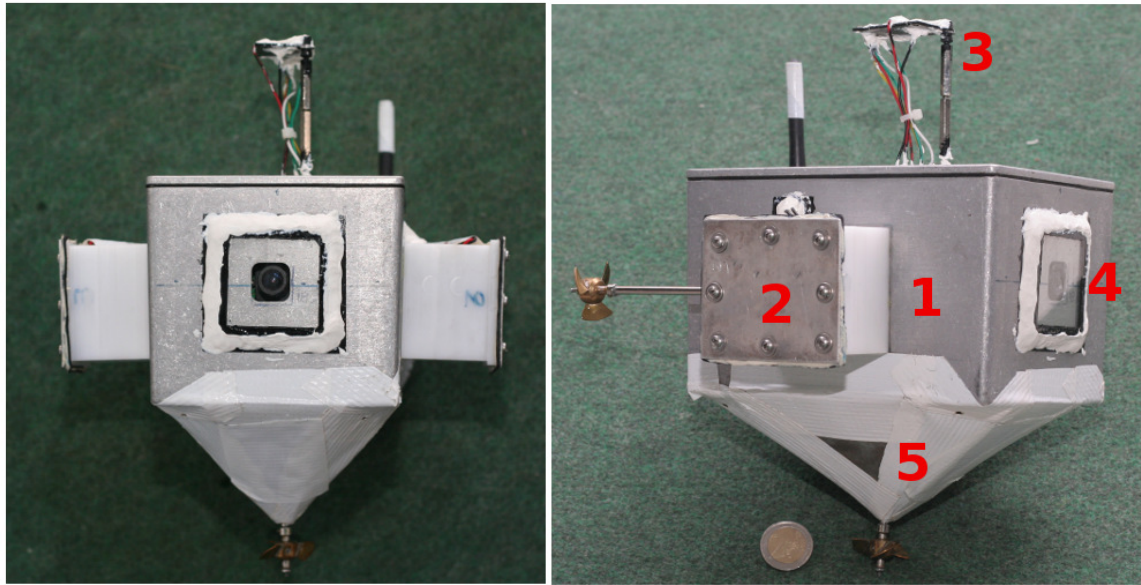


Figure 3.2: *Ale III* autonomous underwater vehicle main parts (1) water tight hull, (2) custom made thruster motor, (3) inertial measurement unit, (4) vision sensor module, (5) ballast unsealed chamber.

- Ease of access inside the enclosure. Detaching the lid is easy by unscrewing its six securing screws. Lid is sealed by a self adhesive, reusable hard neoprene rubber seal. Assembling and securing the box is easy since the lid has seal location lugs to aid seal installation and sealing. Once the lid has been detached there is access at the full area of the enclosure. This is a great advantage for the purposes of construction, maintenance and debugging, as proved throughout AUV construction and operation.

The vessel's box shaped hull contains vehicle electronics, a vision sensor module, the thruster motor drivers, the batteries and the bottom thruster motor. Photographs in Figure 3.4 show an alternative design initially studied and constructed but, finally, rejected. It was comprised of a cylindrical shaped hull made of PVC tube leading to a more streamlined installation of all the electronics inside the waterproof hull. This initial design was rejected due to problems pertaining to the ease of access inside the hull, the attachment of thrusters and the output of batteries' recharging cables. It should be noted that in the current version of the vehicle there was no study for hydrodynamic covers design. Such a work was beyond the scope of this dissertation. Also there were not conducted any pressure tests since *Ale III* was designed to operate in depths up to 3m. The hull, assembled with the heave thruster, weights 0.57kg.

2. Custom made thrusters, two lateral attached externally at the hull's starboard and portside and one placed vertically onto the hull's enclosure bottom. The two lateral thrusters offer two degrees of freedom, surging and yawing in a *differential-like* way. The bottom thruster adds one more for heaving. This propelling scheme was chosen for its agility and ability to operate in small areas, a fact of great importance for the prototype's design decisions. After a market survey on commercially available thruster models, during the AUV's design phase in 2011, the most compact thruster found had dimensions of 23×8cm and a power consumption of 125W [5]. Since these numbers were totally prohibitive, according to vessel's design aims and specifications, a custom made thruster was developed using components from radio controlled scale models. Each thruster uses a 6V, direct current *Graupner Multispeed 140* [77] brushed electric motor, equipped with a 2:1 gear unit. A 12cm shaft and a 4mm diameter stern tube were

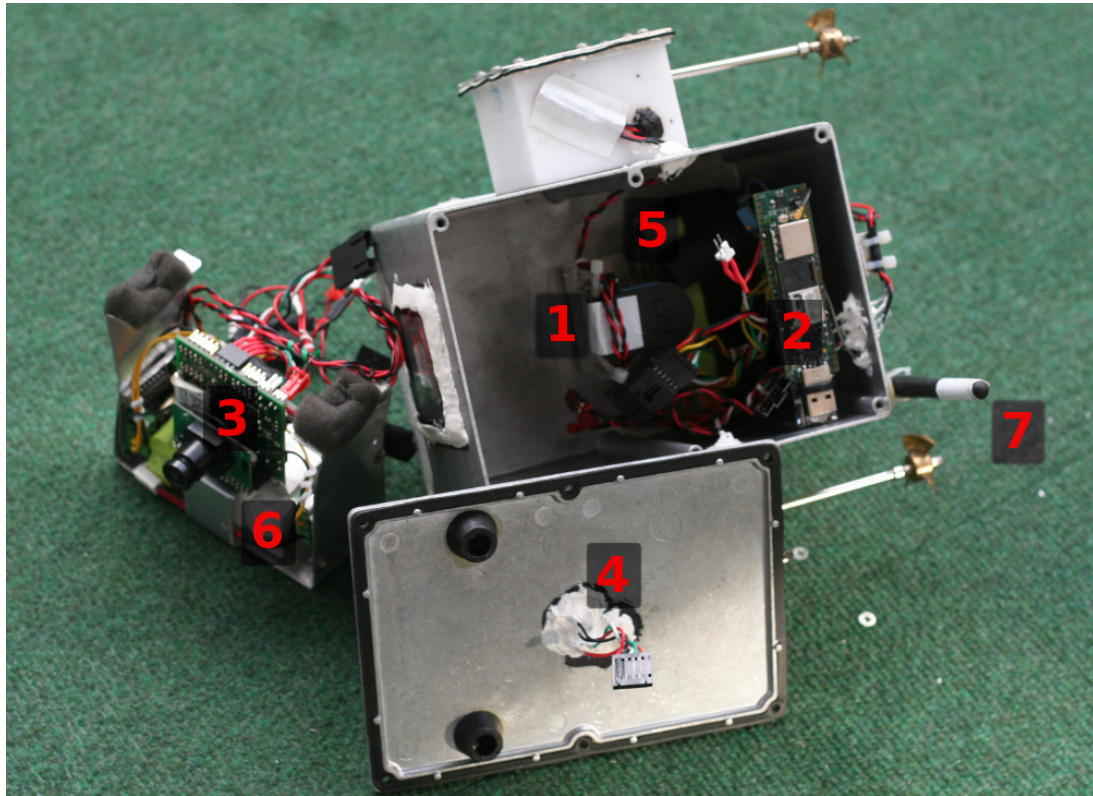


Figure 3.3: *Ale III* main parts inside hull's enclosure, (1) heave thruster placed inside the vehicle's hull, (2) processing unit and power supply circuits, (3) vision sensor module and thruster motor drivers, (4) lid, its seal and IMU's input- output cables, (5) electronics' batteries, (6) motor drivers' batteries, (7) wireless network adapter antenna. (3), (6) and (2), (5) comprise stem and stern assembly modules respectively.

properly assembled to the motor, Figure 3.5. The shaft rotates a 29mm diameter *Raboesch 108-33* [148] two way bow thruster propeller, selected among the vast variety of commercially available model-ships' propellers because of its symmetrical forwards- backwards design: it produces the same propelling thrust either it is rotating left or right. Several other propellers were tested and rejected because they could efficiently propel only when rotating in one direction. These *one way* propellers are not useful for *Ale III* driving- propelling since both turning directions are necessary for alternating forward and backwards in surging movement, diving and emerging in heaving movement and implement differential yaw turning. The ability of such manoeuvres increases vessels agility and facilitates its operation in small indoors laboratories. The two lateral thrusters are attached externally to the hull and so a custom made waterproof housing was designed and constructed for each one of them. Housing's enclosure was constructed by drilling and milling a piece of *ertalon*, a high mechanical strength, stiffness, hardness and toughness machinable nylon material with excellent wear resistance. The enclosure is secured and waterproofed by a 1.2mm aluminium lid and a rubber seal. A completed thruster wit shaft and propeller weights 0.19kg. During experimentation the propelling power of each thruster was estimated at 0.6Nt. Housing's blueprint is included in Appendix A.

3. Central processing unit, an Ångström Linux powered, ARM Cortex-A8 architecture *Gumstix Iron Storm* Computer- On- Module (COM), on a *Gumstix Thumb0* dual in-line style breakout board. Iron Storm



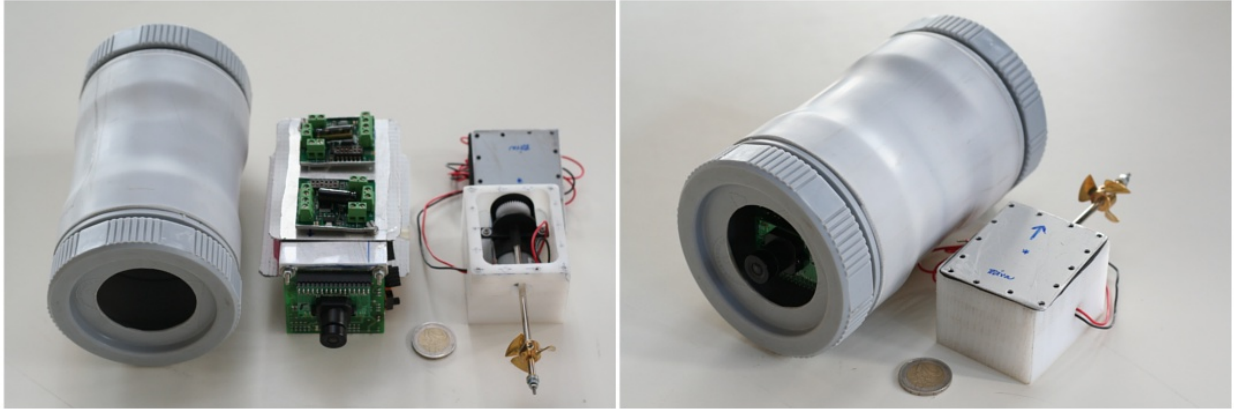


Figure 3.4: *Ale III* first version, featuring a cylindrical shaped hull and a streamlined positioning of the electronic devices and batteries.

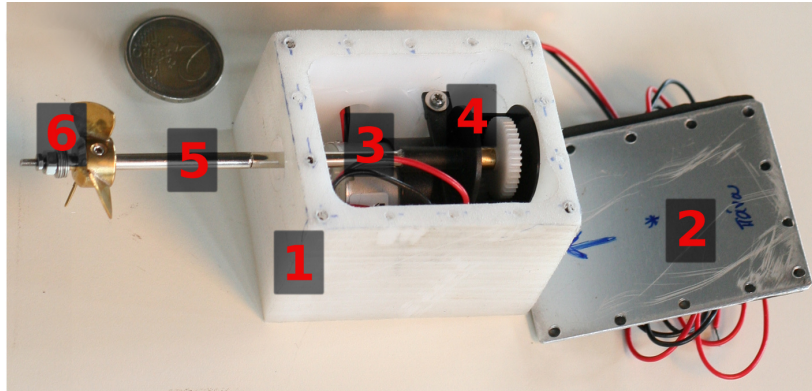


Figure 3.5: *Ale III* custom made thruster, (1) thruster housing, (2) lid and elastic seal, (3) electric motor, (4) gear unit, (5) shaft with stern tube, (6) two way propeller.

COM is a 5.6g and  $58 \times 17 \times 4.5$ mm computer, specifically designed for low power open source embedded applications [78]. It communicates, via breakout board's connectors, with AUV's sensors. It is, also, equipped with a wireless network adapter, enabling remote login for easier programming and debugging, as long as the antenna is not submerged in the water. Vessel's COM is part of the stern assembly, Figure 3.3.

4. Motor drivers, controlling electric motors' speed and rotation direction. *Sabertooth 2x5* dual 5A motor driver [53] from Dimension Engineering was chosen due to his compact dimensions of  $45 \times 40 \times 13$ mm, small weight of 19g, high efficiency and compatibility with the direct current brushed electric motors and the rest of vessel's electronic devices. *Ale III* uses two dual motor drivers, one for its two lateral and one for its bottom thruster. Both of them are part of the stern devices' assembly, Figure 3.3. Vessel's COM generates the appropriate Pulse Width Modulation (PWM) signals and controls both of the drivers.
5. Camera sensor module, supporting AUV's vision sensing, is a *CMUcam3* [43] open source, programmable, embedded colour vision sensor. The main reason for choosing CMUcam3 instead of a web camera is that image grabbing and processing is a task undertaken by the CMUcam3 module itself, leaving COM resources intact for the control process. It is equipped with a fast F2.8 aperture optical lens with  $56^\circ$

horizontal, 42° vertical and 70° diagonal field of view, proved to be adequate for AUV's small sized, low light experimentation area. For the *Ale III* project the camera sensor was adjusted to its low resolution option of 176×144 picture elements. Higher resolutions slowed down the controller execution as an increase at the CMUcam3 picture elements resulted in computational overhead during scanning the image sensor. Its dimensions are 57×55×30mm.

6. Inertial Measurement Unit, (IMU) estimating AUV's orientation in three dimensional space. It is a *Sparkfun Razor SEN-10736* 9 degrees of freedom unit, incorporating triple-axis gyro, triple-axis accelerometer and triple-axis magnetometer [4]. Sharing the same operation philosophy with the CMUcam3 vision module, it is an embedded microprocessor system with reprogrammable firmware. It serves each COMs request with a data packet containing the roll, pitch and yaw orientation of the vehicle. Its dimensions are 28×41×2mm. In this version of *ALE III* the IMU sensor is positioned on the top side of the vehicle, away from the interferences caused by the other electronic devices inside the enclosure of the hull box.
7. Logic level converters, to transform COM's input- output (I/O) 1.8V logic level signals to the levels found at the I/O of the rest electronic devices. *Sparkfun BOB-08745* [4], a two channel logic level converter is used for COM's connection to:
  - the 5V Transistor- Transistor Logic (TTL) level CMUcam3 camera sensor's I/O lines.
  - the 3.3V TTL level IMU sensor's I/O lines.
  - the 5V PWM motor drivers input control signals.

BOB-08745 is an ultra compact device of 15.2×12.7×2mm.

8. Power supply electronics, for transforming and regulating the electronics battery power supply to:
  - 1.8V, necessary for powering an inverter- buffer controlling the serial communication between IMU and COM. A *DE-SWADJ* 10W step down, adjustable, switching regulator from Dimension Engineering [53] is used for this purpose because of its high efficiency of 85%, small dimensions of and its low adjustable output starting at 1.25V. The 10W rating of DE-SWADJ regulator is more than enough for the low power buffer operation. It weights 5.5g and its dimensions are 37×15.5×14mm.
  - 3.9V, necessary for COM's more demanding power supply that can reach 0.5A if the wireless adapter is turned on. A *PTN78060WAH* 3A step down, adjustable, switching regulator from Texas Instruments [168] takes over this power supply adequately with an efficiency of 80%. It weights 3.9g and its dimensions are 25.2×15.75×9mm.

Switching, high efficiency, regulation assures that no excessive electronic heating will happen, even after long operation periods.

9. Energy storage units, inexpensive 2600mAh NiMH AA rechargeable battery packs, one for the supply of electronics at 12V and one for the electric motors at 7.2V. Electronics' power supply battery pack weights 300g, its size is 50×73×30mm and is a part of the stern assembly. Motors' power supply battery pack weights 180g, its size is 50×87×15mm and is part of the stem assembly, Figure 3.3. The two different energy sources assure that the electrical noise, inevitably produced by motors' operation, will not interfere with the rest electronics. Also possible instantaneous electric motors' peak power demands due to *heavy* loads will not affect the power supply to AUV's delicate devices like COM, IMU and CMUcam3. Both packs were positioned at the lowest part of the sealed enclosure, at hull's bottom, trying to achieve as low centre of gravity as possible.
10. Power switch and connector, an eight poles, waterproof *Bulgin 400 Buccaneer* [156] panel mount connector, rated at IP68 BS EN 60529:1992 [170] ingress protection standard, has been assembled at AUV's stern. Four of its poles are connected internally with the two battery packs and the rest four with the electronics' and motors' power supply lines, see Appendix B. When it mates externally with a specially

soldered *blind* cable connector the power supply passes through it, from the internal power supply packs to electronics and motors. Thus, the *blind* connector acts as power switch, turned on when in place and turned off when disconnected. Moreover, when stern's panel mount connector mates with the alternative cable connector sourcing external power, the AUV may operate for unlimited period and- or recharge its battery packs. This *trick* is extremely useful during experimentation and testing because it may supply uninterrupted power, for unlimited time period, without recharging intermissions.

11. Ballast, placed inside a wet external chamber, underneath the hull to compensate vehicle's buoyant force. A ballast of 0.5kg lead spheres was added at the lowest vessel's part, inside a pyramid shaped unsealed wet chamber, to accomplish a slightly positive buoyancy, Figure 3.2. Positive buoyancy is necessary for vehicle recovery reasons. Due to ballast and batteries positioning, vehicles centre of mass is located quite lower than its centre of buoyancy, maximizing the righting moment and offering passive roll and pitch stability, while diving underneath the water surface. The following experiment was repeatedly conducted to assure vessel's underwater stability:

- The vessel is manually positioned in an upside down position underwater.
- It is released to move freely.
- During the experimentation all thrusters are turned off.

During all stability experiments the AUV recovers from the upside down position to a stable near- zero roll and pitch orientation in just a few seconds. A typical stability test is shown in Figure 3.6.

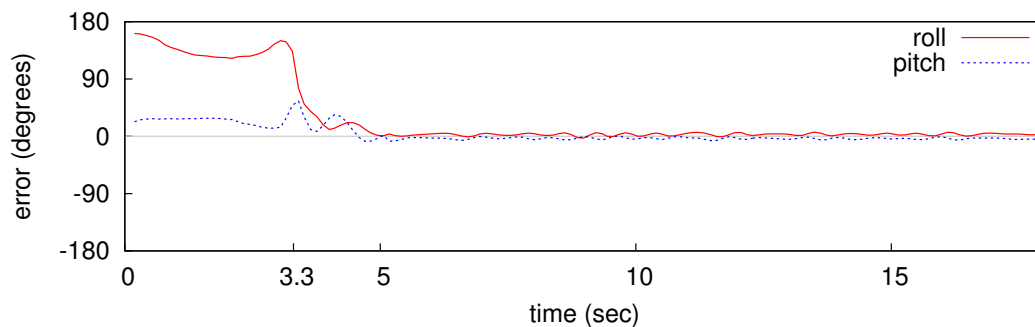


Figure 3.6: Experiment denoting vehicle's roll and pitch stability during diving. Initially the AUV is forced to take an upside down position with all the thrusters turned off. At 3.3 seconds it is released to move freely and it recovers to a stable, near-zero roll and pitch orientation after a few seconds

Appendix B contains the electronic devices interconnections schematics and presents system's hardware architecture.

### 3.2.1 Ale III autonomy

*Ale III* kinematics present three degrees of freedom, surging- yawing and heaving, in a *differential driving* fashion, Figure 3.7. *Ale III* propelling scheme has the same options like ground robots' differential steering, plus the heaving motion:

- when both lateral thrusters are propelling forward and rotating with same speed the vessel is moving straight forward,

Table 3.1: *Ale III* technical specifications

dimensions	17×21×25cm (L×W×H)
hull	aluminium box shaped
thrusters	3, custom made
sensors	vision sensor module inertial measurement unit
processing unit	Ångström Linux Gumstix COM
electronics power consumption	2.5W
thruster power consumption	2.2W at half speed
electronics battery	12V NiMH at 2.6Ah
thrusters battery	7.2V NiMH at 2.6Ah

- when both lateral thrusters are propelling backwards and rotating with same speed the vessel is moving straight backwards,
- when the lateral thrusters are rotating with different speeds and or different directions then the vessel is turning accordingly and
- when both lateral thrusters are rotating with the same speed but opposite directions then the vessel is rotating around its centre.

This propelling method increases vehicle's agility and manoeuvrability. It was chosen for *Ale III* due to the limitations in experimentation area's dimensions. Apart from agility this method offers kinematic control simplicity, aiding this way to the development of navigational algorithms. This was the reason that the experimentation was feasible indoors, inside the limited premises of an academic laboratory. Otherwise, if a commercially available torpedo shaped model was used to test the proposed control models, then underwater experimentation at the coastal line, with the support of surface ship and crew, would have been inevitable. The cost of a mission like this would have, also, been prohibitive.

*Ale III* sensory equipment supports two major perception tasks, typically supporting underwater autonomy: vision and three dimensional orientation capability. Vision is one of the most important vehicle features, facilitating several testing scenarios with coloured light targets. Inside the black opaque experimentation tank filled up with 1.5m<sup>3</sup> of clear water the visual noise levels were in general acceptable, vehicle's VSMP was operating reliably and thus the implementation of several testing scenarios was feasible. IMU heading and attitude continuous estimation is necessary for underwater three dimensional navigation and is a standard in AUVs' equipment. Vehicle's IMU along with vision perception produced sensory information, sufficient enough for the development of autonomous underwater behaviours.

*Ale III* overall dimensions are 17×21×25cm. Its total weight, including ballast, is 2.6kg. The electronics power consumption does not exceed the 2.5W while each thruster consumes 2.2W when spinning at half of the maximum speed underwater. *Ale III* autonomy is at least two hours as measured during experimentation under normal conditions. Using a different technology energy source, for example higher density Lithium Polymer batteries for the motors' greedy battery pack, will improve energy autonomy by far.

Due to its compact dimensions, propelling method and adequate energy autonomy *Ale III* manages to operate successfully inside the experimentation tank, covering the present dissertation's testing needs in full, during a period of three years.



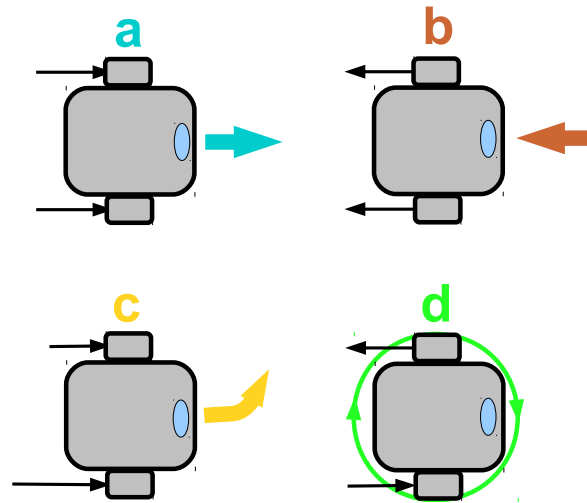


Figure 3.7: *Ale III* surging and yawing propelling model: (a) when both lateral thrusters are propelling forward and rotating with same speed the vessel is moving straight forward, (b) when both lateral thrusters are propelling backwards and rotating with same speed the vessel is moving straight backwards, (c) when the lateral thrusters are rotating with different speeds and or different directions then the vessel is turning accordingly and (d) when both lateral thrusters are rotating with the same speed but opposite directions then the vessel is rotating around its centre.

### 3.3 Log Book

*Ale III* was commissioned in depths up to 1m for more than 500 hours of operation in a period of 16 months. During this period there was no water leakage problems, neither at the hull compartment, nor inside the thrusters. The electronics and sensors were functioning normally without the need of any replacement part until now. The motors also proved their reliability as they were functioning normally over all this time without any maintenance at the electric motors or the rest of the drive train parts. During the experimentation, the power supply devices proved to be adequate and there was never observed electronics power loss or overheating.

### 3.4 Software Architecture

AUV's electronics hardware devices were carefully chosen to accomplish low cost, small dimensions, minimal power consumption, easy customization and open source availability. *Ale III* brains are the combination of a Gumstix COM and an autonomous controller program. This program is stored on board and starts executing right after COM finishes the start up process. There are two more programs incorporating with COM's controller:

- vision sensor module program (VSMP), including frame grabber image processing capabilities. VSMP is stored on board and executed by module's embedded microprocessor as reprogrammable firmware, enabling the implementation of custom frame grabbing and image processing algorithms [43]. At the current implementation VSMP detects colour targets, estimates their position relevant to the image sensor's centre and their size. Sensor's function is triggered and tuned, macroscopically, by COM's serial port communications. CMUcam3 serves each COM's request with a data packet containing the image processing results, leaving COM resources intact, devoted exclusively to the robotic controller process. VSMP is based on open source libraries developed for the CMUcam3 project [43].

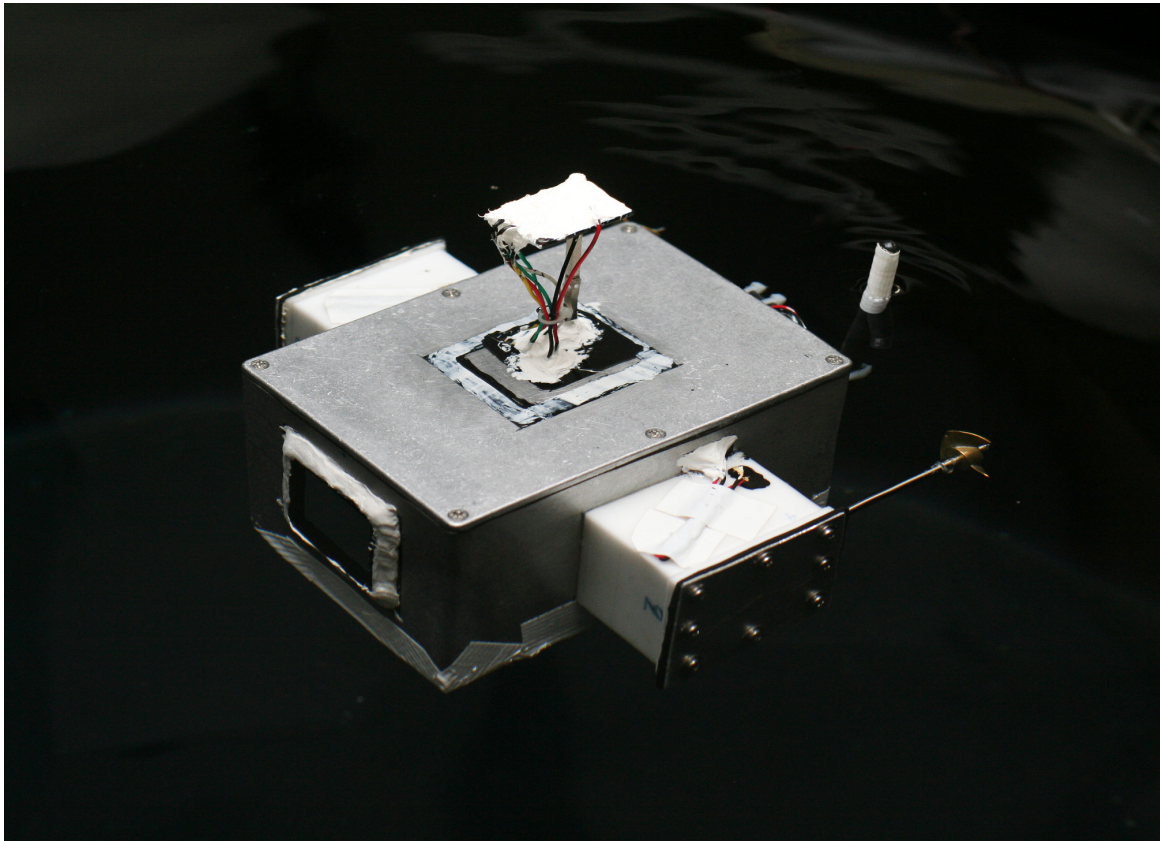


Figure 3.8: *Ale III* near the water surface.

- IMU module program (IMP), sharing the same operation philosophy with the vision module. IMP serves each COM's request with a data packet containing the roll, pitch and yaw orientation of the vehicle, using serial port bi- directional data transmission. COM is macroscopically manipulating IMPs functions to acquire vessel's three dimensional orientation, without consuming any of its computational resources. IMP is a slightly modified version of the open source *sf9domahrs* library [157], functioning as an Attitude Heading Reference System.

VSMP and IMP are embedded C application programs with access to sensors' hardware and communication resources. Open source resources and module- based libraries were used as part of their firmware program routines. COM's Autonomous Controller Software (ACoS) was developed using C++ programming language and object oriented environment. It comprises of two parts:

- the *Robot* class, that uses operating system modules to communicate with the system's input and output devices, i.e. the sensors and the motor drivers respectively. It accesses and manipulates the sensors readings and forwards the control signals to the motor drivers. *Robot* class is modular in the sense that for every new device connected to the system a new class member has to be implemented. The following open source libraries are used in *Robot* class, customized for ASoC's specialized needs:

**libserial 0.6.0rc1** for simplified C++ serial port programming and communication [120]. LibSerial provides a class collection that allows simplified serial ports access as with standard C++ *iostream* objects [99]. It is used for serial port, bi- directional intercommunication between ACoS- VSMP and ACoS- IMP.

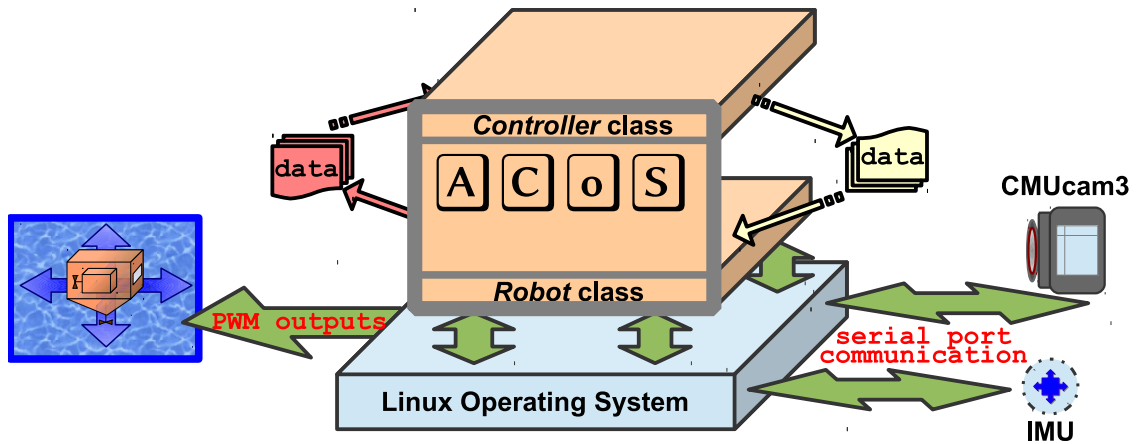


Figure 3.9: ACoS Software Architecture.

**gpio** which allows user- mode access to COM's General Purpose I/O pins (GPIO) [89]. ACoS manipulates a COM's GPIO line to isolate a serial port's receive input line from all electronic signals, during start up process, see Section B.2.

**omap3-pwm** , a driver manipulating PWM outputs from COM's user space environment [58]. *Omap3-pwm* driver is necessary for transmitting the ACoS commands to the motor drivers, via COM's PWM output lines, see Section B.3.

- the *Controller* class, encapsulating vehicle's behaviour. This class receives an input from the *Robot* class containing the sensor readings. It processes these data according to the ACoS algorithm and produces an output that, in turn, is fed back to the *Robot* class. As soon as *Robot* class receives controller's output, it produces the respective PWM signals to command the motor drivers.

Several behaviours may be developed as different members of the *Controller* class. In fact, this architecture permits several behaviours to be programmed and initialized, so that the system user may choose which one of them to use, depending on the circumstances. All these different behaviour objects of the *Controller* class use the same communication protocol to implement the input/output operations with the *Robot* class.

Software resources reused, modified or developed as part of ACoS and ACoS itself are valid through open source licenses.



# Chapter 4

## Testing Controllers on Ale III

This chapter emphasizes on the experimental modularity of software architecture and in particular of the *Controller class*. As a result of the *open* ACoS architecture and COM's efficient computational resources, different kinds of robotic controller may be accommodated on board. ACoS architecture enables, if not encourages, the development of novel control algorithms by the means of:

**Designing and editing** algorithms' source code on board, since ACoS supports wireless remote console communication while the vessel is at surface. This is a great tool, not only for developing algorithms but also for debugging controller programs.

**Use already developed software resources** as ACoS libraries. This feature enables access to an overgrowing number of software modules found in the open source C and C++ repositories, decreasing by far the overall controller development time. It also increases reliability, since repositories' libraries have already been tested, debugged and documented by the worldwide open source community. ACoS is by default fully compatible with software modules implementing methods, typically used in robotics, like fuzzy logic [149], [72], artificial neural networks [138], [50], genetic algorithms [73], [74] and dynamic programming [147], [67].

**Testing alternative algorithms** implemented as distinct *Controller* classes, without the overhead of programming Application Protocol Interfaces (API) for accessing vessel's devices. *Robot* class is ACoS standard API, offering full access to system's hardware resources, at least as long as electronics configuration is the same.

As a proof of concept for *Ale III* seaworthiness and functionality, two different controllers, a classic Proportional Derivative (PD) and a fuzzy PD were successfully implemented and tested. The experimentation took place in an indoors laboratory area, specially designed for *Ale III*.

### 4.1 Experimentation Area

AUV's small dimensions offer the potential for indoors experimentation. As shown in Figures 4.1, 4.2 a plastic, opaque, cylindrical tank, 1.3×0.8m, diameter×height, was established inside the premises of Intelligent Systems & Robotics Laboratory at the School of Production Engineering and Management. The tank has the proper drainage allowing easy control of the water surface level and maintenance. This experimentation area was used extensively, not only for testing and verifying the AUV seaworthiness during the several design and construction levels, but also for the various ASoC versions development. More specifically, the following tasks were conducted inside the experimentation area facility, proving its adequacy:

- Thrusters and hull design evaluation, water proof tests, AUV balance and stability verification, energy efficiency measurements and ballast estimation.



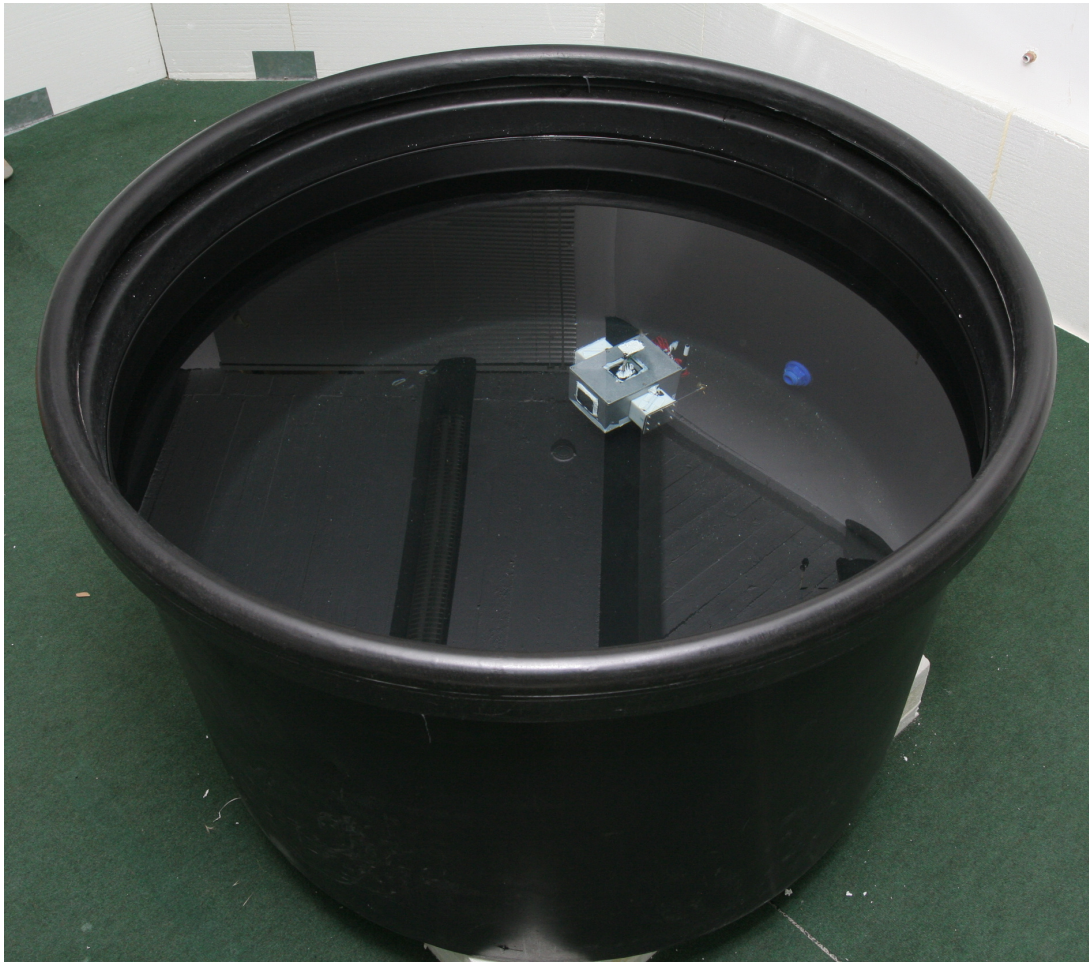


Figure 4.1: *Ale III* operating at the experimentation area inside the laboratory.

- Sensors calibration, sensors firmware development.
- Controllers evaluation and comparison. Development of ACoS versions for yaw oriented and target-following controllers. Development of a biomimetic behaviour based ACoS.

## 4.2 PD Controller Development and Testing

The development of a PD controller, achieving and maintaining a given yaw value, was chosen as a first proof of concept for ALE III project. This task uses a *Controller* class member, dedicated to the task corresponding behaviour and a *Robot* class member for the IMU and thrusters utilization. Every, approximately, 0.04sec *Robot* class communicates with IMP, accesses and forwards IMU's yaw readings to the *Controller* class. After the control process taken under by *Controller*, the *Robot* class receives the results to drive the thrusters. The proportional control factor is given by:

$$O_p = K_{p_{yaw}} * (\phi_e / \phi_{max}) * P_{max} \quad (4.1)$$

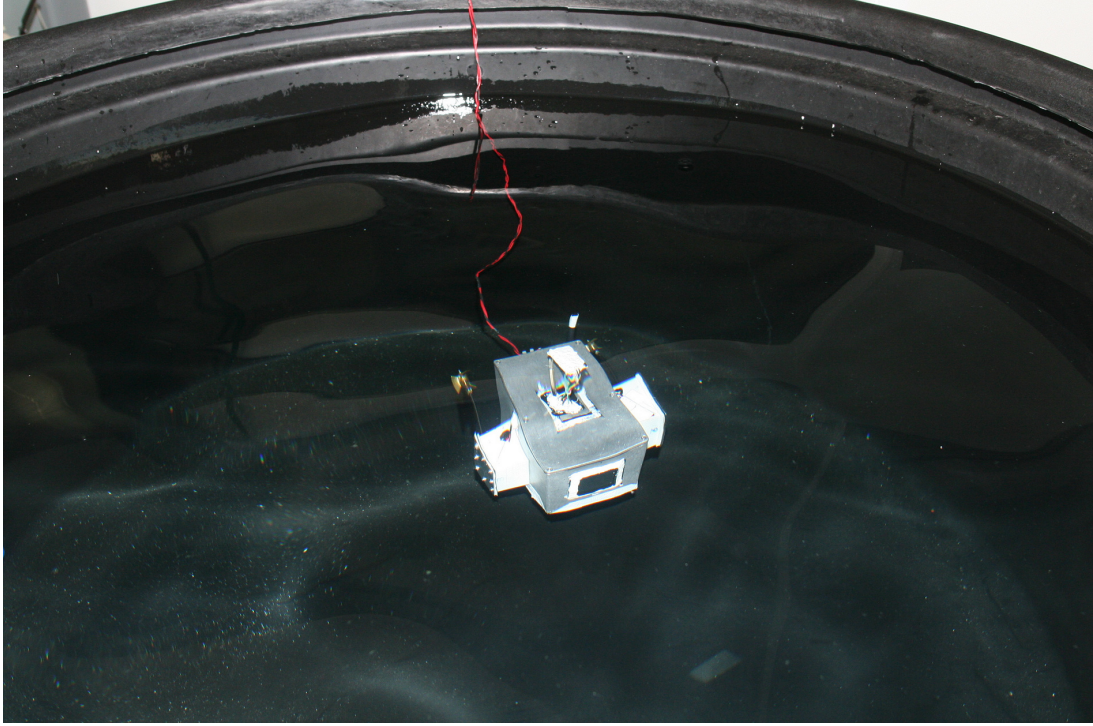


Figure 4.2: The experimentation area inside the laboratory. *Ale III* operating underwater inside the experimentation area. It uses the external power supply for uninterrupted operation and testing.

where  $O_p$  and  $Kp_{yaw}$  are the output and gain respectively,  $\phi_e$  is the yaw heading error,  $\phi_{max}$  its maximum positive value and  $P_{max}$  is thrusters max turning speed. The derivative factor is:

$$O_d = Kd_{yaw} * d\phi_e * P_{max} \quad (4.2)$$

where  $d\phi_e$  is the yaw heading error derivative,  $O_d$  and  $Kd_{yaw}$  are the D controller output and gain respectively. The controller gain values were estimated using experiments, firstly for the proportional one and afterwards for the derivative gain. Figure 4.3 presents the results of an experiment that encapsulates the controller efficiency for different parameter values: the vehicle is underwater, with a yaw heading of approximately  $0^\circ$  when, at zero time, it has to rotate at  $-115^\circ$  and after that, it has to change its yaw orientation for  $90^\circ$  every 12 seconds. The experiment tests controller reflexes after each yaw value reset and verifies controller stability throughout the interval between two subsequent changes to the target value. Also it verifies IMU functionality, ACoS data I/O interchange rates adequacy and intercommunication reliability. During testing the developed controller presented predictable behaviour:

- Increasing  $Kp_{yaw}$  results in a more oscillating behaviour around the target value, top plot of Figure 4.3.
- Increasing  $Kd_{yaw}$  results in a less oscillating behaviour around the target value, middle plot of Figure 4.3.
- Decreasing  $P_{max}$  decreases the oscillating behaviour around the target value and results in a more delayed approach near it, bottom plot of Figure 4.3.

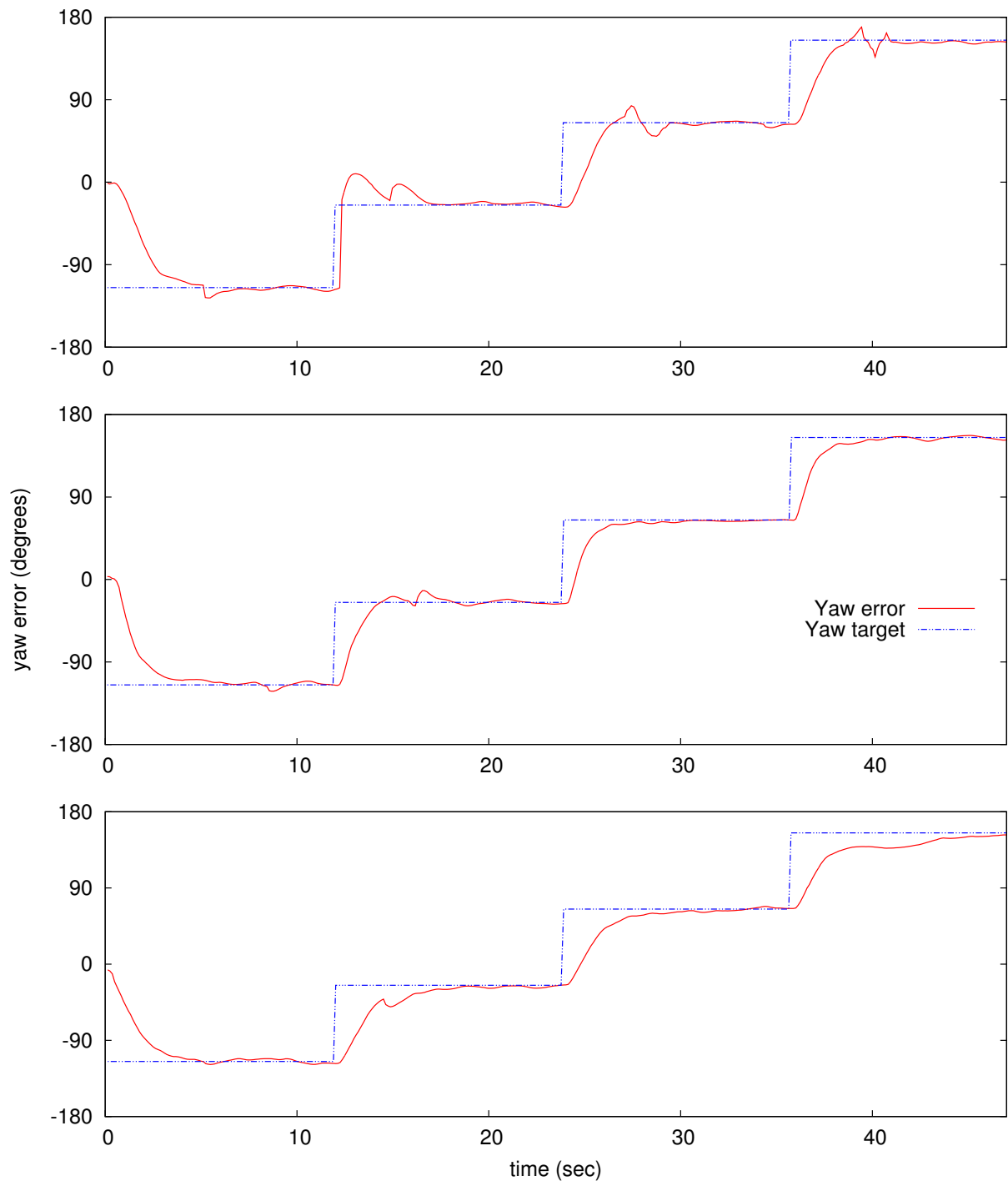


Figure 4.3: Yaw PD Controller experiment results for different gain parameters values. The vehicle is underwater, with a yaw heading of approximately  $0^\circ$ . At 0 seconds, the PD controller takes over and it has to change its yaw heading to  $-115^\circ$ . Subsequently, it has to change its yaw orientation for  $90^\circ$  every 12 seconds. The plot at the top shows the results for  $Kp_{yaw} = 1.2$ ,  $Kd_{yaw} = 0.003$ ,  $P_{max} = 1500$ , the plot at the middle for  $Kp_{yaw} = 1.2$ ,  $Kd_{yaw} = 0.005$ ,  $P_{max} = 1500$  and the plot at the bottom for  $Kp_{yaw} = 0.8$ ,  $Kd_{yaw} = 0.005$ ,  $P_{max} = 1000$ .



## 4.3 Target Following

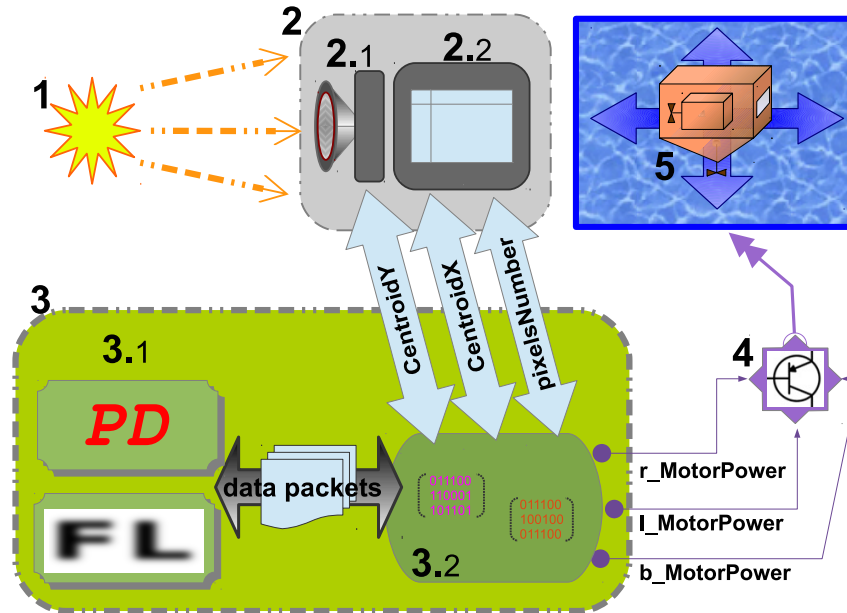


Figure 4.4: System's configuration for the target following scenario: (1) Light Target, (2) Vision Sensor Module, (2.1) Image Sensor, (2.2) VSMP Frame Grabber and Image Manipulation Program, (3) ACoS Engine executing at the memory of Computer On Module, (3.1) Proportional Derivative and Fuzzy Logic alternate *Controller* classes, 3.2 *Robot* class, (4) Motors' Drivers Module, (5) AUV's thruster motors.

The vehicle's second control test is divided in two parts. In the first part *Ale III* uses a PD controller designed in a similar way like the one presented in the previous section. The second part involves the application of a fuzzy logic controller. Both controllers are implemented as two versions of the ACoS that repeatedly manipulate the motors' turning speed, using information collected from the camera sensor, Figure 4.4. More specifically, each controller's tasks involve:

- detecting** a white light target moving freely inside the tank of the experimentation area using vehicle's vision sensor and VSMP light and colour detection capabilities,
- estimating** the detected target's relative orientation in respect to vehicle's position using VSMP tracking capabilities and target's trace on sensor's pixels map,
- approximating** the target's distance from the vehicle by measuring its trace's surface on sensor's pixel map,
- deciding** what will be vehicle's next move according to the target following algorithm rules,
- interpreting** the next move's decision as rotating speed and direction commands for the three thrusters.

This target following task demonstrates the ACoS autonomous capabilities and tests the vessel's overall dynamics. Following the arbitrary three dimensional target course indicates the vessel's seaworthiness, thruster efficiency and underwater stability. When the target stands still vessel's *hovering* capabilities are tested. Low target speeds and steady paths are used for continuously evaluating the ACoS stability and reliability while larger speed levels, sudden target accelerations and steep course changes evaluate its robustness and the vessel's kinematics model efficiency. Finally, the underwater target following scenario tests the vision sensor's functionality, the ACoS- VSMP data I/O interchange rates adequacy and intercommunication reliability.

Figure 4.4 presents the data and signals exchange between the ACoS and system devices, i.e. the vision sensor module, the COM and the motor drivers module. Both alternate controllers repeat approximately 5 control cycles per second. Every one of these cycles consists of the following parts:

1. The *Robot* class program part accesses the vision sensor module's results *CentroidX*, *CentroidY* and *pixelsNumber*. The module computes the former values using the custom firmware VSMP, manipulating the white light target pixels scanned by the camera sensor. *CentroidX* and *CentroidY* values depict target's central point, the so called *centroid* abscissa and ordinate. These values are estimated as the sum of the tracked pixels' abscissas and ordinates, divided by their total number. The centroid point is an indication of the target's position relative to vehicle. *pixelsNumber* is the total number of pixels detected. It is an indication of the distance between vehicle and target.
2. The *Controller* class program part computes the derivatives of *CentroidX* and *CentroidY*.
3. The controller software engine, inside *Controller* class, computes the motor control variables *leftMotorPower*, *rightMotorPower* and *bottomMotorPower* and passes their values to the *Robot* class. These variables correspond to the thrusters rotating speed.
4. The *Robot* class, using COM resources, forwards the output control values to the motors' driver unit. The motor drivers actuate the thrusters and move the vehicle.

### 4.3.1 Target following PD controller

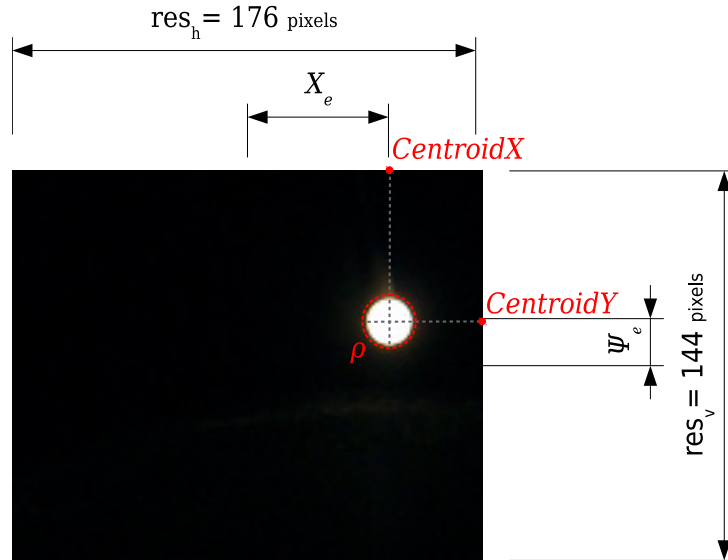


Figure 4.5: CMUcam3 snapshot during target following.

The PD target following controller output is synthesized by the yaw (4.3a) , heave (4.3b) and surge (4.3c)

coordinates,

$$O_{yaw} = Kp_{yaw} * \chi_e * P_{surgeMax} + Kd_{yaw} * d\chi_e * P_{surgeMax} \quad (4.3a)$$

$$O_{heave} = Kp_{heave} * \psi_e * P_{heaveMax} + Kd_{heave} * d\psi_e * P_{heaveMax} \quad (4.3b)$$

$$O_{surge} = Kp_{surge} * (\rho - \rho_{\oplus}) * P_{surgeMax} + Kd_{surge} * d\rho * P_{surgeMax} \quad (4.3c)$$

where

$\chi_e$  is the *CentroidX* error, that is, the target's centroid abscissa deviation from the camera sensor frame's  $x$  axis centre,  $CentroidX - res_h/2$ , where  $res_h$  is the camera's sensor horizontal resolution,

$\psi_e$  is the *CentroidY* error, that is, the target's centroid ordinate deviation from the camera sensor frame's  $y$  axis centre,  $CentroidY - res_v/2$ , where  $res_v$  is the camera's sensor vertical resolution,

$d\chi_e$  is the derivative of  $\chi_e$ ,

$d\psi_e$  is the derivative of  $\psi_e$ ,

$\rho$  is the *pixelsNumber* variable, used by the controller to monitor vehicle's distance from target,

$\rho_{\oplus}$  is  $\rho$ 's target value,

$\rho_{\sqcup}$  is a tolerance margin value depicting  $\rho$ 's acceptable range around the  $\rho_{\oplus}$  target value,

$P_{surgeMax}$ ,  $P_{heaveMax}$  are lateral and bottom thrusters maximum rotating speed,

$Kp_{yaw}$ ,  $Kp_{heave}$ ,  $Kp_{surge}$ ,  $-Kd_{yaw}$ ,  $Kd_{heave}$ ,  $Kd_{surge}$  are the proportional and derivative gains respectively of yaw, heave and surge. Their values were estimated using recurrent experimentation. At first, the proportional gains were figured out using uncoupled motion commands, for example solely for surge movement. Afterwards, the corresponding derivative gain was added to the uncoupled motion command and tuned in with respect to the proportional one. Finally the all the gains were fine tuned using motion commands for coupled surge, heave and yaw turning movements.

Figure 4.5 shows the CMUcam3 sensor image parameters for the target following task described above. The quantity  $d\chi_e$  is related to the submarine's yaw rotating speed and target's lateral movement. If the vehicle is kept stationary, a  $d\chi_e$  value of 200 is equivalent to a left heading target course, causing an increase of 25 pixels to the target's centroid abscissa at the camera sensor frame, during one controller time step of approximately 0.13 seconds. The quantity  $d\psi_e$  is related to the submarine's heave speed and target's vertical movement. If the vehicle is kept stationary, a value of 160 is equivalent to an upwards heading target course, causing an increase of 21 pixels (in analogy to the previous paragraph's higher resolution  $x$  axis movement of 25 pixels) to the target's centroid ordinate at the camera sensor frame. The control process aims at keeping  $\rho$  value as close as possible to  $\rho_{\oplus}$  target value determined by the vehicle user, according to the light target and experimentation area characteristics. Distance PD control is activated whenever  $\rho$  value is outside the range of  $[\rho_{\oplus} - \rho_{\sqcup}, \rho_{\oplus} + \rho_{\sqcup}]$ . When  $\rho_{\oplus}$  is 600 pixels then the submarine is approximately 30cm away from the spherical, 4cm diameter, light target used for the tests.

The target following PD controller was tested inside the experimentation area using a scenario where the white light target was departing away from the vehicle, following a sinusoidal - up/down course. The vehicle managed to follow the target without significant errors that could lead to target disappearance from camera's field of view, Figure 4.6. The third, fourth and fifth plots from top show *leftMotorPower*, *rightMotorPower* and *bottomMotorPower* controller outputs for left, right and bottom thruster propelling power accordingly, where:

$$leftMotorPower = O_{yaw} + O_{surge} \quad (4.4a)$$

$$rightMotorPower = O_{yaw} - O_{surge} \quad (4.4b)$$

$$bottomMotorPower = O_{heave} \quad (4.4c)$$

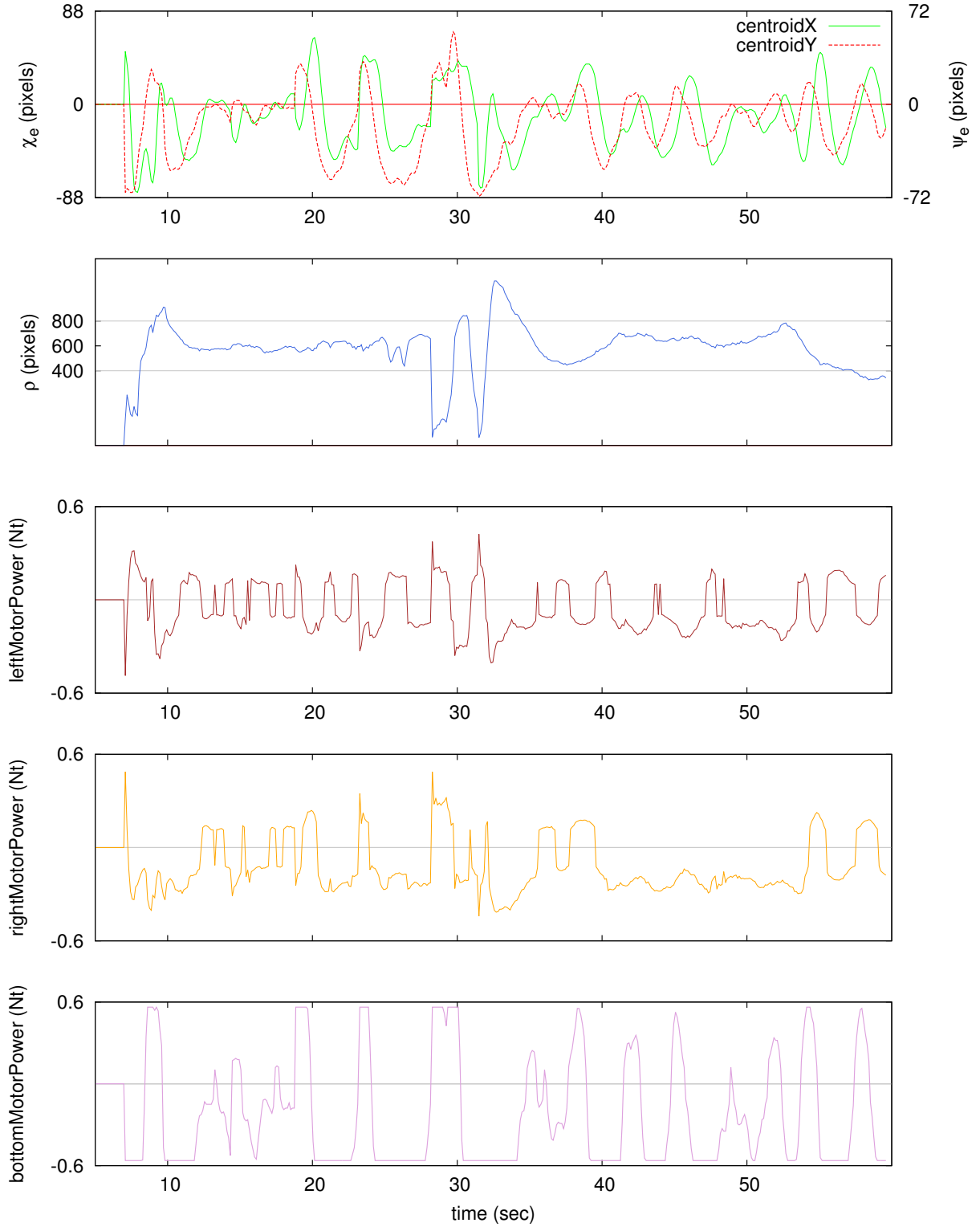


Figure 4.6: ‘Light target following’ Proportional Derivative controller experimental results, where the first plot from top shows target’s centroid  $d\chi_e$  and  $\psi_e$  errors. The second plot shows the  $\rho$  value when  $\rho_{\oplus} = 600$  and  $\rho_{\sqcup} = 200$ . Next three plots show the controller outputs for *leftMotorPower*, *rightMotorPower* and *bottomMotorPower* thruster power controller output variables accordingly.

At this experiment  $\rho_{\oplus}$  and  $\rho_{\sqcup}$  values are 600 and 200 pixels respectively. Distance control is activated whenever *pixelsNumber* value is outside the acceptable range of [400, 800]. The maximum errors are  $\max |\chi_e| = 73$ ,  $\max |\psi_e| = 71$  and their mean values are  $\text{mean} |\chi_e| = 23.2$ ,  $\text{mean} |\psi_e| = 19.8$ . Experimental area's small dimensions result in target movements to a nearby proximity with the vessel, causing inevitable steep changes at  $\chi_e$  and  $\psi_e$  values, that in turn burden controller task and increase  $\text{mean} |\chi_e|$  and  $\text{mean} |\psi_e|$ . An increase at  $\rho_{\oplus}$  value, though, would require a respectively larger underwater area for controller testing and evaluation.

### 4.3.2 Fuzzy Logic target following controller

As an alternative to the PD control, a Fuzzy Logic (FL), 'light target following' controller was developed. In this case a FL ACoS *Controller* class was programmed using *FuzzyLite v1.5*, an open source multi platform Fuzzy Logic Control Library, also written in C++ [149]. The choice of FuzzyLite library was based on the fact that its advantages are fitting well with *Ale III* project's specifications: it is a very simple, compact but easy to extend, object-oriented and *light-weight* fuzzy logic programming framework. It requires no additional libraries other than the Standard Template Library included in the C++ Standard Library and guarantees high speed performance. The FL controller uses the same input and output variables with the PD controller but fuzzifies their values according to the membership functions of Figure 4.7, converting their crisp values to the following fuzzy logic linguistic variables.

- Input Variables

- $\text{Centroid}X_{fz}$ , fuzzifies *CentroidX* variable with range [0, 176], since the horizontal camera resolution is 176.
- $\text{Centroid}Y_{fz}$ , fuzzifies *CentroidY* variable with range [0, 144], since the vertical camera resolution is 144.
- $d\text{Centroid}X_{fz}$ , fuzzifies the *CentroidX* derivative.
- $d\text{Centroid}Y_{fz}$ , fuzzifies the *CentroidY* derivative.
- $\text{pixelsNumber}_{fz}$ , fuzzifies *pixelsNumber*.

- Output Variables

- $\text{left-right-bottomMotorPower}_{fz}$ , fuzzify *left-right-bottomMotorPower* thrusters power values.

Triangular symmetric membership functions were chosen to maximize the fuzzy logic library computation performance [54], [173]. Other kind of functions, as Gaussian for example, led to slightly longer controller step periods due to their increased computational demands compared with the triangular symmetric functions. Recurrent experimentation showed that five membership functions were enough for the quantization of the variables' ranges. Adding more functions had no positive influence to the controller's behaviour, while using four or less functions led to unstable overshooting behaviour. Mamdani type fuzzy rules with centre of gravity defuzzifier produce the controller output. The fuzzy rules database contains rules associating:

- $\text{Centroid}X_{fz}$  and  $d\text{Centroid}X_{fz}$  antecedent inputs with the consequent outputs of  $\text{leftMotorPower}_{fz}$  and  $\text{rightMotorPower}_{fz}$ , Table 4.1. Each table element comprises of two symbols: the first refers to  $\text{leftMotorPower}_{fz}$  and the second to  $\text{rightMotorPower}_{fz}$ . Table 4.1 symbols  $\uparrow, \uparrow, \star, \downarrow, \downarrow$  denote propelling force of *full ahead*, *ahead*, *stop*, *astern*, *full astern* respectively. Linguistic variables symbols for  $\text{Centroid}X_{fz}$  are vR for *very right*, R for *right*, C for *centre*, L for *left* and vL for *very left*. For  $d\text{Centroid}X_{fz}$  are qtR for *quick turn right*, tR for *turn right*, noT for *no turn*, tL for *turn left* and qtL for *quick turn left*.

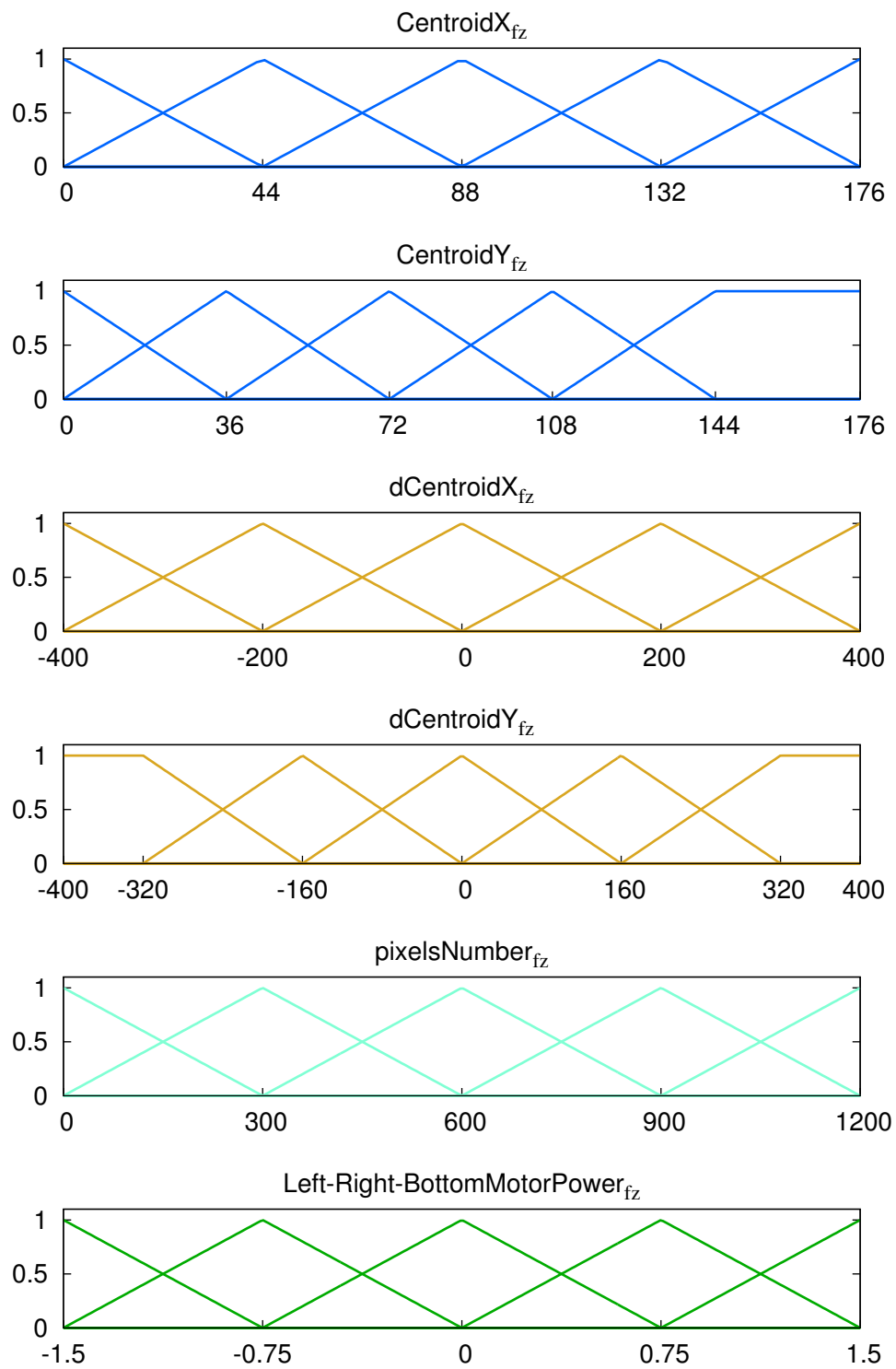


Figure 4.7: FL Target Following Controller linguistic variables membership functions.

Table 4.1: Rules associating  $CentroidX_{fz}$  and  $dCentroidX_{fz}$  antecedent inputs with  $leftMotorPower_{fz}$  and  $rightMotorPower_{fz}$  consequent outputs.

		$CentroidX_{fz}$				
		vR	R	C	L	vL
$dCentroidX_{fz}$	qtR	$\uparrow\downarrow$	$\downarrow\uparrow$	$\downarrow\uparrow$	$\downarrow\uparrow$	$\downarrow\uparrow$
	tR	$\uparrow\downarrow$	$\star\star$	$\downarrow\uparrow$	$\downarrow\uparrow$	$\downarrow\uparrow$
	noT	$\uparrow\downarrow$	$\uparrow\downarrow$	$\star\star$	$\downarrow\uparrow$	$\downarrow\uparrow$
	tL	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\star\star$	$\uparrow\downarrow$
	qtL	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$
		$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$

Table 4.2: Rules associating  $CentroidY_{fz}$  and  $dCentroidY_{fz}$  antecedent inputs with  $bottomMotorPower_{fz}$  consequent output

		$CentroidY_{fz}$				
		vU	U	C	D	vD
$dCentroidY_{fz}$	qhU	$\uparrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
	hU	$\uparrow$	$\uparrow_{0.3}$	$\downarrow$	$\downarrow$	$\downarrow$
	noH	$\uparrow$	$\uparrow$	$\star$	$\downarrow$	$\downarrow$
	hD	$\uparrow$	$\uparrow$	$\uparrow$	$\downarrow_{0.3}$	$\downarrow$
	qhD	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\downarrow$
		$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\downarrow$

- $CentroidY_{fz}$  and  $dCentroidY_{fz}$  antecedent inputs with  $bottomMotorPower_{fz}$  consequent output, Table 4.2. Table 4.2 elements' symbols have the same meaning as in Table 4.1. The subscript number accompanying a table element symbol denotes fuzzy rule's firing strength. Linguistic variables symbols for  $CentroidY_{fz}$  are vU for very up, U for up, C for centre, D for down and vD for very down. For  $dCentroidY_{fz}$  the symbols are qhU for quick heave up, hU for heave up, noH for no heave, hD for heave down and qhD for quick heave down.
- $pixelsNumber_{fz}$  antecedent input with  $leftMotorPower_{fz}$  and  $rightMotorPower_{fz}$  consequent outputs, Table 4.3. Table 4.3 elements' symbols have the same meaning as in Table 4.1. Linguistic variables symbols for  $pixelsNumber_{fz}$  are vS for very small, S for small, T for target value, B for big and vB for very big. Distance control is activated whenever  $pixelsNumber$  variable's crisp value is outside the

Table 4.3: Rules associating  $pixelsNumber_{fz}$  Antecedent Input with  $leftMotorPower_{fz}$  and  $rightMotorPower_{fz}$  Consequent Outputs.

		$pixelsNumber_{fz}$				
		vS	S	T	B	vB
$l/rightMotorPower_{fz}$		$\uparrow\uparrow$	$\uparrow_{0.3}\uparrow_{0.3}$	$\odot$	$\downarrow\downarrow$	$\downarrow_{0.3}\downarrow_{0.3}$

acceptable range of  $[\rho_{\oplus} - \rho_{\sqcup}, \rho_{\oplus} - \rho_{\sqcup}]$ , as denoted in subsection 4.3.1.

The FL controller was tested with the same scenario as the PD one, producing analogous results shown in Figure 4.8. The first two figure's plots show controller input values. These values are fuzzified to the antecedent inputs  $CentroidX_{fz}$ ,  $dCentroidX_{fz}$  and  $pixelsNumber_{fz}$ . Next three plots show FL controller's defuzzified values for the consequent output variables  $leftMotorPower_{fz}$ ,  $rightMotorPower_{fz}$  and  $bottomMotorPower_{fz}$ . During experimentation the vehicle successfully follows the light target's course without overshooting and significant deviations. The FL control procedure is deemed adequate for achieving this kind of behaviour, requiring yaw, heave and surge control. In the experiment described in Figure 4.8, the maximum errors were  $\max |\chi_e| = 83$ ,  $\max |\psi_e| = 65$  and their mean values are  $\text{mean} |\chi_e| = 29.4$ ,  $\text{mean} |\psi_e| = 21.8$ . The maximum roll and pitch errors are  $\max |roll_e| = 18^\circ$ ,  $\max |pitch_e| = 8.3^\circ$  and their mean values are  $\text{mean} |roll_e| = 0,82^\circ$ ,  $\text{mean} |pitch_e| = -4.18^\circ$  respectively. The main difference between the two controllers appears to be a smoother behaviour for the fuzzy one, throughout the diving tests inside the experimentation area. The PD controller, though, had better performance during experiments including sudden light target course changes.

## 4.4 Vessel's Stability and Seaworthiness

As stated in Section 3.2 *Ale III* has quite lower centre of gravity in comparison with its centre of buoyancy, because of the positioning of heavy objects, like the ballast and the two battery packs, as low as possible. This fact offers a passive stability around the horizontal position and immediate roll and pitch correction as a result of vessel's righting moments.

Apart from Figure's 3.6 test, the vessel's underwater stability is indicated by the last plot of Figure 4.8, showing the roll and pitch orientations during the FL controller testing: the deviation from the roll and pitch balance position is negligible throughout the diving experiment. In this plot, the roll value has a continuous deviation from the horizontal level of approximately  $-5^\circ$ . This is not due to the real vessels inclination but due to IMU sensor accumulated *drifting* error, a common issue for this kind of sensors.

During the underwater experimentation with FL and PD controllers the vessel's seaworthiness was verified under conditions of *hovering* and coupled yawing- heaving and surging movements. Controllers' commands, incorporating vessel's passive roll and pitch stability, adequately responded to target's three dimensional course. During all testing scenarios *Ale III* AUV was able to maintain a steady positioning, while its light target was standing still. Also, vessel's propelling method successfully encountered hydrodynamic drag and added mass effects, enabling thus its three dimensional target following navigation without noticeable hysteresis or overhead phenomena. Vehicle's underwater manoeuvres were fully controllable and predictable without oscillations or inclinations, not only under *hovering* conditions but also when following a nimble three dimensional underwater course.

## 4.5 Conclusions

*Ale III* a prototype low cost AUV, sharing the open source philosophy, capable of indoors underwater operation, experimentation and testing was developed. Different versions of ACoS emerged as a proof of concept for vehicle's operational potentials. The proposed ACoS architecture simplifies the design and programming of control processes. PD and FL controllers were easily evolved and proved to be adequate inside the custom made experimentation area. AUV's sensors proved to be reliable underwater. Their embedded design was fully compatible with COM's communication protocols and supports high performance computing. System crashes were extremely rare, even after several hours of experimentation and testing.

*Ale III* ACoS demonstrated its potential for implementing efficient controllers and act as basis for the development of more complicated robot behaviours. Its modularity enables the design, development, debugging of controller programs and also the evaluation of different control methods. Controller programs may be implemented by *controller* classes edited from scratch or by combining source code from already developed



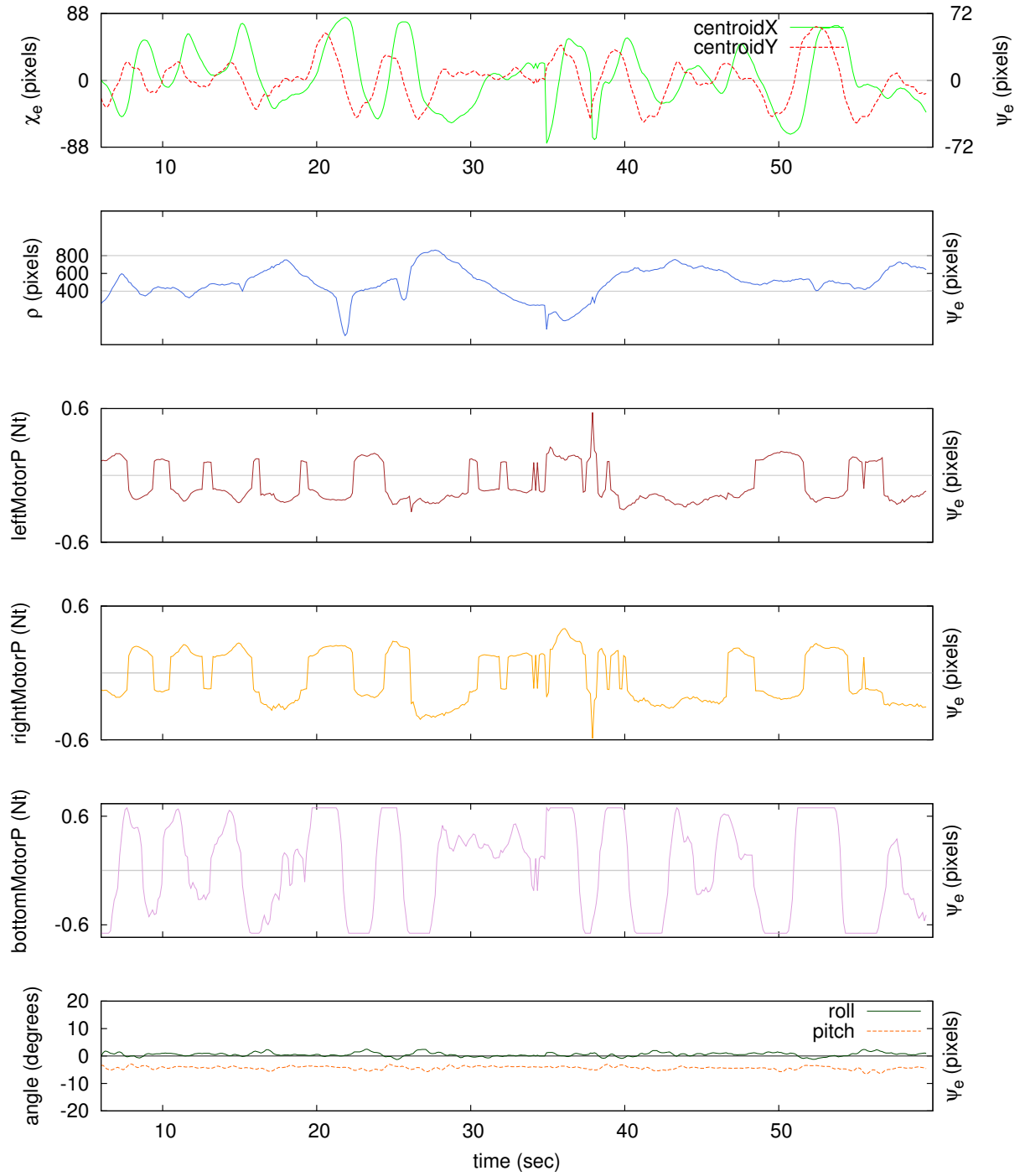


Figure 4.8: 'Light target following' Fuzzy Logic controller experimental results, where the first plot from top shows target's centroid  $d\chi_e$  and  $\psi_e$  errors. The second plot shows  $\rho$  value when  $\rho_{\oplus} = 600$  and  $\rho_{\sqcup} = 200$ . Next three plots show the controller defuzzified outputs for *leftMotorPower*, *rightMotorPower* and *bottomMotorPower* thruster power variables accordingly. The last plot depicts vessel's roll and pitch stability throughout the experiment.

classes. *Robot* class administers ACoS standard API and handles the compatibility issues between ACoS and system I/O devices. The PD controller class was designed and developed by editing a novel source code. As soon as PD controller class was tested and debugged a novel FL controller class was easily developed by including the FL library to a slightly altered version of the PD class source code. It was a straightforward process that resulted to a novel control program based on available tested software resources.

After the several project phases of controller design and implementation, it was clear that the use of a tank with bigger dimensions, at least 2.5m diameter and 1.5m height was necessary. Increased dimensions would provide the necessary area for experimenting with several other scenarios and with more than one *Ale III*- class vehicles. Using the experimentation area described at the previous sections, came out the conclusion that for every new vehicle added to the scenario, an addition of at least  $1\text{m}^3$  should be made to the tank's volume.

Another issue worth dealing, that could improve the experimental results, is the addition of special equipment to estimate each vehicle's absolute position inside the experimentation area. For example, placing an external camera at some height above the tank's centre, watching downwards, would make it possible to track a special light sign located at the vessel's top and would estimate its absolute position on the horizontal plane.

# Chapter 5

## Behaviour Based Biomimetic Control

### 5.1 Behaviour Based Control

Sea creatures, even the most simple ones, take part in a persistent survival race where sensing of the surrounding environment conditions, quick reasoning, and acting are issues of great importance in their everyday living. Natural evolution has endowed underwater creatures with brain abilities like decision making based on incomplete and noisy perception and physical capabilities like swimming, feeding and reproducing, that no robot constructed until now can macroscopically compete. Autonomy is a given characteristic found in all the ranks of underwater life biological classification.

Robot autonomy requires, among others, a robust control method. Contemporary research efforts have developed a large number and variety of distinct robot control schemes that, fundamentally may be divided in four classes [126]:

**Deliberative- Think, Then Act** methods, based on a sequence of sense- plan- act steps. The robot exploits the wholeness of its resources, sensory data, computational capabilities, knowledge and *experience* stored in its internal memory, to make a decision about its next step.

**Reactive- Dont Think, (Re)Act** implements the extremely common physical characteristic of *stimulus- response*. The sensory data are tightly coupled with the robot effector's outputs. Typically there is no intervening reasoning and the control procedure reacts directly to stimuli emerged from changing and unstructured environments. Instead of the deliberative, the reactive method does not feature acquisition or maintenance of world models.

**Hybrid- Think and Act Concurrently** aiming to combine the best aspects of the two previous methods.

**Behaviour Based Control- Think the Way You Act** method is structured with interacting modules, called *behaviours*, that collectively achieve the desired robot behaviour [130]. Behaviours are programmed as control modules that cluster sets of constraints in order to achieve and maintain a goal. Each robot behaviour receives inputs from its sensors and- or other behaviours and provides outputs to its actuators or to some other behaviours [15]. There is no centralized reasoning or world representation used by the deliberative methods. Also, behaviours do have state and can be used to construct representations, thereby enabling reasoning, planning and learning instead of reactive control's immediate and myopic responses, without representation, to a fast changing environment.

The choice of the right robot control methodology should be based on its particular characteristics, its task specifications and the overall environment conditions. In the present work, a behaviour based approach was selected for the implementation of *Ale III* underwater control. The vessel's task was to present inside the laboratory experimentation area a biomimetic behaviour that comprises most common activities among sea creatures that sustain autonomy: *wander, prey, predator avoidance* and *nesting*.

The basic requirement for *Ale III* autonomy is the self sustaining ability to *survive* for long periods without human assistance and intervention inside a dynamic complex environment. Behaviour based systems are best suited for this task where the environment presents continuous dynamic changes and fast response along with adaptivity are crucial and the ability to schedule and planning is also desirable [33]. To accomplish this the vehicle's ACoS follows basic principles of a behaviour based methodology[126]:

1. Behaviours are implemented as modules, *Controller* class members.
2. Each behaviour module may receive data from:
  - CMUcam3 and IMU sensory inputs, interfaced by the ACoS *Robot* class,
  - other *Robot* class members.

The outputs of a behaviour module are passed on to other behaviour modules or to AUV's effectors which are the three thruster motors.

3. Many different behaviour modules may independently receive input from the same sensors and output action commands to the same actuators.
4. Behaviours are encoded to be relatively simple, and are added to the system incrementally.
5. Behaviour modules are executed concurrently in order to exploit parallelism, speed of computation, as well as the interaction dynamics not only among behaviours but also between behaviours and the environment.

### 5.1.1 Ale III autonomous behaviour showcase

The main task of the controller of *Ale III* is to accomplish a safe diving autonomous operation under certain constraints. The vessel's autonomy follows a biomimetic approach inspired by the everyday routine of the majority of underwater living creatures [12]. Several phases of this routine seem to play an important role in preserving autonomy and, as a rule, are present among distinct physical behaviours. The most significant everyday task in underwater life is the search of food. To preserve their autonomous living, sea creatures have to sustain their vital energy level above a certain value by hunting and foraging. This leads to a perpetual role commutation for almost all sea creatures: each species may be a potential predator hunting other species and the same time is a candidate prey. Most of the sea creatures take advantage of a safe, cosy place to protect themselves from possible predators and hostile environmental conditions. Self preservation is an essential part of a creature's autonomy. Inside their nest, sea creatures rest and revitalize their physical potentials so as to be ready for the next cycle of foraging bouts and predators avoidance.

The former phases of underwater life routine along with the behaviour based control methodology principles were the basis of *Ale III* controller design. The controller succeeds to preserve the vessel's energy level, assuring safety and self preservation. At the same time it explores the surrounding area for possible targets- preys and is alerted for the detection of potential threats. This autonomous control method was extensively tested inside the underwater experimentation area of 1m<sup>3</sup>, described in Chapter 3, and proved its reliability not only in coordinating tasks like target following and object avoidance but also its efficiency in self preservation, showing its potentials for undertaking certain kinds of underwater missions. Although at its current configuration *Ale III* may undertake vision and orientation related tasks, ACoS behaviour based architecture, due to its modular and expandable design, enables a posteriori addition of alternative sensors' inputs, command outputs and also the design, implementation and encapsulation of novel behaviour modules. Thus, with a possible addition of equipment, ACoS behaviour based control architecture may potentially undertake missions of:

**oceanography, pollution detection and environmental monitoring** where the vessel should wander inside the bounds of a predefined underwater area, following a target course to collect data, avoiding possible impediments or threats. Such missions demand long lasting diving periods and, whenever the vessel's energy reserves decrease below a safety level, ability to emerge for recovery or return to a recharging base station located at a predefined position.

**security and surveillance** in harbours and underwater facilities where the vessel should either follow a predefined target course or arbitrarily wander to explore a limited area, trying to detect possible threats and invasions. The controller should monitor energy consumption to sustain long lasting autonomy. In the case that a docking station is available the controller should also navigate the vessel to it for recharging.

**inspection** of ship hulls or underwater facilities where the vessel should scan a given area trying to gather visual information about the submerged construction. The autonomous controller should navigate the vessel to repeatedly scan a limited area where some visual pattern, for example colour signs or shapes, should be detected while avoiding obstacles and monitoring the energy reserves. The vessel should return to a nearby base station for recharging or recovery. *Ale III* AUV is suited for such kind of missions thanks not only to the ACoS architecture but also due to its agile kinematics and vision sensing.

Additional ad hoc roles have been developed, as an essential part of the controller, to implement the basic biomimetic behaviours mentioned in the previous paragraphs [71], [171]:

- search- wander,
- act as prey- dynamic object avoidance,
- act as predator- dynamic object attraction,
- nesting as goal oriented behaviour- location attraction,

The controller's task is to manage these roles and their interaction with the vessel's environment and sustain its autonomous operation via supporting the following distinct tasks simultaneously:

- Vessel's energy reserves should never drop below a safety level. Although *Ale III* operation is by default energy consuming, still there are two options for recharging. The first option is to act as predator and follow a moving target as if hunting for food. Vessel's energy level will increase for as long as its distance from the target is inside a predefined range. If no target is available in the nearby environment and the energy reserves are reaching the safety level, then the controller must activate the second option, namely the nesting behaviour module to navigate the vessel towards a base station recharger. Both options have been used in ACoS for the following reasons:
  - Target following is a typical task in underwater robotics, covering fossil fuels' industry intervention missions, missions of cable deployment and route survey, explosive ordnance disposal, antisubmarine warfare, surveillance and reconnaissance.
  - Navigating to a specific underwater or surface location, serving as base, is crucial for every AUV mission for recharging or due to safety and recovery reasons.

At the experimentation scenario, implemented as part of the present dissertation, the recharging base is represented by a red coloured light sign in Figures 5.1, 5.2. The potential target for hunting is represented by a white light following an arbitrary three dimensional course inside the tank. Depending on the vessel's underwater environment and sensory equipment, the light signs may potentially be replaced by more effective and reliable acoustic signals transmitters, or by GPS signal receptions during successive emergences to the surface. Acoustic signals presume the deployment of a quite larger experimentation area than the one used during the *Ale III* testing and furthermore the GPS operation presumes outdoors experimentation. These were the reasons why the visual signs were chosen to represent the vessel's base within the indoors laboratory underwater experimentation area.

- Selectively navigate the vessel inside a predefined and marked underwater diving area. Inside the experimentation area the diving tank was marked with colour light signs indicating the limits of the diving area where the vessel should bound its operation (Figures 5.1, 5.2). The use of visual signs was inevitable due to reasons already described. The control scheme should guarantee this *bounded* navigation, because once the vehicle slips outside these limits it is considered to be lost and the controller procedure fails. Three different behaviour modules, programmed as distinct ACoS *Controller* class members, take over the compliance with this constraint:

1. *Proximity adjuster* is an ACoS behaviour module that equips *Ale III* with the ability to maintain its distance from the diving area's limits light signs within an acceptable range.
2. *Yaw control* is the behaviour module that implements *Ale III* yawing ability.
3. *Search - wander* subsumes the above two modules and navigates the vessel inside the bounded premises of the experimentation area. Apart from avoiding to overcome the experimentation area's limits, *search- wander* navigates the vessel to explore the bounded area.

In AUV offshore or coastal operation, serving missions of oceanography, environmental monitoring, pollution detection, fishery, hull inspection, security and surveillance [124], [102], [82], [7], [83], the issue of staying inside a predefined, somehow bounded, area is of great importance not only for efficiency reasons but also for the AUV's safety and recovery. Depending on the vessel's underwater environment and sensory equipment, the light signs may be replaced by more effective and reliable acoustic signals transmitters, or by GPS signal receptions during successive emergences to the surface.

- Moreover, the controller is able to detect and avoid a specific moving object recognised as threat. As soon as a possible threat is recognised, the controller cancels all other ongoing tasks and navigates the vessel to a safe base station, mimicking a prey's reaction when being hunted. This is another typical requisite of the AUV controller, contributing to vessels safety and recovery. In *Ale III* experimentation scenario this specific object- *threat* was implemented as a brown light sign shown at Figures 5.1 and 5.2, for reasons already mentioned in previous paragraphs.

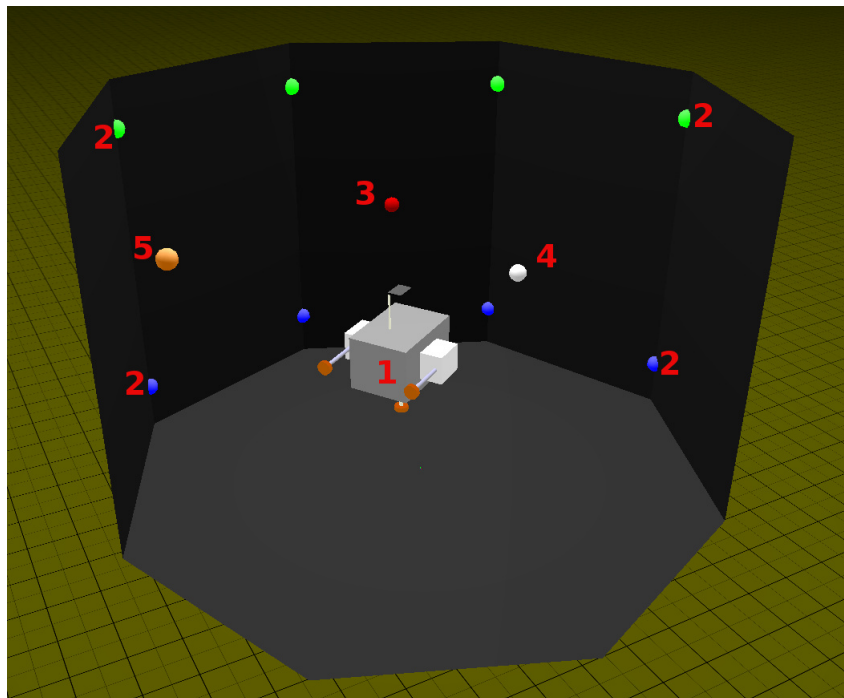


Figure 5.1: Side view of a three dimensional representation of *Ale III* experimentation scenario, (1) *Ale III* AUV inside the experimentation tank, (2) light signs indicating diving area's upper and lower limits, (3) base- nest, (4) moving target- prey, (5) moving threat- predator.

In the ACoS object- oriented environment each behaviour module is an active *Controller* class member, initialized during the creation of *Controller* class at software's start up procedure. During the program's

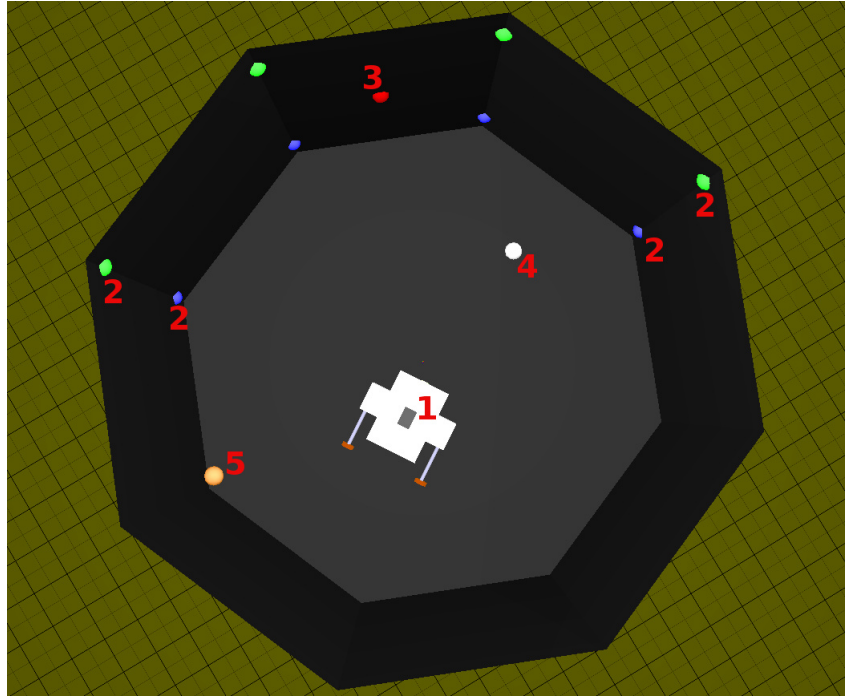


Figure 5.2: Top view of a three dimensional representation of *Ale III* experimentation scenario, (1) *Ale III* AUV inside the experimentation tank, (2) light signs indicating diving area's upper and lower limits, (3) base- nest, (4) moving target- prey, (5) moving threat- predator.

execution, all modules access, via *Robot* class' interface, the same sensory inputs and compete for the same controller outputs during each controller step, as shown in Figures 3.9 and 4.4. The sensory inputs are produced by the vision sensor module and the IMU. The visual data are necessary for the detection of light signs and the estimation of vessel's distance from them. IMU data are used to determine the vessel's three dimensional orientation and navigate inside the limited diving area. The ACoS behaviour modules are layered incrementally in a three- level subsumption architecture. Section 5.2 describes the distinct behaviour modules as simple Finite State Machines (FSM). FSM representation is the standard technique used to describe behaviour based control methods [33]. It was chosen against other representation techniques, like data flow diagrams, flow charts or pseudo code for its simplicity and ability for intuitive graphical representation. All ACoS behaviour modules were described by a FSM containing at most four states and nine transitions. Moreover, designing and representing the ACoS architecture with stand alone FSMs aided the compliance with the behaviour based principles of concurrent execution, modularity, simultaneous reception of sensory inputs and competence for the controller outputs [126].

## 5.2 Behaviours Subsumption Architecture

The behaviour based model comprises of a subsumption architecture of six different behaviours deployed at three levels of competence shown in Table 5.1, involving interconnections, inhibitors and suppressors [33], [126]. Each behaviour is presented as an FSM at the following sections. It may be interconnected with other behaviours of the same or of a different level. Depending on the behaviours interactions and commutations, the level of AUV's energy reserves increases or decreases after each controller time step. Energy consuming behaviours are the ones that have no means of recharging energy, but rather consume the vessel's energy to

Table 5.1: Behaviours Index

level of competence	behaviour number	behaviour name	energy balance
0	01	proximity adjuster	negative
0	02	yaw control	negative
1	11	search- wander	negative
1	12	nesting	negative/positive
2	21	prey	negative/positive
2	22	predator	negative/positive

navigate it. They are listed in Table 5.1 as negatively balanced while those which feed energy to the AUV by hovering inside the recharging base premises or during a successful target following task, are listed as positively balanced. For some behaviours the energy balance may alternate between positive or negative, depending on their progress and outcome. Thus, for example, a successful target following task will feed energy to Ale III in contrast to an unsuccessful one, that will just consume some of its energy reserves. In the current ACoS implementation the energy term is virtual, implemented as an ACoS variable. There is no *hardware* meaning for AUV's energy level and so after a successful *hunt*, for example, the increase of energy level is virtual, by the means of ACoS, without some kind of energy transformation or transfer between the vessel and its surrounding underwater environment. As stated in previous section, the overall control procedure aims at preserving the virtual energy above a safety level. The ACoS performance optimization depends on the fine tuning of the behaviours' interaction parameters, so as to obtain the lowest energy consumption during controller *life* time.

According to the experimentation scenario, ACoS is challenged to *live* as long as possible, swapping behaviours and trying to avoid the danger of *predators*. Green and blue light signs attached on the experimentation tank's top and the bottom internal circumference define the vessel's *living* and diving space limits. Its nest-base is denoted by a red light sign, attached to a specific position inside the tank. A brown light suspended from a rod, able to move underwater, plays the AUV's predator role, while a white one represents the possible prey as shown in Figures 5.1 and 5.2.

ACoS behaviours are competing for the determination of motors' output control commands. Each discrete behaviour may or may not participate to the output commands depending on the behaviours interaction and the dynamically changing environmental conditions. During the AUV's dive, its energy level fluctuates according to the environment conditions and the behaviours interaction. For example, *living* in an environment with redundancy of preys has definitely a positive influence on AUV's energy level. In controller time step this level increases or decreases by a fixed quantity, depending on whether the behaviour in charge features positive or negative energy balance as shown in Table 5.1. *Predator*, *nesting* and *prey* behaviours present positive or negative energy balance according to the progress of their goals. For example the *predator* behaviour may be negatively balanced if the prey escapes or partly negative, during pursuit, and mainly positive if the prey is caught after the hunt. Throughout the consecutive controller steps, behaviours interact and produce controller outputs analogous not only to AUV's position and orientation inside the experimentation tank but also to its energy reserves and the surrounding area stimuli, promoting ACoS autonomy.

### 5.2.1 Level 0

The zero level diagram of Figure 5.3, comprises the following two basic navigational behaviours:

**01 proximity adjuster**, receives sensor inputs from AUV's camera and controls the lateral thrusters power.

**02 yaw control**, navigates the AUV to an orientation of a fixed target yaw value.



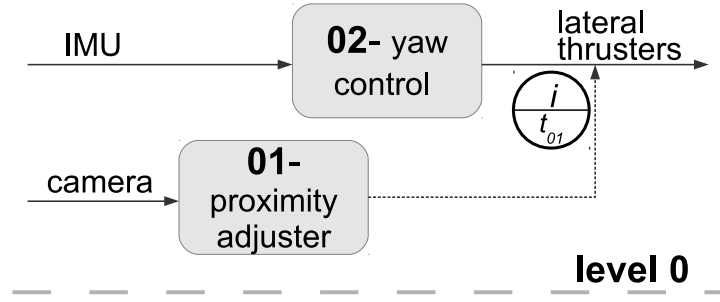
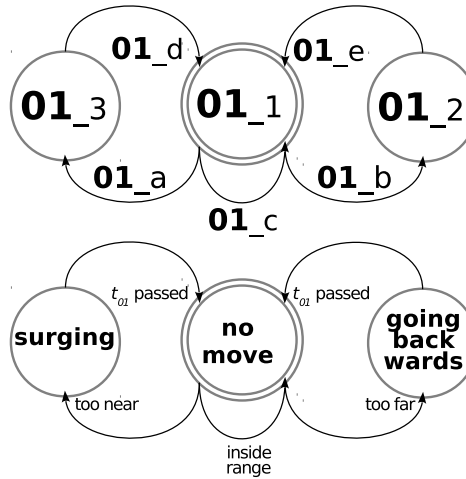


Figure 5.3: Zero level behaviour based control subsumption architecture.

Zero level is the basis for the upper two levels. It implements ACoS potentiality for avoiding impediments, achieving and maintaining an orientation. At FSM's states and inputs numbering, for every distinct behaviour, its level number is used as a prefix. Number and letters are used for the states and inputs numbering accordingly. An FSM initial state is denoted by its double lined circle.

Both zero level's behaviours consume energy as long as they take over the control procedure. The wire next to the circle, connecting the output of the *proximity adjuster* behaviour to the *yaw control*, is an inhibitor [33]. When an output message from the *proximity adjuster* behaviour travels along this wire it inhibits *yaw control* outputs. The letter *i* at the upper hemicycle, depicts inhibition, while the  $t_{01}$  at the bottom hemicycle denotes the inhibition time in controller step numbers. Thus, when the *proximity adjuster* behaviour produces a control output for AUV's lateral thrusters, it inhibits *yaw control* outputs and cancels their transmission to the lateral thrusters, for a time period of  $t_{01}$  controller time steps. A simple example of behaviours interaction is given at the end of this section.

### Behaviour 01: proximity adjuster

Figure 5.4: Zeroth level, 01- *proximity adjuster* behaviour's FSM diagram.

The robot, using its camera, estimates the distance from the stationary coloured signs attached to the tank walls. The *proximity adjuster* behaviour tries to keep this distance inside a fixed range and thus avoids hitting on the tank walls. This behaviour module is presented by an FSM consisting of the following states:

- **01\_1**: vessel is not performing any movement. There is no thruster power manipulation.

- **01.2:** vessel is performing a backwards surge movement for a predefined number of control steps  $t_{01}$ .
- **01.3:** vessel is performing a surge movement for a predefined number of control steps  $t_{01}$ .

The *proximity adjuster* behaviour's FSM commutates its internal states according to the following sensor inputs, as described in Figure 5.4:

- **01.a:** coloured signs traced by the vision sensor show that the AUV's distance from the tank walls is further than the maximum permitted value.
- **01.b:** distance from tank walls is closer than the minimum permitted value.
- **01.c:** distance from tank walls is inside the permitted range.
- **01.d, 01.e:**  $t_{01}$  controller time steps passed.

### Behaviour 02: yaw control

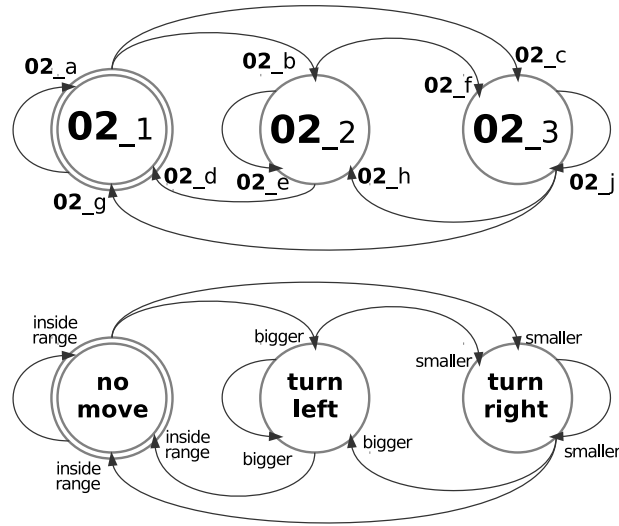


Figure 5.5: Zeroth level, 02- *yaw control* behaviour's FSM diagram.

The AUV controller uses data from the IMU sensor module to estimate its yaw heading. *Yaw control* behaviour manipulates the robot's lateral thrusters, trying to achieve a fixed value of yaw orientation. This behaviour module is presented by an FSM shown in Figure 5.5, consisting of the following states:

- **02.1:** AUV is not performing any movement- no thruster power manipulation.
- **02.2:** AUV is turning left, around its vertical, body oriented axis.
- **02.3:** AUV is turning right, around its, vertical, body oriented axis.

The behaviour's transitions and sensor inputs regarding the above states are:

- **02.a, 02.d, 02.g:** the yaw heading value, estimated by the robot's IMU module, lies between the correct limits, around a predefined target value.
- **02.b, 02.e, 02.h:** yaw value is bigger than the target value
- **02.c, 02.f, 02.j:** yaw value is smaller than the target value, outside the correct limits.

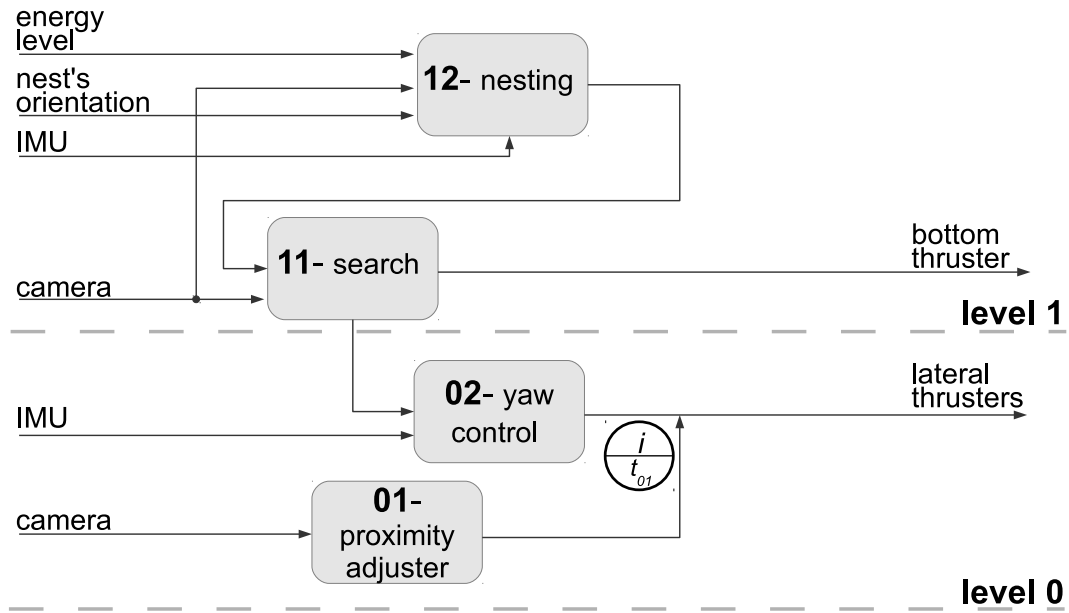
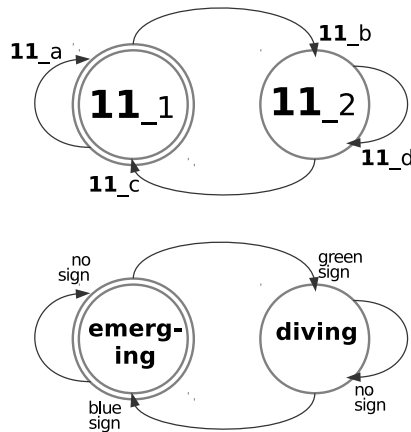


Figure 5.6: First level behaviour based control subsumption architecture.

### 5.2.2 Level 1

Level 1 diagram introduces two new behaviours (Figure 5.6). *Search- wander* for exploring the space between the tank's surface and bottom coloured light signs and *nesting* behaviour for returning to the robot's nest- base. *Search- wander* subsumes the lower level *yaw control* behaviour. A simple example of behaviours interaction is given at the end of this section.

#### Behaviour 11: search- wander

Figure 5.7: Level 1, 11- *search- wander* behaviour's FSM diagram.

There are green signs attached to the walls of the experimentation tank, near the surface and blue ones near the bottom. They indicate the safe diving area limits for the robot. Level 1 *search- wander* behaviour

implements the following two consecutive actions:

1. Chooses a yaw value and manipulates heave thruster in order for the AUV to emerge. The emerge pauses when the camera sensor module detects the green light signs, indicating the upper limit of the AUV's diving zone.
2. When the camera sensor module detects green light signs, the AUV starts to dive, choosing a new yaw value. This diving terminates as soon as the camera module detects the blue light signs, indicating the bottom limit of the diving zone.

The *search-wander* behaviour module is presented by an FSM, Figure 5.7, consisting of the following states:

- **11.1**: emerging, keeping a steady yaw heading,
- **11.2**: diving, keeping a steady yaw heading,

and the following transitions and sensor inputs:

- **11.a**, **11.d**: no sign detected,
- **11.b**: green sign detected,
- **11.c**: blue sign detected.

### Behaviour 12: nesting

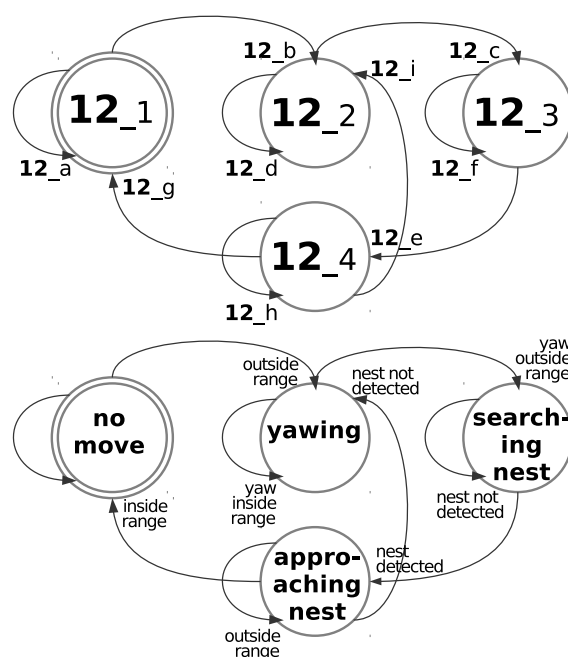


Figure 5.8: Level 1, 12- *nesting* behaviour's FSM diagram.

The *nesting* behaviour implements AUV's homing procedure. It interacts with the *search-wander* behaviour (Figure 5.6) and, depending on energy reserves, may decide to navigate the AUV to its nest- base. A red coloured light sign, placed at a fixed orientation between the tank's surface and bottom coloured light signs, indicates nest's position. *Nesting* successfully completes its task when the value of AUV's distance from the red light is inside a predefined range. It presents a positive energy balance as long as the AUV's position is inside the

nest- base range. While on the search of nest- base, the AUV consumes energy. *Nesting* behaviour module is presented by an FSM (Figure 5.8) consisting of the following states:

- 12.1: inside nest, no movement,
- 12.2: yaw turning,
- 12.3: searching for red coloured nest- base light sign,
- 12.4: nest detected, approaching to nest,

and the following transitions and sensor inputs:

- 12.a: AUV's distance from nest is inside range,
- 12.b: distance from nest is outside range,
- 12.c: AUV's yaw heading value is inside range, heading towards nest,
- 12.d: yaw heading value is not inside range,
- 12.e: Nest's colour sign detected,
- 12.f: Nest's colour sign not detected,
- 12.g: AUV's distance from nest is inside range,
- 12.h: distance from nest is outside range,
- 12.i: nest's colour sign not detected.

### 5.2.3 Level 2

The final top level of competence adds *predator* and *prey* behaviours to the behaviour based robot control model (Figure 5.9). During the experimentation scenario the presence of a white light target represents a prey the AUV has to hunt and catch. *Predator* behaviour searches for energy- food, by means of finding this target-prey, every time the AUV's energy reserves fall below a predefined safety level. The detection of a brown light reverses the roles. As soon as the controller is alerted by the detection of a brown coloured light, the *prey* behaviour navigates the AUV straight to its nest- base. In Figure 5.9 the use of letter *s* in the upper hemicycle, next to the wire, connected to a behaviour's input, depicts suppression of the corresponding input [33]. A suppressed behaviour input is replaced by the signals of the suppressor for a predefined period of time. Level 2 inhibitors and suppressors use the following special symbols:

$t_{02}$  denotes the time period while *predator* behaviour signals inhibit the outputs of *search* and *yaw control* behaviours. This inhibition lasts until the prey is caught or, otherwise, for a predefined number of controller steps since the white light prey target was detected.

$t_{03}$  denotes the time period while *nesting* behaviour suppresses the inputs of *prey* and *predator* behaviours. This suppression lasts for a predefined number of controller steps, in case *nesting* serves a *prey* behaviour call, or until AUV's energy exceeds a prerequisite level.

A simple example of behaviours interaction is given at the end of this section.

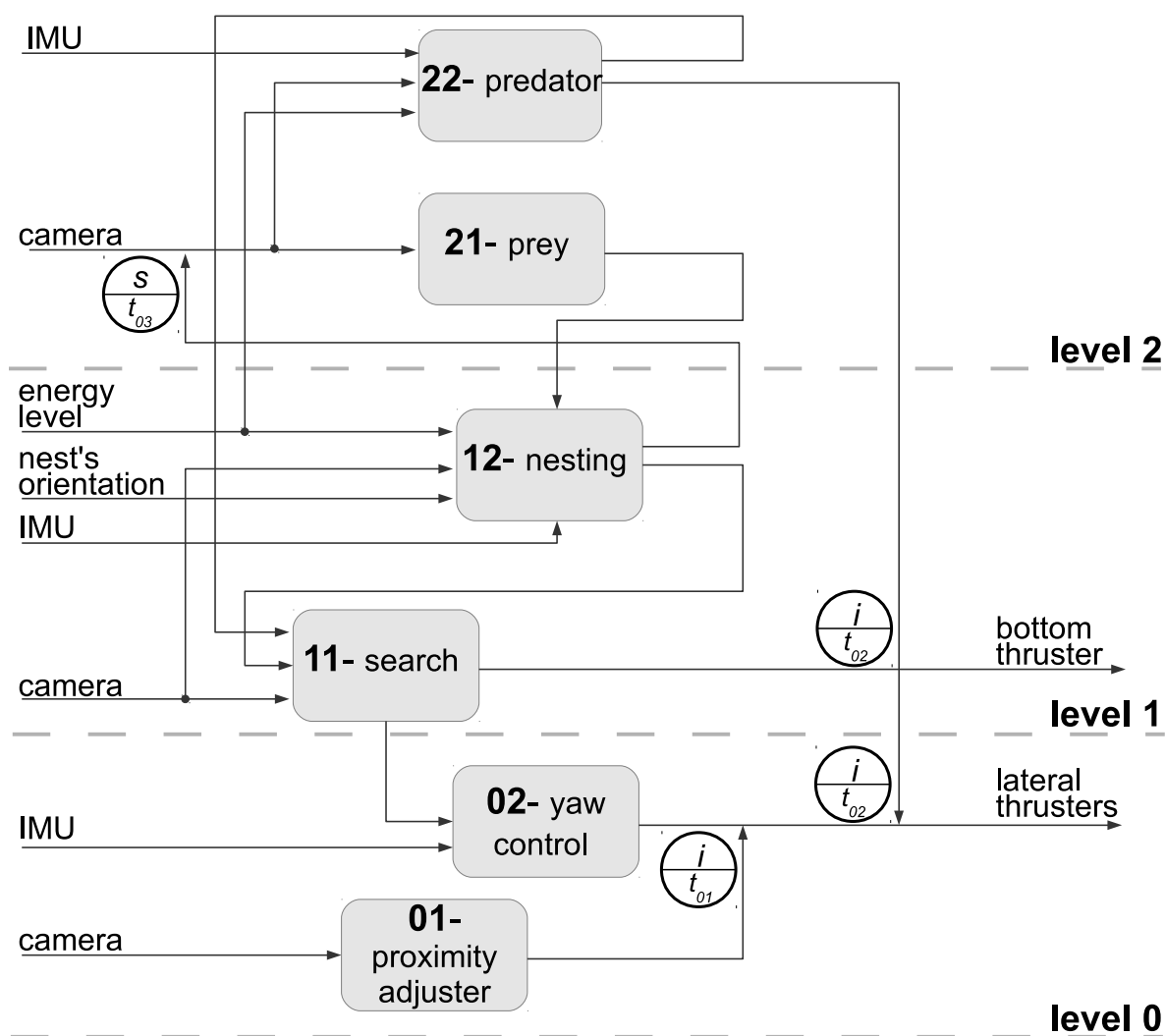


Figure 5.9: Level 2 behaviour based control subsumption architecture.

### Behaviour 21: prey

According to Figure's 5.9 subsumption architecture, the *prey* is connected to the *nesting* behaviour. Thus, when the AUV acts like a prey, it is heading to its nest- base using the subsumed *nesting* behaviour. Camera sensor data, used as inputs for both *predator* and *prey* behaviours are suppressed by *nesting* behaviour signals. These signals cancel respectively all possible prey targets and enemy predators detections, until reaching the safety of the nest- base. *Prey* features positive energy balance as long as AUV's position is inside nest- base range. Otherwise and while on the search of nest- base, the AUV consumes energy. The *prey* behaviour module is presented by an FSM (Figure 5.10), consisting of the following states:

- **21.1:** nesting has been accomplished,
- **21.2:** searching for the nest- base,

and the following transitions and sensor inputs:

- **21.a:** distance from nest- base is inside range,

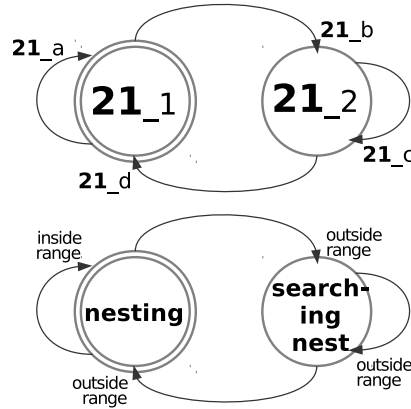


Figure 5.10: Level 2, 21- prey behaviour's FSM diagram.

- 21\_b: distance from nest- base is not inside range,
- 21\_c: distance from nest- base is outside range,
- 21\_d: distance from nest- base is inside range.

### Behaviour 22: predator

The *predator* behaviour monitors energy supplies to decide the time for hunting, searching for energy- food. It is connected to the lower level *search* behaviour for implementing the hunting procedure inside the diving area limits of the experimentation tank. *Predator* behaviour inhibits the outputs of the lower level *search- wander*, *yaw control* and *proximity adjuster* behaviours. The prey is a white light target, moving randomly inside the tank. When the *predator* behaviour takes charge of the AUV control, it consumes energy until locking a prey target for a predefined time, meaning the prey is caught. After this successful outcome, the AUV controller is awarded with a fixed quantity of energy. If the prey is lost before this predefined time passes, then there is no increase in the energy level. The *predator* behaviour module is presented by an FSM (Figure 5.11), consisting of the following states:

- 22.1: on the hunt of detected prey- target,
- 22.2: prey- target locked,
- 22.3: searching for prey,

and the following transitions and sensor inputs:

- 22.a: distance from prey- target is inside range,
- 22.b: distance from prey- target is not inside range,
- 22.c: prey- target is lost,
- 22.d: distance from prey- target is inside range,
- 22.e: distance from prey- target is not inside range,
- 22.f: prey- target is lost,
- 22.g: prey- target is locked for a predefined time period and the hunting is successfully terminated,
- 22.h: prey- target is detected,
- 22.j: prey- target is not detected.

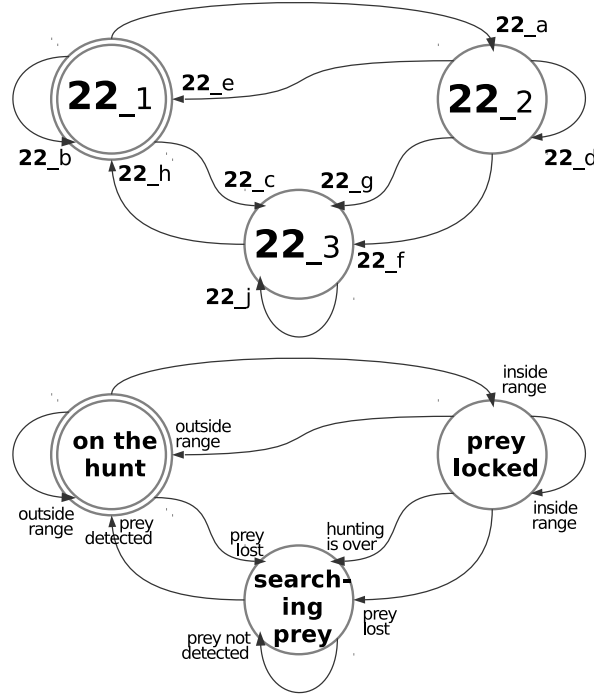


Figure 5.11: Level 2, 22- predator behaviour's FSM diagram.

### 5.2.4 Behaviours Interaction Example

A typical ACoS behaviours' interaction is presented here to depict the controller's functionality. An effort to describe an example of ACoS operating inside the experimentation area using photograph snapshots was rated as unsuccessful due to the experimentation area's small dimensions, tanks opaque black colour and light diffusion inside water. These reasons lead to poor photograph's quality and non descriptive snapshots. Thus such an example will be depicted below, using pictures of a testing scenario inside a three dimensional model of the experimentation area. During the example, the vessel wanders in front of the area of green and blue lights while keeping its distance for them inside a predefined range (Figures 5.1, 5.2). The ACoS continually detects for the invasion of a possible threat or the existence of a potential target, represented by a brown and white light sign accordingly. When the brown light is detected the vessel is heading for its nest- base, located at a predefined place marked by a red light sign, where it will hover for a given time period, before returning to its routine, assuming the threat has disappeared. The controller also manipulates energy reserves and tries to increase them by following the white light target whenever it appears in its field of view. If this is not the case and energy reserves are near the safety level, ACoS navigates *Ale III* to its nest- base. The example phases, summarized in Table 5.2, are described below.

1. The top level behaviour modules define *Ale III* routine of searching for its target inside the limited experimentation area while simultaneously trying to avoid possible threats. *Ale III* starts near the surface of the experimentation tank and the VSMP detects a green light sign (Figure 5.12). The bottom left frame in each picture shows the CMUcam3 sensor's view. *Search- wander* behaviour is in charge and since the vessel has reached the upper limit of the diving area, it chooses vessel's new orientation for the subsequent diving. The same time *proximity adjuster* is manipulating vessel's distance for the light sign, using VSMP data. The vessel is outside range and thus it manipulates lateral thrusters to approach towards the green light signs. During this period, *proximity adjuster* inhibits *yaw control* outputs for a period of  $t_{01}$  controller time steps. During this phase the energy level decreases in a stepwise manner by amounts proportional to the thrusters' output power.



Table 5.2: ACoS reflexes in accordance with the detected light colour signs

<i>Colour</i>	<i>ACoS Behaviours</i>	<i>Next Action</i>	<i>Physical Meaning</i>
green	search- wander yaw control proximity adjuster	stop emerging and start diving	<i>Ale III</i> reached experimentation area's upper limit
blue	search- wander yaw control proximity adjuster	stop diving and start emerging	<i>Ale III</i> reached experimentation area's lower limit
brown	prey nesting search- wander yaw control proximity adjuster	navigate to base- nest	<i>Ale III</i> detected a threat-predator
red	prey nesting search- wander yaw control proximity adjuster	hover in front of the base- nest	<i>Ale III</i> is recharging or trying to protect from a threat-predator
white	predator	follow target	<i>Ale III</i> detected a target- prey

2. After  $t_{01}$  steps the *yaw control* changes the vessel's orientation, increasing its yaw heading by  $20^\circ$  and *Search- wander* manipulates *Ale III* bottom thruster to start diving for exploring a new, adjacent portion of underwater area. During diving, *yaw control* assures proper vessel's orientation until it comes close to the bottom limit of the diving area and *search- wander* detects a blue light (Figure 5.13). As before, the bottom left frame in each picture shows CMUcam3 sensor's view. During this phase the energy level decreases in a stepwise manner by amounts proportional to the thrusters' output power.
3. When the blue light is detected the *proximity adjuster* is manipulating the vessel's distance from the light sign, using VSMP data. The distance is inside the acceptable range and thus *proximity adjuster* produces no output thrusters' commands. *Search- wander* decides vessel's next yaw heading being that of the first from the left limit colour signs, and so the vessel remains inside the limited diving area. The positions of the farthest tank wall signs is a known ACoS parameter. *Yaw control* rotates the vessel to the new yaw heading and in the same time *search- wander* starts emerging. During emerging a brown light, representing a predator threat for *Ale III*, is detected by *prey* behaviour module, Figure 5.14. *Prey* module uses the lower level *nesting* behaviour to navigate the vessel to its base- nest and avoid the predator threat. *nesting* uses, in turn, *search* and *yaw control* to handle vessel's navigation to the given nest heading and locate inside the limits of the diving area. Until the detection of the nest- base, *Ale III* energy level decreases in a stepwise manner by amounts proportional to the thrusters' output power.
4. When ACoS detects its nest- base, represented inside the experimentation area by a red light sign, *prey* behaviour module commands the *nesting* module to hover the vessel near the red light keeping its distance between them inside a predefined range, Figure 5.15. ACoS subsumption architecture uses, in turn, *search* and *yaw control* to accomplish this task. *Nesting* behaviour suppresses the camera inputs to the higher level modules to assure that the vessel will hover in front of the base- nest for the predefined suppressor time period of  $t_{03}$  controller time steps. During the time period of  $t_{03}$  controller time steps

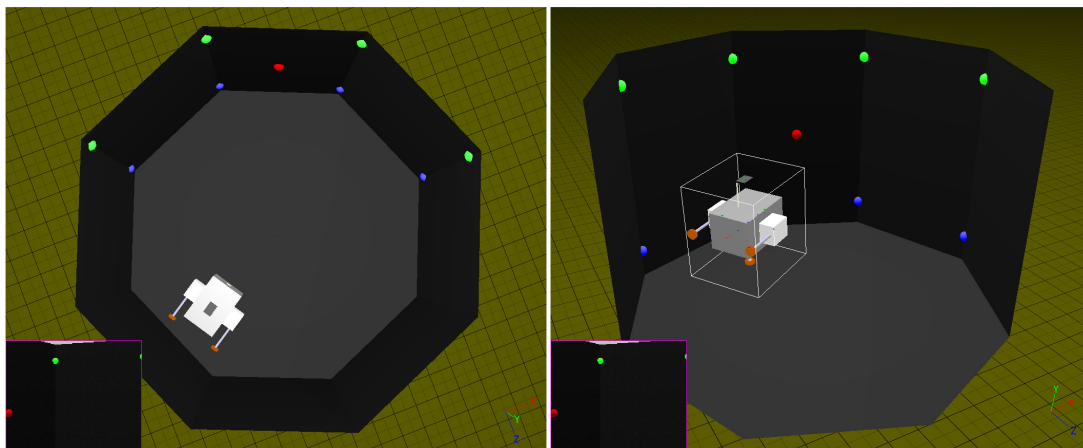


Figure 5.12: Example of ACoS behaviours' interaction, phase 1.

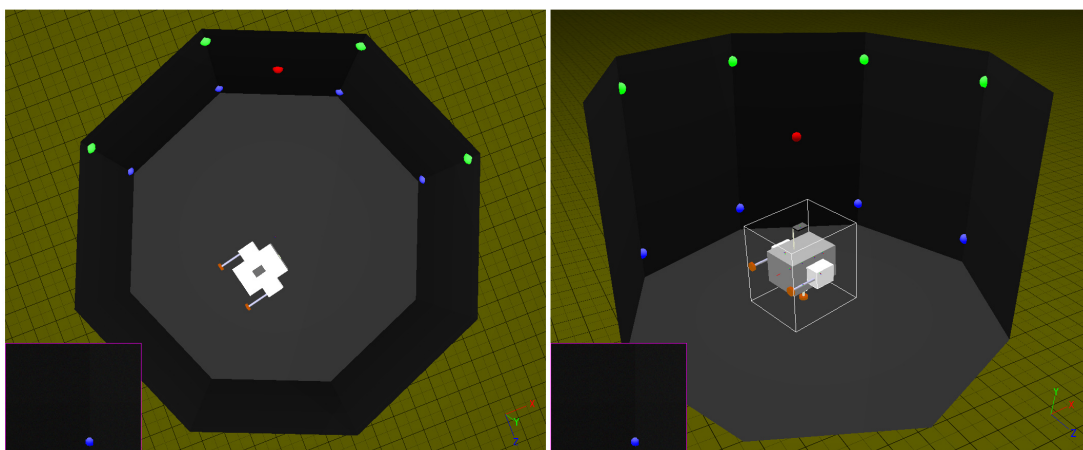


Figure 5.13: Example of ACoS behaviours' interaction, phase 2.

of hovering in front of the nest- base, *Ale III* energy level increases by a predefined amount at every controller step.

5. After  $t_{03}$  time steps, ACoS returns the vessel to its routine of searching for a white light target inside the limits of the experimentation area. Thus, it starts emerging and returns to the initial phase of wandering (Figure 5.16). *Ale III* energy level decreases during this phase. As before, the bottom left frame in each picture in Figure 5.16 shows CMUcam3 sensor's view.
6. During wandering, a white light target is detected (Figure 5.17). The *predator* behaviour module immediately inhibits *search-wander* and *yaw control* outputs, for at most  $t_{02}$  controller time steps, taking full control of vessel's thrusters. The value of  $t_{02}$  time period is a given ACoS inhibitor parameter and assures that the hunt will not exceed a maximum time period, for energy reserves safety. Alternatively, if the *predator* module succeeds in capturing the white target light, that is, to follow it closely for a predefined interval, the inhibition stops and ACoS returns the vessel to its routine operation of searching the next target. During target hunting and until the its distance from vessel is inside an acceptable range, *Ale III* energy level decreases in a stepwise manner by amounts proportional to the thrusters' output power.

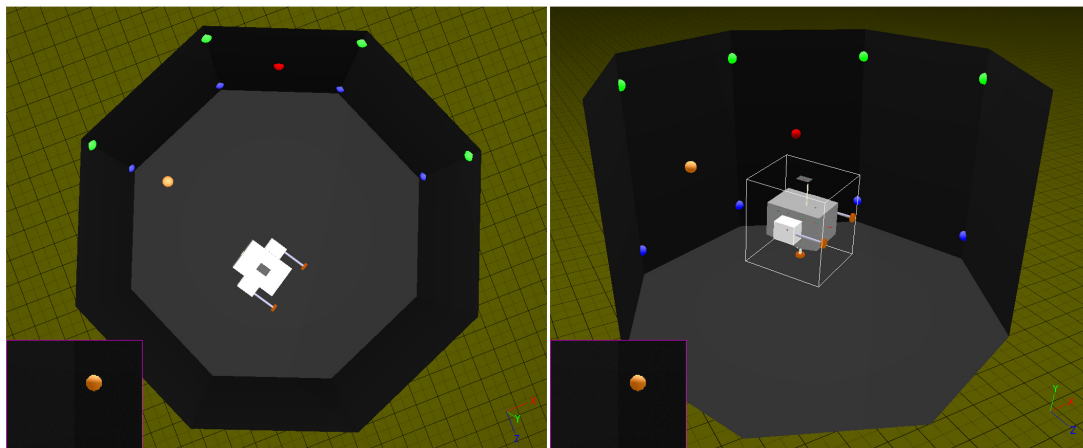


Figure 5.14: Example of ACoS behaviours' interaction, phase 3.

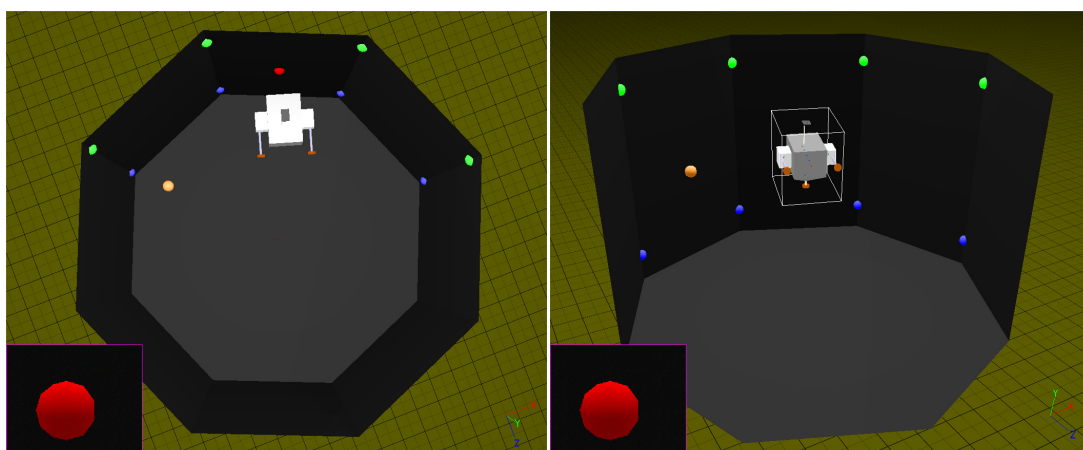


Figure 5.15: Example of ACoS behaviours' interaction, phase 4.

7. ACoS navigates the vessel to follow a white light target along an arbitrary three dimensional course inside the experimentation area (Figure 5.18), aiming at placing the target at the centre of vision sensors field of view and inside a predefined distance range. *Ale III* energy level increases by a predefined amount during the time steps that ACoS manages to:
  - (a) keep its target at the centre of vision sensor's field of view and
  - (b) vessel's distance from target inside range.

As before, the bottom left frame in each picture in Figure 5.18 shows CMUcam3 sensor's view.

## 5.3 Experimental Results

ACoS was extensively tested using *Ale III* AUV in the indoors laboratory experimentation area described in Chapter 4. The experimentation tank was marked with light signs representing the diving area's limits, while a red coloured light sign was the base- nest. A brown and a white light suspended from a rod, appeared randomly

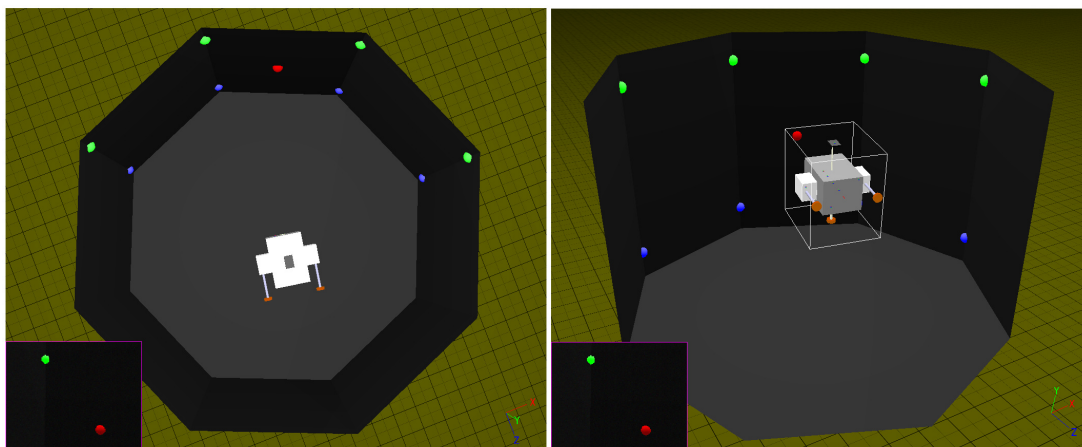


Figure 5.16: Example of ACoS behaviours' interaction, phase 5.

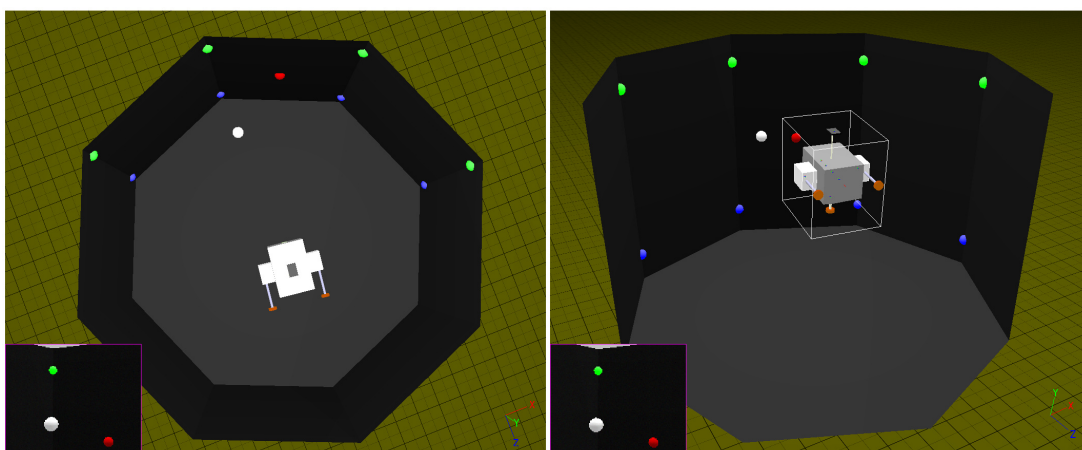


Figure 5.17: Example of ACoS behaviours' interaction, phase 6.

inside the diving area and following arbitrary three dimensional courses, played the role of a potential threat-predator and target- prey accordingly. This kind of experimentation aimed at highlighting the controller's ability to sustain autonomous underwater operation, mimicking the commutation of the several behaviours typically found in the routine of undersea living creatures. The experimentation had several phases of search and wander, acting as prey with dynamic object avoidance, acting as predator with dynamic object attraction, nesting as goal oriented behaviour and location attraction because these phases are common tasks during AUV missions in coastal line and offshore. ACoS was challenged to support *Ale III* autonomous operation and dynamically confront possible threats in its search for potential targets and in the same time monitor vessel's energy reserves, assuring reliability and safe recovery.

Testing with *Ale III* and ACoS began inside the experimentation area with zeroth level behaviours. *Yaw control* was programmed on the basis of the yaw PD controller described at the previous chapter. *Proximity adjuster* uses program modules from the 'light target following' PD controller, also described at the previous chapter. Next levels were implemented and tested incrementally. *Search- wander, nesting, prey* and *predator* were also programmed based on the already tested and evaluated 'light target following' controller. Special consideration regarding the *Controller* behaviour class members design was given to the intercommunication signals management, the use of inhibitors and suppressors. These issues dealt with caution and respect to the

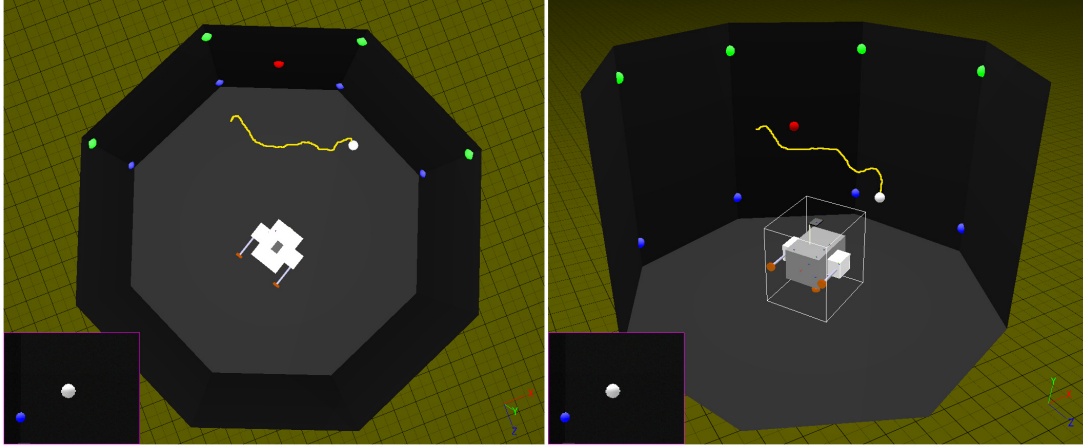


Figure 5.18: Example of ACoS behaviours' interaction, phase 7. The yellow line denotes white light target's three dimensional course.

basic principles of behaviour based control.

The experimentation scenario was chosen to trigger and use alternatively all the behaviours: The vessel begins its underwater commission or, using biomimetic terms, *life* filled up with energy and wanders between its area limits in the experimentation tank. After a while a brown light *predator* or a white light *prey* may appear to the underwater scenery and *Ale III* must respond accordingly by hiding at the safety of its *nest- base* or hunt the *prey*. Wandering or hunting decreases energy reserves and if a *prey* is not available then the vessel must visit its *nest- base* for recharging.

Figures 5.19, 5.20, 5.21, 5.22 show real life experimentations results from tests conducted at the experimentation area following previous paragraph's scenario. Each test lasted several minutes but the plots show only the first three minutes for readability reasons. The first plot of each figure shows behaviours' interaction log throughout the experiment. The second plot shows AUV's virtual energy level and instantaneous consumption measured in virtual energy units. At each controller step a '+' mark is added for those ACoS behaviours that involved in determining the controller outputs. According to the behaviour based design principles, all of them have access to the controller outputs. Still, at each controller time step, not all of them contribute to the control outputs due to the subsumption architecture inhibitors and successors or the lack of the proper stimulus. The following paragraph describes test A and Figure 5.19. Analogous meanings have the figures of the remaining tests: B, C and D.

**Figure 5.19, first plot:** As soon as test A begins, the vessel uses *search- wander* behaviour and dives within the area limits defined by the green light signs near the tank's surface and the blue light signs at the bottom. The test begins with the vessel floating at the surface. It immediately detects surface's green lights and starts diving at 0.2 seconds (blue *search<sub>B</sub>* marks at the plot) until it detects the blue bottom lights, at 8 seconds. After that, it emerges (green *search<sub>G</sub>* marks) until it detects the green surface lights at 16 seconds, and so on. During this time of wandering inside the tank,

- *yaw control* takes also part to the controller outputs' determination (gold- yellow *yawControl* marks at the plot during 1- 16 seconds, 34.5- 36.5 seconds etc) to fix AUV's orientation.
- *proximity adjuster* takes also part to the outputs' estimation, if the vessel is too close or too far from the limit colour light signs. Thus, in some instances between *search<sub>G</sub>* and *search<sub>B</sub>* behaviours' commutations (olive *wall* marks at the plot during 0.5-1 seconds, 36.5- 37.5 seconds etc.) *proximity adjuster* is also in charge, among other behaviours.

ACoS assures that in an interval of 5 controller steps the *prey* behaviour takes full control of the

outputs (scarce *prey* red marks at the plot). If a brown light *predator* is detected, *prey* becomes the dominant behaviour, suppressing CMUcam3 inputs to all the rest behaviours (dense *prey* red points at the plot during 61- 71 seconds) and thus exempting them from the output determination. Along with the subsumed *nesting* behaviour, they are controlling the outputs from the moment a predator was detected until the vessel finds its nest and passes a predefined time period in it (fuchsia *nesting* points during 61- 71 seconds). At a different interval of 5 controller steps, *hunt* behaviour takes outputs' full control (scarce *predator* brown points at the plot) to search for a white light virtual *prey*. If this is the case, *predator* behaviour takes charge and tries to lock the light target at CMUcam3 field of view (dense *predator* brown points at the plot during 16- 34.5 seconds). During this period, *hunt* inhibits other behavioural outputs. Finally when the energy level is below a safety value, *nesting* behaviour takes full control of the outputs (fuchsia *nesting* points during 132- 157 seconds), suppressing CMUcam3 inputs to all other behaviour modules. It searches for the *nest- base* and virtual energy recharging. After recharge, *Ale III* keeps on wandering in search of *predators* and *preys* and its *life* continues according to the experimentation scenario. During experimentation energy level is full at 2000units and the safety level is at 1800units.

**Figure 5.19, second plot:** The red line shows AUV's virtual energy level. Energy level continually decreases except when a prey is locked at the camera's field of view, during 16- 34 seconds, or when recharging at the nest- base, 61- 71 and 132- 157 seconds. The medium- green line shows AUV's instantaneous energy consumption at each controller step. The consumption depends only on thruster motors' turning speed and it is a *real* quantity instead of energy level. The custom measurement unit used to estimate the virtual energy level and the real consumption during each controller time step, is analogous to the motors spinning speed, counting 1 when all motors are spinning at full speed and 0 when all of them are stopped.

During experimentation AUV's external power connector along with external power supply were used for unlimited time testing, at the cost of power supply cables burden. Cables should be arranged with caution throughout the test to avoid entangling or breaking them during vessel's diving missions. As long as *Ale III* had enough *real* energy reserves its operation kept on, following the experimentation scenario, without technical problems or ACoS crashes for more than 15 minutes. Continuous operation duration was restricted not from some vessel's or ACoS deficiency, but from the small dimensions of the experimentation area. The arrangement of power supply cables along with the *prey- predator* lights moving underwater on their rods with at least two persons in charge, is a challenge at the 2m<sup>2</sup> area of experimentation.



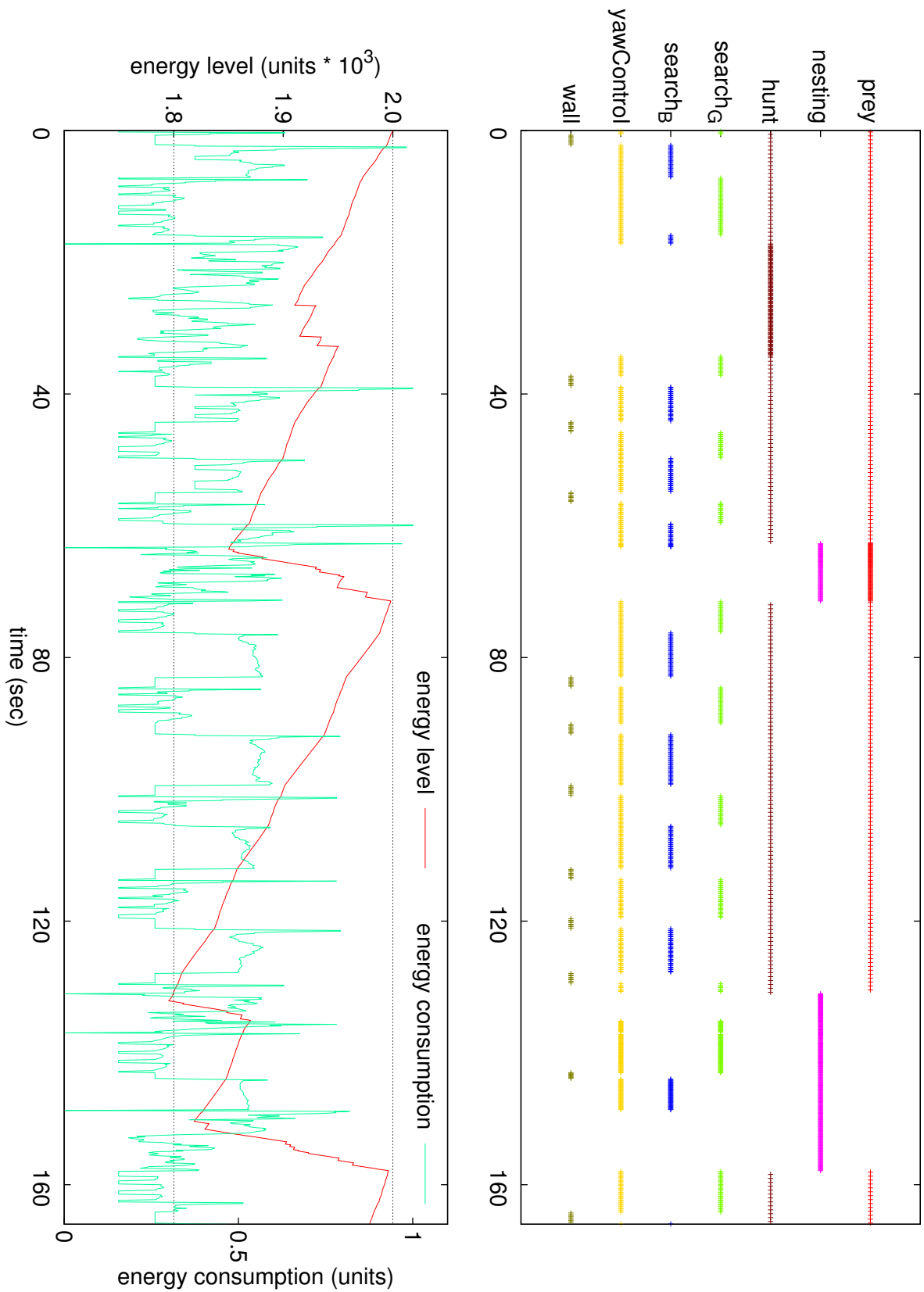


Figure 5.19: Biomimetic behaviour based control experimentation results, test A.

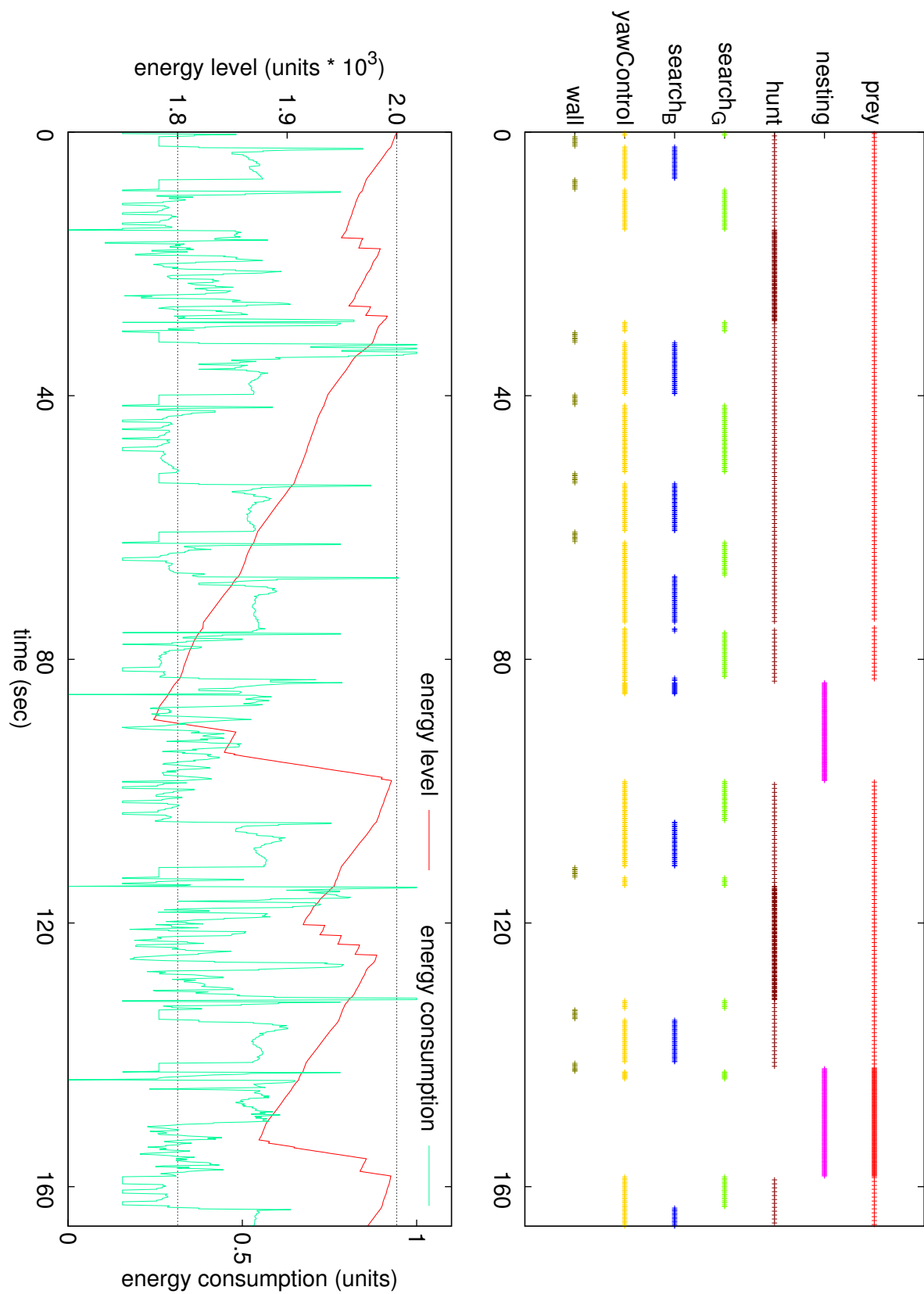


Figure 5.20: Biomimetic behaviour based control experimentation results, test B.



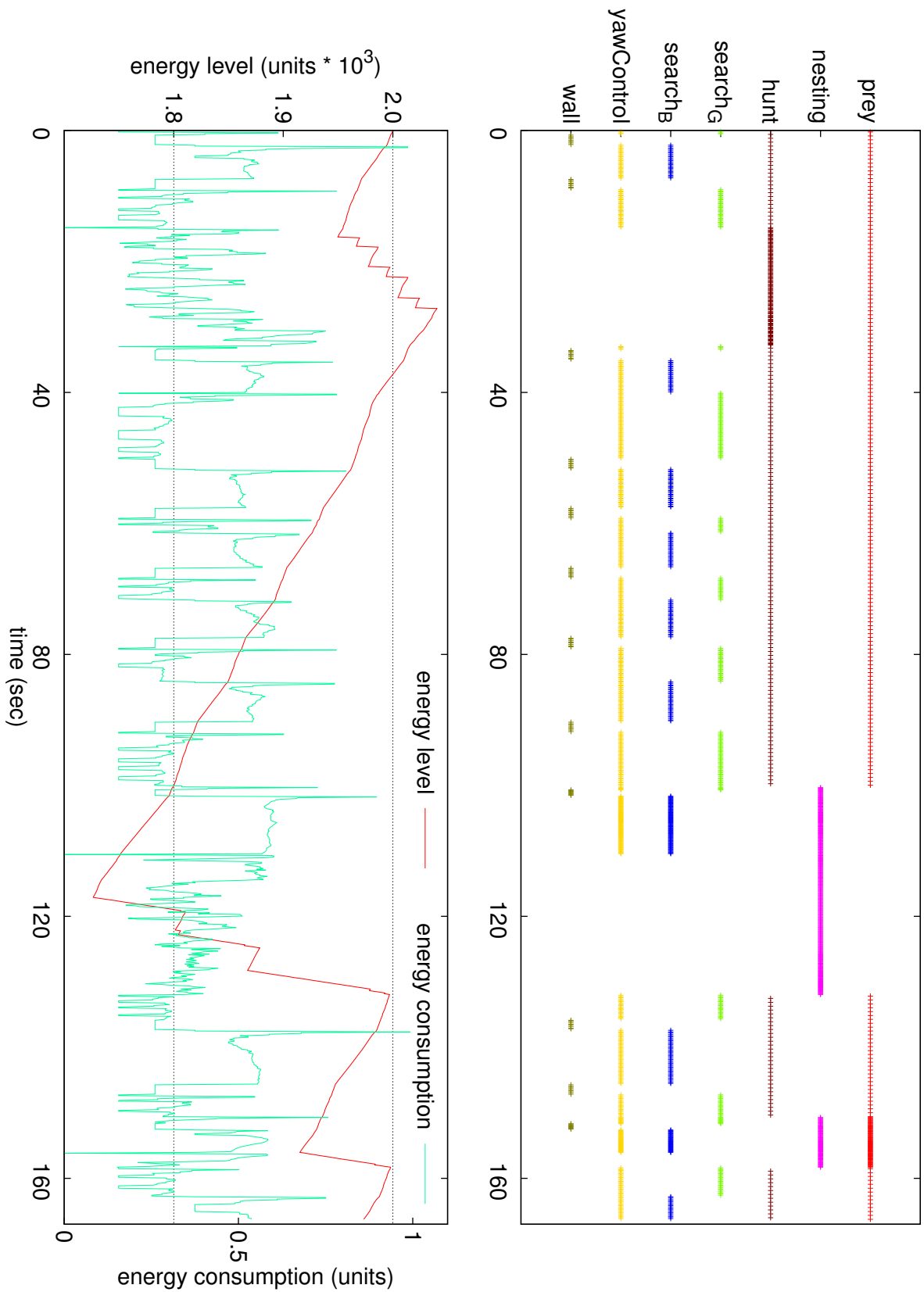


Figure 5.21: Biomimetic behaviour based control experimentation results, test C.

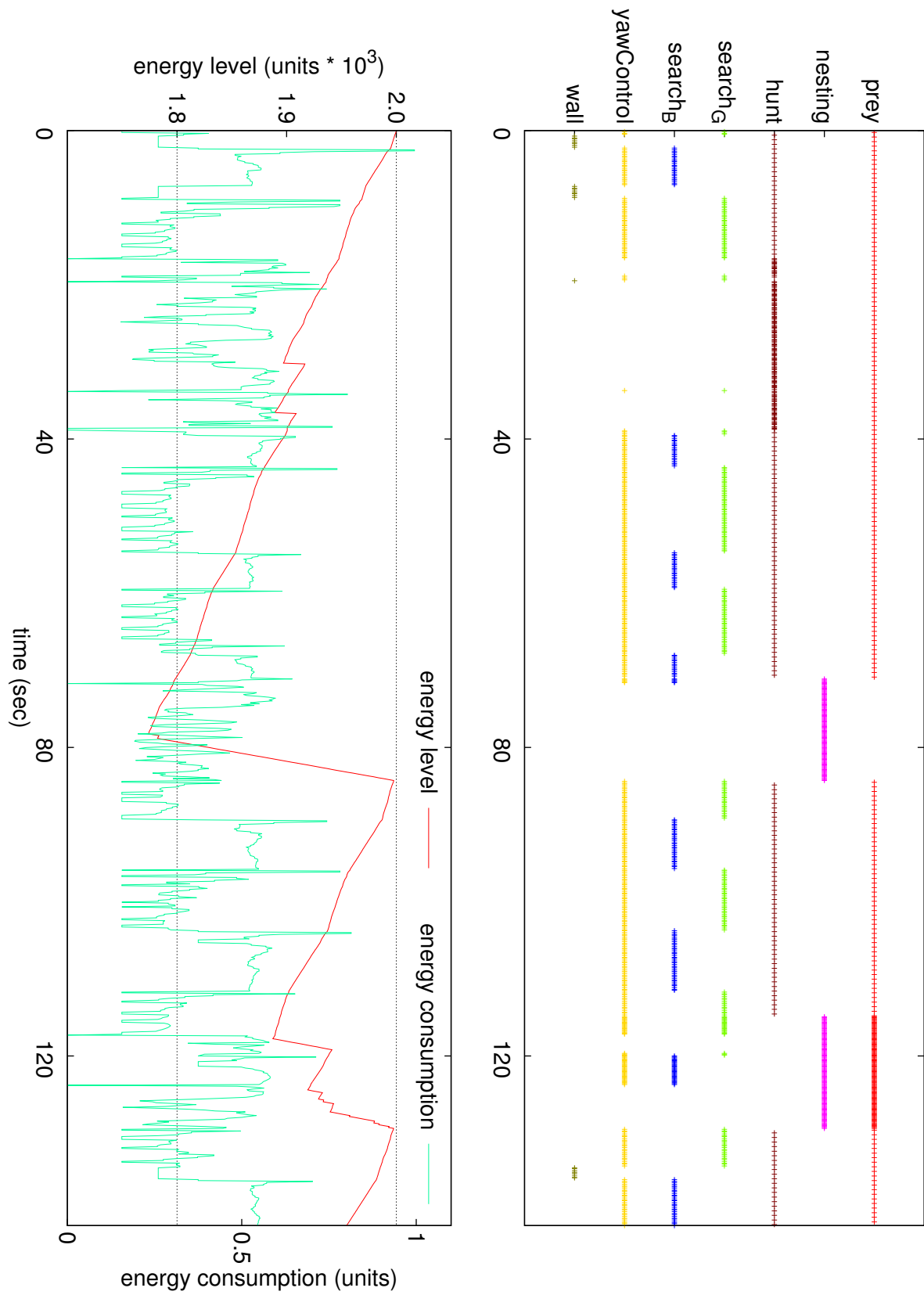


Figure 5.22: Biomimetic behaviour based control experimentation results, test D.

# Chapter 6

## Cooperative Behaviour

### 6.1 Behaviour Based Cooperative Control

This chapter describes the design and implementation of a behaviour based controller specially designed to emerge a cooperative behaviour among a team of AUVs. The successful development of a behaviour based control scheme for implementing and testing a biomimetic behaviour by *Ale III* AUV, was a motivation to extend the already developed scheme and add the potential for controlling more than one AUVs, being members of a team. The team exhibits a biomimetic cooperative behaviour.

#### 6.1.1 Animal Group Hunting

Cooperative or group hunting has been reported in several mammals [111], like *Lycaon pictus* African wild dogs [45], and even in one bird species, *Parabuteo unicinctus* or Harris' hawks [25]. Harris' hawks hunt practicing a *flush and ambush* strategy that involves a division of labour as occurring when individuals, working as a team to complete a task, perform different subtasks [11]. Cooperative hunting with a division of labour has been studied in pods of *Megaptera novaeangliae* humpback whales [42] or at group hunt of *Pan troglodytes* chimpanzees where the team members may engage in particular subtasks such as *driving* or *blocking* their prey [30]. Same behaviour appears in insects too as when a small group of army ants retrieves a prey item as a team [70].

Apart from humpback whales, cooperative hunting has been described in several cetaceans, including *Tursiops truncatus* bottlenose dolphins and *Orcinus orca* killer whales. Accounts of apparent cooperative behaviour in feeding bottlenose dolphins include fishes being herded into a ball [150], fishes driven ahead of dolphins swimming in a crescent formation, against mud banks or trapped between dolphins attacking from either side [117], [180]. Groups of dolphins may even beach themselves to feed on fishes that they have chased onto mud banks [143]. However, none of the previously described cases of group hunting in cetaceans demonstrates a division of labour with role specialization.

Role specialization is found when individuals specialize in their subtasks during repeated team tasks. Group hunting with a division of labour and role specialization is extremely rare. A well documented case in mammals is the study of coordinated group hunts in *Panthera leo* African lionesses [165]. Observation data from lionesses coordinated group hunts were analysed to assess cooperation and individual variation in hunting tactics. Group hunts generally involved a formation whereby some lionesses, the *wings*, circled prey while others, the *centres*, waited for prey to move towards them. Another case of group hunting with role specialization is the wolf pack hunting behaviour model, studied in packs of about 5 to 20 members, with a certain hierarchy in the *wander*, *stalk*, *formation*, *attack* and *eat* phases of the hunting procedure. The wolf pack is divided in two roles: one individual is the *alpha* wolf and the rest team members are *beta* wolves [176], [132]. On the other hand, [125] states that wolf packs' behaviour is not structured as a two role set of strategies, but rather as a generalized *rules of thumb* without role specialization among pack's individuals.

In Cedar Key, Florida, group hunting *Tursiops truncatus* dolphins engage in two types of behaviours while herding fishes. One individual in a group of three to six dolphins, the *Driver*, herds the fishes in circles, while the rest of the team members are the *non Driver* dolphins following the *Driver* and playing the role of a virtual barrier blocking fishes escape efforts [75]. Inside dolphins team the *Driver* is always the same individual while the non driving dolphins are defined as all the rest group members that did not drive.

A feeding bout begins with the initiative of *Driver* dolphin to swim rapidly in tight circles and finishes when the participating dolphins put their heads back under water and roll upright. *Driver* is the most skilled team member. It is capable of herding the fishes in circles. During hunting it leads and traps the school of fish towards the non driving- *barrier* members, that are less than one body length apart and often touching. When dolphins cooperate this way, fish hunting is easier and more beneficial. This extremely rare behaviour consists of team coordination acts like:

1. establishment of the team roles,
2. detecting the presence of *Driver* or the rest *non Driver* team members,
3. searching and navigation,
4. *Driver* following and
5. team foraging.

Next sections describe the implementation of a cooperative behaviour based model controller, based on *Ale III* atomic behaviour based ACoS version, with two different roles specialization: *Driver* (D), robot and *non Driver* (nD) robot team members. This cooperative behaviour is inspired from the bottlenose dolphin cooperative hunting. The behavioural model design explores and studies the two- roles cooperative processes, in terms of different interacting behaviours, used at each one the robot team members. As in the case of bottlenose dolphins, there is only one *D* dolphin inside the team, while the rest of the team members take over *nD* role specialization.

### 6.1.2 Mimicking Dolphins Hunting

The behaviour based design of the atomic ACoS version and the conclusions and observations regarding the bottlenose dolphins hunting procedure were the design basis for a novel control scheme: biomimetic cooperative behaviour that mimics dolphins' cooperating hunting bout basic concepts. Two different roles were designed and implemented with respect to the basic principles of Section 5.1. The first role represents the *D* driver dolphin while the second encapsulates the *nD* barrier or non driving dolphins role of the rest team members.

ACoS framework implemented the two cooperative roles as different *controller* classes, one for each role. *Robot* class is the same, as driver and non driver AUV's share the same I/O devices and operations. Each role is implemented by multi level hierarchy of interacting behaviours, in accordance to previous chapter's atomic role behaviour based subsumption architecture. Zeroth level's behaviour modules are common in both roles and are based in *yaw control* and *predator* modules belonging in the atomic ACoS version. The role discrimination begins at the higher role levels. First and second level at the driver role include different behaviour modules than the first and top non driver role, and both are atomic behaviour modules.

Cooperative ACoS implements in simulation all the distinct phases of the bottlenose dolphins' group hunting. The behaviour based model includes modules for each one of these phases. Thus, cooperative ACoS distinguishes two roles and directs the simulation progress according to the steps of a successful bottlenose group hunting:

1. *D* and *nD* team members demonstrate their identity by colour light signs.
2. During all simulation phases, the main tasks for each team member is searching and navigation. Searching target and navigation plan are set individually for each member according to its role and environmental stimuli, following the bottlenose dolphin group hunting scenario.
3. The first phase of the simulated hunt is the initialization, where *D* searches its surrounding area to detect candidate *nD* members.

4.  $nD$  members, right after the initialization phase follow  $D$  trying to stay close together.
5. The last phase of the simulated hunt takes place after detecting and approaching the target.  $D$  chooses a circular course trying to surround its target using  $nD$  followers as barriers.

## 6.2 Ale III Dynamic Mode

The real world experimentation, as described at the previous chapters, presents certain limitations:

- The size of the experimentation area makes it unsuitable for behaviour evolution and testing in terms of mechanical robustness, automatic evaluation and repetition of time consuming tasks [66]. Further, introduces inevitable limits regarding the number of different AUVs operating concurrently inside the tank. The experience gained throughout the experimentations showed that for every extra AUV an extra volume of at least  $1\text{m}^3$  should be added to the experimentation tank [145].
- The small number of available AUVs, as still these robots are costly and not suited for laboratory experimentation.

Due to these limitations the use of simulation is an attractive solution for studying ACoS cooperative behaviour scenario. Using simulated AUV models and the results from the previous chapter's experimentations on atomic behaviours control, an AUVs' team cooperative underwater predator behaviour, like the ones found bottlenose dolphins hunting bouts, was implemented.

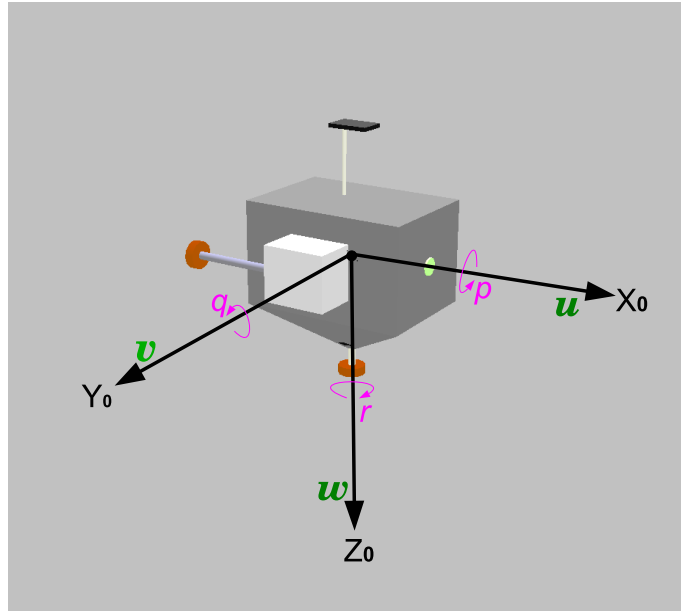


Figure 6.1: Vehicle velocity vector  $\nu = [u, v, w, p, q, r]^T$  in the Body Fixed coordinate system.

The simulated AUVs share the same specifications with the real AUV prototype. The simulation modelling was made using *Webots* simulation environment [175], [145]. The only addition made is a colour light sign on top of every *Ale III* for the exchange of simple messages between the different members of a team. *Ale III* underwater dynamics include the inertial generalized forces, the hydrodynamic effects, the gravity and buoyancy contribution and the thrusters' forces depicted by the following equation [69], [145]:

$$(M_{RB} + M_A)\dot{\nu} + C(\nu)\nu + D(\nu)\nu + G = T \quad (6.1)$$

where  $M_{RB}$  is the rigid body mass matrix,  $M_A$  is the added mass matrix,  $C(\nu)$  is the Coriolis and centripetal terms matrix,  $D(\nu)$  is the hydrodynamic damping matrix,  $G$  is the gravitational and buoyancy vector and  $T$  the thrusters external force and torque input vector.

At the simulated *Ale III* model,  $\nu$  is the velocity vector containing vessel's linear and angular velocities, estimated by the simulation software at each simulated time step that lasts 64msec.

$$\nu = [u, v, w, p, q, r]^T \quad (6.2)$$

The simulated  $6 \times 6$   $M_{RB}$  rigid body mass matrix includes vessel's mass, and inertial moments. It is estimated by Open Dynamics Engine [161], built in Webots simulation environment, according to the mass distribution of *Ale III* several parts. The  $6 \times 6$  added mass  $M_A$ , assuming that *Ale III* is a spherical vessel and its movement may be decomposed into three uncoupled movements along the body fixed coordination system, is approximated by the following diagonal matrix [69],

$$M_A = \begin{pmatrix} \pi\rho\alpha^2 & 0 & 0 & \cdots & 0 \\ 0 & \pi\rho\alpha^2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \quad (6.3)$$

where  $\pi = 3.14159$  is the mathematical constant,  $\rho$  is water density and  $\alpha$  is the assumed spherical vessel's radius.  $C(\nu)$  is the  $6 \times 6$  Coriolis and centripetal terms matrix that was not taken into account because a detailed, exact vessel model was not the scope of *Ale III* simulation. Moreover, the magnitude of Coriolis forces is quite smaller than the rest forces acting upon the vessel.  $D(\nu)$  is the hydrodynamic damping matrix including linear and quadratic damping coefficients. The fact that hydrodynamic drag is always opposed to vessel's movement explains the minus sign in Equation 6.4. The simulated model uses an uncoupled model of damping that is a rough approximation working sufficiently well in the case of *Ale III* low speed simulation [103].  $D(\nu)$  quadratic terms are used to estimate the uncoupled linear velocity damping and the linear terms are used for the uncoupled rotational velocity damping [192]:

$$D(\nu) = -diag \begin{pmatrix} D_{u^2} |u| \\ D_{v^2} |v| \\ D_{w^2} |w| \\ D_p \\ D_q \\ D_r \end{pmatrix} = -diag \begin{pmatrix} 1/2 \times \rho C_{D_u} A_u |u| \\ 1/2 \times \rho C_{D_v} A_v |v| \\ 1/2 \times \rho C_{D_w} A_w |w| \\ D_p \\ D_q \\ D_r \end{pmatrix} \quad (6.4)$$

where,  $C_{D_u}$ ,  $C_{D_v}$ ,  $C_{D_w}$  are the quadratic drag coefficients for the uncoupled velocities of surge, sway and heave,  $A_u$ ,  $A_v$ ,  $A_w$ , are the projected frontal areas of AUV body to the surfaces  $Z_0Y_0$ ,  $Z_0X_0$ ,  $X_0Y_0$  of Figure's 6.1 body fixed coordinate system and  $D_p$ ,  $D_q$ ,  $D_r$  the linear rotational velocities drag terms. In the case of *Ale III* simulation the quadratic drag coefficients values and the the linear rotational velocities drag terms were estimated geometrically [154], [192]:  $C_{D_u}=1.05$ ,  $C_{D_v}=1.05$ ,  $C_{D_w}=0.95$ ,  $D_p=0.015$ ,  $D_q=0.015$ ,  $D_r=0.01$ . The hydrodynamic effects are implemented at every simulated step by estimating and applying to vessel's simulated rigid body dynamics these linear and quadratic drag terms. The gravitational part of vector  $G$  is computed by the simulation physics engine, given the mass and relevant positioning of vessel's several parts as shown at Figure 6.2. The simulated buoyancy effect is composed of three forces:

- vessel's hull buoyancy, computed by its volume and applied at hull's centre of volume.
- left and right thruster enclosures buoyancy, computed by their volume and applied at the left and right thruster enclosures' centre of volume respectively.

The input vector  $T$ , comprising of thrusters external force and torque is also estimated by the simulation physics engine ODE, given the propelling force acting upon each thruster. According to experiments with the

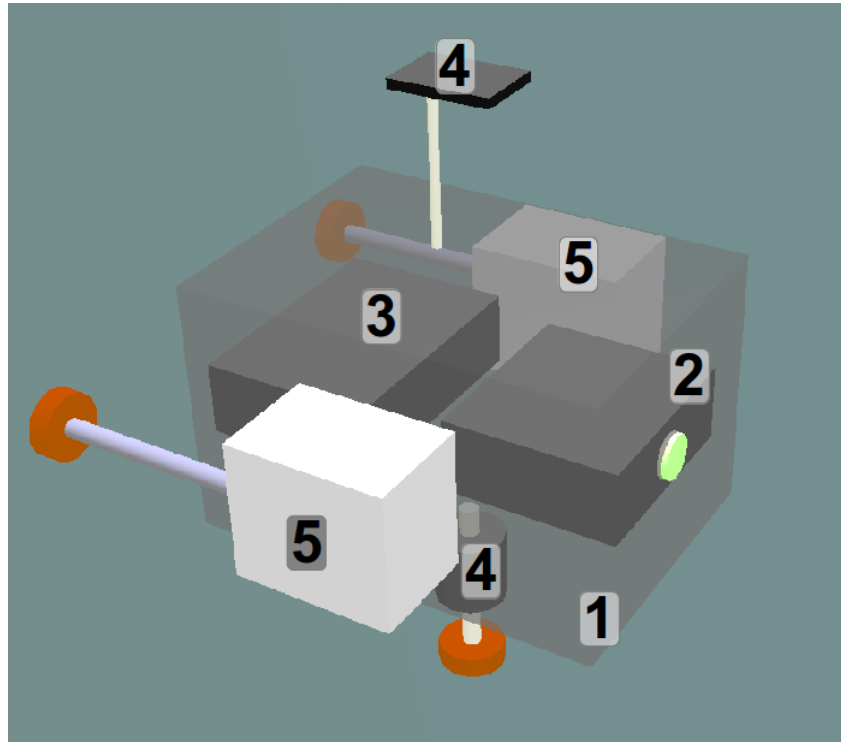


Figure 6.2: Simulated *Ale III* (1) watertight hull, (2) stem assembly module, comprised of the vision sensor, the motor drivers and the motors battery pack. (3) stern assembly module comprised of the power supply circuits, the COM and the electronics battery pack, (4) bottom thruster and ballast, (5) lateral thrusters, (6) IMU.

real *Ale III*, the maximum propeller thrust of each motor is 0.6Nt. The simulated ACoS computes three output values, one for each thruster, in terms of propelling power, ranging from 0Nt to 0.6Nt. These are the propelling forces used by ODE at each controller time step to estimate  $T$  and move the vessel.

Webots simulation environment also implements the distortion caused to visual light undersea. Water is a substance more than 800 times denser than air. As soon as light enters the water, it interacts with the water molecules and suspended particles to cause loss of light, colour changes, diffusion and loss of contrast. At *Ale III* simulated environment the undersea light distortion has been implemented with a linear visibility range of 25m. Objects inside range are drawn blended with a turquoise sea water colour.

### 6.2.1 Simulation software architecture

Webots simulation framework comprises of:

**A controller program** for each one of the simulated world robots. It implements the robot controller and handles all the I/O operations between the robot and its sensors. When a simulation starts, Webots launches the specified controllers, each as a separate process, and it associates the controller processes with the simulated robots. If several robots can use the same controller code, then a distinct process will be launched for each robot.

**A physics plug in** for the implementation of physical effects by accessing the low level API of the ODE physics engine built in Webots simulation Environment [161]. Physics plug- in is a user implemented shared

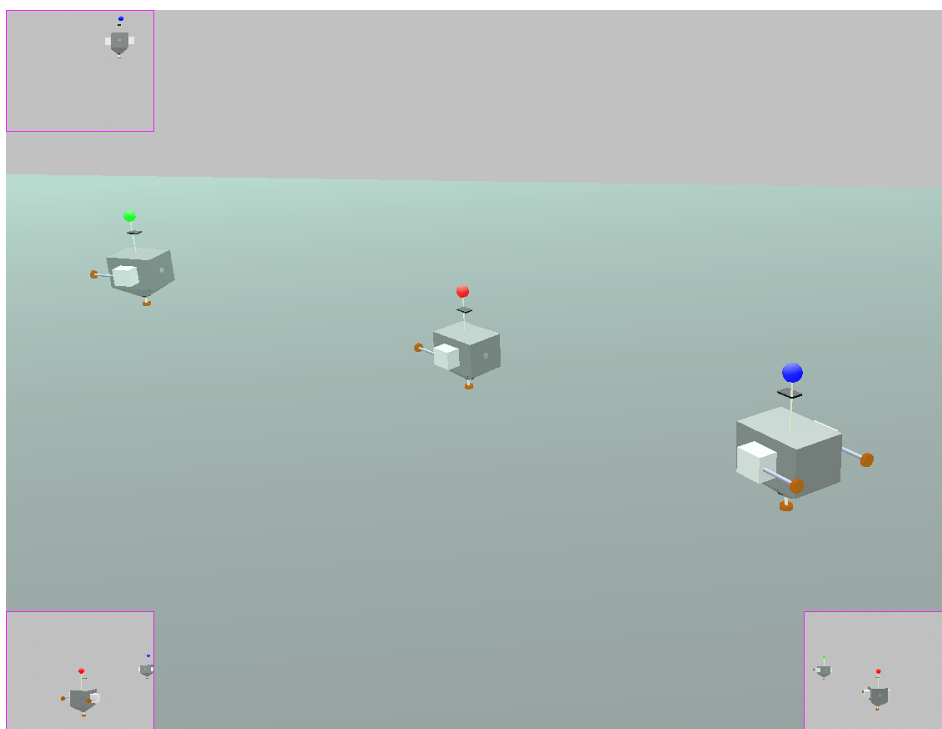


Figure 6.3: *Ale III* AUV team inside Webots simulation environment. The three small frames at picture corners are the views of each vehicle's vision sensor.

library which is loaded by Webots at run time and accesses information about the simulated bodies position, orientation, linear or angular velocity, adds forces, torques and extra joints to a simulation.

*Ale III* simulation involves two different controller programs, one for  $D$  and one for the  $nD$  cooperative behaviour role. Each program comprises the *real* life ACoS framework, incorporating a *Robot* and a *Controller* class [145]. *Robot* class handles I/O operations, has access to all robot resources and is common at both roles. *Controller* class implements the cooperative behaviour based ACoS scheme. In *Ale III* simulation there is always only one robot in the simulated team executing a  $D$  controller, while all rest team members execute the same  $nD$  controller program in distinct simulation processes. A physics plug-in, developed specifically for *Ale III* simulation, encapsulates the vessel's dynamic model described at Section 6.2.

The basic ACoS concepts migrated to the simulation environment were adapted to the Webots API specifications. The novel ACoS cooperative behaviour scheme comprises of the behaviour modules developed for each role, described at the following section.

### 6.3 Cooperative Behaviour Architecture

The cooperative predator behaviour subsumption architecture, developed for the emergence of cooperative behaviour inside an *Ale III* team simulation, in analogy to the bottlenose dolphins hunt with division of labour and role specialization, is distinguished in two roles,  $D$  and  $nD$ .  $D$  role is organized as a three level hierarchy, consisting of six different interacting behaviours in total.  $nD$  role is also organized in three levels, consisting of five  $nD$  behaviours. Due to readability reasons a three member simulated team,  $D$ ,  $nD1$  and  $nD2$  is assumed in the following description.



### 6.3.1 Driver role

#### Level D0

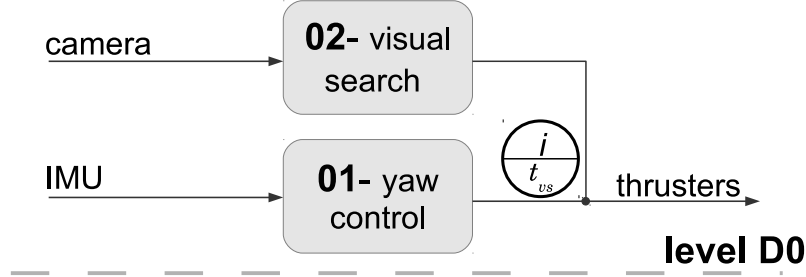


Figure 6.4: Zero level *D0* control architecture.

The zero level diagram, Figure 6.4, contains the following two behaviours:

**D01- yaw control**, tries to achieve an orientation of fixed yaw value.

**D02- visual search**, navigates the AUV towards a target identified using the visual information from the vision sensor module.

*Yaw control* and *visual search* are the only navigational behaviours of the proposed *D* control architecture. The circular symbol at the output of *yaw control* behaviour, Figure 6.4, is an inhibitor. Its meaning is that when *visual search* behaviour produces an output to control AUV's thrusters, it inhibits *yaw control* output. Thus, when a visual target has been identified by the controller, *visual search* is the only behaviour to undertake the navigation task. The use of letter *i* at the upper hemicycle of the inhibitor symbol, depicts the inhibition, while  $t_{us}$  at the bottom hemicycle denotes the inhibition time in controller step numbers. The duration of  $t_{us}$  lasts as long as *visual search* behaviour keeps a light target locked on camera's field of view.

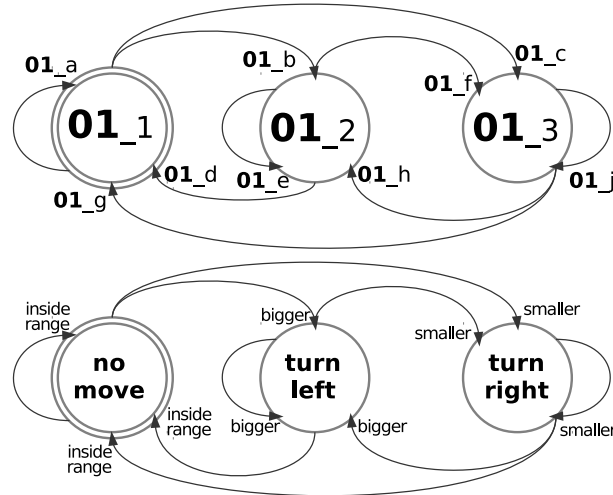


Figure 6.5: *D01- yaw control* behaviour's FSM diagram.

**Behaviour D01- yaw control** *Yaw control* is a navigational behaviour. It uses data from the IMU sensor module to estimate AUV's yaw heading and manipulates robot's lateral thrusters, trying to maintain a fixed yaw orientation value. *Yaw control* behaviour's FSM, shown in Figure 6.5, consists of the following states.

- **01.1:** AUV is not performing any turning movement- no lateral thruster power manipulation.
- **01.2:** AUV is turning left, around its vertical, body oriented axis.
- **01.3:** AUV is turning right, around its, vertical, body oriented axis.

IMU sensor inputs trigger the transitions between the above states as follows:

- **01.a, 01.d, 01.g:** the yaw heading value, estimated by the robot's IMU module, lies between the correct limits, around a predefined target value.
- **01.b, 01.e, 01.h:** yaw value is bigger than the target value, outside the correct limits.
- **01.c, 01.f, 01.j:** yaw value is smaller than the target value, outside the correct limits.

*Yaw control* behaviour module utilizes a *PD* controller for yaw rotation. To test the controller efficiency a simulated experimental scenario was implemented where an initially standstill, 0rad heading *Ale III* vessel moves using solely the *yaw control* behaviour to achieve a 3rad yaw heading target value (Figure 6.9). Figures 6.6, 6.7, 6.8 show results for different  $K_P$  and  $K_D$  controller gain parameter values, demonstrating the controller's performance. In each figure the plot at the top adds a '+' mark at every controller step for each behaviour module that takes part to the determination of output commands and shows behaviour modules interaction. The second plot from top shows the PD controller output value for the left thruster, along with yaw error values. The third and fourth plot from top show the uncoupled P and D controller term respectively. Figure 6.6 shows a well tuned efficient yaw controller while next two figures show the predictable experimentation results when P or D gain parameters are altered.

**Behaviour D02- visual search** *Visual search* uses the camera sensor module to seek a predefined light colour target and lock it inside its field of view. It is a navigational behaviour that inhibits *yaw control* output and navigates the AUV, for as long as the visual target that has been detected remains inside the camera's field of view. An identified visual target inside the simulation environment may be a:

- *nD* team mate. Every *nD* team member is equipped with an omnidirectional light, suspended 10cm above its hull. The light is turned on whenever the *nD* member has detected and locked inside its camera's field of view the leader *D*. For the purposes of the current simulation scenario *nD1* is equipped with green omni directional light while the other, *nD2*, has a blue one. During the simulated feeding bout initialization phases the team driver *D* searches for possible teammates, trying to find their lights.
- a self luminous, spherical shaped, yellow coloured school of fish, called prey.

*Visual search* behaviour's FSM, Figure 6.10, consists of the following states.

- **02.1:** seeking a light prey- target of a predefined colour.
- **02.2:** the coloured light target has been detected- trying to lock the target inside the camera's field of view.

Triggering conditions for the transitions between the above states:

- **02.a:** prey has been detected.
- **02.b:** prey has not been detected.
- **02.c:** prey has been locked.
- **02.d:** prey has not been locked.
- **02.e:** prey lost.

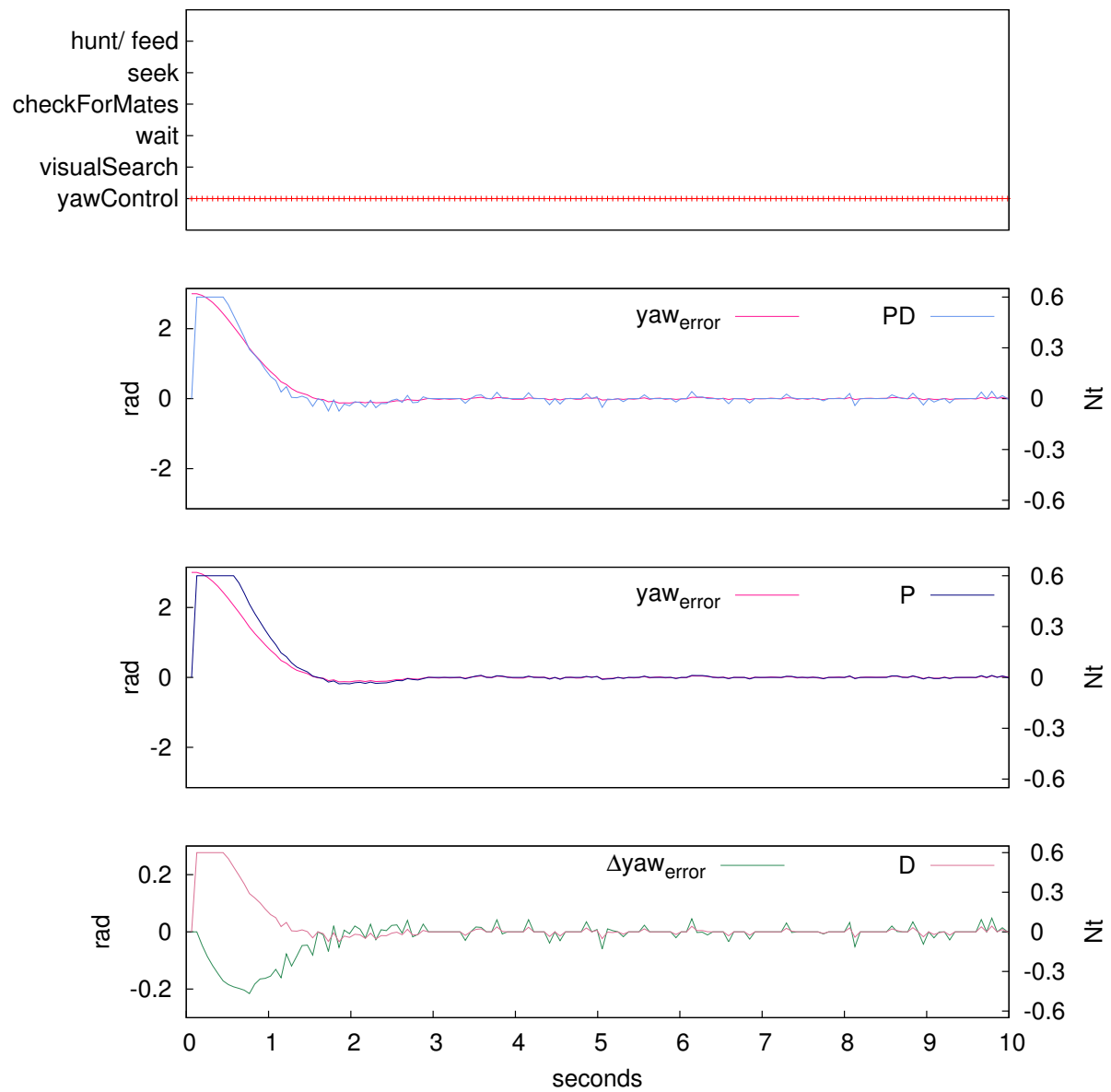


Figure 6.6: *D01- yaw control* experimental results, test A. The controller gain parameters are  $K_P=0.5$  and  $K_D=7 \times 10^{-3}$ .

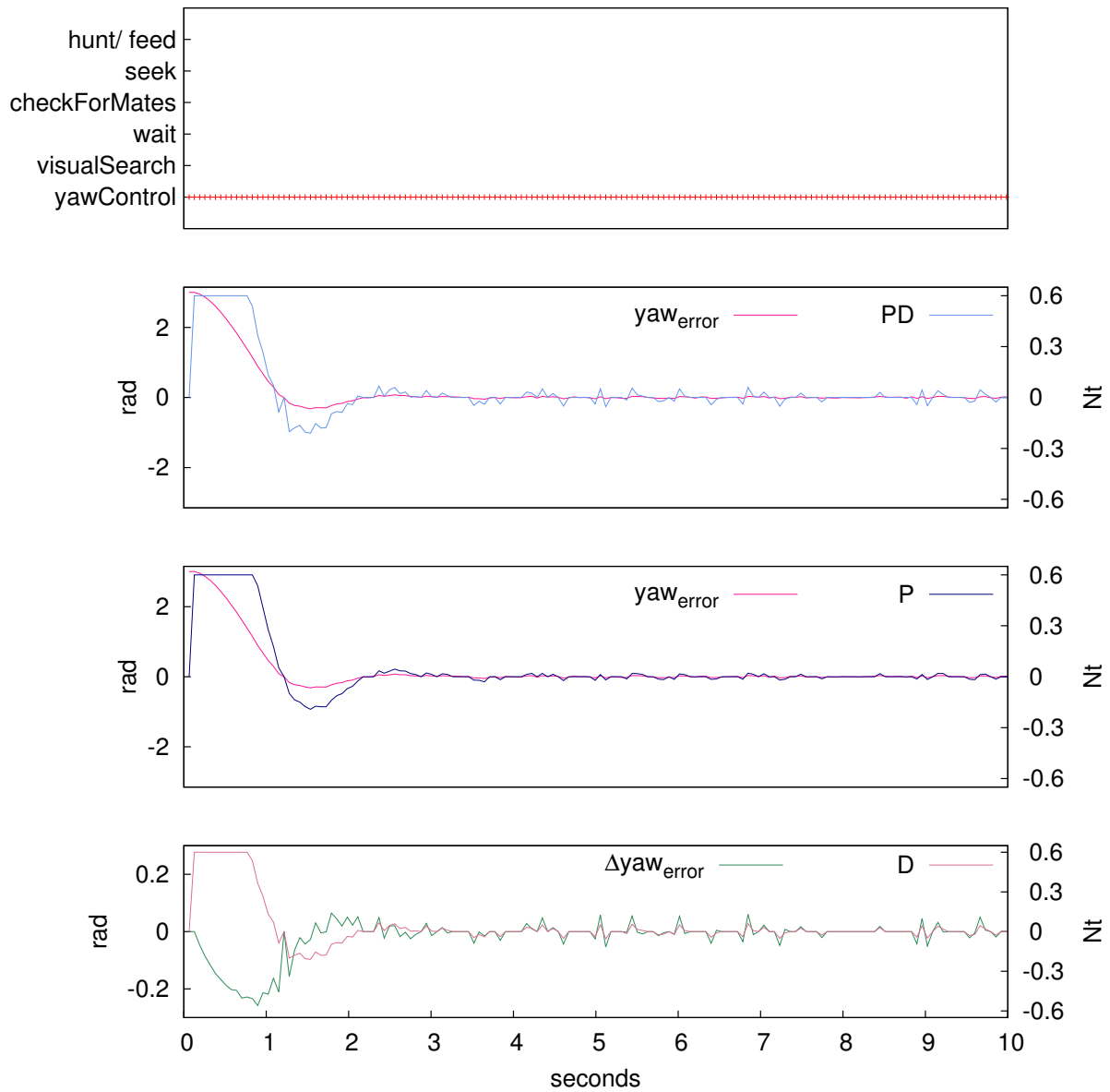


Figure 6.7: *D01- yaw control* experimental results, test B. The controller gain parameters are  $K_P=1$  and  $K_D=7 \times 10^{-3}$ . As expected, doubling Figure's 6.6  $K_P$  value results in a more intensive oscillation around the target values.

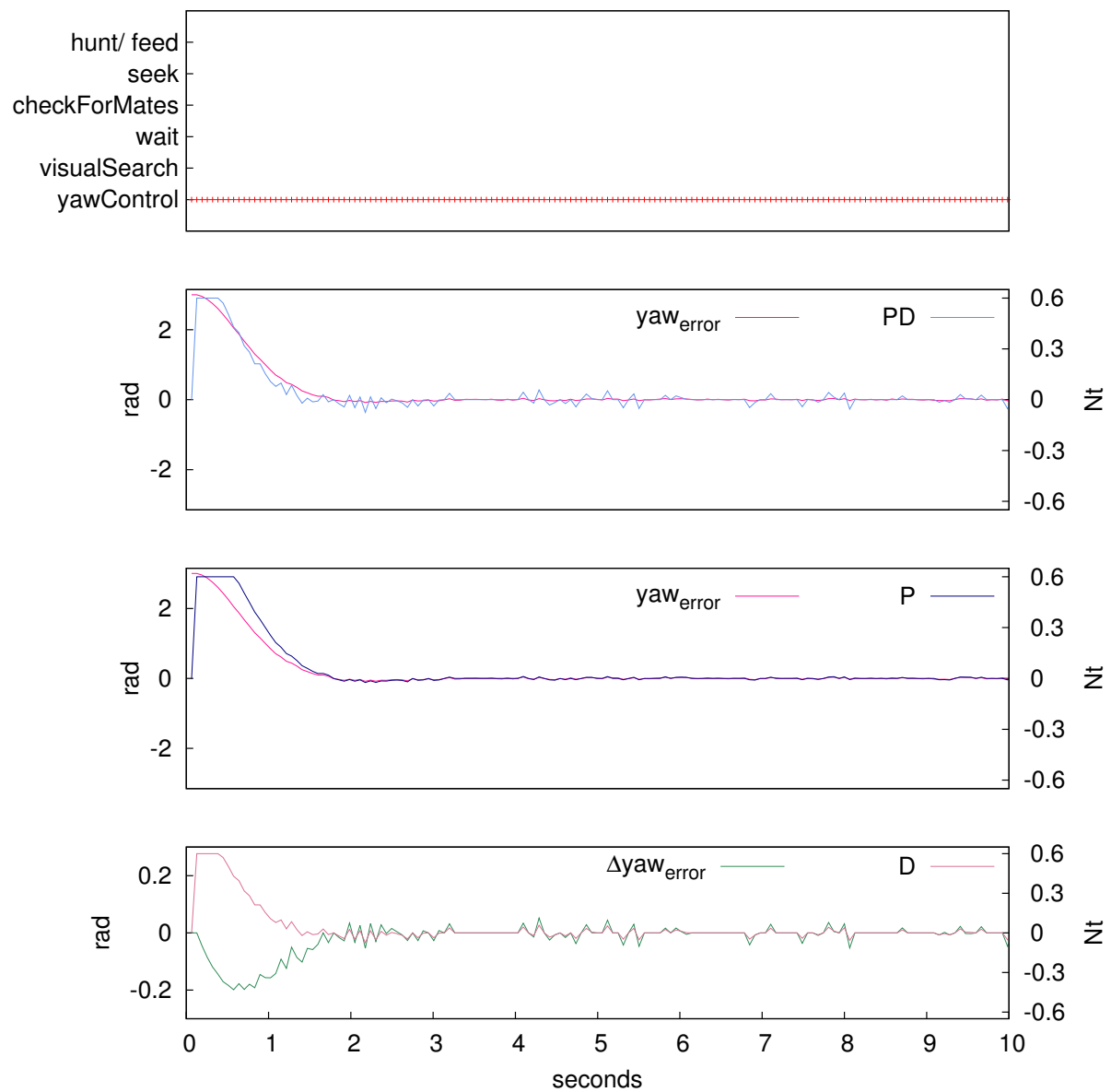
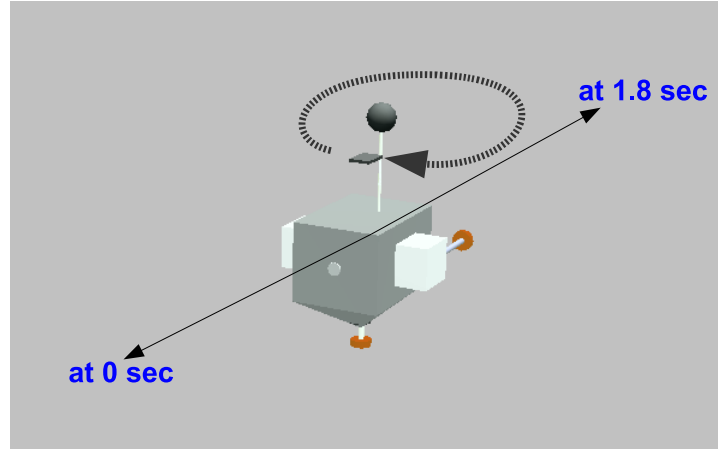
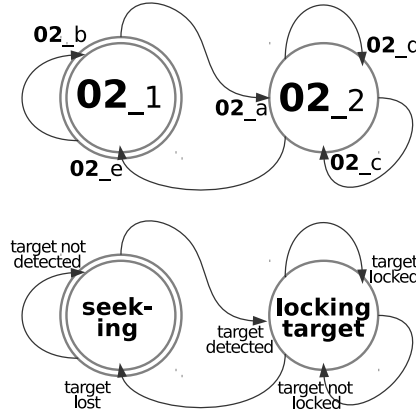


Figure 6.8: *D01- yaw control* experimental results. The controller gain parameters are  $K_P=0.5$  and  $K_D=1.4 \times 10^{-2}$ . As expected doubling Figure's 6.6  $K_D$  value results in less nervous controller reflexes.

Figure 6.9: *D01- yaw control* experiment.Figure 6.10: *D02- visual search* behaviour's FSM diagram.

In the *visual search* behaviour module a PD controller takes over seeking and target locking procedures. To test the controller, a simulated experimental scenario was implemented. According to the scenario an, initially standstill, AUV locks and follows a moving light target using this PD controller. After ten seconds the light target changes its direction (Figure 6.14). Figures 6.11, 6.12, 6.13 show the results for different  $K_P$  and  $K_D$  controller gain parameter values demonstrating controller's performance. As previously, in each figure the plot at the top adds a '+' mark at every controller step for each behaviour module that takes part to the determination of output commands and shows behaviour modules interaction. The second plot from top shows the uncoupled P, D controller left thruster output values for the correction of  $x_\varepsilon$  target's centroid horizontal error, along with  $x_\varepsilon$  target's centroid horizontal error. The third shows the uncoupled P, D controller bottom thruster output values for the correction of  $\psi_\varepsilon$  target's centroid vertical error, along with  $\psi_\varepsilon$ . The fourth plot shows the uncoupled controller outputs for the correction of  $\alpha_\varepsilon$  targets detected pixels number, and  $\alpha_\varepsilon$ .  $\alpha_\varepsilon$  implies the distance between the vessel and the target. Figure 6.11 shows a well tuned efficient yaw controller while next two figures show the predictable experimentation results when P or D controller gain parameters are altered.

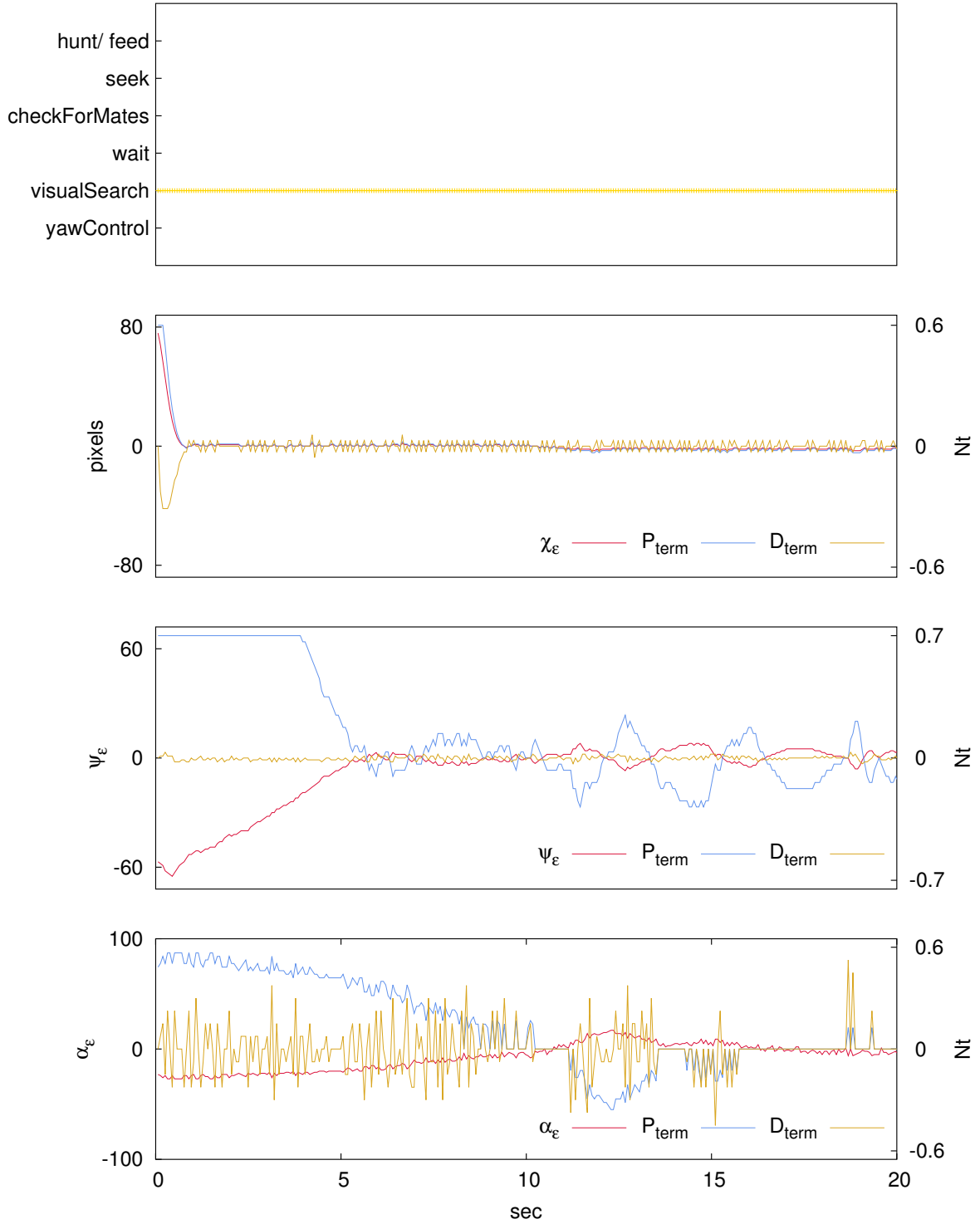


Figure 6.11: *D02- visual search* experimental results, test A: the AUV locks a moving light target that follows a curved, three dimensional trajectory. At the 10th second the light target changes its direction. The controller gain parameters for  $x_\epsilon$  are  $K_P=1.8 \times 10^{-2}$  and  $K_D=3 \times 10^{-3}$ , for  $\psi_\epsilon$  are  $K_P=0.5 \times 10^{-1}$  and  $K_D=0.1 \times 10^{-1}$  and for  $\alpha_\epsilon$  are  $K_P=3.5 \times 10^{-2}$  and  $K_D=8 \times 10^{-3}$ .

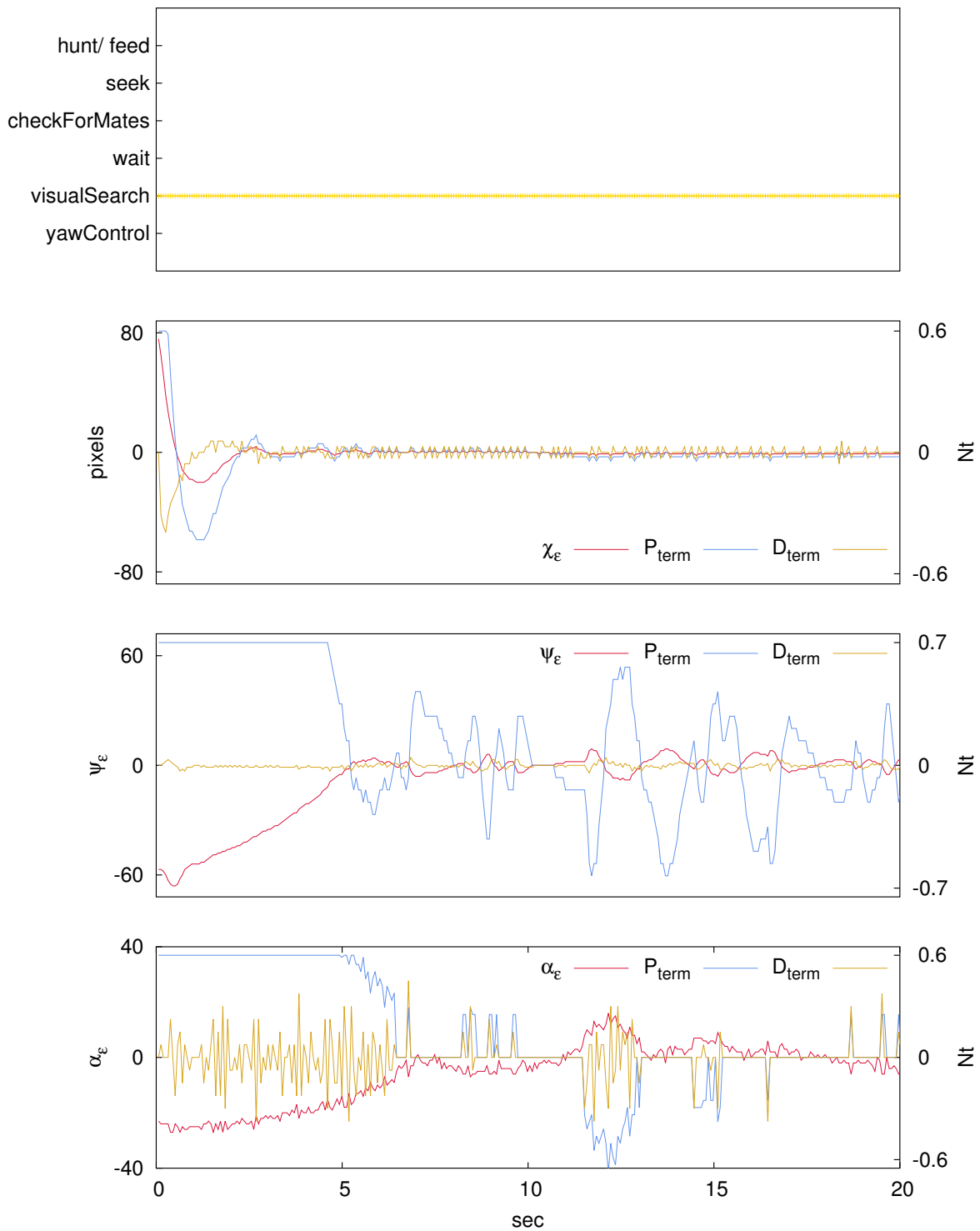


Figure 6.12: D02- visual search experimental results, test B. All controller parameters are the same apart from  $K_P$  values that is double for the correction of  $x_\epsilon$ ,  $\psi_\epsilon$  and  $\alpha_\epsilon$ . As expected, doubling Figure's 6.11  $K_P$  values results in a more intensive oscillation around the target values.



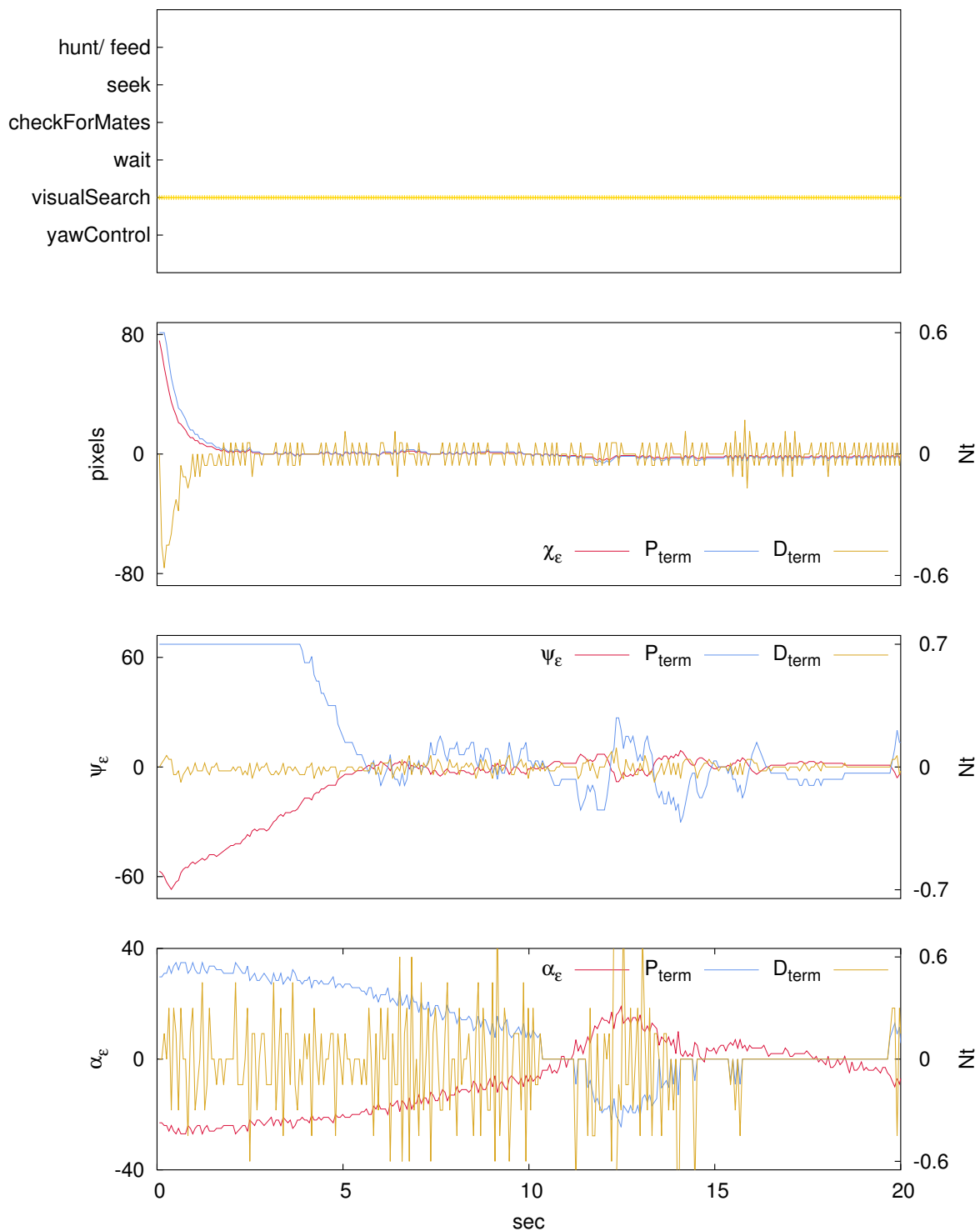
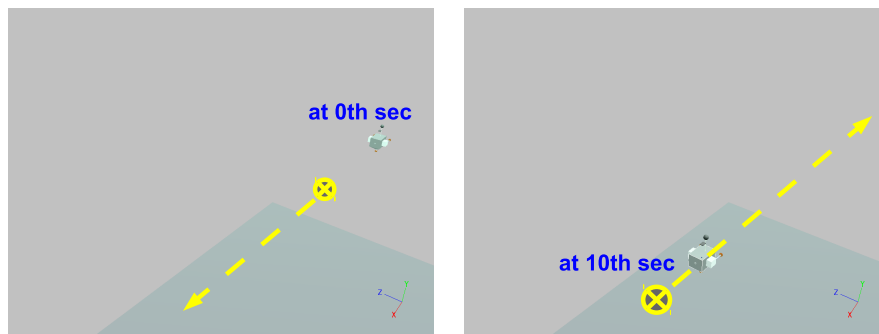
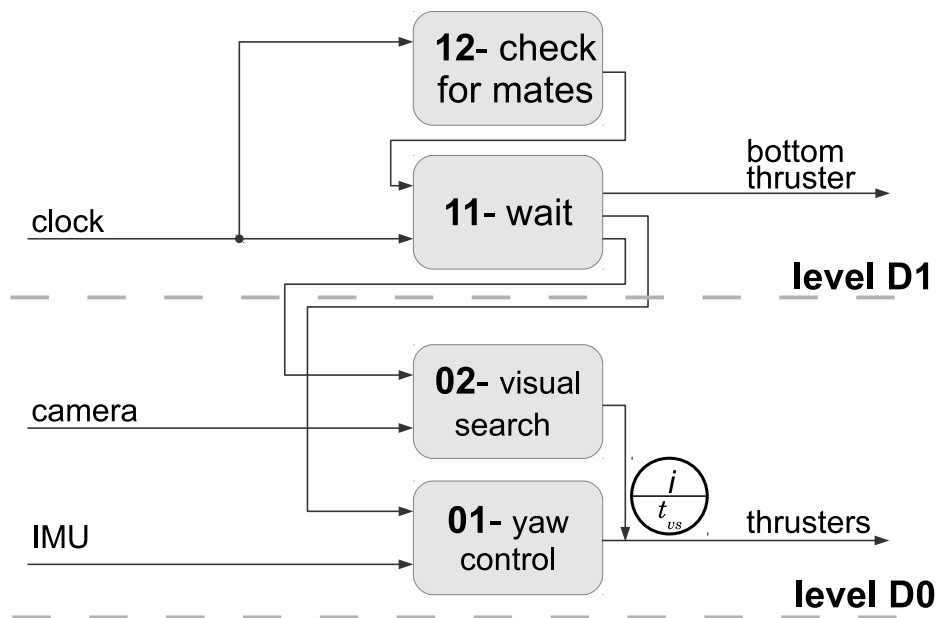
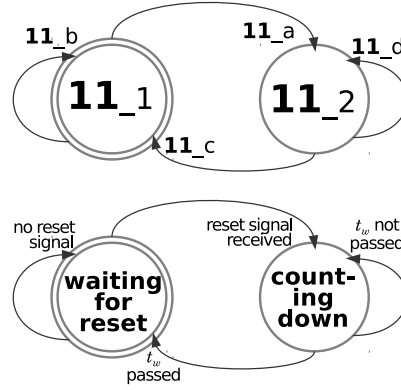


Figure 6.13: *D02- visual search* experimental results, test C. All controller parameters are the same apart from  $K_D$  values that is double for the correction of  $x_\epsilon$ ,  $\psi_\epsilon$  and  $\alpha_\epsilon$ . As expected doubling Figure's 6.11  $K_D$  values results in less nervous controller reflexes.

Figure 6.14: *D02- visual search* experiment.Figure 6.15: Level *D1* control architecture.

## Level D1

Level 1 behaviours, Figure 6.15, initialize each simulated bout. *Check for mates* verifies that at least two *nD* AUVs have joined the team. If this is not the case, the bout start up is delayed, for a predefined time period, by the *wait*. After that, *check for mates* repeats the procedure. As shown at Figure 6.15, *wait* behaviour subsumes 01- *yaw control* and is subsumed by the, same level's, behaviour *check for mates*.

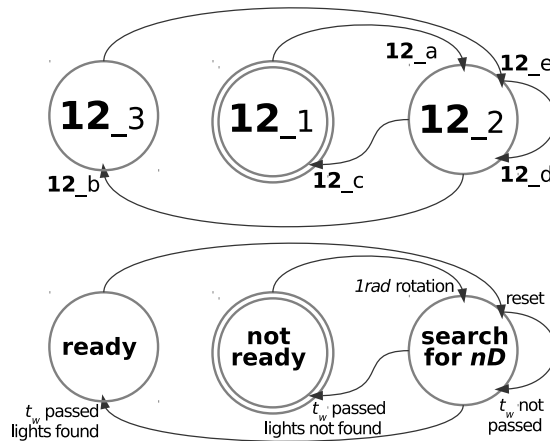
Figure 6.16: *D11- wait* behaviour's FSM diagram.

**Behaviour D11- wait** During bout's initialization phase, the leader *D* waits for the rest of the *nD*- mates to join the team. With the help of the subsumed *D02- visual search* it detects the presence of *nD* members, searching for their coloured lights. The interval of waiting is  $t_w$  controller time steps. At the current simulation scenario,  $t_w=40$  controller steps or 2.56sec. During this interval, *wait* behaviour aims at keeping *D*'s absolute position as stationary as possible. That is why, apart from using the subsumed *yaw control* behaviour for keeping a steady yaw orientation, it also manipulates the bottom thruster to counterbalance AUV's, slightly, positive buoyancy. *Wait* behaviour's FSM (Figure 6.16) consists of the following states:

- **11.1:** waiting for a countdown reset signal,
- **11.2:** countdown for a predefined period of  $t_w$  controller time steps,

and the following transitions and sensor inputs:

- **11.a:** a countdown reset signal has been received,
- **11.b:** a countdown reset signal has not been received yet,
- **11.c:**  $t_w$  time period has passed,
- **11.d:**  $t_w$  time period has not passed yet.

Figure 6.17: *D12- check for mates* behaviour's FSM diagram.

**Behaviour D12- check for mates** At the simulation scenario described in this chapter, there are two  $nD$  team members. A bout is initialized successfully if  $D$  detects them, with their lights turned on, denoting their readiness for hunting.  $nD1$  is equipped with green omni directional light while  $nD2$  has a brown one. Every  $t_w$  controller time steps *check for mates* changes AUV's yaw orientation, rotating it clockwise, by an angle of  $1rad$ . Thus, *check for mates*, subsuming *wait* behaviour, scans for the *nonDrivers'* lights. If both, green and brown, lights are detected then everything is ready for a new hunting. The leader  $D$  starts moving, choosing a searching course and so the bout is beginning. *Check for mates* behaviour's FSM (Figure 6.17) consists of the following states:

- 12.1: not yet ready for bout initialization,
- 12.2: yaw rotating- searching for green and brown lights,
- 12.3: bout initialized,

and the following transitions and sensor inputs:

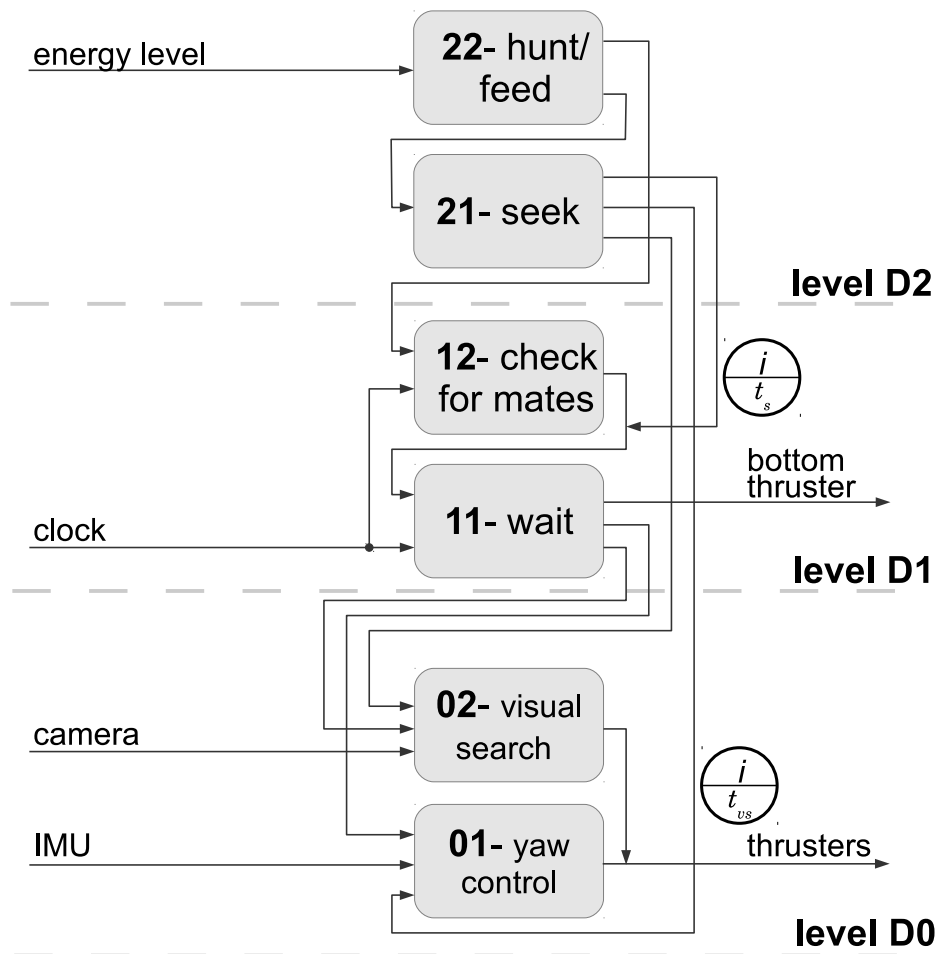
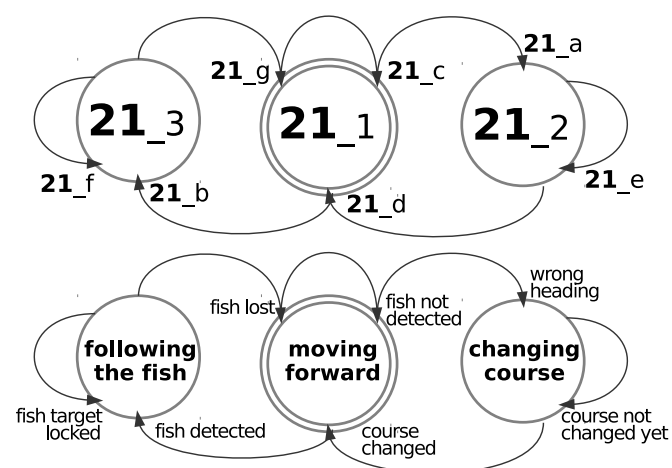
- 12.a:  $1rad$  yaw rotation has been accomplished,
- 12.b:  $t_w$  controller steps passed and  $nD$  lights found,
- 12.c:  $t_w$  controller steps passed and  $nD$  lights not found,
- 12.d:  $t_w$  not completed yet,
- 12.e: hunting procedure reset signal, whenever a hunting bout has to be initialized.

## Level D2

Level 2 implements the two higher hierarchy hunting behaviours:

**D21- seek**, searches for the simulated school of fish. Subsumes behaviours 01- *yaw control* and 02- *visual search*, Figure 6.18, to detect fishes by the yellow colour and spherical shape of their schooling cluster. Immediately after bout initialization, performed by 12-*check for mates*, *seek* behaviour chooses a course following a predefined pattern, trying to explore AUV's surrounding environment. This is the reason for the subsumption of *check for mates* lower level behaviour output, lasting  $t_s$  controller time steps, where  $t_s$  is the time needed for the bout procedure to complete.

**D22- hunt/ feed**, after the bout target has been detected, *hunt/ feed* subsumes *seek* behaviour and chooses a circular course, trying to surround the simulated school of fish. The rest of the team members, i.e. the two  $nD$  followers, play the role of barriers.  $D$  leads the foraging team to its feeding target, implementing circular courses.

Figure 6.18: Level *D2* control architecture.Figure 6.19: *D21- seek* behaviour's FSM diagram.

**Behaviour D21- seek** The course followed by *seek* behaviour may be a meandrous or spiral shaped, Figures 6.20, 6.21. It subsumes *D01- yaw control* and *D02- visual search* to seek for the simulated school of fish. During this searching period,  $t_s$ , *seek* inhibits the output of the lower level behaviour *D12- check for mates* and thus prohibits all the rest, visual based, behaviours from taking over thrusters' control and stop the hunting. Behaviour's FSM (Figure 6.19) consists of the following states:

- 21.1: moving forward- search for fish,
- 21.2: changing course following a predefined pattern,
- 21.3: following the school of fish,

and the following transitions and sensor inputs:

- 21.a: vessel's heading is outside course,
- 21.b: school of fish has been detected,
- 21.c: school of fish has not been detected,
- 21.d: course changed,
- 21.e: course change has not been completed yet,
- 21.f: school of fish target locked,
- 21.g: school of fish target lost.

Driver *D* uses two different navigation mechanisms to scan the surrounding underwater area. The first, Figure 6.20, selects an initial heading and follows it using a meandrous course. At Figure's 6.20 simulated experiment *D* uses a meandrous pattern to maintain a resultant  $0^\circ$  heading. This mechanism explores half AUV's *horizon*, proceeding towards a preselected heading. That is why it is efficient when there is an anticipation about target's position. The second mechanism implements an incremental radius spiral course, Figure 6.21. This method explores AUV's *horizon* in total and is the default scanning method used by *seek* behavior. At both methods the AUV, successively, emerges and submerges between -6m and -2m, that is near the surface. Near the surface is where the target is gathered to form school of foraging fishes and is simulated by a yellow colour sphere with a radius of 3m.

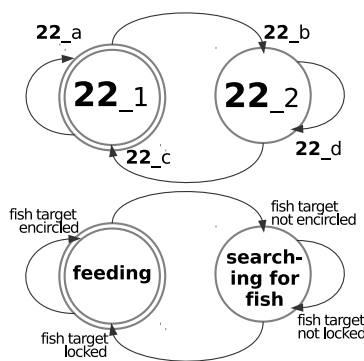
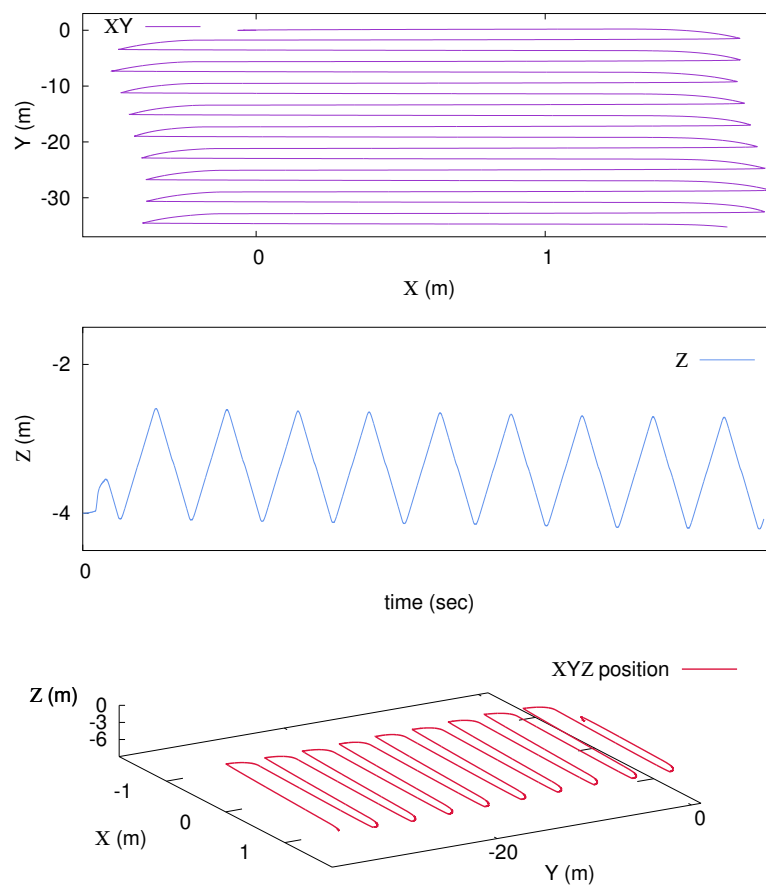


Figure 6.22: *D22- hunt/ feed* behaviour's FSM diagram.

**Behaviour D22- hunt/ feed** leads to the accomplishment of a successful simulated hunting bout. Bout's leader, *D*, leads all the rest  $nD$  team members to an encircling course, surrounding the detected school of fish. Feeding is simulated by repetitious circular movements around the fishes.  $nD$  followers play the role of barriers to block fishes' break away attempts. Behaviour's FSM (Figure 6.22) consists of the following states:

Figure 6.20: *D21- seek* behaviour's meandrous shaped course.

- 22.1: hunting/ feeding,
- 22.2: seeking for fishes,

and the following transitions and sensor inputs:

- 22.a: *D*'s course is encircling the school of fish,
- 22.b: *D*'s course is not encircling the school of fish,
- 22.c: school of fish target is locked on sight,
- 22.d: school of fish target is not locked on sight,

### 6.3.2 nonDriver role

The role of the *nonDrivers*, *nD*, team members is a subset of the *Driver*, *D*, role. It is less complicated, since apart from the two basic navigational behaviours, *yaw control* and *visual search*, a *nD* team member has just to be able to locate and follow its leader, *D*, that takes care for the rest of the hunting procedure.

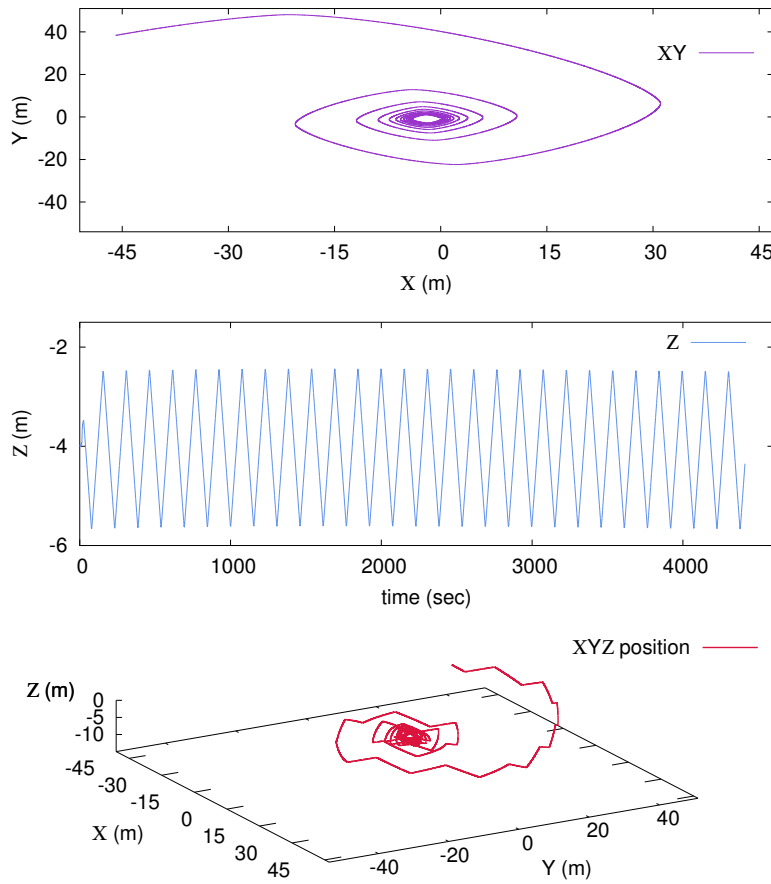


Figure 6.21: *D21- seek* behaviour's spiral pattern course.

### Levels *nD0*, *nD1*, *nD2*

*nD* role diagram, shown in Figure 6.24, shares the same *D* behaviour based architecture and behaviours' finite state machines as described in the previous sections. *nD* role's is a commutative copy of *D* role, apart from the following issues:

**Behaviours *nD02- visual search*, *nD11- wait* and *nd12- check for mates*** : the visual target to detect and track is *D*'s red omnidirectional light. As soon as *D*'s red light is detected and locked on sight, the *nD* member turns on its own light, green coloured for *nD1* and blue for *nD2*. *nD* members do not search for fish, they just follow the leader.

**Behaviour *nD12- check for mates*** : as soon as *nD* detects and locates the leader, it just follows its green light, using *21- follow D*, until the hunting bout finishes.

**Level *nD2*** : The only behaviour that belongs to this level is *nD21- follow D*, as the behaviour *hunt- feed* is not part of level *nD2*.



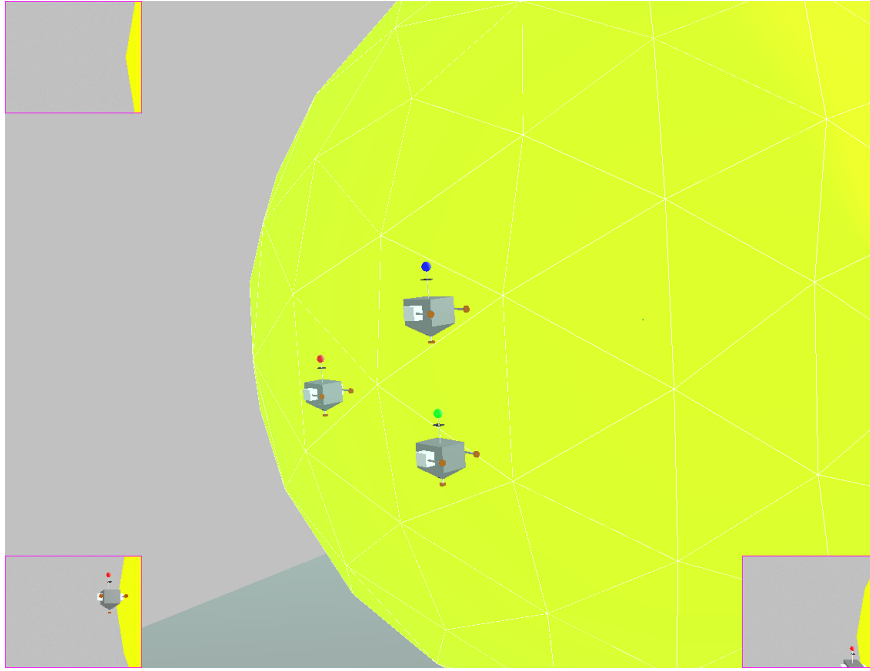


Figure 6.23: *Ale III* AUV team foraging by a school of fish inside Webots simulation environment. All vessels are moving on the edge of the school of fish sphere. The three small frames at picture corners are the views of each vehicle's vision sensor. The top left is *D*'s frame with the school of fish locked on its far left side, the bottom left is *nD1*'s and the bottom right *nD2*'s. Both *nD* members have locked *D*'s red light on their CMUcam3 point of view.

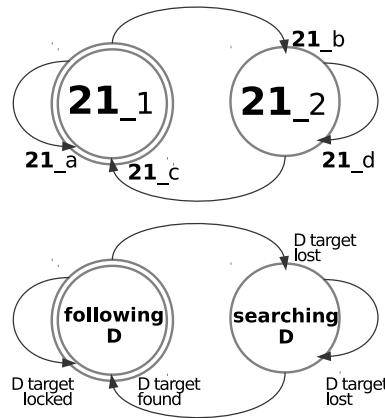
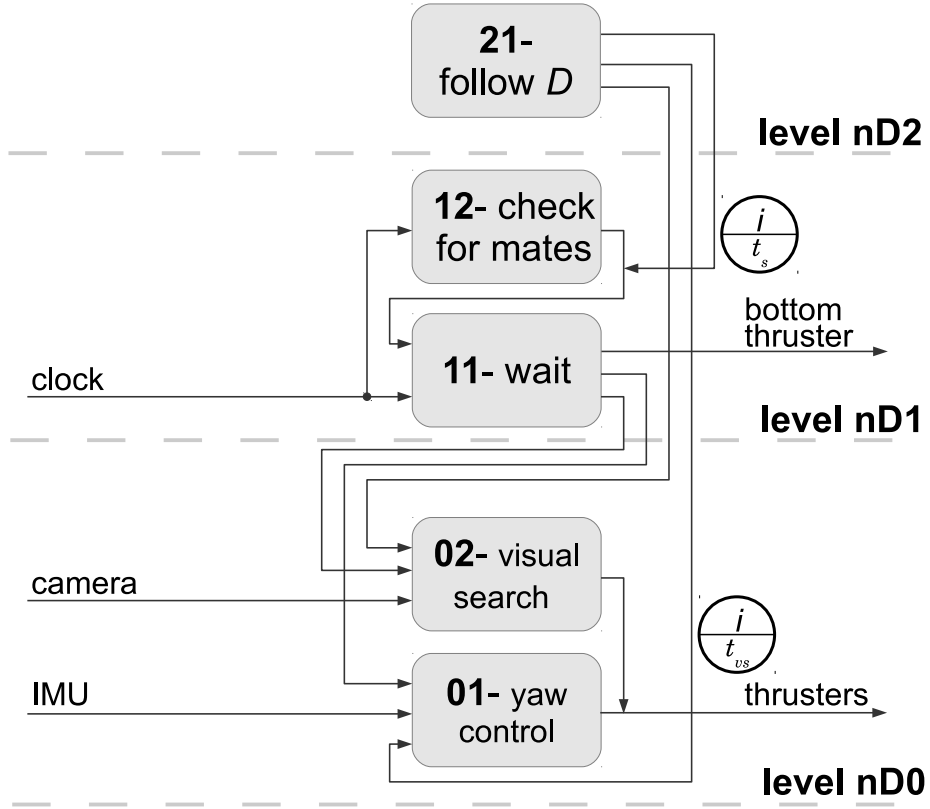


Figure 6.25: *D22-hunt/feed* behaviour's FSM diagram.

**Behaviour *nD21-follow D*** : is a variation of *D21-seek*. It subsumes *nD02-visual search*, trying to lock and follow *D*'s light target. Behaviour's FSM (Figure 6.25) consists of the following states:

- **21\_1**: following *D*,
- **21\_2**: searching *D*,

Figure 6.24:  $nD$  control architecture.

and the following transitions and sensor inputs:

- **21\_a:**  $D$  target locked,
- **21\_b:**  $D$  target lost,
- **21\_c:**  $D$  target found,
- **21\_d:**  $D$  target lost.

## 6.4 Experimental Results

The proposed behaviour based control scheme was tested extensively at the simulation environment. It was proved to be efficient and robust. Both  $D$  and  $nD$  roles accomplish their missions according to the experimental simulation scenarios.

### 6.4.1 Initialization Phase

Figures 6.27, 6.28 and 6.29 describe  $D$ ,  $nD1$  and  $nD2$  behaviours interaction during the initialization phase of a simulated feeding bout. A dot plotted at a diagram marks a behaviour that, during the corresponding controller step, takes part at the determination of AUV's thruster power. At the beginning of the experimentation test depicted by the diagrams,  $D$  has a yaw orientation of  $1.7\text{rad}$  and lies between the two non driver members, at

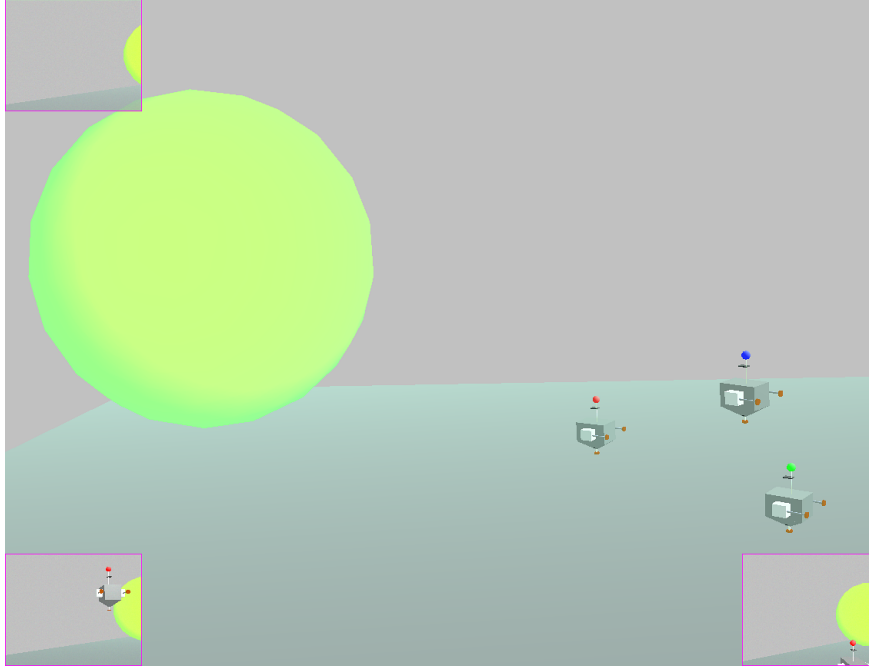


Figure 6.26: *Ale III* AUV team and a school of fish inside Webots simulation environment. Team driver has already detected a school of fish target and leads the two non driving vessels towards it. The driver  $D$  has a red light on, the  $nD1$  has a green light and  $nD2$  has a blue one. The three small frames at picture corners are the views of each vehicle's vision sensor. The top left is  $D$ 's frame, the bottom left is  $nD1$ 's and the bottom right  $nD2$ 's.

$X = 0m, \Psi = 0m, Z = -4m$ , where  $X\Psi$  is the horizontal plane. Both non driver members of the team,  $nD1$  and  $nD2$ , have the same yaw orientation of  $0rad$  and their position is  $(1, -4.5, -2.5)$  and  $(-2.5, -4.2, -2.5)$  respectively. It takes  $23.8sec$  for the driver  $D$  to detect both of its future followers with their lights turned on.  $nD1$  finds  $D$  in  $4sec$ , while  $nD2$  needs more than  $13.3sec$ . This delay is due to the different relative orientation between the team members. All this time  $D$  and  $nD$  members are using:

- *yaw control* to change their heading,
- *visual search* to scan the surrounding area with the camera sensor module,
- *wait* alternatively with *check for mates* to find out the rest of the team members.

### 6.4.2 Feeding Bout Phase

After the bout has been initialized, driver  $D$  tries to discover the school of fish that is simulated by a  $3m$  radius yellow sphere, positioned at  $(-10, -4, -13)$ . Figures 6.30, 6.31 and 6.32 depict the seek- hunt phase that follows feeding bout's initialization. The driver role behaviours that take over AUV's thrusters power control during this phase, as shown in Figure 6.30 are:

- $D02$ - *visual search* to manipulate the camera sensor module. It is subsumed to  $D21$ - *seek* and  $D22$ - *hunt/ feed* effort to discover and lock the school of fish target inside the camera sensor's field of view.
- $D21$ - *seek* to discover and
- $D22$ - *hunt/ feed* to encircle the school of fish target.

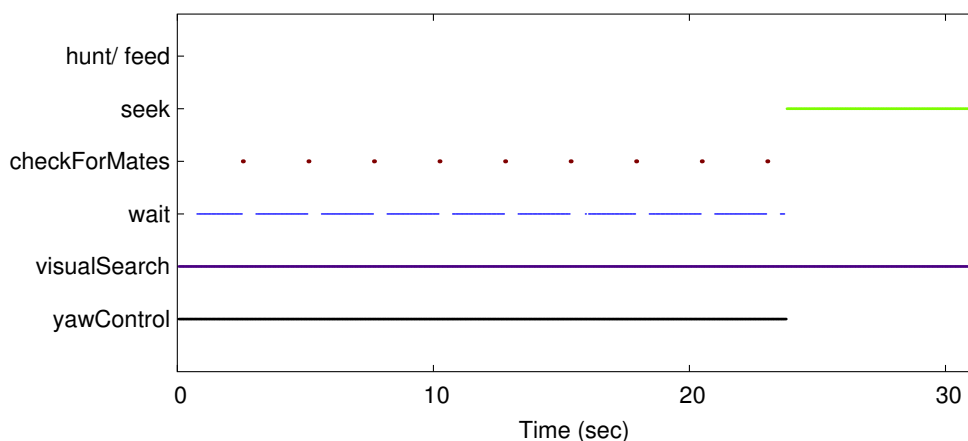


Figure 6.27: *D* role behaviours' interaction, during bout's initialization phase.

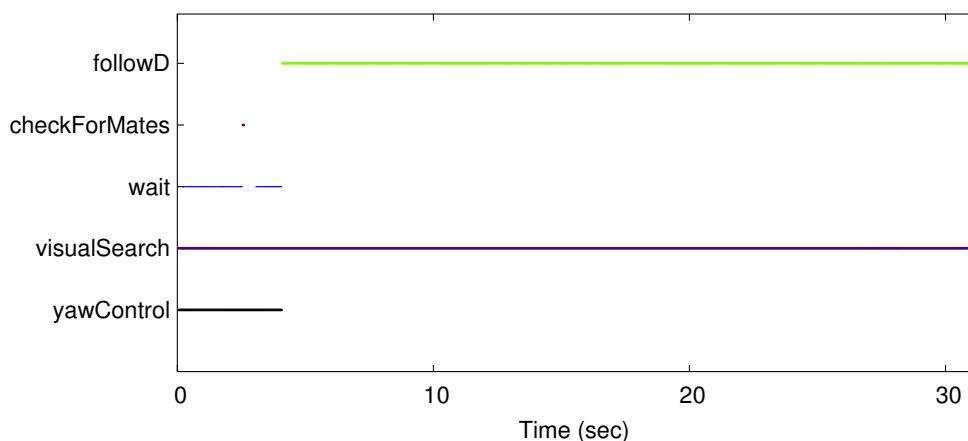


Figure 6.28: *nD1* role behaviours' interaction, during bout's initialization phase.

449sec after experiment's initiation *D* discovers the school of fish target. Thereafter, it approaches and finally encircles the school of fish, leading the team to a successful hunting bout. *D* completes 4 spiral rotations and performs 3 encircling trajectories around the school of fish in 1000sec (Figure 6.30).

The behaviours used after the initialization phase at the non driver role of the rest team members, *nD1* and *nD2*, are *nD02- visual search* and *nD21- follow D*. Using these two behaviours, both *nD* members lock driver's light at their camera sensor field of view and follow *D* to its encircling course surrounding the school of fish. As expected, according to the design of the behaviour based control scheme, after the initialization phase both non driver members present the same behaviour interaction.

## 6.5 Simulation Issues

Webots simulation experimental results showed that the proposed behaviour based cooperative scheme was a successful migration from the real *life* ACoS. Cooperative behaviour's modules were designed on the basis of the three level hierarchy, behaviour based subsumption architecture described in Chapter 5. They implemented

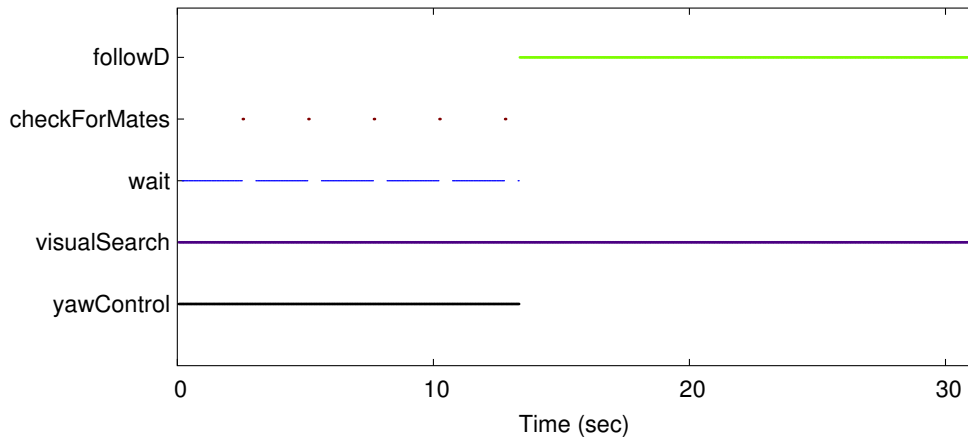


Figure 6.29: *nD2* role behaviours' interaction, during bout's initialization phase.

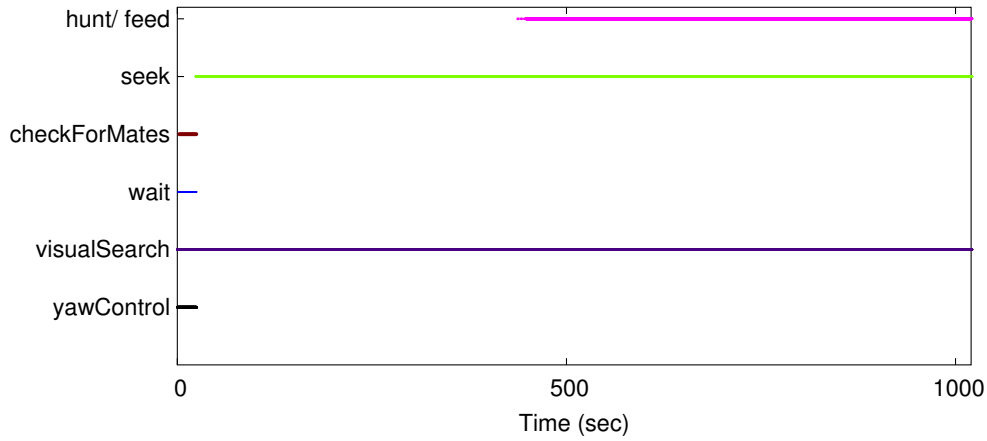


Figure 6.30: *D* role behaviours' interaction, during feeding bout phase.

a two roles team task mimicking the underwater bottle nose dolphins cooperative hunt with division of labour and role specialization, an extremely rare behaviour among living creatures.

Due to experimentation area limitations the proposed biomimetic ACoS was developed and tested successfully in simulation as a proof for its consistency, robustness and functionality. Webots simulation environment proved to be an agile and extremely useful tool for robotic controllers development and debugging. There were not any problems like ACoS crashes or behaviours malfunction throughout the extensive experimentation and testing in simulation. Cooperative behaviour ACoS was tested with one *D* instance and two, three or four *nD* instances interacting and cooperating for a successful *hunting bout*.

The migration from the real *life* ACoS was an unusual procedure, since in robotics the opposite usually happens: controllers are developed in simulation and afterwards, real world experiments test and confirm the simulated results. In the *Ale III* case, it was challenging to expand the behaviour based ACoS design concepts to a more complicated, biomimetic predator cooperative behaviour, even in the *fake*, simulated underwater world.

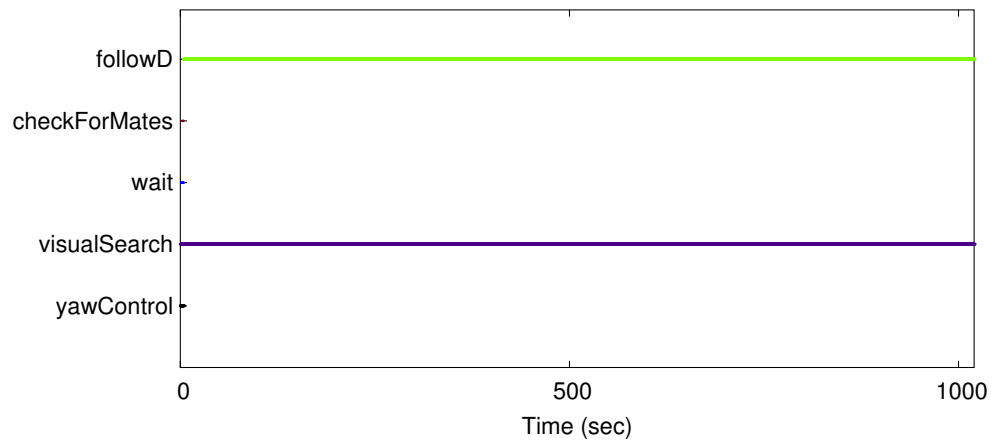


Figure 6.31: *nD1* role behaviours' interaction, during feeding bout phase

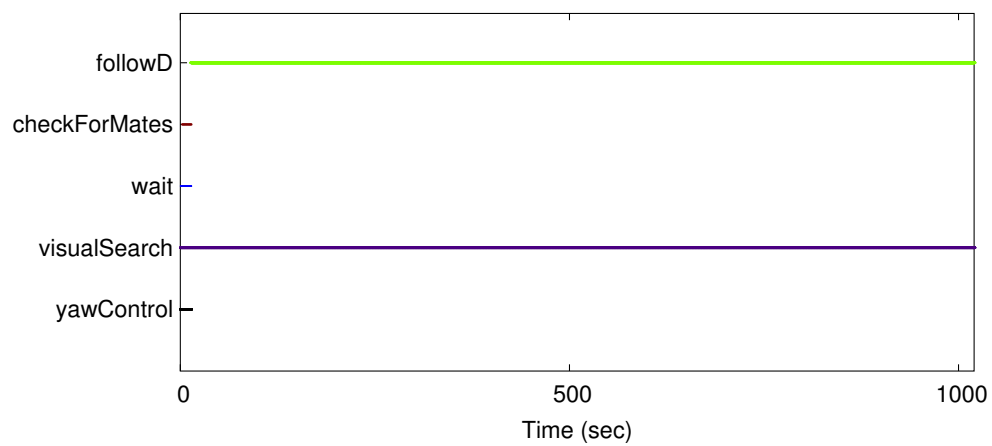


Figure 6.32: *nD2* role behaviours' interaction, during feeding bout phase.

# Chapter 7

## Conclusions

### 7.1 Ale III AUV

*Ale III* was designed to be a low cost, compact vessel, agile enough for experimentation in a laboratory space of 1m<sup>3</sup>. All its hardware and software resources share the open source philosophy. It is equipped with a sensor suite capable of performing basic navigation and visual perception tasks. Its computational performance is more than enough and its three custom made thrusters enable a nimble, *differential* like underwater propelling. Due to its custom design electronics solutions and the carefully chosen low power hardware *Ale III* features a total operational consumption of up to 10W and enjoys a two hours of autonomous underwater operation using a standard economy NiMH battery power supply. The combination of these specifications compose the profile of a unique compact, low logistics vessel for indoors experimentation able to serve as a modular platform for robotic research and development.

After a long period of self testing, *Ale III* was used as a research tool for developing autonomous underwater controllers. Right after the successful implementation of PD yaw and target following controllers, a behaviour based biomimetic controller was designed, programmed and tested proving *Ale III* to be a robust, agile, reliable underwater vessel carrying smart brains, able to implement an underwater *life* mimetic behaviour.

### 7.2 Behaviour Based Biomimetic Control

A biomimetic behaviour, inspired from underwater creatures' real life routine, was chosen as a design target for a novel behaviour based control scheme. This behaviour commutates its states between searching, feeding, hiding and nesting mimicking the everyday life phases of most sea creatures in underwater life. The behaviour based approach designed atomic ACoS architecture that was implemented and debugged using *Ale III* platform. The incremental design steps were programmed as a behaviour based subsumed architecture ACoS, stored and executing on board, with respect to all behaviour based principles. The behaviour based representation simplicity is remarkable. The easiness of programming the behaviour modules, based on their FSM, and implementing the whole control scheme at an object oriented programming environment, significantly reduced project's development time. During all design, development and testing phases this scheme demonstrated one more advantage: modularity. A new behaviour may easily be added upon an already functioning level of competence inside the subsumed architecture. A module may also be removed or altered without any malfunction at the lower level modules.

The experimentation results not only confirmed the behaviour based design correctness but also demonstrated ACoS ability to sustain underwater autonomous operation undertaking typical AUV tasks and satisfying certain constraints met at common underwater missions. This control model may emerge autonomy at underwater operation in a sense of artificial life with sequential phases of *feeding*, *hunting*, *resting* etc. According to

the observations during testing these typical phases of underwater life may effectively compose the solution to AUV control. ACoS performed efficiently without the need of modelling the vessel, as this was not inside the scope of this dissertation.

The proposed atomic ACoS has the ability to handle the most important underwater control issues and at the same time maintain simplicity, modularity, robustness and adaptability. Its basic advantage is the intuitive way of handling autonomy by commutating behaviours observed in the wild underwater life. The outcome of the experimentation results proved ACoS potential to sustain underwater operation in conjunction with the performance of typical tasks. Moreover the controller's representation and design was remarkably efficient: three levels of behaviour based architecture, including no more than six modules successfully supported *Ale III* autonomy.

### 7.3 Behaviour Based Cooperative Control

The reliability and modular functionality of the biomimetic *Ale III* ACoS promoted the concept of practising the same technique for the design of a novel cooperative control scheme. Behaviour based principles made simple the rather complicated task of team cooperation with individually specialized roles devoted to each team member. The control is decentralized and robust. Experimentation results in simulation confirmed the cooperative ACoS functionality. Potentially, an AUV team controlled by the cooperative ACoS may be commissioned in underwater tasks like:

**Oceanography** for monitoring environmental maritime conditions and pollution. The *D* vessel could handle task's initialization to form a team with several *nD* members and follow a course according to the survey's scenario. The team would scan an underwater territory and emerge for recovery. Alternatively *D* could *hunt* a spill and use *nD* members as barriers to encircle and monitor the pollution.

**Security** for surveilling territories of interest, in search for possible intruders and threats.

**Industry** for fault detection in underwater facilities, oil piping and cables deployment.

### 7.4 Further Research and Future Plans

In this dissertation, the development of atomic or cooperating biomimetic roles was based on an interaction between behaviour based modules organized in a multi level hierarchy. This interaction is programmed in a static way inside ACoS framework. ACoS at the current state of development does not contain any sense of *memory* or *learning*. This would be an interesting further research target: artificial neural network and- or genetic algorithms evolution of an intelligent biomimetic behaviour, based on and enriched with the expertise gained during the *life* time of each real or simulated experiment [59], [118]. The individual behaviour based modules would be borrowed from the already developed ACoS versions and their interaction will be the result of an evolution process. Under this point of view, the evolved behaviour will encapsulate the concepts of memory and learning [158], [17]. This combination will be an *intelligent* behaviour based control scheme, based on the static approach, real tests or simulation results.

The next experimentation step with *Ale III* will be at seashore. New scenarios will test the vessel's potentials in this extremely noisy and unstructured environment. The lighting conditions undersea will be a challenge for *Ale III* vision system. Sea waving and currents will test vessel's seaworthiness in the hard way, far from the tranquil laboratory experimentation area. New control issues will arise and the behaviour based model will need anamorphosis to compete the new more hostile conditions.

Apart from the *wild*, the domestic laboratory experimentation area needs expansion. Indoors experimentation presents overwhelming advantages in relevance with open sea in terms of availability, feasibility, money and time cost. To exploit the full range of them the acquisition of a tank with volume at least  $3\text{m}^3$  will be a great progress for the experimentation area. As realized after several months of tests in the existing laboratory



facility, for every Ale III class AUV added to the experimentation scenario at least  $1\text{m}^3$  of additional underwater space volume is necessary. As soon as a bigger experimentation area is available, more than one vessels may be operated simultaneously inside the tank. This would be a great opportunity to test the bottlenose dolphins predator cooperative behaviour ACoS with the scenario already performed in simulation, or some other cooperative task.



# Thruster Blueprints

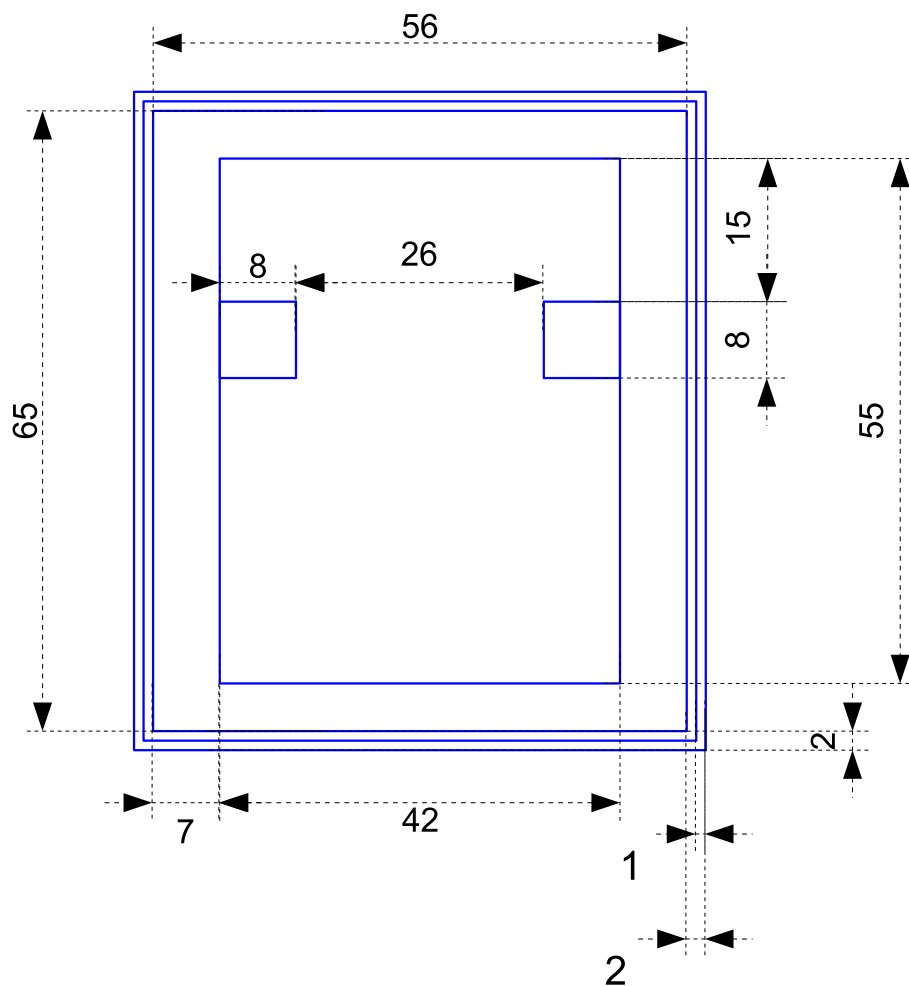


Figure A.1: *A/e III* thruster top view. All dimensions are in mm.

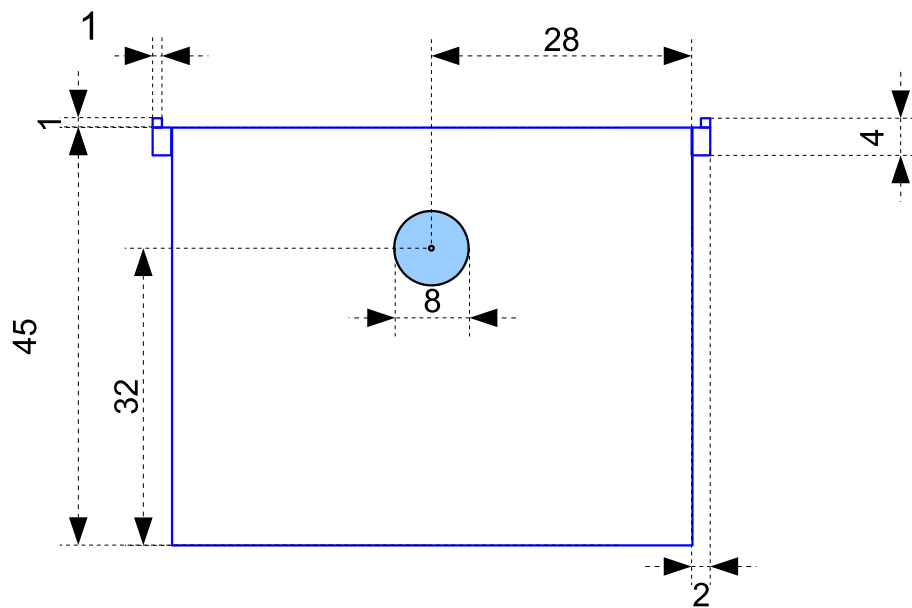


Figure A.2: *Ale III* thruster back view with the opening for the propeller stern tube. All dimensions are in mm.

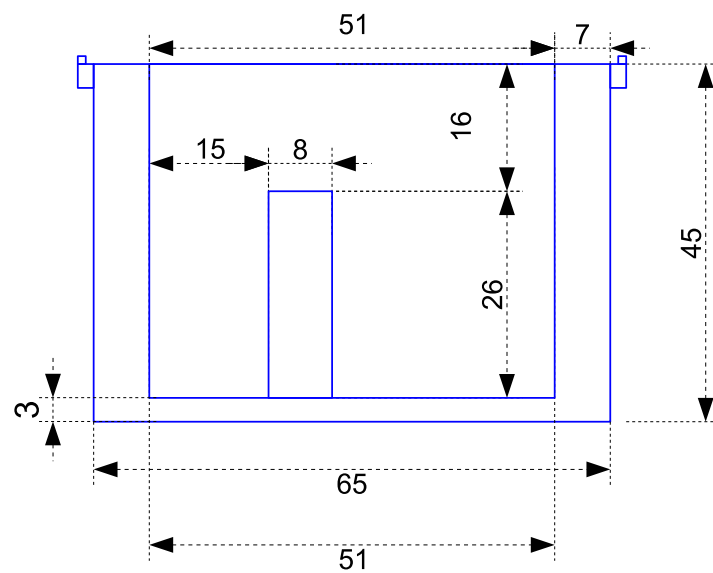


Figure A.3: *Ale III* thruster side view. All dimensions are in mm.

# Appendix B

## Electronics Schematics

### B.1 Power Supply Circuit

The power supply circuit, Figure B.1, comprises from the following components:

**12V** NiMH 2600mAh 10AA rechargeable battery pack. Supplies electricity to the electronic devices, separately from the noisy motors. If the *external blind connector* mates with the *power connector*, electricity flows from the 12 battery to the 12V *electronics power supply* bus. If the *external power & recharging connector* mates with the *power connector*, then:

- if the 12V *charging switch* is turned on, the 12V battery starts recharging via the 12V *NiMH charger*.
- if the 12V *external power switch* is turned on, AUV's 12V *electronics power supply* bus is connected to the 12V *external power supply* sourcing uninterrupted power supply for unlimited time period.

**7.2V** NiMH 2600mAh 6AA rechargeable battery pack. Supplies electricity to the noisy motors, separately from the delicate electronic devices. If the *external blind connector* mates with the *power connector*, electricity flows from the 7.2 battery to the 7.2V *electronics power supply* bus. If the *external power & recharging connector* mates with the *power connector*, then:

- if the 7.2V *charging switch* is turned on, the 7.2V battery starts recharging via the 7.2V *NiMH charger*.
- if the 7.2V *external power switch* is turned on, AUV's 7.2V *motors power supply* bus is connected to the 7.2V *external power supply* sourcing uninterrupted power supply for unlimited time period to the motor drivers.

**power connector** Bulgin 400 Buccaneer 8 poles panel mount connector, rated at IP68 BS EN 60529:1992 [170] ingress protection standard. The *power connector* is mounted at vessels stern and functions as a power switch or external power supply and recharging connector, depending on its mating cable connector.

**external blind connector** shares the same specification with the former. When it mates with vessel's stern *power connector* then it turns on AUV's electric power, sourcing electricity from its batteries to its power supply buses.

**external power & recharging connector** shares the same specification with the former. When it mates with vessel's stern *power connector* then:

- 7.2V *motors power supply* and 12V *electronics power supply* buses are connected to the 7.2V *external power supply* and 12V *external power supply* respectively.

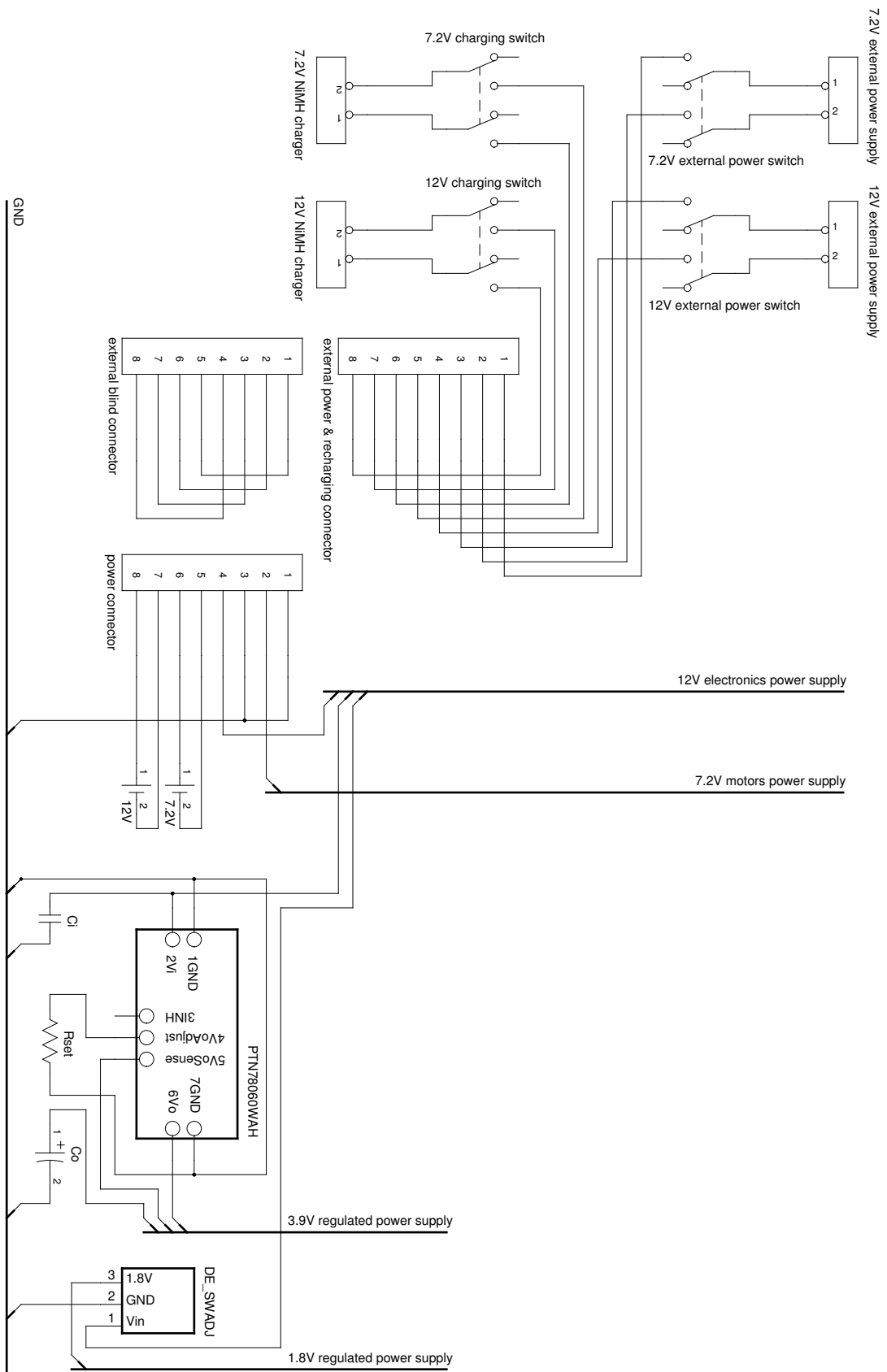


Figure B.1: Power Supply Circuit.

- 7.2V and 12V batteries are connected to the 7.2V NiMH charger and 12V NiMH charger respectively.

**12V, 7.2V NiMH chargers** used for vessels batteries' recharging process.

**12V, 7.2V external power supply** used for uninterrupted- unlimited power supply. Useful for long period experimentation.

**12V and 7.2V external power switch** , turn on- off the 12V, 7.2V external power supply, respectively.

**12V and 7.2V charging switch**, turn on- off the 12V, 7.2V battery recharging process, respectively.

**12V electronics power supply**, bus for supplying power to the electronics.

**7.2V electronics power supply**, bus for supplying power to the motor drivers.

**PTN78060WAH** 3A step down, adjustable, switching regulator from Texas Instruments [168]. Supplies power to vessel's COM via the 3.9V regulated power supply. It is connected to the following components:

- $C_i$ , 3 $\mu$ F ceramic input capacitor.
- $C_o$ , 180 $\mu$ F electrolytic output capacitor.
- $R_{set}$ , 43 $\Omega$  set- point resistor regulating the  $V_o$  output voltage to 3.9V.

**DE-SWADJ** 10W step down, adjustable, switching regulator from Dimension Engineering [53]. Supplies 1.8V regulated power to the 1.8V regulated power supply bus for the needs of the inverter- buffer controlling the serial communication between IMU and COM, see Section B.2.

**3.9V regulated power supply** bus for supplying COM.

**1.8V regulated power supply** bus for supplying the inverter- buffer controlling the serial communication between IMU and COM, see Section B.2.

## B.2 COM and Sensors Circuit

The COM and sensors interconnection circuit, Figure B.2, comprises from the following components:

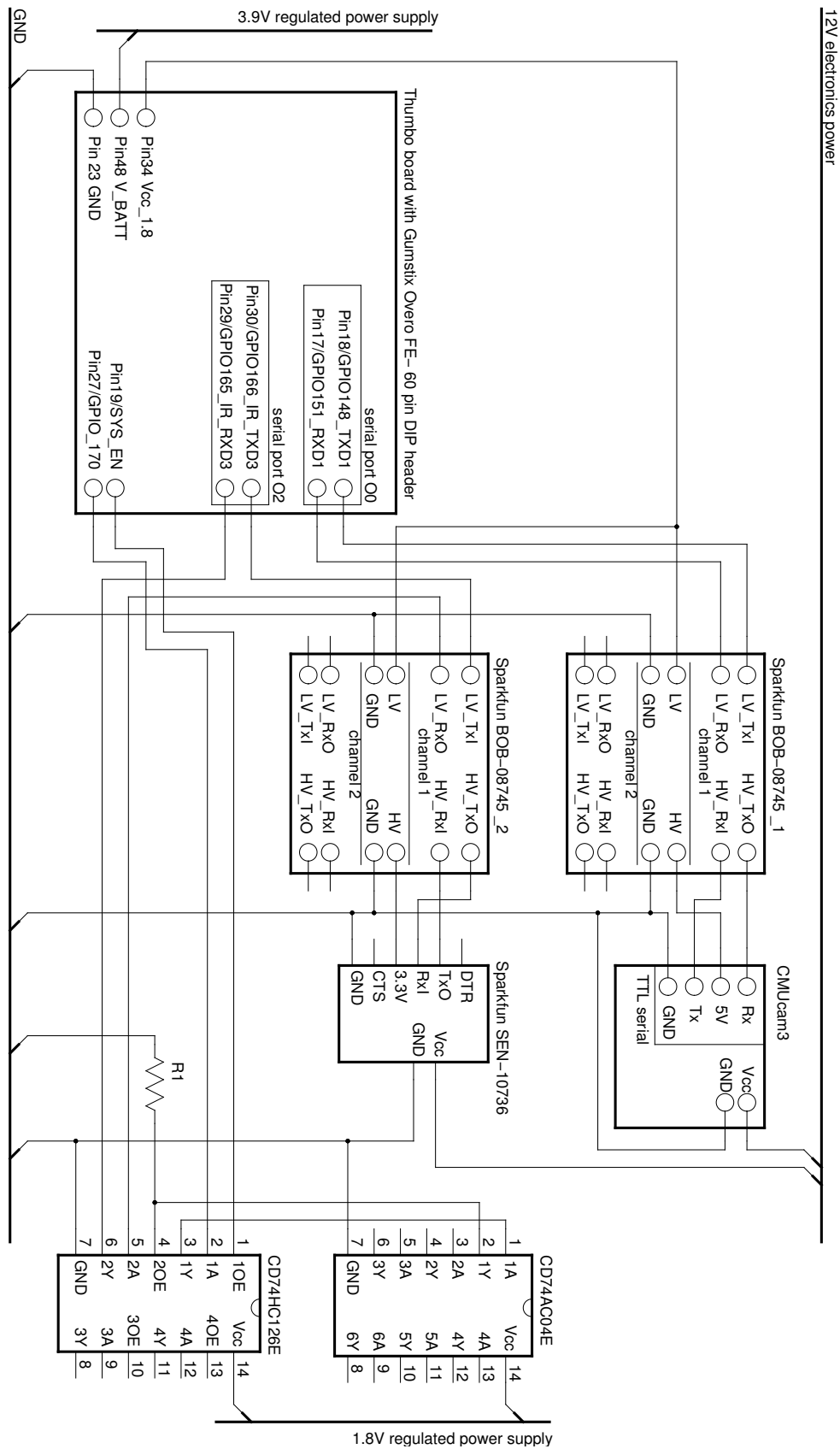
**Thumbo motherboard 60 pin header** hosting Gumstix IronSTORM I/O pins [78]. The IronSTORM COM uses two serial ports, *O0* and *O2* to communicate with the *CMUcam3* and *Sparkfun SEN-10736* IMU respectively, using the appropriate logic level conversions, since COM's logic level is at 1.8V, while *CMUcam3* uses a 5V and IMU uses a 3.3V logic. All pin numbers refer to the Thumbo 60- pin DIP header numbering. Noise on the *Pin29/GPIO165\_IR\_RXD3* line during boot can interrupt the normal booting sequence. For that reason the *CD74HC126E* buffer isolates this receive line until the system is ready for I/O communications. The *CD74HC126E* buffer protects

**Sparkfun BOB-08745\_1** logic level converter [4] to convert *serial port O0* 1.8V to 5V logic level used at *CMUcam3* vision sensor.

**Sparkfun BOB-08745\_2** logic level converter [4] to convert *serial port O2* 1.8V to 3.3V logic level used at *Sparkfun SEN-10736* IMU sensor.

**CMUcam3** an open source, programmable, embedded colour vision sensor [43]. Communicates with COM's *serial port O0* via *Sparkfun BOB-08745\_1* logic level converter.

**Sparkfun SEN-10736** an open source, programmable, embedded IMU [4]. Communicates with COM's *serial port O2* via *Sparkfun BOB-08745\_1* logic level converter and *CD74HC126* buffer. Noise or communication signals transmitted at *serial port O2 Pin29/GPIO165\_IR\_RXD3* pin during AUV's electronics power- on and COM's boot process, can interrupt the normal booting sequence and lead to a system





crash. This is the reason for isolating, using the *CD74HC126E* buffer, the *serial port O2* receive signal from the rest of the electronics, until COM has finished its boot procedure. As soon as COM has started up, the controller program sets *Pin27/GPIO\_170* output to enable the buffered communication for the *serial port O2 Pin29/GPIO165\_IR.RXD3* pin.

**CD74AC04E** inverter from Texas Instruments [168], used to invert *Pin27/GPIO\_170*.

**CD74HC126E** high speed, three state buffer from Texas Instruments [168], used to protect *Pin29/GPIO165\_IR.RXD3* from noise and communication signals during COM's booting process. The complete sequence from system's start up, until the beginning of receiving communication signals at *serial port O2 Pin29/GPIO165\_IR.RXD3* is:

1. COM starts booting process. As soon as it is ready for I/O communication, sets the system enable *Pin19/SYS\_EN* pin that drives first buffer's enable *IOE* input. Before that *1Y* and *2Y* are in high impedance state.
2. *Pin37/GPIO\_170* data pass from buffer's *1A* input to *1Y* output. *Pin37/GPIO\_170* value, HIGH by default, is inverted at *CD74AC04E* output *1Y*, holding buffer's *2Y* output in high impedance state.
3. The user space controller program sets the *Pin27/GPIO\_170* output value to LOW. This value is inverted and sets buffer's *2OE* value to HIGH, enabling buffer's *2A* input from *Sparkfun SEN-10736* to pass from *2Y* to *Pin29/GPIO165\_IR.RXD3*.
4. Bi- directional communication through *serial port O2* has been established.

Second buffer's enable *2OE* input is LOW

**R<sub>1</sub>** , 1K $\Omega$  pull down resistor disabling buffer's output *2Y*.

## B.3 COM and Motor Drivers Circuit

The COM and motor drivers interconnection circuit, Figure B.3, comprises from the following components:

**Thumbo motherboard 60 pin header** hosting Gumstix IronSTORM PWM output pins. Three PWM lines, one for each motor, have been activated and programmed according to motor drivers' input signal specifications. Once again a conversion is necessary to connect COM's 1.8V logic level to the drivers' 5V logic level.

**Sparkfun BOB-08745.3** logic level converter [4]. Converts *Pin36/GPIO145\_PWM10* and *Pin35/GPIO146\_PWM11* 1.8V to 5V logic level used at *Sabertooth 2 $\times$ 5* driver.

**Sparkfun BOB-08745.4** converts *Pin37/GPIO145\_PWM10* and *Pin37/GPIO147\_PWM8* 1.8V to 5V logic level used at *Sabertooth 2 $\times$ 5* driver.

**Sabertooth 2 $\times$ 5 LR** dual 5A motor driver [53] from Dimension Engineering controlling vessel's two lateral thrusters.  $S_1$  is the right thruster's motor command line and  $S_2$  is for the left one, driven by *Pin36/GPIO145\_PWM10* and *Pin35/GPIO146\_PWM11* respectively. *7.2V motors power supply* powers all motor drivers.

**Sabertooth 2 $\times$ 5 B** dual driver controlling vessel's bottom thrusters.  $S_1$  is the command line driven by *Pin37/GPIO147\_PWM8*.

**M.L, R and B** 6V direct current *Graupner Multispeed 140* [77] brushed electric motors, controlled by *Sabertooth 2 $\times$ 5* motor drivers.

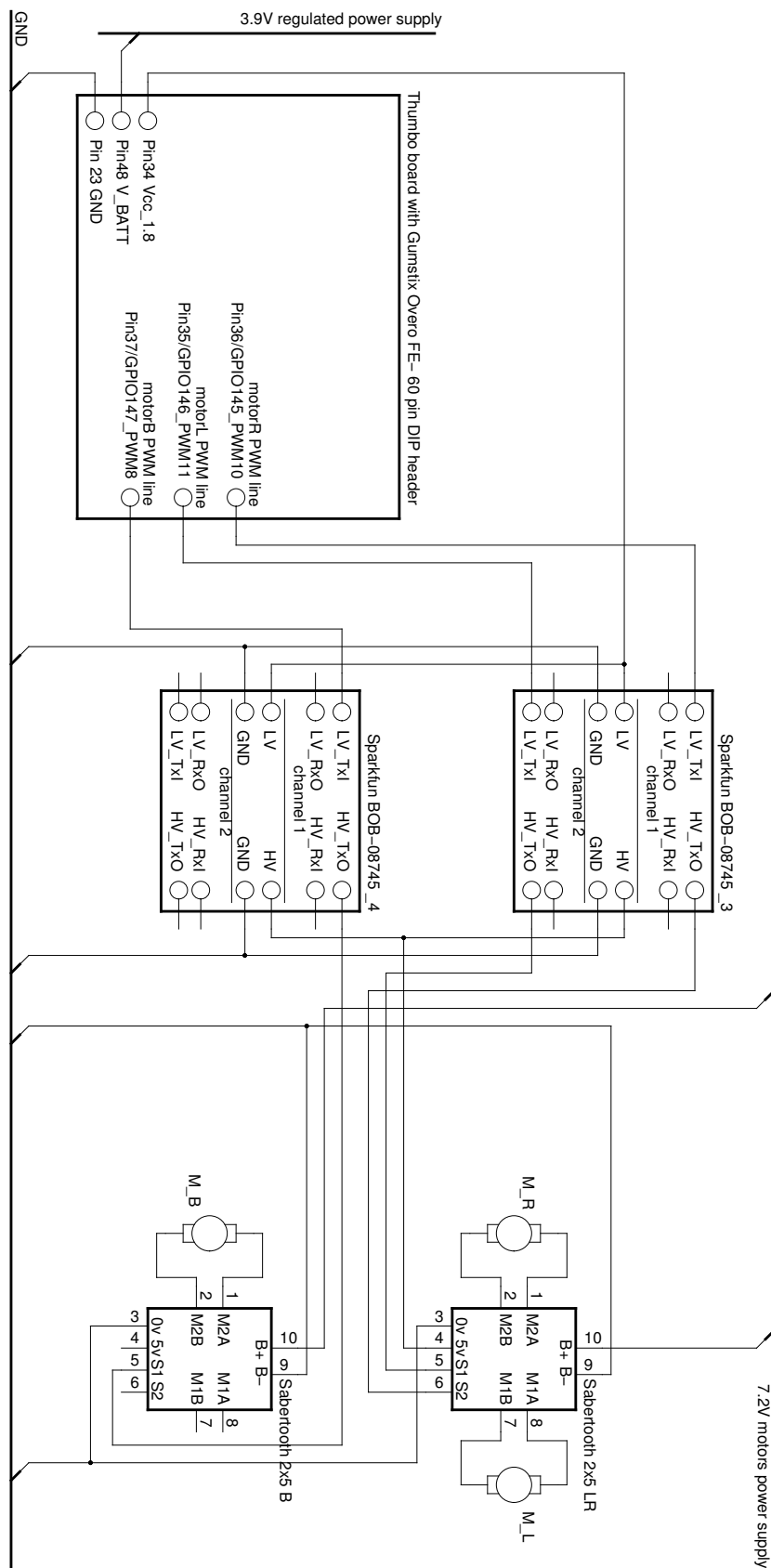


Figure B.3: COM and Electric Motors Circuit.

# Bibliography

- [1] Bluefin Robotics. <http://www.bluefinrobotics.com> [Online; accessed December 2013].
- [2] Gavia Autonomous Underwater Vehicles. <http://www.gavia.is> [Online; accessed December 2013].
- [3] Kongsberg Maritime. <http://www.km.kongsberg.com/hydroid> [Online; accessed December 2013].
- [4] Sparkfun Electronics. <https://www.sparkfun.com> [Online; accessed December 2013].
- [5] Tecnadyne. <http://www.tecnadyne.com> [Online; accessed December 2013].
- [6] ACSA-ALCEN. The SeaExplorer underwater glider breaks World Record. Technical report, Aix-en-Provence, France, December 2013.
- [7] A. Alvarez, A. Caffaz, A. Caiti, G. Casalino, L. Gualdesi, A. Turetta, and R. Viviani. Fòlaga: A low-cost autonomous underwater vehicle combining glider and AUV capabilities. *Ocean Engineering*, 36(1):24–38, January 2009.
- [8] A. Alvarez, J. Chiggiato, and K. Schroeder. Mapping sub-surface geostrophic currents from altimetry and a fleet of gliders. *Deep Sea Research Part I: Oceanographic Research Papers*, 74:115 – 129, 2013.
- [9] A. Amory, B. Meyer, C. Osterloh, T. Tosik, and E. Maehle. Towards Fault-Tolerant and Energy-Efficient Swarms of Underwater Robots. In *Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2013 IEEE 27th International*, pages 1550–1553, 2013.
- [10] B. Anderson and J. Crowell. Workhorse AUV A cost-sensible new Autonomous Underwater Vehicle for Surveys/Soundings, Search & Rescue, and Research. In *Proceedings of OCEANS 2005 MTS/IEEE*, pages 1–6. IEEE, 2005.
- [11] Carl Anderson and Nigel R. Franks. Teamwork in Animals, Robots, and Humans. volume 33 of *Advances in the Study of Behavior*, pages 1 – 48. Academic Press, 2003.
- [12] Tracy L. Anderson and Max Donath. Animal behavior as a paradigm for developing robot autonomy. *Robotics and Autonomous Systems*, 6(1-2):145–168, June 1990.
- [13] William Anderson. *Nautilus 90 North*. Tab Books, 1989.
- [14] Gianluca Antonelli, Thor I Fossen, and Dana R. Yoerger. Underwater robotics. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, volume 15, chapter 43, pages 987–1008. Springer Berlin Heidelberg, Berlin, Heidelberg, January 2008.
- [15] Ronald C. Arkin. *Behavior-Based Robotics*. MIT Press, 1998.
- [16] T. Assaf, C. Stefanini, and P. Dario. Autonomous Underwater Biorobots: A Wireless System for Power Transfer. *Robotics Automation Magazine, IEEE*, 20(3):26–32, 2013.
- [17] Christopher G. Atkeson and Stefan Schaal. Memory-based neural networks for robot learning. *Neurocomputing*, 9(3):243–269, December 1995.

- [18] Josep Aulinas, Yvan R Petillot, Xavier Lladó, Joaquim Salvi, and Rafael Garcia. Vision-based underwater SLAM for the SPARUS AUV. In *International Conference on Computer and IT Applications in the Maritime Industries (COMPIT)*, pages 171–181, 2011.
- [19] Autonomous Systems and Control Laboratory. Virginia Tech, Department of Electrical and Computer Engineering. <http://www.asc1.ece.vt.edu> [Online; accessed December 2013].
- [20] Autosub6000 AUV. National Oceanography Centre. <http://noc.ac.uk> [Online; accessed December 2013].
- [21] AUVAC. Autonomous Undersea Development Center. <http://auvac.org> [Online; accessed December 2013].
- [22] Juan Pablo Julca Avila, Julio Cezar Adamowski, Newton Maruyama, Fabio Kawaoka Takase, and Milton Saito. Modeling and Identification of an Open-frame Underwater Vehicle: The Yaw Motion Dynamics. *Journal of Intelligent & Robotic Systems*, 66(1-2):37–56, August 2011.
- [23] Christopher Barngrover, Ryan Kastner, Thomas Denewiler, and Greg Mills. The stingray AUV: A small and cost-effective solution for ecological monitoring. In *OCEANS 2011*, pages 1–8, Waikoloa, HI, 2011. IEEE.
- [24] Edward Beach. *Around the World Submerged: The Voyage of the Triton*. Naval Institute Press, 2001.
- [25] J. C. Bednarz. Cooperative Hunting in Harris' Hawks (*Parabuteo unicinctus*). *Science*, 239:1525–1527, 1988.
- [26] Michael R Benjamin, Henrik Schmidt, Paul M Newman, and John J Leonard. Nested autonomy for unmanned marine vehicles with MOOS-IvP. *Journal of Field Robotics*, 27(6):834–875, November 2010.
- [27] Andrew A Bennett and John J Leonard. A behavior-based approach to adaptive feature detection and following with autonomous underwater vehicles. *Oceanic Engineering, IEEE Journal of*, 25(2):213–226, 2000.
- [28] Rainer Bischoff and Volker Graefe. Learning from Nature to Build Intelligent Autonomous Robots. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3160–3165. IEEE, October 2006.
- [29] D. R. Blidberg. The Development of Autonomous Underwater Vehicles (AUVs); A Brief Summary. In *IEEE ICRA*, volume 6500, Seoul, Korea, 2001.
- [30] Christophe Boesch and Hedwige Boesch. Hunting Behavior of Wild Chimpanzees in the Tai National Park. *American Journal of Physical Anthropology*, 78:547–573, 1989.
- [31] Joe Borden and Jeffery DeArruda. Long range acoustic underwater communication with a compact AUV. In *2012 Oceans*, pages 1–5. IEEE, October 2012.
- [32] Douglas Botting. *The U-boats*. Time Life UK, 1981.
- [33] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23, 1986.
- [34] Javier Busquets, Dionisio Tudela, Francisco Perez, Jesus Busquets-Carbonell, Alvaro Barbera, Carlos Rodriguez, Antonio Javier Garcia, and Javier Gilabert. Low-cost AUV based on Arduino open source microcontroller board for oceanographic research applications in a collaborative long term deployment missions and suitable for combining with an USV as autonomous automatic recharging platform. In *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*, number 1, pages 1–10. IEEE, September 2012.
- [35] A. Caiti, V. Calabro, F. Di Corato, D. Meucci, and A. Munafo. Cooperative distributed algorithm for AUV teams: A minimum entropy approach. In *OCEANS - Bergen, 2013 MTS/IEEE*, pages 1–6, 2013.

- [36] A. Caiti, V. Calabro, T. Fabbri, D. Fenucci, and A. Munafo. Underwater communication and distributed localization of AUV teams. In *OCEANS - Bergen, 2013 MTS/IEEE*, pages 1–8, 2013.
- [37] A. Caiti, T. Fabbri, D. Fenucci, and A. Munafo. Potential games and AUVs cooperation: First results from the THESAURUS project. In *OCEANS - Bergen, 2013 MTS/IEEE*, pages 1–6, 2013.
- [38] Andrea Caiti, Vincenzo Calabrò, Andrea Munafo, Gianluca Dini, and Angelica Lo Duca. Mobile Underwater Sensor Networks for Protection and Security: Field Experience at the UAN11 Experiment. *Journal of Field Robotics*, 30(2):237–253, March 2013.
- [39] N. Carlesi, F. Michel, B. Jouvencel, and J. Ferber. Generic Architecture for Multi-AUV Cooperation Based on a Multi-Agent Reactive Organizational Approach. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 5041–5047, 2011.
- [40] M Carreras, J Batlle, P Ridao, and G N Roberts. An overview on behaviour-based methods for AUV control. *Behaviour*, pages 141–146, 2000.
- [41] M. Carreras, Junku Yuh, J. Batlle, and P. Ridao. A behavior-based scheme using reinforcement learning for autonomous underwater vehicles. *Oceanic Engineering, IEEE Journal of*, 30(2):416–427, 2005.
- [42] P.J. Clapham. *Cetacean Societies: Field Studies of Dolphins and Whales*, volume 239, chapter The Humpback Whale: Seasonal Feeding and Breeding in a Baleen Whale, pages 173–198. University of Chicago Press, Chicago, 2000.
- [43] CMUcam3. CMUcam: Open Source Programmable Embedded Color Vision Sensors. <http://www.cmucam.org> [Online; accessed December 2013].
- [44] Committee on Undersea Vehicles and National Needs, National Research Council. *Undersea Vehicles and National Needs*. The National Academies Press, 1996.
- [45] Scott Creel and Nancy Marusha Creel. Communal hunting and pack size in African wild dogs, *Lycaon pictus*. *Animal Behaviour*, 50(5):1325 – 1339, 1995.
- [46] Douglas Creighton and Saeid Nahavandi. Kinematic modeling of a bio-inspired robotic fish. In *2008 IEEE International Conference on Robotics and Automation*, pages 695–699, Pasadena, May 2008. IEEE.
- [47] Alessandro Crespi, Konstantinos Karakasiliotis, Andre Guignard, and Auke Jan Ijspeert. Salamandra Robotica II: An Amphibious Robot to Study Salamander-Like Swimming and Walking Gaits. *IEEE Transactions on Robotics*, 29(2):308–320, 2013.
- [48] R. Cui, S. S. Ge, B. V. E. How, and Y. S. Choo. Leader-follower Formation Control of Underactuated AUVs with Leader Position Measurement. In *2009 IEEE International Conference on Robotics and Automation*, page 979984. IEEE, 2009.
- [49] F.R. Dagleish, S. Tetlow, and R.L. Allwood. Experiments in laser-assisted visual sensing for AUV navigation. *Control Engineering Practice*, 12(12):1561–1573, December 2004.
- [50] John Darrington. Artificial Neural Network Library. <http://www.nongnu.org/libann> [Online; accessed December 2013].
- [51] Robert H. Davis. *Deep Diving and Submarine Operations Part 1 and 2*. Siebe Gorman and Co Ltd, 9th edition, 1995.
- [52] Deltron Enclosures. DEM Manufacturing. <http://www.dem-uk.com/deltron-enclosures> [Online; accessed December 2013].
- [53] Dimension Engineering LLC. <http://www.dimensionengineering.com> [Online; accessed December 2013].
- [54] L. Doitsidis, N.C. Tsourveloudis, and S. Piperidis. Evolution of fuzzy controllers for robotic vehicles: The role of fitness function selection. *Journal of Intelligent and Robotic Systems*, 56(4):469–484, 2009.

- [55] Dorado AUV. Monterey Bay Aquarium Research Institute. <http://www.mbari.org/auv> [Online; accessed December 2013].
- [56] DSME E&R. Daewoo Shipbuilding & Marine Engineering Co. Ltd. <http://dsmeu.en.ec21.com> [Online; accessed December 2013].
- [57] ECA Robotics. <http://www.eca-robotics.com> [Online; accessed December 2013].
- [58] Scott Ellis. PWM library for OMAP3 based Linux systems, 2012. <https://github.com/scottellis/omap3-pwm> [Online; accessed December 2013].
- [59] Andries P. Engelbrecht. *Computational intelligence: an introduction*. John Wiley & Sons Ltd, 2007.
- [60] Eprons ROV Ltd. RovBuilder. <http://www.deepocean.com> [Online; accessed December 2013].
- [61] Nathaniel Fairfield, George Kantor, and David Wettergreen. Real-time SLAM with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, 24:3–21, 2007.
- [62] J.J. Fernandez, M. Prats, P.J. Sanz, J.C. Garcia, R. Marin, M. Robinson, D. Ribas, and P. Ridao. Grasping for the Seabed: Developing a New underwater Robot Arm for Shallow-Water Intervention. *Robotics Automation Magazine, IEEE*, 20(4):121–130, 2013.
- [63] E. Fiorelli, N. E. Leonard, P. Bhatta, D. a. Paley, R. Bachmayer, and D. M. Fratantoni. Multi AUV Control and Adaptive Sampling in Monterey Bay. *Journal of Oceanic Engineering*, 31:935–948, 2006.
- [64] E. Fiorelli, N.E. Leonard, P. Bhatta, D.A. Paley, R. Bachmayer, and D.M. Fratantoni. Multi-AUV Control and Adaptive Sampling in Monterey Bay. *Oceanic Engineering, IEEE Journal of*, 31(4):935–948, 2006.
- [65] Bernard Fitzsimons. *The Illustrated Encyclopedia of 20th Century Weapons and Warfare*, volume 12. Phoebus, 1977.
- [66] D Floreano and F Mondada. Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot. In *3rd International Conference on Simulation of Adaptive Behavior : From Animals to Animats 3*, pages 421–430. MIT Press, 1994.
- [67] Tutorial for Dynamic Programming. [http://www.codechef.com/wiki/tutorial-dynamic-programming#Problem:\\_Longest\\_Increasing\\_subsequence](http://www.codechef.com/wiki/tutorial-dynamic-programming#Problem:_Longest_Increasing_subsequence) [Online; accessed December 2013].
- [68] Forum Energy Technologies. <http://www.f-e-t.com> [Online; accessed December 2013].
- [69] Thor I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley & Sons, 1994.
- [70] N.R. Franks, A. Sendova-Franks, and C. Anderson. Division of labour within teams of New World and Old World army ants. *Animal Behaviour*, 62:635 – 642, 2001.
- [71] Matthias O. Franz and Hanspeter A. Mallot. Biomimetic robot navigation. *Robotics and Autonomous Systems*, 30(1-2):133–153, January 2000.
- [72] C++ Fuzzy Logic Programming Library. <http://sourceforge.net/projects/cpp-fuzzy-logic> [Online; accessed December 2013].
- [73] GAlib. A C++ Library of Genetic Algorithm Components. <http://lancet.mit.edu/ga/GAlib.html> [Online; accessed December 2013].
- [74] GAUL. Genetic Algorithm Utility Library. <http://gaul.sourceforge.net/> [Online; accessed December 2013].
- [75] Stefanie K Gazda, Richard C Connor, Robert K Edgar, and Frank Cox. A division of labour with role specialization in group-hunting bottlenose dolphins (*Tursiops truncatus*) off Cedar Key, Florida. *Proceedings. Biological sciences / The Royal Society*, 272(1559):135–40, January 2005.

- [76] GNOM. Underwater Remotely Operated Vehicle. <http://www.gnom-rov.com> [Online; accessed December 2013].
- [77] Graupner/SJ GmbH. <http://www.graupner.de> [Online; accessed December 2013].
- [78] Gumstix Overo. Gumstix, Inc. <https://www.gumstix.com> [Online; accessed December 2013].
- [79] Jenhwa Guo. Maneuvering and control of a biomimetic autonomous underwater vehicle. *Autonomous Robots*, 26(4):241–249, April 2009.
- [80] Shuxiang Guo, Liwei Shi, Nan Xiao, and Kinji Asaka. A biomimetic underwater microrobot with multi-functional locomotion. *Robotics and Autonomous Systems*, 60(12):1472–1483, December 2012.
- [81] Brayton Harris. *Navy Times Book of Submarines*. Berkley Trade, 2001.
- [82] Thomas Hiller, Arnar Steingrímsson, and Robert Melvin. Expanding the Small AUV Mission Envelope; Longer, Deeper & More Accurate. In *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 1–4. IEEE, September 2012.
- [83] G. a. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme. Active Planning for Underwater Inspection and the Benefit of Adaptivity. *The International Journal of Robotics Research*, 32(1):3–18, November 2012.
- [84] Wenling Huang, Huajing Fang, and Li Liu. Obstacle Avoiding Policy of Multi-AUV Formation Based on Virtual AUV. *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, pages 131–135, 2009.
- [85] Ya-wen Huang, Yuki Sasaki, Yukihiro Harakawa, Edwardo F Fukushima, and Shigeo Hirose. Operation of Underwater Rescue Robot Anchor Diver. In *OCEANS 2011*, pages 1–6, Waikoloa, HI, 2011. IEEE.
- [86] T. Husøy, M. Pettersen, B. Nilsson, T. Oberg, N. Warakagoda, and A. Lie. Implementation of an underwater acoustic modem with network capability. In *OCEANS, 2011 IEEE - Spain*, pages 1–10, 2011.
- [87] T. Hyakudome, H. Yoshida, T. Nakatani, Y. Ohta, T. Tani, K. Sugihara, T. Moriga, T. Iwamoto, Y. Kawaharazaki, T. Oda, and Y. Fujita. Development of fuel cell system for underwater power source. In *OCEANS - Bergen, 2013 MTS/IEEE*, pages 1–6, 2013.
- [88] Hydrovision Limited. <http://195.11.212.163> [Online; accessed December 2013].
- [89] Dave Hylands. GPIO library for Gumstix Overo, 2004. <http://svn.hylands.org/linux/gpio> [Online; accessed December 2013].
- [90] i-Tech. Subsea 7 Limited. <http://www.interventiontechnology.com> [Online; accessed December 2013].
- [91] Deep Sea Systems International Inc. <http://www.deepseasystems.com> [Online; accessed December 2013].
- [92] Deep Trekker Inc. <http://www.deeptrekker.com> [Online; accessed December 2013].
- [93] Desistec Inc. <http://www.desistek.com.tr> [Online; accessed December 2013].
- [94] Oceaneering International Inc. <http://www.oceaneering.com> [Online; accessed December 2013].
- [95] Submersible Systems Inc. <http://www.ssirovs.com> [Online; accessed December 2013].
- [96] International RoboSub Competition. AUVSI Foundation. <http://www.auvsifoundation.org/competitions/robosub> [Online; accessed December 2013].
- [97] International Submarine Engineering Limited. <http://www.ise.bc.ca> [Online; accessed December 2013].

- [98] Zool H. Ismail. Tracking Control Scheme for Multiple Autonomous Underwater Vehicles Subject to Union of Boundaries. *Procedia Engineering*, 41(Iris):1176–1182, January 2012.
- [99] Ivor Horton. *Ivor Horton's Beginning ANSI C++: The Complete Language*. Apress, third edition, 2004.
- [100] J. Jalbert, M. Shevenell, S. Chappel, R. Welsh, and R. Blidberg. EAVE III-untethered AUV submersible. In *OCEANS '88. A Partnership of Marine Interests. Proceedings*, pages 1259–1264 vol.4, 1988.
- [101] Japan Agency for Marine-Earth Science and Technology. <http://www.jamstec.go.jp/maritec/e> [Online; accessed December 2013].
- [102] Chris Kaminski, Tristan Crees, James Ferguson, Alexander Forrest, Jeff Williams, David Hopkin, and Garry Heard. 12 days under ice - An Historic AUV Deployment in the Canadian High Arctic. In *2010 IEEE/OES Autonomous Underwater Vehicles*, number 1, pages 1–11. IEEE, September 2010.
- [103] V. Kanakakis. *Path Planning and Navigation of Autonomous Underwater Vehicles Using Fuzzy Logic and Evolutionary Algorithms*. PhD thesis, Department of Production Engineering and Management, Technical University of Crete, Greece, 2007. In Greek.
- [104] V. Kanakakis and N. Tsourveloudis. Evolutionary Path Planning and Navigation of Autonomous Underwater Vehicles. In *Control Automation, 2007. MED '07. Mediterranean Conference on*, pages 1–6, June 27–29 2007.
- [105] V. Kanakakis and N. Tsourveloudis. Evolutionary Path Planning and Navigation of Autonomous Underwater Vehicles. In *Mediterranean Conference on Control and Automation. MED '07.*, pages 1–6, 2007.
- [106] V. Kanakakis, K. P. Valavanis, and N. C. Tsourveloudis. Fuzzy-Logic Based Navigation of Underwater Vehicles. *Journal of Intelligent and Robotic Systems*, 40(1):45–88, May 2004.
- [107] Xiaodong Kang, Hongli Xu, and Xisheng Feng. Fuzzy logic based behavior fusion for multi-AUV formation keeping in uncertain ocean environment. In *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*, pages 1–7, 2009.
- [108] A.C. Kapoutsis, S.A. Chatzichristofis, L. Doitsidis, J. Borges de Sousa, and E.B. Kosmatopoulos. Autonomous navigation of teams of unmanned aerial or underwater vehicles for exploration of unknown static & dynamic environments. In *Control Automation (MED), 2013 21st Mediterranean Conference on*, pages 1181–1188, 2013.
- [109] George Karras, Dimitra Panagou, and Kostas Kyriakopoulos. Target-referenced Localization of an Underwater Vehicle using a Laser-based Vision System. In *OCEANS 2006*, pages 1–6. IEEE, September 2006.
- [110] Kihun Kim and Hang S. Choi. Analysis on the controlled nonlinear motion of a test bed AUVSNUUV I. *Ocean Engineering*, 34(8-9):1138–1150, June 2007.
- [111] D. M. Kitchen and C. Packer. *Levels of Selection in Evolution*, volume 239, chapter Complexity in vertebrate societies, pages 176–196. Princeton University Press, 1999.
- [112] Hayato Kondo and Tamaki Ura. Navigation of an AUV for investigation of underwater structures. *Control Engineering Practice*, 12(12):1551–1559, December 2004.
- [113] Elias B. Kosmatopoulos. Brief paper: An adaptive optimization scheme with satisfactory transient performance. *Automatica*, 45(3):716–723, March 2009.
- [114] R. Kumar and J.A. Stover. A behavior-based intelligent control architecture with application to coordination of multiple underwater vehicles. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 30(6):767–784, 2000.



- [115] M. Kyo, E. Hiyazaki, S. Tsukioka, H. Ochi, Y. Amitani, T. Tsuchiya, T. Aoki, and S. Takagawa. The sea trial of 'KAIKO', the full ocean depth research ROV. In *OCEANS '95. MTS/IEEE. Challenges of Our Changing Global Environment. Conference Proceedings*, volume 3, pages 1991–1996 vol.3, 1995.
- [116] Lionel Lapierre. Robust diving control of an AUV. *Ocean Engineering*, 36(1):92–104, January 2009.
- [117] S. Leatherwood. Some Observations of Feeding Behavior of Bottle-Nosed Dolphins (*Tursiops truncatus*) in the Northern Gulf of Mexico and (*Tursiops cf. T. gilli*) off Southern California, Baja California, and Nayarit, Mexico. *Marine Fisheries Review*, 37(9):10–16, 1975.
- [118] Malrey Lee. Evolution of behaviors in autonomous robot using artificial neural network and genetic algorithm. *Information Sciences*, 155(1-2):43–60, October 2003.
- [119] Jia-wang Li, Bao-wei Song, and Cheng Shao. Tracking Control of Autonomous Underwater Vehicles with Internal Moving Mass. *Acta Automatica Sinica*, 34(10):1319–1323, 2008.
- [120] libserial. Serial port programming in C++. <http://sourceforge.net/projects/libserial> [Online; accessed December 2013].
- [121] Jindong Liu and Huosheng Hu. Biological Inspiration: From Carangiform Fish to Multi-Joint Robotic Fish. *Journal of Bionic Engineering*, 7(1):35–48, March 2010.
- [122] VideoRay LLC. <http://videoray.com> [Online; accessed December 2013].
- [123] DWTEK Co. Ltd. Chieng Ming Industries Co. Ltd. <http://www.dwtek-rov.com/> [Online; accessed December 2013].
- [124] Martin Ludvigsen, Geir Johnsen, Petter a. Lagstad, Asgeir J Sorensen, and Oyvind Odegard. Scientific operations combining ROV and AUV in the Trondheim Fjord. In *2013 MTS/IEEE OCEANS - Bergen*, pages 1–7. IEEE, June 2013.
- [125] John D. Madden, Ronald C Arkin, and Daniel R. MacNulty. Multi-robot system based on model of wolf hunting behavior to emulate wolf and elk interactions. In *2010 IEEE International Conference on Robotics and Biomimetics*, pages 1043–1050. IEEE, December 2010.
- [126] Maja J. Mataric and François Michaud. Behavior-Based Systems. In *Springer Handbook of Robotics*, chapter 38, pages 891–909. Springer, 2008.
- [127] MAKO AUV, Ocean Technology Lab. University of Victoria. <http://web.uvic.ca/~lacir/ocean> [Online; accessed December 2013].
- [128] DOER Marine. Deep ocean exploration and research. <http://www.doermarine.com> [Online; accessed December 2013].
- [129] Maritime & Ocean Engineering Research Institute. Korea Ocean Research and Development Institute. [http://www.moeri.re.kr/kordi\\_daeduck/eng](http://www.moeri.re.kr/kordi_daeduck/eng) [Online; accessed December 2013].
- [130] M. J. Mataric. Designing and Understanding Adaptive Group Behavior. *Adaptive Behavior*, 4(1):51–80, September 1995.
- [131] Miles G. McPhee. A Personal View of the ONR High Latitude Upper Ocean Physics Programs. *Arctic Research of the United States*, 18, 2004.
- [132] D.L Mech. *The Wolf. The Ecology and Behaviour of Endangered Species*. The Natural History Press, 1970.
- [133] Jean-Arcady Meyer and Agnes Guillot. Biologically-inspired robots. In Bruno Siciliano and Oussama Khatib, editors, *Handbook of Robotics*, chapter 61, pages 1–38. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [134] David Miller. *The Illustrated Directory of Submarines of the World*. MBI Pub. Co, 2002.

- [135] M. Morgado, P. Oliveira, and C. Silvestre. Tightly coupled ultrashort baseline and inertial navigation system for underwater vehicles: An experimental validation. *Journal of Field Robotics*, 30(1):142–170, January 2013.
- [136] Baptiste Mourre and Alberto Alvarez. Benefit assessment of glider adaptive sampling in the Ligurian Sea. *Deep Sea Research Part I: Oceanographic Research Papers*, 68:68 – 78, 2012.
- [137] Naval Undersea Warfare Center. Naval Sea Systems Command US Navy. <http://www.navsea.navy.mil/nuwc> [Online; accessed December 2013].
- [138] Steffen Nissen. Fast Artificial Neural Network Library. <http://leenissen.dk/fann/wp> [Online; accessed December 2013].
- [139] OceanServer Technology, Inc. <http://ocean-server.com> [Online; accessed December 2013].
- [140] Roatan Institute of Deepsea Exploration. Deepsea submarine idabel. <http://www.stanleysubmarines.com> [Online; accessed December 2013].
- [141] Navy Department Library: Naval Documents of the Revolutionary War. The submarine turtle. [http://www.history.navy.mil/library/online/sub\\_turtle.htm](http://www.history.navy.mil/library/online/sub_turtle.htm) [Online; accessed December 2013].
- [142] Yong-jai Park, Useok Jeong, Jeongsu Lee, Seok-ryung Kwon, Ho-young Kim, and Kyu-jin Cho. Kinematic Condition for Maximizing the Thrust of a Robotic Fish Using a Compliant Caudal Fin. *IEEE Transactions on Robotics*, 28(6):1216–1227, December 2012.
- [143] R. O. Petricig. *Bottlenose dolphin (Tursiops truncatus) in Bull Creek, South Carolina*. PhD thesis, University of Rhode Island, USA, 1995.
- [144] John Pike. Submarine History - The New Navy. <http://www.globalsecurity.org/military/systems/ship/sub-history4.htm> [Online; accessed December 2013].
- [145] Savas Piperidis and Nikolaos Tsourveloudis. Design and Control of an AUV in a virtual environment (in Greek). In *2nd Hellenic Robotics Conference*, 2010.
- [146] Savas Piperidis and Nikos C. Tsourveloudis. Testing controllers on ALE III: A low cost mini Autonomous Underwater Vehicle. In *21st Mediterranean Conference on Control and Automation*, pages 551–557. IEEE, June 2013.
- [147] Dynamic Programming. <http://chinmaylokesk.wordpress.com/2011/01/25/egg-dropping-puzzle-dynamic-programming-c> [Online; accessed December 2013].
- [148] Raboesch Models. Marine Accessories. <http://raboeschmodels.com> [Online; accessed December 2013].
- [149] Juan Rada-Vilela. fuzzylite: a fuzzy logic control library. <http://www.fuzzylite.com> [Online; accessed December 2013].
- [150] Kelly Ann Rossbach. Cooperative feeding among bottlenose dolphins (*Tursiops truncatus*) near Grand Bahama Island, Bahamas. *Aquatic Mammals*, 25:163–168, 1999.
- [151] SAAB Seaeye LTD. <http://www.seaeye.com> [Online; accessed December 2013].
- [152] B.K. Sahu, M.M. Gupta, and B. Subudhi. Fuzzy separation potential function based flocking control of multiple AUVs. In *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, pages 1429–1434, 2013.
- [153] Norimitsu Sakagami, Kouhei Ishimaru, Sadao Kawamura, Mizuho Shibata, Hiroyuki Onishi, and Shigeo Murakami. Development of an Underwater Robotic Inspection System using Mechanical Contact. *Journal of Field Robotics*, 30(4):624–640, July 2013.
- [154] Ingrid Schjølberg and Thor I. Fossen. Modelling and Control of Underwater Vehicle-Manipulator Systems. In *Conference on Marine Craft maneuvering and control*, pages 45–57, 1994.

- [155] SeaExplorer AUV Glider. ACSA ALCEN. <http://www.acsa-alcen.com> [Online; accessed December 2013].
- [156] 400 Series Buccaneer. Bulgin. <http://www.bulgin.co.uk/Products/Buccaneer/Buccaneer.html> [Online; accessed December 2013].
- [157] sf9domahrs. AHRS for Sparkfun's '9DOF razor IMU'. <http://code.google.com/p/sf9domahrs> [Online; accessed December 2013].
- [158] C.K. Shin and S.C. Park. Memory and neural network based expert system. *Expert Systems with Applications*, 16(2):145–155, February 1999.
- [159] D. Shinzaki, C. Gage, S. Tang, M. Moline, B. Wolfe, C.G. Lowe, and C. Clark. A multi-AUV system for cooperative tracking and following of leopard sharks. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4153–4158, 2013.
- [160] Sika Group. Marine Sealing Products. <http://www.sika.com> [Online; accessed December 2013].
- [161] Russell Smith. Open Dynamics Engine. <http://www.ode.org> [Online; accessed December 2013].
- [162] Laura Sorbi, Graziano Pio De Capua, Jean-Guy Fontaine, and Laura Toni. A Behavior-Based Mission Planner for Cooperative Autonomous Underwater Vehicles. *Marine Technology Society Journal*, 46(2):32–44, 2012.
- [163] C.C. Sotzing, J. Evans, and D.M. Lane. A Multi-Agent Architecture to Increase Coordination Efficiency in Multi-AUV Operations. In *OCEANS 2007 - Europe*, pages 1–6, 2007.
- [164] Ageotec s.r.l. <http://www.ageotec.com> [Online; accessed December 2013].
- [165] PE Stander. Cooperative hunting in lions: the role of the individual. *Behavioral Ecology and Sociobiology*, 29(6):445–454, February 1992.
- [166] P. B. Sujit and S. Saripalli. An Empirical Evaluation of Co-ordination Strategies for an AUV and UAV. *Journal of Intelligent & Robotic Systems*, 70(1-4):373–384, August 2012.
- [167] Subsea Tech. Mini ROV. <http://www.subsea-tech.com> [Online; accessed December 2013].
- [168] Texas Instruments Incorporated. <http://www.ti.com> [Online; accessed December 2013].
- [169] The Atlas Elektronik Group. Unmanned Vehicles. <http://www.atlas-elektronik.com> [Online; accessed December 2013].
- [170] The British Standards Institution. <http://www.bsigroup.com> [Online; accessed December 2013].
- [171] O Trullier, S I Wiener, A Berthoz, and J a Meyer. Biologically based artificial navigation systems: review and prospects. *Progress in neurobiology*, 51(5):483–544, April 1997.
- [172] Antonio Vasilijevic, Bruno Borovic, and Zoran Vukic. Underwater Vehicle Localization with Complementary Filter: Performance Analysis in the Shallow Water Environment. *Journal of Intelligent & Robotic Systems*, 68(3-4):373–386, August 2012.
- [173] Nikos I. Vitzilaios and Nikos C. Tsourveloudis. An Experimental Test Bed for Small Unmanned Helicopters. *Journal of Intelligent and Robotic Systems*, 54(5):769–794, 2009.
- [174] David Weaver. *The Encyclopedia of Ecotourism*. CABI Publishing, 2001.
- [175] Webots Commercial Mobile Robot Simulation Software. Cyberbotics Ltd, 2013. <http://www.cyberbotics.com> [Online; accessed December 2013].
- [176] Alfredo Weitzenfeld, Alberto Vallesa, and Horacio Flores. A Biologically-Inspired Wolf Pack Multiple Robot Hunting Model. In *2006 IEEE 3rd Latin American Robotics Symposium*, pages 120–127. IEEE, October 2006.

- [177] R. Wernli. AUVs-the maturity of the technology. In *OCEANS '99 MTS/IEEE. Riding the Crest into the 21st Century*, volume 1, pages 189–195 vol.1, 1999.
- [178] P. Woock. Deep-sea seafloor shape reconstruction from side-scan sonar data for auv navigation. In *OCEANS, 2011 IEEE - Spain*, pages 1–7, 2011.
- [179] P. Woock. Survey on suitable 3d features for sonar-based underwater navigation. In *OCEANS, 2012 - Yeosu*, pages 1–6, 2012.
- [180] B. Wursig. *Dolphin Cognition and Behavior: a Comparative Approach*, chapter Delphinid Foraging Strategies, pages 347–359. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.
- [181] Cao Xiang and Zhu Daqi. A survey of cooperative hunting control algorithms for multi-auv systems. In *Control Conference (CCC), 2013 32nd Chinese*, pages 5791–5795, 2013.
- [182] Qin Yan, Zhen Han, Shi-wu Zhang, and Jie Yang. Parametric Research of Experiments on a Carangiform Robotic Fish. *Journal of Bionic Engineering*, 5(2):95–101, June 2008.
- [183] Hiroshi Yoshida, Taro Aoki, Hiroyuki Osawa, Shojiro Ishibashi, Yoshitaka Watanabe, Junichiro Tahara, Tsuyoshi Miyazaki, and Kazuaki Itoh. A Deepest Depth ROV for Sediment Sampling and Its Sea Trial Result. In *Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies, 2007. Symposium on*, pages 28–33, 2007.
- [184] Junzhi Yu, Rui Ding, Qinghai Yang, Min Tan, and Jianwei Zhang. Amphibious Pattern Design of a Robotic Fish with Wheel-propeller-fin Mechanisms. *Journal of Field Robotics*, 30(5):702–716, September 2013.
- [185] Hu Yuli and Wang Jiajun. Study on Power Generation and Energy Storage System of a Solar Powered Autonomous Underwater Vehicle(SAUV). *Energy Procedia*, 16, Part C:2049 – 2053, 2012.
- [186] Dandan Zhang, Long Wang, Junzhi Yu, and Guangming Xie. Robotic fish motion planning under inherent kinematic constraints. In *2006 American Control Conference*, page 6 pp. IEEE, 2006.
- [187] Shaowei Zhang, Jiancheng Yu, Aiqun Zhang, and Fumin Zhang. Spiraling motion of underwater gliders: Modeling, analysis, and experimental results. *Ocean Engineering*, 60:1 – 13, 2013.
- [188] Side Zhao and Junku Yuh. Experimental study on advanced underwater robot control. *Robotics, IEEE Transactions on*, 21:695 – 703, 2005.
- [189] Zhou Zhong-Hai, Yuan Jian, Zhang Wen-Xia, and Zhao Jin-Ping. Virtual-Leader-follower Structure and Finite-Time Controller Based Cooperative Control of Multiple Autonomous Underwater Vehicles. In *Control and Decision Conference (CCDC), 2012 24th Chinese*, pages 3670–3675, 2012.
- [190] Marc Ziegler, Fumiya Iida, and Rolf Pfeifer. "Cheap" underwater locomotion: Roles of morphological properties and behavioural diversity. In *9th International Conference on Climbing and Walking Robots (CLAWAR 2006)*, 2006.
- [191] Kexu Zou, Chen Wang, Guangming Xie, Tianguang Chu, Long Wang, and Yingmin Jia. Cooperative control for trajectory tracking of robotic fish. In *2009 American Control Conference*, pages 5504–5509. IEEE, 2009.
- [192] Jean-Christophe Zufferey, Alexis Guanella, Antoine Beyeler, and Dario Floreano. Flying over the reality gap: From simulated to real indoor airships. *Autonomous Robots*, 21(3):243–254, September 2006.