

Design of an Autonomous Underwater Vehicle with Vision Capabilities

By

Ali Jebelli

A thesis presented to the Ottawa-Carleton Institute
for Electrical and Computer Engineering
in partial fulfillment to the thesis requirement for the degree requirements of

**PhD degree
in Electrical and Computer Engineering**



uOttawa

University of Ottawa
Ottawa, Ontario, Canada

June 2016

© Ali Jebelli, Ottawa, Canada, 2016

Abstract

In the past decade, the design and manufacturing of intelligent multipurpose underwater vehicles has increased significantly. In the wide range of studies conducted in this field, the flexibility and autonomy of these devices with respect to their intended performance had been widely investigated.

This work is related to the design and manufacturing of a small and lightweight autonomous underwater vehicle (AUV) with vision capabilities allowing detecting and contouring obstacles.

It is indeed an exciting challenge to build a small and light submarine AUV, while making tradeoffs between performance and minimum available space as well as energy consumption. In fact, due to the ever-increasing in equipment complexity and performance, designers of AUVs are facing the issues of limited size and energy consumption.

By using a pair of thrusters capable to rotate 360° on their axis and implementing a mass shifter with a control loop inside the vehicle, this later can efficiently adapt its depth and direction with minimal energy consumption. A prototype was fabricated and successfully tested in real operating conditions (in both pool and ocean). It includes the design and embedding of accurate custom multi-purpose sensors for multi-task operation as well as an enhanced coordinated system between a high-speed processor and accustomed electrical/mechanical parts of the vehicle, to allow automatic controlling its movements.

Furthermore, an efficient tracking system was implemented to automatically detect and bypass obstacles. Then, fuzzy-based controllers were coupled to the main AUV processor system to provide the best commands to safely get around obstacles with minimum energy consumption. The fabricated prototype was able to work for a period of three hours with object tracking options and five hours in a safe environment, at a speed of 0.6 m/s at a depth of 8 m.

Acknowledgements

First, I would like to express my honestly gratitude and appreciation to my supervisor, Professor Mustapha C.E. Yagoub for his encouragement, guidance, critics and friendship. Without his continued support and interest, this thesis would not have been the same as presented here. My sincere appreciation is extended to everyone from School of Electrical Engineering and Computer Science and Department of Mechanical Engineering of the University of Ottawa for their advice and comments.

Finally, a special thanks to my family for the support they provided me through my entire life and in particular, I must acknowledge my wife, Nafiseh, without whose love and encouragement, I would not have finished this thesis.

Table of Contents

Chapter 1. INTRODUCTION.....	1
1.1 Motivations	1
1.2 Contributions.....	2
1.3 Thesis Outlines.....	3
1.4 Publications and Awards	3
 Chapter 2. REVIEW OF AUV DESIGNS.....	5
2.1 Introduction	5
2.2 State of the Art and Suggested Designs	6
2.3 Fluid Mechanic for Submarines.....	7
2.3.1 Fluid dynamics	8
2.3.1.1 Buoyancy	8
2.3.1.2 Stability	8
2.3.2 Hydrodynamics	11
2.3.2.1 Volume Control Relationship	11
2.3.2.2 Conservation Equation (Conservation of Mass)	12
2.3.2.3 Momentum Equation (Conservation of Momentum).....	13
2.3.2.4 Drag.....	14
2.4 Thrust	15
2.4.1 Propellers	16
2.4.2 Podded propulsion devices.....	17
2.5 Energy	18
2.6 Navigation.....	20
2.6.1 Path Planning	20
2.6.2 Radio navigation	21
2.6.3 Navigation and Positioning Error.....	22
2.6.4 Inertial Measurement System.....	22
2.6.6 Dead Reckoning	23
2.7 Control Systems	23
2.7.1 Sliding mode control.....	23
2.7.2 PID controller.....	24
2.7.3 Fuzzy controller	26
2.9 Conclusion	28

Chapter 3. MECHANICAL DESIGN.....	29
3.1 Mechanical Design.....	29
3.1.1 Body Design.....	30
3.1.2 Locating Parts	32
3.2 Body Material and Body Strength	34
3.2.1 Applied pressures	35
3.2.2 Reliability Factor and Allowable Load.....	35
3.2.3 Anchor.....	37
3.2.4 Simulation Results	37
3.2.5 Conclusion	41
3.3 Mass shifter	42
3.4 Thrusters.....	44
3.5 Rotating Body	48
3.6 Modeling the System	49
3.6.1 Introduction to Coordinates System	49
3.6.2 Modeling the Gravity	51
3.6.3 Modeling the Forces.....	51
3.7 Display Control	55
3.7.1 Simulation Platform	56
3.7.2 GUI Environnement.....	56
3.7.3 VR Environnement	63
3.7 AUV Insulation	63
3.5 Conclusion	66
 Chapter 4. DESIGN OF ELECTRONICS PARTS	 68
4.1 Main Board and Sensors	68
4.2 Sensors and Sensor Boards	70
4.2.1 IMU Sensor	71
4.2.2 Compass Sensor	72
4.2.3 Pressure Sensor	74
4.2.4 Humidity and Temperature Sensor	74
4.2.5 Sensor Board	76
4.3 Transceiver.....	77
4.4 Camera and Light Board	79
4.5 Power and Battery	82
4.6 Access Valve Board.....	85
4.7 Engines.....	88
4.8 Engine Driver Board	89
4.9 Conclusion	92
 Chapter 5. OBJECT DETECTION AND TRACKING.....	 93
5.1 Introduction.....	93

5.2 Object Tracking Methods.....	94
5.2.1 Object Based Detection Method	94
5.2.2 Color-Based Methods	94
5.2.3 Feature-Based Methods.....	94
5.2.3 Overall Performance of the Above Mentioned Methods	94
5.3 Proposed Method	95
5.3.1 Motion-Based Segmentation Methods.....	95
5.3.2 Detecting Moving Objects: Background Subtraction Methods	96
5.3.3 Mean Shift.....	97
5.3.4 Color-Texture Histogram	98
5.4 Describing the Target.....	99
5.4.1 Tracking with Mean Shift	100
5.4.2 Local Binary Patterns.....	101
5.4.3 Target Description by Color-Texture Histogram.....	102
5.4.4 Tracking by Color-Texture Histogram.....	105
5.5 Results.....	107
5.6 Conclusion	109
 Chapter 6. CONTROL SYSTEM	 110
6.1 Introduction.....	110
6.2 Controller Design.....	111
6.2.1 Speed Controller	112
6.2.2 Pitch Controller	112
6.2.3 Depth Controller.....	113
6.2.4 Level Controller	115
6.3 Experimental Tests.....	116
6.3.1 Experimental Setup	116
6.3.2 Speed Command	119
6.3.3 Depth Command	121
6.3.4 Integrating Speed and Depth Commands.....	124
6.3.5 Integrating Speed, Pitch and Depth Commands	127
6.3.6 Direction Command.....	131
6.4 Controller Integration.....	133
6.5 Breakthrough.....	142
6.5.1 Fuzzy Logic Controller (FLC)	143
6.5.2 Fuzzy Controller Design	144
6.6 Vision Tests.....	149
6.6.1 First Test	149
6.6.2 Second Test	156
6.6.3 Third Test.....	161
6.7 Conclusion	167

Chapter 7. CONCLUSIONS AND FUTURE WORK.....	168
7.1 Conclusion	168
7.2 Future Works	169
 APPENDIX I.....	171
APPENDIX II	181
APPENDIX III.....	186
 REFERENCES.....	192

List of Figures

Figure 2.1: Schematic of an immersed body that is stable in roll [40]	9
Figure 2.2: Schematic of an immersed body that is unstable in roll [40]	9
Figure 2.3: Various conditions of longitudinal stability [42].....	10
Figure 2.4: Control volume [41]	11
Figure 2.5: Control volume for one-dimensional flow [43].....	12
Figure 2.6: Relative contributions to total drag as a function of L/d [46]	15
Figure 2.7: Propeller examples [41]	17
Figure 2.8: Podded propulsors [41]	
(a) Podded propulsor on a ship (b) Previous AUVSI competition thruster	18
Figure 2.9: The general block diagram of the fuzzy controller structure diagram	26
Figure 3.1: AUV: block diagram of the mechanical parts	29
Figure 3.2: Isometric scheme of the designed AUV.....	30
Figure 3.3: Different views of the designed AUV	30
Figure 3.4: Tilt Rotor V-22 [76]	31
Figure 3.5: Access Valve (1), Antenna Holes (2), Pressure Sensor (3), and Dolly Block of Servo Motor (4)	32
Figure 3.6: Six Designed Valves.....	33
Figure 3.7: Location of the Sensors	33
Figure 3.8: The Bottom Openings: (1) Bottom Camera, (2) and (3) Sensors	34
Figure 3.9: Locating Internal Components	34
Figure 3.10: The Location of Anchor	37
Figure 3.11: (a) Loading, (b) Used Mesh.....	37
Figure 3.12: Strain Changes.....	39
Figure 3.13: Stress Changes.....	39
Figure 3.14: Region of Maximum Stress	40
Figure 3.15: Stress More than 3 MPa	40
Figure 3.16: Displacement of the Points in the Body	41
Figure 3.17: The Point of Maximum Displacement	41
Figure 3.18: The Mass Shifter Mechanism.....	42
Figure 3.19: Mass Shifter.....	43
Figure 3.20: Mass Shifter: Location of the Cube Dumbbells, Ball Screw, and Stepper Motor	44
Figure 3.21: Fan Design in OpenProp	46
Figure 3.22: Propellers Designed in SolidWorks.....	46
Figure 3.23: The Isometric Scheme of the Housing	46

Figure 3.24: Design of the Housing	47
Figure 3.25: Different View of the Thrusters.....	47
Figure 3.26: The Propellers in the Thruster	48
Figure 3.27: The Servo Motor in the Dolly Blocks	48
Figure 3.28: Rotating Body and Location for Top Camera: (1) Cable Compartment, (2) Stepper Motor for 360 Degree, (3) Camera Body, (4) Camera Board, (5) Threaded Cap Body, (6) Top Camera, and (7) Transparent Bubble.	49
Figure 3.29: Earth-Centered Inertial (ECI) [95]	50
Figure 3.30: Earth-Centered Earth-Fixed (ECEF) [96].....	50
Figure 3.31: Physical Coordinates System	51
Figure 3.32: Archimedes' Principle [100]	52
Figure 3.33: Propulsion.....	53
Figure 3.34: Added Mass [103]	54
Figure 3.35: Initial Conditions	57
Figure 3.36: Platform Settings	58
Figure 3.37: Settings Parameters	58
Figure 3.38: Environmental Parameters.....	59
Figure 3.39: Setting Device Parameters.....	59
Figure 3.40: Linear Drag.....	60
Figure 3.41: Quadratic Drag	61
Figure 3.42: Added Mass	61
Figure 3.43: The Important Points' Location Vector.....	62
Figure 3.44: Control.....	62
Figure 3.45: VR Environment.....	64
Figure 3.46: The Paste Between the Seal.....	65
Figure 3.47: The Screws	66
Figure 4.1: Electronics Parts	69
Figure 4.2: Block Diagram of Main Processor	70
Figure 4.3: MPU-9250 Sensor [123].....	71
Figure 4.4: SparkFun HMC6343 Breakout [124], [125].....	73
Figure 4.5: PCB of the MPU Sensor Board	73
Figure 4.6: PCB of the Compass Board.....	74
Figure 4.7: Block diagram of MS5803-14BA [126] and Sensor Board	75
Figure 4.8: Humidity and Temperature Sensor [127]	75
Figure 4.9: Block Diagram and Schematic of the Sensor Board	76
Figure 4.10: Sensor Board, top Board with Dimensions $67.40 \times 48.40 \times 2 \text{ mm}^3$ and Bottom Board with Dimensions $69.85 \times 49.80 \times 2 \text{ mm}^3$	77
Figure 4.11: PCB of the Bottom Board.....	78
Figure 4.12: PCB of the Top Board	79
Figure 4.13: The Main Camera Board	80
Figure 4.14: The Bottom Camera	80
Figure 4.15: LED Board: (a) Schematic, (b) PCB, (c) Final Circuit	81

Figure 4.16: (a) View of the Robot at Night, (b) LEDs Performance with Maximum Power	82
Figure 4.17: a) Block Diagram and b) Schematic of the Power Distribution board	84
Figure 4.18: Power Distribution Board (a), Top View, (b) PCB.	85
Figure 4.19: (a) Schematic of the Horizontal Access Valve Board, (b) Access Valve Board, (c) PCB of the Horizontal Access Valve Board with Dimensions $52.40 \times 87.40 \times 2 \text{ mm}^3$	86
Figure 4.20: (a) Schematic of the Vertical Access Valve Board, (b) PCB of the Vertical Access Valve Board with Dimensions $17.95 \times 87.40 \times 2 \text{ mm}^3$	87
Figure 4.21: Motor Performance at 12 V and 24 V [135]	88
Figure 4.22: Dynamic Torque Curves [137]	89
Figure 4.23: a) Block Diagram and b) Schematic of the Engine Driver Board	90
Figure 4.24: Engine Driver Board with Dimensions $157.50 \times 120 \times 2 \text{ mm}^3$ (a) Top View, (b) PCB, (c) BTN7971B Driver Module	91
Figure 5.1: An Example of LBP Operator with $P = 8$ and $R = 1$ [158]	103
Figure 5.2: 9 Flat Patterns in the Model $\mathbf{LBP}_{8,1}^{riu2}$ [142]	103
Figure 5.3: (a) Tracking Windows, (b) Mask Extracted by Uniform Patterns of LBP Majority	105
Figure 5.4: The Original Image Captured by the Robot Camera	107
Figure 5.5: Result of Applying the Derivative Operator	107
Figure 5.6: Detecting the Desired Object in the Image	108
Figure 5.7: Result of the Test on Underwater Video with the Proposed Method (From (1) to (5), the Pictures Show the Robot Approaching the Obstacle)	109
Figure 6.1: System Response to the Proportional Controller	114
Figure 6.2: System Response with Proportional-Integral Controller	115
Figure 6.3: Final Prototype	117
Figure 6.4: Internal View of the Robot	118
Figure 6.5: The AUV Remains on the Water When We Are Sending the Commands to Begin Its Mission	118
Figure 6.6: AUV Response (Black) to Speed Command (Blue)	119
Figure 6.7: Speed Error	120
Figure 6.8: Left Engine Revolution Level	120
Figure 6.9: Right Engine Revolution Level	121
Figure 6.10: Acceleration Towards x-axis	121
Figure 6.11: Robot Response (Black) to Selected Depths (Blue)	122
Figure 6.12: Position of the Moving Weight (Mass Shifter)	123
Figure 6.13: Pitch Angle	123
Figure 6.14: Rotational Speed Around y-axis	124
Figure 6.15: Vertical Speed	124
Figure 6.16: Depth	125
Figure 6.17: Vertical Speed	125

Figure 6.18: Left Servo Angle	126
Figure 6.19: Right Servo Angle	126
Figure 6.20: Left Servo Angle	127
Figure 6.21: Right Servo Angle	127
Figure 6.22: Speed Response	128
Figure 6.23: Depth Response	129
Figure 6.24: Pitch Angle	129
Figure 6.25: Mass Shifter Variation.....	130
Figure 6.26: Left Servo Angle	130
Figure 6.27: Right Servo Angle	131
Figure 6.28: Robot Response (Black) to Direction Angle Command (Blue)	131
Figure 6.29: Rotational Speed Around z-axis	132
Figure 6.30: Left Servo Angle	133
Figure 6.31: Right Servo Angle	133
Figure 6.32: Final Control Algorithm	134
Figure 6.33: Desired Speed	135
Figure 6.34: Desired Depth.....	135
Figure 6.35: Desired Direction Angle.....	136
Figure 6.36: Robot Response (Black) to Speed Command (Blue)	136
Figure 6.37: Vertical Speed	137
Figure 6.38: x-axis Acceleration	137
Figure 6.39: y-axis Acceleration	138
Figure 6.40: z-axis Acceleration	138
Figure 6.41: Robot Response (Black) to Depth Command (Blue)	139
Figure 6.42: Robot Response (Black) to Direction Angle Command (Blue)	139
Figure 6.43: Rotational Angle Around x-axis.....	140
Figure 6.44: Rotational Speed Around y-axis.....	140
Figure 6.45: Rotational Speed Around z-axis.....	141
Figure 6.46: The Status of the Moving Weight	141
Figure 6.47: Left Servo Angle	142
Figure 6.48: Right Servo Angle	142
Figure 6.49: Block Diagram of a Fuzzy Controller Structure.....	143
Figure 6.50: An Example of Image Scoring system. The original Image (Top left) is Subdivided into Six Sections (Top Right), i.e., Three Vertical Bands (Bottom Left) and Three Horizontal Bands (Bottom Right)	144
Figure 6.51: Input Membership Functions for Fuzzy Systems of DC Engines	146
Figure 6.52: Output Membership Functions for Fuzzy Systems of DC Engines.....	147
Figure 6.53: Input Membership Functions for Servo Motors' Fuzzy System	148
Figure 6.54: Output Membership Functions for Servo Motors' Fuzzy System.....	149
Figure 6.55: Control Algorithm Flowchart	150
Figure 6.56: A Sample of Control and Display Unit When the Robot at a Depth of 2m	151

Figure 6.57: Image Recognition Without Obstacle	151
Figure 6.58: Depth Evolution vs. Time.....	152
Figure 6.59: Speed Evolution vs. Time.....	152
Figure 6.60: Side Angle	153
Figure 6.61: Roll Angle	153
Figure 6.62: Pitch Angle	154
Figure 6.63: Mass Shifter Movement	154
Figure 6.64: Left Servo Angle	155
Figure 6.65: Right Servo Angle	155
Figure 6.66: Left Engine Speed	156
Figure 6.67: Right Engine Speed	156
Figure 6.68: A Sample of Control and Display Unit When the Robot at a Depth of 1m	157
Figure 6.69: Detection of an Obstacle.....	157
Figure 6.70: Robot Response (Black) to Depth Change (Blue).....	158
Figure 6.71: Speed Variation	158
Figure 6.72: Roll angle Variation	159
Figure 6.73: Pitch angle Variation	159
Figure 6.74: Mass Shifter Variation.....	160
Figure 6.75: Left Engine Speed Variation	160
Figure 6.76: Right Engine Speed Variation	161
Figure 6.77: A Sample of Control and Display Unit When the Robot at a Depth of 1.5 m	162
Figure 6.78: Robot Performance and Obstacle Detection in the Third Test Original images: blue background, processed images: black and white. Top set: The robot detects an obstacle on its way, Middle set: According to the fuzzy decision, the robot turns right, Third set: no more obstacle in front of the robot and the robot can move forwards	163
Figure 6.79: Robot Response (Black) to Depth Change (Blue).....	164
Figure 6.80: Robot Response (Black) to Speed Change (Blue).....	164
Figure 6.81: Robot Response (Black) to Side Angle Change (Blue).....	165
Figure 6.82: Roll Angle Variation	165
Figure 6.83: Pitch Angle Variation	166
Figure 6.84: Mass Shifter Variation.....	166

List of Tables

Table 2.1: PID controller features.	25
Table 3.1 PTFE mechanical properties and selected safety factor.....	36
Table 4.1: Description of the bottom board.	78
Table 4.2: Description of the top board.	78
Table 4.3: Description of the LED board.....	81
Table 4.4: Required Voltages.....	83
Table 4.5: Description of the Power Distribution Board	83
Table 4.6: Description of the Horizontal Access Valve Board	87
Table 4.7: Description of the Vertical Access Valve Board	87
Table 4.8: General Specifications of Baureihe 1.13.044.XXX for 12V & 24V [135].....	88
Table 4.9: General Specifications of 16HS4003-06N and 16HS4007-01N [137].....	89
Table 4.10: General Specifications of XQ-S4016D and XQ-S4020D [138]	89
Table 4.11: Description of the Engine Driver Board	92
Table 5.1: Assessment of Accuracy and Speed in Tracking the Target by Three Available Methods.....	108
Table 5.2: Number of iterations in mean shift algorithm until there is no blank space left unlabeled	108
Table 6.1: PID Controller values from the Ziegler-Nichols method [161].....	111
Table 6.2: Controlling coefficients for the whole system.	134
Table 6.3: Fuzzy rules to decide dc motors performance decision to move the robot in yaw degrees of freedom (P: Forward rotation of the motors, Z: OFF, N: Backward rotation of the motors	146
Table 6.4: Fuzzy rules for decision performance of servomotors to move the robot in a predefined pitch angle (Negative numbers indicate downward motion and positive numbers indicate upward movement of servo motors	147
Table 6.5: The Fuzzy systems' features [166]-[169]	148
Table 6.6: Image scoring system for this test.....	162
Table 6.7: Fuzzy decision system inputs.....	163

List of Variables

A	Area
A	Surfaces
AJ	Advance coefficient
A_B^C	Status of the system B to the device C
B	Reference system
$b(x_i^*)$	Relates x_i^* to the histogram
B_1	X axis of the system B
B_3	Z axis of the system B
B_2	Y axis of the system B
C	Constant
C^*	Desired external amount of y
CB	Center of Buoyancy
C_D	Drag coefficient
CG	Center of Gravity
C_h	Pressure of the water column
CS	Control surface
CV	Control Volume
c	Acceleration
c_B^H	Acceleration of system B to system H
C_A''	Added Coriolis matrix
$d\forall$	Element of volume in CV
d	Distance
D	Drag
D''	Propeller diameter
$\frac{dNF}{dt}$	Total rate of change of N
$\overrightarrow{dA}$	Area vector
dF	Differential force
dM	Differential moment
DS	Deviation of the sensor
dS	Differential surface
Dse	Measured depth by sensors
D	Derivative

D_{SF}	Skin friction drag force
E	Earth-Centered Earth-Fixed system
E'	Coefficient
$e_1 (X_1^E),$	System E z axis
$e_2 (X_2^E)$	System E x axis
$e_3 (X_3^E)$	System E y axis
e_p	Nose angle error
e_u	Speed error
e_y	Direction error
e^*	Feedback error
ET	Energy
EV	Error of vehicle location
e_D	Depth error
$\overrightarrow{F_B}$	Vector which includes all body forces acting on the CV
$\overrightarrow{F_S}$	Vector which includes all surface forces acting on the CS
F_r	Right thrust force
F_A	Added mass force
F_W	Weight force
$F_{hydrodynamics}$	Hydrodynamic force
F	Force
FB	Buoyancy Force
FH	Force on the system under the hotel power
F_l	Left thrust force
F_B	Force to the system B
$F_{Buoyancy}$	Buoyancy force
g''	Gravitational Acceleration
G	Coordinate system
$G(s)^*$	Laplace transform transfer function
g_c	Color intensity
g_p	Center of circle
h	Bandwidth
H	System H
h''	Height
IE	Initial error of the direction of movement,
I	Moment of Inertia
J	Vector
$k.RD_{D_1}$	Depth controller integral coefficient at rest.
$k.PC_{D_1}$	Depth controller proportional coefficient at rest
K'	Steady state relation between rudder angle and heading.

$k.PC_{D_2}$	Depth controller proportional coefficient during movement
$k.PC_y$	Proportional coefficient of the controller
kPB_p	Controller proportional coefficient
$k.LC_y$	Integral coefficient.
K_X'	Moment at Rotation on x axis
K_Y'	Moment at Rotation on y axis
K_Z'	Moment at Rotation on z axis
$K(s)$	Feedback compensator
$k(x)$	Kernel function
K_d ,	Differential coefficient
K_i	Integral coefficient
K_p ,	Proportional coefficient
kS_u	Speed controller integral coefficient.
ki_{D_2}	Depth controller integral coefficient during movement.
L_p^{BH}	Total angular momentum
lD_d	Desired depth
$\dot{m}$	Fluid flow rate through the propulsion device
m	Mass
m''	Number of feature space dimension
ME	Modulus of elasticity
m_B	System mass
M_A	Added mass matrix
NF	Net flux
N_u	Linear viscosity along x
N_r	Linear viscosity along r
N_v	Linear viscosity along y
N_w	Linear viscosity along z
N_q	Linear viscosity along q
$N_{q q }$	Quadratic coefficients along q
N_p	Linear viscosity along p
$N_{p p }$	Quadratic coefficients along p
$N_{r r }$	Quadratic coefficients along r
$N_{u u }$	Quadratic coefficients along x
$N_{w w }$	Quadratic coefficients along z
n^*	Sensor noise
N'	Moment at Rotation on z axis
O_B^B	Resultant momentum exerted on system B
$\hat{p}(y)$	Target candidate model
$\hat{p}_u(y)$	Probability of feature u in model $\hat{p}(y)$

P_{water}	Pressure of the water column
P_{atm}	Atmospheric pressure
$P^*(s)$	Pre-filter function
Pe	Power
PH	Hotel power
P_{in}	Pressure on the inner surface of the body
P_{out}	Pressure on the outer surface of the body
P_t	Total pressure on the body
p'	Angular velocity vector on x axis
p_B^H	Linear momentum of the system B versus the system H
$\hat{q}$	Target
q'	Angular velocity vector on y axis
q	Elements of the quaternion
$\hat{q}_u$	Probability of feature u in model $\hat{q}$.
riu2	Immutable with rotation
R	Parameters bandwidth
RV	Distance of travel or range of vehicle
R_e	Earth's radius
r'	Angular velocity vector on z axis
SA	Speed error along the path,
SW	Speed error within the path
S_y	Yield stress
t	Time
T	Thrust
$T(s)^*$	Closed-loop transfer function
TC	Thrust coefficient
TH_2	Controller output that indicates the engines' angle towards horizon
TP	Pitch controller output
TH_1	Controller output determined by the engines' revolution (%)
TL	Controller output, showing the difference between the servo motors
TS	Controller output in terms of engine revolution percentage
us_d	Desired speed
use	Speed measured by the sensors.
u'	Velocity vector on x axis
$\vec{V}$	Velocity vector
V'	Volume
v	Velocity
v_G^H	Velocity of the center G to the system H
v'	Velocity vector on y axis
v_e	Exit velocity
v_i	Inlet velocity

v_p	Propeller velocity
V_{pw}	propeller velocity in the water
VS	Shear force
W	Weight
w_i	Weights
W_{body}	Weight of body
$W_{components}$	Weight of components
ω^{BH}	Angular velocity of B relative to system H
ω^{BA}	Angular velocity of the system B relative to system A
ω^{BE}	Angular velocity of B relative to the Earth
W_B^R	Transfer matrix of system B to system W
w'	Velocity vector on z axis
X^*	Current vehicle location on North
X_C	Vector from the circle to the vehicle X axis
x_i	Location of pixels in the target
X_L	Position of loiter on X axis
X_p^*	Waypoint in the local navigation reference on North
X_s^*	Start position of the tracking trajectory on North
X_X'	Position at movement along x axis
X_Y'	Position at movement along y axis
X_Z'	Position at movement along z axis
X_u	Linear viscosity along x
$\{x_i\}_{i=1...n_h}$	Pixel location in the target candidate region with center y
X''	Force at movement along x axis
Y^*	Current vehicle location on East
y_1	Location
Y_C	Vector from the Leviter circle to the vehicle Y axis
Y_L	Position of loiter Y axis
Y_p^*	Waypoint in the local navigation reference on East
Y_s^*	Start position of the tracking trajectory on East
$Y_{v v }$	Quadratic coefficients along y
yl_d	Desired direction angle
$Y_{v v }$	Quadratic coefficients along y
yl_{sen}	Direction angle determined by the sensors.
Y''	Force at movement along y axis
Z''	Force at Movement along z axis
Z'	Position at Movement along z axis
zs	Elevation of the section
β	Intersection angle
β_0	Initial intersection angle
δ	Impulse function

Δ	Off-track position
δ^*	Actuator signal
η	NF per unit mass
μ_b	Bottom horizontal band scoring in fuzzy system
μ_L	Left vertical band scoring in fuzzy system
μ_m	Middle horizontal band scoring in fuzzy system
μ_M	Middle vertical band scoring in fuzzy system
μ_R	Right vertical band scoring in fuzzy system
μ_t	Top horizontal band scoring in fuzzy system
ρ	Density
$\sigma_{Ultimate}$	Ultimate stress load
$\sigma_{Working}$	Working stress load
σ	Stress
σ_y	Stress on y axis
ε	Strain
σ'	Von Mises stress
θ_d	Pitch angle
ρ	Fluid density
ψ	Euler angles at Rotation on z axis
θ_{sen}	pitch angle measured by the sensors
σ_z	Stress on z axis
σ_x	Stress on x axis
ρ	Density
μ	Earth's gravitational constant
ϕ	Euler angle at Rotation on x axis
θ	Euler angle at Rotation on y axis
$\dot{\Omega}_{e.t}^E$	Hour angle
τ^*	Dominant system time constant
Ψ^*	Carrier heading
Ψ_{com}	Heading command
Ψ_c	Track relative heading from the vehicle
Ψ_T	Track relative heading (x, y)

List of Acronyms

2D	Two-dimensional
3D	Three-dimensional
ADC	Analog-to-digital converter
AUV	Autonomous Underwater Vehicle
ECEF	Earth-Centered Earth-Fixed
ECI	Earth-Centered Inertial
F.S	Safety factor
FLC	Fuzzy Logic Controller
GFSK	Gaussian Frequency Shift Keying
GPS	Global positioning system
GUI	Graphical User Interface
HD	High-definition
I2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IR LED	infrared light-emitting diode
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
LBP	Local Binary Pattern
LDR	Light Dependent Resistor
LORAN	Long Range Navigation
MPU	Magnetic pick-up
MS	Mass Shifter
MSV	Manned Submersible Vehicles
MULE	Micro-Underwater Low-cost Explorer
N	Negative,
NB	Negative Big.
NED	North East Down
NS	Negative Small
P	Positive
PB	Positive Big
PI	Proportional integral
PID	Proportional Integral Differential
PS	Positive Small
PSB	Printed circuit board
PTFE	Polytetrafluoroethylene
PWM	Pulse Width Modulation
RDF	Radio Direction Finder
ROV	Remotely operated vehicle

SPI	Serial Peripheral Interface
UART	universal asynchronous receiver/transmitter
USBL	Ultra-Short Base Line
VR	Virtual reality

CHAPTER 1

INTRODUCTION

1.1 Motivations

In the past few decades, the number of autonomous underwater vehicles (AUVs) has significantly increased and many research centers have focused on this field, mainly because of the high ability of these devices in analyzing and understanding water environment as well as in collecting information. Consequently, their cost is significantly decreasing while their performance is increasing, making them very attractive to both researchers and industrials [1]-[15].

Based on their mission, these vehicles are capable to map the ocean floor for oil and gas facilities, to help oceanographic research, and to develop a better understanding of the oceans. Underwater vehicles can have unique characteristics; for instance, they are able to explore places inaccessible to humans. However, their design is still challenging mainly because of their intrinsic property: autonomy. In fact, issues related to vehicle balance and movement, obstacle detection, energy saving, etc., substantially reduce their development and usage [16]-[20].

On the other side, to successfully conduct long-term missions, AUVs should be able to detect and bypass obstacles. This impose embedding cameras and image processing tools to scan the ocean and capture/process data; this later being closely dependent on the light conditions in the water and/or the nature of the obstacles [21]-[25].

1.2 Contributions

In this thesis, an enhanced small and light autonomous vehicle has been designed and implemented with the ability of imaging and passing through obstacles. The main contributions of this thesis include designing and implementing a pair of thrusters capable to rotate 360° on their axis and a mass shifter system with a control loop inside the device. Thus, allowing the AUV to efficiently adapt its depth and direction with minimal energy consumption. Besides the above contributions, the present work includes:

- Modelling the AUV dynamics and the coupling between the rotating thrusters and the internal mass.
- designing and embedding customized multi-purpose sensors for multi-task operation.
- implementing an enhanced coordinated system between a high-speed processor and customized electrical/mechanical parts to allow automatic control of the AUV movements.
- designing customized electronic boards with the objective of significantly reducing connections and cables, without sacrificing the overall device performance.
- designing a mini-PC board to control the overall operation of the AUV.
- designing an efficient tracking system to automatically detect and bypass obstacles. Based on data collected by two embedded cameras with night vision capabilities, an image processing technique was developed to target obstacles. Then, fuzzy-based controllers were coupled to the main processor system to provide reliable commands to safely get around obstacles.
- Building a working prototype and successfully testing it in real operating conditions (pool and ocean).

1.3 Thesis Outlines

The remainder of the thesis is organized as follows. Chapter 2 introduces different AUV types and compares their performance. Also, the forces acting on a submarine and the types of thrusters and as well as controlling systems are also investigated.

Chapter 3 presents the designed AUV, all its components including the designed thrusters and mass shifter. This section includes a dynamic analysis of the AUV with various simulation tests of its body.

Chapter 4 deals with the design of the main processor system with all the used sensors in it. It presents the custom electronic boards especially manufactured for optimal functionality.

Chapter 5 reviews the existing methods for object tracking. In this section, an enhanced technique is proposed by using both color and texture histogram to trace a target and apply it to a mean shift algorithm for underwater images.

Chapter 6 presents the designed controlling system for AUV balance in different depths and swings and the coordinating between the thrusters and the embedded AUV's mass shifter in order to adapt the device movement and depth to detected obstacles. Then, tests demonstrated the efficiency of the above approach.

Chapter 7 summarizes the contributions of this thesis and briefly explores some directions of future work on this subject.

1.4 Publications and Awards

Journal Papers – Published

1. A. Jebelli *et al.*, “Development of Sensors and Microcontrollers for Small Temperature Controller Systems” *Journal of Automation and Control Engineering* Vol. 3, No. 4, pp. 322-328, August 2015.

2. A. Jebelli *et al.*, “Fuzzy Logic PID Based Control Design for a Small Underwater Robot with Minimum Energy Consumption” International Journal of Mechanical Engineering and Robotics Research Vol. 5, No. 3, July 2016.

Conference Papers – Published

1. A. Jebelli *et al.*, “Fuzzy Logic PID Based Control Design for a Small Underwater Robot with Minimum Energy Consumption” 2nd International Conference on Robotics and Automation Engineering (ICRAE 2015), Beijing, China.
2. A. Jebelli *et al.*, “Submarine robot sealing with minimum cost and environmental pollution”. International Ocean and Polar Engineering Conference (ISOPE 2016) Rhodes, Greece.
3. A. Jebelli *et al.*, “Efficient Robot Vision System for Underwater Object Tracking”. 2nd IEEE International Conference on Control Science and Systems Engineering (CCSSE 2016), Singapore.

Awards and Honours

1. Best Design Project for the “Design of an Intelligent Underwater Robot with Flexible Thrusters,” *Faculty of Engineering*, University of Ottawa, Canada, March 2016.

CHAPTER 2

REVIEW OF AUV DESIGNS

2.1 Introduction

Underwater vehicles can be grouped into three major groups, namely:

- Manned Submersible Vehicles (MSVs), capable to perform complex tasks with the help of human interventions and intelligence;
- Remotely Operated Vehicles (ROVs), remotely controlled and corded vehicles to transfer power, sensor data, and control commands between the operators on the surface and the underwater vehicle;
- Autonomous Underwater Vehicles (AUVs), unmanned, tether-free vehicles controlled by onboard computers to perform certain tasks.

Both ROVs and AUVs are effective in deep-sea monitoring. ROVs can be controlled in real time: an experienced pilot can control the vehicle while a scientist can provide him/her with mission-level directives. The pilot usually uses live videos provided by on-board cameras to monitor the vehicle.

AUVs are commonly known as unmanned underwater vehicle. AUVs can be used for underwater survey without operator intervention. When a mission is complete, the AUV will return to a pre-programmed location where the data can be downloaded and processed.

2.2 State of the Art and Suggested Designs

Two main types of Autonomous Underwater Vehicles (AUVs) have been proposed in the last few years. The main contribution of the first group, classified as “amateur” [26]-[28], is to propose new ideas/directions for researchers and therefore, are worth to mention. A common factor in almost all amateur platforms is using piping equipments, which is not only light and cost-effective but can be installed quite easily and there is no need for waterproofing them as well. Outside wiring is another property of these platforms.

On the other side, professional designs used in industry [29]-[36] exhibit, as expected, much higher capabilities and performance vs. the “amateur” designs. They find applications in construction, platform cleaning and inspection, subsea cable burial and maintenance, deep water salvage, remote tool deployment, subsea pipeline completion, to name a few.

However, they usually use a number of vertical and horizontal fixed thrusters, which increases weight and power consumption and reduces the device maneuverability in narrow and winding paths. Therefore, one can opt for mobile thrusters instead of fixed vertical/horizontal thrusters in order to both increase maneuverability and save energy. In fact, when thrusters are fixed, all thrusters should simultaneously be “on” to create a simultaneous vertical-horizontal movement to reach the desired depth/position; it will then lead to prodigality of energy and reduce device speed and maneuverability.

In this work, we designed a research prototype which main characteristics are the simultaneous use of two thrusters located in a specific angle so both vertical and horizontal forces needed are provided simultaneously. Furthermore, the change of angle can be made possible by the instant movement of an engine stopper, which consumes much less energy than the constant movement of a thruster.

In designing this AUV, we also included a mass shifter, first because of the thruster movement and second to make the maneuver possible in the direction of the pitch. The whole set can have two movement modes. In the first mode, the body should, by the help of the mass shifter, have a constant horizontal movement and its movement towards vertical and horizontal directions should be done through a change in the thruster angle. In the second mode, the thrusters should be kept fixed in a horizontal direction and the vertical movement should be made possible through a change in the body angle in the pitch direction. The first mode is used wherever there is little room of maneuver or to pass obstacles while the second mode is used to preserve and store the source of energy and change in great depths.

Note that this structure is innovative, even if some complications in terms of controlling issues can be expected. We have also implemented an efficient method for tracking underwater objects using two cameras to execute the expected tasks safely with a wide field of view and also because sensing in aquatic environments results in image distortion occurring by refraction of light due to the difference of refractive indices of air, watertight container and water. To address these issues, we introduced a method to correct the images and a ray tracing method to remove the effect of refraction.

2.3 Fluid Mechanic for Submarines

Designing an AUV to carry out a specific mission requires detailed investigation of each part and determining its effect on the entire system and other processor systems. For example, the carrier body affects the amount of drag that occurs in the carrier and thus requires a special thrust to push it in water. The amount of required thrust will be effective on the propulsion systems and CPU power requirements and system design. Power system affects the processor and computer sensors that will be supported. This section provides a fundamental understanding on the following issues [37]:

- Fluid dynamics
- Hydrostatic

2.3.1 Fluid dynamics

2.3.1.1 Buoyancy

The buoyancy force F_B is an upward net force applied on an object created by the fluid (in this case water) surrounding it:

$$F_B = \rho g V \quad (2.1)$$

Here ρ is the water density, g the gravitational acceleration and V the object volume. In other words, the buoyancy force equals the weight of the displaced fluid. Net buoyancy is the difference between buoyancy force and the weight of the carrier body (W_{hull}) and its parts ($W_{components}$).

$$NetBuoyancy = F_B - W_{hull} - W_{components} \quad (2.2)$$

If a floating object is neutral when it is submerged, it can be considered as weightless. By this assumption, the object must move the amount of water equal to its own weight. Neutral buoyancy is often targeted by submarine carriers. Achieving this target is possible by the adjustment of the carrier weight through adding or removing heavy materials dumped at the bottom of the carrier (ballast). The submarines use reservoirs drowned or evacuated so that the buoyancy level is set in terms of the requirements [38].

2.3.1.2 Stability

Center of buoyancy (CB) is a point on which buoyancy force is applied presumably. This occurs at the center of the displaced mass and when the submerged carrier comes in, it is left at the same spot. However, when the carrier comes to the surface, the center of buoyancy is modified. Carrier weight (W) is applied on the center of gravity (CG) and this point does not change unless the mass is displaced in the carrier [39]-[41].

In a stable condition, F_B and W produce a couple (torque) that tends to the original position (balance) of the ship known as vertical anchor or restorer. There is hydrostatic stability for a submerged carrier that CG is lower than CB as presented in Figure 2.1. It determines distance between these two points and dictates the stability characteristics of the device. Under unstable conditions the anchor prevents the device to return the correct direction as shown in Figure 2.2. The practical problem here is to determine the minimum distance to provide sufficient vertical anchor. Under static conditions, the vertical anchor should be large enough so that the interior parts do not lead to unstable conditions. The distance between CB and CG provides effective conditions for the equations of motion which creates resistance to flow (vertical rotation). However, increasing this distance will increase the device stability, leading to an impractical amount of ballast [39]-[41].

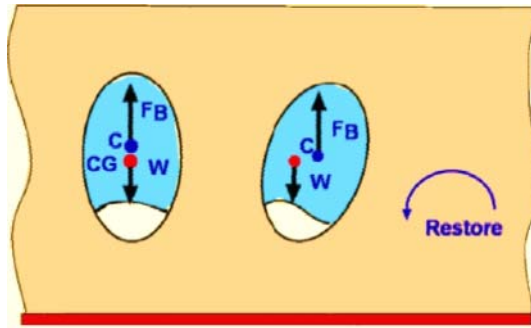


Fig. 2.1: Schematic of an immersed body that is stable in roll [40].

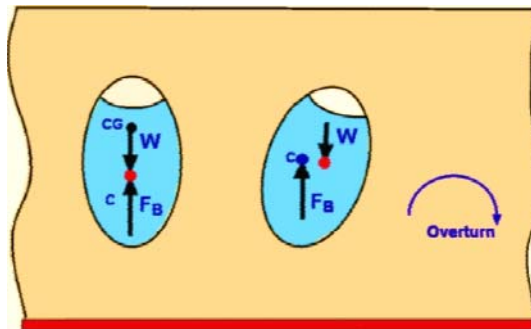


Fig. 2.2: Schematic of an immersed body that is unstable in roll [40].

Ballast can be described as fixed or variable. The fixed ballast includes any internal part and weights with positive buoyancy, so that the device reaches the desired buoyancy.

Ballast is also used to coincide the longitudinal positions of CG and CB (pitch stability), as shown in Figure 2.3, and to adjust the vertical distance between CG and CB (pitch and roll stability). If a cylindrical shape is used for the carrier body, CB will be close to the cylinder and the negative ballast at the bottom of the body transfers the CG downward. Tubular frame can be used to increase flexibility. However, in case of resistance and the likelihood of impact damage, they should be weaker.

Variable ballast is used to modify buoyancy more than modifying the weight or depth (surface or submerged). Variable ballast systems can include soft or hard (rigid) tanks. The problem with soft tanks is the volume change when changing carriers' depth. The system that fill and empty the rigid tank can be implemented easily. Opening a valve allows the water to enter the tank; a pump or piston can be used to remove the water with force. Ballast systems including tanks, motors, pistons and switches are available in the market.

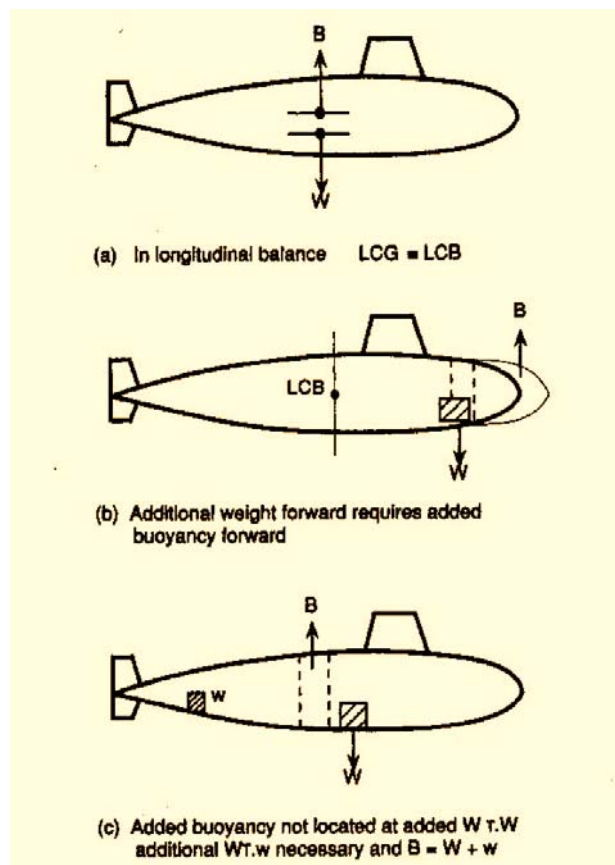


Fig. 2.3: Various conditions of longitudinal stability [42].

2.3.2 Hydrodynamics

2.3.2.1 Volume Control Relationship

The control volume includes a part of the space that is studied. This space may be real or hypothetical. Also the border between this space and the surrounding environment is known as the control surface. A control volume relationship is especially applied in hydrodynamic problems because the analysis focuses on a region in space where fluid flows rather than trying to follow a unique mass of fluid over time. This method facilitates the study of fluid flow effects in a device or structure. For example, a control volume can be defined as a volume surrounding a device and moving with it. For a specified control volume CV in Figure 2.4 and related control surface CS , the rate of change of any developed specification NF (Depending on the mass of the system) is based on the following equation [43]:

$$\left. \frac{dNF}{dt} \right|_{system} = \frac{\partial}{\partial t} \int_{CV} \eta \rho d\forall + \int_{CS} \eta \rho \vec{V} \cdot \vec{dA} \quad (2.3)$$

where t denotes the time, $NF = \eta$ for the unit mass, ρ is the fluid density, $d\forall$ is a volumetric element within CV , $\vec{V}$ denotes the velocity vector, and $\vec{dA}$ is the surface vector, which direction is perpendicular to the outside of the control surface. The left side of the equation represents the change rate of NF over time in the control. Figure 2.5 shown one-dimensional control volume [43].

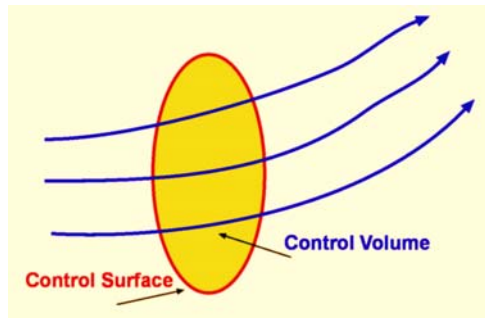


Fig. 2.4: Control volume [41].

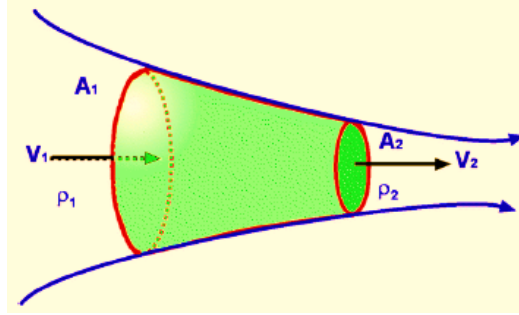


Fig. 2.5: Control volume for one-dimensional flow [43].

2.3.2.2 Conservation Equation (Conservation of Mass)

The Conservation Equation, or conservation of mass, states that the mass in a closed system remains constant and unchanged (regardless of the processes that are happening within it). Conservation of mass of the Reynolds transport theorem creates a relationship between the rate of system changes, control surface and the volume integrals but the system derivatives are related to basic equations of mechanics. Applying this concept to a control volume CV will result to the following equation [44]:

$$\left. \frac{dm}{dt} \right|_{system} = 0 = \frac{\partial}{\partial t} \int_{CV} \rho d\forall + \int_{CS} \rho \vec{V} \cdot \vec{dA} \quad (2.4)$$

In other words, the specific speed of mass flux in the middle of the control surface equals the speed or the rate of volume change inside the control volume. This equation can be simplified in certain cases. For uncompressed flow, which is a common hypothesis for water, density is not a function of space or time. Equation 2.4 can be then simplified as follows [44], [45]:

$$0 = \frac{\partial \forall}{\partial t} + \int_{CS} \vec{V} \cdot \vec{dA} \quad (2.5)$$

For a non-transformed control volume, $\forall$ is fixed and the mass conservation equation can be set as:

$$0 = \int_{CS} \vec{V} \cdot \vec{dA} \quad (2.6)$$

It should be noted that this equation is used for both stable and unstable flow. Integral quantity in the above equation is commonly called the rate of control volume. Note also that the dot product sign depends on the direction of the velocity vector to the surface vector. So $\vec{V} \cdot \vec{dA}$ is negative when the fluid flows in through the control surface and positive when it flows out. In many applications, the assumption of uniform flow in a section (constant velocity at all levels) is sufficient. For the CV in Figure 2.5, the mass conservation equation can be simplified as follows:

$$v_1 A_1 = v_2 A_2 \quad (2.7)$$

It is assumed that the current decompressed flow and the smooth flow are in A_1 and A_2 , respectively [44], [45].

2.3.2.3 Momentum Equation (Conservation of Momentum)

Using Newton's second law for a water particle, results in the momentum equation. This is a clear illustration of the fact that the sum of all forces affecting a system equals the time rate of linear momentum changes in System $\vec{P}$. The equation of control volume for a slow CV (No acceleration) [44], [45] can be written as follows:

$$\left. \frac{d\vec{P}}{dt} \right|_{system} = \vec{F}_s + \vec{F}_b = \frac{\partial}{\partial t} \int_{CV} \vec{V} \rho d\forall + \int_{CS} \vec{V} \rho \vec{V} \cdot \vec{dA} \quad (2.8)$$

$\vec{F}_s$ includes all forces affecting CS and $\vec{F}_b$ includes all volume forces affecting CV . In other words, the time rate of linear momentum changes equals the time rate of linear

momentum changes of the control volume components plus the specific speed of momentum flow at the control surface [44], [45].

Equation 2.8 is a vector equation that can be parsed into the scalar equation compared to the defined coordinate system. The careful selection of coordinate system and border control volume can simplify the analysis. A simple form of the momentum equation for steady, uncompressed and frictionless flow along a flow line is called Bernoulli equation.

In this case, $\frac{\partial}{\partial t} = 0$ and ρ and $\overline{F_s}$ are only caused by compressive forces.

In addition, if CV is limited to flow lines, the flow is only created in the end sections. According to Figure 2.28, blue arrows are flow lines that define two surfaces of control volume and A_1 and A_2 define the other two surfaces; it is also assumed that a steady stream passes through it. With these assumptions and using the continuity equation, the momentum equation is reduced to the following equation:

$$\frac{p}{\rho} + \frac{v^2}{2} + g.zs = const. \quad (2.9)$$

where p denotes the average pressure, v is the uniform flow velocity, and zs represents the section level. It is worth mentioning that, one can show that the Bernoulli equation is also valid between two points along a stable, incompressible, friction-prone, non-rotational flow. Such analyses can also be applied to the first (conservation of energy) and second laws of thermodynamics to obtain an integral-based equation [44], [45].

2.3.2.4 Drag

Drag is surface force that object moving in the water are faced with. The total drag on an object is caused by compressive forces (form drag) and shear forces (skin friction drag). Form drag force is caused as carrier and usually tall and thin objects have lower form drag. skin-friction drag is created by the surface area of the object that is in contact with the fluid. Therefore, there is an optimum body shape that leads to the least drag.

Figure 2.6, presents the relative contribution of the cast and skin-friction drag when the ratio of length to diameter of the carrier increases. However, deviation from the ideal may be necessary to improve the building or inclusion of internal equipment. Also, discontinuities, control surfaces and other side devices will increase the total drag. Scouring the surface, smoothing the side devices or creating a smooth transition to the main object minimizes their participation in the total drag [46], [47].

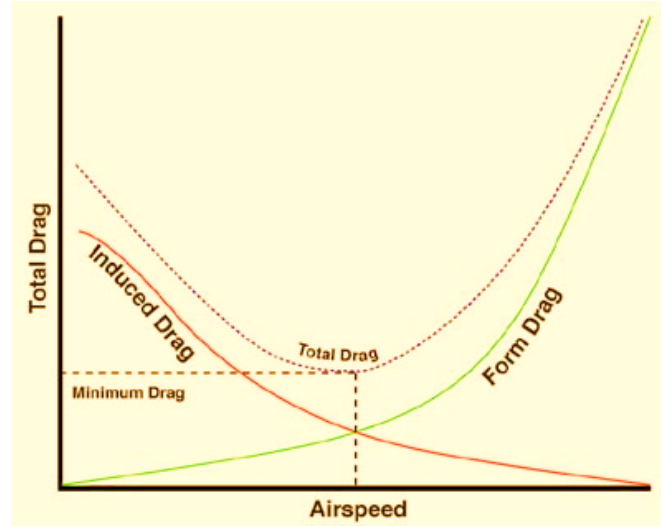


Fig. 2.6: Relative contributions to total drag as a function of L/d [46].

Shell friction drag is calculated as follows:

$$D_{SF} = \frac{1}{2} \rho C_D A_s v^2 \quad (2.10)$$

Here C_D is the carriers drag coefficient that can be calculated by a measured drag force. The surface area A_s is the total surface area exposed to the flow and v is the velocity [46], [47].

2.4 Thrust

Each submarine has a mechanism to move in water. This mechanism creates thrust by moving in water at a certain speed.

Thrust follows the below equation [41]:

$$T = \dot{m}(v_e - v_i) \quad (2.11)$$

Here T is the created Thrust, $\dot{m}$ the fluid velocity inside the thrust device, v_e the output speed, and v_i the input speed. In selecting the device, the dimensions of size, cost, required power, thrust device efficiency and the produced thrust should be considered. The thrust devices used in submarine that may be useful for vehicle design are discussed in [41].

2.4.1 Propellers

Propellers are connected to a shaft or cylinder, which is connected to an engine. Propeller blades can be designed in many ways, but the propellers produce thrust by pushing water through the blades and create vortices behind them. Propellers' loading equation is effective in determining the required force [48], [49]:

$$Loading = \left(\frac{TC}{AJ^4} \right)^{1/4} \quad (2.11)$$

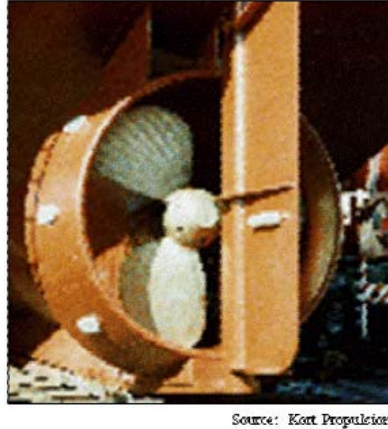
where the thrust coefficient TC is defined as:

$$TC = \frac{T}{\rho(v_p)^2(D'')^4} \quad (2.12)$$

Here ρ is the water density, v_p the propeller speed, and D'' the propeller diameter. The Advance coefficient AJ can be stated as [48], [49]:

$$AJ = \frac{v_{pw}}{v_p D''} \quad (2.13)$$

with V_{pw} the advancing velocity of the propeller in water. Figure 2.7 presents an example of a propeller with a shroud surrounding it to minimize energy loss and thrust by propeller flow towards the optimal thrust. For AUV, shrouds are needed for safety reasons.



Source: Kort Propulsion

Fig. 2.7: Propeller example [41].

Their types are as follows [41]:

- Fixed: The propeller blades are not adjustable.
- Variable Pitch: the hub blades are rotatable so that a different pitch can be allowed that alters the angle of the blades hit.
- Contra Rotating: In this scheme, there are two sets of blades on an axis. The blades rotate in opposite directions so that the propeller behind the carrier could compensate part of the rotational energy that has reached the water by the front propeller and make it more efficient than the first mode. two sets of blades allow smaller propeller diameter.

2.4.2 Podded propulsion devices

Podded propulsion device uses a propeller to provide thrust but the propeller is behind a pod that allows the uniform fluid in that volume. A pod (Figure 2.8) is a polished cylindrical chamber that contains engines for the propeller behind it. This extra space

allows more flexibility in layout and more space for other machines in the carrier wall. The pod can be attached to the wall or main body of the carrier with a blade or fixed or rotational bar. Few examples of the pods can be seen below (Figure 2.8).



Fig. 2.8: Podded propulsors [41].

(a) Podded propulsor on a ship (b) Previous AUVSI competition thruster

2.5 Energy

Submarine carriers fed by batteries have limited supply, which means that the processor has a limited amount of power available in a limited time. The amount of energy required to move the device can be estimated in terms of weight and defined speed due to the mission the carrier is designed for. It can be defined as [50], [51].

$$Pe = F * v \quad (2.14)$$

Here Pe is the supply energy, F the force and v the carrier velocity. Therefore, as initial approximation, the thrust energy required can be obtained based on carrier drag and vehicle velocity. According to carrier hydrodynamics elasticity is proportional to the square of carrier velocity. As a result, the required power is equal to [41], [50], [51]:

$$Pe = D * v = C_D * A * \frac{1}{2} \rho v^2 * v \quad (2.15)$$

Here C_D is the carrier's drag coefficient defined earlier in the hydrodynamics. In addition, the amount of the applied energy is based on the effective force on the carrier multiplied by the distance or altitude in which the carrier moves. Therefore, the amount of energy that should be saved in the carrier can be estimated as follows [41], [52]:

$$ET = F * d = D * d = C_D * A * \frac{1}{2} \rho v^2 * RV \quad (2.16)$$

with E the available energy. The distance d can be assumed the same as R , which is the altitude range of the carrier. To estimate the altitude attainable for a carrier with a specific battery, the above equation can be rearranged so that RV is determined as [41], [52].

$$RV = \frac{ET}{C_D * A * \frac{1}{2} \rho v^2} \quad (2.17)$$

Other processor sensors can use the energy from the battery pack. In order to consider the effect of the energy used in the processor in the obtainable altitude range, the altitude range equation can be modified to consider the "Hotel electric power" or the required supply energy to maintain carrier processor functions. Based on the above definition of supply energy, the hotel power can be estimated as [41], [52]:

$$FH = \frac{PH}{v} \quad (2.18)$$

Thus using the energy equation, the total energy needed to move within the range RV is:

$$ET = (D + FH) * RV \quad (2.19)$$

This is achieved by arranging the altitude range as [41], [52]:

$$RV = \frac{ET}{D + FH} = \frac{ET}{C_D * A * \frac{1}{2} \rho v^2 + \frac{PH}{v}} \quad (2.20)$$

So, when choosing the batteries for an AUV, the task should be to best determine the required supply energy [41], [53]. Also, the system can be divided into two sub-systems: a first one may be used to supply the propulsion systems and the second to provide power supply to sensors, computers and other related devices [54].

2.6 Navigation

Typically moving between two points (apart from the problem of control) can be considered as a navigation problem. Today, many AUVs continuously work with surface ships for navigational purposes. However, ultra-low-power and long-range AUVs, such as underwater gliders, can work unattended for a certain period by periodically relaying data to the command center before recollecting more data. When a reference surface such as a support ship is available, ultra-short baseline (USBL) positioning is used to calculate the location of the sub-sea vehicle relatively to the known position of the surface craft by means of acoustic range and bearing measurements. Also, for precise maneuvering, an inertial navigation system installed on AUV can measure the vehicle acceleration and thus its velocity. These observations are then treated to determine a final navigation solution.

Positioning technologies can be broadly divided into two major streams: relative positioning and absolute positioning. Absolute positioning means that the calculated current position does not depend on previous positions. An example of this absolute positioning system is the global positioning system (GPS). The advantage of this system is that there is no accumulation of position deviation error. However, the GPS signal cannot be used indoors and its production rate is relatively low. Therefore, we used the relative positioning system dead reckoning method to find the vehicle location [55]-[57].

2.6.1 Path Planning

Guidance and control functions are related to higher levels of software intelligence through route planning. A route planner can obtain information from the obstacles outlined in the map as well as friendly or hostile environments, and produces a paved way for the vehicle to follow.

In this operation, when the vehicle encounters with unknown objects or when the mission requirements changes, there is a certain level of feedback using sonar in order to reschedule the route. Based on the position and orientation of the vehicle at a certain point, it is possible to obtain multiple classes of paved paths containing a set of pieces of straight lines and circular arcs or 3rd grade curves. Separation of the control functions between guidance and control are based on the concept that an autopilot is responsible for the dynamic stabilization of the vehicle in terms of speed, direction and depth. It combines the rule of motion guidance control commands for the position and direction as well as other requirements.

This separation of guidance functions and motion monitor has its own issues. However, to keep accurate track of the dynamic guidance law should be as quick as possible. This provides a lower bound for the motion controller reaction. Ocean vehicles suffer from a number of dynamic delays in motion response and actuator value revision, and these delays determine an upper bound for their reaction time [57], [58].

2.6.2 Radio navigation

Radio navigation is the use of radio frequencies to determine a location. Similar to radiolocation radio navigation, is a radio determining method. The first radio navigation system was the Radio Direction Finder (RDF) by tuning in a radio station and then using a directional antenna to find the direction of the broadcaster antenna source. It has been later replaced by more advanced systems. LORAN (Long Range Navigation) System is a ground-based navigation system using low-frequency radio transmitters that use the time interval between radio signals received from two or more stations to determine the position of a ship or aircraft. The current version of LORAN used by the public is the LORAN-C, which operates in a low frequency band of 90 to 110 kHz. Navigation method provided by LORAN is based on the time difference between receiving a signal from a radio transmitter pair. The time difference between signals from two stations can be presented by a hyperbolic line of information (LOP). If the position of two synchronized stations is determined, the receiver position can be determined as a place on the hyperbolic curve where the time difference between signals is fixed [55], [59], [60].

2.6.3 Navigation and Positioning Error

Navigation precision and positioning required by an AUV will be determined by the requirements of the mission. Positional accuracy is any error caused by AUV determining the geographic location. Navigation accuracy is the precision by which the AUV can move from one geographical point to another.

Basic navigation error model uses the following terms for AUV navigation [55]:

$$\text{Total Error } (1\sigma) = \sqrt{EV^2 + (d \sin IE)^2 + (d \sin DS)^2 + (d SW)^2 + (d SA)^2} \quad (2.21)$$

where EV is the error of vehicle location, SA the speed error along the path, IE the initial error of the direction of movement, SW the speed error within the path, and DS the deviation of the sensor. These terms are combined to estimate the vehicle location error as function of the traveled distance d . They are independent of the vehicle speed and do not include higher-order terms such as directional speed deviation. It should be noted that in the worst-case scenario it is assumed that the vehicle has a constant movement direction during the mission. In many operations, the direction of the vehicle could change or be reversed. In these cases, other error models are used to remove navigation errors [55], [61].

2.6.4 Inertial Measurement System

One of the conventional relative positioning systems is the inertial navigation system (INS). Positioning dead reckoning by gyroscopes and accelerometers is called inertial navigation. Gyroscope measures the angular velocity and accelerometer measures acceleration. Integration of angular velocity versus time produces the angular data. Distance data can be obtained by double integration of the acceleration versus time. An inertial navigation system (INS) is a complete unit (without the need for auxiliary devices) and does not require any external electromagnetic signals. So, INS does not have the problem of signal coverage like in GPS. In addition, the data produced in INS can be much faster than GPS. However, the INS disadvantage is its position deviation bias. The errors are accumulated

and can worsen over time due to integration. Methods such as Kalman filtering are used to reduce errors caused by random bias deviation [62].

2.6.5 Dead Reckoning

Dead reckoning estimates the global position of a vehicle by moving forward from a known position using the route, speed, time and distance that must be traveled. With this method, it is possible to determine the current position as well as at a certain time if we keep the speed and direction.

Dead reckoning is a navigation method used in ships, aircraft, trucks, cars, construction sites and, more recently, in moving AUVs. Basically, this method is used to estimate the position of the object based on the distance it has travelled from its previous position.

A navigation system that uses this method uses the last known position (constant) of the vehicle then plots the expected position of vehicles for a certain constant distance (elapsed time from a constant to the next constant) based on the compass course it is steering, the speed, and allowance for currents [63].

2.7 Control Systems

2.7.1 Sliding mode control

Sliding mode control is a non-linear control method that guarantees control strategy against uncertainties. In this method the system stability is obtained by keeping system states on the sliding surface. Sliding mode controller is very important to design robust controllers for linear and non-linear systems. Their advantage is the ease and flexibility in design that guarantees stability by reducing system degree and robustness against disturbances and uncertainties. In addition, low cost implementation of the controller and its applicability in discontinuous systems has made it appealing to the control engineers and there is an increasing tendency to use it.

Initial experiments indicated that the Super Twisting Control Algorithm can track well but the position it cannot adequately overcome the AUV inertia. The problem has been resolved by considering the constraint on the AUV acceleration in a Real Twisting Control Algorithm. This helped overcoming the inertia by providing more control energy.

In the modified classical sliding mode, a second degree sliding mode was considered.

$$\dot{s}(t) + \beta s(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \dot{e}(t) \quad (2.22)$$

where k_p , k_i and k_d are positive constants, β is the death rate of the surface, and e is the state or tracking error. In this mode, in addition to satisfy the zero level condition ($S=0$), the first derivative of the surface should be zero ($\dot{s} = 0$). Also, by adding the integral term to the sliding surface and increasing the surface degree, chattering phenomenon was reduced [64].

Another type of modification of sliding mode control is the surface fuzzification technique in which the designed fuzzy system parameters are based on the fuzzy decisions the signal produced. This method, considered as a classic method, has better speed compared to other sliding mode controllers [65].

2.7.2 PID controller

PID controllers were used in the shipping industry in the early twentieth century. Today, this type of controllers is applied in many industrial machinery and applications. PID is formed of three parts called Proportional, Integral and Derivative that each one obtains the error signal as the input and perform actions on it and finally their output is totaled. The output of this set is the same as PID controller output sent to the system to correct the error. PID controller is among the most common Feedback control algorithm that is applied in many control processes such as DC motor speed control, pressure control, temperature control and so on. PID controller estimates the error between the process output and input value (set-point). The controller purpose is to minimize the error by adjusting the process control inputs.

Standard PID formula is as follows:

$$Output(t) = K_p \left(e(t) + \frac{I}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de}{dt} \right) \quad (2.23)$$

Thus the conversion function of a PID controller is as follows:

$$G_c = K_p + \frac{K_i}{s} + K_d s \quad (2.24)$$

The effect of PID controller in terms of coefficients is defined as follows: the coefficient K_p increases the system velocity and reduces the steady state error to some extent (but not up to zero). Adding the integral term K_i makes the permanent steady state error zero but adds a great part of overshoots to the transient response zero. The derivative term K_d makes the transient fluctuations zero and makes the step response close to the ideal point [66]-[68]. (Table 2.1).

Note that because of the low convergence speed of PID controllers in complex systems, they are often combined with other advanced controllers.

Table 2.1: PID controller features.

Type	Advantages	Disadvantages
P	Quick, easy, inexpensive	The output cannot reach the final desired amount and the decisions are based on momentum error
I	The output can reach the final desired amount, it is not sensitive to noise and acts based on the error history	It is too slow and reduces system stability, it is not simple and inexpensive
D	It is very quick and stabilizes the system	The output cannot reach the final desired amount, it is not simple and it is sensitive to noise

For a faster response and to reduce steady-state error, the controller coefficients must be set. To this aim, there are a number of techniques that mainly involve numerical optimization algorithms such as genetic algorithms, swarm, neural networks ... [69]-[72]. In these methods, a fitness function is defined to measure the appropriate performance of the response against PID parameter variations.

2.7.3 Fuzzy controller

A fuzzy controller is composed of four main parts: Fuzzifier, database, decision making block and defuzzifier (Figure 2.9). Usually before and after the fuzzy controller, a pre-processor and a post-processor are used.

The input is first presented to the fuzzifier, which allocates a value between 0 and 1 to the input based on the defined membership functions. The value is determined according to the linguistic rules that get the “if – then” form also called the fire strength. These values are then combined in the inference engine to give a number as the output for the fuzzy system. In the end the obtained value is defuzzified and presented to the next part as a meaningful output.

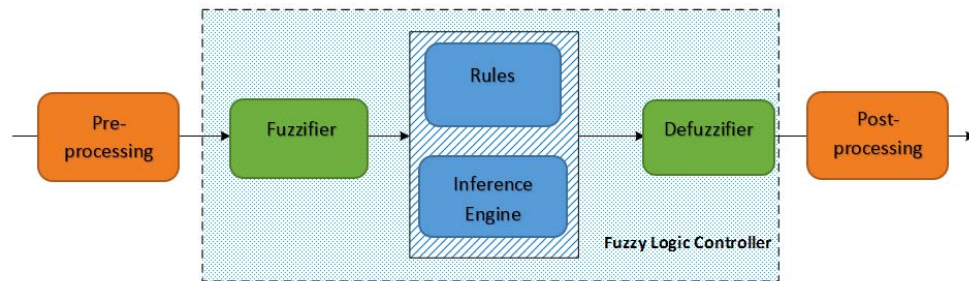


Fig. 2.9: The general block diagram of the fuzzy controller structure diagram.

The first blocks inside the controller is the fuzzifier in which the degree of input values's membership degree is calculated. There are several methods to do so. The first method is to

fuzzify the single value variable. In this mode the variable's membership function is "1" at one point and "0" in the rest of points.

Each fuzzy controller is formed of a number of "if -then" as the database. These rules are presented in different forms. Suppose a fuzzy controller has two input errors and an output signal as the output:

- NEGATIVE, ZERO and POSITIVE could be associated with the fuzzy sets for both inputs and
- NB, NS, PS, and PB (stating for large negative, small negative, positive, small positive, and large positive, respectively) could be the output fuzzy sets.

The fuzzy set elements are taken from a set called the global set. The global set includes all members that can be considered for the system. Before designing the membership function, it is necessary to consider a global set for the input and the output. This is in fact finding the maximum and minimum for the inputs and outputs so that all measured input and output values are understood by the fuzzy controller. Another point is that the input membership functions can be discrete or continuous. Continuous membership functions on a global set are continuous and discrete membership functions are discrete on a global set.

Each global set member is the member of the membership function with one degree of membership that the preprocessing does this work. Each member is the member of the membership function with one degree.

The inference engine used in the fuzzy system includes a Multiplication inference engine and a Minimum inference engine. Then, the inference result fuzzy set should be converted to a non-fuzzy number so that the proportional output is applied on the process as the control signal. This practice is called defuzzification.

Fuzzy controllers are, after the PID ones, the most widely used as industrial controllers, mainly because of their low cost. Also there is no need for a fuzzy controller to precisely identify inputs because there is an uncertainty in the nature of the fuzzy system.

Applied to AUVs, a zero order Takagi – Sugeno fuzzy controller has been used to control deviation and velocity [73]. In this controller, the output is expressed as a number and presented to the actuators. The results presented in [73] claim that the use of fuzzy controller is much faster and has less error than PID controller while the controller is more robust. However, only two degrees of freedom are controlled. In [74], the membership parameters have been improved using a faster tilt.

2.8 Conclusion

In this chapter, we reviewed some existing underwater AUVs, highlighting their advantages and disadvantages. Then, we reviewed the basics of fluid dynamics and their rules in designing the AUV body and its propellers. We also discussed about common types of navigation systems for AUVs and, finally, we presented different types of controllers highlighting their advantages and limitations. Among all controllers, fuzzy controllers, despite their simple design, exhibit a high efficiency in implementation and present a good flexibility in the face of uncertainty and noise.

The next chapters will be devoted to the design of the different mechanical and electrical parts of the proposed AUV.

CHAPTER 3

MECHANICAL DESIGN

3.1 Mechanical Design

While designing AUVs, finding the most suitable component is one of the key factors that strongly affect the device performance. In fact, such design is deeply based on a strong interdependence between all constituting parts, which can be subdivided into three main parts namely, Mechanical, Hardware/Electronic, and Software. The mechanical section can be subdivided in turn into three main tasks (Figure 3.1).

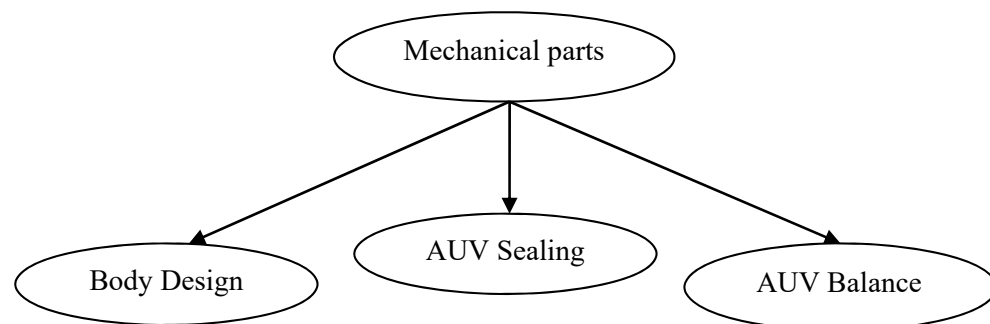


Fig. 3.1: AUV: block diagram of the mechanical parts.

3.1.1 Body Design

To successfully design the body, one should establish a primary plan of the body structure with the location of each sub-part, knowing that it will be made of one piece with cameras on the top and bottom, and engines attached on the sides. In this work (Figures 3.2 and 3.3), the structure has been inspired by the V-22 planes thrusters (Figure 3.4) [75] while the body shape is inspired by the whale's body.

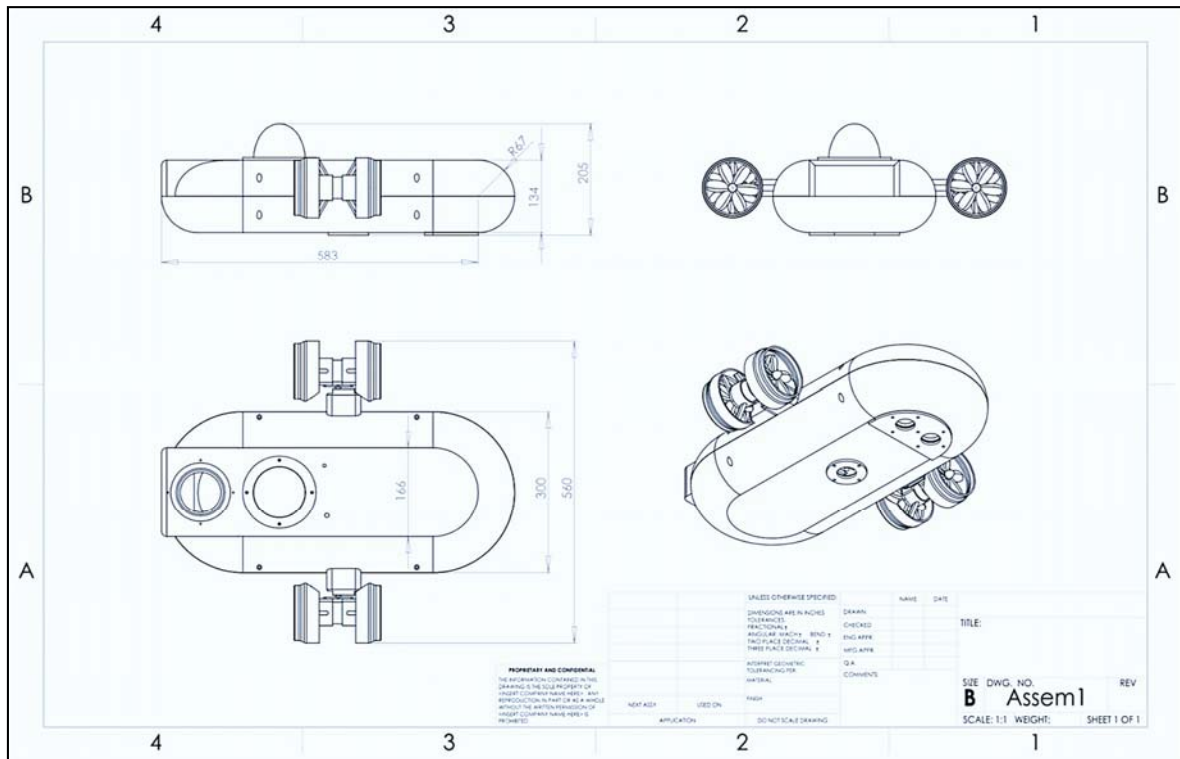


Fig. 3.2: Isometric scheme of the designed AUV.

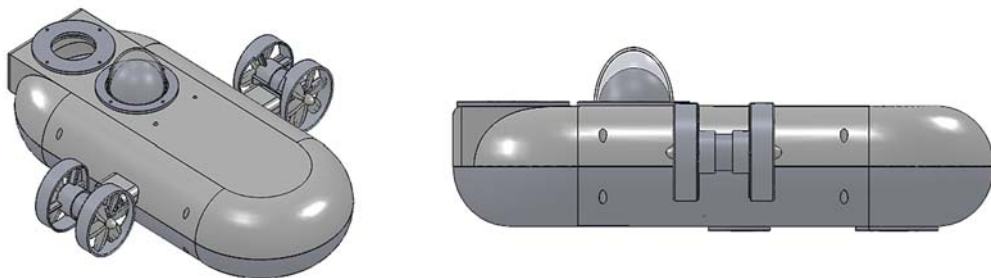


Fig. 3.3: Different views of the designed AUV.

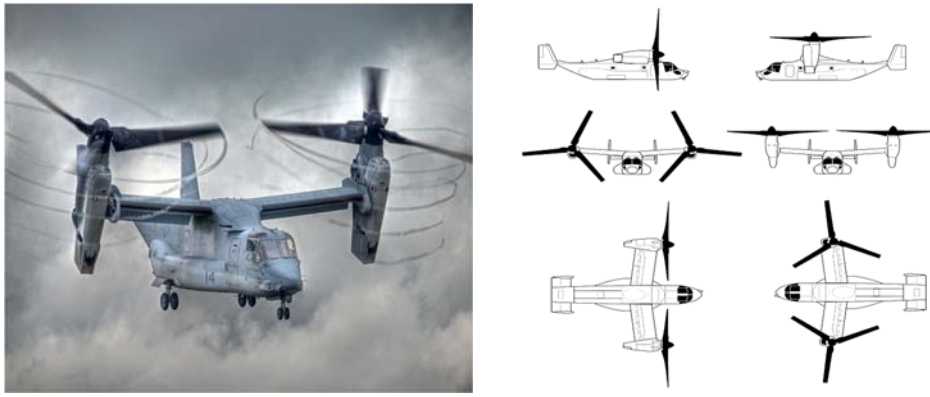


Fig. 3.4: Tilt rotor V-22 [76].

As previously mentioned, this form is exclusive and the whole set will become naturally balanced with the Mass Shifter (MS) unit beside it.

Most existing devices use both vertical and horizontal fixed thrusters. However, in this case, all thrusters should be put “on” at the same time to create a simultaneous vertical-horizontal movement; it will then lead to high power consumption. Our first contribution was to use a pair of mobile thrusters instead of fixed vertical/horizontal ones, thus saving energy. In practice, the two thrusters shown in Figure 3.3 could be oriented within a specific angle based on the vertical and horizontal forces needed to move the device in a predefined direction. Any change of angle should be made possible by the instant movement of a servo motor which consumes much less energy than the constant movement of a thruster.

As seen in Figure 3.3, the curving all around the body minimizes the drag force, which will be explained in the next parts. The valve #1 in Figure 3.5 is used for placing charger cables, microcontrollers and processor data cables. This valve enables the user to have access to the whole system without any need to open the whole body and follow-up damage to the sealing. The holes #2 and #3 in Figure 3.5 are for the pressure sensor and the antenna, respectively. It is to be noted that the form of body at the end, which is the access valve, is cubic to prepare more space for the valve.

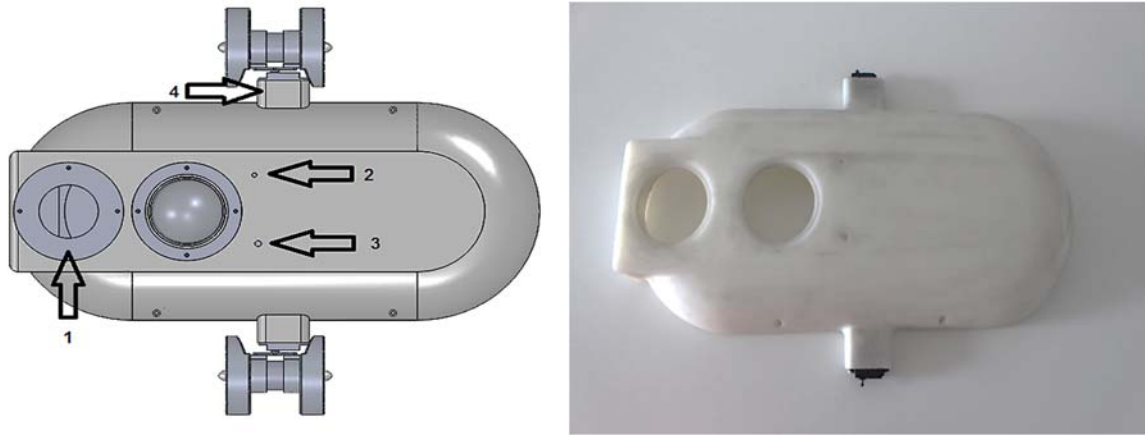


Fig. 3.5: Access valve (1), Antenna holes (2), Pressure sensor (3), and Dolly block of servo motor (4).

The body is designed with the two dolly blocks #4 shown in Figure 3.5, in which servo motors will be placed to change the thruster angle.

The thickness of the body is 1 cm and the body is made of Teflon. There are two cameras, one above (main camera) and the other on the bottom. Considering the thickness of the body, some extra tools may be attached to the body upon necessity. Therefore, adding more thrusters in the required directions or a AUV arm would be possible.

3.1.2 Locating Parts

As mentioned, the different components of the AUV have been located in a way such that minimum noise is directed toward sensitive parts. In this AUV, six valves have been designed to allow access to the different parts. Such valves are isolated from the rest of the body by walls to avoid any damage due to possible leakage (Figure 3.6).

In Figure 3.6, valve #1 is a valve under the access valve, separated from the other sections by a wall to avoid possible leakage. Also, this space contains a step at the end to easily design the required board or sockets without any fear from making a hole in the body.

Also, the batteries are in this space. The engine driver board is placed in the space provided by valve #2. Valve #3 is dedicated to locate the Mini PC and valve #4 to initially locate the sensors. However, due to their high sensitivity, they have been relocated as in Figure 3.7: the circular space near the cape has been used for placing the Inertial Measurement Unit (IMU) and the compass sensors, as in Figure 3.8. In this Figure, #1 is related to the bottom camera. #2 and #3 are for the sensors. Also, these parts have holes at the bottom covered by clear and thinner-than-body lens. The reason for this is to weaken the earth's electromagnetic waves inside the body.

Valve #5 is for the parts that are connected to the middle sheet: the mass shifter and its motor. Finally, valve #6 is for the main camera space as well as for the sensor board. Figure 3.9 shows the placement of the driver board, the bottom camera, the sensor board, the mass shifter, and the main camera.

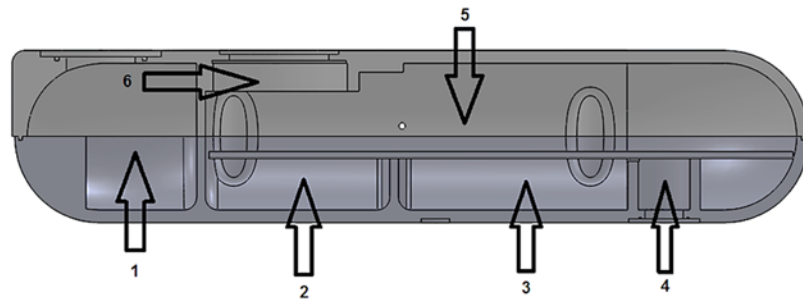


Fig. 3.6: Six designed valves.

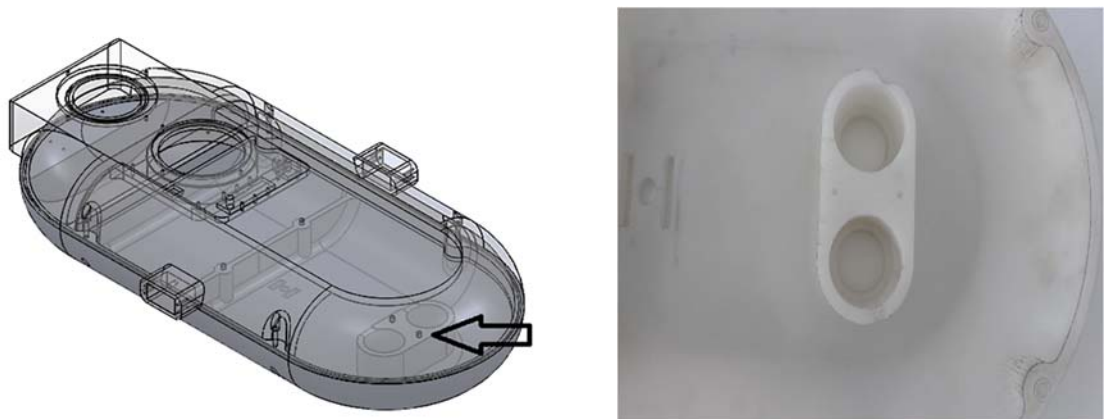


Fig. 3.7: Location of the sensors.

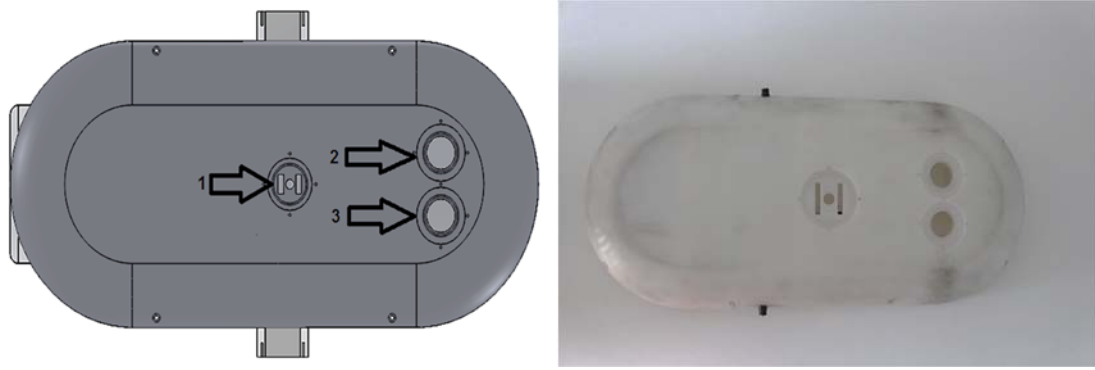


Fig. 3.8: The Bottom openings: (1) Bottom camera, (2) and (3) Sensors.

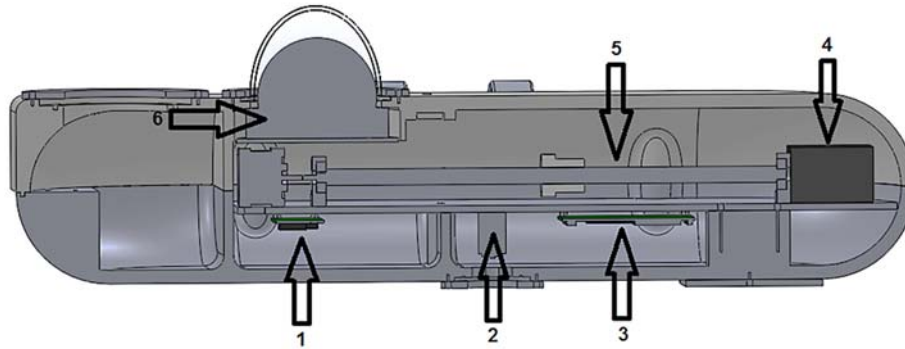


Fig. 3.9: Locating internal components.

3.2 Body Material and Body Strength

The body and its components are made of PTFE (Polytetrafluoroethylene) Teflon [77]. Teflon is a polymer with a molecule made of carbon fluorine atoms strongly connected to each other. This material is very strong and highly recommended due to its numerous plastic features. It is widely used in industry and has practical suitable features such as high resistance against chemical factors, high resistance against adherence, and high dielectric features. However, one of its main advantages remains its flexibility. This material does not break even at a pressure of 0.7 N/mm^2 (700 kPa). However, in accordance with the ASTM D790 standard, plastic form and flexibility of Teflon in natural temperature is about $50\text{-}650 \text{ N/mm}^2$ [78], [79]. Teflon is also one of the best and valuable electrical insulators. It should be also mentioned that Teflon retains this feature in different atmospheric, and temperature conditions as well as at different frequencies [78], [79].

3.2.1 Applied Pressures

In this part, the underwater vehicle is analyzed in terms of pressure tolerance. The maximum depth for the vehicle was set to 8 m. Considering the amount of 1,000 kg/m³ for water density, for each 1 m increase in depth, we have 10-kPa pressure applied on the outer surface of the body. The total pressure P_t exerted on the body can be set as the difference between the pressure on the outer surface of the body, P_{out} , and the pressure on the inner surface of the body, P_{in} [80], [81] as:

$$P_t = P_{out} - P_{in} \quad (3.1)$$

The outer pressure can be expressed as:

$$P_{out} = P_{atm} + P_{water} \quad (3.2)$$

with P_{atm} the atmospheric pressure (100 kPa) and P_{water} the pressure of the water column. Given that the maximum working depth of the vehicle is 8 m, we get for an acceleration of gravity g of 9.81 m/s²:

$$P_{water} = 1000 * g * 8 \cong 80 \text{ kPa} \rightarrow P_{out} = 180 \text{ kPa} \quad (3.3)$$

Assuming that the internal pressure of the vehicle is the same as the atmospheric pressure of 100 kPa, the total pressure exerted on the body is:

$$P_t = 100 + 80 - 100 = 80 \text{ kPa} \quad (3.4)$$

3.2.2 Reliability Factor and Allowable Load

In order to ensure the structure's safety, it is required to select the allowable stress tolerated by the structure that gives the appropriate load. The load based on which the structure is designed may be different from the load actually applied to the structure. For example, measurements performed on a structure may be inaccurate due to errors in connecting the

parts or to unknown vibrations, shocks or loads that were not considered during the design. The air corrosion, degradation and weather conditions can also alter materials conditions during their usage.

One way to determine the allowable load to design and analyze a structure is the use of a value called safety factor. The called safety $F.S$ can be obtained by [82]:

$$F.S. = \frac{\sigma_{ultimate}}{\sigma_{working}} \quad (3.5)$$

where $\sigma_{ultimate}$ is the ultimate stress load determined by the material test (can be found in tables related to the mechanical properties of the materials) and $\sigma_{working}$ the working stress load based on the design.

The failure load for PTFE is set as 23 to 30 MPa according to preliminary tests performed on the material. Therefore, according to the desired maximum depth (8 m), the lowest failure load value of 23 MPa was considered as well as a safety factor of 3.5 [83], [84].

Also, according to the failure load related to PTFE material, the maximum load to be applied on the body is 6.57 MPa. Therefore, $\sigma_{working} \cong 6.57 \text{ MPa}$. The properties required by PTFE material as well as the selected safety factor and allowable load by the structure are summarized in Table 3.1.

Table 3.1 PTFE Mechanical properties and selected safety factor.

$\sigma_{ultimate}$ - ultimate stress load of PTFE	23 MPa
ME - the modulus of elasticity for PTFE	0.5 GPa
ρ - PTFE density	2300 kg/m ²
$F.S$ safety factor	3.5
$\sigma_{working}$ working stress load	6.57 MPa

3.2.3 Anchor

Figure 3.10 shows the location of the constraints while Figure 3.11-a represent the loading based on pressure and applied on the entire surface of the body (red arrows). Figure 3.11-b shows the used meshing. The green arrows on both sides of the vehicle and in the motors' arms are the places for the body anchor.

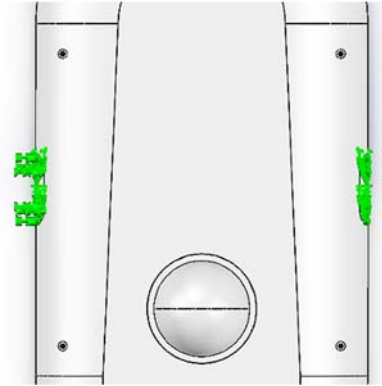


Fig. 3.10: Location of the anchors.

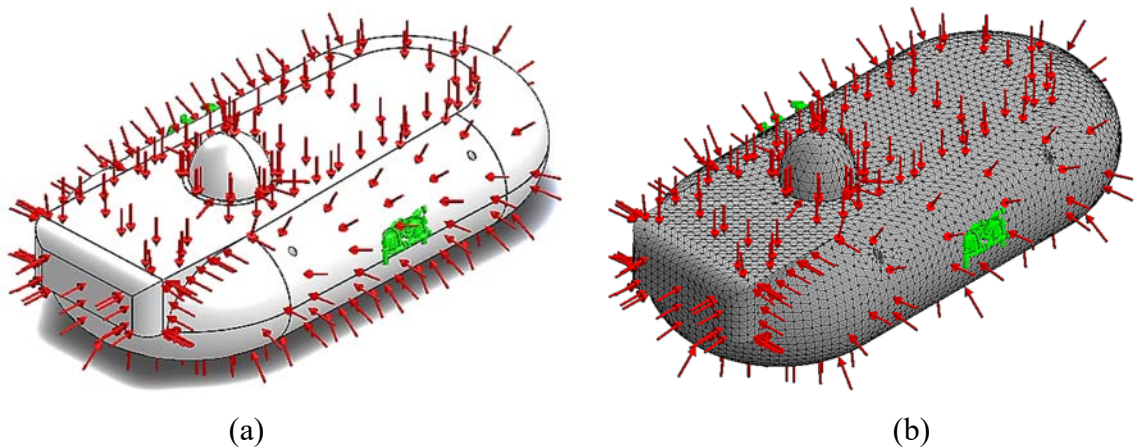


Fig. 3.11: (a) Loading, (b) Used mesh.

3.2.4 Simulation Results

The force on the surface is usually called stress. There are two types of stress: Normal and shear.

Normal stress is the energy intensity applied perpendicularly on a unit area [85]:

$$\sigma = \frac{F}{A} \quad (3.6)$$

where F is the force and A is the area. If the direction of the force is toward the inside, it is called compressive stress and noted σ_c . Otherwise, it is called tensile stress and noted σ_t . Energy intensity per unit area is tangent to the surface and called shear stress. For a shear force VS , the shear stress τ is given as [85]-[87]:

$$\tau = \frac{VS}{A} \quad (3.7)$$

According to Hooke's Law, the stress σ is given by

$$\sigma = E' * \varepsilon \quad (3.8)$$

where E' is a coefficient of proportionality known as the modulus of elasticity and ε the strain (a dimensionless parameter) [85]-[87]. Figure 3.12 shows the strain changes in the designed body. One of the applied failure hypotheses is the strain or distortion energy hypothesis, which predicts that failure due to yielding happens when the strain stress in a unit volume is equal to or greater than the strain energy related to yield tensile strength. This suggests that failure occurs when Von Mises stress σ' is greater than the matter yield stress S_y [88], with

$$\sigma' = \frac{1}{\sqrt{2}} \left[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2) \right]^{\frac{1}{2}} \quad (3.9)$$

where σ_x , σ_y , σ_z are stress on x, y, and z axis, respectively. Figure 3.13 shows the Von Mises stress in different parts of the body with a maximum value of 6.228 MPa, occurring in the region encircled in black in Figure 3.14.

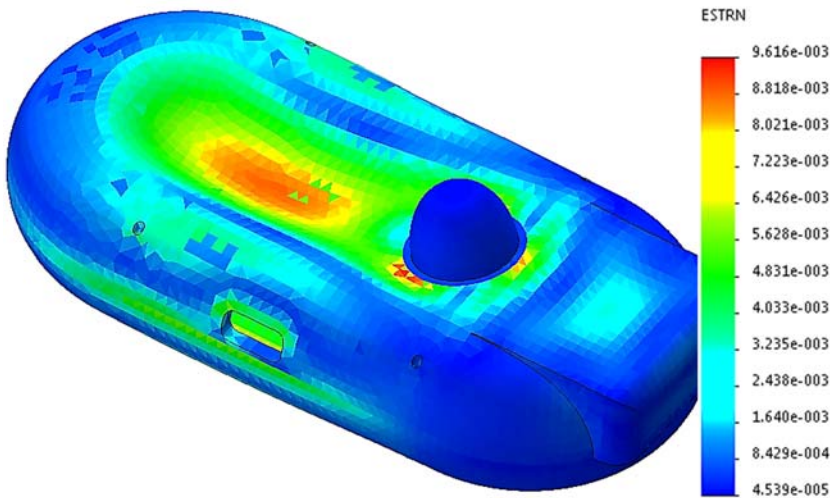


Fig. 3.12: Strain changes.

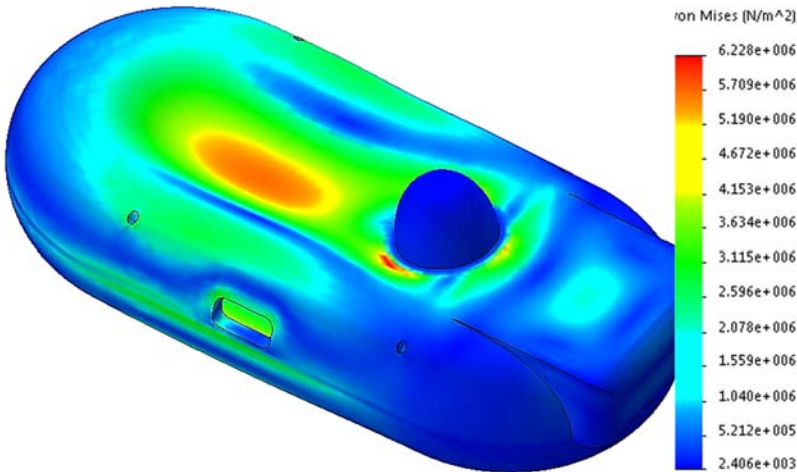


Fig. 3.13: Stress Changes.

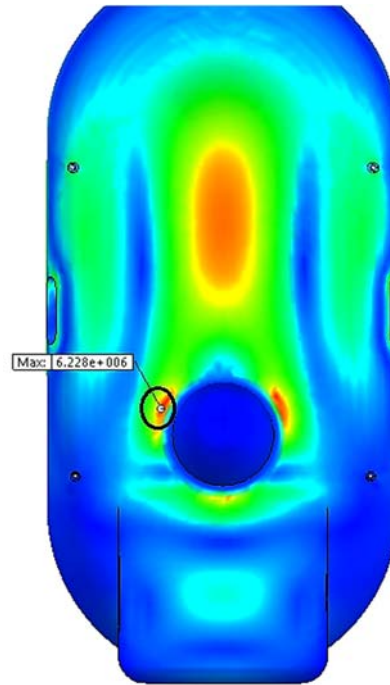


Fig. 3.14: Region of Maximum Stress.

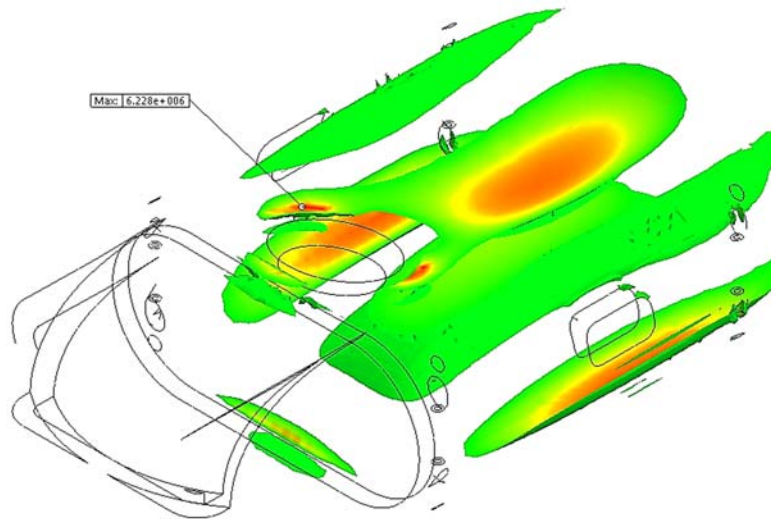


Fig. 3.15: Stress More than 3 MPa.

Figure 3.15 presents the part of the body on which the stress is greater than 3 MPa, i.e., the central parts of the body (up and bottom). Figure 3.16 shows the displacement of various parts of the body with a maximum of 5.205 mm for the point zoomed in Figure 3.17.

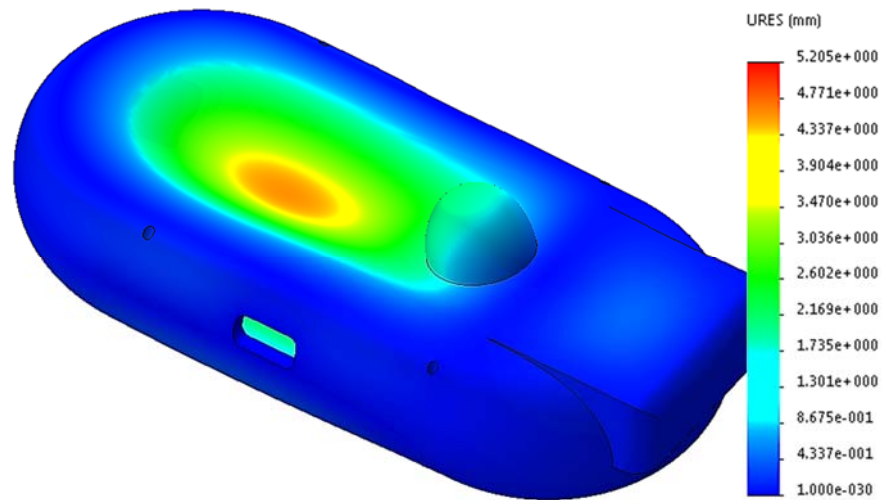


Figure 3.16: Displacement of the Points in the Body.

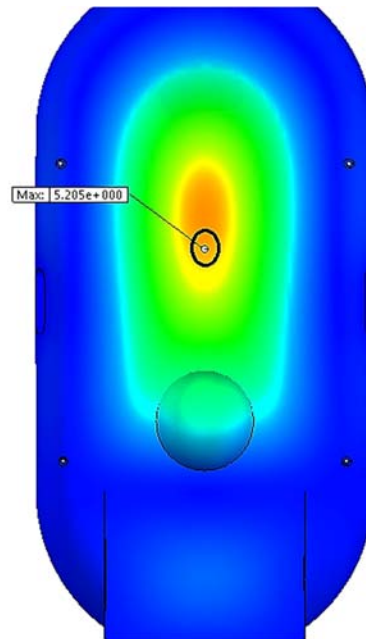


Fig. 3.17: The Point of maximum displacement

3.2.5 Conclusion

As noted, the maximum allowable stress applied on the body is 6.57 MPa while the maximum applied stress on the body at a depth of 8 m is lower (6.228 MPa), thus validating our design.

3.3 Mass Shifter

In designing this AUV, we included a mass shifter first because of the thruster's movement and second to make the maneuver possible in the direction of the pitch. The whole set can have two movement modes. In the first mode, the body will have, with the help of the mass shifter, a constant horizontal movement and its movement towards vertical and horizontal directions should be done through a change in the thruster's angle. In the second mode, the thrusters should be kept fixed in a horizontal direction and the vertical movement should be made possible through a change in the body angle in the pitch direction. The first mode is used to maneuver or pass obstacles while the second mode is used to preserve and store the source of energy and for changes in relatively large depths.

As shown in Figure 3.18, the mass shifter consists of a stepper motor (#1), two ball bearings (#2), a screw (#3), a Nut (#4), and two bearing structures (#5). The controller of this system is a PI controller receiving its feedback from IMU, designed to regulate the pitch angle. Based on initial design estimations, a weight of 3 kg was selected for the change of the mass center of the system to balance the weight and buoyancy force. Note that this weight can be easily changed if additional elements have to be included.

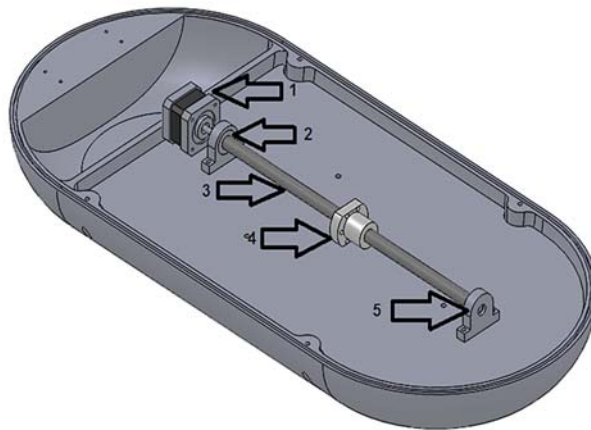


Fig. 3.18: The Mass shifter mechanism.

The significant point is the structure of the ball screw which does not use guide. In fact, in ball screw datasheets, emphasis is put on the weakness of ball screw against radial load, implying that it is better to put the weight on the guide, if any. Based on preliminary calculations, a weight of approximately 12 kg can be placed on the nut without deforming the screw. The specifications of the ball screw are as follows (Figure 3.19):

- Ball Screw: Length: 35 cm Lead: 5 mm Diameter: 12 mm
- Ball Bearing: Deep Grove: Inside Diameter: 8 mm Outside Diameter: 22 mm

Thus, based on the above, we designed two cube dumbbells of length 25 cm at a distance of 13 cm, each weighing about 1,300 g, which can be moved on two rails. The total weight of the AUV with all the equipment is about 21 kg. Figure 3.20 shows the Mass shifter mechanism.

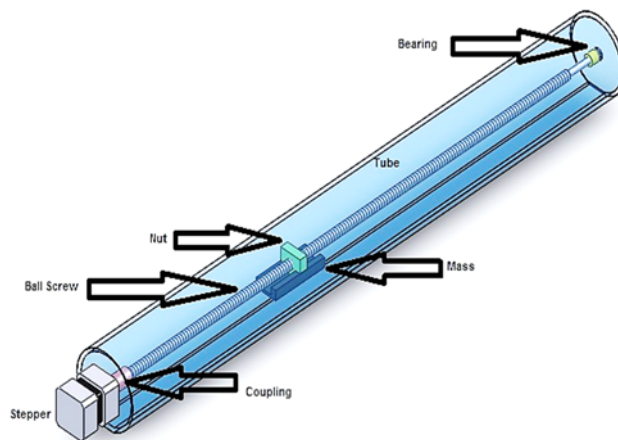


Fig. 3.19: Mass shifter.

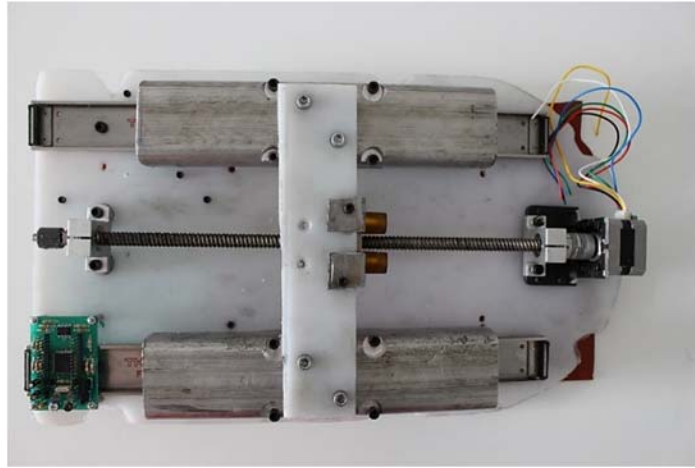


Fig. 3.20: Mass shifter: Location of the cube dumbbells, Ball screw, and Stepper motor.

3.4 Thrusters

In the previous chapter, some practical structures of AUVs have been discussed and the common point among them was the use of vertical and/or horizontal fixed thrusters. Knowing the limitations of the AUVs and trying to make it as small and light as possible, we opted for a pair of mobile thrusters with 360° rotation in order to both make an innovation and save the energy source. While thrusters are fixed, all thrusters should simultaneously be “on” to create a simultaneous vertical-horizontal movement; this will lead to high power consumption while, in the proposed structure, the two thrusters can be oriented in a specific angle so both vertical and horizontal forces needed can be provided simultaneously. This change of angle is made possible by the instant movement of an engine stopper, which consumes much less energy than the constant movement of a thruster, and there are two DC motors inside the thrusters at both sides.

In the designed AUV, four motors are used: two DC motors for moving the AUV, one stepper motor for moving the mass shifter and creating balance, and one motor for the 360° rotation of the top camera.

In this design, the vertical forces are balanced so that the buoyancy force and the weight of system are roughly the same [89]. In order for the system to be inclined to diving, we

conducted some preliminary experiments and found that a force of about 3 kg_F should be necessary to move the system, which then led to the design of appropriate thrusters and corresponding fans.

The required propeller was designed using OpenProp V3.3.4 coding [90]. These systems of codes are written based on Matlab [90], [91] which outputs are in the form of dots. They were then entered using Macro codes in SolidWorks, to finally lead to understandable levels.

The features of design a 6 blades propeller using these codes are:

Rotor speed: 1500 RPM,
Trust: 1 kg_F,
Vehicle speed: 5 cm/s,
Hub diameter: 1.6 Cm,
Required power: 11.6 W,
Required torque: 0.07 Nm.

The sizes of the router and hub were calculated based on the form of the thruster placement beside the body. We should mention that the selected motor has two shafts with an encoder for motor speed control installed on one of them, and then, the propellers have to be placed on the other shaft. This will make the motor revolution using less power than indicated in the motor datasheet. Figures 3.21 and 3.22 show the process for designing fans in OpenProp and SolidWorks, respectively. Then the fan was made of ABS using a 3D printer [92]. After performing tests, it was found out that each motor produces a thrust force equal to 1.1 to 1.5 kg_F per propeller, which exceed the minimum specifications. It is expected that this amount reaches to 2kg_F upon placing the second propeller on the motor, practically reaching to 4 kg_F for the whole system.

For designing the thruster, housing made of Aluminum was used to play the role of thruster connector to the Servo (Figures 3.23 and 3.24). Figures 3.25 and 3.26 show the design of the thruster. As shown in these Figures, part of the motor body will be in contact with water, which will assure that the temperature of the engine will not increase too high.

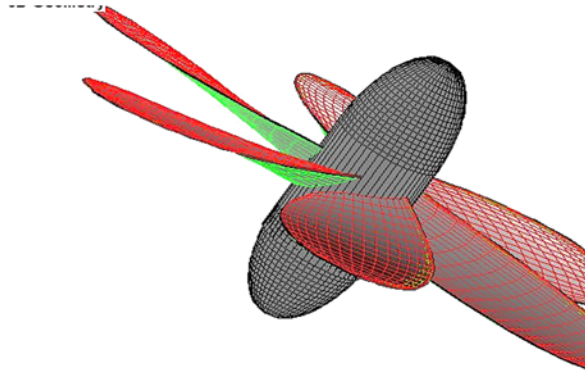


Fig. 3.21: Fan design in OpenProp.

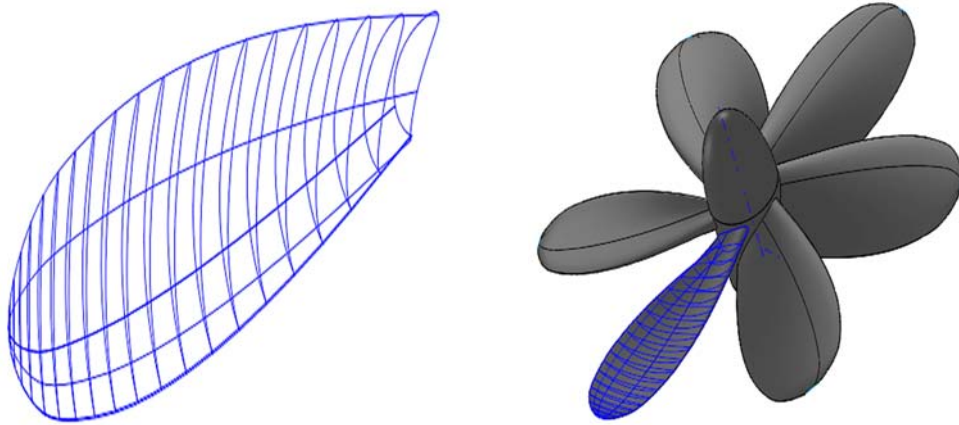


Fig. 3.22: Propellers designed in SolidWorks.

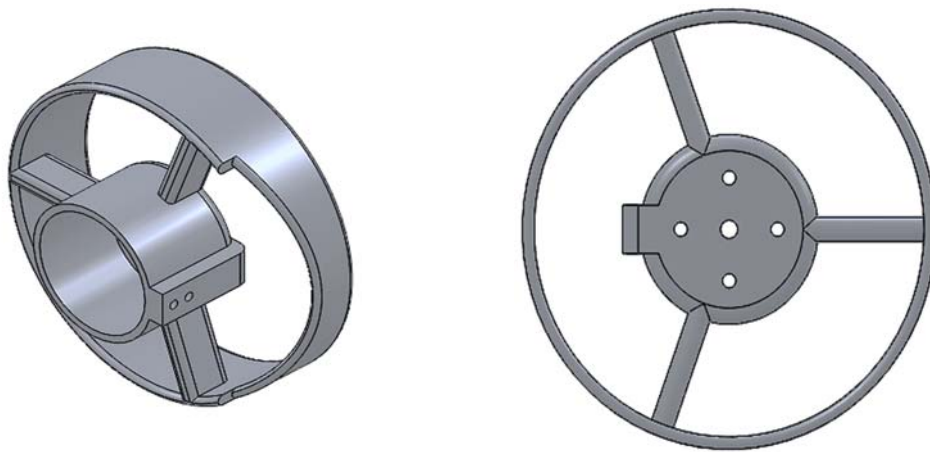


Fig. 3.23: The isometric scheme of the housing.



Fig. 3.24: Design of the housing.



Fig. 3.25: Different view of the thrusters.

The groove beside the thrusters has been created for the shaft of the servomotor. The place of the servo shaft is not at the middle of the space. This causes the edge of housing be stuck to the dolly block of servomotor on the body. Therefore, grooves were considered as in Figure 3.27.



Fig. 3.26: The propellers in the thruster.



Fig. 3.27: The servo motor in the dolly blocks.

3.5 Rotating Body

The bottom camera has been designed to rotate over 360° horizontally and 90° vertically. So a rotating body should be designed. For this aim, we used a readymade structure on which two stepper motors are set to rotate in the PAN (horizontal) and Tilt (vertical) directions. It also includes a communication protocol to control the rotation. The body has a transparent bubble mounted on the body as shown in Figure 3.28. It requires a 12V power supply with Pelco-D protocol [93]. Also, since the communication type is RS485, a HEXIN converter

[94] to RS232 is used for sending/receiving commands from the main processor board via its RS232 serial port.

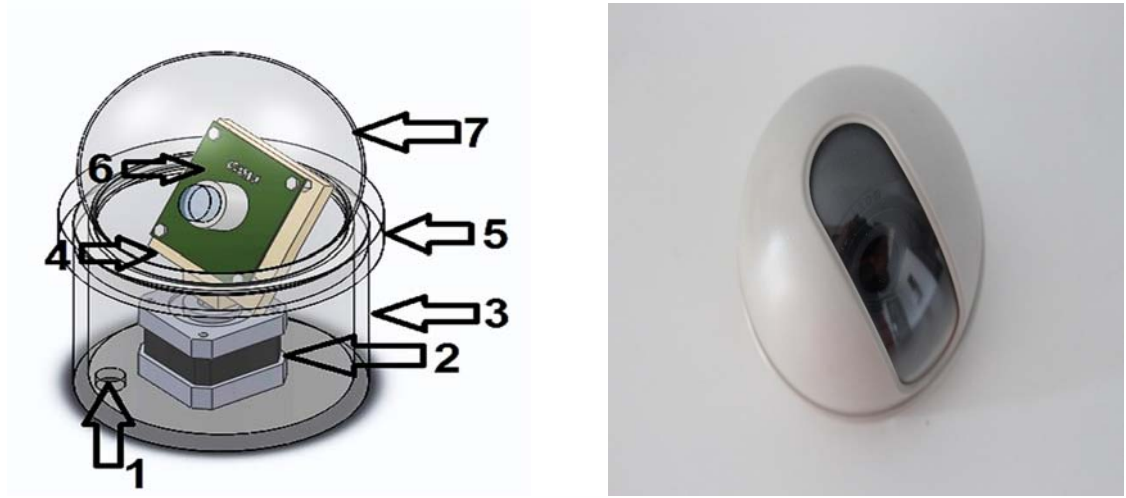


Fig. 3.28: Rotating body and location for top camera: (1) Cable compartment, (2) Stepper motor for 360 degree, (3) Camera body, (4) Camera board, (5) Threaded cap body, (6) Top camera, and (7) Transparent bubble.

3.6 Modeling the System

3.6.1 Introduction to Coordinate System

- *Earth-Centered Inertial (ECI)*: According to Figure 3.29, the center is the center of the earth and the x-axis is on the equator, referring to the 0° longitude and latitude at 0 point. However, with the earth rotation, the system framework remains fixed, its z-axis is toward the North and the y-axis completes the right hand rule. This system is called I and its axes are $i_1(X_1^E)$, $i_2(X_2^I)$ and $i_3(X_3^I)$.
- *Earth-Centered Earth-Fixed (ECEF)*: According to Figure 3.30, it is assumed that the system is connected to the earth and is entirely consistent with the moment of inertia; but over time, it rotates with the earth. The x-axis is on the equator and refers to prime meridian while the y-axis remains consistent with the z-axis inertial coordinates. This system is called E and its axes are $e_1(X_1^E)$, $e_2(X_2^E)$ and $e_3(X_3^E)$.

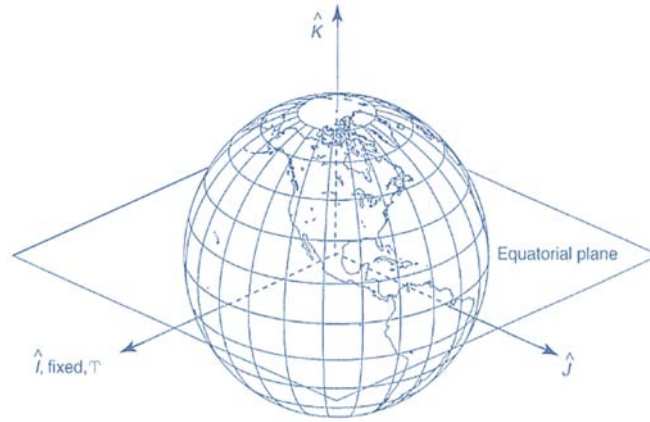


Fig. 3.29: Earth-Centered Inertial (ECI) [95].

It should be noted that the angle between ECI and ECEF is considered as the hour angle presented by $\dot{\Omega}_e^E \cdot t$. The angular velocity is constant and includes [95]:

$$\dot{\Omega}_e^E \cdot t = 7.2921150 \times 10^{-5} \frac{\text{rad}}{\text{s}} \quad (3.2)$$

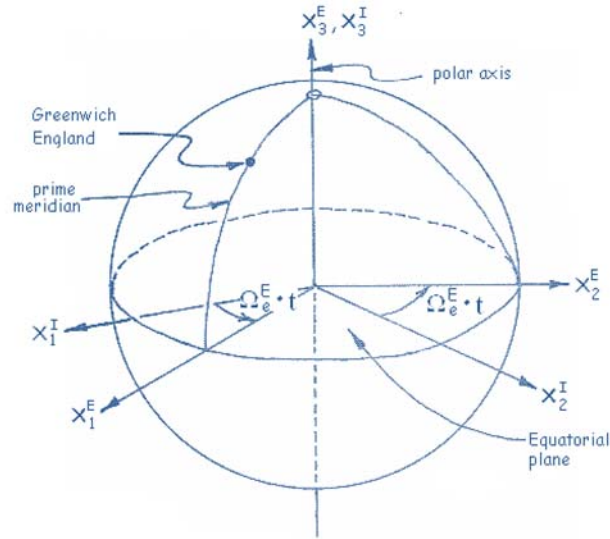


Fig. 3.30: Earth-Centered Earth-Fixed (ECEF) [96].

According to Figure 3.31, the center of the physical device matches with the AUV reference point. Its y-axis comes out of the front end of the vehicle, its x axis extends to the right and its z axis completes the right hand rule. The system is called B and its axes are B_1 , B_2 and B_3 .

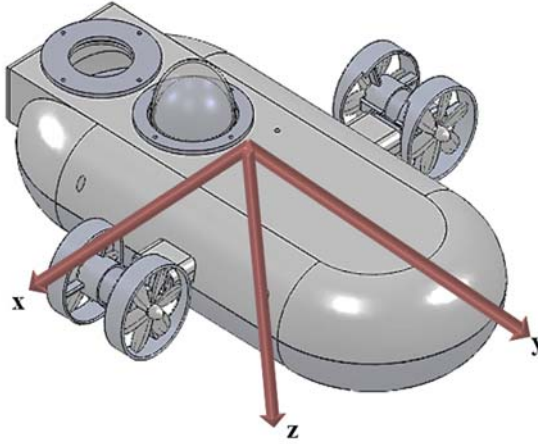


Fig. 3.31: Physical coordinates system.

3.6.2 Modeling the Gravity

- **Gravity Model:** Assuming that the Earth is a perfect sphere, the amount of gravity in terms of height is determined by the following equation:

$$g'' = \frac{\mu}{R^2} \quad (3.3)$$

where μ is the earth's gravitational constant ($3.986005 \times 10^{14} \text{ m}^3/\text{s}^2$) and R the distance that an object is from the center of the earth. It can be stated as the sum of h'' , the height of the object above the earth's surface and R_e , the earth's radius, as [97], [98].

$$R = R_e + h'' \quad (3.4)$$

In our case, h can be neglected.

3.6.3 Modeling the Forces

In this project it is assumed that the fluid in which the vehicle is run has a steadily and fixed density $\rho = 1000 \text{ kg/m}^3$ c.

The external forces applied on the vehicle include [99]:

- ✓ Weight force. For a system mass m_B and a gravity g , the weight force is equal to

$$F_W = m_g g \quad (3.5)$$

- ✓ Buoyancy force. According to Archimedes' principle, for a fluid density ρ and volume V' , the Buoyancy force is (Fig. 3.32) [99],[100]:

$$F_{Buoyancy} = \rho V' g \quad (3.6)$$

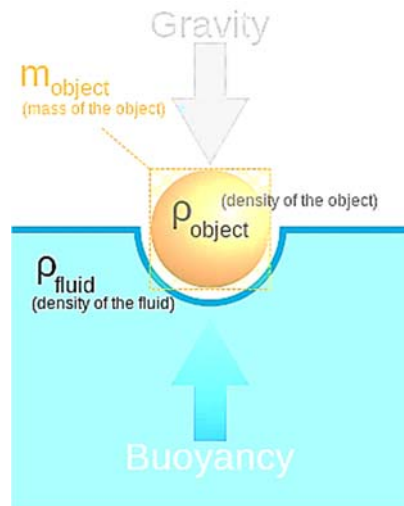


Fig. 3.32: Archimedes' principle [100].

- ✓ Thrust force. As shown in Figure 3.33, two engines on either side of the vehicle are used that provide the thrust force. The forces produced by the right and left engines are noted F_r and F_l , respectively.
- ✓ Added mass force. As the device moves within the fluid, a certain amount of liquid will move with it [101]. As inertial and Coriolis matrices relate the accelerations and angular/linear velocities of the body to the forces applied to the device, the added mass matrices and added Coriolis relate the accelerations and angular and linear velocities to the hydrodynamic force arising from the liquid displacement is applied to the device.

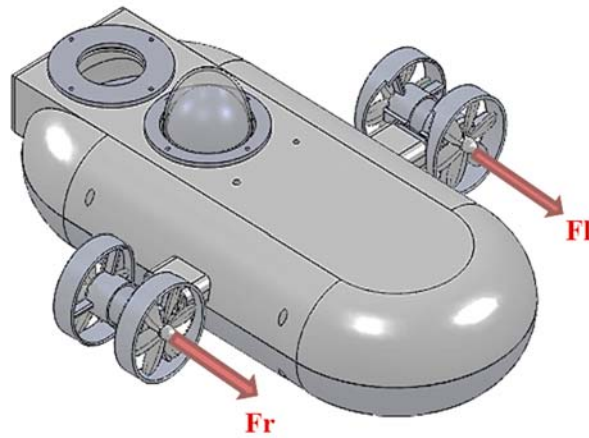


Fig. 3.33: Propulsion.

When a vehicle moves inside the fluid, a dynamic pressure distribution is created around it. Bernoulli's law states that the pressure applied on a differential surface dS depends on the fluid particle velocity on the differential surface and also to the water column height above it and this pressure applies the differential force dF and differential moment dM on the differential surface dS . The force and differential moment are called the force and added mass moment (Figure 3.34).

Assuming that the vehicle moves in a constant fluid, calculating its hydrostatic pressure will only depend on the water column above the differential surface dS . On the other side, to calculate the fluid velocity distribution on the surface of the device requires to evaluate the hydrodynamic pressure.

Fluid particles can move by an external force field. When force is applied to the fluid particles, its adjacent particles are accelerated due to a viscosity. That is why when a device moves in the fluid, the fluid particles which are located exactly on the device move at the same rate of the vehicle and the particles that are far from the vehicle move with different velocities. In fact, there is a velocity distribution on the differential surface dS and the particles with the farthest have a zero velocity. Such a phenomenon is explained by Navier Stokes equations [102], in which the velocity of the fluid and its partial derivatives are related to the force and the hydrostatic pressure.

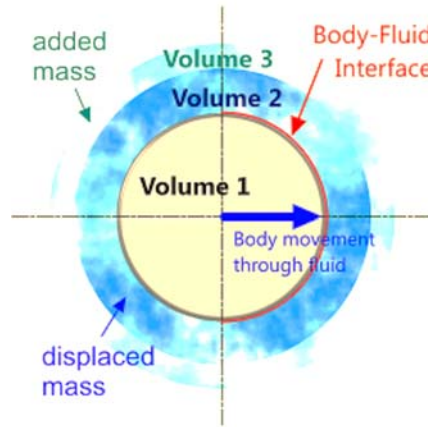


Fig. 3.34: Added mass [103].

According to [104], the force due to the added mass is

$$F_A = M_A \dot{v} + C_A''(v)v \quad (3.7)$$

where F_A is the added mass, M_A the added mass matrix, v the velocity and C_A'' the added Coriolis matrix. Added mass matrix features are as follows:

- Although the matrix is symmetric, it has twenty-one independent factors that should be calculated. However, this number can be reduced in the presence of apparent symmetry.
 - If the device lacks the apparent symmetry, the coupling coefficients of the added mass matrix are not zero that causes an acceleration in the direction other than the direction of the imposed acceleration.
 - The amount of added mass matrix coefficients depends only on the appearance of the product.
- ✓ Hydrodynamic force. As stated in [104], [105] the main hydrodynamic forces that are applied on a subsurface vessel include Quadratic drag forces and lineal skin friction forces. Although the equations of a vehicle with six degrees of freedom are highly non-linear, a series of simplifications are usually used.

The most common simplification is to assume that the linear and quadratic forces in the direction i_1 depend on the velocity vector components in i_1 direction. Thus, the hydrodynamic forces are calculated as follows:

$$F_{hydrodynamics} = - \begin{bmatrix} N_u u' \\ N_v v' \\ N_w w' \\ N_p p' \\ N_q q' \\ N_r r' \end{bmatrix} - \begin{bmatrix} N_{u|u}|u'|u'| \\ N_{v|v}|v'|v'| \\ N_{w|w}|w'|w'| \\ N_{p|p}|p'|p'| \\ N_{q|q}|q'|q'| \\ N_{r|r}|r'|r'| \end{bmatrix} \quad (3.8)$$

where:

- u' , v' and w' are the components of the velocity vector on the respective directions x, y and z.
- p' , q' and r' , are physical components of the angular velocity vector on the respective directions x, y and z.
- N_u and $N_{u|u}|$ are the linear viscosity and quadratic coefficients along x.
- N_v and $N_{v|v}|$ are linear viscosity and quadratic coefficients along y.
- N_w and $N_{w|w}|$ are linear viscosity and quadratic coefficients along z.
- N_p and $N_{p|p}|$ are linear viscosity and quadratic coefficients along p.
- N_q and $N_{q|q}|$ are linear viscosity and quadratic coefficients along q.
- N_r and $N_{r|r}|$ are linear viscosity and quadratic coefficients along r.

3.7 Display Control

In this part, the models used to simulate the AUV dynamics, which include the inertia mass and added mass model, the hydrodynamic forces model, the Euler's transformation model, the gravitational and buoyancy forces model, and the thruster model, are implemented in Matlab [106], [107] (Appendix 1). For solving the equations, we used the Runge-Kutta method with a time step size of 20 ms [108]. This value was tuned to simultaneously assure relatively fast convergence and acceptable accuracy. After completing the coding in Matlab, the LabVIEW (Laboratory Virtual Instrument Engineering Workbench) was used to design

a Graphical User Interface (GUI) as well as a virtual reality (VR) to virtually observe the maneuvers of the vehicle [107], [109]-[111]. The simulator designed in this thesis includes three main subsystems namely, a simulation platform and equations solving code, a GUI Environment, and a VR Environment.

3.7.1 Simulation Platform

The system on which the simulator has been installed has a quad-core processor. Given that the equation solving part should not be delayed, two cores (#0 and #1) have been assigned to this part while cores #2 and #3 have been allocated to the parts GUI and VR. Also the platform was responsible to assure the required communication between the different sub-systems. These communications are as follows:

- Creating a two-way communication between the solving and the GUI sub-systems
- Creating a one-way communication between the solving and the VR sub-systems
- Creating a one-way communication between the GUI and VR sub-systems

In general, it can be said that the simulation platform is a subsystem on which other subsystems are located and running.

3.7.2 GUI Environment

In this section, the user can communicate directly with the equation part, determining the different initial conditions before running the program that analyzes the movement behavior. This environment consists of five main components that include:

- *Initial Conditions*: This part is presented in Figure 3.35. This part should be set before running the simulation. Here the user can set the initial conditions of the vehicle including the following cases:
 - Initial latitude and longitude and height
 - Initial velocity
 - Initial physical angle
 - Initial angular velocity

Parameter	Value
lat_0 lon_0 alt_0 (deg deg m)	0
u_0 v_0 w_0 (m/s)	0
fi_0 teta_0 say_0 (deg)	0
p_0 q_0 r_0 (deg/s)	0

Fig. 3.35: Initial conditions

- *Simulation Monitor*: In the image presented in Figure 3.36, it is possible to perform some of the settings related to the platforms including setting the frequency of the loops or choosing the processor.
- *Set Parameters*: as seen in Figure 3.37, this part includes four main components:
 - Setting environmental parameters (Figure 3.38), In this part as seen in Figure 3.39, the simulation parameters can be set that include the following cases:
 - Determining the Greenwich longitude and the radius of the earth
 - Determining the rotational velocity of the earth: If this value is zero, it means that the Earth is not rotating
 - Determining the earth chamfer: If this value is zero, it means that the Earth is assumed a perfect sphere
 - Determining the fluid density in which the vehicle moves
 - Setting the volume mass parameters of the vehicle: In this part, as seen in Figure 3.39, the volume mass parameters of the vehicle are set as cases:

- Determining the total volume of the vehicle
- Determining the total volume and the moment of inertia of the entire body regardless of moving mass
- Determining the mass and the moment of inertia of the moving mass

Fig. 3.36: Platform settings.

Initial Condition Simulation Monitor Set Parameters Control

Greenwich Celestial longitude (deg): 0

R_earth (m): 6.37814E+6

w_E (rad/s): 7.29211E-5

Flattening: 0.00335281

density (kg/m³): 1000

Background: ☐

Underwater: ☐

Tmax (N): 7

M_A:

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Volume (m³): 0.0215

total mass: 0

Mass1 (kg): 19

Mass2 (kg): 2.57

1_I_G (kg.m²):

0.17128991	-0.00000062	-0.00382298
-0.00000062	0.62260902	-0.00000038
-0.00382298	-0.00000038	0.73746655

2_I_G (kg.m²):

0.00207345	0	0
0	0.00207345	0
0	0	0.00207345

XG1B_B (m): 0, 0, 0

XcbB_B (m): 0, 0, 0

XcpB_B (m): 0, 0, 0

Xt1B_B (m): 0, -0.2, 0

Xt2B_B (m): 0, 0.2, 0

vG2B_B: 0, 0, 0

aG2B_B: 0, 0, 0

XG2 max (mm): 100

Linear Damping:

X_u: 9

Y_v: 11

Z_w: 13

K_p: 0.25

M_q: 1

N_r: 0.5

Quadratic Damping:

X_u|u|: 40

Y_v|v|: 60

Z_w|w|: 100

K_p|p|: 0.2

M_q|q|: 0.8

N_r|r|: 0.4

Fig. 3.37: Settings parameters

Greenwich Celestial longitude (deg)
0
R_earthb (m)
6.37814E+6
w_E (rad/s)
7.29211E-5
Flattening
0.00335281
density (kg/m^3)
1000

Fig. 3.38: Environmental parameters

Volume (m^3)	1_I_B (kg.m^2)		
0.0215	0.17128991	-0.00000062	-0.00382298
total mass	-0.00000062	0.62260902	-0.00000038
0	-0.00382298	-0.00000038	0.73746655
Mass1 (kg)	2_I_G (kg.m^2)		
19	0.00207345	0	0
Mass2 (kg)	0	0.00207345	0
2.57	0	0	0.00207345

Fig. 3.39: Setting device parameters.

- Setting the hydrodynamic parameters: In this part, the hydrodynamic parameters of the device are determined by the user. This part includes three main components:
 - Determining linear drag coefficients (Figure 3.40)

- Determining quadratic drag coefficients (Figure 3.41)
- Setting the added mass elements (Figure 3.42)
- Locating important points versus the reference point: In this part, Figure 3.43, the user should determine the location vector of important points of the vehicle versus the reference point B. These important points include:
 - The center of the moving mass location, the buoyancy location, and the pressure location
 - The left and right engine force location

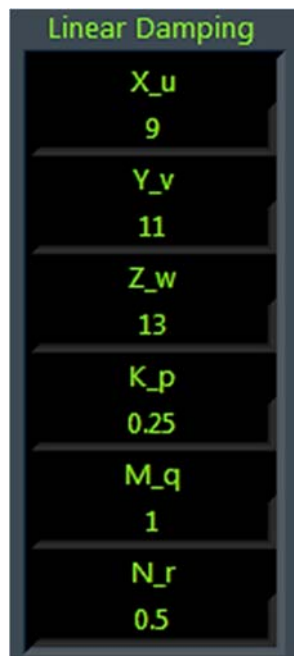


Fig. 3.40: Linear drag.

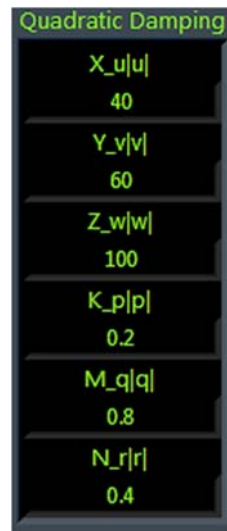


Fig. 3.41: Quadratic drag.

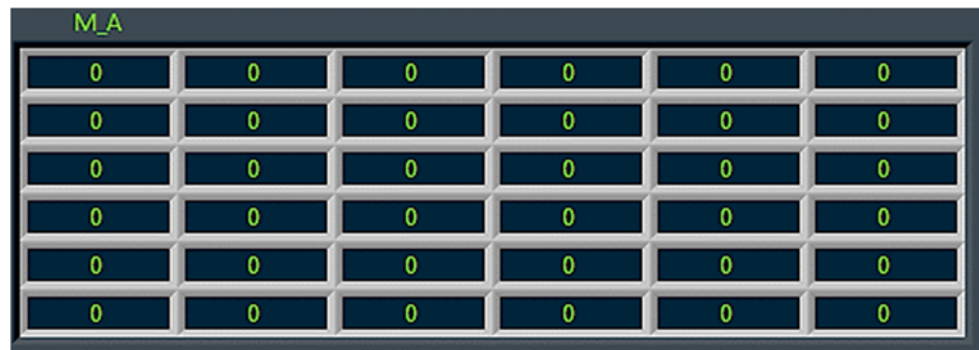


Fig. 3.42: Added mass.

- *Control*: This part is composed of three sets that include (Figure 3.44).
 - Displays (Figure 3.44): this part displays the main data of the vehicle:
 - Physical angular velocity
 - The latitude and longitude
 - Physical angle versus the North East Down (NED) coordinate system
 - Vertical velocity that resents the rate of reduced or increased height
 - Height

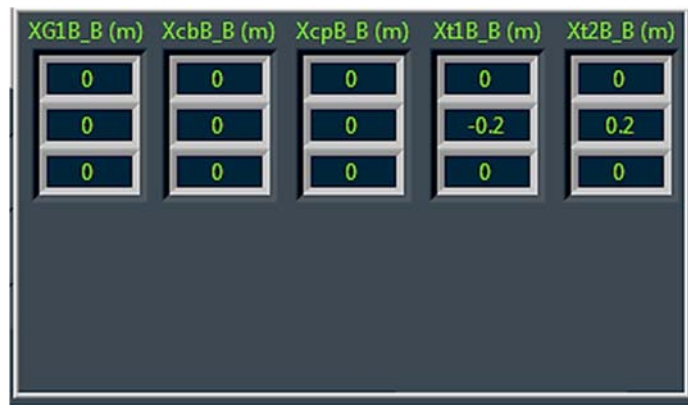


Fig. 3.43: The important points' location vector.



Fig. 3.44: Control

- VR control: To control the VR environment, two controllers have been implemented to include the perspective and can set the distance between the two cameras located on the back of the device (eye) versus the body.

- Vehicle control tools: Using these tools, the user can control the device in the GUI environment, i.e., determining:
 - the right engine speed
 - the right engine speed relative to the body
 - the left engine speed
 - the left engine speed relative to the body
 - the moving mass location
- *Data*: In this part, the diagrams of the movement data are visible to the operator. These data include:
 - Velocity,
 - Rotational velocity,
 - Body angle to the magnetic north and horizon,
 - Acceleration

3.7.3 VR Environment

In this part a three-dimensional model of the body is created in 1-scale. This graphic model is directly connected to the equation solution subsystem and displays the movement changes of the vehicle including the translational and rotational movements (Figure 3.45).

3.7 AUV Insulation

One of the crucial points to solve was how to efficiently insulate the AUV from any possible leakage. First, a groove around the body was spotted and sealed with an O-ring. However, the body was deformed after machining the Teflon [112],[113] leading to leakage. We then considered a range of silicon glues with various percentages and Gasket Makers [114].

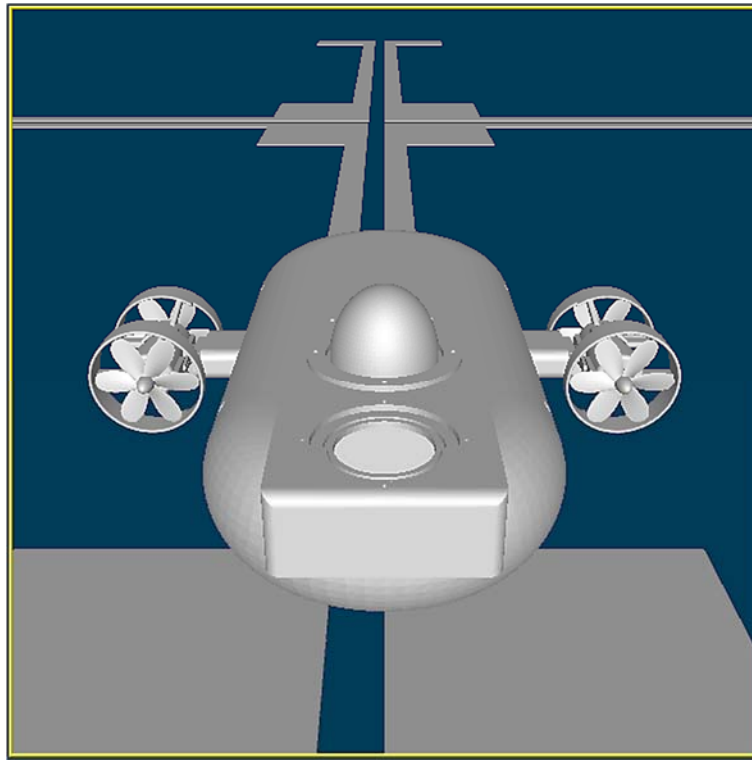


Fig. 3.45: VR environment

Yet, test showed that body twist imposed another leakage. Another alternative was to seal the AUV from outside using paraffin [115], [116]. The problem of using Paraffin is its proper use. Given the body size, the possibility of making improper paraffin pouring is high, which would lead to tiny holes. Then, instead of paraffin, silicon basic materials were tested too. The merit of these materials compared to paraffin is they require less time for being dried. Yet, a problem is the possibility of leaving vents, thus the sealing cannot be completely guaranteed.

Finally, we opted for Butyl, a polymer also known as Acronal [117]. This paste, used in the form of plastic sticky bands, was applied after filling the external groove with a thin layer of base silicon glue. Note that each band of Butyl can be attached and removed few times, which presents a significant advantage in case one need to access to the internal elements of the AUV for repairs, replacements, or adjustments.

Given their low friction level, Gasket Makers are among the best types of silicone materials for sealing Teflon. These materials differ in terms of operating temperature or tonnage they can bear. In the suggested sealing, and based on preliminary tests, we combined two types, i.e., the V-Tech Clear RTV Gasket Maker and the Mazda Gasket Maker, for optimal sealing. Then, a layer of Polyurethane was placed upon them. Therefore, the adopted method of putting silicon materials for sealing is as follows:

- First, the Mazda Gasket Maker must be put between the surfaces as a washer. Placing the paste on the bottom edge before placing the upper edge or vice versa is a difficult task. Therefore, the two parts of the body were first placed on each other, then, but just before fastening the screws, the paste was placed between the seal. Figure 3.46 shows this step.
- In the second step, screws are fastened in order for the paste to stick as much as possible between the two surfaces. Figure 3.47 shows this step. After 30 minutes, i.e., after the first maker becomes dry, the second maker (the V-Tech Clear RTV Gasket Maker) is put on it.
- After another 20 minutes (passing the dryness of the second layer), a Polyurethane layer is applied.



Fig. 3.46: The paste between the seal.



Fig. 3.47: The screws.

Note that we had to make a hole for the lens of the camera in the bottom as well as two other holes for the IMU and Magnetometer sensors to decrease the dielectric effect of Teflon. Fortunately, the bottom part of the body experienced less twist due to the existence of the wall, making the use of O-rings possible.

This approach was also used to seal the thrusters. O-Rings are indeed the most common technique for sealing moving parts whether be rotating or reciprocating. The fundamental problem with them in rotating shafts is power dissipation, in the sense that drivers should always spend some power to overcome the resistance caused by O-Ring. Therefore, we would have to add this amount of resistance to the torque and power requirements in our calculations as well as to coat the shaft and its above chamber with grease [118], [119].

3.8 Conclusion

In this chapter, we addressed the design of the AUV body including thrusters, propellers, pickets (the location of thrusters), internal and external parts of the AUV. The body material and the reason to choose that was explained and regarding the material and carried out design, we conducted an accurate analysis of the AUV body strength in maximum pressure, stress,

tension and displacement on that and results indicate the higher body strength safety factor within the range of the AUV mission.

By design and placement of the mass shifter inside the AUV which are complementary to thrusters performance, we modeled the applied forces on AUV and we addressed the static and dynamic mathematical modeling of the body and we implemented the achieved equations in MATLAB and we used the LabVIEW environment to design a user interface environment or GUI and also we could observe maneuvers of the device virtually and also in real environment by designing a VR environment. We also could observe and analyze the performance of the system at each moment by the designed control display and finally, by introducing an efficient measure to insulate the AUV and the components, we insulated the AUV body against each type of water leaks and infiltration into it.

The next chapter will be devoted to the design of the electrical parts.

CHAPTER 4

DESIGN OF ELECTRONICS PARTS

Building an AUV comprises three parts namely, Mechanical, Electronic Hardware, and Software. We discussed in Chapter three about issues related to mechanical. In this chapter, the designing and manufacturing procedure of electrical and electronic AUV parts will be investigated (Figure 4.1). Their datasheets are summarized in Appendix II.

The objective of the present work is to design an AUV that can safely navigate through obstacles. For this purpose, several devices should be investigated.

4.1 Main Board and Sensors

All AUV processes including camera/sensor data, detecting obstacles, controlling motor operation, etc., should be managed via a main processing board. In this work, we used the mini PC board Giada i200-BG000, Celeron [120]. Considering the amount of data to manage in real time, the AUV operation needs a high-speed processor to efficiently receive and send commands. With its high processing speed, supporting USB ports, and 802.11ac dual band wireless network, Giada i200-BG000 board is in fact one of the most appropriate devices. Furthermore, its light weight, small size, and low power consumption make it the most

suitable for our application. This board has the following features:

- Mounting: optional VESA mounting kit.
- Adapter1: optional video adapter for HDMI Connector.
- Adapter2: optional video adapter for DisplayPort.
- Wi-Fi: 802.11ac dual band wireless network with Bluetooth 4.0.
- 12V to 19V supply voltage.

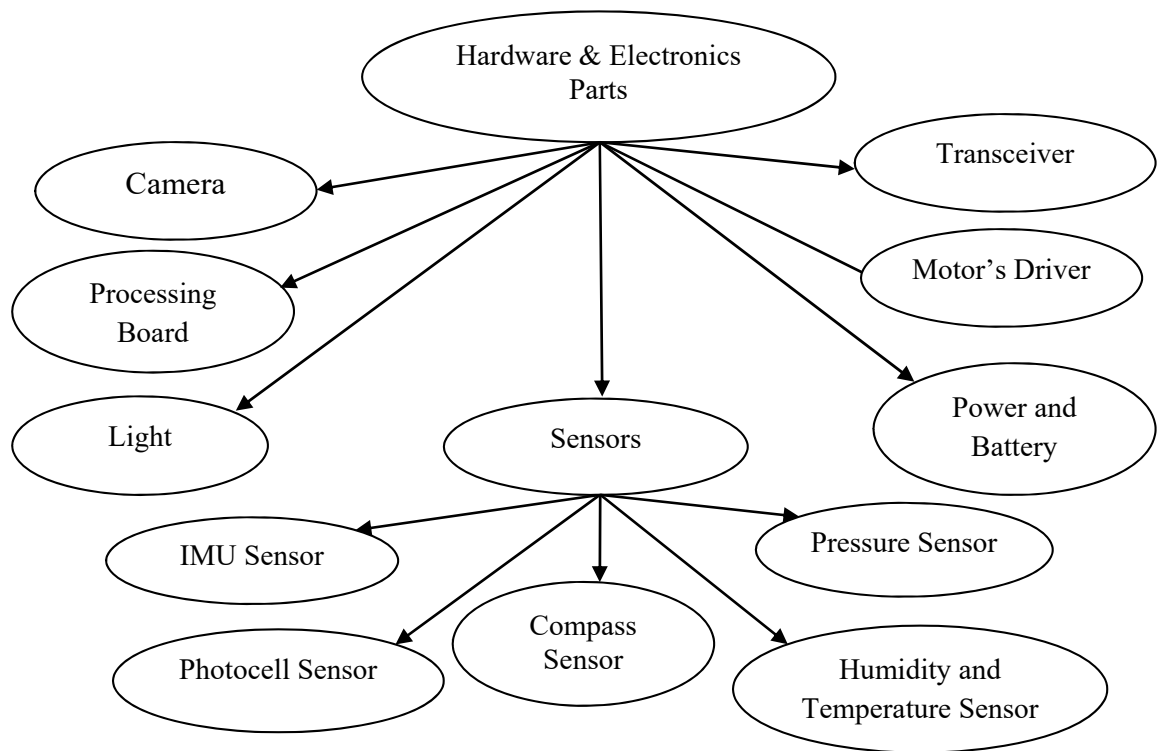


Fig. 4.1: Electronics parts.

All ports of this board will be used. The serial port is dedicated to the rotating body while the two cameras, the sensor board, and the transceiver module will be connected to the USB ports. Figure 4.2 shows the block diagram of Main Processor connections.

To navigate safely, an AUV should be provided with a minimum of information about its environment so that it can take appropriate decisions.

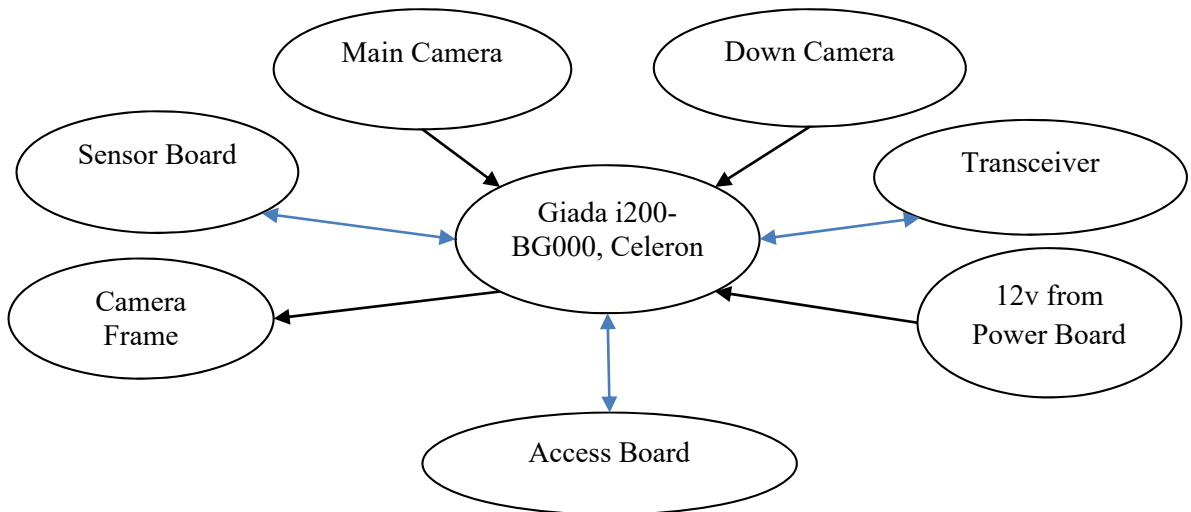


Fig. 4.2: Block diagram of main processor.

Such data mainly come from sensors. In this work, the following sensors have been used:

- Compass Sensor for knowing the direction of the magnetic North.
- IMU sensor for collecting data on Roll and Pitch angles and acceleration.
- Pressure sensor to know the water pressure and depth of the device.
- Humidity and temperature sensors for measuring the temperature and humidity of the chamber of the device.
- Photocell sensor to measure ambient light.

A two-wire TWI protocol placed on a shared bus has been used to reduce wiring.

4.2 Sensors and Sensor Boards

In this work, we used several different sensors. The common factors in selecting these sensors are their reasonable price, very small size, and good performance. They should also exhibit high sensitivity and resistance against heat, as well as low current consumption and be able to work with I2C communication protocol [121], [122]. Because of their sensitivity, and in order to minimize the risk of communication errors with the main processor, each sensor board was designed separately.

4.2.1 IMU Sensor

IMU sensor is composed of three sensors: acceleration counter, gyroscope, and magnetometer. All three perform measurements in the three directions of the coordinate system. The acceleration counter measures the acceleration of the device, its balance, and its deviation. The gyroscope measures the circulation rate of the device and the magnetometer determines the position of the device in comparison with the North Pole.

The selected sensor is the MPU-9250, one of the most advanced sensors (Figure 4.3) with a 3-axis gyroscope, 3-axis accelerometer, 3-axis compass, internal temperature sensor, and equipped with an advanced system of digital motion processing [123].

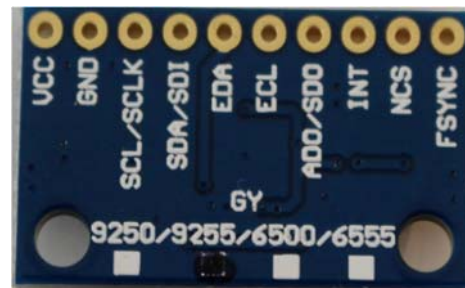
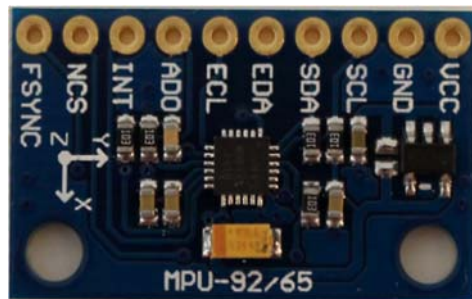
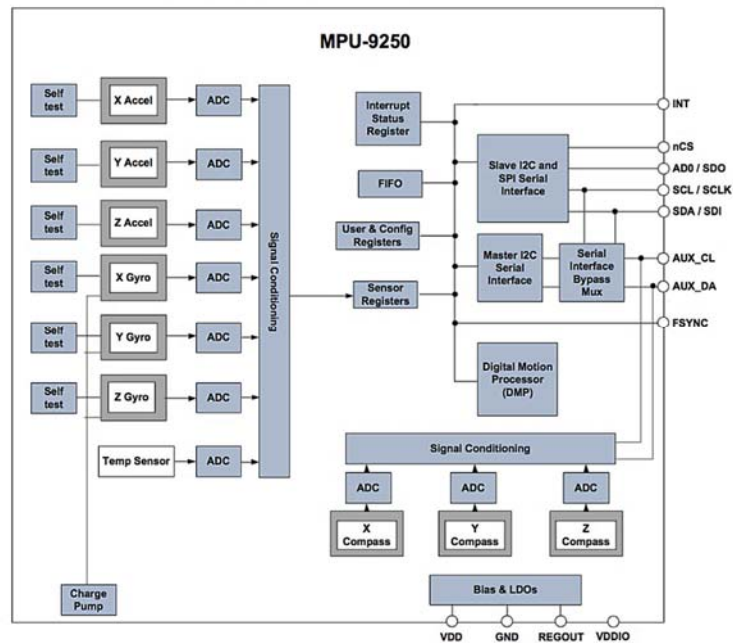


Fig. 4.3: MPU-9250 sensor [123].

Unlike its predecessor chips, the MPU-9250 chip is equipped with the I2C communication protocol that leads to very high speed communication, extremely low current consumption and protection against noise. In addition, its thermal performance range is -40 to 85 °C [123].

4.2.2 Compass Sensor

This sensor is used to find the heading point of the AUV. Because the magnetometer sensor is highly sensitive to electromagnetic noise and the earth magnetic field intensity quite weak under water, we had to use a separate digital compass sensor in addition to the IMU sensor to increase accuracy. The selected compass sensor is the HMC6343 (Figure 4.4), a fully integrated high-end electronic compass module that can compute and give an accurate heading direction within a couple of degrees. It is tilt compensated and calibrated to handle magnetic distortions. This breakout board allows an easy use. All what is required is a power supply and I2C connections to a microcontroller so that the module can receive commands and send data back to the user.

The IC combines 3-axis magneto-resistive sensors and 3-axis MEMS accelerometers, analog and digital support circuits, a microprocessor and algorithms in firmware required for heading computation. The HMC6343 Breakout needs to be supplied with 3.3V and 4.5mA and can measure and compute a heading direction every 200ms (5Hz) [124].

As shown in Figures 3.7 and 3.8 a circular space in the front of the AUV has been considered to locate the IMU and the Compass. These containers have holes on the bottom with transparent lens. This is due to the weakening of the ground's electromagnetic waves inside the body because of its material, thickness and the possibility of compass and IMU error. This part is protected by aluminum foil against noise. According to the available space and the sensors involved, two electronic boards, each with radius of 27 mm, were designed to connect the sensor board with an IDE cable (Figures 4.5 and 4.6).

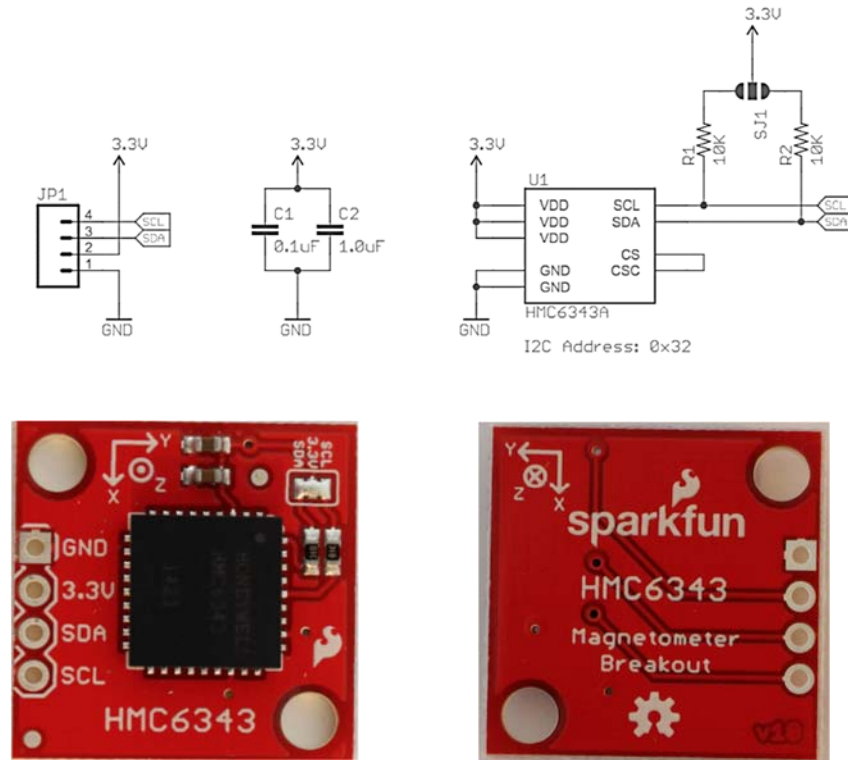


Fig. 4.4: SparkFun HMC6343 breakout [124],[125].

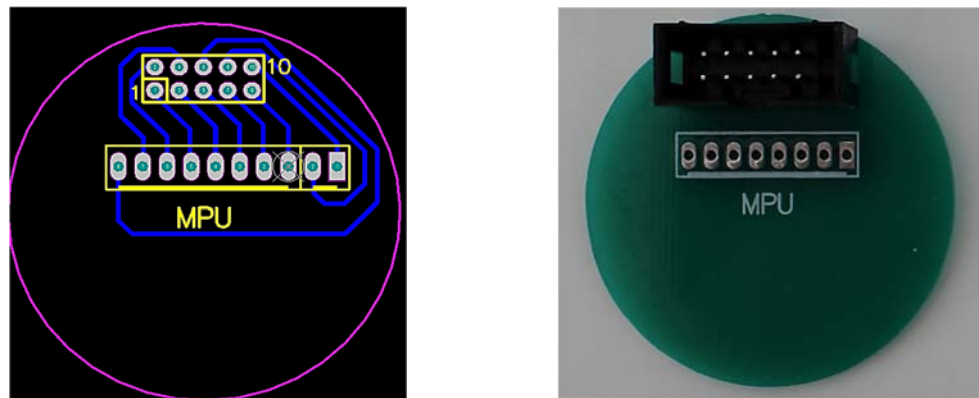


Fig. 4.5: PCB of the MPU sensor board.

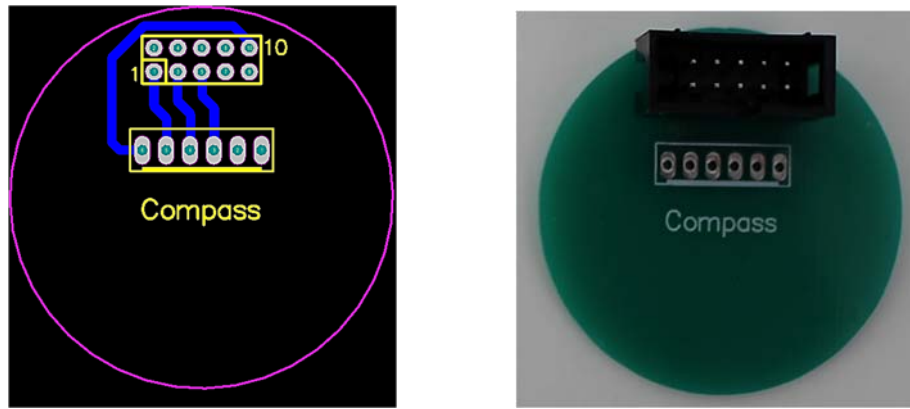


Fig. 4.6: PCB of the compass board.

4.2.3 Pressure Sensor

Because there is a linear relation between water pressure and water depth, we used a pressure sensor to measure the device depth. If an utmost depth of 20m, the utmost pressure will be around 3bar. The selected pressure sensor is the MS5803-14BA (Figure 4.7), a new generation of high resolution pressure sensors with I2C interface. It is optimized for depth measurement systems with a water depth resolution of 1cm and below. A high resolution temperature output allows the implementation of a depth measurement system and thermometer function without any additional sensor. A gel protection and antimagnetic stainless steel cap protects against 30 bar overpressure waterproof [126].

4.2.4 Humidity and Temperature Sensor

In this digital sensor (SHT-11), a unique capacitive sensor element is used to measure relative humidity while the temperature is measured by a band-gap sensor. Serial I2C interface and factory calibration, allow easy and fast system integration. Board is set to use 3.3V power supply by default. Solder PWR SEL SMD jumper to 5V position if used with 5V systems. is a high precision digital sensor to measure temperature and humidity (Figure 4.8), the sensor has a 5V power supply [127], [128].

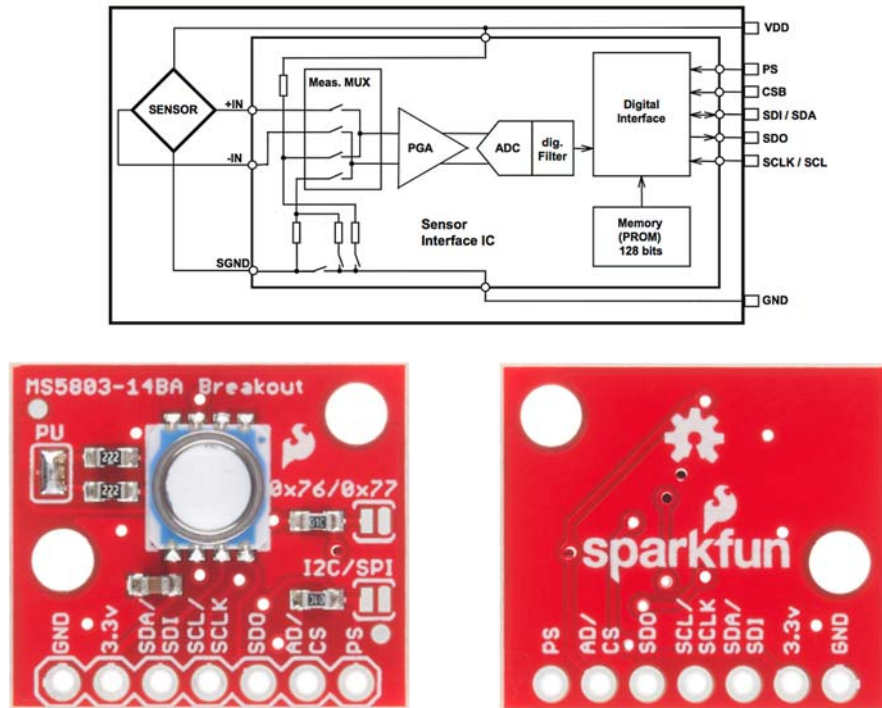


Fig. 4.7: Block diagram of MS5803-14BA [126] and sensor board.

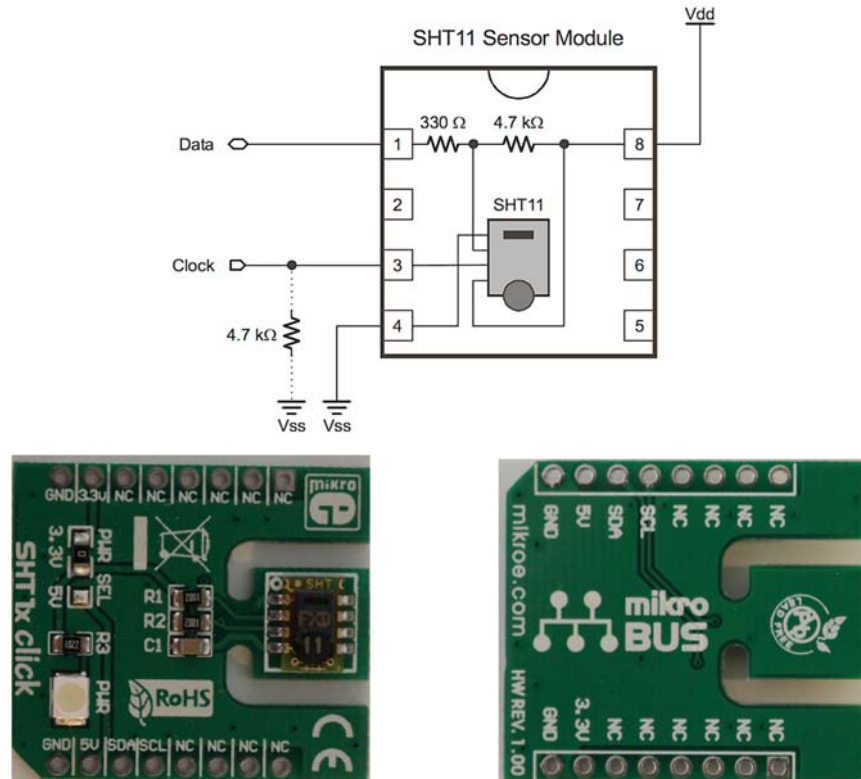


Fig. 4.8: Humidity and temperature sensor [127].

4.2.5 Sensor Board

Based on the microcontroller ATMEGA64 [129] (Figure 4.9), the aim of the sensor board is to collect/send data from/to the above sensors via a serial port as well as to receive the required commands from the main processing board and send them to the motor driver board via two UART communication serials. The mass shifter circuit, composed of two switches at both ends of the rod, as well as the LED lighting control circuit are mounted in this board.

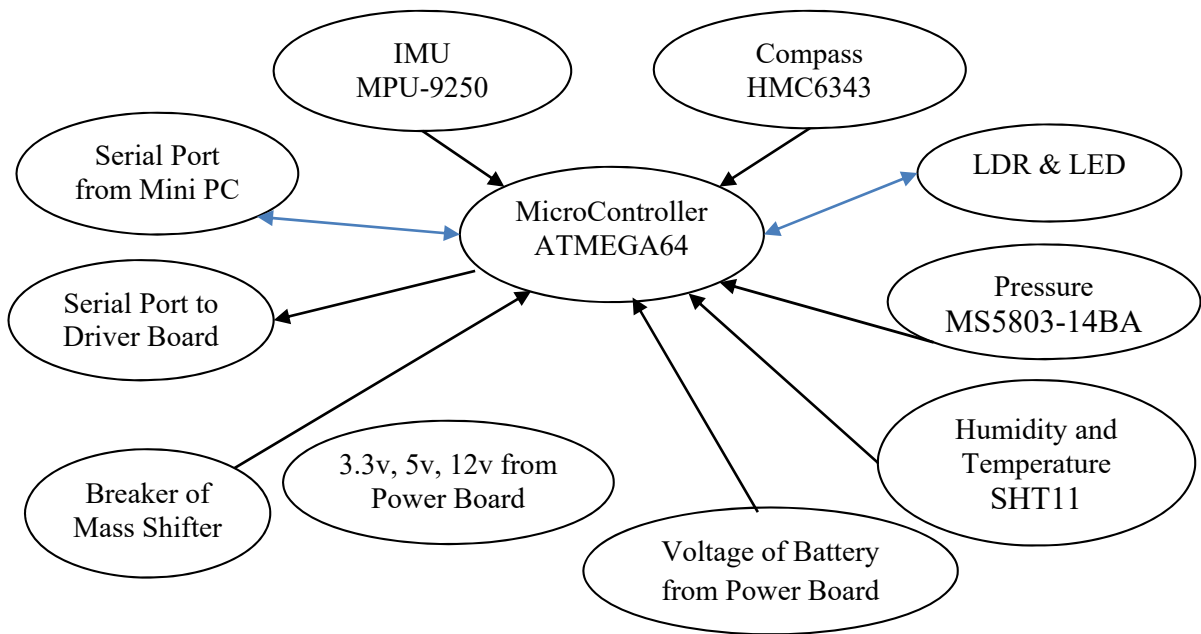


Fig. 4.9: Block diagram and schematic of the sensor board.

The programmer socket, the reset circuit, the battery voltage detection socket, and the LDR photocell sensor used to detect changes in the resistance of the circuit board are also mounted on this board, along with the sensor socket HMC6343, MPU-9250, MS5803 and SHT11 (two). A vacant socket (for future use) is also part of this board.

Due to the high space required by all fittings and connectors, this board has been designed in two levels to save space. The circuits are located in the bottom level while the sockets and connectors are located on the top level of the board (Figure 4.10).

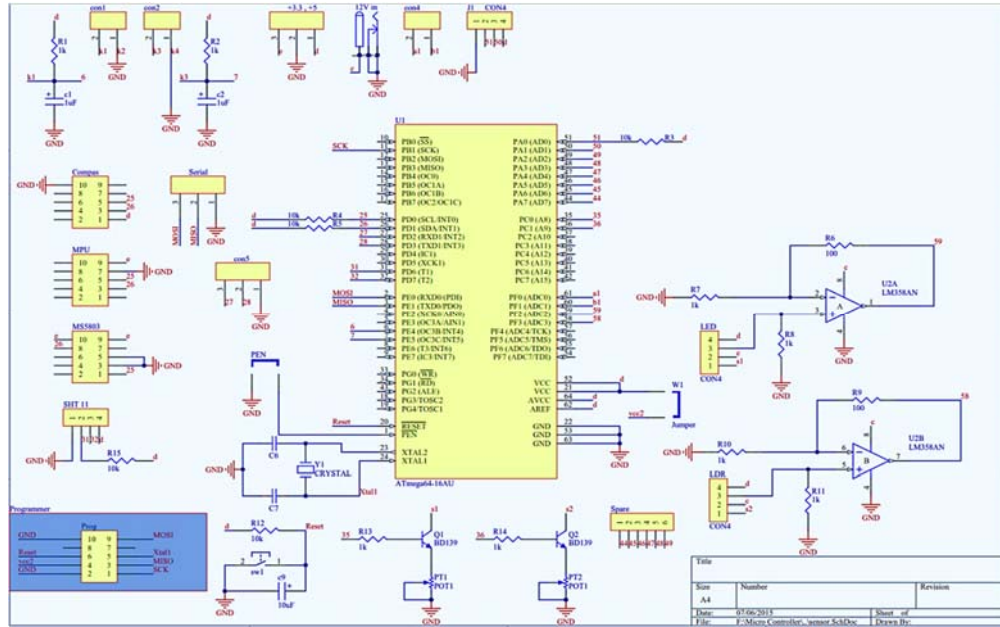


Fig. 4.10: Sensor board, top board with dimensions $67.40 \times 48.40 \times 2 \text{ mm}^3$ and bottom board with dimensions $69.85 \times 49.80 \times 2 \text{ mm}^3$.

4.3 Transceiver

As mentioned before, the AUV's function and mission is to work in water at a maximum depth of 8m. The Mini PC (Giada i200-BG000, Celeron) uses Wi-Fi and Bluetooth systems. However, due to underwater antenna issues in terms of cost and power supply, we decided to use two transceiver modules in order to evaluate the system performance during AUV mission and controlling the AUV in emergency modes. We chose the RF7020 (ADF7020), a low power of size $37.5 \times 18.3 \times 2.54 \text{ mm}$. The reason for choosing this module is its AUV's good mileage performance (about 3 km), its USB port connection, and its light weight as well. One of the two modules is connected to the external controller while the second is inside the AUV and is connected to the Mini PC through a USB port [130]-[132].

The PCB of the bottom and the top board are shown in Figures 4.11 and 4.12 and their specifications summarized in Tables 4.1 and 4.2, respectively.

The transceiver uses the GFSK modulation and is directly connected to the main processor board via a Serial USB adapter and a USB port. It can be fed from the same port.

Table 4.1 : Description of the bottom board.

Quantity	LibRef	Footprint	Designator	Description	Comment
2	cap10u	cap10u	C1, C2		1uF
2	Cap ceramic	My_cap100n	C6, C7	CAP-100Nf	Cap
1	cap10u	cap10u	C9		c?
2	Jumper	My_IDC-2	P.S, PEN	Jumper Wire	Jumper
2	Header 19	HDR1X19	P1, P2	Header, 19-Pin	Header 19
2	POT1	POT1	PT1, PT2	Variable Potentiometer standing	POT1
2	BD139	SOT-32	Q1, Q2	NPN Transistor	BD139
8	Res	My_axial-0.4	R1, R2, R7, R8, R10, R11, R13, R14	resistor 1/4 wat	1k
5	Res	My_axial-0.4	R3, R4, R5, R12, R15	resistor 1/4 wat	10k
2	Res	My_axial-0.4	R6, R9	resistor 1/4 wat	100
1	ATmega64-16AU	64A_N	U1	8-bit AVR Microcontroller, 4.5-5.5V, 64KB Flash, 2KB EEPROM, 4KB SRAM, 64-pin TQFP, Industrial Grade (-40°C to 85°C), Pb-Free	ATmega64-16AU
1	LM358AN	DIP8	U2	Low-Power Dual Operational Amplifier	LM358AN
1	CRYSTAL	My_Crystal	Y1	Crystal	CRYSTAL

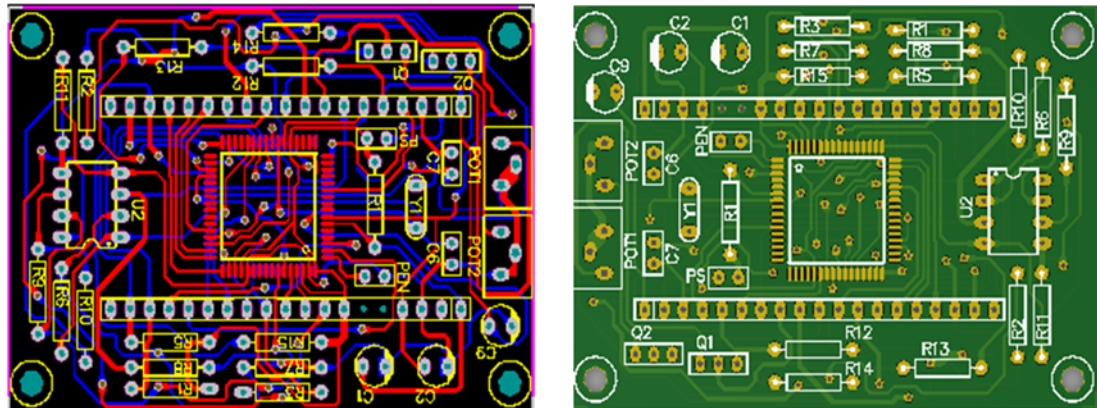


Fig. 4.11: PCB of the bottom board.

Table 4.2: Description of the top board.

Quantity	LibRef	Footprint	Designator	Description	Comment
3	con 3_3.9	con 3_3.9mm	+3 , +5, Serial 1, Serial 2		con
1	PWR2.5	KLD-0202	12V in	Low Voltage Power Supply Connector	12V in
3	con 2_3.9mm	con 2_3.9mm	Battery, Key 1, Key 2		con
4	WE-2.54 mm IDC & Transition Connector 61201022323	61201022323	Cmps, Mpu, Ms5803, Prog	WR-BHD 2.54mm Board To Cable Transition Connector	WE-2.54 mm IDC & Transition Connector 61201022323
4	CON4	MT6CON4V	LED 1, LED 2, Sht 1, Sht 2	Connector	CON4
2	Header 19	HDR1X19	P1, P2	Header, 19-Pin	Header 19
1	CON6	MT6CON6V	Spare	Connector	CON6
1	Tac switch	MY_TACTILE-PTH	sw1	small tac switch	Tac switch

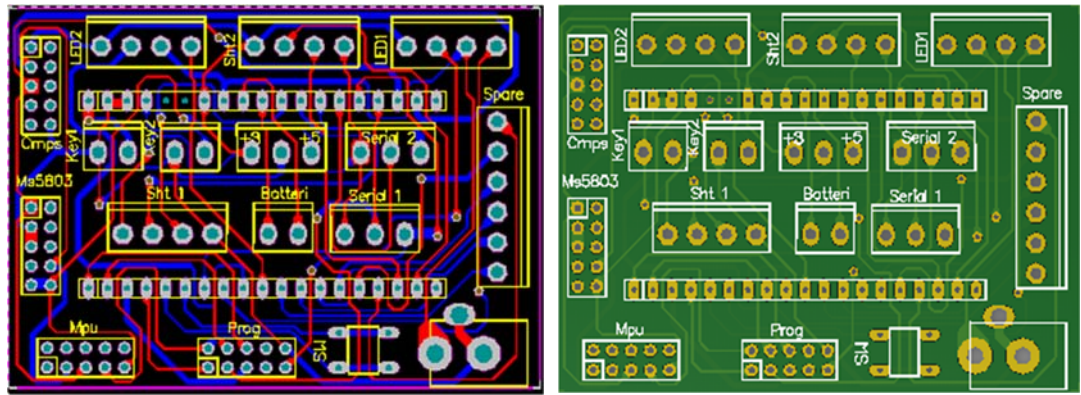


Fig. 4.12: PCB of the top board.

4.4 Camera and Light Board

In recent years, demands for underwater tasks, such as excavating of ocean resources, exploration of aquatic environments, and inspection of underwater structures, have increased. We have then included two cameras in the AUV design: the first in a transparent dome-like space at the top of the AUV with a capability of a 360° circulation and the second on the AUV floor, both for observing the surroundings and detecting obstacles. These cameras, the HD Webcam C615 (Figure 4.13), have the following specifications [133]:

- low-voltage and low current consumption
- high resolution of 1080p and 720p
- USB port for connection to the main processor board
- automatic recognition by the main processor board
- automatic setting up of light and resolution

Only the internal board of the camera is used in our design for ease of installation and for cumbersomeness of the case of the camera.

The down camera, a 1.0 Megapixel 720p USB Camera with Infrared (IR) Cut and IR LED for Day & night Smart Video Surveillance (Figure 4.14).

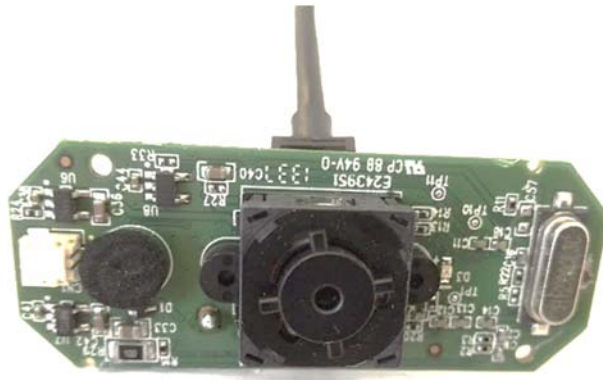


Fig. 4.13: The main camera board.

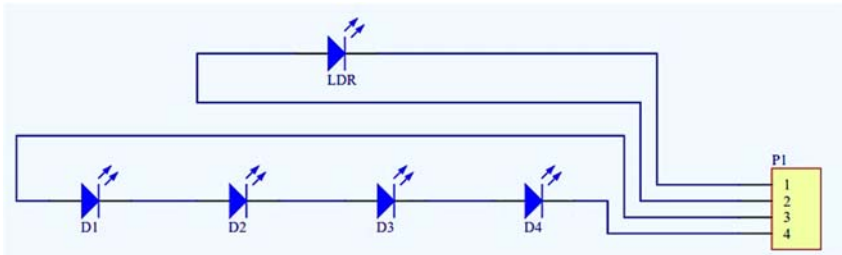


Fig. 4.14: The bottom camera.

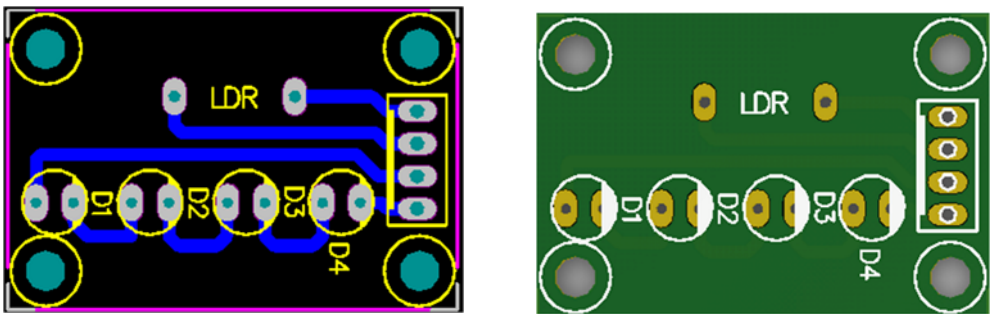
Also, LEDs have to be added around the cameras to shed light into the water environment. A light board was then designed and installed on the top of the camera (Figure 4.15 and Table 4.3). It consists of an LDR (Light Dependent Resistor) photocell sensor [134] to measure the ambient light and four LEDs for providing light. Figure 4.16 shows the LEDs performance at night in a pool. The internal resistance of photocell sensors depends on the ambient light. So, the output of this sensor will go to the sensor board and any change in resistance will be turned into variation in voltage and measured with an ADC microcontroller. If the ambient light is less than a given threshold, the microcontroller would prompt the LEDs to be turned on.

Table 4.3: Description of the the LED board.

Quantity	LibRef	Footprint	Designator	Description	Comment
5	LED 5mm	LED-1	D1, D2, D3, D4, LDR	Typical LED	LED 5mm
1	Header 4	My_1X04	P1	Header, 4-Pin	Header 4



(a)



(b)



(c)

Fig. 4.15: LED board: (a) Schematic, (b) PCB, (c) Final circuit.



Fig. 4.16: (a) View of the AUV at night, (b) LEDs performance with maximum power.

4.5 Power and Battery

The above circuits and other electronic components used in the AUV require different voltages as summarized in Table 4.4. Therefore, we decided to use two rechargeable batteries: a 20V/14Ah battery and a 12V/12Ah battery (with regulators). As seen, different parts of the circuit require different voltages. For this purpose, a 69.70*48.60*2 mm power distribution board or Power Board was built to provide the requested voltages (Figure 4.17 and Table 4.5). A 7805 regulator was used to produce 5V and a LF33 regulator for producing 3.3V from the main 12V power source, this later being made of two paralleled 7812 regulators to form a 20V power source in addition to the 12V battery.

Two 20V outputs, three 12V outputs (from the battery), two 12V outputs (from the regulator) and two 5V and 3.3V outputs are embedded in this board (Figure 4.18).

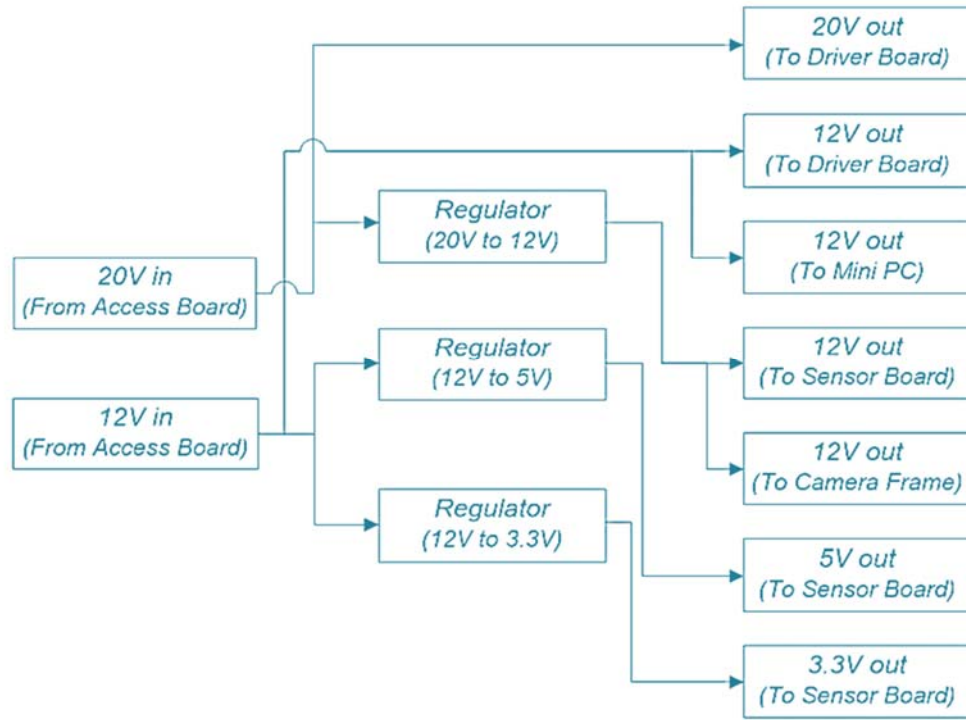
Furthermore, two safety fuses have been included. Also, since the 12V battery voltage should not fall below 9V and the 20V battery voltage should not fall below 15V, two voltage detection circuits have been embedded via making use of Op-Amp LM358. Their outputs should then be measurable by the analog to digital converter of the sensor board microcontroller. As a result, the output from these circuits would go to the sensor board.

Table 4.4: Required voltages.

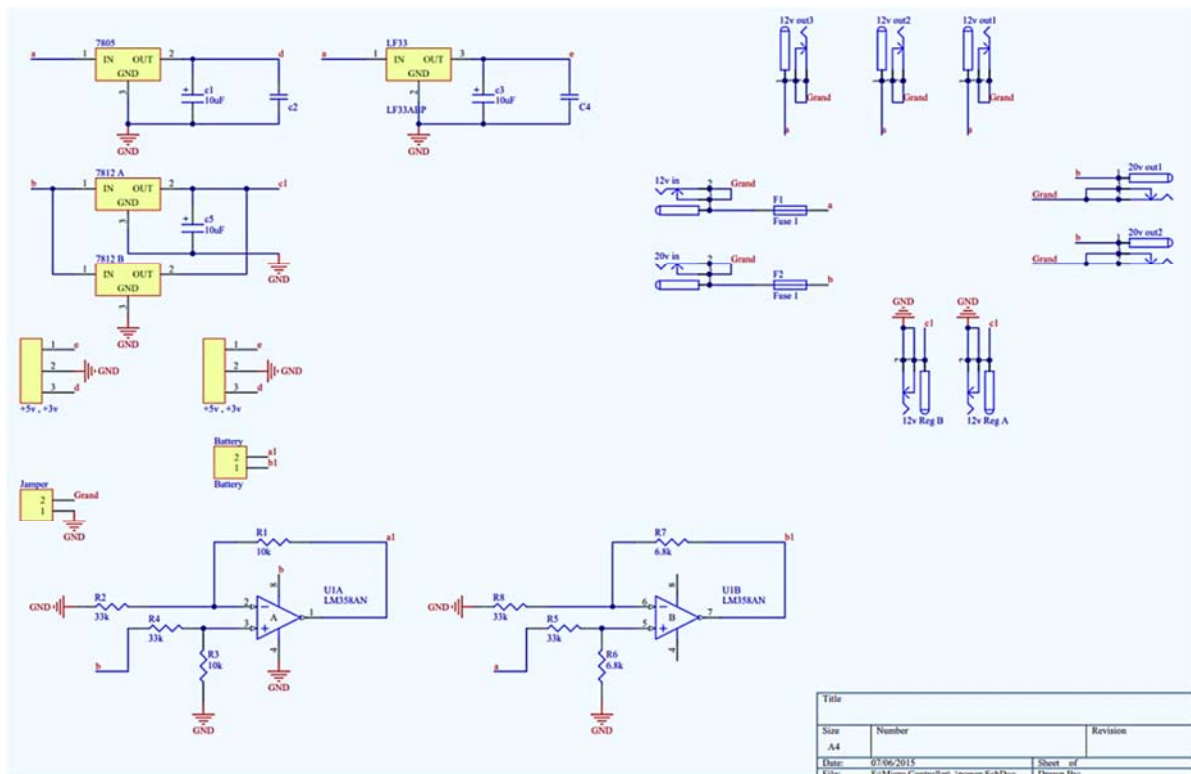
Component Name	Required voltage	Approximate required current
Camera	5	below 100mA
LED	3	10mA
Sensors	3.3-5	below 100mA
Microcontroller	5	below 100mA
Transceiver	5	below 50mA
Processing Board	12-19	1A to 1.5A of
Servo Motor	6	0.5A
Camera Rotor	12	below 200mA
Mass Shifter Motor	12	1A
Driving Device	18	3A

Table 4.5: Description of the power distribution board.

Quantity	LibRef	Footprint	Designator	Description	Comment
1	con 3_3.9	con 3_3.9mm	+5v , +3v		+5 , +3B
1	con 3_3.9	con 3_3.9mm	+5v , +3v a		+5v , +3v
1	PWR2.5	KLD-0202	12v in	Low Voltage Power Supply Connector	12 in
1	PWR2.5	KLD-0202	12v out1	Low Voltage Power Supply Connector	12v out1
1	PWR2.5	KLD-0202	12v out2	Low Voltage Power Supply Connector	12v out2
1	PWR2.5	KLD-0202	12v out3	Low Voltage Power Supply Connector	12v out3
1	PWR2.5	KLD-0202	12v Reg A	Low Voltage Power Supply Connector	12v Reg A
1	PWR2.5	KLD-0202	12v Reg B	Low Voltage Power Supply Connector	12v Reg B
1	PWR2.5	KLD-0202	20v in	Low Voltage Power Supply Connector	20 in
1	PWR2.5	KLD-0202	20v out1	Low Voltage Power Supply Connector	20v out1
1	PWR2.5	KLD-0202	20v out2	Low Voltage Power Supply Connector	20v out2
1	L7805CV	TO220ABN	7805	Positive Voltage Regulator	L7805CV
2	L7812CP	ISOWATT220AB	7812 A, 7812 B	Positive Voltage Regulator	L7812CP
1	Header 2	HDR1X2	Battery	Header, 2-Pin	Battery
1	cap10u	cap10u	c1		c1
1	Cap ceramic	My_cap100n	c2	CAP-100Nf	c2
1	cap10u	cap10u	c3		c3
1	Cap ceramic	My_cap100n	C4	CAP-100Nf	c4
1	cap10u	cap10u	c5		c5
2	Fuse 1	PIN-W2/E2.8	F1, F2	Fuse	Fuse 1
1	Header 2	My_1X02	Jumper	Header, 2-Pin	Header 2
1	LF33ABP	TO-220FP	LF33	500mA, Very Low Drop (0.45V) Voltage Regulator	LF33ABP
2	Res	My_axial-0.4	R1, R3	resistor 1/4 wat	10k
4	Res	My_axial-0.4	R2, R4, R5, R8	resistor 1/4 wat	33k
2	Res	My_axial-0.4	R6, R7	resistor 1/4 wat	6.8k
1	LM358AN	DIP8	U1	Low-Power Dual Operat	LM358AN



(a)

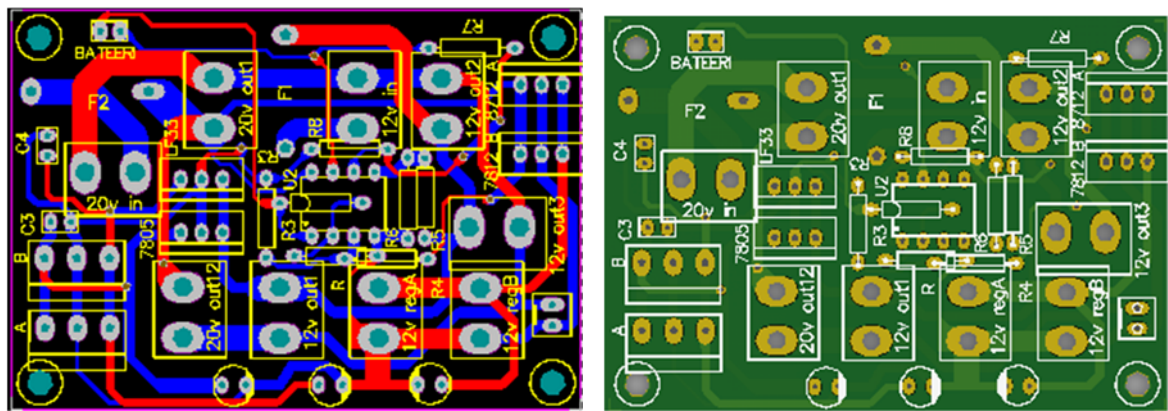


(b)

Fig. 4.17: a) Block diagram and b) Schematic of the power distribution board.



(a)

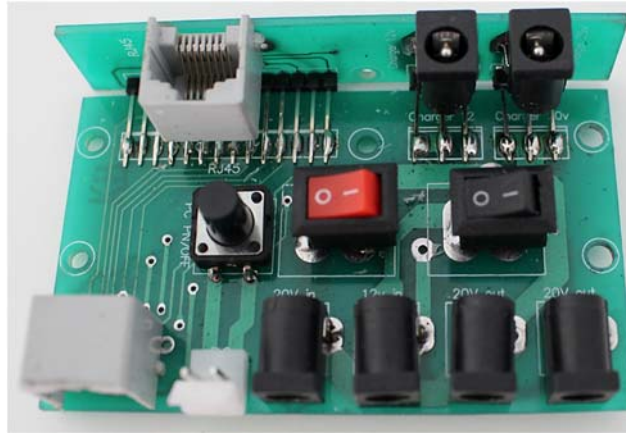


(b)

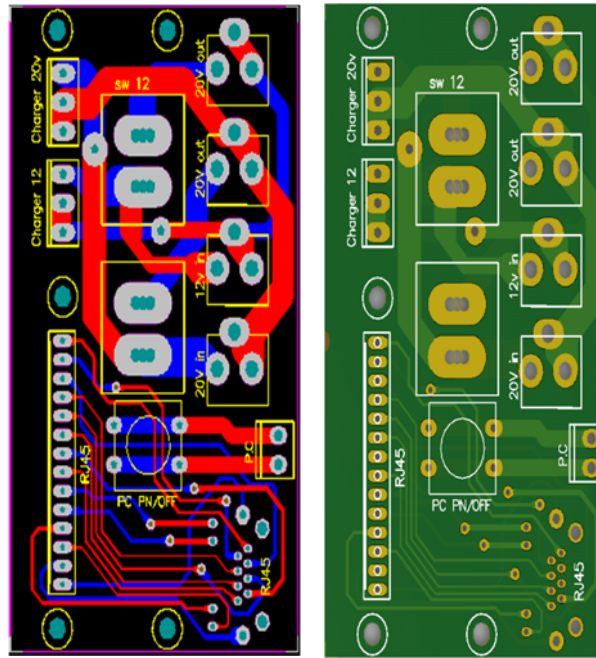
Fig. 4.18: Power distribution board (a), Top view, (b) PCB.

4.6 Access Valve Board

An access valve board has been placed in the access valve chamber to turn the device on, access the information stored in the device, and charge the battery after being drained. This board is made up of two perpendicular boards so that the user can plug the network cable into the charger cable. Also, a micro-switch is embedded in the horizontal board to directly turn on or off the main processing board (Figure 4.19 and Table 4.6). Two headers are used for connecting the input sockets to the on-board battery charger on the vertical board. Figure 4.20 shows the PCB of the vertical board (Table 4.7), which includes two sockets for connecting chargers and a network socket for connecting to an external computer.



(a)



(c)

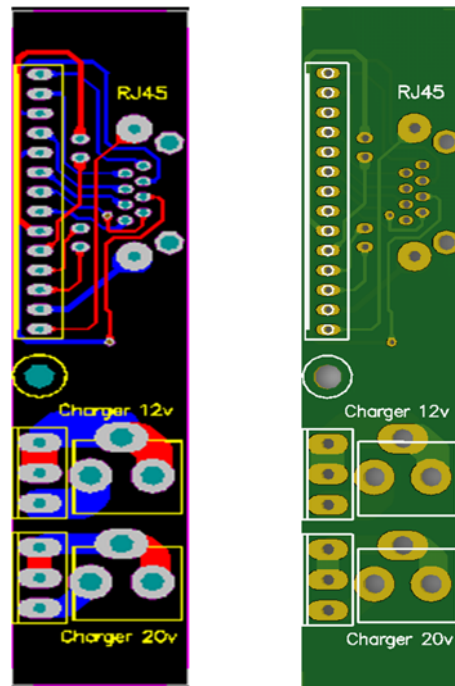
Fig. 4.19: (a) Access valve board, (b) PCB of the horizontal access valve board with dimensions 52.40*87.40*2 mm³.

Table 4.6: Description of the horizontal access valve board.

Quantity	LibRef	Footprint	Designator	Description	Comment
1	PWR2.5	KLD-0202	+1	Low Voltage Power Supply Connector	+12 out
1	PWR2.5	KLD-0202	+2	Low Voltage Power Supply Connector	+20 out
1	PWR2.5	KLD-0202	+3	Low Voltage Power Supply Connector	+12 in
1	PWR2.5	KLD-0202	+4	Low Voltage Power Supply Connector	+20 in
1	con 3_3.9	con 3_3.9mm	charger 1		charger 12
1	con 3_3.9	con 3_3.9mm	charger 2		charger 20
1	Header 14	My_1X14	P1	Header, 14-Pin	Header 14
2	con 2_3.9mm	con 2_3.9mm	power pc1, power pc2		power pc
1	RJ45	RJ45	RJ1		RJ45
2	power switch	power switch	sw1, sw2		power switch

Table 4.7: Description of the vertical access valve board.

Quantity	LibRef	Footprint	Designator	Description	Comment
1	PWR2.5	KLD-0202	Charger 12	Low Voltage Power Supply Connector	charger 12
1	PWR2.5	KLD-0202	charger 20	Low Voltage Power Supply Connector	charger 20
1	con 3_3.9	con 3_3.9mm	charger_12		charger_12
1	con 3_3.9	con 3_3.9mm	charger_20		charger_20
1	Header 14	My_1X14	P1	Header, 14-Pin	Header 14
1	RJ45	RJ45	RJ1		RJ45

Fig. 4.20: PCB of the vertical access valve board with dimensions 17.95*87.40*2 mm³.

4.7 Engines

In order to provide different types of movements to the device we need engines. We used:

- Two Buhler 1.13.044.023.03 24 Volt DC Motors for moving the main body of the device, (Figure 4.21 and Table 4.8) [135].
- One Moons Hybrid Stepper Motor 16HS Series 1.8° /12V DC to move the ball screw and Mass Shifter, its dynamic torque curves shown in Figure 4.22 (Table 4.9) [136], [137].
- Two XQ-S4020D 21.8kg highly waterproof digital servo motors for the horizontal and vertical rotation of thrusters and placing them in a specific angle (Table 4.10) [138].

Table 4.8: General Specifications of Baureihe 1.13.044.XXX for 12v and 24v [135].

Characteristics*	Nenndaten*				
Rated voltage	Nennspannung	U/V	V	12	24
Rated power	Nennleistung	P _N	W	20	20
Rated torque	Nenndrehmoment	T _N /M _N	Ncm	6.0	6.0
Rated speed	Nenndrehzahl	n _N	rpm/min ⁻¹	3200	3200
Rated current	Nennstrom	I _N	A	2.6	1.3
No load characteristics*	Leerlaufdaten*				
No load speed	Leerlaufdrehzahl	n ₀	rpm/min ⁻¹	4000	4000
No load current	Leerlaufstrom	I ₀	A	0.30	0.15

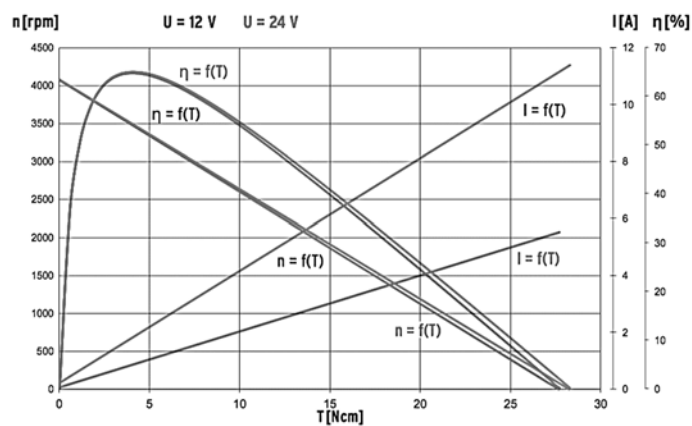


Fig. 4.21: Motor performance at 12 V and 24 V [135].

Table 4.9: General Specifications of 16HS4003-06N and 16HS4007-01N [137].

Series & Length	Model Number	Holding Torque		Rated Current	Resistance per Phase	Inductance per Phase	Detent Torque		Rotor Inertia	
		mNm	oz-in	A	ohm	mH	mNm	oz-in	g.cm ²	oz-in ²
16HS4 36 mm (1.40 in.)	16HS4003-06N	260	36.85	0.40	29	38	15	2.12	30	0.17
	16HS4007-01N	220	31.18	0.65	7	9.6	15	2.12	30	0.17

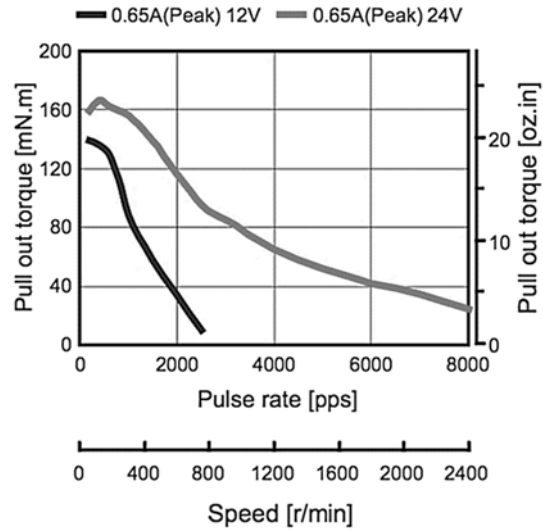


Fig. 4.22: Dynamic torque curves [137].

Table 4.10: General specifications of XQ-S4016D and XQ-S4020D [138].

Idle current(at stopped)	8.5mA	10.0mA
No load speed	0.15sec/60°	0.13 sec/60°
Runnig current(at no load)	180 mA	210 mA
Peak stall torque	17.2 kg.cm / 238.9 oz.in	20 kg.cm / 277.8 oz.in
Stall current	2200 mA	2450 mA

4.8 Engine Driver Board

A single driver is used for controlling all the engines. This driver is responsible for receiving commands in a serial form from the main processor board and via the sensors board, and controlling the direction and speed of all engines. It includes two BTN7971B motor driver

modules for control of DC motors [139]. This board produces the PWM signals required by all engines via a mega64 microcontroller (Table 4.11). Block diagram and Schematic of the engines driver board is shown in Figure 4.23 and PCB of the top and the BTN7971B Motor Driver Module are shown in Figure 4.24.

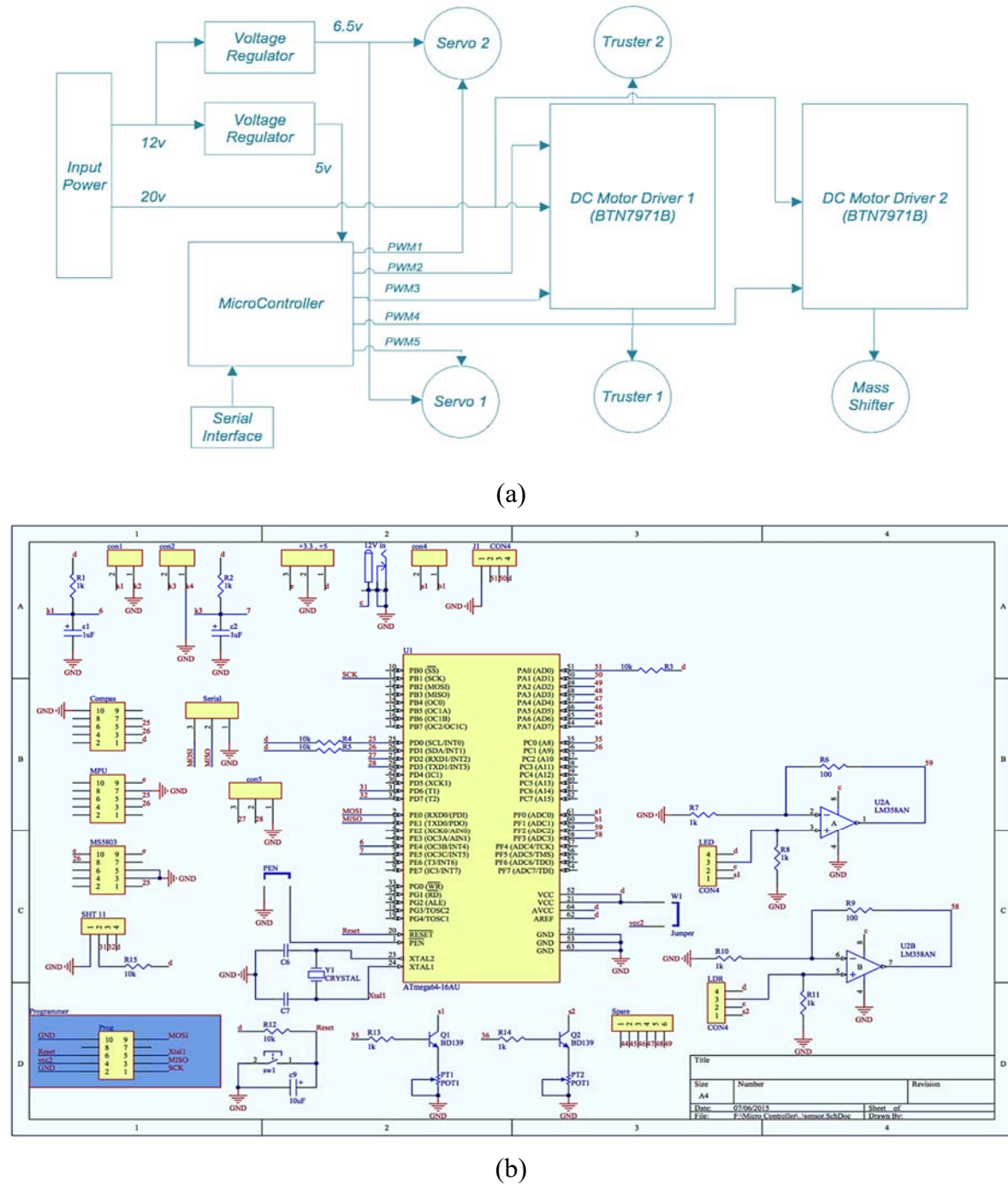
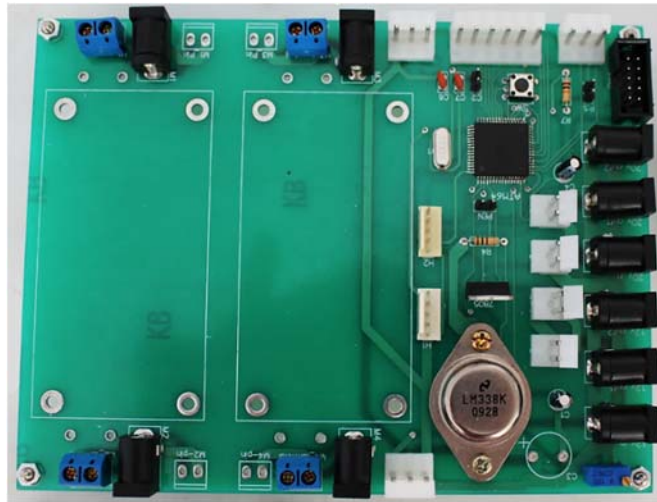
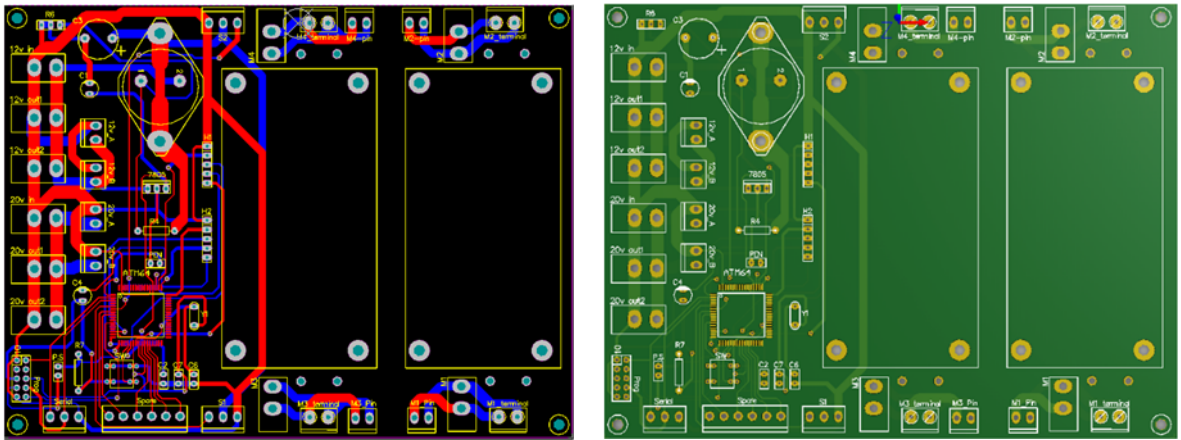


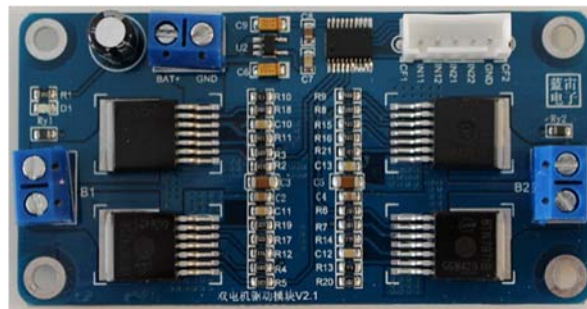
Fig. 4.23: a) Block diagram and b) Schematic of the engine driver board.



(a)



(b)



(c)

Fig. 4.24: Engine driver board with dimensions 157.50*120*2mm (a) Top view, (b) PCB, (c) BTN7971B driver module.

Table 4.11: Description of the Engine Driver Board.

Quantity	LibRef	Footprint	Designator	Description	Comment
3	con 3_3.9	con 3_3.9mm	+3.3 , +5, con5, Serial		con
1	PWR2.5	KLD-0202	12V in	Low Voltage Power Supply Connector	12V in
2	cap10u	cap10u	c1, c2		1uF
1	cap10u	cap10u	c9		c?
2	Cap ceramic	My_cap100n	C6, C7	CAP-100Nf	Cap
4	WE-2.54 mm IDC & Transition Connector 61201022323	61201022323	Compas, MPU, MS5803, Prog	WR-BHD 2.54mm Board To Cable Transition Connector	WE-2.54 mm IDC & Transition Connector 61201022323
3	con 2_3.9mm	con 2_3.9mm	con1, con2, con4		con
4	CON4	MT6CON4V	J1, LDR, LED, SHT 11	Connector	CON4
2	Jumper	My_IDC-2	PEN, W1	Jumper Wire	Jumper
2	POT1	POT1	PT1, PT2	Variable Potentiometer standing	POT1
2	BD139	SOT-32	Q1, Q2	NPN Transistor	BD139
8	Res	My_axial-0.4	R1, R2, R7, R8, R10, R11, R13, R14	resistor 1/4 wat	1k
5	Res	My_axial-0.4	R3, R4, R5, R12, R15	resistor 1/4 wat	10k
2	Res	My_axial-0.4	R6, R9	resistor 1/4 wat	100
1	CON6	MT6CON6V	Spare	Connector	CON6
1	Tac switch	MY TACTILE-PTH	sw1	small tac switch	Tac switch
1	ATmega64-16AU	64A_N	U1	8-bit AVR Microcontroller, 4.5-5.5V, 64KB Flash, 2KB EEPROM, 4KB SRAM, 64-pin TQFP, Industrial Grade (-40°C to 85°C), Pb-Free	ATmega64-16AU
1	LM358AN	DIP8	U2	Low-Power Dual Operational Amplifier	LM358AN
1	CRYSTAL	My_Crystal	Y1	Crystal	CRYSTAL

4.9 Conclusion

In this chapter, the electrical parts embedded in the AUV have been discussed in detail. A main processor as well as customized boards for each sensor, were designed with the objective to reduce electronic connections and, consequently, to optimize the interior space. Further, others boards such as power, access, and motors driver boards were also designed.

Finally, two cameras as well as a transmitter system were implemented as part of the whole AUV vision and control block.

CHAPTER 5

OBJECT DETECTION AND TRACKING

5.1 Introduction

Trackers are inherently facing with a string of random images that change over time. Although most existing algorithms can track objects in a controlled environment, they fail when drastic changes happen in target appearance or ambient light.

One reason for this failure is that these algorithms use fixed apparent models for the target. These models are taught by the object's appearance data before starting the tracking process, thus limiting the apparent range of the modeled target and ignoring the amount of information (such as deformation and lighting conditions) that exists at the time of tracking [140]-[144].

It is then crucial to first review the performance of existing methods and then, based on our particular application (underwater environment with frequent changes in ambient light and image resolution), to select the most suitable approach.

5.2 Object Tracking Methods

5.2.1 Object Based Detection Method

The first class of methods is based on machine learning and need training [140]. It converts the object-tracking problem to object detection for each frame. Its main limitation is in the fact that it cannot detect an angled view of the object (a full object image (front view) should be used). Note that we can use a colored model of the object to cope with this issue, but still this solution cannot be suitable when light is changing or when we have occlusion [141].

5.2.2 Color-Based Methods

The second class uses the color characteristics for tracking. Therefore, the object should be determined manually or automatically by a detector to start the process (usually based on a color histogram to describe it). Two of the most popular approaches are Mean Shift and camshaft. Ning *et al.* [142] used textural features extracted by Local Binary Patterns (LBP) technique in addition to the color histogram, thus providing a better description of the target. Shearer *et al.* [145] combined two tracking methods to achieve a more robust tracking. Nevertheless, they cannot be applied in presence of light changes or occlusion [145].

5.2.3 Feature-Based Methods

The last class of tracking systems uses object's features for tracking. Iraj *et al.* [144] have presented a color-independent method that uses the morphological characteristics of the object and edge detector to locate it. This class is better than the other two but is computationally expensive since we have to detect features in every frame [143].

5.2.4 Overall Performance of the Above Mentioned Methods

Although most existing algorithms can track objects in a controlled environment, they fail when drastic changes happen in target appearance or ambient light. One reason for this failure is that most algorithms use fixed apparent models for the target. These models are taught by the object's appearance data before starting the tracking process, thus limiting the apparent

range of the modeled target and ignoring the amount of information (such as deformation and lighting conditions) that exists at the time of tracking.

5.3 Proposed Method

The main challenge in tracking is to interact with the target variations as in our application. The inherent variations of the object include changes in the mode and form along with the non-inherent changes in lighting, camera motion, camera angle and occlusion that lead to many changes in the appearance of the object. Selecting the most suitable technique mainly depends on the targeted application. Therefore, we used in this work a combination of color and texture histograms to trace the target and applied it to the *Mean shift* context. Unlike the conventional methods for extracting the objective characteristics, we used the LBP (Local Binary Pattern) technique to describe the target. The proposed method effectively extracts corners and edges in the target area, so that a more resilient and better description of the target is obtained. It exhibits significant advantages over the above mentioned methods since it is computationally inexpensive and works when the target is very similar to its background.

5.3.1 Motion-Based Segmentation Methods

One of the most important tools in tracking is motion analysis [146]. Tracking might be based on the detection of the feature points of the region or the two or three- dimensional models. One of the important points in tracking moving objects is the partial/complete overlapping. In the partial overlap, a part of an object is covered and, in the complete overlap, the object is completely hidden behind another object. In partial overlap, instead of tracking the whole object, its features are tracked.

Two essential issues in motor-based machine vision systems such as video surveillance or monitoring, are motion segmentation and tracking moving objects. The purpose of the segmentation is:

- to detect and segment the moving object in an image,
- remove noise and shadow,
- connect the areas related to the object.

The output of the motion segmentation is the current frame binary image. Then, in the tracking stage, the correspondence between the segmented objects in the current frame and the objects in the previous frame is established.

Also in this stage the motion and appearance models of objects are updated. The unit output is tracking the path taken by the objects and their features.

5.3.2 Detecting Moving Objects: Background Subtraction Methods

The most common methods for the detection of moving objects in an image are the background subtraction methods [147]. Among such methods, even if temporal averaging and simple alpha combination are relatively fast and simple, they are not effective for complex images that contain many moving objects especially if these moving objects move smoothly [147]-[148]. Methods based on Gaussian modeling are, on the other side, capable of dealing with change in brightness, and fluctuations caused by minor repetitions in the background objects and small camera movements. However, although their model is resistant to changes in light intensity, such methods are still subject to limitations regarding background.

To address this problem, the multi-colored comparative model can be used to deal with effects caused by minor changes associated with leaves of trees, bushes, flags or water level [149]. Stochastic modeling such as Markov random field or distribution of Gibbs and Bayesian inference are used to separate complex images and textures. It is equivalent to Gibbs distribution in terms of association with energy minimization problem. Most of the stochastic methods need the probability density estimation. Most of them are powerful in function, although most of them are computationally intensive. Apart from the use of models, there are still methods non-based on the model, which are widely used especially for those which do not need supervision.

5.3.3 Mean Shift

Among statistical methods, mean shift has the advantages to include its statistical properties and its relationship with density estimates. What is of interest, are the use of local conditions instead of density distributions. Mean shift method is an accurate method for estimating local

conditions without the need to estimate the density function. Image and video processing based on Mean shift acts on the principle that pixels in a given region share similar states. Given the properties of an image, different environments with different similarities, such as uniformity in the intensity of colors and textures, can be estimated. Recent studies have shown that Mean shift is associated with dual filter and non-linear diffusion [150],[151].

Mean shift method in video process can be achieved based on some key frames: color, shape, and position, although spatial break followed by timing track may be preferred. Although Mean shift section break is suitable for videos, it requires high-CPU time. If we consider the probability density function independently of the image sequence or spatial region, the joint probability density function can be used for three-dimensional (3D) video size instead of two-dimensional (2D) probability density functions. Therefore, Mean shift section break of video volume can be simplified and an enhanced Mean shift filter with common time spatial range for video section break can be defined. The coordinated state combination of inner frames and the estimation of outer frame state in the same spatial and time ranges, enhance separating animated objects [152]-[155].

One of the methods to identify images is using a probability density function that may have many states and unknown forms. A non-parametric method may be used to estimate the probability density function. In many applications, we can estimate only the relations of the local state of the function or the density gradient instead of the precise measurement of the density function. Mean shift method is a very effective method to find the density estimation states without estimating the function.

Kernel selection is a fundamental issue in Mean shift. In image processing applications, filtering based on Mean shift is used mainly as a smoothing filter by preserving the edges. In this case, the filter can be visualized as a nonlinear low-pass filter. Two types of kernels are used to calculate the middle transmission. The Epanechnikov kernel, which can be obtained from the minimum mean square error, and the Multivariate Gaussian kernel, a shadow of the Gaussian kernel. To reduce calculations, a kernel with an infinite length like the Gaussian kernel is often used to transfer a kernel with limited support. If a short length is perfect for kernel bandwidth, truncation will be trivial.

A non-linear filtering can be also used instead to find the density state because there is a direct relationship between the dual filter and the filter based on Mean shift. A filter can be relevant with a range or a domain filter, or a dual filter, depending on the property vector settings. One of the major applications of the Mean shift filter is smoothing images by preserving edges.

Since the Mean shift segmentation method only uses the film or image features obtained by the pixel level, there will be a gap between the segmented area and the concepts of the high level of the image of the human conception. Therefore, it is expected that all segmentation methods are based on low level image information. The mean shift segmentation advantages include obtaining the regions while maintaining the edges, obtaining data that are very sensitive to the primary process and at lower cost than the other methods.

5.3.4 Color-Texture Histogram

Quick tracking of objects is considered as an essential issue in image processing applications and machine vision. We can be facing with many issues such as noise, image blocking, flicker or drastic changes in the background or foreground for tracking of objects in video images. Many algorithms have been proposed to address them. As mentioned, mean shift is one of the most popular because of its simplicity and efficiency. Also, the most common way to display and describe an object or intended target in an image is to use histograms, considered as a density function of the target area. This approach is relatively strong to describe the appearance of objects. However, using only histogram in mean shift can raise some issues: one of them being that the target location information can disappear or, if the target has an appearance close to background, it cannot be well recognized. Therefore, additional features such as gradients or edge information can be used in alongside the histogram to resolve this problem [156].

Texture patterns represent the spatial structure of an object in the image. They are a proper feature to identify and display a target. Image texture features provide more and better data against image color histogram. Hence, the simultaneous use of color and texture histogram of image can increase the capacity of tracking confidence, particularly in complex images. But effective use of both texture and color features together is still a challenge [157].

In this work, the *Local Binary Patterns (LBP)* approach was thus used mainly because it is an effective and efficient technique to describe the texture characteristics of image with high computational features and unchangeable speed with rotation. It has been applied in a variety of areas including image texture analysis, facial recognition, image segmentation, etc. More recently, LBP pattern has been also used to track objects.

5.4 Describing the Target

A target is usually characterized by a rectangle in the image. In most existing methods, the image color histogram is used for tracking the target in describing the target. In [206], an enhanced method based on color-texture histogram is used. It displays the location of pixels in the target area as $\{x_i^*\}_{i=1\dots n}$ and considers the center of target area as the origin. Target model $\hat{q}$ corresponding to target area is calculated as follows [157],[158]:

$$\begin{cases} \hat{q} = \{\hat{q}_u\}_{u=1\dots m} \\ \hat{q}_u = C \sum_{i=1}^n k(\|x_i^*\|^2) \delta[b(x_i^*) - u] \end{cases} \quad (5.1)$$

in which $\hat{q}_u$ represents the probability of feature u in model $\hat{q}$, m is the number of feature space dimension, δ the impulse function and $b(x_i^*)$ relates x_i^* to the histogram bin, $k(x)$ is the kernel function and C is a constant representing the normalized function defined as

$$C = 1 / \sum_{i=1}^n k(\|x_i^*\|^2) \quad (5.2)$$

Similarly, the nominated target model corresponding to the candidate area is calculated as follows:

$$\begin{cases} \hat{p}(y) = \{\hat{p}_u(y)\}_{u=1\dots m} \\ \hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \end{cases} \quad (5.3)$$

and

$$C_h = 1 / \sum_{i=1}^{n_h} k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right) \quad (5.4)$$

in which $\hat{p}_u(y)$ represents the probability of feature u in model $\hat{p}(y)$ and $\{x_i\}_{i=1 \dots n_h}$ represents the pixel location in the target candidate region with center y , h is the bandwidth and the constant C_h is the normalized function. We introduced the following measure to calculate the likelihood of the target candidate model $\hat{p}(y)$ and target model $\hat{q}$:

$$\rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \hat{q}_u} \quad (5.5)$$

Then the distance between $\hat{p}_u(y)$ and $\hat{q}$ is calculated as follows:

$$d[\hat{p}(y), \hat{q}] = \sqrt{1 - \rho[\hat{p}(y), \hat{q}]} \quad (5.6)$$

5.4.1 Tracking with Mean Shift

From the above, minimizing equation (5.6) is equivalent to maximizing equation (5.5). This iterative optimization process starts using initial values of the target location in the previous frame. Using the Taylor series around $\hat{p}(y_0)$, the linear approximation of (5.5) is

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u} + \frac{1}{2} C_h \sum_{i=1}^{n_h} w_i k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right) \quad (5.7)$$

with

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \delta[b(x_i) - u] \quad (5.8)$$

The first term in (5.7) is independent of y , so the objective is to maximize the second term in order to minimize (5.6). In this iterative process, the estimated target changes its location from the location y to a new location y_1 given by:

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g \left(\left\| \frac{y - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n_h} w_i g \left(\left\| \frac{y - x_i}{h} \right\|^2 \right)} \quad (5.9)$$

When we use the Epanechnikov kernel, (5.9) will reduce as follows:

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i} \quad (5.10)$$

Finally, the closest area to the desired object can be selected by using (5.10) in each new frame.

5.4.2 Local Binary Patterns

Through thresholding, the neighboring pixels with central pixel, LBP operator allocates a number to each pixel; this operator can be described as [157]:

$$\text{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (5.11)$$

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

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (5.12)$$

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

$$\text{LBP}_{P,R}^{\text{riu2}} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{if } U(\text{LBP}_{P,R}) \leq 2 \\ P + 1 & \text{otherwise} \end{cases} \quad (5.13)$$

with

$$\begin{aligned} U(\text{LBP}_{P,R}) &= |s(g_{P-1} - g_c) - s(g_0 - g_c)| \\ &\quad + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \end{aligned} \quad (5.14)$$

5.4.3 Target Description by Color-Texture Histogram

One of the LBP weaknesses is that when the changes in color intensity are very low in certain regions, e.g., in flat areas [157]. A way to address this issue is to change the term $s(g_p - g_c)$ in (5.11), (5.13), and (5.14) with $s(g_p - g_c + a)$; thus for a higher $|a|$, the amount of color intensity changes will be greater without significantly changing the result of thresholding.

In our approach, we used the LBP operator with 8 neighbors with a radius equals to 1 (Figure 5.1). So, we calculated the value of LBP for each point by using the above method, which is between 0 and 9. In fact, model $\text{LBP}_{8,1}^{\text{riu2}}$ has 9 uniform structures: points, flat areas, edges, end of line, and corners that can be found by 9 patterns shown in Figure 5.2. Each of these nine structures is called micro-texton. In Figure 5.2, black and white circles represent 0 and 1 respectively.

Note that this approach alone does not improve the tracking practice by mean shift compared to the conventional method of the target description based on color [142],[157]. This becomes even more evident when the target is not that different of its background. So, we still have to seek a better solution to combine the color and LBP properties.

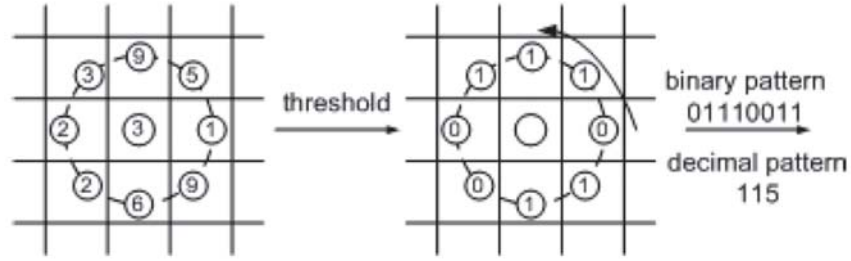


Fig. 5.1: An example of LBP operator with $P = 8$ and $R = 1$ [158].

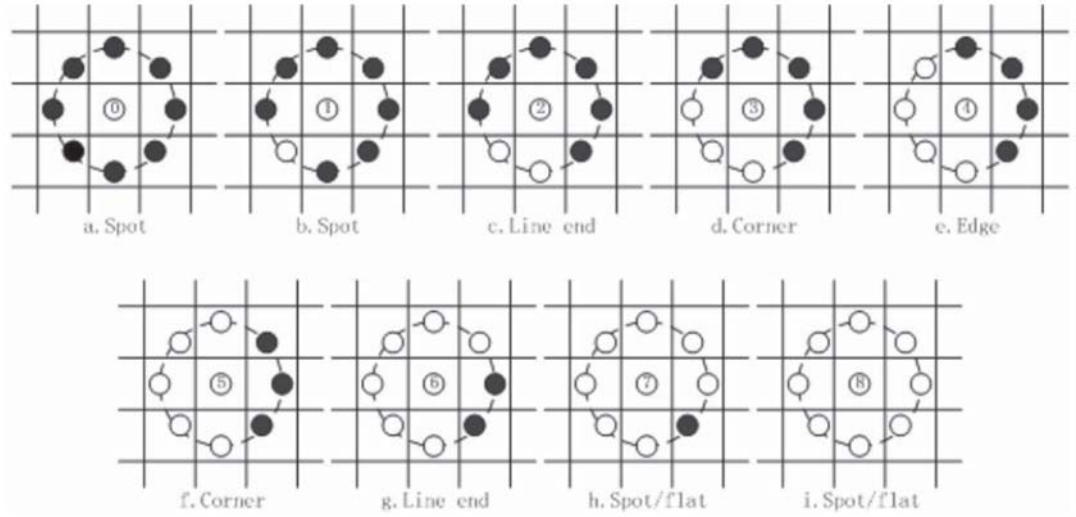


Fig. 5.2: 9 Flat patterns in the model $\mathbf{LBP}_{8,1}^{riu2}$ [142].

In describing the target, the micro-textons at the end of lines and corners, as flat patterns of the majority, represent the main characteristics of the target while, on the other hand, flat points and areas, as uniform patterns of the minority, are quantitative textures. In general, majority patterns are more important than the minority ones in describing the target, so we need to extract the majority patterns using the following formula [157].

$$\mathbf{LBP}_{8,1}^{riu2} = \begin{cases} \sum_{p=0}^7 s(g_p - g_c + a) & U(\mathbf{LBP}_{8,1}) \leq 2 \text{ and} \\ \sum_{p=0}^7 s(g_p - g_c + a) \in \{2, 3, 4, 5, 6\} & \\ 0 & \text{otherwise} \end{cases} \quad (5.15)$$

Patterns No. 0, 1, 7, and 8 are related to uniform patterns of the minority, No. 9 is related to non-uniform pattern, and No. 2 to 6 are related to original uniform patterns. Equation (5.15) classifies the minority patterns in the category of non-uniform patterns. Then, these patterns are used to describe the target. Originally, it can be said that we initially used the (5.15) as a mask and then we modeled the target by using the color features and LBP inside this mask. Using the model provided, only pixels of the desired object which have been extracted by Equation (5.15) is used to describe it. In this report, these points combine the color histogram of the image with LBP histogram and model the target. This method eliminates the flat backgrounds very well and also remove the noise interference effect; on the whole, this method extracts the main features of the image.

Figure 5.3 shows a test example. Figure 5.3 (a) shows the target area in consecutive frames and Figure 5.3 (b) shows the result of (5.15) after applying the mask in which the non-black pixels represent key features of the image.

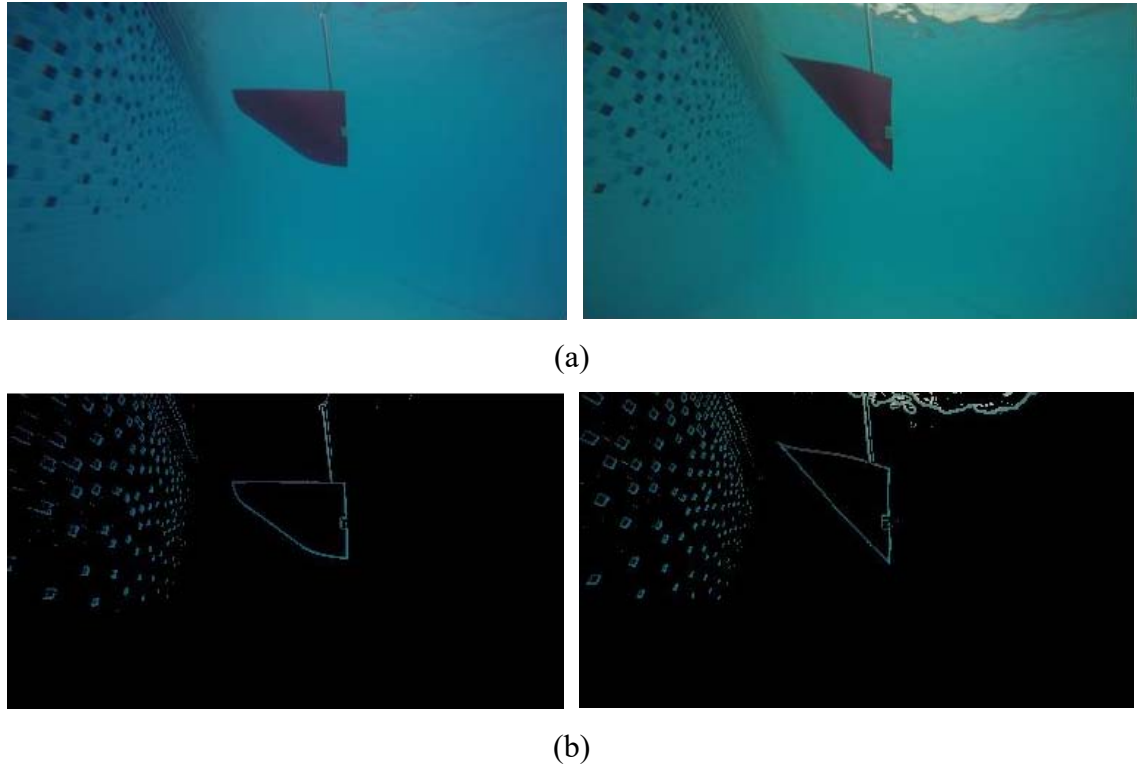


Fig. 5.3: (a) Tracking windows, (b) Mask extracted by uniform patterns of LBP majority.

5.4.4 Tracking by Color-Texture Histogram

The target is then described by using the RGB channel and LBP pattern extracted through (5.15) and applied in mean shift. To determine the distribution of texture and color in the target area, (5.1) was used. Then, the target candidate model $\hat{p}(y)$ was calculated using (5.3). The corresponding algorithm can be summarized as follows:

Input: Target model ($\hat{q}$) and its location (y_0) in the previous frame.

1. Initialize the number of iterations ($k \rightarrow 0$)
2. Set the threshold value ε and the maximum number of iterations N
3. Determine the target candidate model $\hat{p}(y)$ in the current frame
4. Calculate the weights w_i using (5.8).
5. Calculate the new target location (y_1) using (5.10).
6. Increment $k = k+1$ with $d = \|y_1 - y_0\|$ and $y_0 = y_1$. if $d < \varepsilon$ or $k \geq N$, go to step 7. Otherwise, go to step 3.
7. Recall the next frame as the current frame and go to step 3.
8. If $k = N$ or if last frame is reached, then end.

It should be noted that, at this stage, the target location should be manually adjusted in the first frame. This means that if we are want to track an object in a video, we should introduce this location into the system in the first frame. To automate the tracking process, we introduced an enhanced method by using the derived image and identified the desired object through it so that it can be used as input for the tracking algorithm.

Edges determine the borders; therefore, they are considered as an essential parameter in image processing. Edges are areas with strong contrast of color intensity in the image. In other words, we have a mutation in color intensity of a pixel which is not edge from a pixel which is edge. Edge detection in the image means reducing non-useful information from the data, while maintaining important structural features. One way to edge detection is the use of derivative operator [159],[160].

It is clear that the derivative shows a maximum located on the edge center in the original signal. This method of edge detection is known as gradient filtering. If the value of gradient

is greater than a threshold, that pixel determines the location of an edge because edge pixels have greater intensity values compared to neighboring pixels. Note that when the first derivative is maximum, the second one is zero. So, another way to find edges is the use of second derivatives (Laplacian). Also, before detection, the noise should be removed from the original image. To do this, a median filter with a simple mask can be used. Figure 5.4 shows the original image, which is a frame of a video provided by the robot under the water and the object seen in the image. Figure 5.5 shows the result after applying the derivative operator. Also, we used a window of size 3×3 so that the information does not reduce significantly. Figure 5.6 shows the result of applying this method on Figure 5.4.

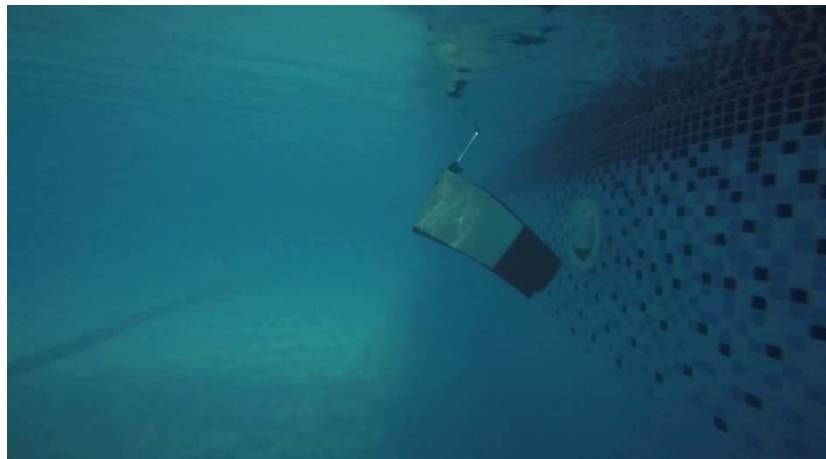


Fig. 5.4: The original image captured by the robot camera.

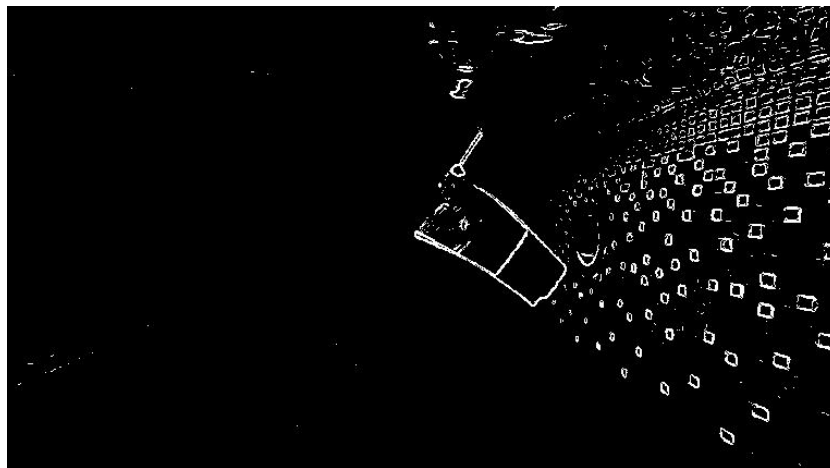


Fig. 5.5: Result after applying the derivative operator.

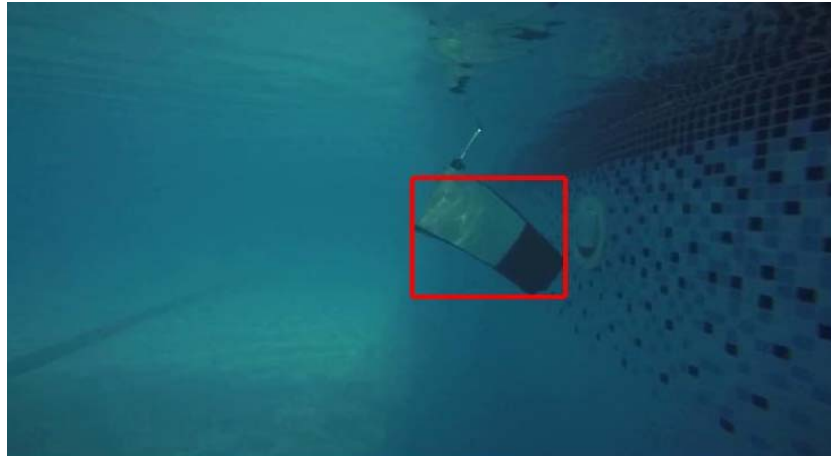


Fig. 5.6: Detecting the desired object in the image.

5.5 Results

To demonstrate our approach, let us call the method presented in equation (5.10), i.e. the one using only the color histogram, as M1, by M2 the method using both the color histogram and the LBP (Equation (5.11)), and by M3 the method combining the color model, the LBP, and the mask of majority pattern.

We tested the underwater robot movement with all these three methods and found that the number of iterations necessary to converge in the mean shift algorithm was 5, 4, and 2, for respectively M1, M2, and M3. Since the target area is very different from the background in terms of color, the target can be tracked with high precision. Figure 5.8 shows the result of this test while Table 5.1 summarizes the values of average and standard deviation in local error as well as tracking speed based on frames per second for the three methods. Results show that M3 has the lowest values of average and standard deviations. In addition, it can be seen in Table 5.2 that although the computational complexity has increased in method M3, its tracking speed is better than M1, probably due to the smaller number of iterations needed to converge. Because they are converged earlier. So, method M3 is more precise and reliable than the two other methods.

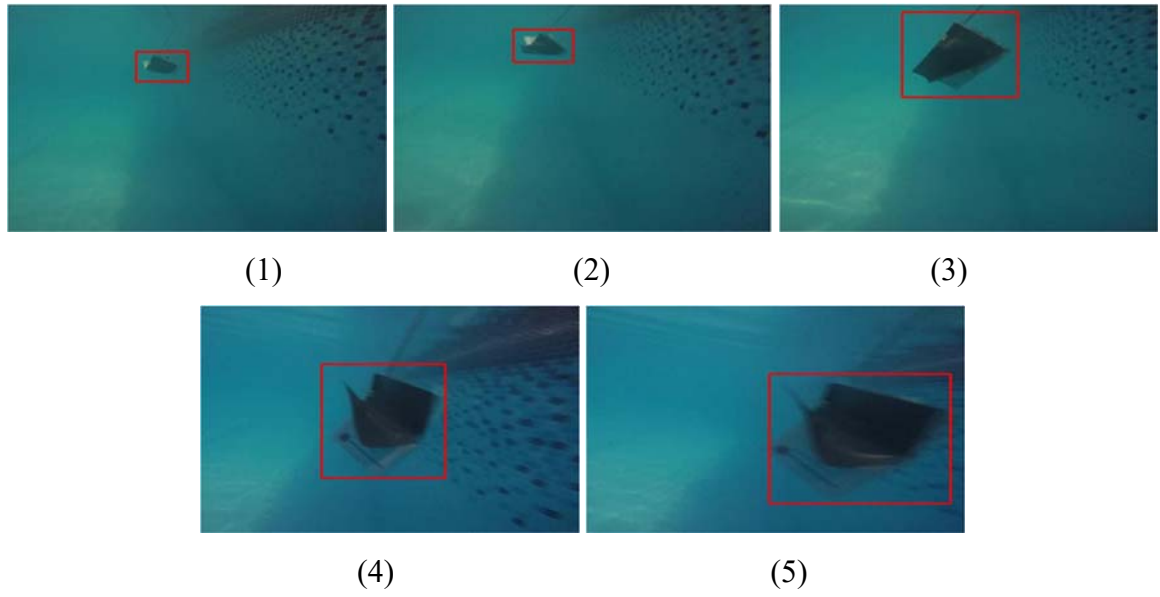


Fig. 5.7: Result of the test on underwater video with the proposed method (From (1) to (5), the pictures show the robot approaching the obstacle).

Table 5.1: Assessment of accuracy and speed in tracking the target by the three available methods.

Method	Majority pattern mask (M3)	Color histogram and LBP (M2)	Color histogram (M1)
Average error (pixels)	3.01	11.3	7.40
Standard deviation (pixels)	4.2	12.4	8.10
Speed (frame/s)	110	80	121

Table 5.2: Number of iterations in mean shift algorithm until there is no blank space left unlabeled.

Video	Number of repetitions average	Number of repetitions	Number of frames
Underwater	2.72	155	58

5.6 Conclusion

In this chapter, an enhanced method to describe the target was presented in which both color and LBP characteristics were combined and then, a new tracking algorithm was built by using these features. This led to reduction of computational costs and increasing in accuracy of target display. A mask was made from the target by using the majority uniform patterns and then, the target was described by using color texture within the mask. The description of the extracted target showed the important features such as corners and edges well and was able to remove the background effect. The experimental results of the tests demonstrated the proposed method with less number of iterations and higher quality than existing approaches.

CHAPTER 6

CONTROL SYSTEMS

This chapter details the algorithms implemented to control the AUV movement and the experimental tests performed to validate them.

6.1 Introduction

The designed vehicle has five degrees of freedom for moving,

- Horizontal/Vertical movements.
- Rotations around the x-axis, the y-axis, and the z-axis.

Different controlling algorithms were then implemented to control the AUV's speed (associated with the device horizontal movement) and depth (corresponding to the device vertical movement), as well as Pitch, Roll and Yaw angles. However, we need to first review the procedure we used to determine the controller parameters. We will focus on the PID controller, knowing that all other types of controllers (P, PI, PD) can be deduced from this generic one.

6.2 Controller Design

Determining the PID controller coefficients is usually based on both the user experience and the knowledge of the system. But apart from this, the Ziegler – Nichols method is one of the most common approaches to estimate the PID controller values [161]. In this method, the system is a closed-loop and a simple proportional controller is considered in the loop. Then the system gain is increased by the proportional controller to the point where the system reaches the unstable level and starts oscillating. This gain is called KU and the corresponding period is noted TU.

Let us consider the closed loop controller input as:

$$u(t) = KP \left(e(t) + \frac{1}{TI} \int_0^t e(\tau) d\tau + TD \frac{de(t)}{dt} \right) \quad (6.1)$$

where $KI = \frac{KP}{TI}$ is the controller integral coefficient. $KD = KP \cdot TD$ and KP are the proportional controller coefficients. From this general formulation, one can determine the P, PI or PD controller coefficients according to Table 6.1.

In other words, the open-loop Ziegler-Nichols method can be described as follows. First the derivative and integral blocks should be separated from the circuit and the proportional block (equivalent to P) should be set in the controlled circuit block.

Table 6.1- PID Controller values from the Ziegler-Nichols method [161].

Control Type	KP	TI	TD
P	0.5KU	-	-
PI	0.45KU	TU/1.2	-
PD	0.8KU	-	TU/8

Removing other controlled factors means considering TI as infinite and TD as zero. It should be noted that the main PID control equation in both cases is:

$$G(s) = KP \left(1 + \frac{1}{TIs} + TDs \right) \quad (6.2)$$

Then, the step input is applied starting with small values of K_p until the output graph becomes oscillating. In this case, the period is T_u and the obtained oscillator gain is K_u . Thus, given the estimates of K_i and K_p , it is possible to find the other parameters depending on the type of controllers we used.

6.2.1 Speed Controller

This controller adjusts the speed of the left and right engines to provide the required speed depending on the programmed route. For this part, a proportional integral (PI) controller has been used, which output can be set as [66]-[68],[162]:

$$TS = kPAU \times e_u + kSU \int e_u \quad (6.3)$$

with $kPAU$ the speed controller proportional coefficient and kSU the speed controller integral coefficient. TS is expressed in terms of engine revolution percentage and e_u the speed error, calculated as:

$$e_u = usd - use \quad (6.4)$$

with usd the desired speed and use the speed measured by the sensors.

6.2.2 Pitch Controller

A mass moving system was designed to control the pitch angle, i.e., the x-axis towards the horizon. The intended mass moves only towards the longitudinal direction of the body (the

x-axis) while creating positive or negative angles towards horizon. Due to its inherent stability, the device tends to keep the pitch angle close to zero.

This means that if the instrument is out of balance, it tends to return to its steady state, the zero angle. Moreover, to dive or rising, it is desirable that the angle of the nose being changed upwards or downwards to decrease the body angle with its path, thus decreasing the friction with the fluid, and therefore, increasing speed.

For this purpose, the pitch controller adjusts the moving poise close to the center of balance to make the nose angle zero, when the device moves horizontally and if the device tends to change its depth upwards or downwards it replaces the moving poise backwards or forwards, respectively [163]-[165].

A proportional controller has been used to adjust the nose angle as:

$$TP = kPB_p \times e_p \quad (6.5)$$

where TP is the controller output, which includes the poise distance towards the center of balance. kPB_p is the controller proportional coefficient and e_p is the nose angle error between the desired pitch angle θ_d and the pitch angle θ_{sen} measured by the sensors:

$$e_p = \theta_d - \theta_{sen} \quad (6.6)$$

Note that preliminary simulations showed that there is no need for an integrator term.

6.2.3 Depth Controller

The depth controller maintains the device at the desired depth. It has two modes, at rest and during movement:

- *Depth Controller at rest:* If the AUV is at rest, the speed is zero and the angle TH_1 relatively to the horizon is set automatically by the servos to 90°. The depth is

maintained at the desired value by adjusting the level of engines' revolution. For this, a proportional integral controller has been used [66]-[68],[164]:

$$TH_1 = kPC_{D_1} \times e_D + kRD_{D_1} \int e_D \quad (6.7)$$

with kPC_{D_1} the depth controller proportional coefficient at rest and kRD_{D_1} the depth controller integral coefficient at rest. TH_1 is the controller output determined by the engines' revolution in terms of percentage; e_D is the depth error calculated as:

$$e_D = lD_d - Dse \quad (6.8)$$

where lD_d is the desired depth and Dse is the measured depth by sensors.

To find the control parameters, we started with a P controller. First, the system critical gain was obtained ($KU=611$) and the related coefficients deduced. However, with the P controller, the system did not respond correctly (Figure 6.1). Therefore, a low integral controller value was added to the system to slightly increase the response speed of the system. Finally, the values $kPC_{D1}=300$ and $kRD_{D1}=0.1$ were retained for the PI controller to get the response shown in Figure 6.2.

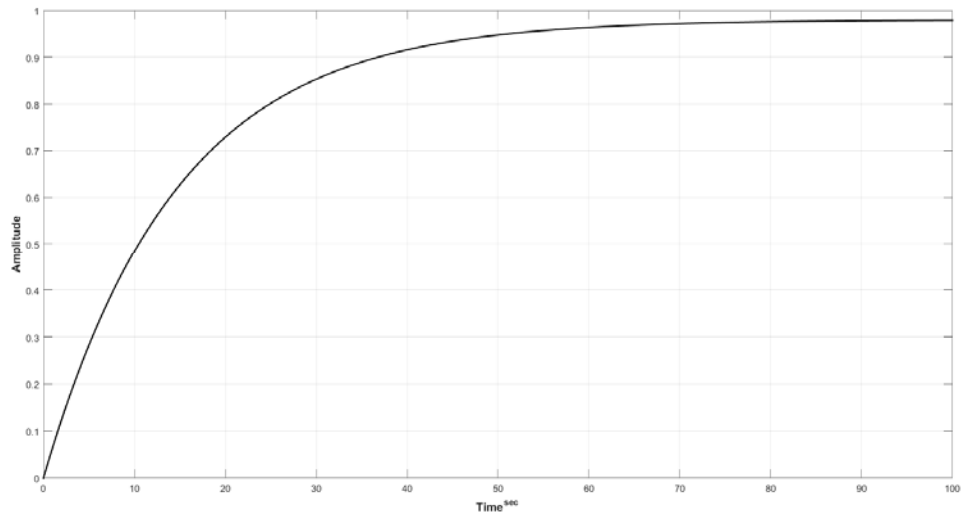


Fig. 6.1: System response to the proportional controller.

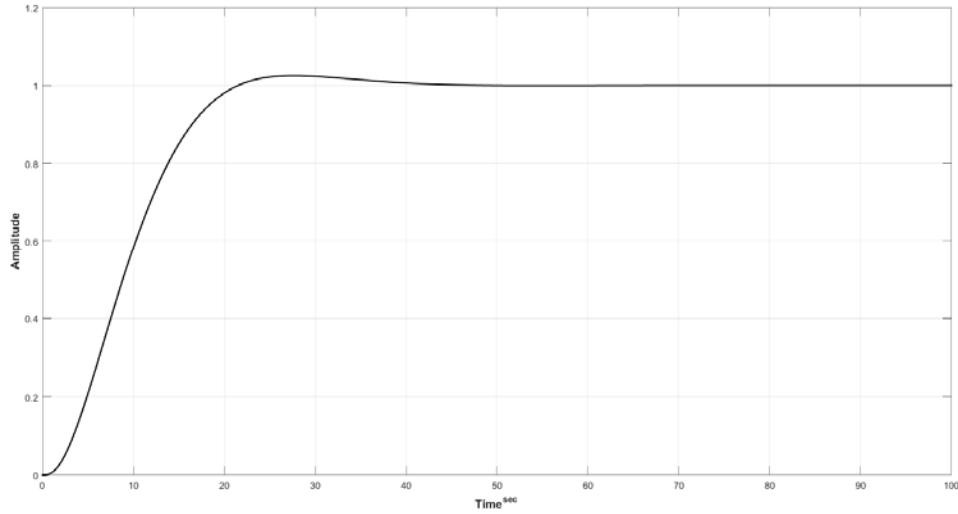


Fig. 6.2: System response with proportional-integral controller.

- *Depth controller during movement*: when the speed is not zero, the TH_2 parameter is used to control the depth by changing the engines' angle towards horizon and the required vertical force is provided. For this purpose, a proportional integral controller has been used with [68]-[68], [164].

$$TH_2 = kPC_{D_2} \times e_D + kRD_{D_2} \int e_D \quad (6.9)$$

with kPC_{D_2} the depth controller proportional coefficient during movement and kRD_{D_2} the depth controller integral coefficient during movement. TH_2 is the controller output that represents the engines' angle towards horizon.

Here the system critical gain is $KU=194$ leading to a proportional controller coefficient of $kPC_{D_2} \approx 97$.

6.2.4 Level Controller

The logic of controlling the AUV direction is based on the difference between the left and the right servo's angle. For example, if the desired rotation is towards right, the left servo is

placed upper than the right servo and becomes closer to horizon. Similarly, if the desirable rotation of the device is towards left, the right servo comes up and becomes closer to horizon. To implement this logic a proportional-integral controller has been used with [66], [68]:

$$TL = kPC_y \times e_y + kLC_y \int e_y \quad (6.10)$$

Here kPC_y is the proportional coefficient of the controller and kLC_y its integral coefficient. TL is the controller output, showing the required difference between the servo motors, e_y is the direction error given by

$$e_y = y_{l_d} - y_{l_{sen}} \quad (6.11)$$

where y_{l_d} is the desired direction angle and $y_{l_{sen}}$ the direction angle determined by the sensors. According to the Ziegler-Nichols method, we got a critical gain $KU = 0.71$, leading to $KPC_y = 0.32$ and $TU = 35s$. Here, since the system is slow and the period high, the above value was adjusted to $KPC_y = 0.35$. Next, an integral part was added to obtain $KPC_y = 0.3$ and $kLC_y = 0.001$.

6.3 Experimental Tests

6.3.1 Experimental Setup

The above algorithms were first implemented and tested separately. Then, the AUV was built with all required mechanical and electrical blocks (Figures 6.3 and 6.4).

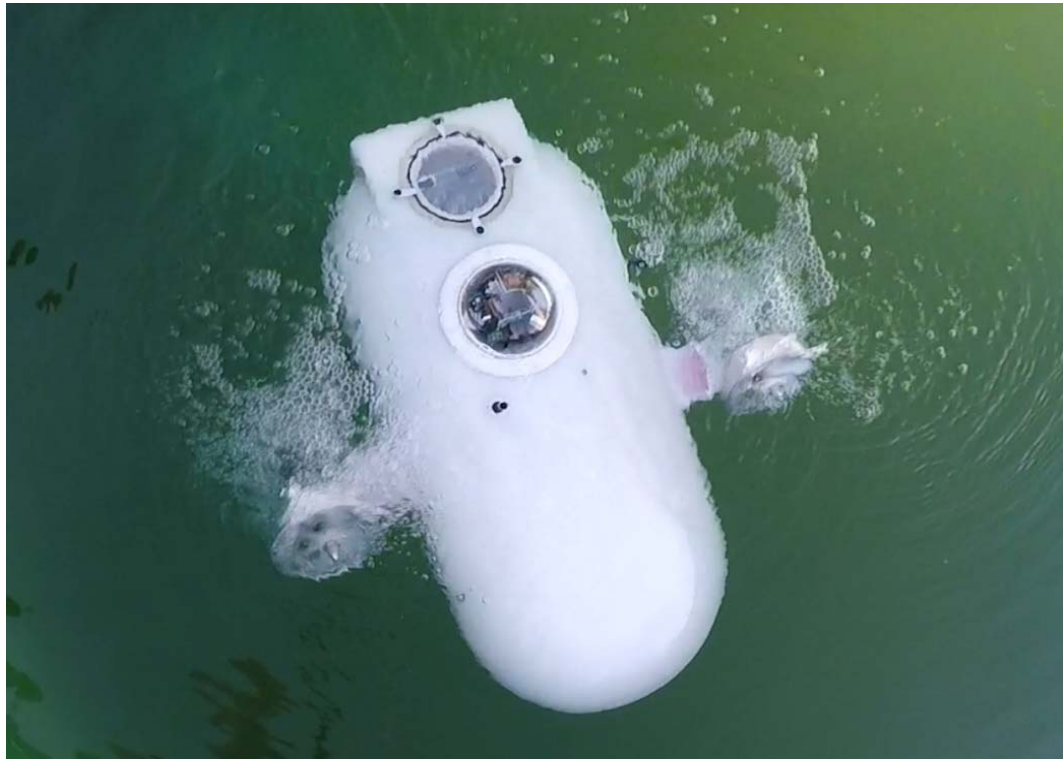


Fig. 6.3: Final prototype.

Integrating so different blocks is indeed challenging. So, we first tested the interaction between blocks to predefined commands before embedding them. Once assembled and insulated, we can access the AUV through the access board. At rest, the servomotors and the mass shifter return to their default mode (the mass shifter return to the center of gravity of the AUV and servomotors return to horizontal state or zero angle). Also, the rotating body (figure 3.29) of the camera chamber is placed on the AUV nose in default mode.

We first connected the Mini-PC to a computer to perform initial verifications. Then, we activated the wireless connection mode for the Mini-PC and removed the cable network from the access board. Next, we placed the cap access valve and sealed it. Once placed on water, the AUV will float at the surface until we send the first set of commands (Fig. 6.5).

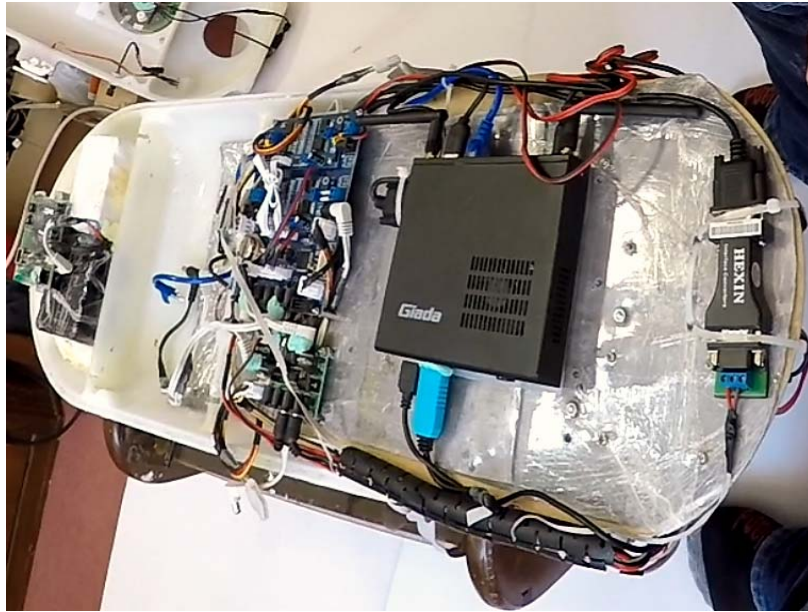


Fig. 6.4: Internal view of the AUV.



Fig. 6.5: The AUV remains on the water when we are sending the commands to begin its mission.

6.3.2 Speed Command

In this part, we tested the AUV's response in controlling speed. In Figure 6.6, the dotted chart (in blue) is the desired speed and the second curve the response of the device. As highlighted in Figure 6.7, the response follows closely the command. Figures 6.8 and 6.9 show the left and right engines corresponding command in terms of engine revolution in %.

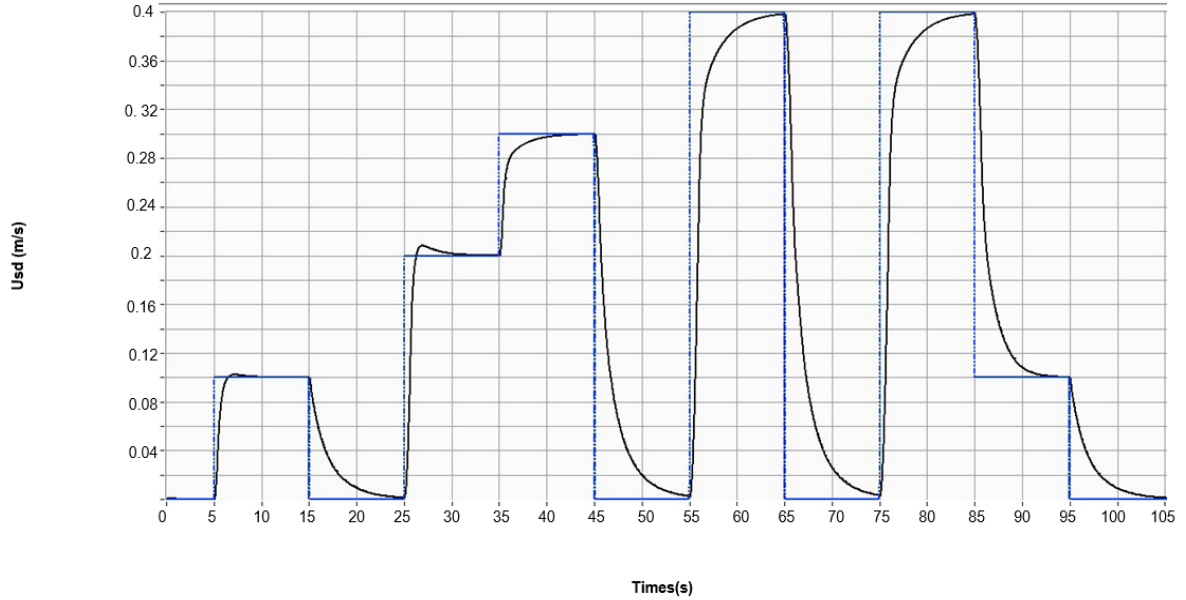


Fig. 6.6: AUV response (Black) to speed command (Blue).

Figure 6.10 shows the body acceleration towards the longitudinal axis of the body. As expected, when the engine revolution is increasing, the acceleration also increased, rising the speed. Consequently, the drag force from the fluid will also increase, reducing the acceleration and thus, making the device moving at a relatively constant speed.

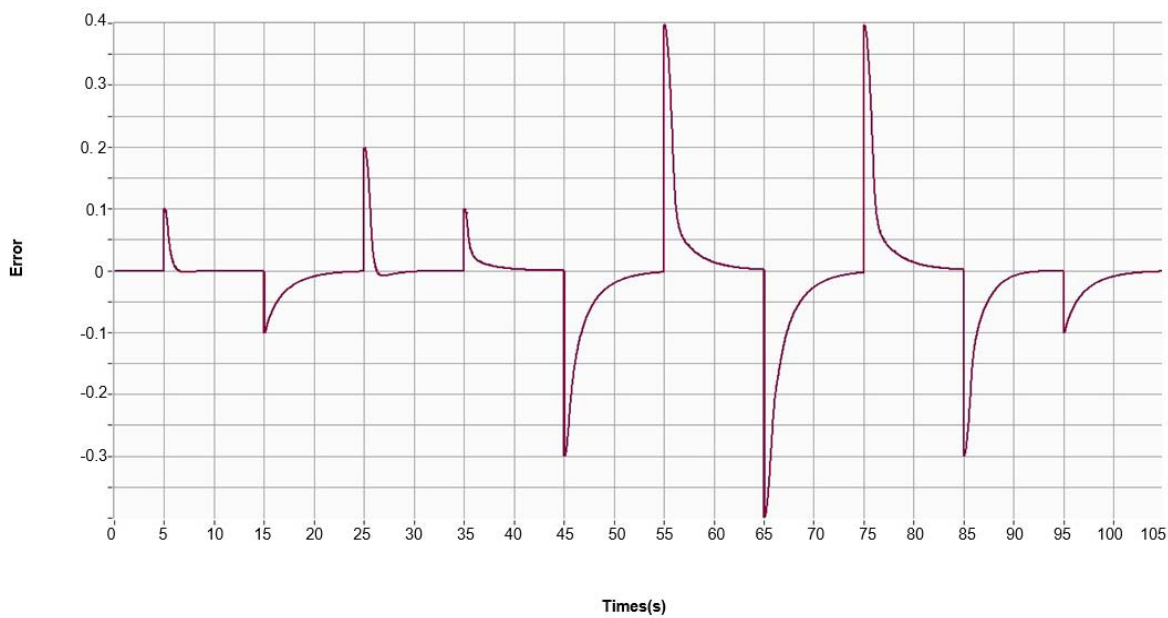


Fig. 6.7: Speed error.

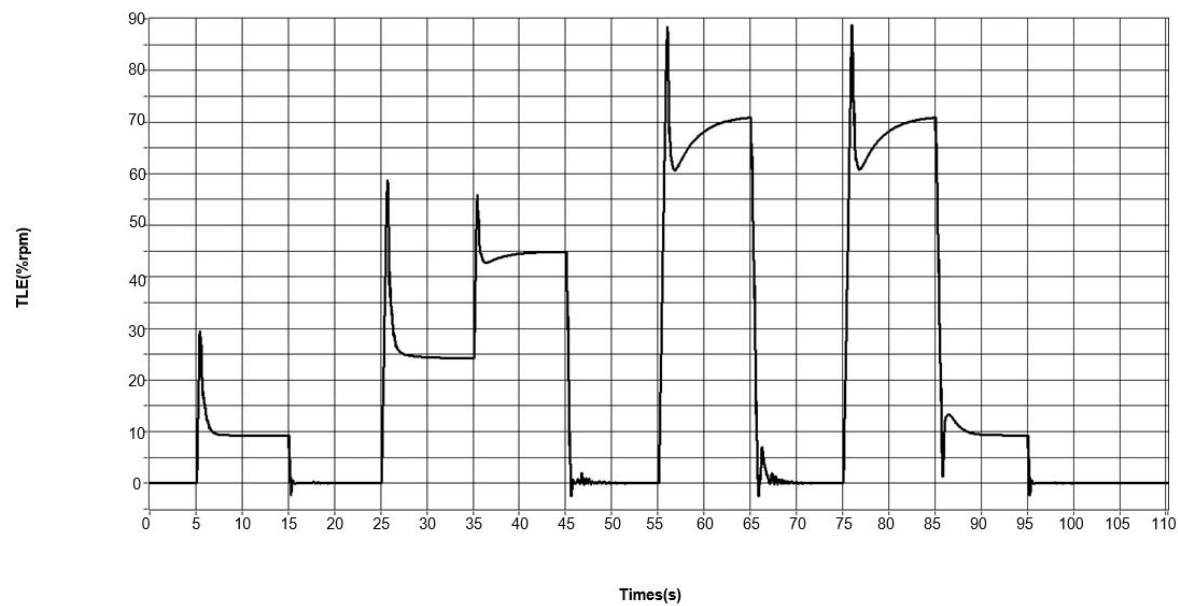


Fig. 6.8: Left engine revolution level.

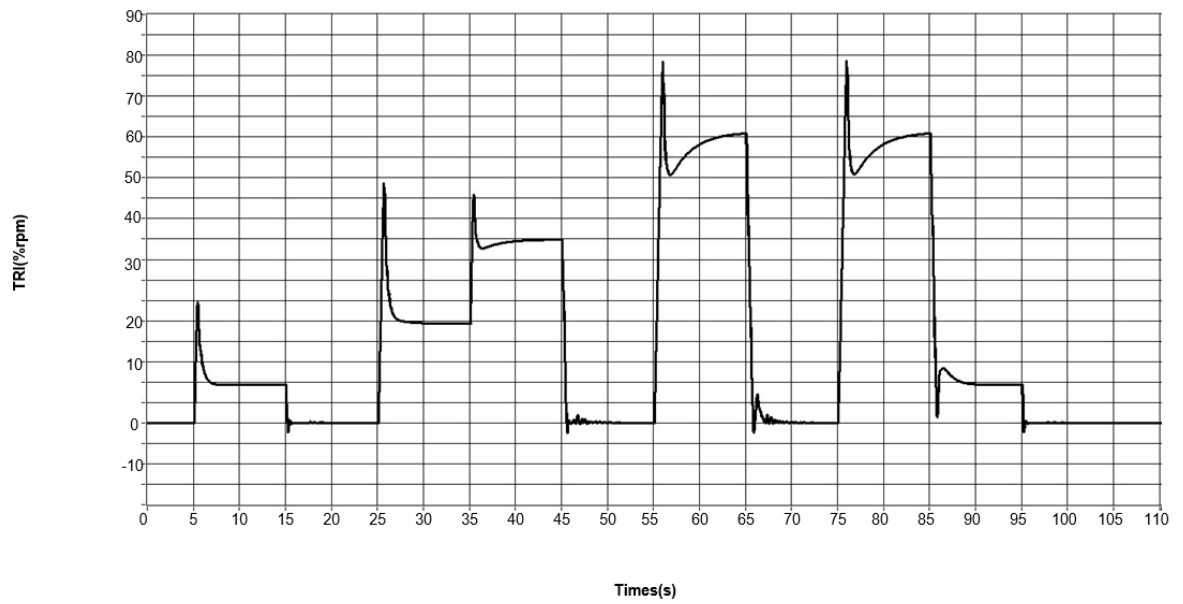


Fig. 6.9: Right engine revolution level.



Fig. 6.10: Acceleration towards x-axis.

6.3.3 Depth Command

To test the AUV in terms of depth command, we selected different depths. As shown in Figure 6.11, the AUV's response follows the command.

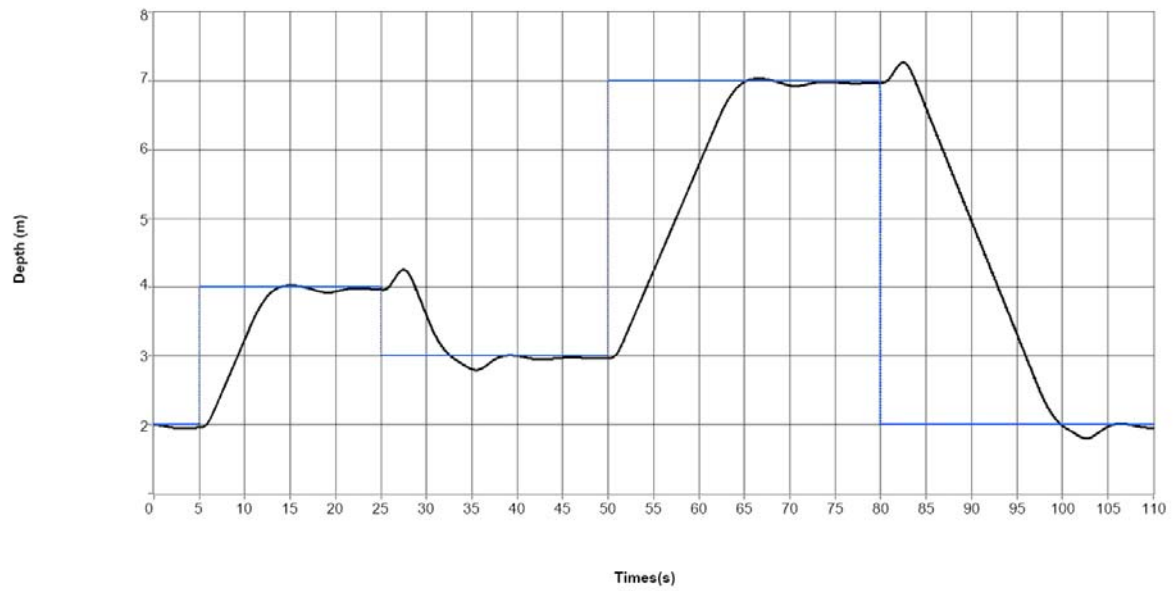


Fig. 6.11: AUV response (Black) to selected depths (Blue).

Figure 6.12 shows the location of the moving weight (Mass shifter) over time. As shown, at any depth change, the poise is moving either backwards or forwards to balance the AUV, then coming back to its steady state position.

Figure 6.13 shows the nose angle for the above depth command. This Figure is comparable to the precedent one. When the poise is moving forward (positive displacement), the nose comes down and when the weight is moving backward (negative displacement), the nose goes up.

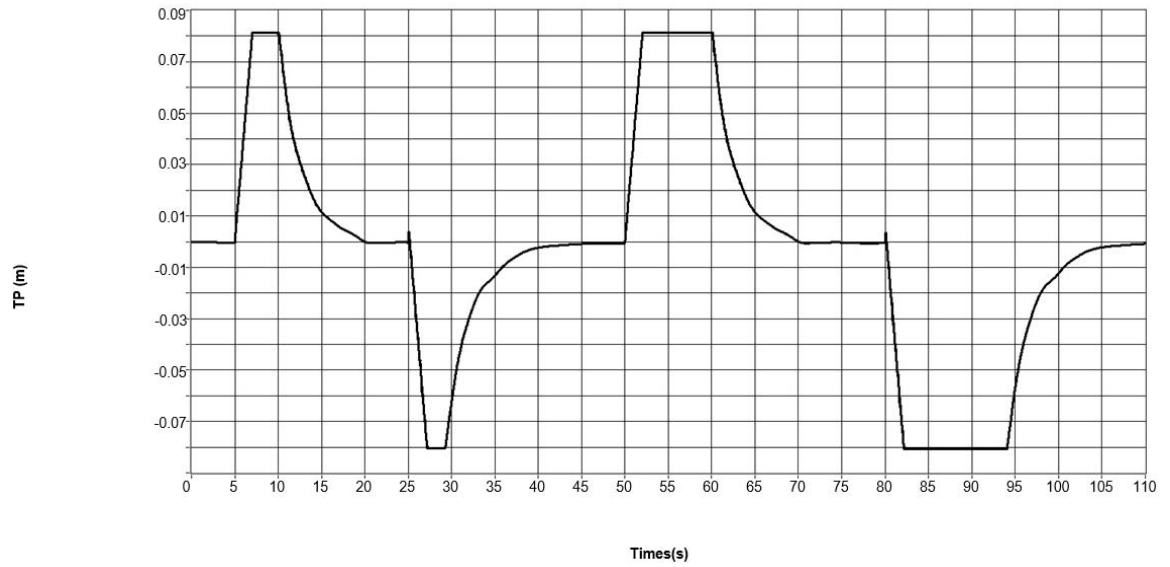


Fig. 6.12: Position of the moving weight (Mass shifter).

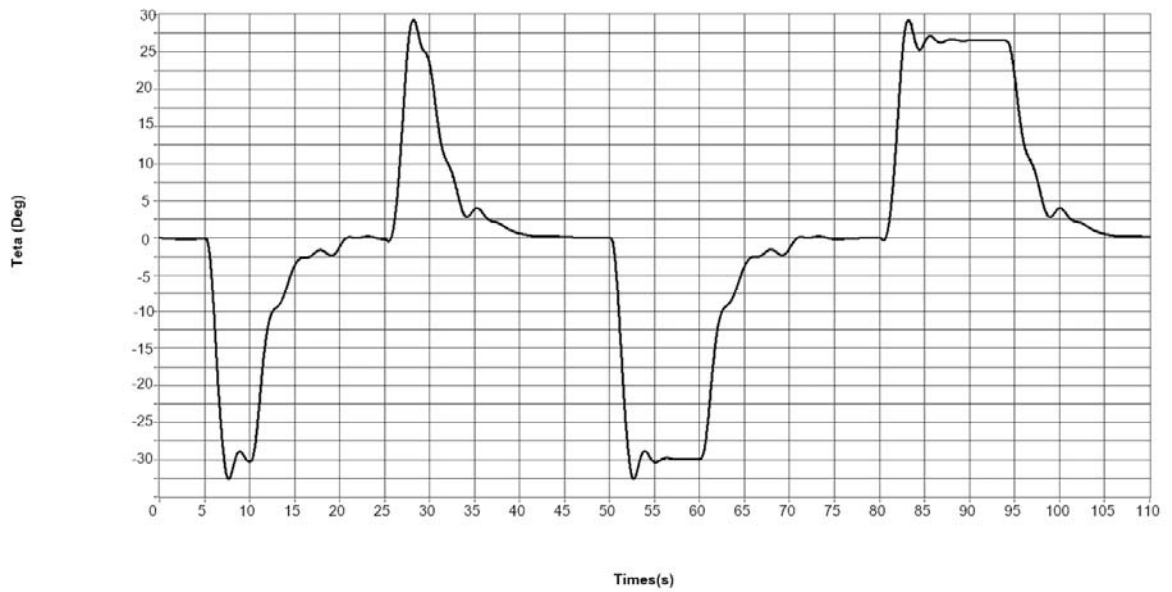


Fig. 6.13: Pitch angle.

Accordingly, Figures 6.14 and 6.15 show the respective engine angular velocity around the y-axis and the speed in the vertical direction.

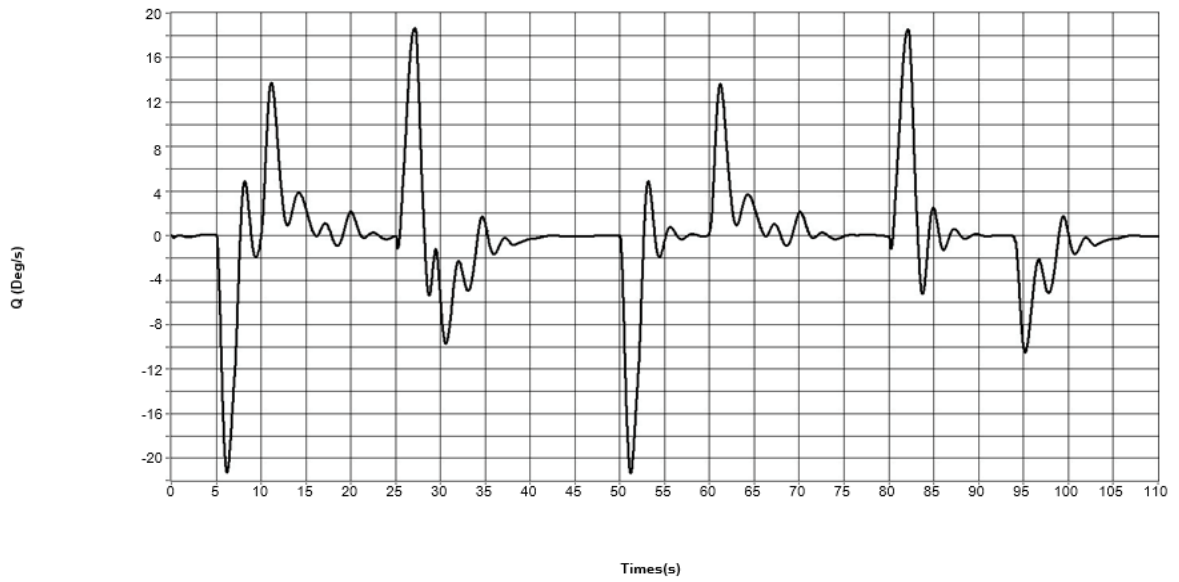


Fig. 6.14: Angular velocity around y-axis.

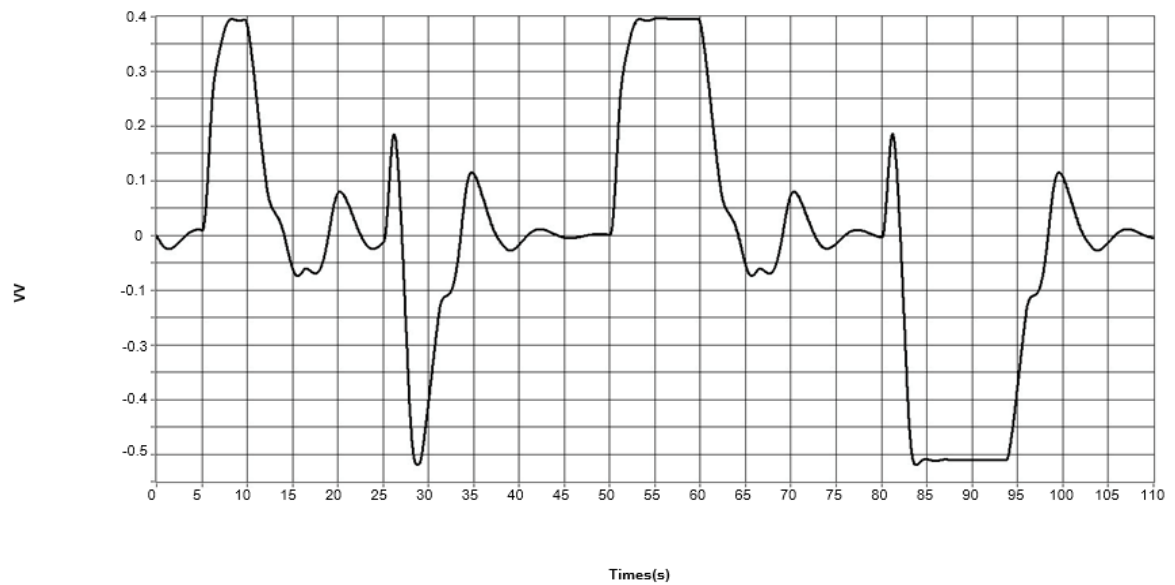


Fig. 6.15: Vertical speed.

6.3.4 Integrating Speed and Depth Commands

Next, the objective was to maintain a constant speed of 0.2 m/s while moving at a desired depth of 5 m. As shown in Figures 6.16 and 6.17, the AUV follows closely the commands with an almost null vertical speed.

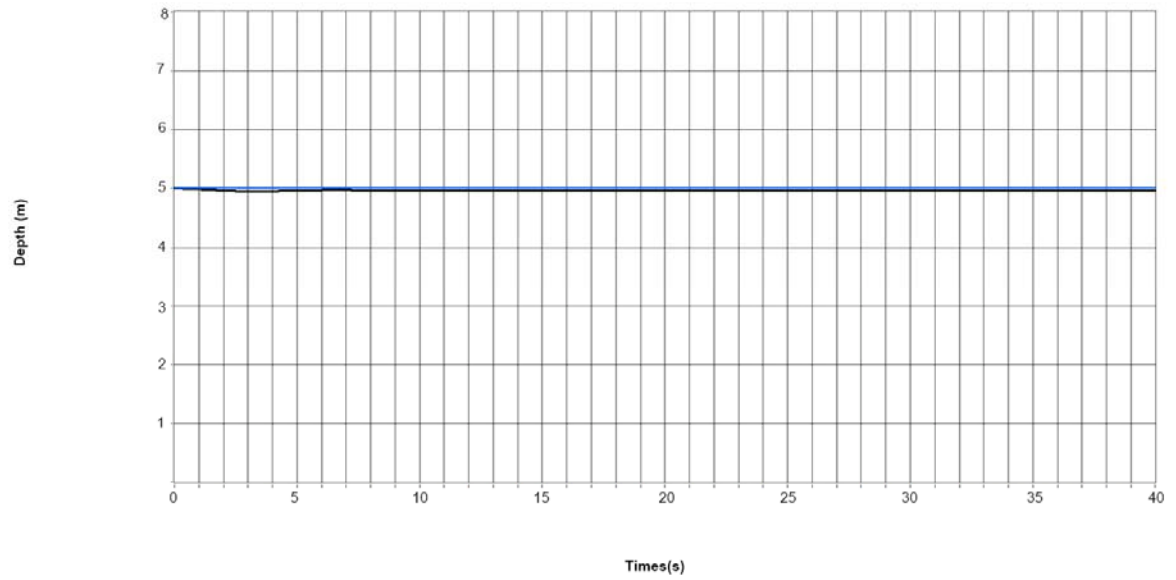


Fig. 6.16: Depth.

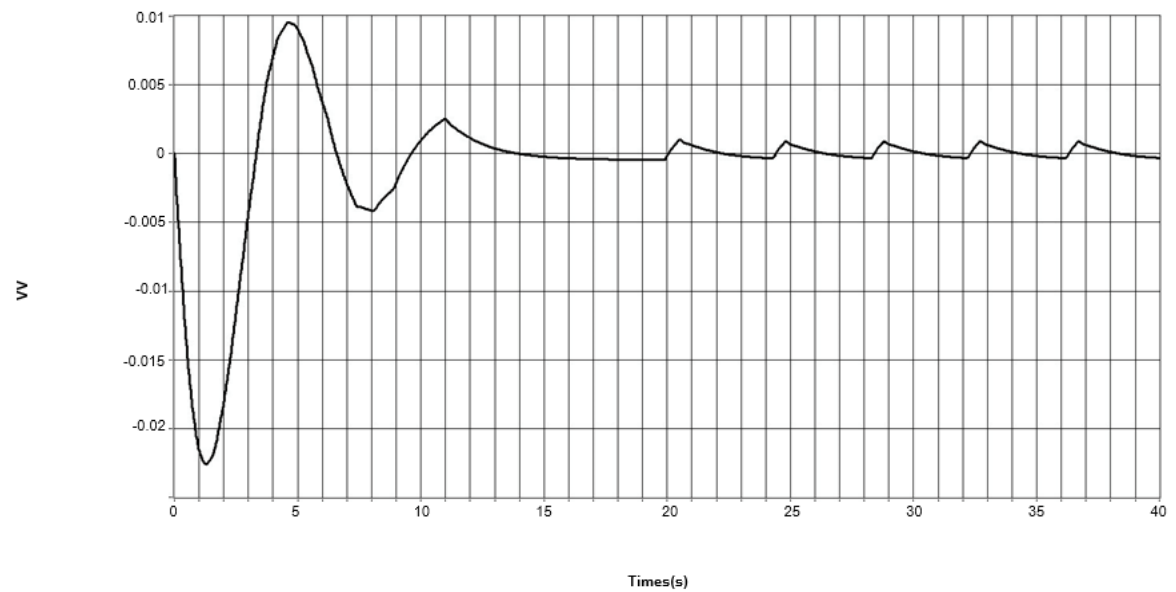


Fig. 6.17: Vertical speed.

Figures 6.18 and 6.19 show the evolution of the angle of the left and right servos towards the body. As noted, the servos angles are stabilized to a value of -12.5° to ensure a constant speed of 0.2 m/s at a depth of 5 m.

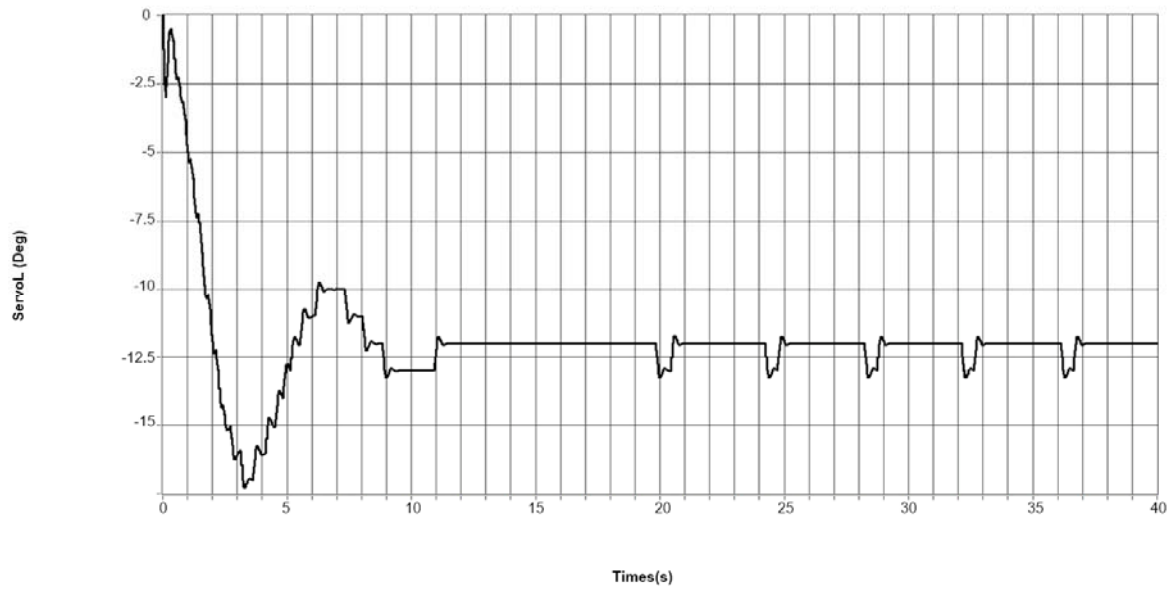


Fig. 6.18: Left servo angle.

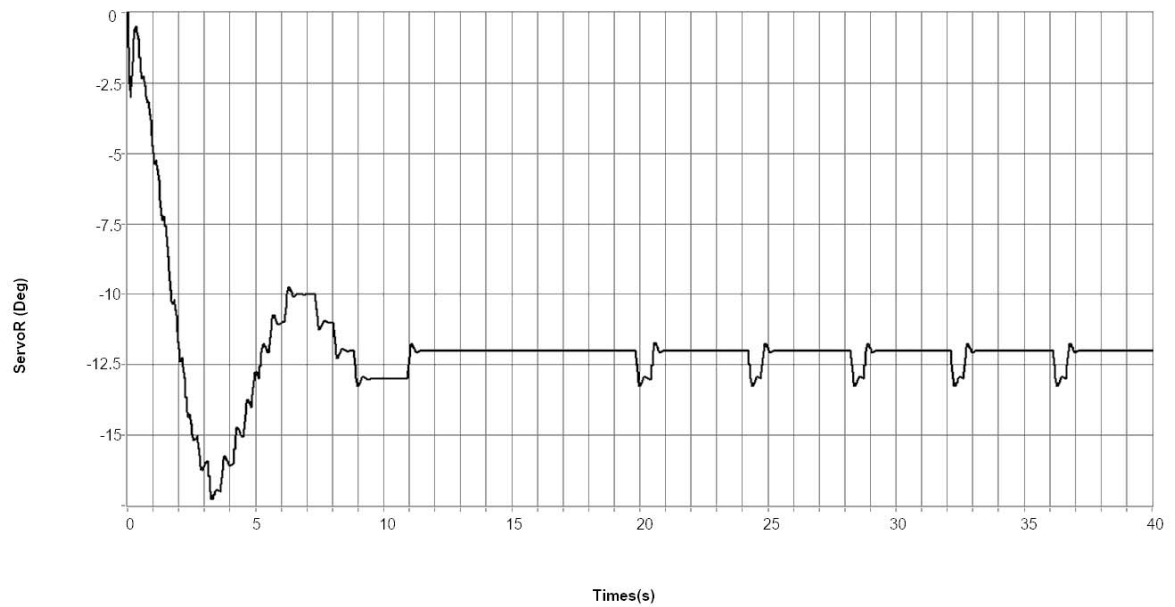


Fig. 6.19: Right servo angle.

Then, the desired depth was maintained to 5 m while reducing the speed command to 0.1m/s. From Figures 6.20 and 6.21, we can note that with this low speed, the servos are more reactive and a larger servo angle is required (-30°) to keep the desired objectives.

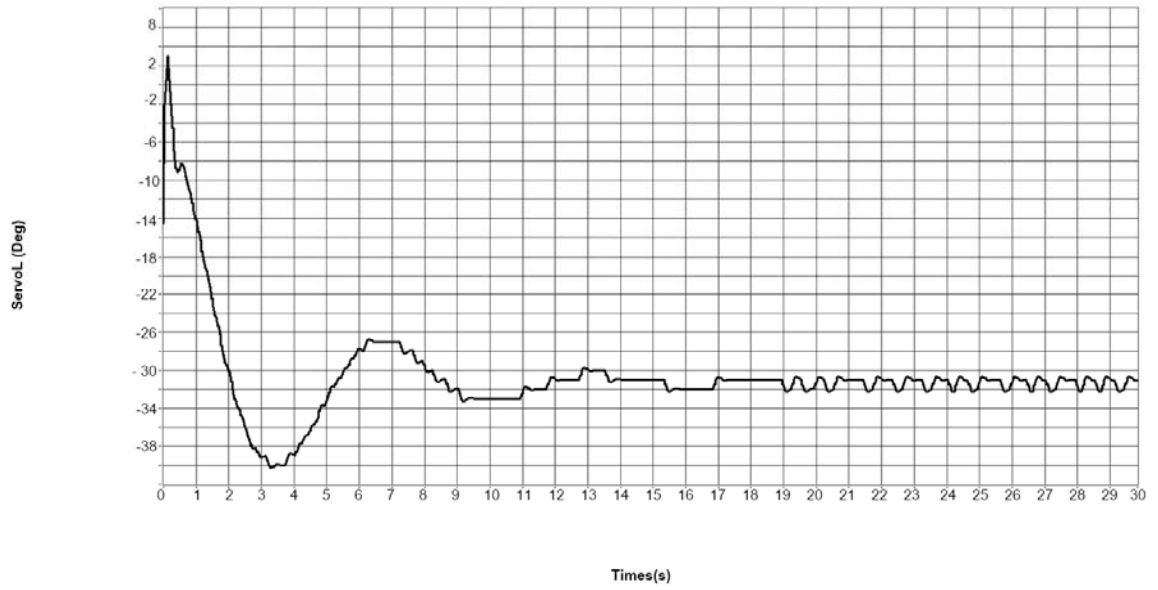


Fig. 6.20: Left servo angle.

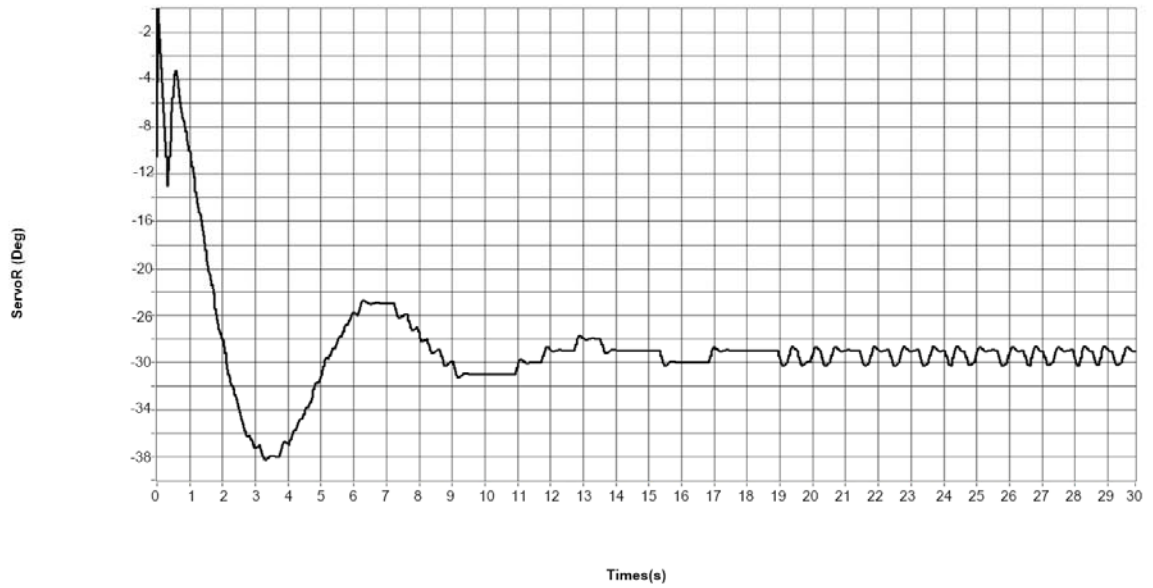


Fig. 6.21: Right servo angle.

6.3.5 Integrating Speed, Pitch and Depth Commands

In this test, the AUV was set in different operating conditions integrating all three commands: speed, depth and pitch at different periods.

The commands are:

- $t = 0$ s, $u_{sd} = 0$ m/s, $lD_d = 1$ m
- $t = 5$ s, $u_{sd} = 0.2$ m/s, $lD_d = 3$ m
- $t = 20$ s, $u_{sd} = 0.3$ m/s, $lD_d = 7$ m
- $t = 50$ s, $u_{sd} = 0.1$ m/s, $lD_d = 1$ m
- $t = 90$ s, $u_{sd} = 0.45$ m/s, $lD_d = 6$ m

As shown in Figures 6.22 to 6.27, the AUV response adequately follows the commands in terms of desired speed and depth. The mass shifter as well as the left and right servos contributed jointly to maintain the AUV operation within the required constraints.

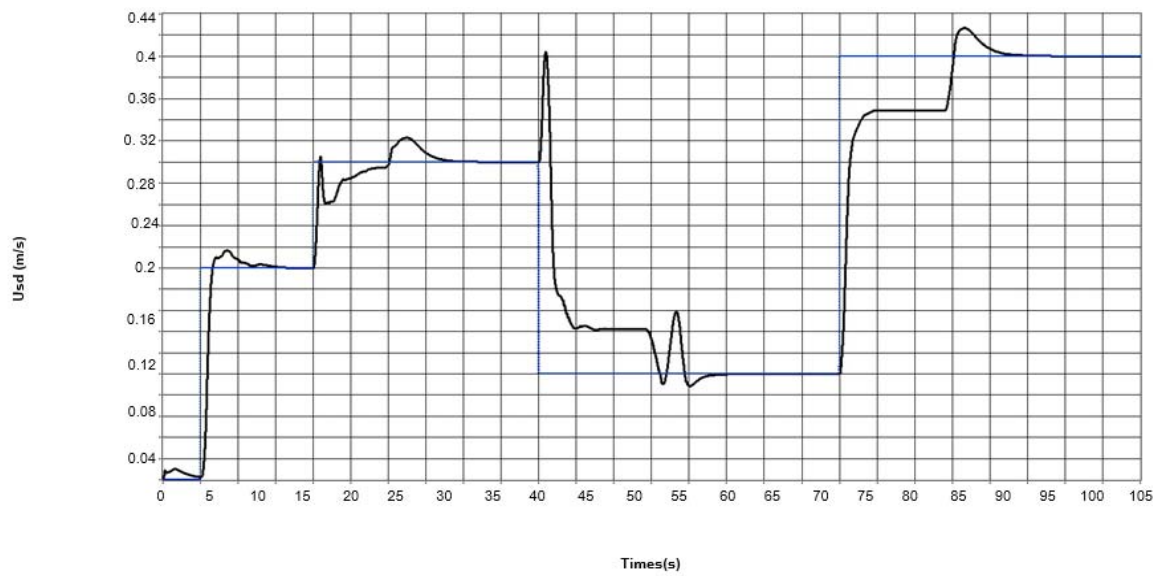


Fig. 6.22: Speed response.

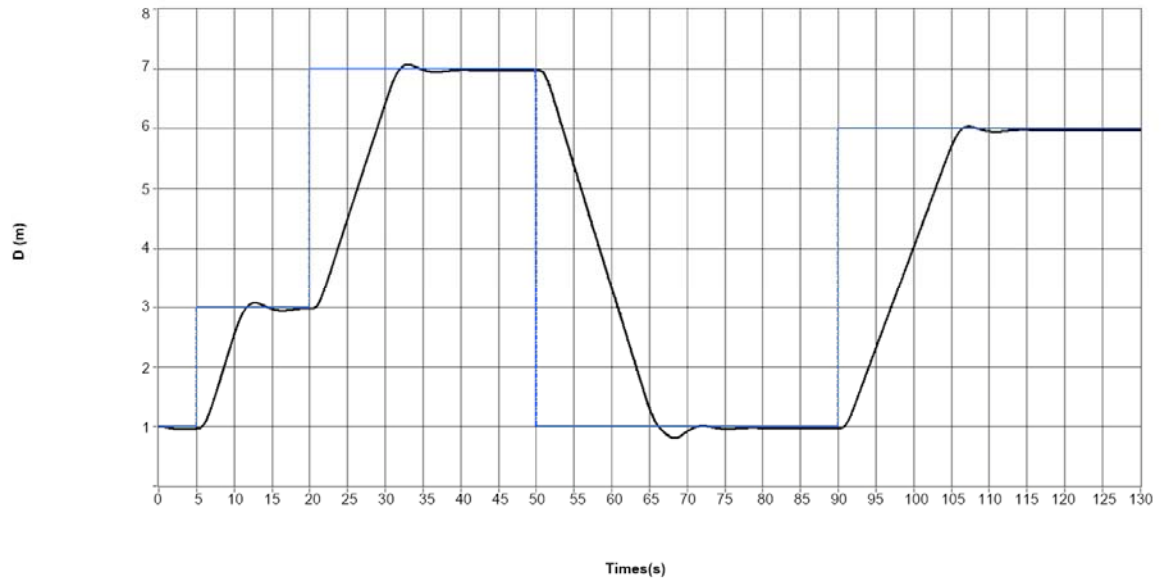


Fig. 6.23: Depth response.

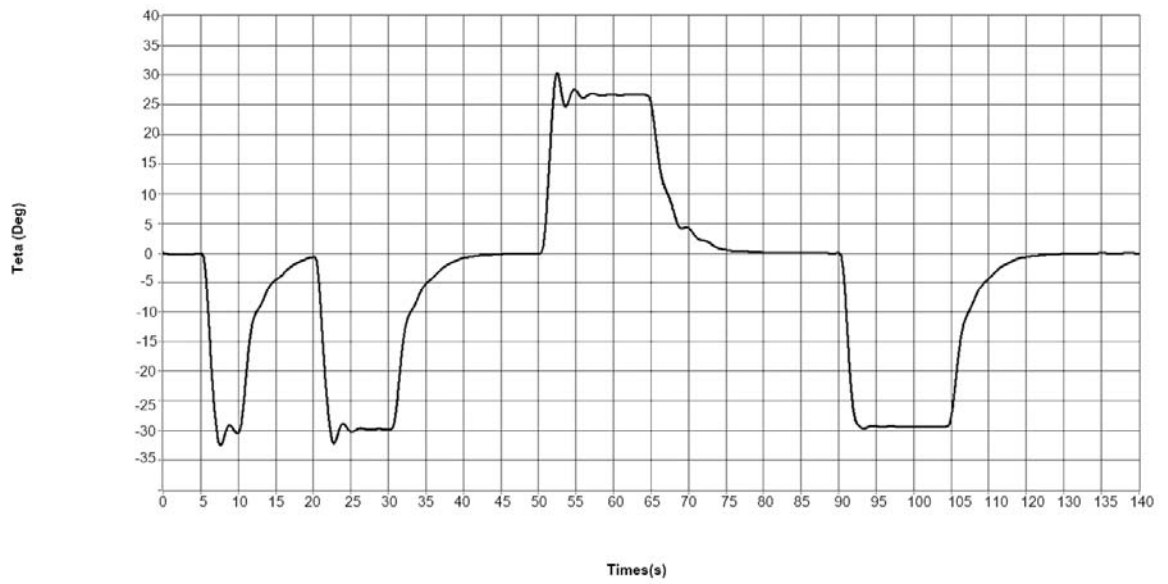


Fig. 6.24: Pitch angle.

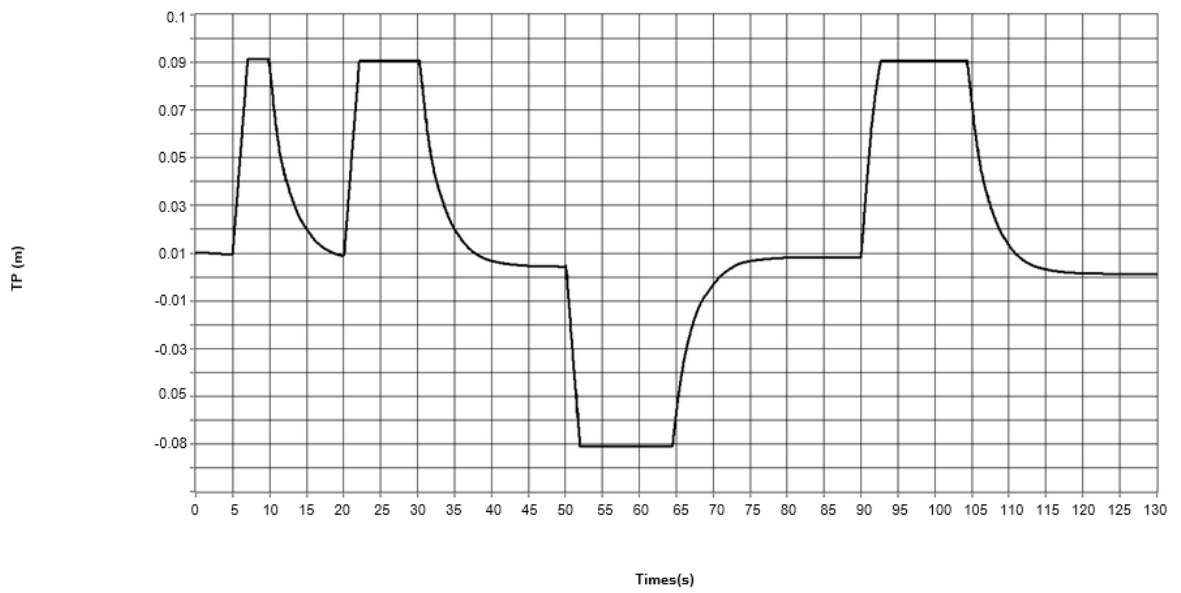


Fig. 6.25: Mass shifter variation.

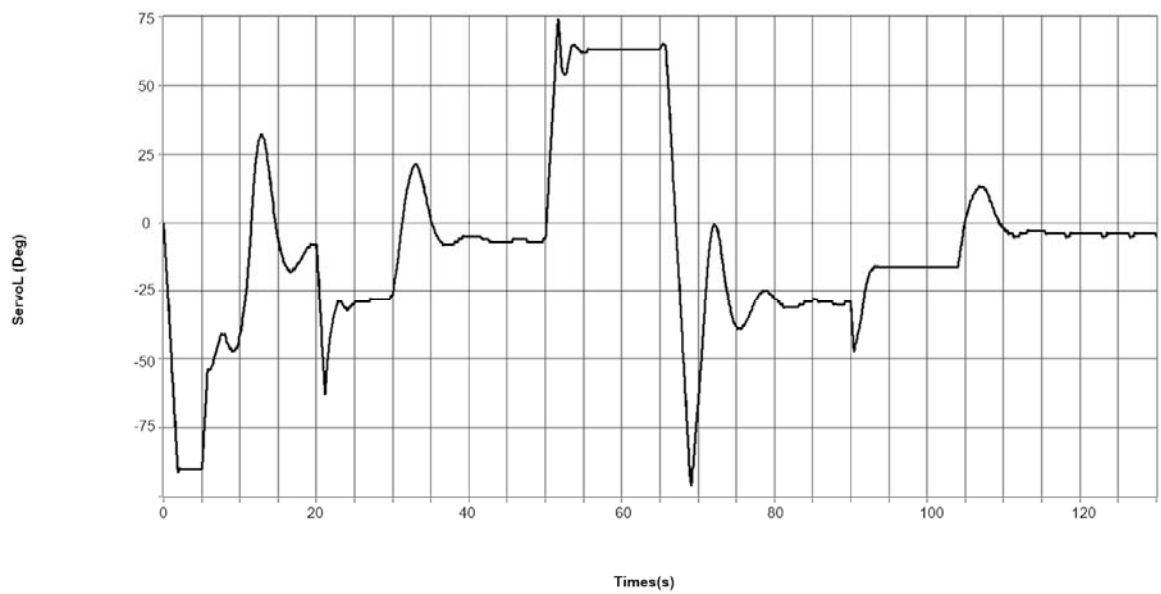


Fig. 6.26: Left servo angle.

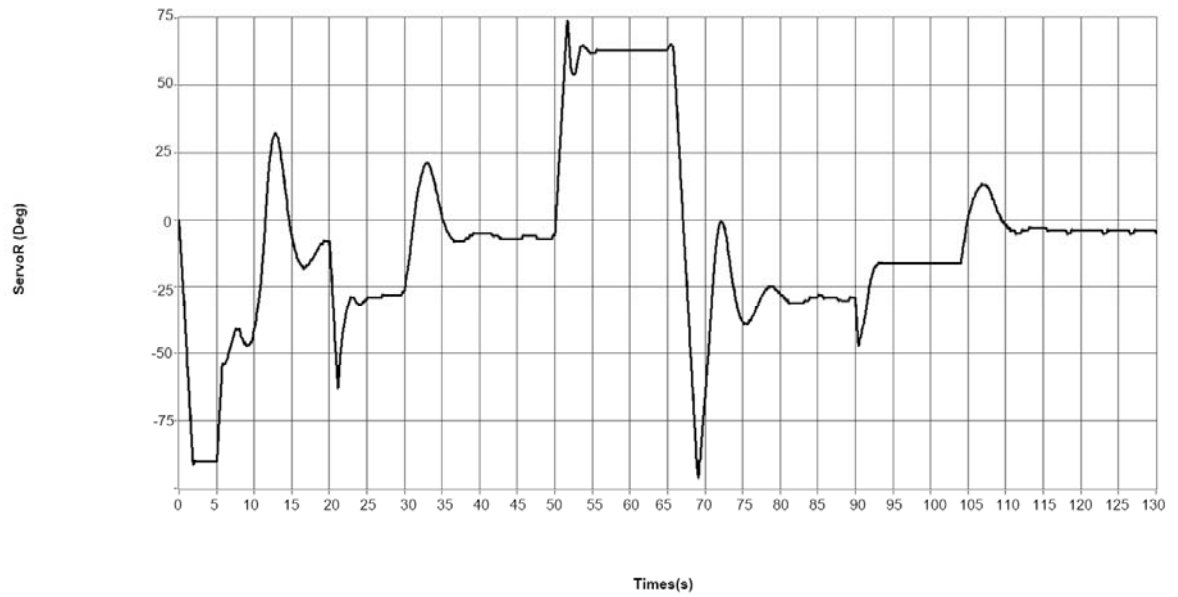


Fig. 6.27: Right servo angle.

6.3.6 Direction Command

In this test, the command was to follow a desired path defined in terms of direction angles (Figure 6.28). Here again, the prototype outputs follow the commands.

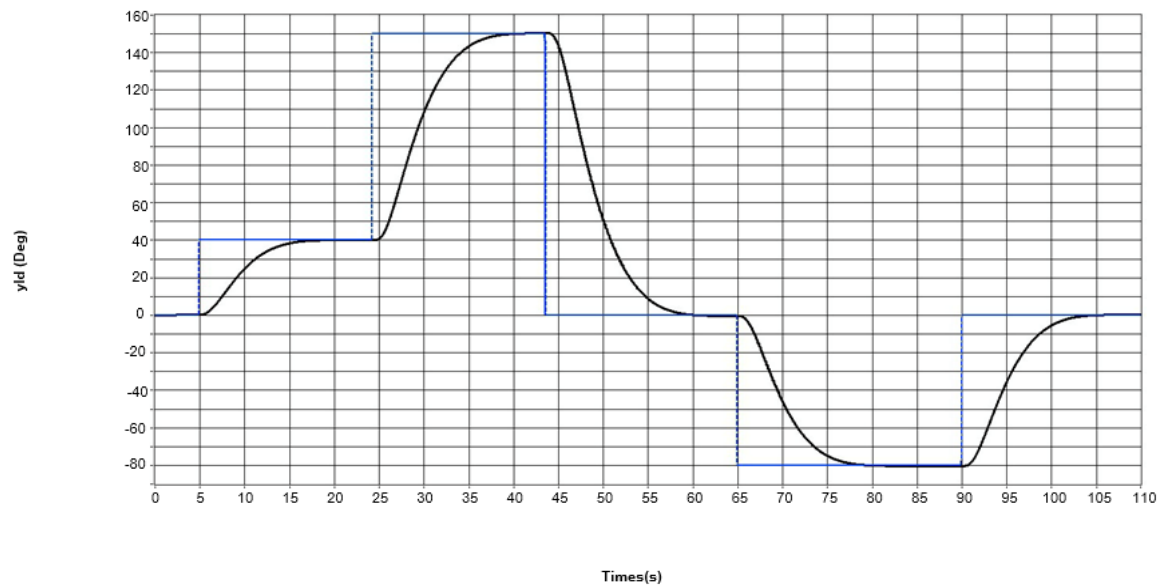


Fig. 6.28: AUV response (Black) to desired direction angle (Blue).

In fact, as in Figure 6.29, a positive angular velocity implies an increase of the direction speed in the positive direction and when the angular velocity is negative, the direction speed increased in the negative direction.

Also, as shown in Figures 6.30 and 6.31, when the AUV should rotate towards the right direction to reach the desired direction angle, the left servo angle increases and approaches to horizontal state (i.e., 0° angle) and when the AUV should rotate towards left, the angle of the right servo increases and becomes closer to the horizon.

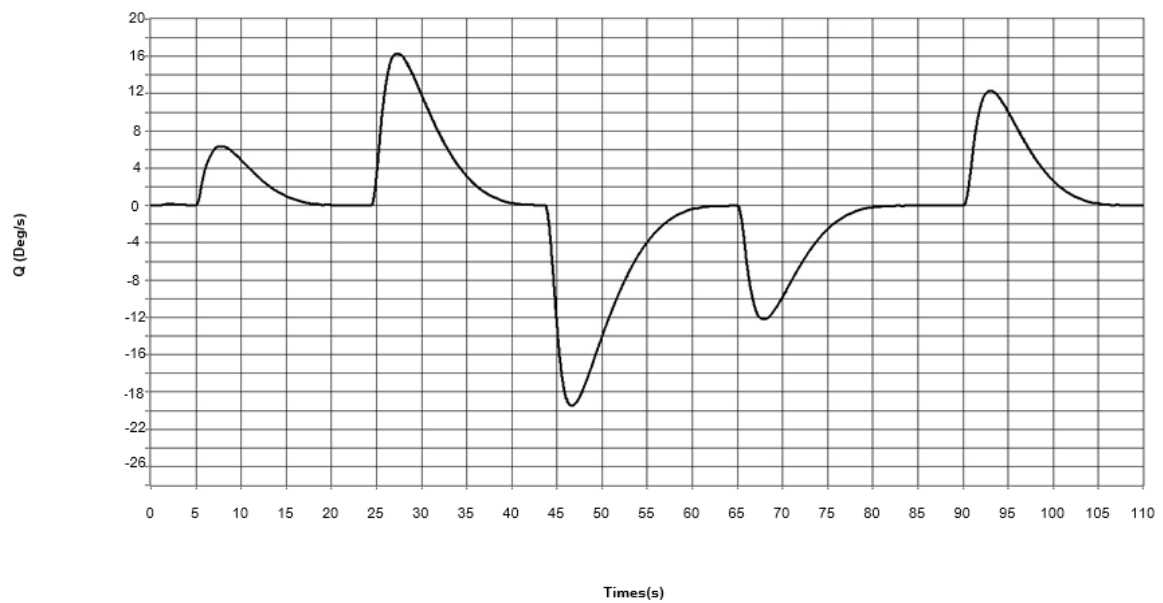


Fig. 6.29: Angular velocity around z-axis.

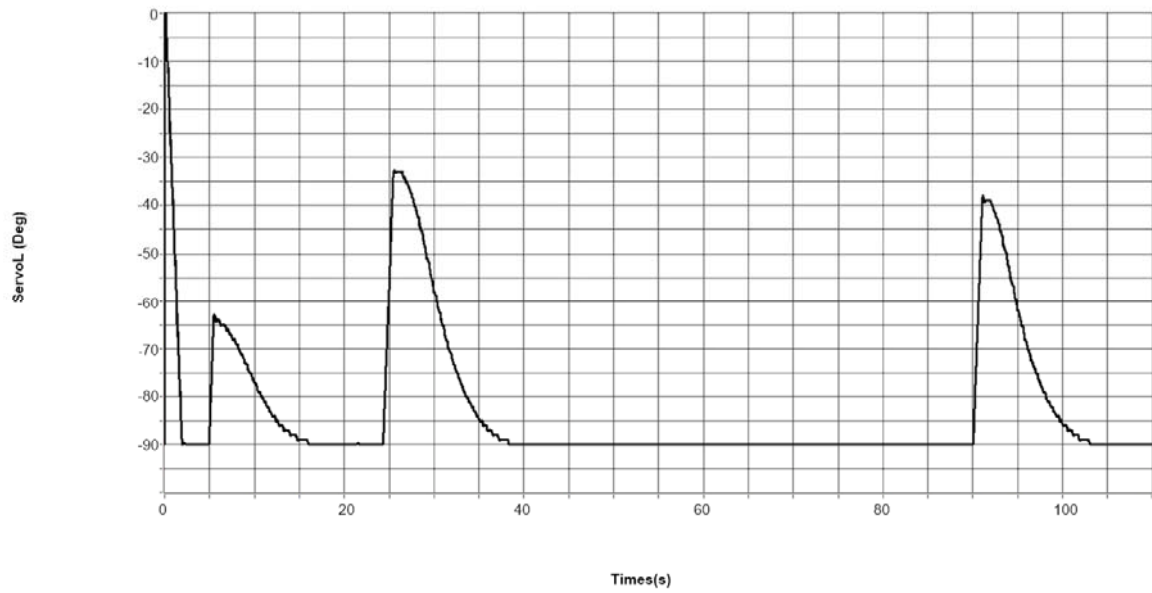


Fig. 6.30: Left servo angle.

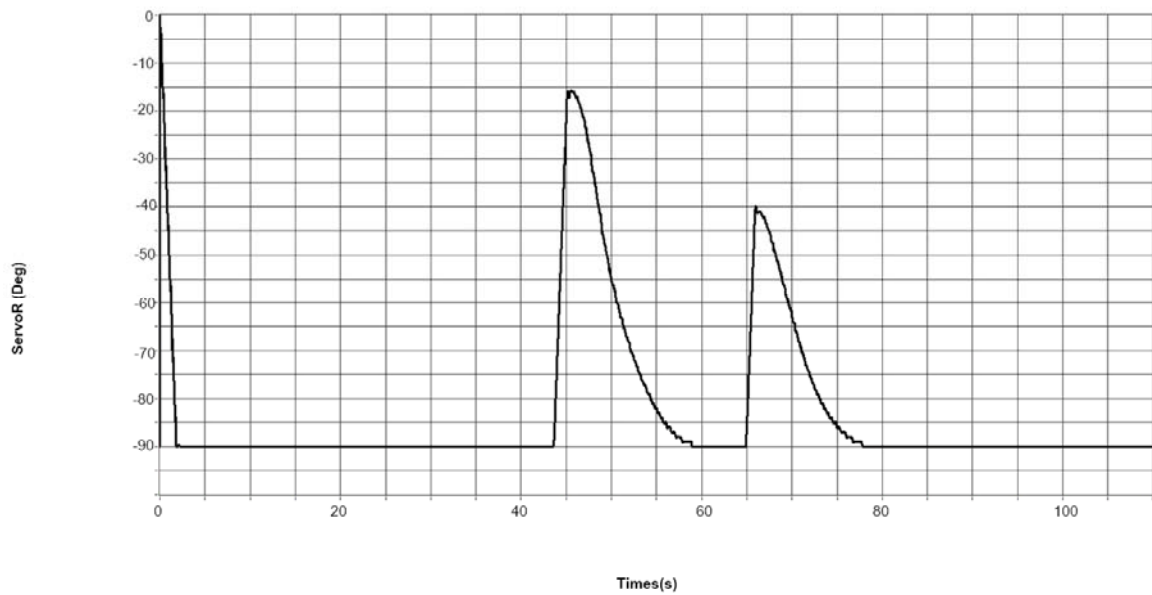


Fig. 6.31: Right servo angle.

6.4 Controller Integration

In this section, all the above controllers (speed, pitch, depth and direction) were integrated into a single algorithm (Figure 6.32) and the whole system behavior evaluated.

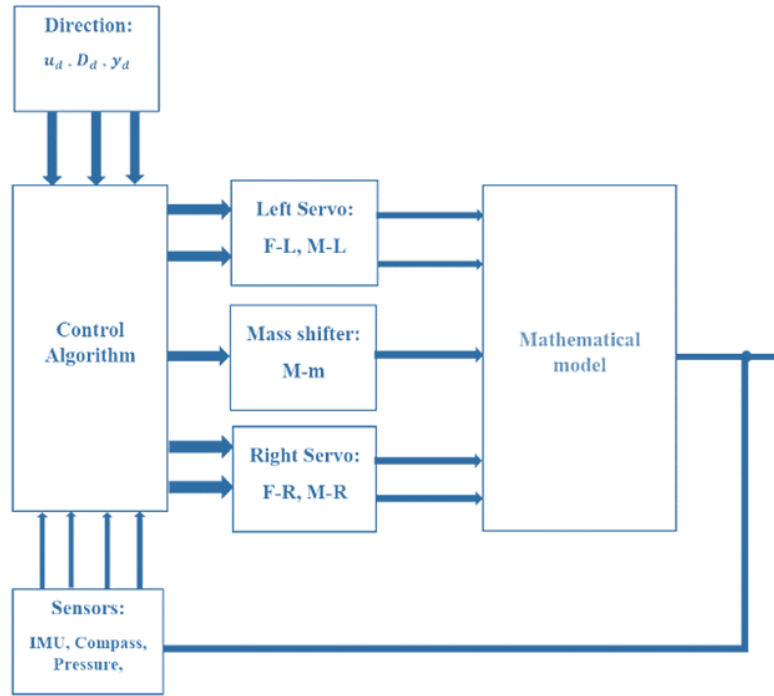


Fig. 6.32: Final control algorithm.

For this test, a predefined set of commands was sent to the prototype in terms of desired speed (Figure 6.33), depth (Figure 6.34) and direction angle (Figure 6.35). As shown in Figures 6.36 to 6.48, the AUV's response follows the given commands, thus allowing to validate the algorithm implementation. The related controllers' coefficients are summarized in Table 6.2.

Table 6.2: Controlling coefficients for the whole system.

Controlling Coefficients	Value
$kPAU$	500
kSU	40
kPB_p	0.001
kPC_{D_1}	300
kRD_{D_1}	0.1
kPC_{D_2}	100
kRD_{D_2}	0.1
kPC_y	0.3
kLC_y	0.001

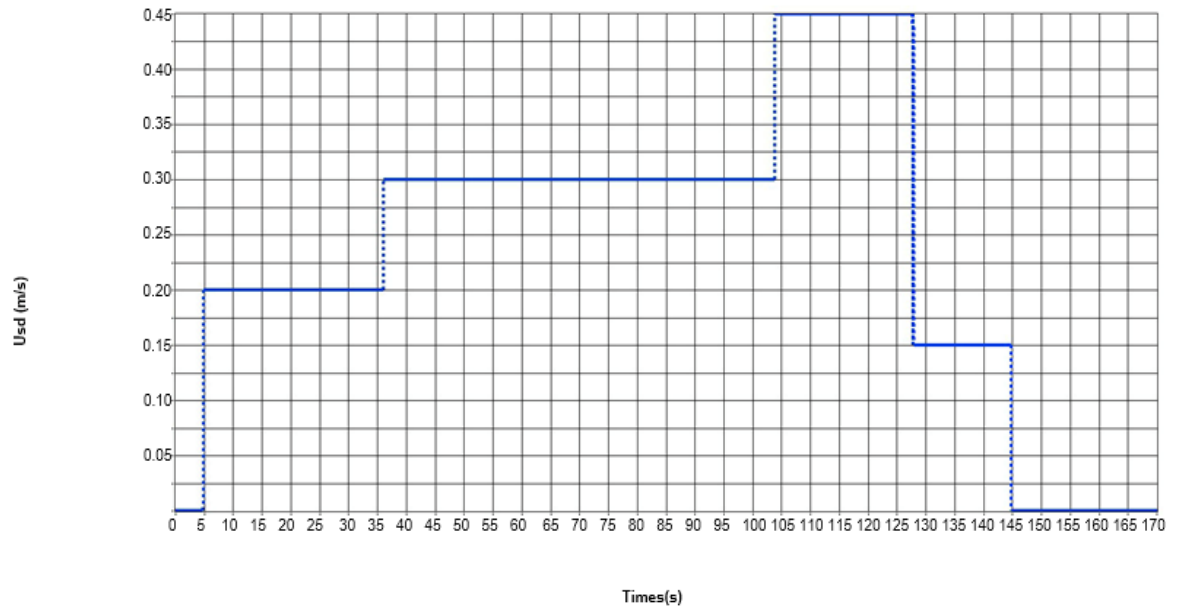


Fig. 6.33: Desired speed.

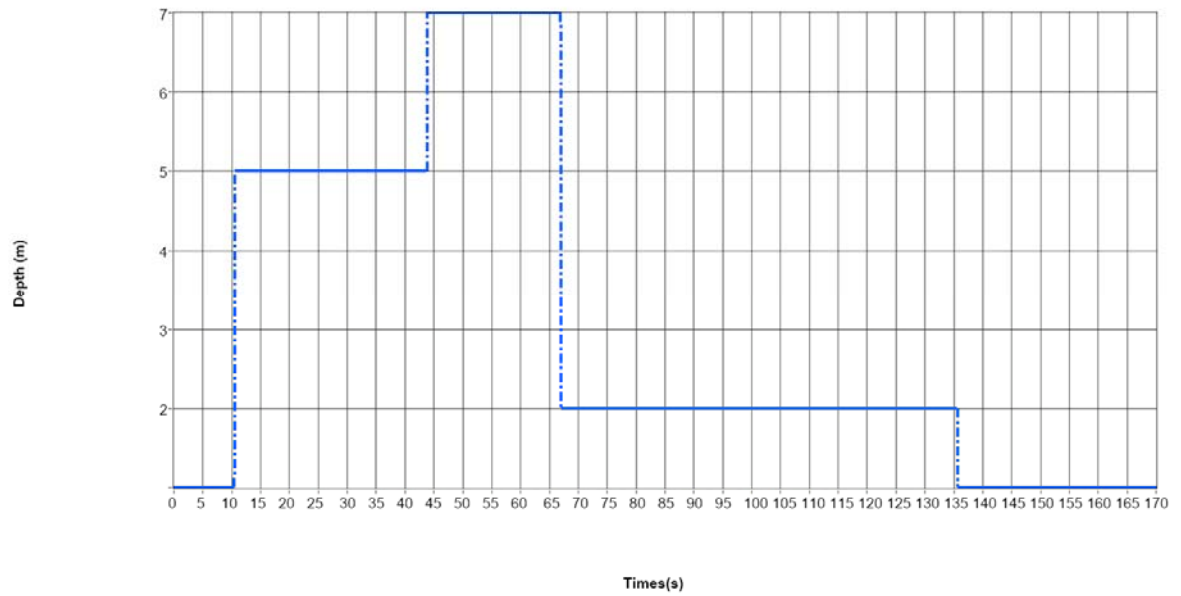


Fig. 6.34: Desired depth.

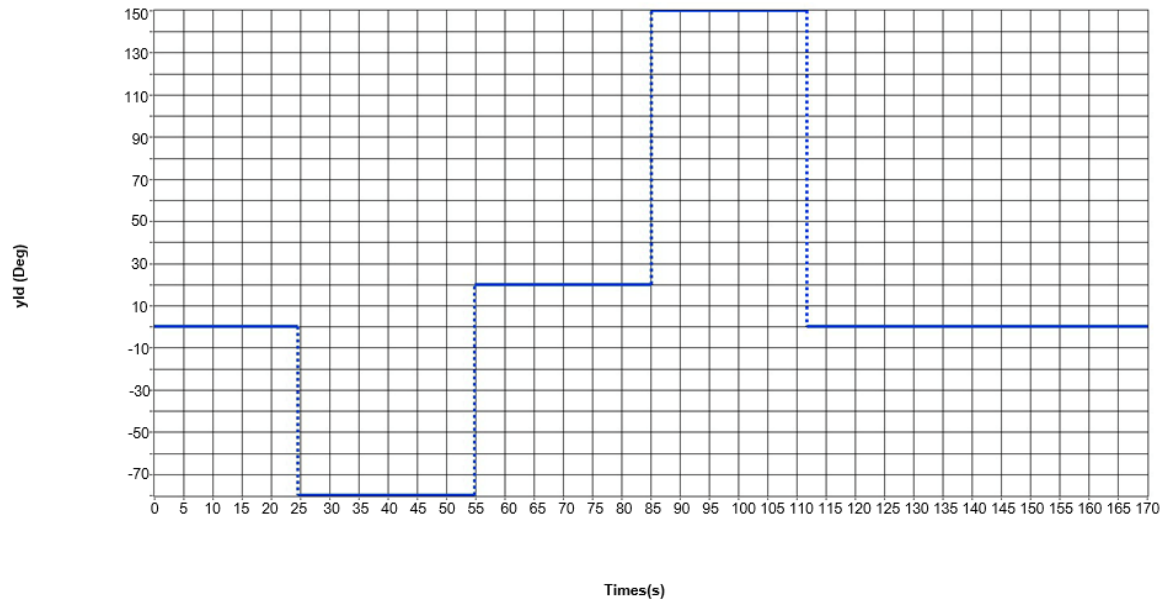


Fig. 6.35: Desired direction angle.

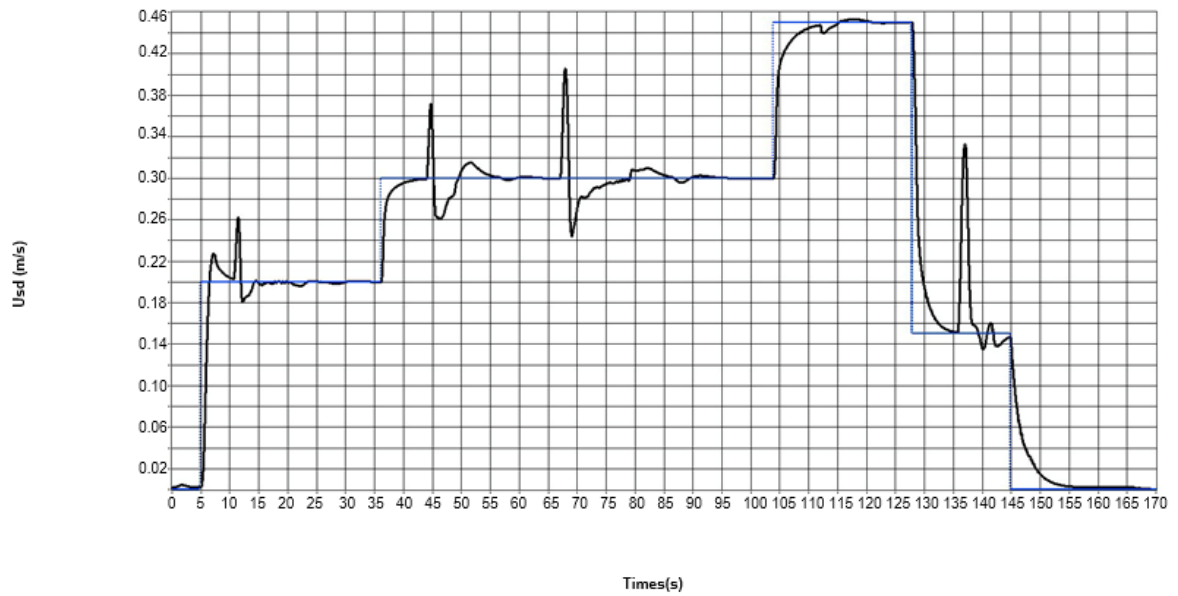


Fig. 6.36: AUV response (Black) to speed command (Blue).

In fact, while comparing figures 6.36 and 6.37, we can notice that the evolution of the speed prototype to the speed command agrees with the vertical speed. In spots where the depth change occurs, this later rises to balance the AUV's moving.

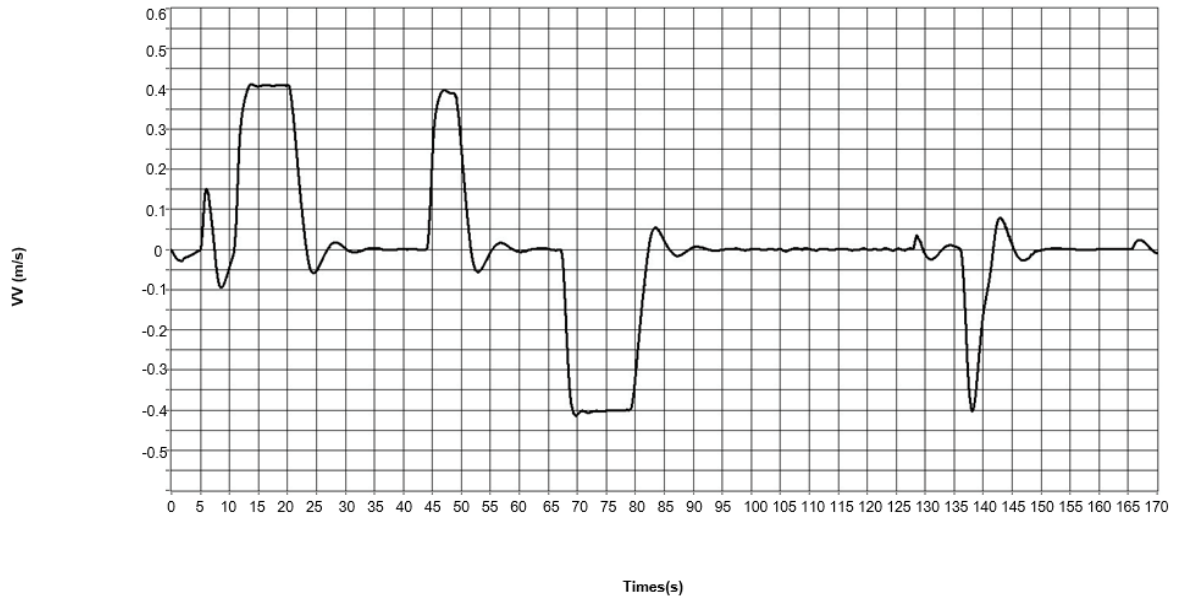


Fig. 6.37: Vertical speed.

This has been further confirmed by the acceleration curves (Figures 6.38 to 6.40). Indeed, when the AUV changes the angle of its pitch and rotates towards left and right, an acceleration is applied to the AUV in the y-axis direction of the body to closely follow its commands. Similarly, for the other acceleration components.

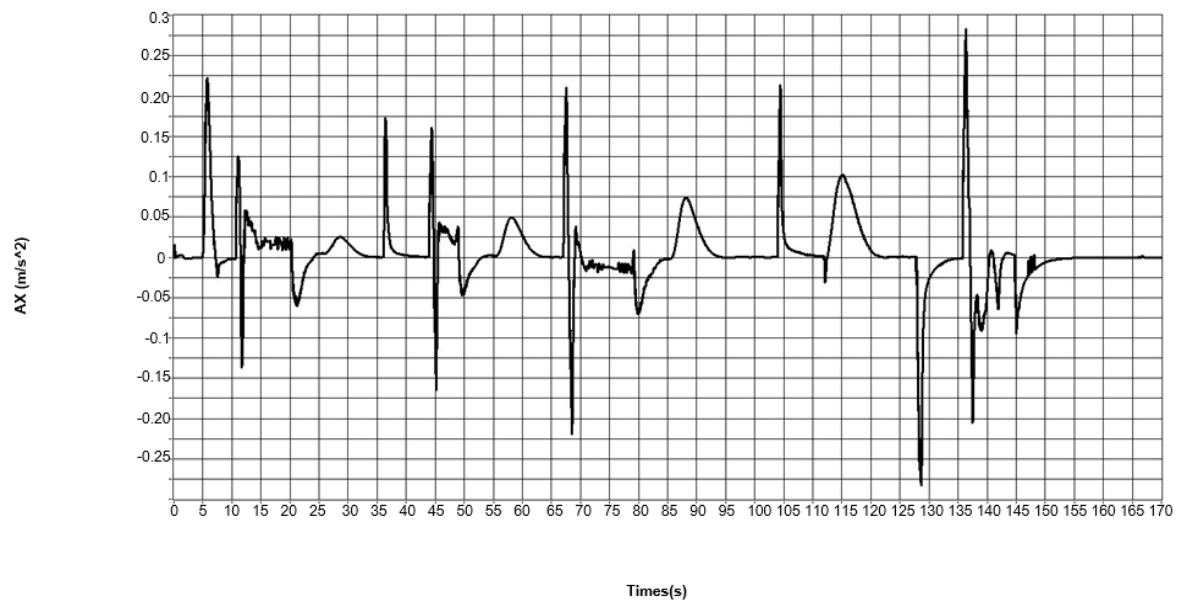


Fig. 6.38: Acceleration towards x-axis.

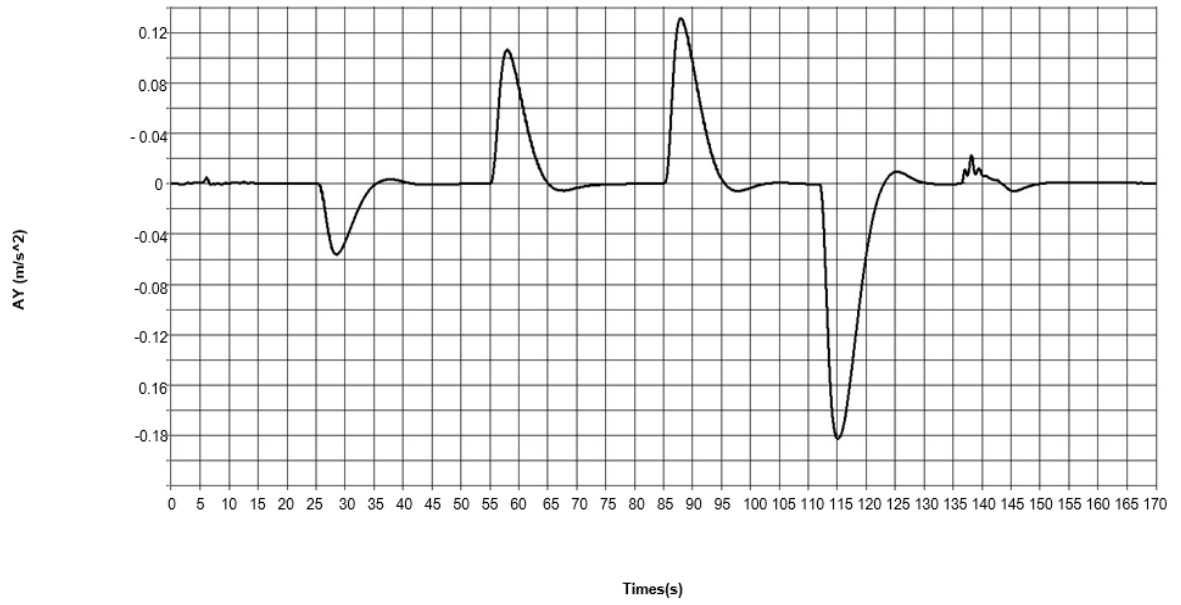


Fig. 6.39: Acceleration towards y-axis.



Fig. 6.40: Acceleration towards z-axis.

Same conclusions can be made for the depth and pitch angle (figures 6.41 and 6.42).

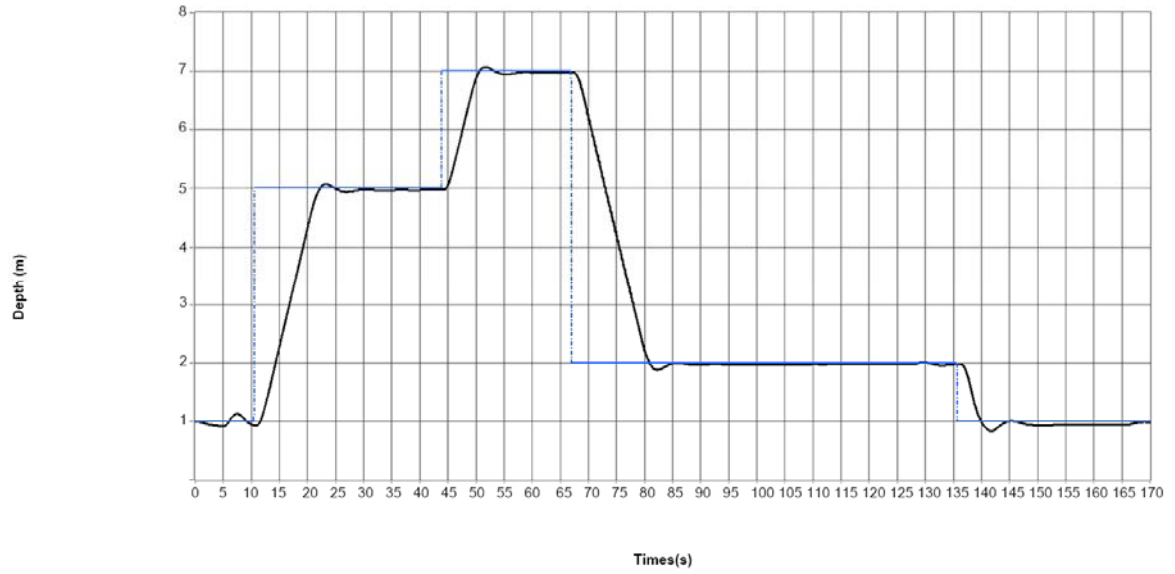


Fig. 6.41: AUV response (Black) to depth command (Blue).

We should note that to maintain the stability of the AUV, angular velocity around the x-axis occurred to keep it stable around its longitudinal axis (figure 6.43). The variations of the y- and z-axis angular velocities as well as the servo angles and the corresponding moving of the mass shift weight confirm the good operation of the prototype (Figures 6.44 to 6.48).

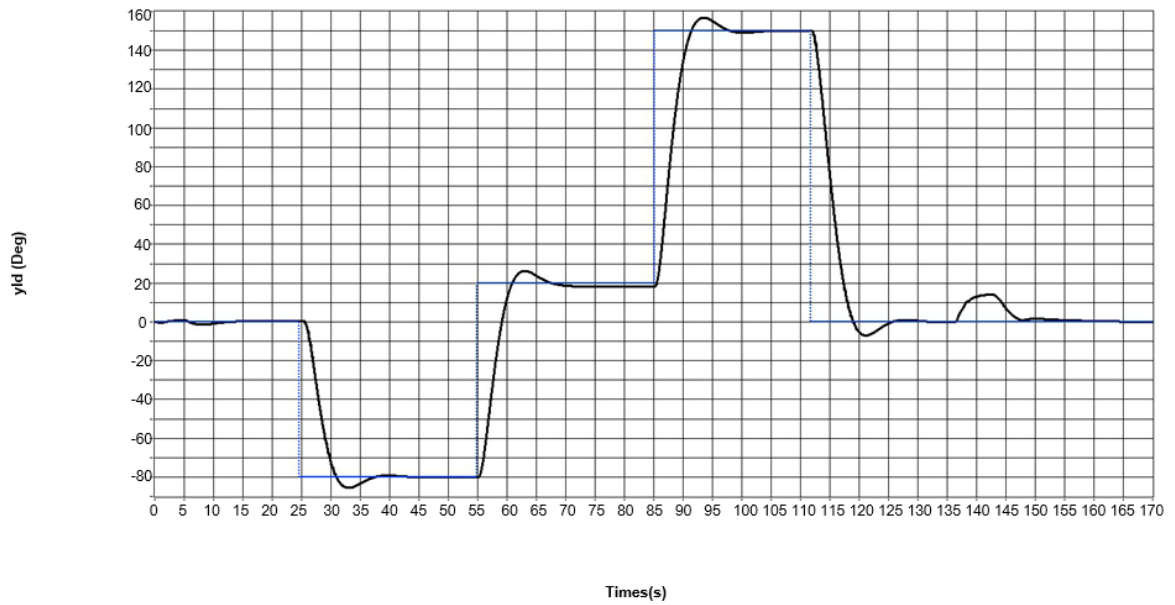


Fig. 6.42: AUV response (Black) to direction angle command (Blue).

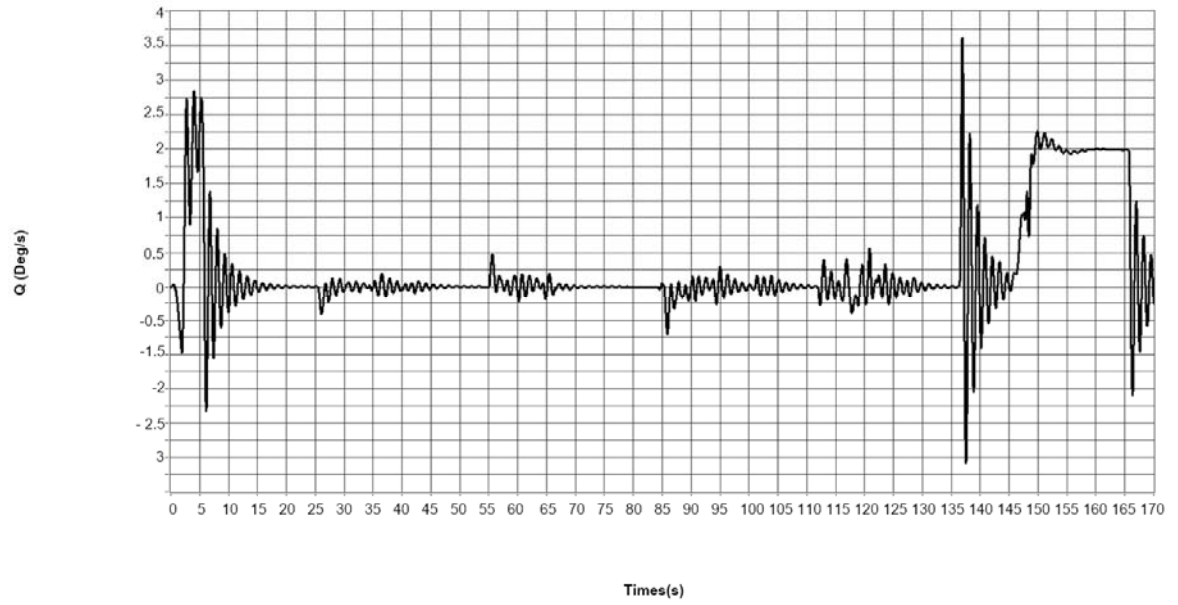


Fig. 6.43: Angular velocity around x-axis.



Fig. 6.44: Angular velocity around y-axis.

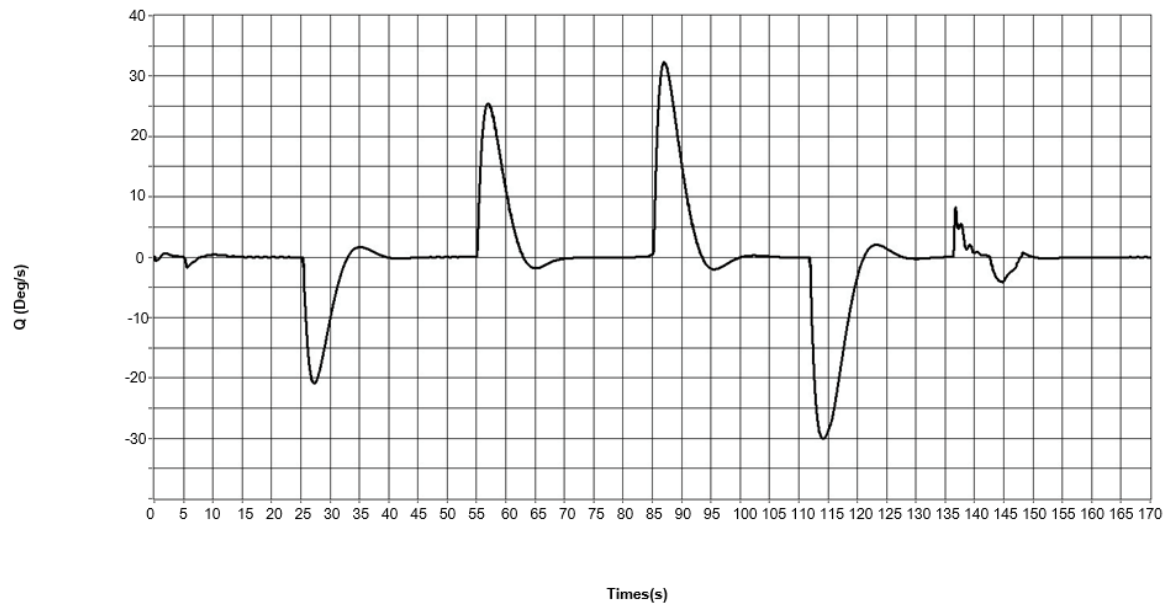


Fig. 6.45: Angular velocity around z-axis.

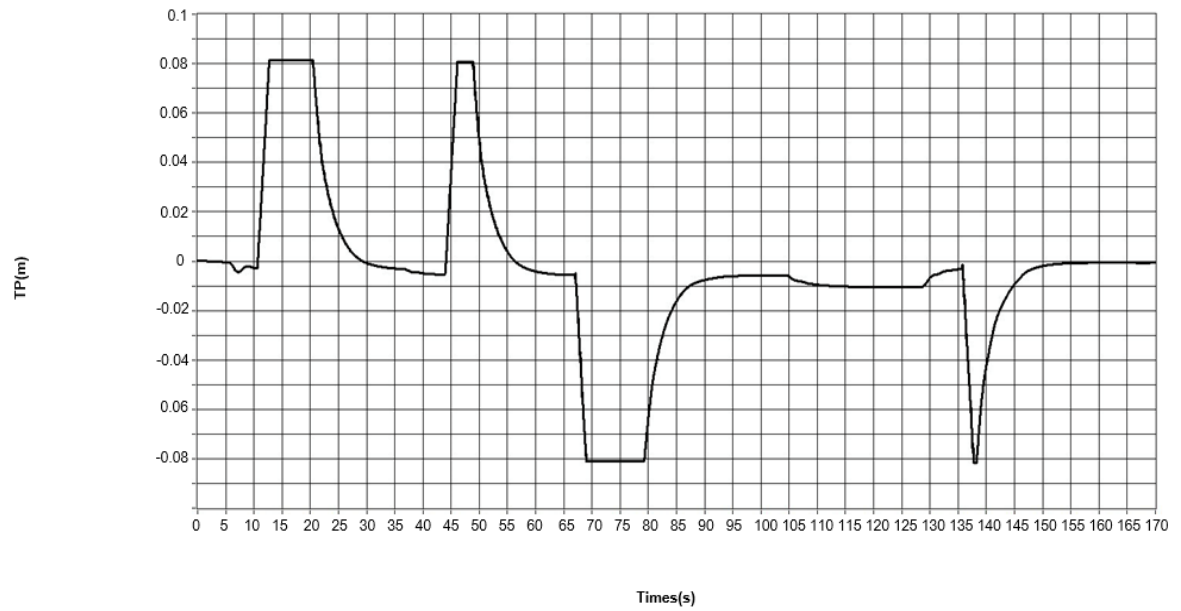


Fig. 6.46: The status of the moving weight.

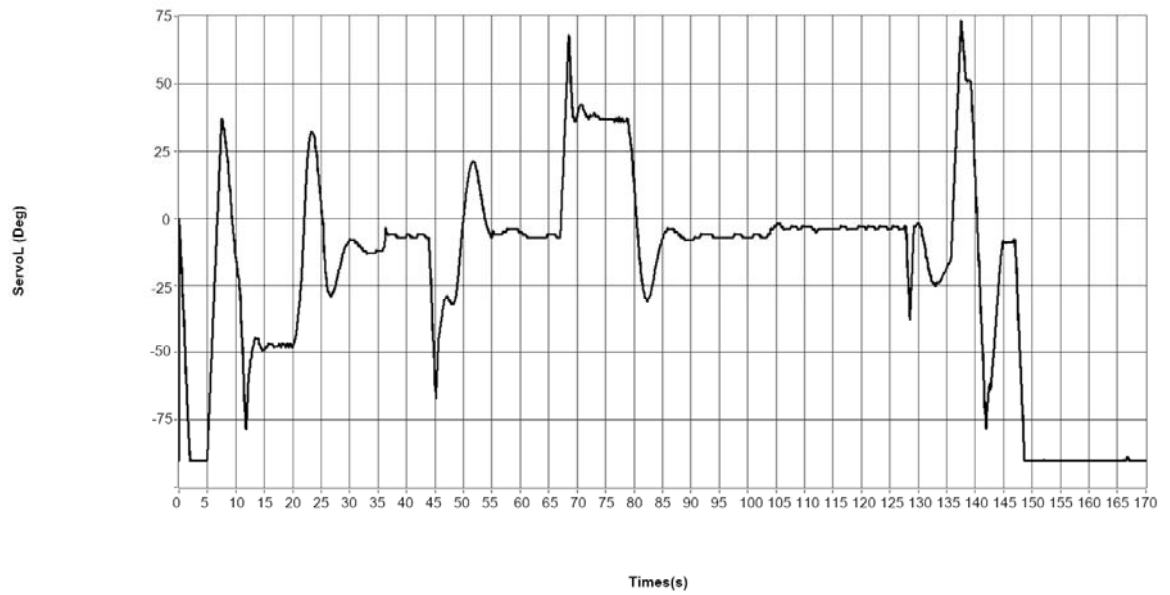


Fig. 6.47: Left servo angle.

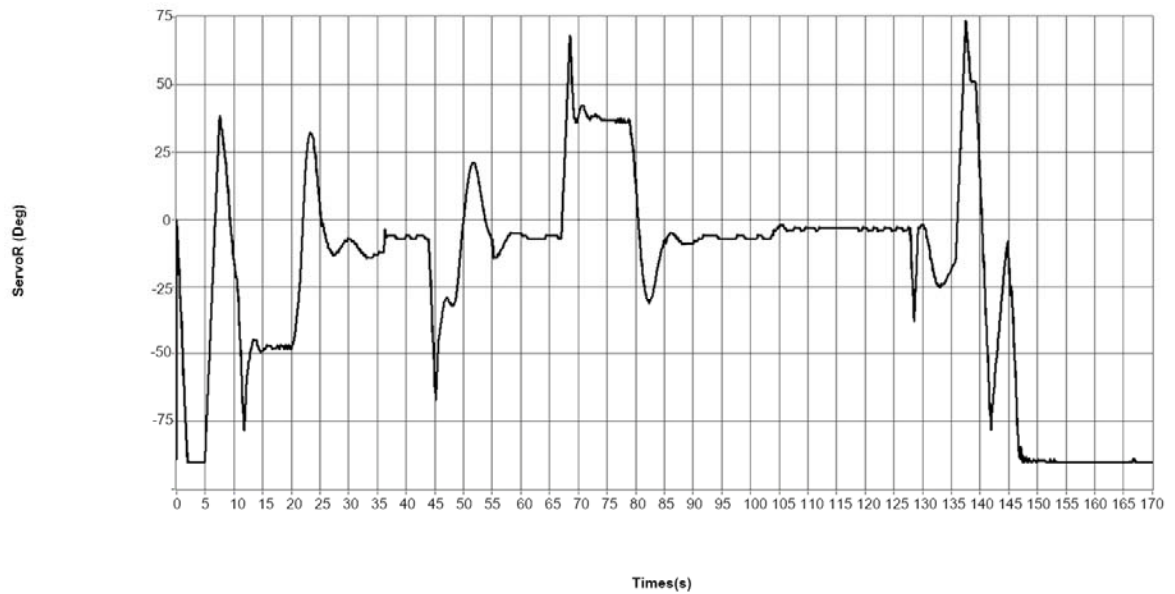


Fig. 6.48: Right servo angle.

6.5 Breakthrough

At this stage, the control process was validated. Then, we added two more features to make it relatively autonomous, i.e., a Fuzzy-based control decision system and a central control

signal system. In the control decision stage, a phase system should take the best decision for the next move after diagnosing an obstacle ahead. Then the decision is transmitted to the central unit, which send the required commands to the AUV engines for proper movement.

6.5.1 Fuzzy Logic Controller (FLC)

Fuzzy Logic was developed by Dr. Zadeh in 1960 [166]. Unlike others who believed that approximations could decrease a system performance, Dr. Zadeh demonstrated that model ambiguity is a fundamental part of an engineering system.

Fuzzy systems are knowledge- or rule-based systems. The heart of a fuzzy system is composed of fuzzy IF-THEN rules. A fuzzy if-then rule is an if-then term if its words are determined by continuous membership functions [166]-[169].

A fuzzy controller is composed of four main parts: Fuzzifier, rules base, decision making part and Defuzzifier (Figure 6.49). Note that before and after the fuzzy controller, preprocessor- and post-processor are usually included.

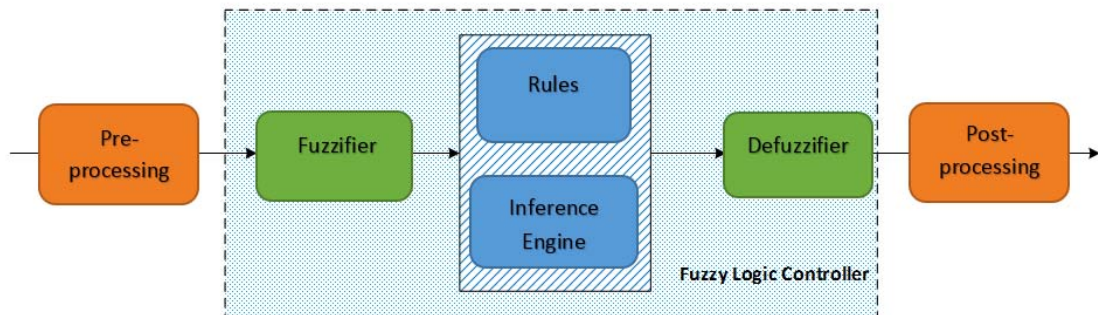


Fig. 6.49: Block Diagram of a fuzzy controller structure.

In this structure, the input is first presented to the Fuzzifier. The Fuzzifier then relates a value between 0 and 1 to the input based on the defined membership functions. This value is determined according to the linguistic rule as if-then term and obtains a value known as

fire strength. These values are combined through numerical methods in the inference engine and build a number as the output of the fuzzy system. Finally, the value is Defuzzified and presented to the next level as a significant output.

6.5.2 Fuzzy Controller Design

To apply fuzzy logic to obstacle detection, each picture is first subdivided into three horizontal/vertical bands and a score assigned to each of the six bands. This number is equal to the ratio of white pixels to the total number of pixels in the section. To prevent noise error in image processing, we set a threshold: if this ratio is less than 2% for the whole image, this later is assumed free of obstacles.

To illustrate this concept, let us consider the image in Figure 6.50.

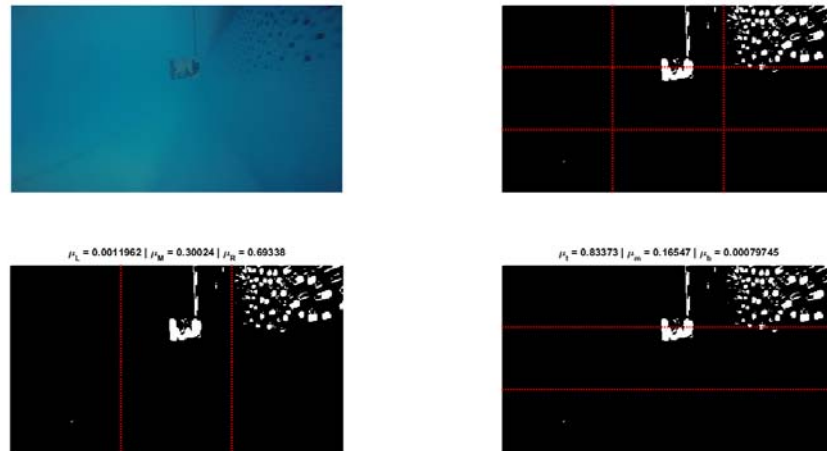


Fig. 6.50: An example of image scoring system. The original image (Top left) is subdivided into six sections (Top right), i.e., three vertical bands (Bottom left) and three horizontal bands (Bottom right).

Then to set the fuzzy system inputs, each score is associated with a parameter as follows:

- μ_L for the left vertical band,
- μ_M for the middle vertical band,

- μ_R for the right vertical band,
- μ_t for the top horizontal band,
- μ_m for the middle horizontal band, and
- μ_b for the bottom horizontal band.

The fuzzy system decision is based on the fact that each band is compared to its corresponding middle band. For example, let us start with the vertical bands. If the result of the term $\mu_L - \mu_M$ is negative, the vertical middle part obtains a higher score in detecting an obstacle and a higher share than the left band in terms of placement.

Thus, the left band might be a better choice than the vertical middle band for the AUV's next movement to contour the obstacle. At the same time, the term $\mu_L - \mu_M$ is calculated and the fuzzy system makes its decision to finally determine the Yaw direction of the AUV.

The decision output is the speed difference between the left and right engine. The number provided by the fuzzy system is reduced from the average speed of the left DC engine and added to the right DC engine. The fuzzy rules are presented in Table 6.3 with:

PB: Positive Big,
 PS: Positive Small,
 P: Positive,
 Z: Zero,
 N: Negative,
 NS: Negative Small, and
 NB: Negative Big.

The decision table is designed such that the AUV will select the direct track. It means that if the side bands are not significantly different from the middle band in terms of scoring, the direct track is the AUV's next decision to move. Accordingly, sudden and large movements in one direction are prevented and waste commands reduced.

Input and output membership functions of the fuzzy decision system are given in Figures 6.51 and 6.52. Note that such input/output membership functions have been set after performing preliminary tests on known images.

Table 6.3: Fuzzy rules to decide dc motors performance decision to move the AUV in yaw degrees of freedom (**P**: Forward rotation of the motors, **Z**: OFF, **N**: Backward rotation of the motors).

		$\mu R-\mu M$				
		PB	PS	Z	NS	NB
$\mu L-\mu M$	PB	Z	Z	Z	Z	P
	PS	Z	Z	Z	P	P
	Z	Z	Z	Z	P	P
	NS	Z	N	N	P	P
	NB	N	N	N	N	P

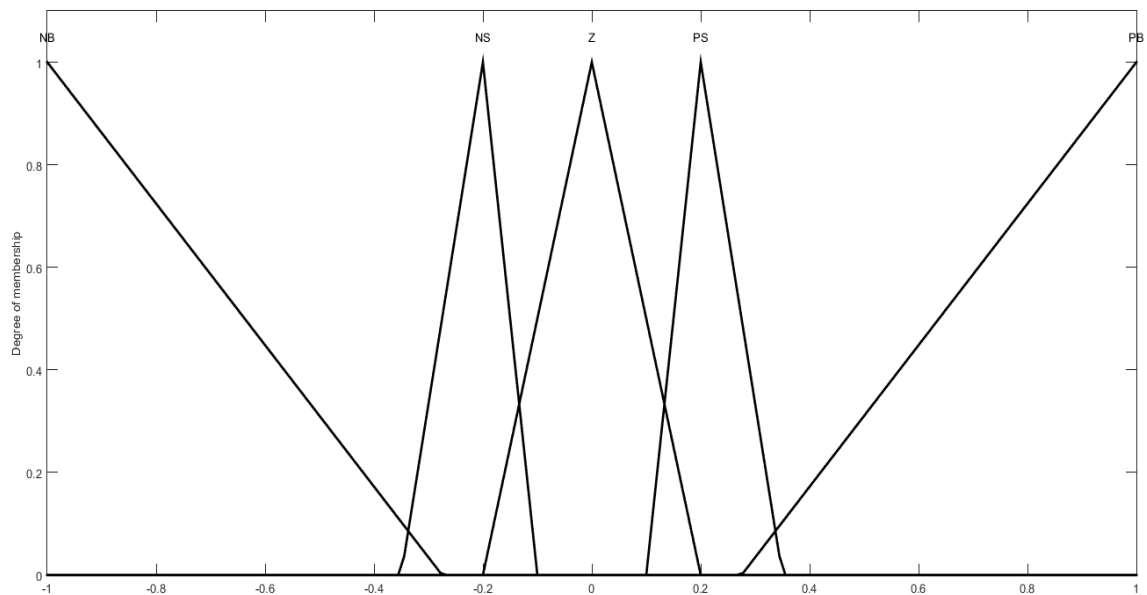


Fig. 6.51: Input membership functions for fuzzy systems of DC engines.

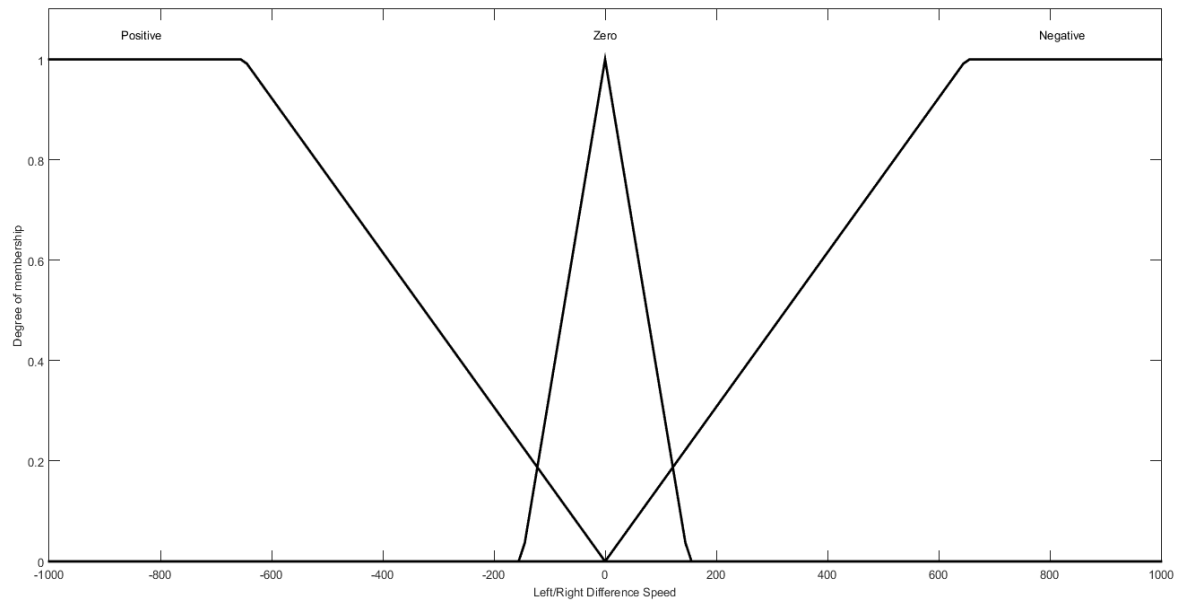


Fig. 6.52: Output membership functions for fuzzy systems of DC engines.

Similarly, we implemented a fuzzy controller input to quantify the scoring between the horizontal bands. After performing the fuzzy operation, the fuzzy system output is an angle to command the servo motor angles (Table 6.4).

Table 6.4: Fuzzy rules for decision performance of servomotors to move the AUV in a predefined pitch angle (Negative numbers indicate downward motion and positive numbers indicate upward movement of servo motors).

		$\mu_b - \mu_m$				
		PB	PS	Z	NS	NB
$\mu_t - \mu_m$	PB	0°	0°	0°	-20°	-60°
	PS	0°	0°	0°	-40°	-60°
	Z	0°	0°	0°	-40°	-60°
	NS	+20°	+40°	+40°	+40°	-60°
	NB	+60°	+60°	+60°	+60°	+60°

Similar to the previous fuzzy system, the rules of the fuzzy controller have been designed to guide the AUV. Input/output membership functions are shown in Figures 6.53 and 6.54.

According to the mentioned control process, the control strategy can be summarized in Figure 6.55 while Table 6.5 summarizes the fuzzy systems' features.

Table 6.5: The Fuzzy systems' features [166],[169].

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

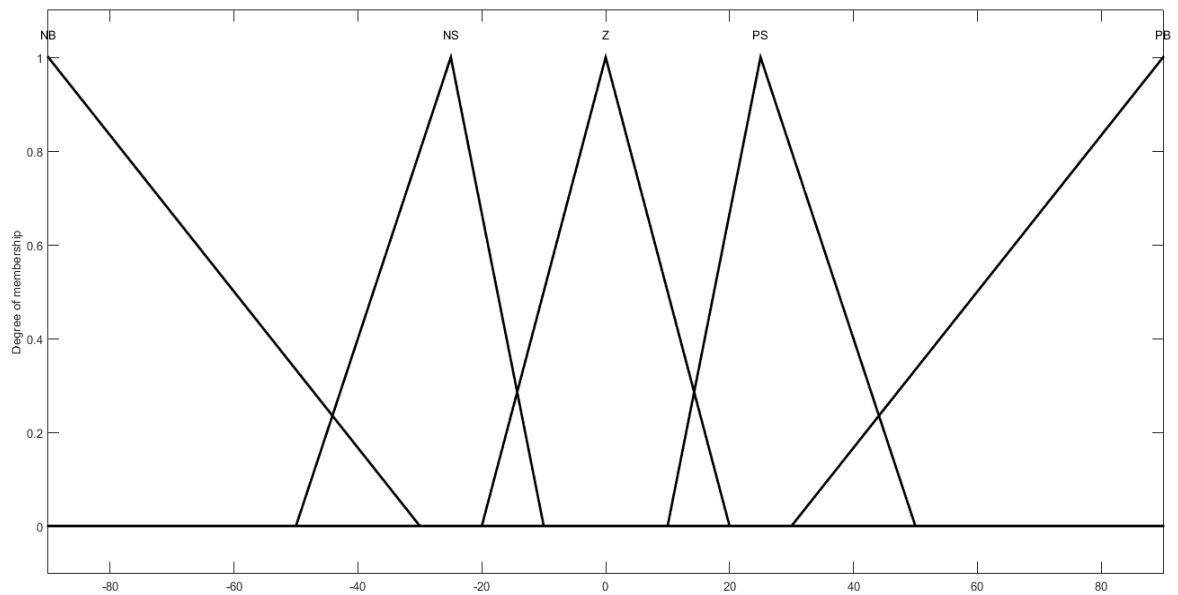


Fig. 6.53: Input membership functions for servo motors' fuzzy system.

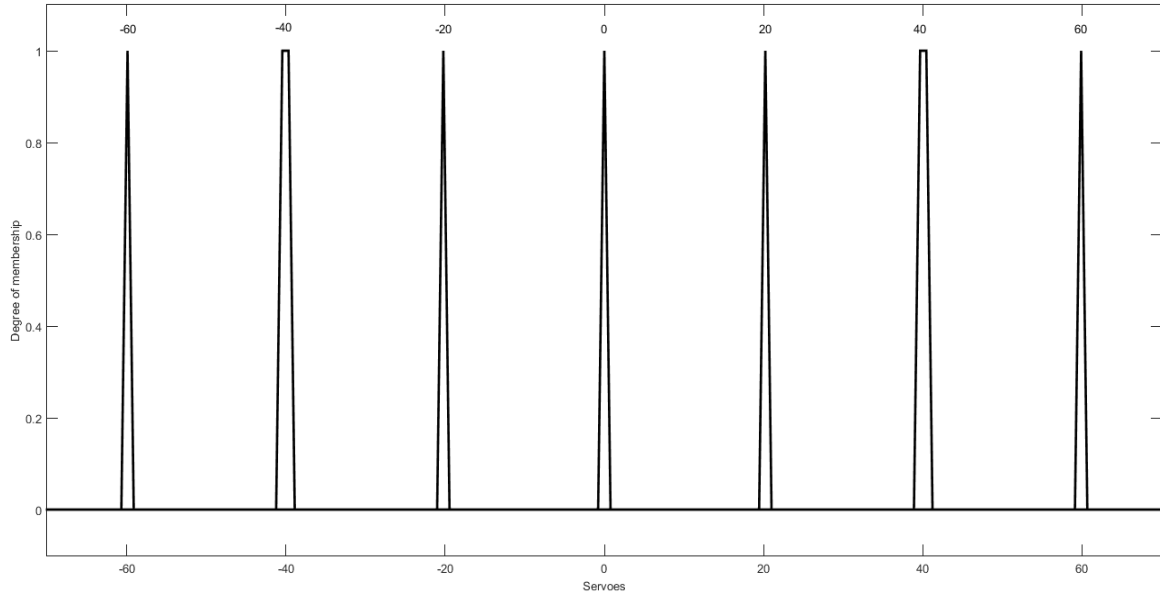


Fig. 6.54: Output membership functions for servo motors' fuzzy system.

6.6 Vision Tests

6.6.1 First Test

As shown in Figure 6.56, the AUV is moving in the pool with a speed of 0.2 m/s and a side angle of 0° at a depth of 2 m. According to the images kept by the AUV and illustrated in Figure 6.57, the AUV did not detect an obstacle and then followed the path determined by the compass sensor. However, due to an external turbulence from an unsought wave, it had to turn left 45° , at $t = 1$ s.

Despite this unexpected problem, the controller response was adequate. It was indeed anticipated that the controlling algorithm dispels this turbulence by sending suitable commands and the AUV be returned to its stable side angle of 0° .

Also, at $t = 11$ s, the algorithm took appropriate action to stabilize the AUV: it rolled around its longitudinal axis by about an angle of 30° s shown in Figures 6.58 to 6.67.

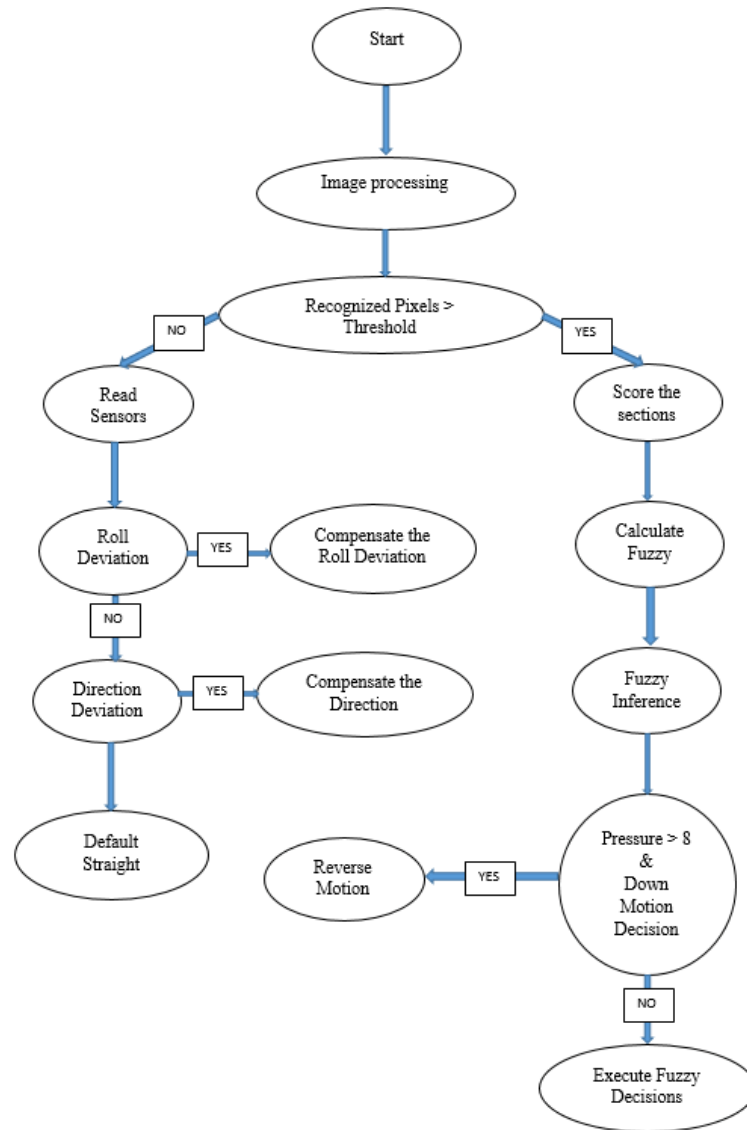


Fig. 6.55: Control algorithm flowchart.



Fig. 6.56: A sample of control and display unit when the AUV at a depth of 2m.

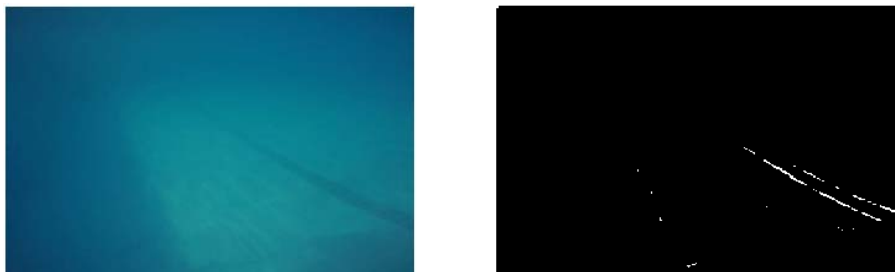


Fig. 6.57: Image recognition without obstacle.

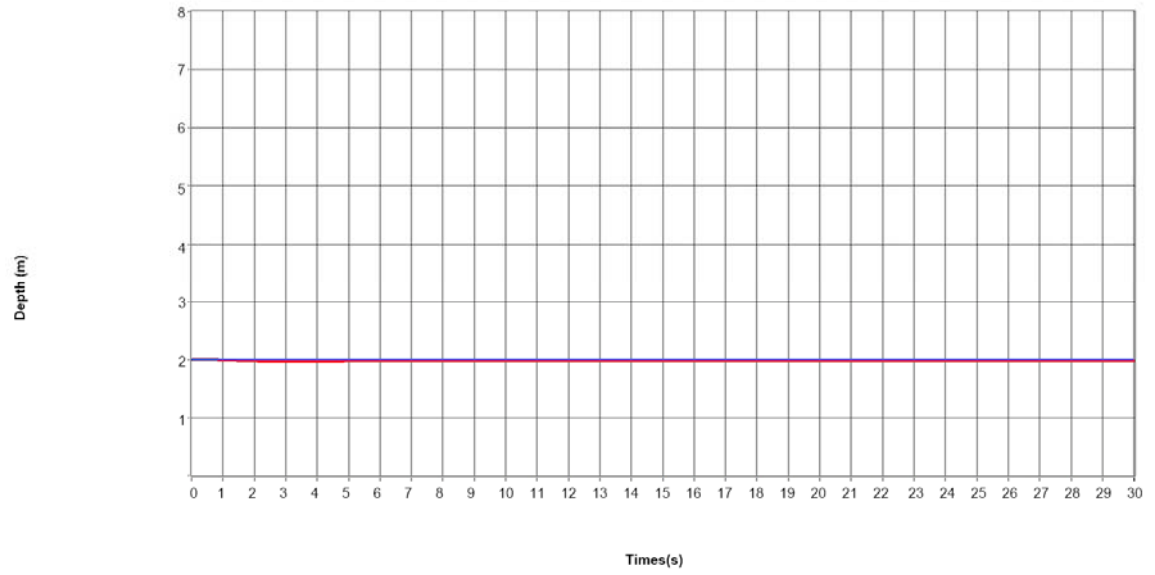


Fig. 6.58: Depth evolution vs. time.

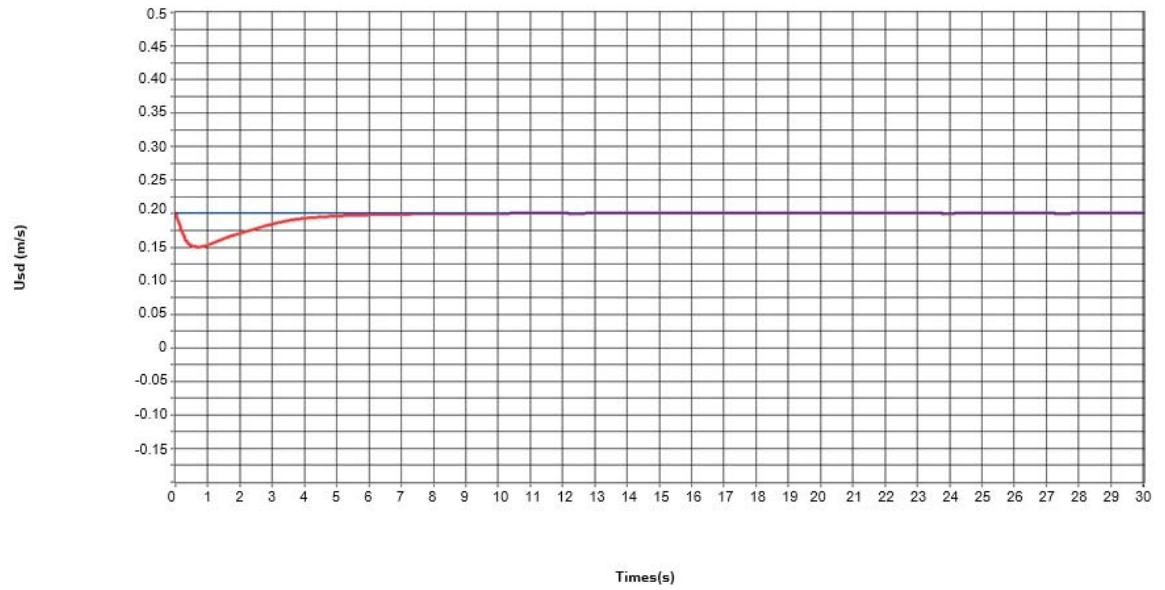


Fig. 6.59: Speed evolution vs. time.

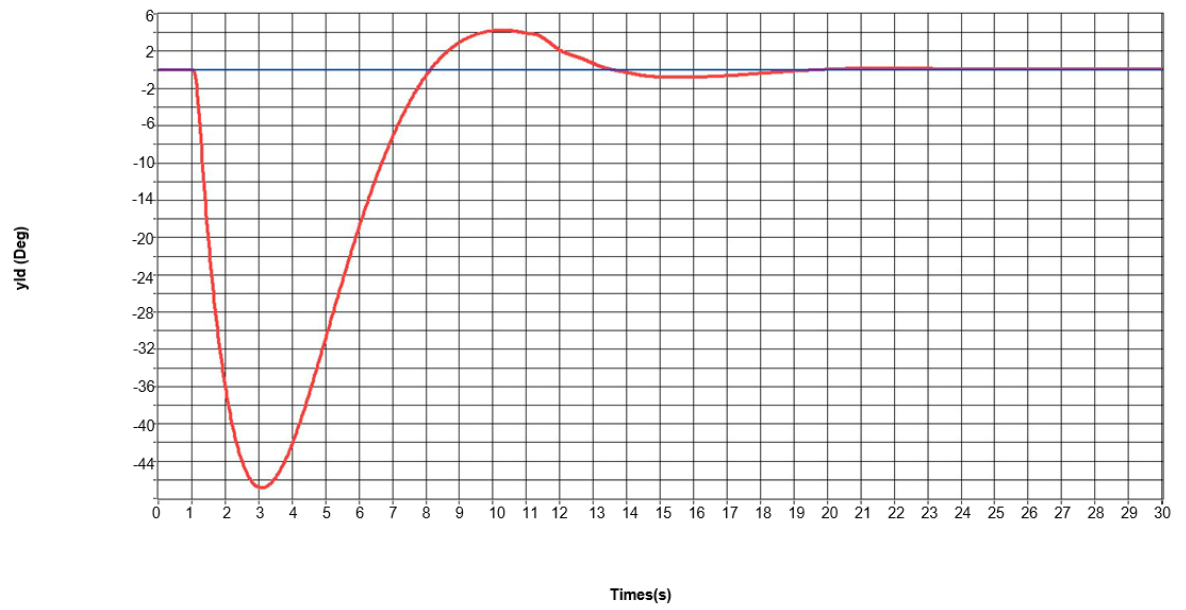


Fig. 6.60: Direction angle.

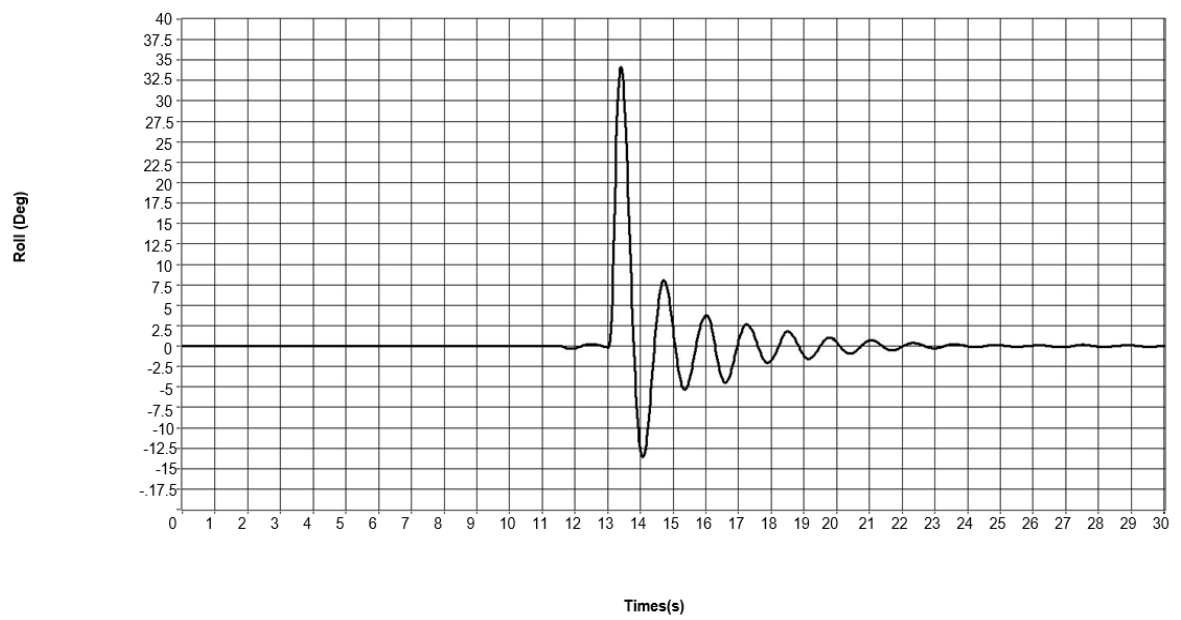


Fig. 6.61: Roll angle.

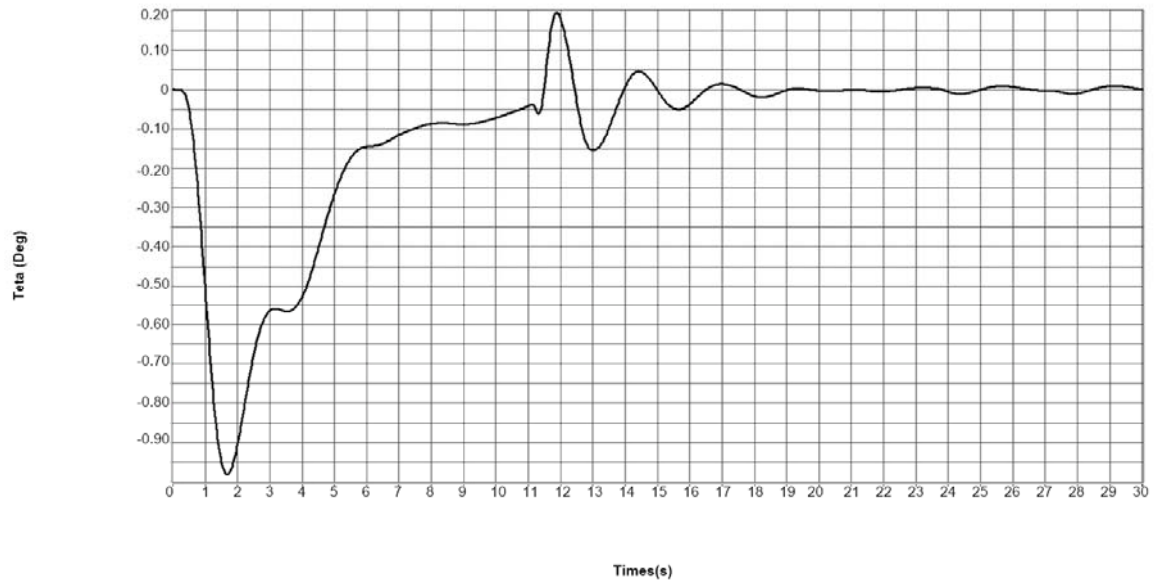


Fig. 6.62: Pitch angle.

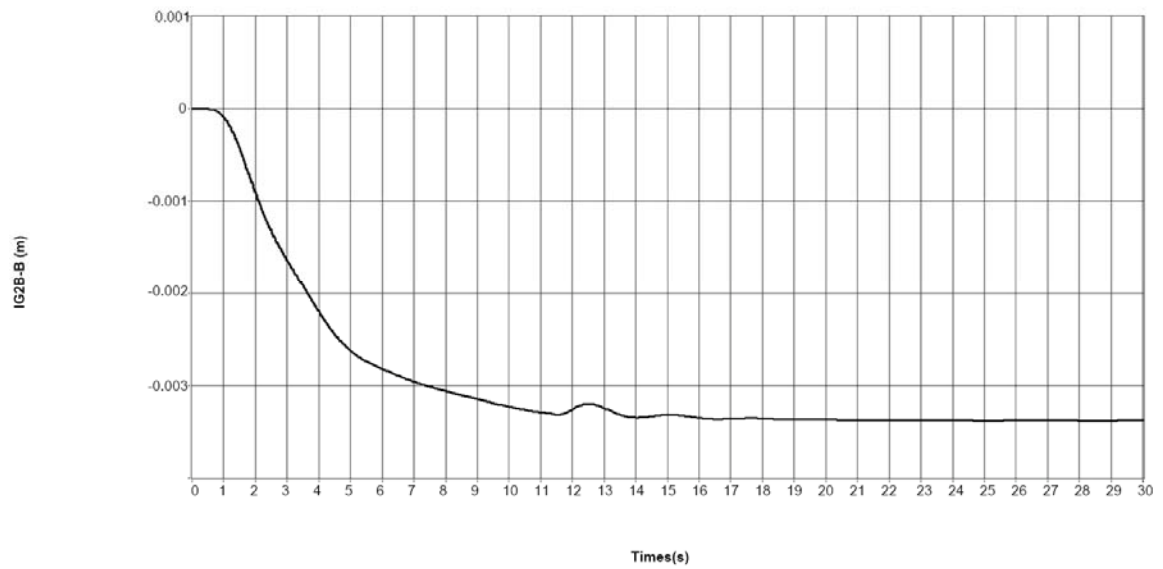


Fig. 6.63: Mass shifter movement.

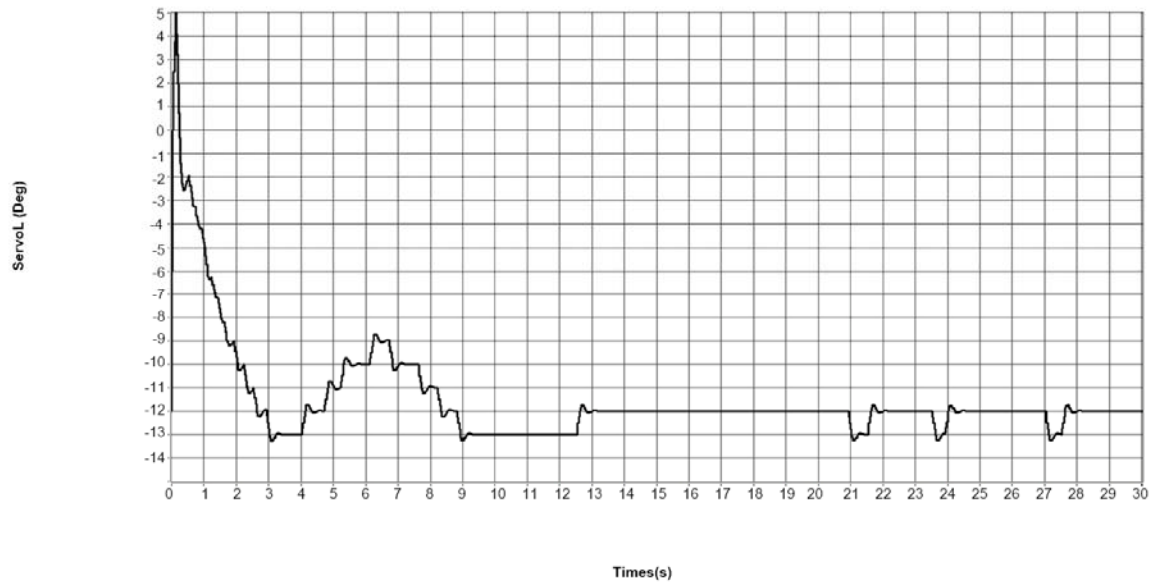


Fig. 6.64: Left servo angle.

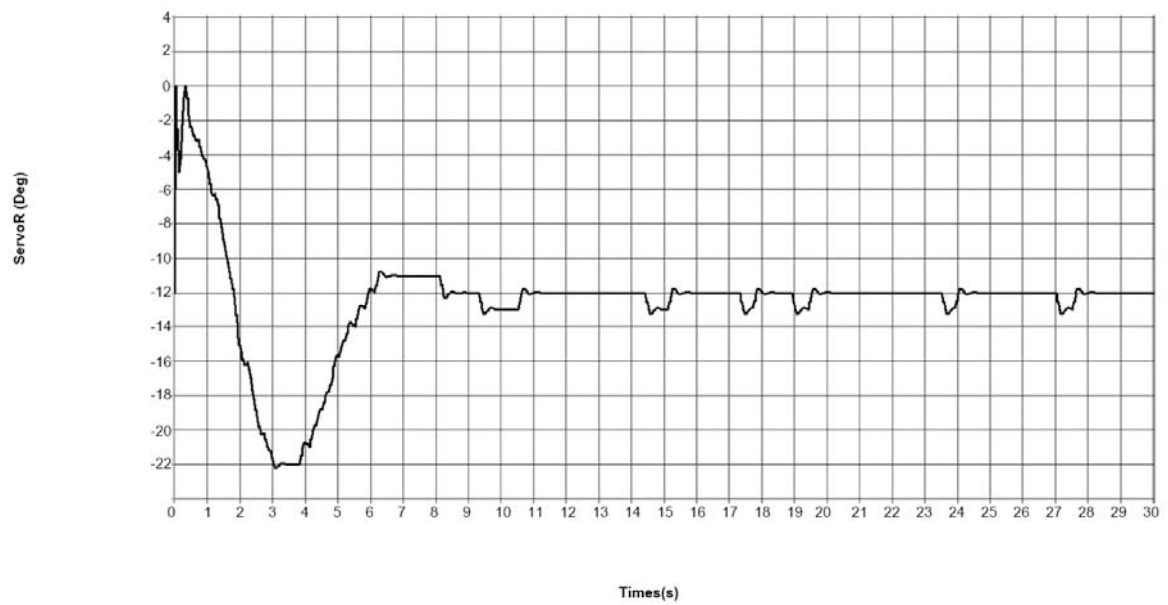


Fig. 6.65: Right servo angle.

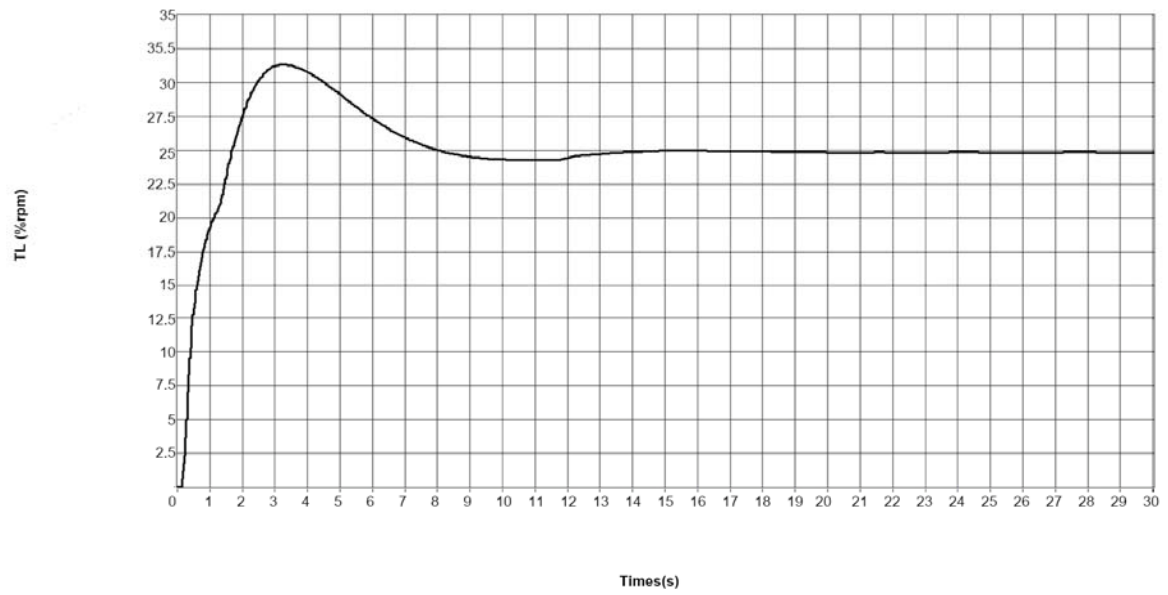


Fig. 6.66: Left engine speed.

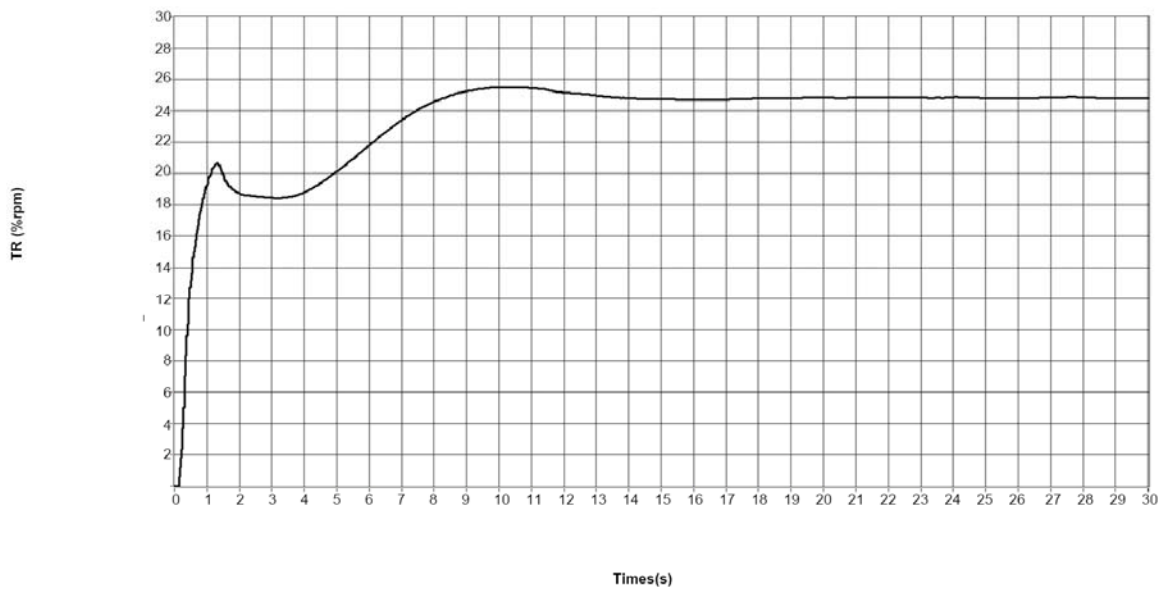


Fig. 6.67: Right engine speed.

6.6.2 Second Test

As shown in Figure 68, the AUV command was set to a speed of 0.3 m/s, a direction angle of 60° and at a depth of 1m. The main difference is now the presence of an obstacle (Figure

6.69). As shown in Figures 6.70 to 6.76, the AUV decided to slightly turn toward right and select a greater depth (1.5 m). Then, after 5s, it returned to its initial path but at a depth of 1.5 m.



Fig. 6.68: A sample of control and display unit when the AUV at a depth of 1m.

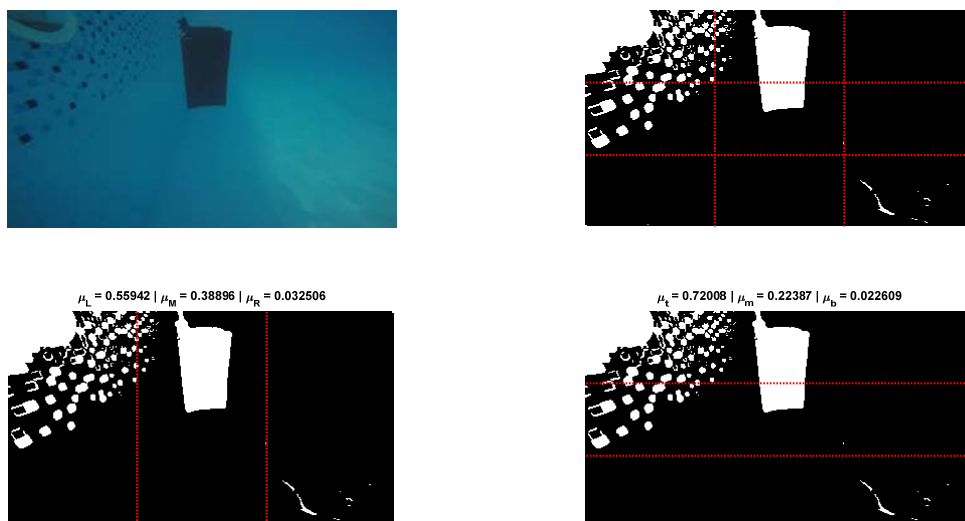


Fig. 6.69: Detection of an obstacle.

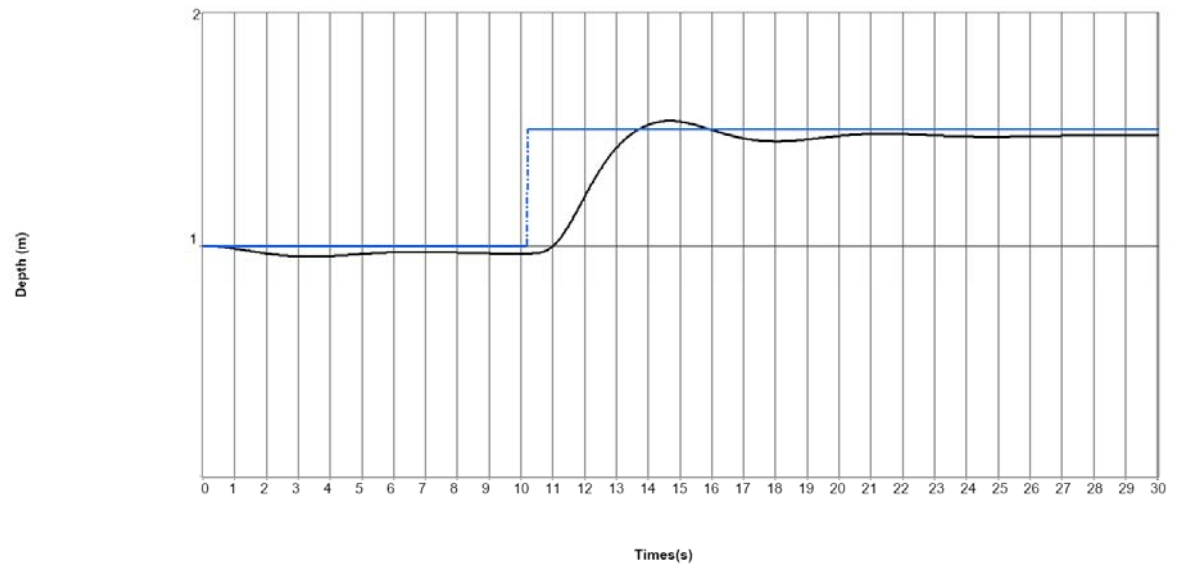


Fig. 6.70: AUV response (Black) to depth change (Blue).

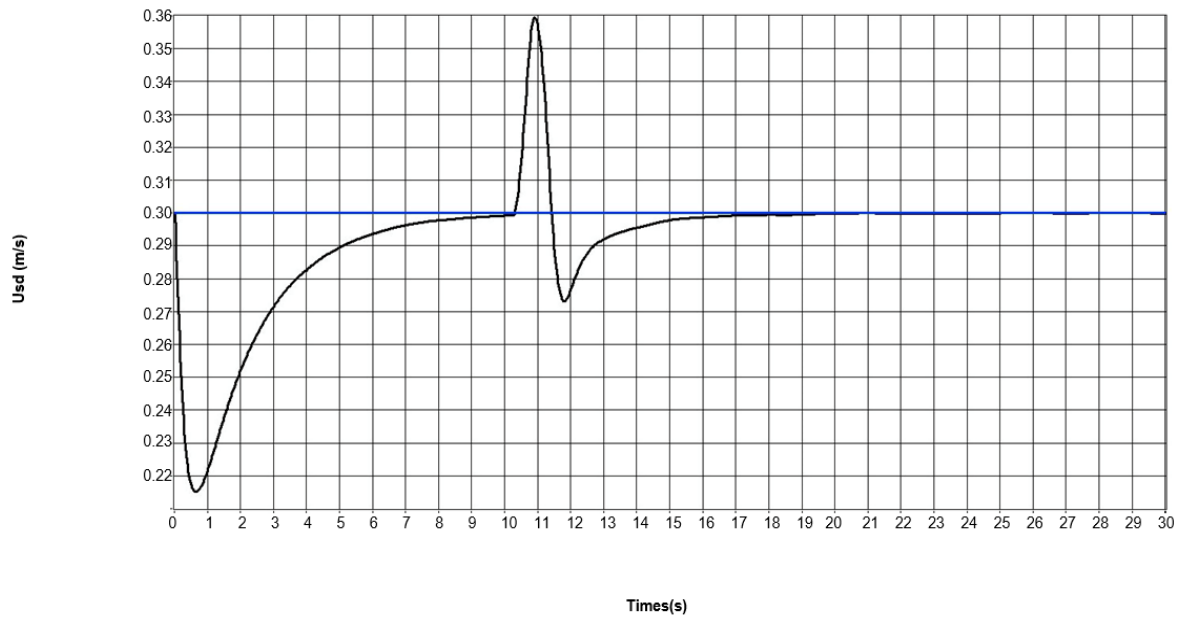


Fig. 6.71: Speed variation.

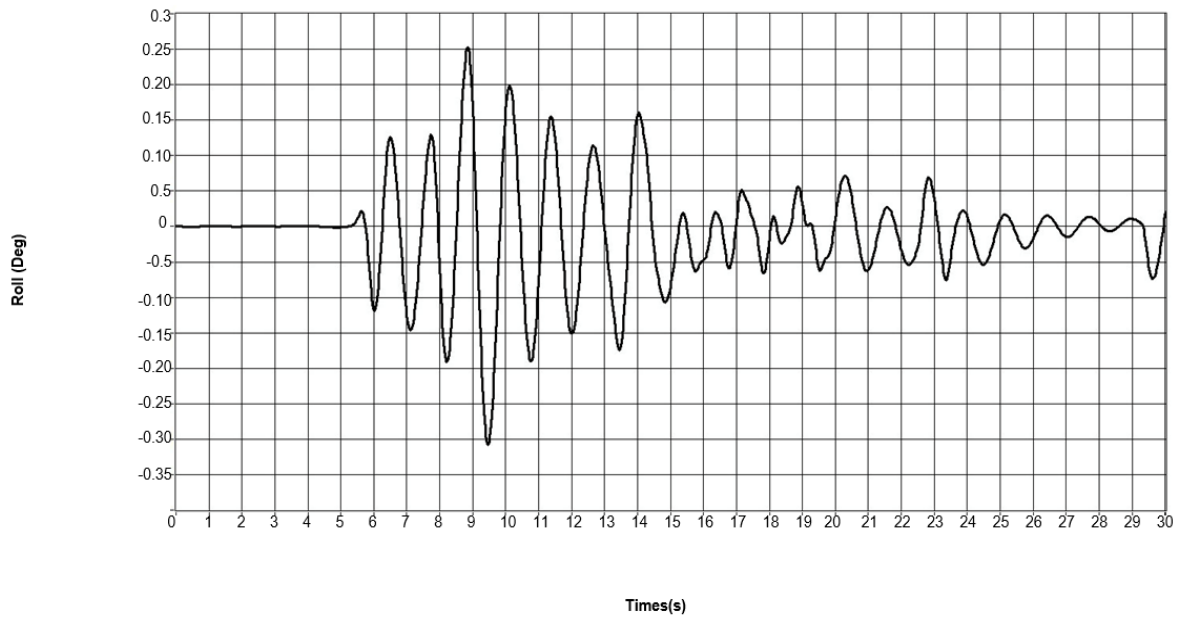


Fig. 6.72: Roll angle variation.

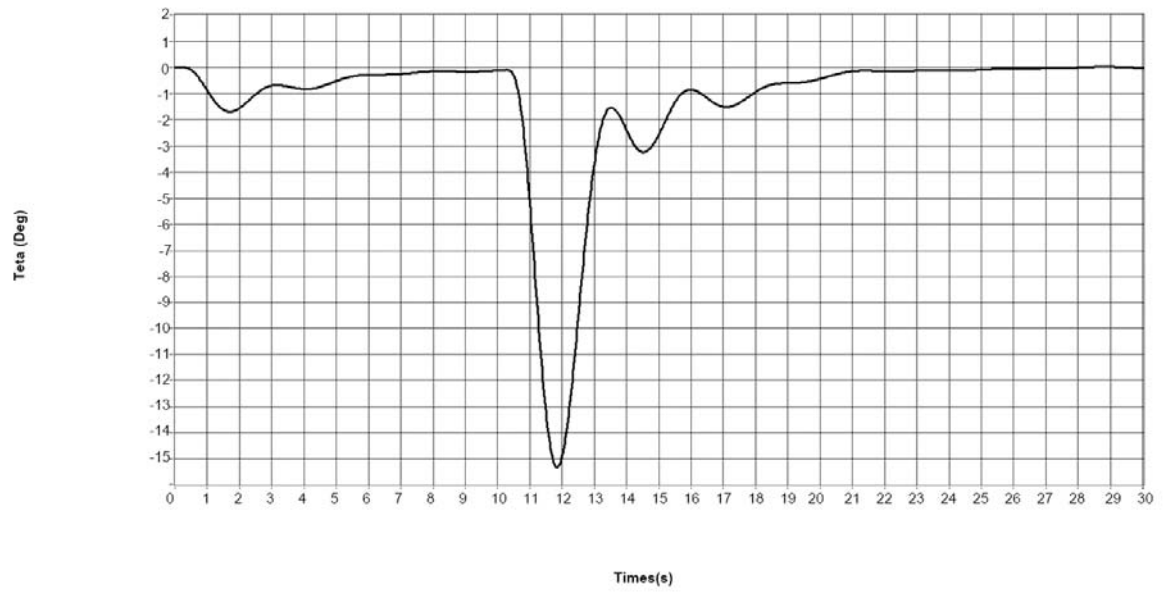


Fig. 6.73: Pitch angle variation.

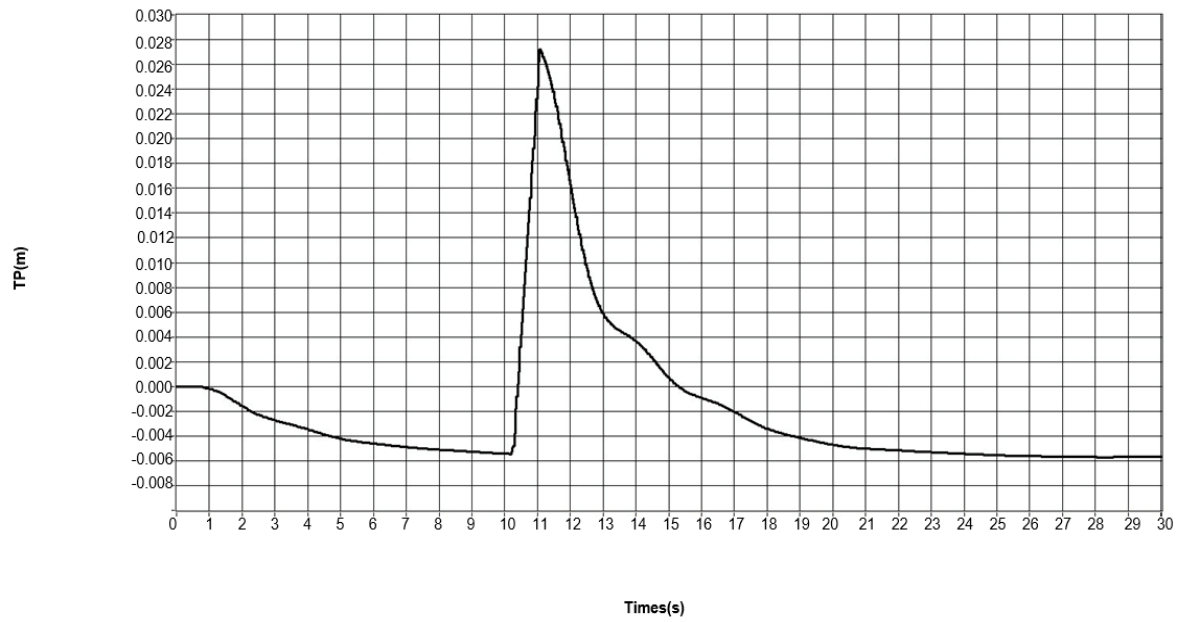


Fig. 6.74: Mass shifter variation.

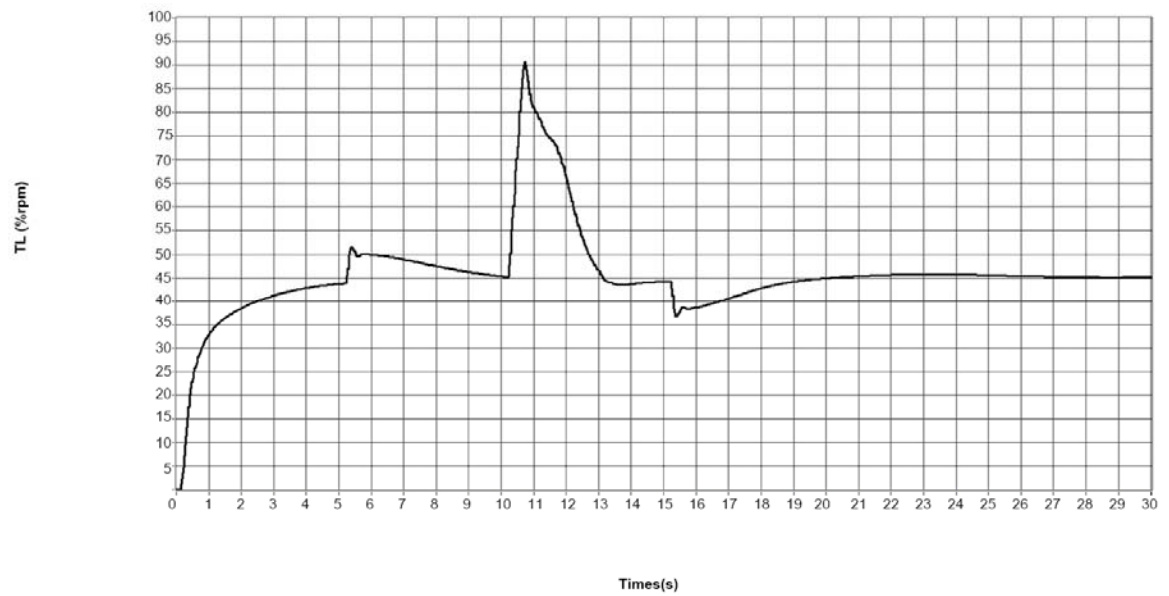


Fig. 6.75: Left engine speed variation.

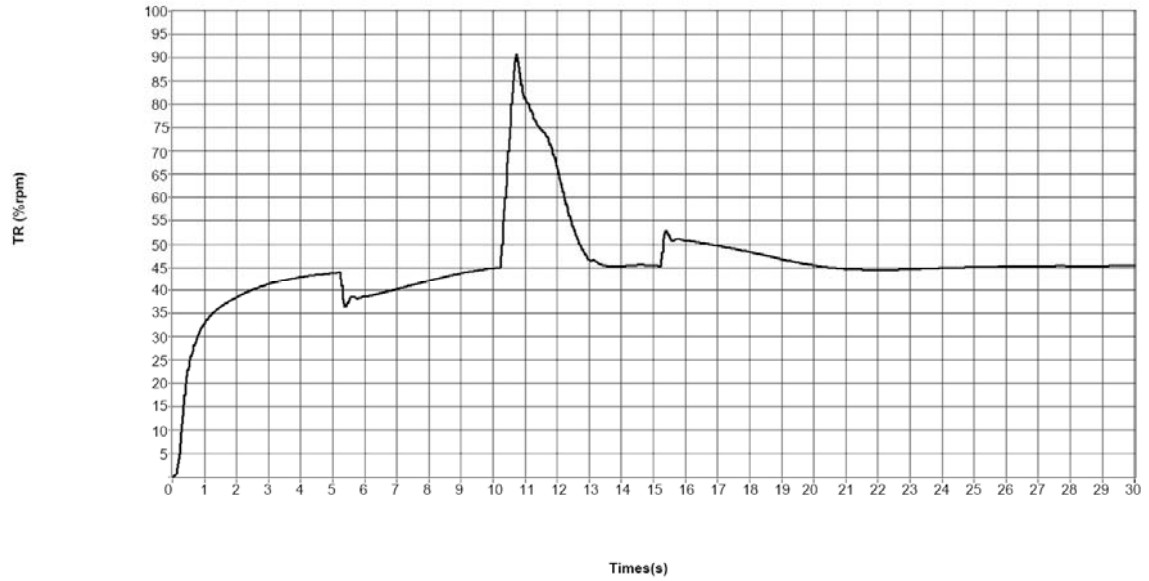


Fig. 6.76: Right engine speed variation.

6.6.3 Third Test

As shown in Figure 6.77, the AUV is now moving with a speed of 0.1 m/s and a direction angle of -40° in 1.5m depth. Once an obstacle has been detected (Figure 6.78), the image scoring system response is summarized in Table 6.6. As shown in Table 6.7 and Figures 6.79 to 6.84, after passing the obstacle, at $t = 11$ s, the AUV went toward right at a depth of 2 m with a little increasing in the speed. At $t = 13$ s, the sensor reported a change in the angle (30°) in the roll direction, thus the AUV pitched to the left side (at an angle of about 30°) to return to its initial path.

We can also note the controlling effort to compensate the turbulence. At $t = 16$ s, a turbulence in the Yaw direction occurred and the AUV strayed to its right side, to finally returning to the desired operation conditions.



Fig. 6.77: A Sample of control and display unit when the AUV at a depth of 1.5 m.

Table 6.6: Image scoring system for this test.

μ_L	μ_M	μ_R	$\mu_L - \mu_M$	$\mu_R - \mu_M$	μ_t	μ_m	μ_b	$\mu_t - \mu_m$	$\mu_b - \mu_m$
0.5812	0.4121	0.0067	0.1692	-0.4053	0.7754	0.2181	0.0065	0.5573	-0.2116
0.5809	0.4068	0.0123	0.1741	-0.3946	0.7597	0.2284	0.0120	0.5313	-0.2164
0.4368	0.5491	0.0140	-0.1123	-0.5351	0.7094	0.2772	0.0134	0.4321	-0.2638
0.3412	0.6495	0.0093	-0.3083	-0.6402	0.6839	0.3072	0.0089	0.3767	-0.2983
0.3552	0.6274	0.0174	-0.2722	-0.6099	0.6366	0.3491	0.0144	0.2875	-0.3347
0.6241	0.3534	0.0226	0.2707	-0.3308	0.6232	0.3595	0.0172	0.2637	-0.3423
0.8590	0.1319	0.0091	0.7271	-0.1228	0.5745	0.4170	0.0085	0.1576	-0.4085
0.9819	0.0041	0.0140	0.9778	0.0099	0.5111	0.4807	0.0082	0.0305	-0.4724
0.9817	0.0041	0.0142	0.9776	0.0102	0.3713	0.6180	0.0107	-0.2467	-0.6073
0.9133	0.0114	0.0753	0.9663	0.0206	0.2215	0.7394	0.0392	-0.5179	-0.7002
0.7436	0.0688	0.1876	0.9018	0.0639	0.0203	0.9098	0.0699	-0.8894	-0.8398

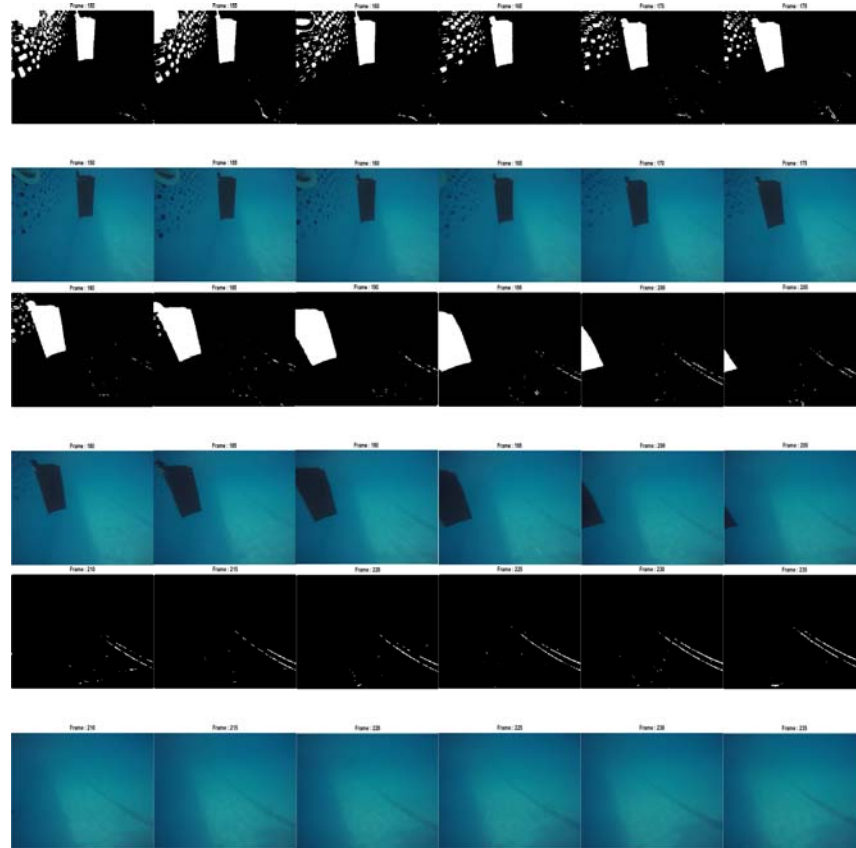


Fig. 6.78: AUV performance and obstacle detection in the third test.
Original images: blue background, processed images: black and white.

Top set: The AUV detects an obstacle on its way,

Middle set: According to the fuzzy decision, the AUV turns right,

Third set: no more obstacle in front of the AUV and the AUV can move forwards.

Table 6.7: Fuzzy decision system inputs.

Time (second)	t = 0	t = 11	t = 13	t = 15	t = 16
Speed (m/s)	0.1	0.2	Turbulence in direction of roll degree equals to +30°	0.2	Turbulence in direction of side degree +40°
Depth (m)	1.5	1.5		2	
Direction angle (°)	-40	+10		+10	

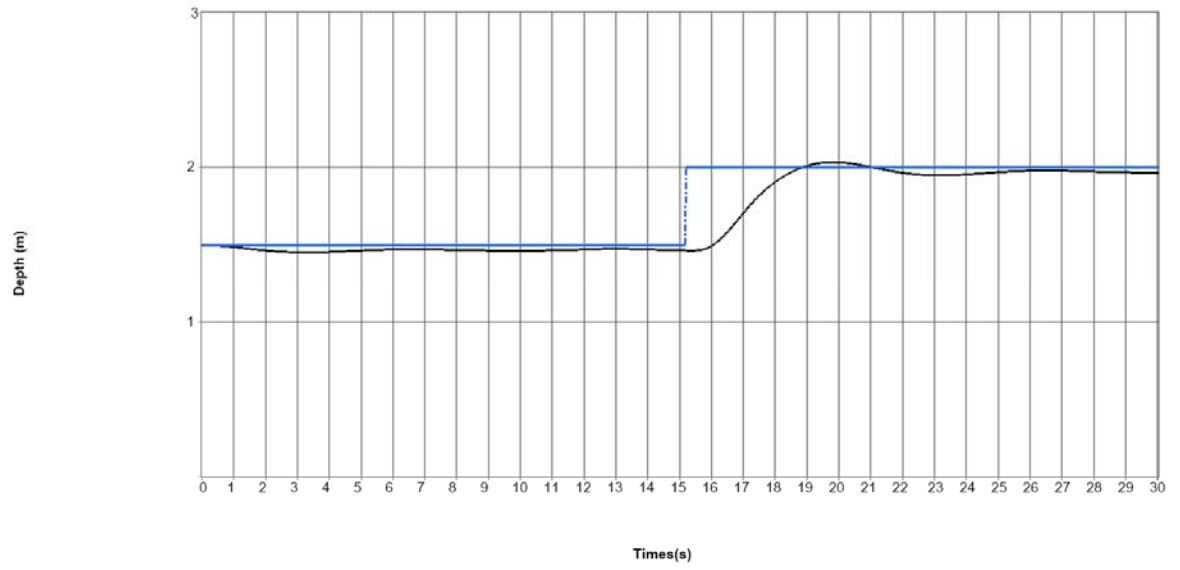


Fig. 6.79: AUV response (Black) to depth change (Blue).

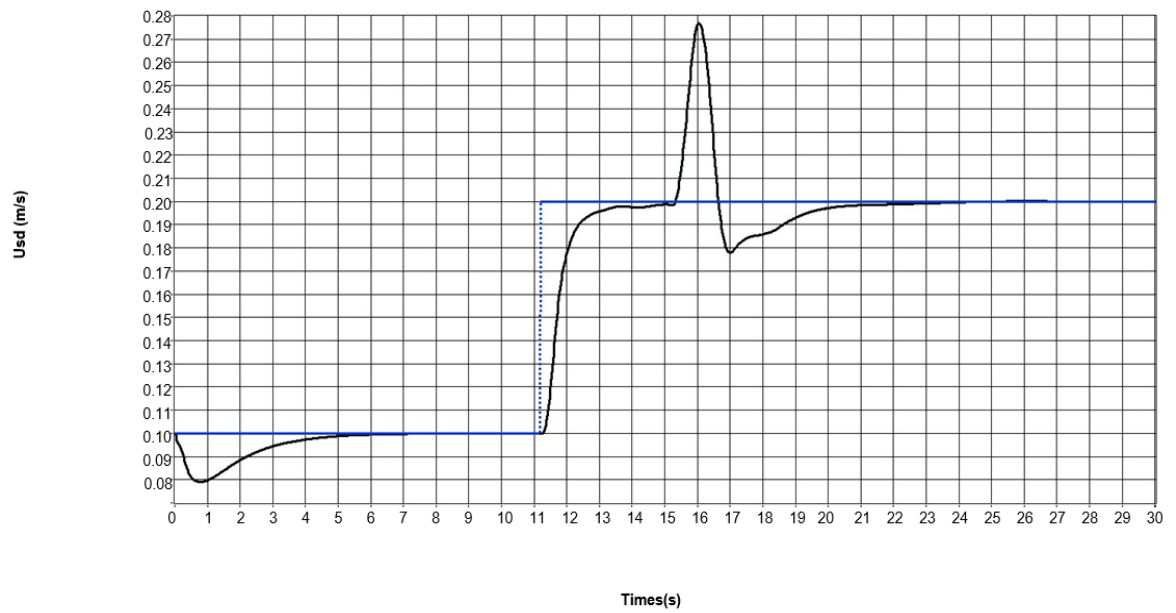


Fig. 6.80: AUV response (Black) to speed change (Blue).

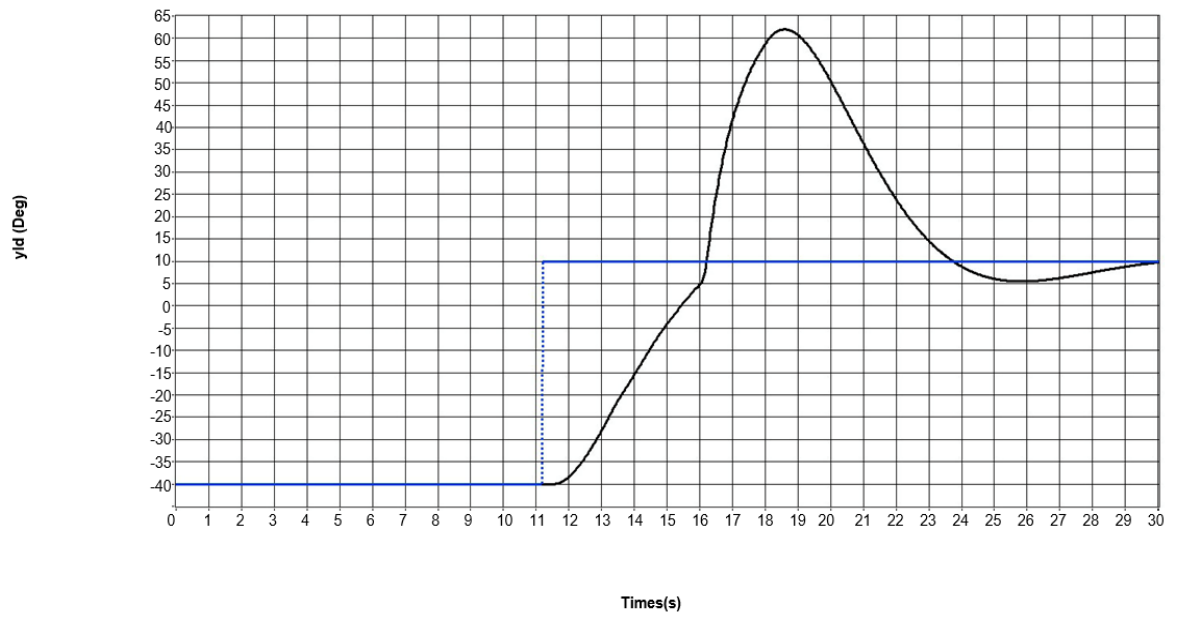


Fig. 6.81: AUV response (Black) to direction angle change (Blue).

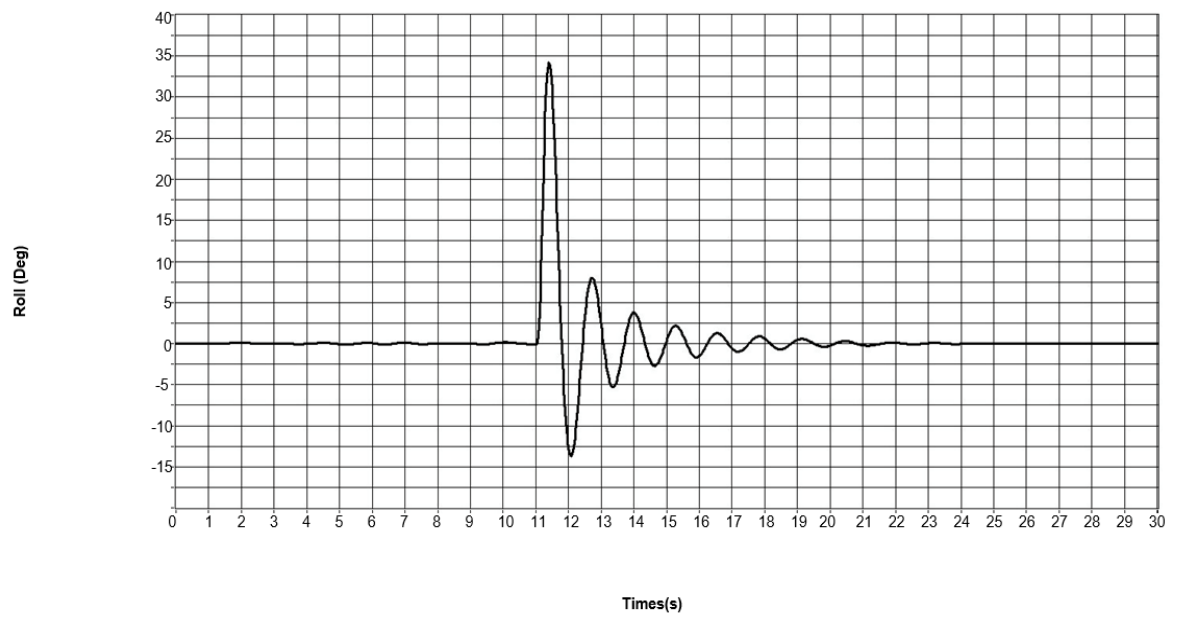


Fig. 6.82: Roll angle variation.

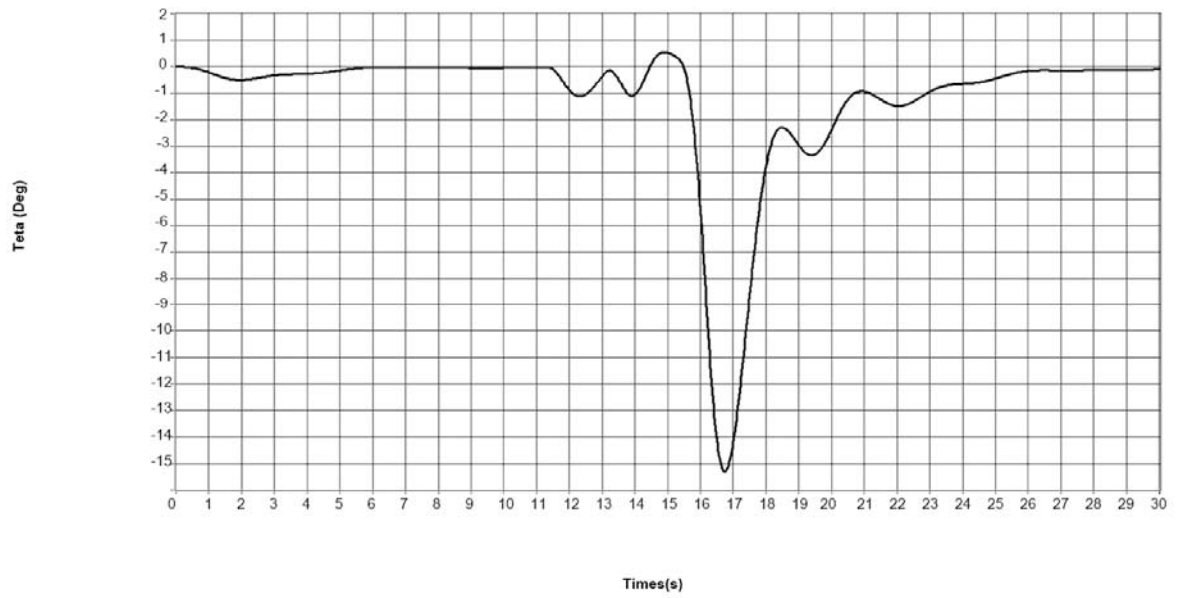


Fig. 6.83: Pitch angle variation.

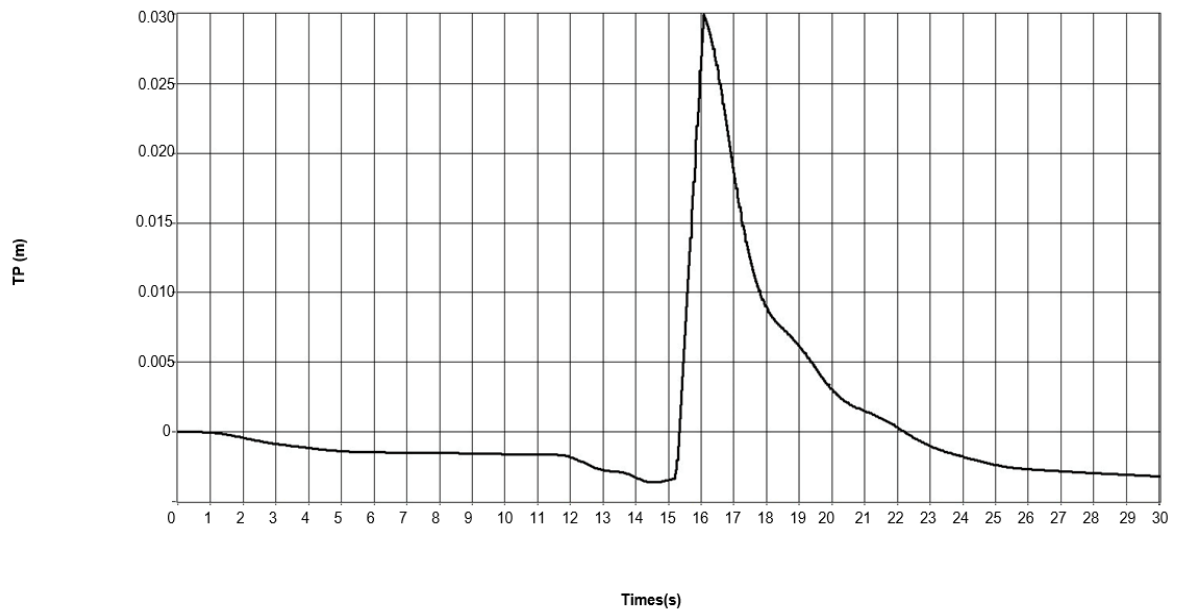


Fig. 6.84: Mass shifter variation.

At the end, the AUV successfully returned into its first operating conditions and all the parameters (depth, speed, direction angle ...) were stabilized back around their desired values.

6.7 Conclusion

In this chapter, different algorithms and related controllers were discussed and implemented to efficiently control the AUV response to predefined commands. They include speed, depth, roll, pitch, and yaw. The AUV capabilities in terms of obstacle detection and tracking were also investigated and a robust algorithm proposed.

After preliminary tests deemed necessary to adjust the internal parameters, the AUV was successfully tested under real operating conditions to investigate its behavior to different series of commands.

First, the controllers were evaluated separately in manual mode (without the cameras) to validate their individual implementation. In a second step, the AUV response was validated in terms of obstacles detection and avoidance.

CHAPTER 7

CONCLUSIONS AND FUTURE WORKS

7.1 Conclusions

The main purpose of this work was to design an autonomous underwater AUV prototype with vision and detection capabilities. The tasks involved the design of various mechanical parts, particularly two rotating thrusters and an interior mass shifter to control the device movement. Sensor boards were also embedded to share information with a mini-PC board and thus, to assure proper AUV operation. Control-based algorithms were developed to control the AUV operation and assure proper responses to external commands and/or surrounding conditions. Also, two cameras were used to collect information about the surrounding environment and an image processing technique implemented to treat the images sent by the cameras and then, to detect potential obstacles.

To do so, we started with the AUV body by selecting a shape and subdividing it into blocks in order to locate all embedded mechanical and electrical components required for proper

operation under real environmental conditions; the objective being to achieve a good trade-off between size and weight.

Then, by setting the maximum desired depth to 8 m, loading, strain, and stress analyses and simulations were performed to ensure the body safety while submitted to external forces.

Next, we included the mass shifter to control the AUV movement and assure its stability. We also designed a sensor board, a power board, an access board, and a motor driver board as well as a mini-PC to manage the data exchange with the different mechanical and electronic parts like the IMU, pressure, and compass sensors. Note that a particular effort has been made to significantly reduce the electronic connections and avoid energy waste.

After receiving predefined-user commands regarding the desired arrival point to reach at a certain depth and with a certain speed, the AUV was able to perform the task by efficiently changing its driving motors' speed and servomotors' angle. This played an important role in preserving the energy resources of the AUV, thus increasing its autonomy.

Also, a vision system for tracking obstacles was developed, embedded, and successfully tested. It uses a technique based on the Color-Texture Histogram and implemented within the mean shift method. Unlike conventional existing methods, Local Binary Patterns (LBP) was used to recognize the characteristics of the obstacle in order to achieve better understanding of its edges and corners. Note that this technique allowed reducing the number of iterations to search for targets resulting in a significant improvement in the AUV decision to cross the different obstacles with different widths and edges.

Finally, after more than 600 hours of testing under various operating conditions including both pool and ocean environments, the designed AUV successfully responded to all commands and efficiently detected fixed obstacles by taking appropriate decisions.

7.2 Future Works

Despite its good performance, the designed AUV remains just a prototype and then, opens the door to significant improvements in terms of both design and capabilities.

Future research directions could include the following:

- A more adequate material could be selected for the body and, therefore, more rigorous dynamic and control analyses can be performed to ensure the body will support and absorb any extra pressure due to external perturbations (rough or stormy sea).
- Increasing the AUV autonomy by designing flexible and waterproof solar panels to charge the batteries when deemed necessary.
- A similar work can be done on the batteries to improve their performance.
- The sound made by the thrusters is monotone and then, can be easily identified by the AUV. Starting from that, one can add a sound sensor to detect any external noise and therefore, track noisy obstacles. In fact, the current prototype can handle fixed obstacles, but it has not been designed to deal with approaching moving obstacles such as motorized floating objects.
- One of the thesis contributions is the use of mobile thrusters. However, during some tests, the existence of floating herbs and floating debris on water surface substantially affected the propellers' movement, causing the AUV breakdown and its return to the water surface in its rest position. It is therefore suggested to design a shield that will not harm the performance of the propellers and thus, the thruster system.
- Use more advanced sensors and design their related boards, having in mind the reduction of the AUV internal space and, on a long-term objective, to design a smaller and lighter prototype while keeping the existing features.
- Improve the vision system. This system plays an important role in decision-makings along with the movement in intelligent mode. During the tests, it has been observed that the visual system failed in some circumstances, like in highly polluted water. Enhanced algorithms may be implemented to improve the robustness of the current tracking system. Also, changing the existing camera with a Time of Flight (TOF) Basler device can be another direction to explore (with 3D images in one shot).

APPENDIX I

MODELING OF THE ROBOT

a) Kinematics

To derive a vector J vs. time in any coordinates system (Table AI-1), the following operator $D = d/dt$ is used:

$$D^A v = D^B v + \omega^{BA} \times J \quad (\text{AI-1})$$

Then, the transfer matrix form of a system B to a system W can be stated as

$${}^W_B R = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (\text{AI-2})$$

To calculate the angular speeds, the following equations are used:

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \quad (\text{AI-3})$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \quad (\text{AI-4})$$

$$\dot{\psi} = \frac{q \sin \phi + r \cos \phi}{\cos \theta} \quad (\text{AI-5})$$

Table AI-1: Variables for subsurface floating vessels with:

(1) Surge, (2) Sway, (3) Heave, (4) Roll, (5) Pitch, and (6) Yaw

Degrees of freedom	Movement Direction	Position and Euler Angle	Linear and Angular Speed	Force and Moment
1	Movement along x ⁽¹⁾	X_X'	u	X_X''
2	Movement along y ⁽²⁾	X_Y'	v	X_Y''
3	Movement along z ⁽³⁾	X_Z'	w	X_Z''
4	Rotation along x ⁽⁴⁾	ϕ	p	K_X'
5	Rotation along y ⁽⁵⁾	θ	q	K_Y'
6	Rotation along z ⁽⁶⁾	φ	r	K_Z'

Physical speeds p , q and r are calculated from the following equation [2]:

$$p = \dot{\phi} - \dot{\psi} \sin \theta \quad (\text{AI-6})$$

$$q = \dot{\theta} \cos \phi + \dot{\psi} \sin \phi \cos \theta \quad (\text{AI-7})$$

$$r = -\dot{\theta} \sin \phi + \dot{\psi} \cos \phi \cos \theta \quad (\text{AI-8})$$

To model the rotation in three dimensions, the concept in classical mechanics is to use rotational kinematics. One of the ways to describe rotation is the quaternion method. The four elements of the quaternion, q_0 , q_1 , q_2 and q_3 , are related to the physical speed components p , q and r through the following relation:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (\text{AI-9})$$

b) Dynamics

The linear momentum of a system B versus a system H can be expressed as:

$$p_B^H = m_B v_G^H \quad (\text{AI-10})$$

The system under discussion in this study contains two objects namely the floating vehicle, considered as m_1 and the weights, considered as m_2 that move relatively to m_1 . According to the Newton's second law, the resultant forces acting on m_B are equal with the linear momentum changes over time in the inertia system.

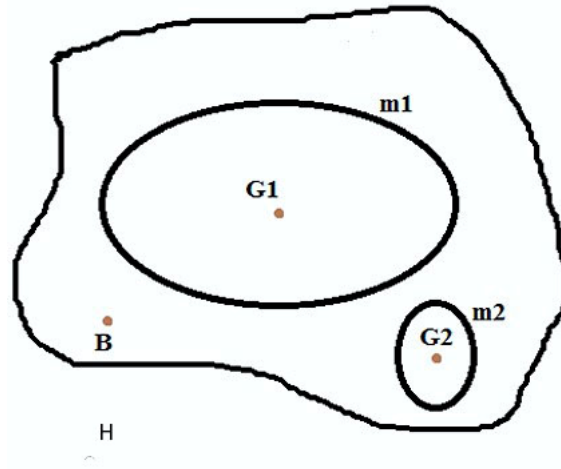


Fig. AI-1: System B and System H.

Assuming that both objects have constant mass, we get:

$$F_B = m_1 D^H v_{G_1}^H + m_2 D^H v_{G_2}^H \quad (\text{AI-11})$$

with $v_{G_1}^H$ the velocity of the center of G_1 and $v_{G_2}^H$ the velocity of the center of G_2 to the mass,

$$\begin{aligned} D^H v_{G_1}^H &= D^H D^H l_{G_1 H} = D^H (D^B l_{G_1 B} + \omega^{BH} \times l_{G_1 B}) + D^H D^H l_{BH} \\ &= D^B D^B l_{G_1 B} + (D^B \omega^{BH}) \times l_{G_1 B} + 2\omega^{BI} \times (D^B l_{G_1 B}) + \omega^{BH} \times (\omega^{BH} \times l_{G_1 B}) + c_B^H \end{aligned}$$

(AI-12)

$$D^H v_{G_2}^H = D^H D^H l_{G_2 H} = D^B D^B l_{G_2 B} + (D^B \omega^{BH}) \times l_{G_2 B} + 2\omega^{BH} \times (D^B l_{G_2 B}) + \omega^{BH} \times (\omega^{BH} \times l_{G_2 B}) + c_B^H \quad (\text{AI-13})$$

Since m_2 is moving compared to the reference system B connected to the mass m_1 , some of the terms removed in equation (AI-12) will not be removed in this equation, thus leading to:

$$f_B = m_1(D^B \omega^{BE}) \times l_{G_1 B} + m_1 \omega^{BH} \times (\omega^{BE} \times l_{G_1 B}) + m_1 c_B^H + m_2 c_{G_2}^B + m_2(D^B \omega^{BH}) \times l_{G_2 B} + 2m_2 \omega^{BH} \times (v_{G_2}^B) + m_2 \omega^{BH} \times (\omega^{BH} \times l_{G_2 B}) + m_2 c_B^H \quad (\text{AI-14})$$

Using the relation $m_B = m_1 + m_2$, and reorganizing the equations, gives

$$\begin{aligned} F_B &= (D^B \omega^{BE}) \times (m_1 l_{G_1 B} + m_2 l_{G_2 B}) + \omega^{BE} \times (\omega^{BH} \times (m_1 l_{G_1 B} + m_2 l_{G_2 B})) \\ &+ m_2 c_{G_2}^B + 2m_2 \omega^{BH} \times v_{G_2}^B + m_B c_B^H \\ \rightarrow c_B^H &= D^E (v_B^E + \omega^{EH} \times l_{BE}) + \omega^{EH} \times (v_B^E + \omega^{EH} \times l_{BE}) \\ &= D^E v_B^E + (D^E \omega^{EH}) \times l_{BE} + 2\omega^{EH} \times v_B^E + \omega^{EH} \times (\omega^{EH} \times l_{BE}) \\ \rightarrow c_B^H &= D^B v_B^E + \omega^{BE} \times v_B^E + 2\omega^{EH} \times v_B^E + \omega^{EH} \times (\omega^{EH} \times l_{BE}) \quad (\text{AI-15}) \end{aligned}$$

After substitution,

$$\begin{aligned} F_B &= (D^B \omega^{BE}) \times (m_1 l_{G_1 B} + m_2 l_{G_2 B}) + \omega^{BH} \times (\omega^{BH} \times (m_1 l_{G_1 B} + m_2 l_{G_2 B})) + \\ &m_2 c_{G_2}^B + 2m_2 \omega^{BH} \times v_{G_2}^B + m_B D^B v_B^E + m_B \omega^{BE} \times v_B^E + 2m_B \omega^{EH} \times v_B^E + \\ &m_B \omega^{EH} \times (\omega^{EH} \times l_{BE}) \quad (\text{AI-16}) \end{aligned}$$

According to the definition of the center of mass:

$$l_{GB} = \frac{m_1 l_{G_1 B} + m_2 l_{G_2 B}}{m_B} \rightarrow m_B l_{GB} = m_1 l_{G_1 B} + m_2 l_{G_2 B} \quad (\text{AI-17})$$

with $\omega^{BH} = \omega^{BE} + \omega^{EH}$

Then,

$$\begin{aligned}
 F_B &= (D^B \omega^{BE}) \times m_B l_{GB} + (D^B \omega^{EH}) \times m_B l_{GB} + \omega^{BE} \times (\omega^{BE} \times m_B l_{GB}) + \omega^{BE} \\
 &\quad \times (\omega^{EH} \times m_B l_{GB}) + \omega^{EH} \times (\omega^{BE} \times m_B l_{GB}) + \omega^{EH} \\
 &\quad \times (\omega^{EH} \times m_B l_{GB}) + m_2 c_{G_2}^B + 2m_2 \omega^{BE} \times v_{G_2}^B + 2m_2 \omega^{EH} \times v_{G_2}^B \\
 &\quad + m_B D^B v_B^E + m_B \omega^{BE} \times v_B^E + 2m_B \omega^{EH} \times v_B^E + m_B \omega^{EH} \\
 &\quad \times (\omega^{EH} \times l_{BE}) \\
 \rightarrow F_B &= m_B D^B v_B^E + m_B (D^B \omega^{BE}) \times l_{GB} + m_B \omega^{BE} \times v_B^E + 2m_B \omega^{EH} \times v_B^E + \\
 &\quad m_B \omega^{BE} \times (\omega^{BE} \times l_{GB}) + m_B \omega^{BE} \times (\omega^{EH} \times l_{GB}) + m_B \omega^{EH} \times \\
 &\quad (\omega^{BE} \times l_{GB}) + 2m_2 \omega^{BE} \times v_{G_2}^B + m_B (D^B \omega^{EH}) \times l_{GB} + m_B \omega^{EH} \times \\
 &\quad (\omega^{EH} \times l_{GB}) + m_2 c_{G_2}^B + 2m_2 \omega^{EH} \times v_{G_2}^B + m_B \omega^{EH} \times (\omega^{EH} \times l_{BE})
 \end{aligned} \tag{AI-18}$$

Knowing that

$$D^B \omega^{EH} = \omega^{EH} \times \omega^{BE} \tag{AI-19}$$

we get:

$$\begin{aligned}
 F_B &= m_B D^B v_B^E + m_B (D^B \omega^{BE}) \times l_{GB} + 2m_B \omega^{EH} \times v_B^E + m_B \omega^{BE} \times (\omega^{BE} \times l_{GB}) + \\
 &\quad m_B \omega^{BE} \times (\omega^{EH} \times l_{GB}) + m_B \omega^{EH} \times (\omega^{BE} \times l_{GB}) + 2m_2 \omega^{BE} \times v_{G_2}^B + \\
 &\quad m_B (\omega^{EH} \times \omega^{BE}) \times l_{GB} + m_B \omega^{EH} \times (\omega^{EH} \times l_{GB}) + 2m_2 \omega^{EH} \times v_{G_2}^B + m_B \omega^{EH} \times \\
 &\quad (\omega^{EH} \times l_{BE})
 \end{aligned} \tag{AI-20}$$

which leads to

$$\begin{aligned}
 F_B &= m_B D^B v_B^E + m_B (D^B \omega^{BE}) \times l_{GB} + m_B \omega^{BE} \times v_B^E + 2m_B \omega^{EH} \times v_B^E + \\
 &\quad m_B \omega^{BE} \times (\omega^{BE} \times l_{GB}) - 2m_B \omega^{EH} \times (\omega^{BE} \times l_{GB}) + 2m_2 \omega^{EH} \times v_{G_2}^B + \\
 &\quad m_B \omega^{EH} \times (\omega^{EH} \times l_{GB}) - m_2 c_{G_2}^B + 2m_2 \omega^{EH} \times v_{G_2}^B + m_B \omega^{EH} \times (\omega^{EH} \times l_{BE})
 \end{aligned}$$

(AI-21)

c) Rotational dynamics equations

Angular momentum: The total angular momentum of the system B can be defined as follows:

$$L_P^{BH} = I_G^B \omega^{GH} + l_{GP} \times m v_G^H \quad (\text{AI-22})$$

On the other hand, the relative angular momentum of the system B is defined according to:

$$L_{P(\text{rel})}^{BH} = I_P^B \omega^{PH} + l_{GP} \times m v_G^P \quad (\text{AI-23})$$

According to [222], the angular momentum of the system B on its center of mass is given by,

$$L_G^{BH} = I_G^B \omega^{GH} \quad (\text{AI-24})$$

and around the point P, we have

$$L_P^{BH} = I_G^B \omega^{GH} + l_{GP} \times m v_G^H \quad (\text{AI-25})$$

leading to the following equality:

$$\begin{aligned} L_P^{BI} &= I_P^B \omega^{PH} + l_{GP} \times m v_P^H \\ \rightarrow L_B^{BI} &= (I^B) \omega^{BH} + l_{G_1B} \times m_1 v_B^H + l_{G_2B} \times m_2 v_{G_2}^H \end{aligned} \quad (\text{AI-20})$$

Euler law: The Euler law says that the resultant momentum exerted on system B is equal to the time derivative of the total angular momentum in the inertia device as:

$$O_B^B = D^H L_B^{BH} \quad (\text{AI-26})$$

$$\rightarrow O_B^B = D^H \left((I^B) \omega^{BH} + l_{G_1B} \times m_1 v_B^H + l_{G_2B} \times m_2 v_{G_2}^H \right) \quad (\text{AI-27})$$

which leads to

$$O_B^B = D^H (I^B \omega^{BH}) + (D^H l_{G_1B}) \times m_1 v_B^H + l_{G_1B} \times m_1 (D^H v_B^H) + (D^H l_{G_2B}) \times m_2 v_{G_2}^H + l_{G_2B} \times m_2 (D^H v_{G_2}^H) \quad (\text{AI-28})$$

Each term can be calculated separately,

$$\begin{aligned} D^H (I^B \omega^{BH}) &= \\ H^B (D^B \omega^{BE}) &+ H^B (\omega^{EI} \times \omega^{BE}) + \omega^{BE} \times (H^B \omega^{BE}) + \omega^{EH} \times (H^B \omega^{BE}) + \omega^{BE} \times \\ & (H^B \omega^{EI}) + \omega^{EH} \times (H^B \omega^{EH}) \end{aligned} \quad (\text{AI-28.1})$$

$$\begin{aligned} (D^H l_{G_1B}) \times m_1 v_B^H &= \\ &= (\omega^{BH} \times l_{G_1B}) \times m_1 (D^E l_{BE} + \omega^{EH} \times l_{BE}) \\ &= ((\omega^{BE} + \omega^{EH}) \times l_{G_1B}) \times m_1 (v_B^E + \omega^{EH} \times l_{BE}) \\ &= m_1 (\omega^{BE} \times l_{G_1B}) \times v_B^E + m_1 (\omega^{EH} \times l_{G_1B}) \times v_B^E + m_1 (\omega^{BE} \times l_{G_1B}) \times \\ & (\omega^{EH} \times l_{BE}) + m_1 (\omega^{EH} \times l_{G_1B}) \times (\omega^{EH} \times l_{BE}) \end{aligned} \quad (\text{AI-28.2})$$

$$\begin{aligned} l_{G_1B} \times m_1 (D^H v_B^H) &= \\ &= m_1 l_{G_1B} \times D^B v_B^E + m_1 l_{G_1B} \times (\omega^{BE} \times v_B^E) + 2m_1 l_{G_1B} \times (\omega^{EH} \times v_B^E) + m_1 l_{G_1B} \times \\ & (\omega^{EH} \times (\omega^{EH} \times l_{BE})) \end{aligned} \quad (\text{AI-28.3})$$

$$\begin{aligned} (D^H l_{G_2B}) \times m_2 v_{G_2}^H &= \\ m_2 (D^B l_{G_2B} + \omega^{BH} \times l_{G_2B}) \times (D^H l_{BE}) &= \\ &= m_2 v_{G_2}^B \times v_B^E + m_2 (\omega^{BE} \times l_{G_2B}) \times v_B^E + m_2 (\omega^{EH} \times l_{G_2B}) \times v_B^E + m_2 v_{G_2}^B \times \\ & (\omega^{EH} \times l_{BE}) + m_2 (\omega^{BE} \times l_{G_2B}) \times (\omega^{EH} \times l_{BE}) + m_2 (\omega^{EH} \times l_{G_2B}) \times (\omega^{EI} \times l_{BE}) \end{aligned} \quad (\text{AI-28.4})$$

$$D^H v_{G_2}^H = c_{G_2}^B + (D^B \omega^{BE} + \omega^{EH} \times \omega^{BE}) \times l_{G_2B} + 2(\omega^{BE} + \omega^{EH}) \times v_{G_2}^B +$$

$$\begin{aligned}
& (\omega^{BE} + \omega^{EH}) \times \left((\omega^{BE} + \omega^{EH}) \times l_{G_2B} \right) + D^B v_B^E + \omega^{BE} \times v_B^E + 2\omega^{EH} \times v_B^E + \\
& \omega^{EH} \times (\omega^{EH} \times l_{BE}) \\
& = c_{G_2}^B + D^B \omega^{BE} \times l_{G_2B} + (\omega^{EH} \times \omega^{BE}) \times l_{G_2B} + 2\omega^{BE} \times v_{G_2}^B + 2\omega^{EH} \times v_{G_2}^B \\
& \quad + \omega^{BE} \times (\omega^{BE} \times l_{G_2B}) + \omega^{BE} \times (\omega^{EH} \times l_{G_2B}) + \omega^{EH} \\
& \quad \times (\omega^{BE} \times l_{G_2B}) + \omega^{EH} \times (\omega^{EH} \times l_{G_2B}) + D^B v_B^E + \omega^{BE} \times v_B^E \\
& \quad + 2\omega^{EH} \times v_B^E + \omega^{EH} \times (\omega^{EH} \times l_{BE})
\end{aligned}$$

$$\begin{aligned}
& l_{G_2B} \times m_2(D^H v_{G_2}^H) = \\
& m_2 l_{G_2B} \times a_{G_2}^B + m_2 l_{G_2B} \times (D^B \omega^{BE} \times l_{G_2B}) + m_2 l_{G_2B} \times \left((\omega^{EH} \times \omega^{BE}) \times l_{G_2B} \right) + \\
& 2m_2 l_{G_2B} \times (\omega^{BE} \times v_{G_2}^B) \\
& + 2m_2 l_{G_2B} \times (\omega^{EH} \times v_{G_2}^B) + m_2 l_{G_2B} \times \left(\omega^{BE} \times (\omega^{BE} \times l_{G_2B}) \right) + m_2 l_{G_2B} \times \\
& \left(\omega^{BE} \times (\omega^{EH} \times l_{G_2B}) \right) \\
& + m_2 l_{G_2B} \times \left(\omega^{EH} \times (\omega^{BE} \times l_{G_2B}) \right) + m_2 l_{G_2B} \times \left(\omega^{EH} \times (\omega^{EH} \times l_{G_2B}) \right) + \\
& m_2 l_{G_2B} \times D^B v_B^E + m_2 l_{G_2B} \times (\omega^{BE} \times v_B^E) + 2m_2 l_{G_2B} \times (\omega^{EH} \times v_B^E) + m_2 l_{G_2B} \times \\
& (\omega^{EH} \times (\omega^{EH} \times l_{BE})) \tag{AI-28.5}
\end{aligned}$$

$$\begin{aligned}
& D^H(m_B l_{GB} \times v_B^H) = \\
& = m_B(v_G^B + \omega^{EH} \times l_{GB}) \times (v_B^E + \omega^{BH} \times l_{BE}) + m_B l_{GB} \times c_B^H \\
& = m_B v_G^B \times v_B^E + m_B v_G^B \times (\omega^{EH} \times l_{BE}) + m_B(\omega^{BE} \times l_{GB}) \times v_B^E + m_B(\omega^{EH} \times l_{GB}) \\
& \quad \times (\omega^{EH} \times l_{BE}) + m_B(\omega^{EH} \times l_{GB}) \times v_B^E \\
& + m_B(\omega^{EH} \times l_{GB}) \times (\omega^{EH} \times l_{BE}) + m_B l_{GB} \times D^B v_B^E + m_B l_{GB} \times (\omega^{BE} \times v_B^E) + \\
& 2m_B l_{GB} \times (\omega^{EH} \times v_B^E) + m_B l_{GB} \times (\omega^{EH} \times (\omega^{EH} \times l_{BE})) \tag{AI-28.6}
\end{aligned}$$

By combining the above terms, we obtain:

$$\begin{aligned}
O_B^B & = I^B(D^B \omega^{BE}) + I^B(\omega^{EH} \times \omega^{BE}) + \omega^{BE} \times (I^B \omega^{BE}) + \omega^{EH} \times (I^B \omega^{BE}) + \\
& \omega^{BE} \times (I^B \omega^{EH}) + \omega^{EH} \times (I^B \omega^{EH}) + m_B v_G^B \times v_B^E + m_B v_G^B \times (\omega^{EH} \times l_{BE}) + \\
& m_B(\omega^{BE} \times l_{GB}) \times v_B^E + m_B(\omega^{BE} \times l_{GB}) \times (\omega^{EH} \times l_{BE}) + m_B \omega^{EH} \times (l_{GB} \times v_B^E) + \\
& m_B(\omega^{EH} \times l_{GB}) \times (\omega^{EH} \times l_{BE}) + m_B l_{GB} \times D^B v_B^E + m_B l_{GB} \times (\omega^{BE} \times v_B^E) +
\end{aligned}$$

$$\begin{aligned}
& m_B l_{GB} \times (\omega^{EH} \times v_B^E) + m_B l_{GB} \times (\omega^{EH} \times (\omega^{EH} \times l_{BE})) + m_2 l_{G_2B} \times c_{G_2}^B + \\
& m_2 l_{G_2B} \times ((D^B \omega^{BE}) \times l_{G_2B}) + 2m_2 l_{G_2B} \times (\omega^{BE} \times v_{G_2}^B) + 2m_2 l_{G_2B} \times \\
& (\omega^{EH} \times v_{G_2}^B) + m_2 l_{G_2B} \times (\omega^{BE} \times (\omega^{BE} \times l_{G_2B})) + 2m_2 l_{G_2B} \times (\omega^{EH} \times \\
& (\omega^{BE} \times l_{G_2B})) + m_2 l_{G_2B} \times (\omega^{EH} \times (\omega^{EH} \times l_{G_2B}))
\end{aligned} \tag{AI-29}$$

Thus,

$$m_B v_G^B = m_B D^B l_{GB} = D^B (m_1 l_{G_1B} + m_2 l_{G_2B}) = m_1 D^B l_{G_1B} + m_2 D^B l_{G_2B} = m_2 v_{G_2}^B \tag{AI-30}$$

Making the main equation as:

$$\begin{aligned}
O_B^B &= I^B (D^B \omega^{BE}) + I^B (\omega^{EH} \times \omega^{BE}) + \omega^{BE} \times (I^B \omega^{BE}) + \omega^{EH} \times (I^B \omega^{BE}) + \\
& \omega^{BE} \times (I^B \omega^{EH}) + \omega^{EH} \times (I^B \omega^{EH}) + m_2 v_{G_2}^B \times v_B^E + m_2 v_{G_2}^B \times (\omega^{EH} \times l_{BE}) + \\
& m_B (\omega^{BE} \times l_{GB}) \times v_B^E + m_B (\omega^{BE} \times l_{GB}) \times (\omega^{EH} \times l_{BE}) - m_B \omega^{EH} \times (l_{GB} \times v_B^E) + \\
& m_B (\omega^{BE} \times l_{GB}) \times (\omega^{BE} \times l_{BE}) + m_B l_{GB} \times D^B v_B^E + m_B l_{GB} \times (\omega^{BE} \times v_B^E) - \\
& m_B l_{GB} \times (\omega^{EH} \times v_B^E) + m_B l_{GB} \times (\omega^{EH} \times (\omega^{BE} \times l_{BE})) + m_2 l_{G_2B} \times c_{G_2}^B + \\
& m_2 l_{G_2B} \times ((D^B \omega^{BE}) \times l_{G_2B}) + 2m_2 l_{G_2B} \times (\omega^{EH} \times v_{G_2}^B) - 2m_2 l_{G_2B} \times \\
& (\omega^{EH} \times v_{G_2}^B) - m_2 l_{G_2B} \times (\omega^{BE} \times (\omega^{EH} \times l_{G_2B})) + 2m_2 l_{G_2B} \times (\omega^{EH} \times \\
& (\omega^{BE} \times l_{G_2B})) + m_2 l_{G_2B} \times (\omega^{EH} \times (\omega^{EH} \times l_{G_2B}))
\end{aligned} \tag{AI-31}$$

we get the following set of dynamic equations

$$\begin{aligned}
[F_B]^B &= m_B [\dot{v}_B^E]^B - m_B [l_{GB}]_\times^B [\dot{\omega}^{BE}]^B - m_B [v_B^E]_\times^B [\omega^{BE}]^B - \\
& m_B \{[\omega^{BE}]_\times^B [l_{GB}]^B\}_\times [\omega^{BE}]^B + 2m_B [\omega^{EH}]_\times^B [v_B^E]^B - 2m_B [\omega^{EH}]_\times^B [l_{GB}]_\times^B [\omega^{BE}]^B - \\
& 2m_2 [v_{G_2}^B]_\times^B [\omega^{BE}]^B + m_B [\omega^{EH}]^B \times ([\omega^{EH}]^B \times [l_{GB}]^B) + m_2 [c_{G_2}^B]^B - 2m_2 [\omega^{EH}]^B \times \\
& [v_{G_2}^B]^B + m_B [\omega^{EH}]^B \times ([\omega^{EH}]^B \times [l_{BE}]^B)
\end{aligned} \tag{AI-32}$$

$$\begin{aligned}
[O_B^B]^B = & \\
& m_B[l_{GB}]_{\times}^B[\dot{v}_B^E]^B + [I^B]^B[\dot{\omega}^{BE}]^B + m_B[l_{GB}]_{\times}^B[\omega^{BE}]_{\times}^B[v_B^E]^B \\
& - \{[I^B]^B[\omega^{BE}]^B\}_{\times}[\omega^{BE}]^B + \\
& [I^B]^B[\omega^{EH}]_{\times}^B[\omega^{BE}]^B + [\omega^{EH}]_{\times}^B[I^B]^B[\omega^{BE}]^B - \{[I^B]^B[\omega^{EH}]^B\}_{\times}[\omega^{BE}]^B \\
& - m_2[v_{G_2}^B]_{\times}^B[v_B^E]^B + m_B\{[\omega^{BE}]_{\times}^B[l_{GB}]^B\}_{\times}[v_B^E]^B \\
& + m_B\{[\omega^{EH}]_{\times}^B[l_{BE}]^B\}_{\times}[l_{GB}]_{\times}^B[\omega^{BE}]^B - m_B[\omega^{EH}]_{\times}^B[l_{GB}]_{\times}^B[v_B^E]^B \\
& + m_B[l_{GB}]_{\times}^B[\omega^{EH}]_{\times}^B[v_B^E]^B + m_2[l_{G_2B}]_{\times}^B[l_{G_2B}]_{\times}^B[\dot{\omega}^{BE}]^B \\
& - 2m_2[l_{G_2B}]_{\times}^B[v_{G_2}^B]_{\times}^B[\omega^{BE}]^B + m_2[l_{G_2B}]_{\times}^B[\omega^{BE}]_{\times}^B[l_{G_2B}]_{\times}^B[\omega^{EH}]^B \\
& - 2m_2[l_{G_2B}]_{\times}^B[\omega^{BH}]_{\times}^B[l_{G_2B}]_{\times}^B[\omega^{BE}]^B \\
& + [\omega^{EH}]^B \times ([I^B]^B[\omega^{EH}]^B) + m_2[v_{G_2}^B]^B \times ([\omega^{EH}]^B \times [l_{BE}]^B) - m_B([\omega^{EH}]^B \times \\
& [l_{GB}]^B) \times ([\omega^{EH}]^B \times [l_{BE}]^B) + m_B[l_{GB}]^B \times ([\omega^{EH}]^B \times ([\omega^{EH}]^B \times [l_{BE}]^B)) + \\
& m_2[l_{G_2B}]^B \times [c_{G_2}^B]^B + 2m_2[l_{G_2B}]^B \times ([\omega^{EH}]^B \times [v_{G_2}^B]^B) + m_2[l_{G_2B}]^B \times \\
& ([\omega^{EH}]^B \times ([\omega^{EH}]^B \times [l_{G_2B}]^B))
\end{aligned} \tag{AI-33}$$

APPENDIX II

ELECTRONIC PARTS: SUPPLEMENTARY INFORMATION

a) Main processor:



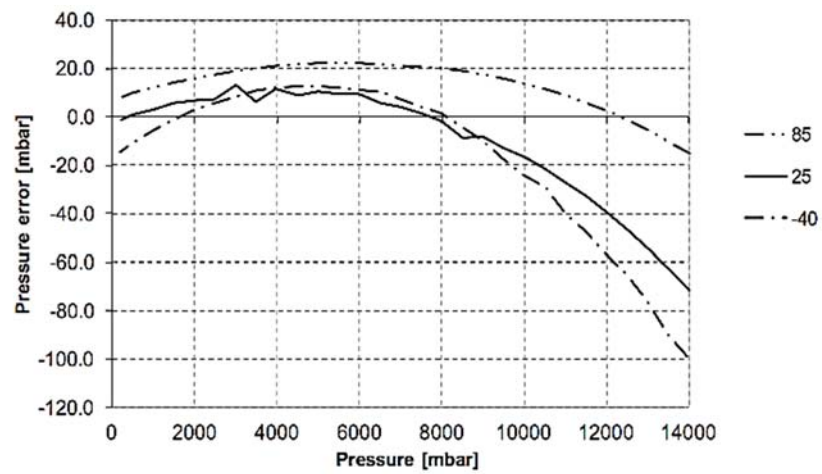
Fig. AII-1: The Giada i200 (5.1" x 5.1" x 1") that the Intel Core i7 Haswell Processors.

b) Camera housing:

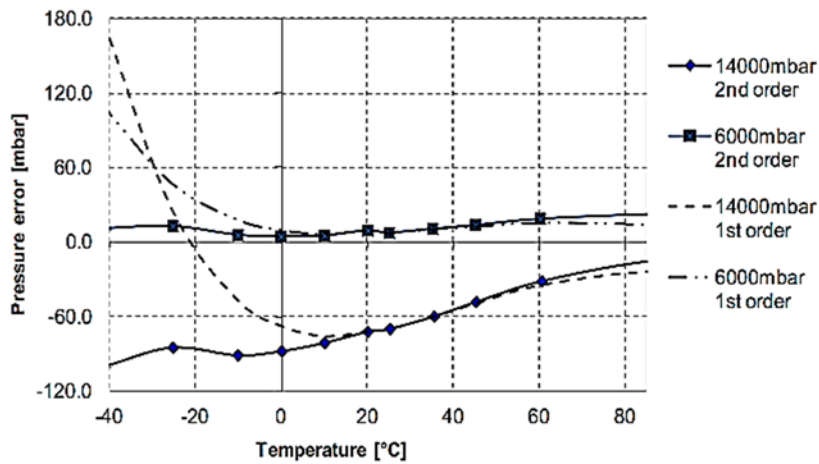


Fig. AII-2: Rotating Body and Board.

c) Pressure sensor datasheet (MS5803-14BA):



(a)



(b)

Fig. AII-3: (a) Absolute Pressure Accuracy and (b) Pressure Accuracy Vs Temperature [126].

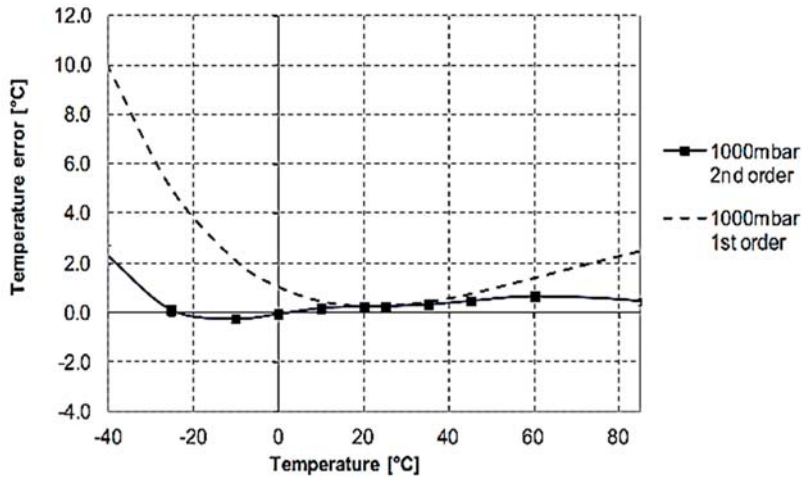
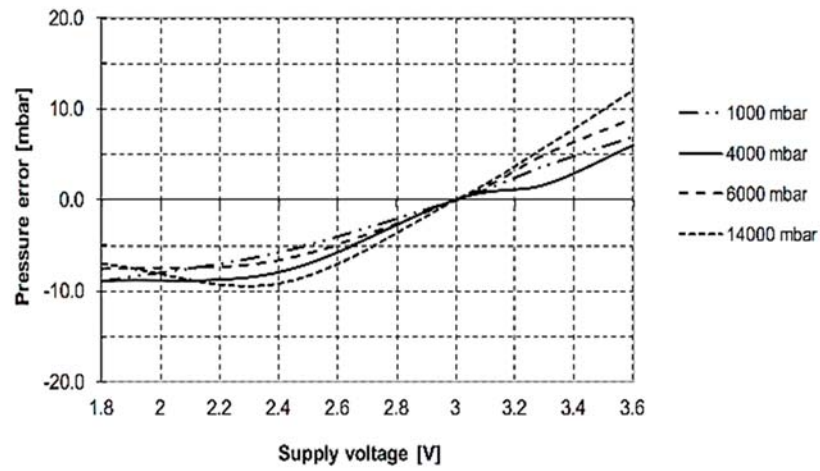
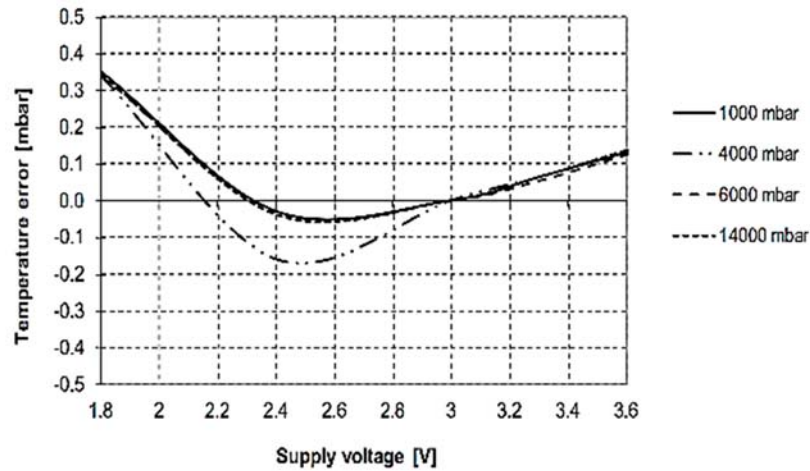


Fig. AII-4: Temperature Accuracy Vs Temperature [126].



(a)



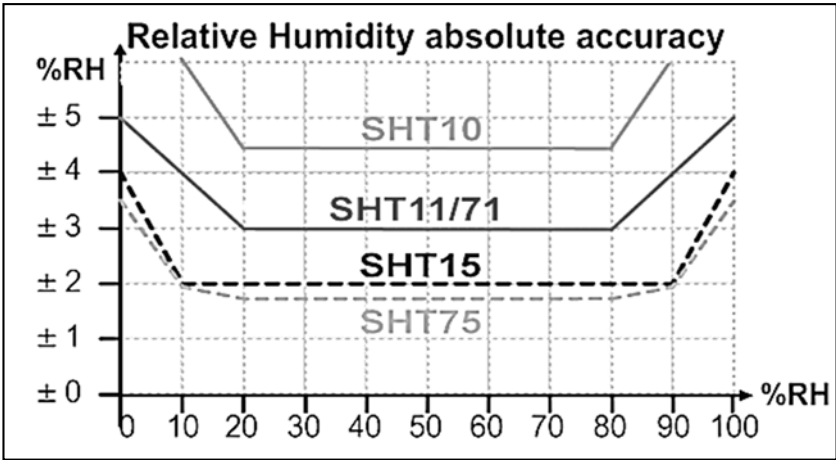
(b)

Fig. AII-5: (a): Pressure Error, (b): Temperature Error Vs Supply Voltage at 25°C [126].

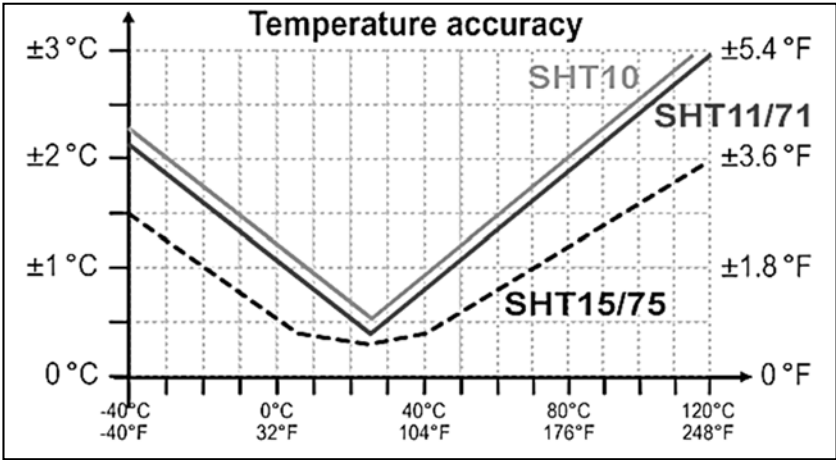
d) Humidity/temperature sensor datasheet (SHT-11):

Part Number	Humidity accuracy [%RH]	Temperature accuracy [K] @ 25 °C	Package
SHT10	±4.5	±0.5	SMD (LCC)
SHT11	±3.0	±0.4	SMD (LCC)
SHT15	±2.0	±0.3	SMD (LCC)
SHT71	±3.0	±0.4	4-pin single-in-line
SHT75	±1.8	±0.3	4-pin single-in-line

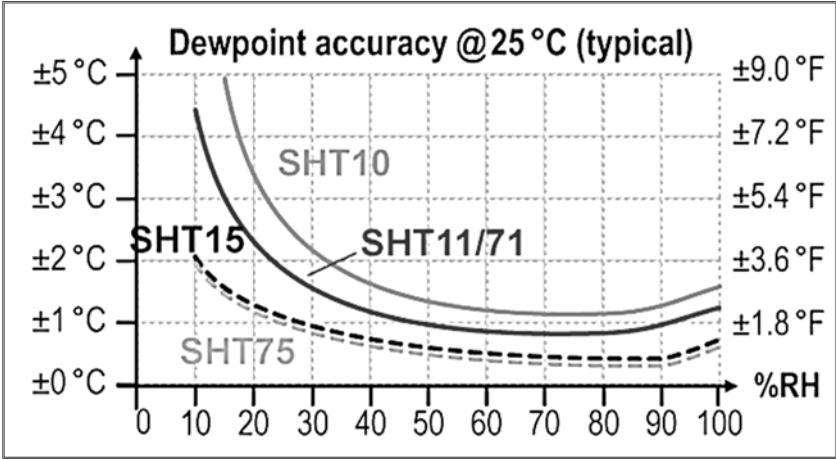
Fig. AII-6: SHT Family [128].



(a)



(b)



(c)

Fig. AII-7: (a): Relative Humidity, (b): Temperature and (c): Dewpoint Accuracies [128].

APPENDIX III

MATLAB IMPLEMENTATION OF OBJECT TRACKING

a) Describing the target

```

function
q_u=flbp81_rgb_PDF(Image,center,w_halfsize,lbpThreshold,redBins,greenBins,blueBins)

histo=zeros(redBins,greenBins,blueBins,5);

% initialize the target histogram
rmin=center(1)-w_halfsize(1);
rmax=center(1)+w_halfsize(1);
cmin=center(2)-w_halfsize(2);
cmax=center(2)+w_halfsize(2);

wmax=(rmin-center(1)).^2+(cmin-center(2)).^2+1;
for i=rmin:rmax %
    for j=cmin:cmax
        if textureMat(i,j)~=0
            d=(i-center(1)).^2+(j-center(2)).^2;
            w=wmax-d; % weight value corresponding to pixel(i,j)
            R=floor(Image(i,j,1)/(256/redBins))+1;
            G=floor(Image(i,j,2)/(256/greenBins))+1;
            B=floor(Image(i,j,3)/(256/blueBins))+1;
            T=textureMat(i,j);
            histo(R,G,B,T)=histo(R,G,B,T)+w; % color-texture joint
        histogram
    end
end
end
end

```

```

% convert histo (matrix) into 1D vector
for i=1:redBins
    for j=1:greenBins
        for k=1:blueBins
            for s=1:5
                index=(i-1)*greenBins*blueBins*5+(j-1)*blueBins*5+(k-
1)*5+s;
                q_u(index)=histo(i,j,k,s);
                sum_q=sum_q+q_u(index);
            end
        end
    end
end

q_u=q_u/sum_q;
return;

for i=1:redBins
    for j=1:greenBins
        for k=1:blueBins
            for s=1:5
                index=(i-1)*greenBins*blueBins*5+(j-1)*
                blueBins*5+(k-1)*5+s;
                p_u(index)=histo(i,j,k,s);
                sum_p=sum_p+p_u(index);
            end
        end
    end
end

p_u=p_u/sum_p; % normalize

```

b) Tracking with mean shift

```
MS=sqrt(sum((center_r-center_old).^2));    % normal of mean shift
vector
```

```
% compute the weight value of each pixel in the candidate window
for i=rmin:rmax
    for j=cmin:cmax
        if (i>=1 & i<=height & j>=1 & j<=width)
            if textureMat(i,j)~=0
                R=floor(Image(i,j,1)/(256/redBins))+1;
                G=floor(Image(i,j,2)/(256/greenBins))+1;
                B=floor(Image(i,j,3)/(256/blueBins))+1;
                T=textureMat(i,j);
```

```
center_r=(x*w_i'/sum(w_i))';    % new center
```

c) Target description by color-texture histogram

```

p1(3:sy+2,3:sx+2) = X ;
p2(3:sy+2,2:sx+1) = w2*double(X) ;
p3(3:sy+2,1:sx) = X ;

p4(2:sy+1,3:sx+2) = w2*double(X) ;
p5(2:sy+1,2:sx+1) = (1-1/sqrt(2))^2*double(X) ;
p6(2:sy+1,1:sx) = w2*double(X) ;

p7(1:sy,3:sx+2) = X ;
p8(1:sy,2:sx+1) = w2*double(X) ;
p9(1:sy,1:sx) = X ;

Xi1 = w1*p1+ p2+p4 + p5 + 0.000001;
Xi2(3:sy+2,2:sx+1) = X;
Xi3 = w1*p3 + p2 + p6 + p5 + 0.000001;
Xi4(2:sy+1,1:sx) = X;
Xi5 = w1*p9 + p8 + p6 + p5 + 0.000001;
Xi6(1:sy,2:sx+1) = X;
Xi7 = w1*p7 + p8 + p4 + p5 + 0.000001;
Xi8(2:sy+1,3:sx+2) = X;

Xi=
(Xi4>=Xi+T)+2*(Xi5>=Xi+T)+4*(Xi6>=Xi+T)+8*(Xi7>=Xi+T)+16*(Xi8>=Xi+T)+3
2*(Xi1>=Xi+T)+64*(Xi2>=Xi+T)+128*(Xi3>=Xi+T);

X=Xi(3:sy,3:sx);

M=size(X,1);
N=size(X,2);

```

```
for i=1:M
    for j=1:N
        textureMat(i,j)=patternMap(X(i,j)+1);
    end
end
```

d) Image derivative

```
im1 = medfilt2(im1,[3 3]);
```

```
Gmag = imgradient(im1,'prewitt');
im2 = Gmag > 25;
```

```
prop=regionprops(im2);
P_A = {prop.Area};
P_A = cell2mat(P_A);
ind = find(P_A == max(P_A));
rectangle('Position',prop(ind).BoundingBox,'EdgeColor','red')
```


REFERENCES

- [1] D.R. Blidberg, "The development of autonomous underwater vehicles (AUV); a brief summary," *Int. Conf. on Robotics and Automation*, Seoul, Korea, 2001.
- [2] R. Burgoyne, X. Huang, C. North, N. Schaal, and K. Shaw, "Final year project report; unmanned underwater vehicle," *University of Canterbury*, New-Zealand, 2007.
- [3] G.N. Roberts and R. Sutton, *Advances in Unmanned Marine Vehicles*, IEE Control Series, vol. 69, 2006.
- [4] H.-T. Choi and P.-M. Lee, "Development of a system architecture for an advanced autonomous underwater vehicle, ORCA," *Int. Conf. on Communications, Circuits and Systems*, Bangkok, Thailand, 2004.
- [5] H.-T. Choi *et al.*, "New project and new underwater robot for new missions," *Int. Conf. on Advanced Mechatronics*, Osaka, Japan, 2010.
- [6] K.-M. Han and H.-T. Choi, "Shape context based object recognition and tracking in underwater environment," *IEEE Int. Geoscience and Remote Sensing Symp.*, Vancouver, Canada, 2011.
- [7] K. Han and H. Choi, "Recognition and tracking of texture-free objects for submerged robots in structured underwater environment," *Int. Unmanned Untethered Submersible Technology Conf.*, Portsmouth, NH, 2011.
- [8] K. Han and H. Choi, "Vision based 3D pose estimation for underwater vehicle docking problem," *Int. Unmanned Untethered Submersible Technology Conf.*, Portsmouth, NH, 2011.
- [9] C. Silpa-Anan, "Autonomous underwater vehicle: vision and control," Master's thesis, *The Australian National University*, Australia, 2001.
- [10] D. Wettergreen *et al.*, "Development of autonomous underwater vehicle towards visual servo control," *Australian Conf. on Robotics and Automation*, Melbourne, Australia, 2000.
- [11] W. Naeem, R. Sutton, and S. M. Ahmad, "LQG/LTR control of an autonomous underwater vehicle using a hybrid guidance law," *Int. Federation on Automatic Control*, Workshop on Guidance and Control of Underwater Vehicles, Newport, UK, 2003.
- [12] T.I. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*, Marine Cybernetics, 2002.
- [13] R. Bachmayer *et al.*, "Oceanographic research using remotely operated underwater robotic vehicles: Exploration of hydrothermal vent sites on the mid-Atlantic ridge at 37° North 32° West," *Marine Technology Society J.*, vol. 32, pp. 37-47, 1996.

- [14] S.B. Williams *et al.*, “A decoupled, distributed AUV control architecture,” Technical Report, *University of Sydney*, 2006.
- [15] K. Hur, “Brushless motor controller for an AUV,” *University of Texas*, 2006.
- [16] Z. Vukić and N. Mišković, “State and perspectives of underwater robotics - role of laboratory for underwater systems and technologies,” *Pomorski Zbornik Posebno Izdanje*, pp. 15-27, 2016.
- [17] A. Pascoal *et al.*, “Robotic ocean vehicles for marine science applications: the European ASIMOV project,” *Oceans 2000 MTS/IEEE Conf.*, Narragansett, RI, 2000.
- [18] R. Garcia *et al.*, “3-D mapping for the characterization of the seafloor using an underwater robot within the MUMAP project,” *Int. Symp. on Naval Architecture and Maritime*, Istanbul, Turkey, 2014.
- [19] D.A. Anisi, E. Persson, and C. Heyer, “Real-world demonstration of sensor-based robotic automation in oil and gas facilities,” *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Francisco, CA, 2011.
- [20] L. Lapierre, “Underwater robots Part I: Current systems and problem pose,” in *Mobile Robotics - Towards New Applications*, A. Lazinica Ed., InTechOpen, pp. 335-360, 2006.
- [21] M. Kemp, M. Palanza, C. Skibski, J. Ormsby, and M. Estaphan, “Persistence at full ocean depth,” *IEEE/OES Autonomous Underwater Vehicles Conf.*, Southampton, UK, 2012.
- [22] H.T. Choi, A. Hanai, S.K. Choi, and J. Yuh, “Development of an underwater robot, Odin-III,” *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, NV, 2003.
- [23] T. Clarke, “Robots in the deep,” *Nature*, vol. 30, pp. 468–470, 2003.
- [24] B.H. Robison, “The coevolution of undersea vehicles and deep-sea research,” *Marine Technology Society J.*, vol. 33, pp. 69-73, 2000.
- [25] D. Walther, D.R. Edgington, and C. Koch, “Detection and tracking of objects in underwater video,” *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Washington, DC, 2004.
- [26] <http://www.homebuiltrovs.com>, [Last access May 20, 2016].
- [27] <http://www.instructables.com>, [Last access May 20, 2016].
- [28] <http://openrov.dozuki.com>, [Last access May 20, 2016].
- [29] <http://www.imaging-resource.com>, [Last access May 20, 2016].
- [30] K.M. Kelly, “A deep diving remotely operated vehicle for research in the Pacific,” *University of Hawaii*, 1991.
- [31] http://www.rov.net/pages/hydro_demon_spec1.htm, [Last access May 20, 2016].
- [32] http://www.rov.net/pages/Perry_Triton_Spec1.htm, [Last access May 20, 2016].
- [33] <http://www.rov.net/pages/mrv1.htm>, [Last access May 20, 2016].
- [34] http://www.rov.net/pages/Perry_TritonXL_spec1.htm, [Last access May 20, 2016].
- [35] http://www.rov.net/pages/Amatek_Scorpio_spec1.htm, [Last access May 20, 2016].
- [36] http://www.rov.net/pages/hydro_hyball_spec1.htm, [Last access May 20, 2016].
- [37] T.R. Gooding, “A framework for evaluating advanced search concepts for multiple autonomous underwater vehicle (AUV) mine countermeasures (MCM),” *Massachusetts Institute of Technology*, MA, 2001.

- [38] College Physics. OpenStax College. 2012: <http://cnx.org/content/col11406/latest/C>, [Last access May 20, 2016].
- [39] E. C. Tupper, *Introduction to Naval Architecture: Formerly Muckle's Naval Architecture for Marine Engineers*, Butterworth-Heinemann, 5th Ed., 2013.
- [40] <https://sites.google.com/site/aerodynamics4students/>, [Last access May 20, 2016].
- [41] AUVSI Foundation and ONR Engineering Primer Document, *RoboSub Competitions*, 2014.
- [42] R. Burcher and L. Rydill, *Concepts in Submarine Design*, Cambridge University Press, 1999.
- [43] Aerodynamics for Students, www.aerodynamics4students.com, [Last access May 20, 2016].
- [44] Y.A. Cengel and J.M. Cimbala, *Fluid Mechanics: Fundamentals and Applications*, McGraw-Hill Education, 3rd Ed. 2013.
- [45] D. Olbers, J. Willebrand, and C. Eden, *Ocean Dynamics*, Springer-Verlag, 2012.
- [46] P.N. Joubert, "Some aspects of submarine design Part 1 – Hydrodynamics," DSTO Report DSTO-TR-1622, *Defence Science and Technology Org.*, Melbourne, Australia, 2004.
- [47] M. Sadraey, *Aircraft Performance Analysis*, VDM Verlag, 2009.
- [48] A.H. Techet, "Hydrodynamics for Ocean Engineers," Course Notes, *Massachusetts Institute of Technology*, 2014.
- [49] C. Barry, "Propeller Selection for Boats and Small Ships," *E Marine Training - Prop Matching*, 2005
- [50] https://en.wikipedia.org/wiki/Autonomous_underwater_vehicle, [Last access May 20, 2016].
- [51] T.L. Lowe and J. Rounce, *Calculations for A-level Physics*, Nelson Thornes, 2002.
- [52] http://web.mit.edu/snively/www/2_006%20Coursenotes.pdf, [Last access May 20, 2016].
- [53] <http://batteryuniversity.com/learn/>, [Last access May 20, 2016].
- [54] P. Fortescue, G. Swinerd, and J. Stark, *Spacecraft Systems Engineering*, Wiley, 4th Ed., 2011.
- [55] M. Drtil, "Electronics and sensor design for an autonomous underwater vehicle," Diplom-Ingenieur Thesis, *University of Western Australia*, 2006.
- [56] E. Abbott and D. Powell, "Land-vehicle navigation using GPS," *Proc. IEEE*, vol. 87, pp. 145-162, Jan. 1999.
- [57] K.P. Valavanis, D. Gracanin, M. Matijasevic, and R. Kolluru, "Control architectures for autonomous underwater vehicles," *IEEE Control Systems Magazine*, vol. 17, pp. 48-64, 1997.
- [58] E. Galceran, "Coverage path planning for autonomous underwater vehicles," Ph.D. Thesis, *University of Girona*, 2014.
- [59] X. Yun, E.R. Bachmann, and S. Arslan, "An inertial navigation system for small autonomous underwater vehicles," *IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, 2000.
- [60] J.C. Kinsey, R.M. Eustice, and L.L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," *Int. Conf. on Manoeuvring and Control of Marine Craft*, Lisbon, Portugal, 2006.

- [61] D. Loebis, J. Chudley and R. Sutton, "A fuzzy Kalman filter optimized using a genetic algorithm for accurate navigation of an autonomous underwater vehicle," *Manoeuvring and Control of Marine Craft Int. Conf.*, Girona, Spain, 2003.
- [62] B. Barshan and H.F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Trans. Robotics and Automation*, vol. 11, pp. 328-342, 1995.
- [63] F.C. Teixeira and A.M. Pascoal, "Geophysical navigation of autonomous underwater vehicles using geomagnetic information," *Int. Workshop on Navigation, Guidance and Control of Underwater Vehicles*, Killaloe, Ireland, 2008.
- [64] I. Khan *et al.* "Sliding mode control of lateral dynamics of an AUV," *Int. Bhurban Conf. on Applied Sciences and Technology*, Islamabad, Pakistan, 2012.
- [65] E. Sebastian and M.A. Sotelo, "Adaptive fuzzy sliding mode controller for the kinematic variables of an underwater vehicle," *J. of Intelligent and Robotic Systems*, vol. 49, pp 189–215, 2007.
- [66] K. Kemal Ari, F.T. Asal, and M. Coşgun, "PI, PD, and PID controllers," Project Report, *Middle East Technical University*, Ankara, Turkey, 2012.
- [67] K.J. Åström, "Control system design," Lecture Notes, *University of California*, CA, 2002.
- [68] S. Singh and A.K. Pandey, "Design of PI Controller to minimize the speed error of D.C. servo motor," *Int. J. of Scientific and Technology Research*, vol. 1, pp. 95-98, 2012.
- [69] Y. Chen, "Modular modeling and control for autonomous underwater vehicle (AUV)," Master of engineering, *National University of Singapore*, Singapore, 2008.
- [70] E. Dong *et al.* "A neural network-based self-tuning PID controller of an autonomous underwater vehicle." *IEEE Int. Conf. on Mechatronics and Automation*, Chengdu, China, 2012.
- [71] Q. Chen, T. Chen, and Y. Zhang. "Research of GA-based PID for AUV motion control." *IEEE Int. Conf. on Mechatronics and Automation*, Changchun, China, 2009.
- [72] M.H. Khodayari and S. Balochian. "Modeling and control of autonomous underwater vehicle (AUV) in heading and depth attitude via self-adaptive fuzzy PID controller." *J. of Marine Science and Technology*, vol. 20, pp. 559-578, 2015.
- [73] J. González, S. Gomáriz, C. Battle, and C. Galarza, "Fuzzy controller for the yaw and velocity control of the Guanay II AUV." *Int. Federation on Automatic Control*, Girona, Spain, 2015.
- [74] B. Raeisy, A.A. Safavi, and A.R. Khayatian. "Optimized fuzzy control design of an autonomous underwater vehicle." *Iranian J. of Fuzzy Systems*, vol. 9, pp. 25-41, 2012.
- [75] https://en.wikipedia.org/wiki/Bell_Boeing_V-22_Osprey, [Last access May 20, 2016].
- [76] <https://www.wikipedia.org>, [Last access May 20, 2016].
- [77] <https://en.wikipedia.org/wiki/Polytetrafluoroethylene>, [Last access May 20, 2016].
- [78] M. Wikol *et al.*, *Expanded Polytetrafluoroethylene Membranes and Their Applications*, CRC Press, 2007.
- [79] Teflon® PTFE Fluoropolymer Resin, Properties Handbook, 2013.
- [80] B.R. Munson, D.F. Young, and T.H. Okiishi, *Fundamentals of Fluid Mechanics*, 5th Ed., Wiley, 2005.

- [81] M.R. Lindeburg, *Mechanical Engineering Reference Manual for the PE Exam*, Professional Publications, 2013.
- [82] https://en.wikipedia.org/wiki/Factor_of_safety, [Last access May 20, 2016].
- [83] H.E. Sliney, "Performance of PTFE-lined," *NASA Technical Memorandum 82779*, Lewis Research Center, Cleveland, OH, 1982.
- [84] P.J. Raea and D.M. Dattelbaum, "The properties of poly(tetrafluoroethylene) (PTFE) in compression," *Int. J. for the Science and Technology of Polymers*, vol. 45, pp. 7615–7625, 2004.
- [85] A.S. Khan and X. Wang, *Strain Measurements and Stress Analysis*, Pearson, 1st Ed., 2000.
- [86] D. Roylance, *Mechanical Properties of Materials*, Massachusetts Institute of Technology, 2008.
- [87] M. Vable, *Mechanics of Materials*, Michigan Technological University, 2nd Ed., 2012.
- [88] I.C. Jong, B.G. Rogers, and W. Springer, "Teaching von Mises stress: from principal axes to non-principal axes", *American Society for Engineering Education*, 2009.
- [89] T. McGraw, *Focus on Physical Science*, McGraw-Hill, 2015.
- [90] B.P. Epps, M.J. Stanway, and R.W. Kimball, *OpenProp: An open-source design tool for propellers and turbines*, Massachusetts Institute of Technology, 2009.
- [91] R.S. Duelley, "Autonomous underwater vehicle propulsion design," Master's thesis, *Virginia Polytechnic Institute and State University*, VA, 2010.
- [92] F. Fischer, "Thermoplastics: The Best Choice for 3D Printing", *Stratasys Inc.*, 2011.
- [93] <http://bruxy.regnet.cz/programming/rs485/pelco-d.pdf>, [Last access May 20, 2016].
- [94] http://www.hexin-technology.com/shared/manual/HXSP-485_RS-232_RS-485_UserManual.pdf, [Last access May 20, 2016].
- [95] <http://what-when-how.com/space-science-and-technology/earth-orbiting-satellite-theory/>, [Last access May 20, 2016].
- [96] <http://www.perfectlogic.com/articles/avionics/flightdynamics/flightpart2.html>, [Last access May 20, 2016].
- [97] F.J. Regan, S.M. Anandakrishnan, N.S.F. Regan, and S. Anandakrishnan, *Dynamics of Atmospheric*, American Institute of Aeronautics and Astronomy, 1993.
- [98] R. Goodman and Zavorotniy, *Newton's Law of Universal Gravitation*, New Jersey Center for Teaching and Learning, NJ, 2009.
- [99] F.M. White, *Fluid Mechanics*, McGraw-Hill Education, 8th Ed., 2015.
- [100] <https://en.wikipedia.org/wiki/Buoyancy>, [Last access May 20, 2016].
- [101] S.K. Lee and T.H. Joung, "Evaluation of the added mass for a spheroid-type unmanned underwater vehicle by vertical planar motion mechanism test", *Int. J. of Naval Architecture and Ocean Eng.*, vol. 3, pp. 174-180, 2014.
- [102] G.P. Galdi, *An Introduction to the Mathematical Theory of the Navier-Stokes Equations*, Springer Monographs in Mathematics, 2011.
- [103] <http://www.calqlata.com/productpages/00021-help.html>, [Last access May 20, 2016].
- [104] P. Ridao, J. Batlle, and M. Carreras, *Dynamics Model of an Underwater Robotic Vehicle*, University of Girona, Spain, 2001.

- [105] S.T. Chuan, "Modeling and simulation of the autonomous underwater vehicle Autolytus," Master's Thesis, *Massachusetts Institute of Technology*, 1999.
- [106] C.S. Chin, M.W.S. Lau, E. Low, and G.G. Seet, "Software for modelling and simulation of a remotely operated vehicle," *Int. J. Simulation Modeling*, vol. 5, pp. 114-125, 2006.
- [107] C.S. Chin, "Systematic Modeling and model-based simulation of a remotely-operated vehicle using MATLAB and Simulink," *Int. J. of Modeling, Simulation, and Scientific Computing*, vol. 2, pp. 481-511, 2011.
- [108] I. Faragó, *Numerical Methods for Ordinary Differential Equations*, http://www.cs.elte.hu/~faragois/ODE_angol.pdf, [Last access May 20, 2016].
- [109] L.G. García-Valdovinos *et al.*, "Modelling, design and robust control of a remotely operated underwater vehicle," *Int. J. of Advanced Robotic Systems*, vol. 11, pp. 1-16, 2014.
- [110] S.M. Saravanakumar, T. Thiruppathi, and T. Asokan, "Design of interactive simulation system and graphical user interface for underwater vehicle control and navigation," *ISOPE Ocean Mining Symp.*, Chennai, India, 2009.
- [111] R.D. Christ and R.L. Wernli Sr, *The ROV Manual: A User Guide for Observation Class Remotely Operated Vehicles*, Elsevier, 2007.
- [112] <http://www.mfg.mtu.edu/cyberman/environment/mpw/wastestreams/trad/was.html>, [Last access May 20, 2016].
- [113] V.P. Astakhov and S. Shvets, "The assessment of plastic deformation in metal cutting," *J. Materials Processing Technology*, vol. 146, pp. 193-202, 2004.
- [114] <https://en.wikipedia.org/wiki/Gasket>, [Last access May 20, 2016].
- [115] https://en.wikipedia.org/wiki/Liquid_paraffin, [Last access May 20, 2016].
- [116] M. Martinez-Maldonado, *Renal Pharmacology*, Springer Science, 1978.
- [117] Acronal® acrylic resins, BASF Corporation, North Carolina, NC, 2008.
- [118] R.C. Briggs, "Mechanical design of a self-mooring autonomous underwater vehicle", Master's thesis, *Virginia Polytechnic Institute and State University*, VA, 2010.
- [119] https://www.parker.com/literature/ORD%205700%20Parker_O-Ring_Handbook.pdf, [Last access May 20, 2016].
- [120] i200, 4th Gen. Intel® Core™ i3/i5/i7, Intel, 2015.
- [121] UM10204I2C-bus specification and user manual, NXP Semiconductors N.V. 2014.
- [122] R. Kamal, *Devices and Communication Buses for Devices Network*, Chapter-3, McGraw-Hill Education, 2008.
- [123] MPU-9250 Product Specification, Document Number: PS-MPU-9250A-01, InvenSense Inc, 2014.
- [124] Three-axis Compass HMC6343, Honeywell's Magnetic Sensors, 2014.
- [125] <https://cdn.sparkfun.com/datasheets/Sensors/Magneto/HMC6343%20Breakout-v10.pdf>, [Last access May 20, 2016].
- [126] MS5803-14BA Miniature 14 bar Module, Measurement Specialties Inc., 2012.
- [127] Sensirion SHT11 Sensor Module #28018, *Precision Temperature and Humidity Measurement*, Parallax, Inc., 2003.
- [128] SHT1x Humidity and Temperature Sensor IC, Sensirion Sensor Comp., 2011.
- [129] http://www.atmel.com/images/atmel-2490-8-bit-avr-microcontroller-atmega64-1_datasheet.pdf, [Last access May 20, 2016].

- [130] *FSK/ASK Transceiver IC*, Analog Devices, Inc., 2012.
- [131] *KeyStone Architecture Universal Asynchronous Receiver/Transmitter*, Texas Instruments, 2010.
- [132] <http://www.dorji.com/docs/data/DRF7020D13.pdf>, [Last access May 20, 2016].
- [133] <http://www.oktabit.gr/images/pdfs/176-85-LOCC615.pdf>, [Last access May 20, 2016].
- [134] <https://en.wikipedia.org/wiki/Photoresistor>, [Last access May 20, 2016].
- [135] [http://www.buehlermotor.de/C12572D40025EAF8/vwContentByKey/W274AHF5391WEBRDE/\\$File/DC-Motor_51x73_1.13.044.0XX.pdf](http://www.buehlermotor.de/C12572D40025EAF8/vwContentByKey/W274AHF5391WEBRDE/$File/DC-Motor_51x73_1.13.044.0XX.pdf), [Last access May 20, 2016].
- [136] <http://www.ma-boite-a-archives.fr/Moons-hybrid-steppermotors.pdf>, [Last access May 20, 2016].
- [137] http://www.motionusa.com.s3-website-us-east-1.amazonaws.com/moons/Hybrid_Stepper_Motors_16HS_Series.pdf, [Last access May 20, 2016].
- [138] <http://www.xq-power.com/XQ-S40/>, [Last access May 20, 2016].
- [139] BTN7971B High Current PN Half Bridge NovalithIC, *Infineon Techn.*, 2008.
- [140] V. Paul, and J. Jones, M, "Robust real-time face detection," *Int. J. of Computer Vision*, vol. 57, pp. 137-154, 2004.
- [141] H.A. Kwang, D.H. Yoo, S.U. Jung, and M.J. Chung, "Robust multi-view face tracking," *IEEE Int. Conf. on Intelligent Robots and Systems*, St. Louis, MO, 2009.
- [142] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Robust object tracking using joint color-texture histogram," *Int. J. of Pattern Recognition and Artificial Intelligence*, vol. 23, pp. 1245–1263, 2009.
- [143] Y.M. Lui, J.R. Beveridge, and L.D. Whitley, "Adaptive appearance model and condensation algorithm for robust face tracking," *IEEE Trans. Systems, Man and Cybernetics - part A*, vol. 40, pp. 437–448, 2010.
- [144] R. Iraj, M.T. Manzuri-Shalmani, A.H. Jamal, and A.R. Sefidpour, "A real-time color-independent method for multiple faces tracking," *IEEE Int. Conf. on Cognitive Informatics*, Tehran, Iran, 2007.
- [145] K. Shearer, K. D. Wong, and S. Venkatesh, "Combining multiple tracking algorithms for improved general performance," *Pattern Recognition*, vol. 34, pp. 1257-1269, 2001.
- [146] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Institute of Automation, Chinese Academy of Sciences, Beijing*, pp. 585-601, 2001.
- [147] S.H. Shaikh, K. Saeed, and N. Chaki, *Moving Object Detection Using Background Subtraction*, Springer Int. Publishing, 2014.
- [148] P. Wang, M.B. Green, Q. Ji, and J. Wayman, "Automatic eye detection and its validation," *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Washington, DC, 2005.
- [149] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology J.*, vol. 2, 1998.
- [150] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603–619, 2002.

- [151] Y. Cheng, "Mean shift, mode seeking and clustering," *IEEE Trans. Pattern Recognition*, vol. 17, pp. 790–799, 1995.
- [152] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, pp. 564–575, 2003.
- [153] I. Haritaoglu and M. Flickner, "Detection and tracking of shopping groups in stores," *IEEE Conf. Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001.
- [154] K. Nummiaro, E. Koller-Meier and L. V. Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, pp. 99–110, 2003.
- [155] C. Yang, D. Ramani, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," *IEEE Conf. Computer Vision and Pattern Recognition*, 2005.
- [156] M.-P. Dubuisson-Jolly and A. Gupta, "Color and texture fusion: application to aerial image segmentation and GIS updating," *Image and Vision Computing*, vol. 18, pp. 823-832, 2000.
- [157] C. Pallavi, C. Hingane, and S.A. Shirsat, "Object tracking using joint color-texture histogram," *Int. J. of Science and Research*, vol. 4, pp. 2603-2606, 2015.
- [158] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–986, 2002.
- [159] M.K. Vairalkar and S.U. Nimbhorkar, "Edge detection of images using Sobel operator," *Int. J. of Emerging Technology and Advanced Eng.*, vol. 2, pp. 291-296, 2012.
- [160] R. Maini, H. Aggarwal, "Study and comparison of various image edge detection techniques," *Int. J. of Image Processing*, vol. 3, pp. 1-11, 2009.
- [161] M. Shahrokhi and A. Zomorodi, *Comparison of PID Controller Tuning Methods*, http://www.ie.itcr.ac.cr/einteriano/control/clase/Zomorodi_Shahrokhi_PID_Tunning_g_Comparison.pdf, [[Last access May 20, 2016].
- [162] Y. Sharma and A. Patra, "Comparative analysis of PI and PID controllers for speed control of DC shunt motor," *Int. J. of Science, Engineering and Technology Research*, vol. 4, pp. 2736-2739, 2015.
- [163] J. Petrich, W.L. Neu, and D.J. Stilwell, "Identification of a simplified AUV pitch axis model for control design: Theory and experiments," *IEEE Oceans Conf.*, Vancouver, BC, pp. 1-7, 2007.
- [164] J.H.A.M. Vervoort, "Modeling and control of an unmanned underwater vehicle, Master's Report, *University of Canterbury*, New Zealand, 2008.
- [165] J. Petrich and D.J. Stilwell, "Model simplification for AUV pitch-axis control design," *Ocean Engineering*, vol. 37, pp. 638–651, 2010.
- [166] D.E. Tamir, N.D. Rishe, and A. Kandel, *Fifty Years of Fuzzy Logic and its Applications*, Springer, 2015.
- [167] L.A. Zadeh, "Fuzzy logic computing with words," *IEEE Trans. Fuzzy Systems*, vol. 4, pp. 103-111, 1996.
- [168] W. Banks and G. Hayward, *Fuzzy Logic in Embedded Microcomputers and Control Systems*, Byte Craft Ltd, 2002.
- [169] G. Metcalfe, N. Olivetti, and D.M. Gabbay, *Proof Theory for Fuzzy Logics*, Springer, 2008.