DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEMARINE AUV: DYNAMIC MOTION CONTROLLED UNDERWATER ROBOT FOR SURVEYING VESSELS AND SHALLOW WATER ENVIRONMENT

by
Kerim Deniz KAYA

March, 2023
İZMİR

DEMARINE AUV: DYNAMIC MOTION CONTROLLED UNDERWATER ROBOT FOR SURVEYING VESSELS AND SHALLOW WATER ENVIRONMENT

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in
Mechatronics Engineering, Mechatronics Engineering Program

by
Kerim Deniz KAYA

March, 2023
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"DEMARINE AUV: DYNAMIC MOTION CONTROLLED UNDERWATER ROBOT FOR SURVEYING VESSELS AND SHALLOW WATER ENVIRONMENT"** completed by **KERİM DENİZ KAYA** under supervision of **ASSOC. PROF. DR. AYTAÇ GÖREN** we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

..……

Assoc. Prof. Dr. Aytaç GÖREN

Supervisor

…….

Prof. Dr. Zeki KIRAL

Thesis Committee Member

.....

Prof. Dr. Yalçın ÇEBİ

Thesis Committee Member

…….

Prof. Dr. Levent ÇETİN

Examining Committee Member

…...……..……

Assoc. Prof. Dr. S. Murat BAĞDATLI

Examining Committee Member

……

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

First and foremost, I would like to express my appreciation to my supervisor Assoc. Prof. Dr. Aytaç GÖREN for giving me the opportunity to work in a significant area of engineering science and supporting me during every section of my study. His confidence in my abilities and the time he dedicated to me for his guidance on my work enabled me to be a better researcher at the point where I am now. I also have to thank him for encouragement in project management and team leadership and for expanding my research network in multidisciplinary fields.

My sincere gratitude also goes for my doctoral thesis committee member Prof. Dr. Zeki KIRAL with whom I have been following his lectures since my undergraduate education years and whose role I modeled in my academic career. In addition to his contributions and guidance regarding the content and course of my study, I built the sections of my thesis that require knowledge of system modeling and automatic control theory on the foundation he provided.

I would like to express my deepest gratitude to my doctoral thesis committee member Prof. Dr. Yalçın ÇEBİ for his precious contributions and beneficial discussions that provide elegant solutions during doctoral thesis committee meetings, especially in the computer science field.

And I would like to thank Lec. Semih YILMAZ and Res. Asst. Kubilay BAYRAMOĞLU for sharing their scientific experience required for completing the theoretical and numerical parts of my thesis.

I would like to thank the esteemed researcher Halit Ege CEYHUN for his vital contributions during the design and implementation of advanced controllers.

I would also like to thank Fehim KUŞ and Jamal ESENKANOVA for helping me about technical issues throughout my thesis.

I am also indebted to undergraduate and graduate students who assisted in the implementation and testing phases of the first and second DEMarine AUV prototype as research and graduation work groups.

I would like to state that I owe a lot to my beloved Gamze AKPINAR, who always takes care to keep me alive with her beautiful energy and endless love.

And I am grateful for everything to my brother Onur KAYA, my father Mehmet KAYA and my devoted mother Elif Biray KAYA, all of whom are valuable members of my family, for being with me throughout my life.

This work was supported by Dokuz Eylul University Scientific Research Coordination Unit [Grant Number: 2019.KB.FEN.011]; and The Scientific and Technological Research Council of Turkey [Grant Number: 111E689]

Kerim Deniz Kaya

DEMARINE AUV: DYNAMIC MOTION CONTROLLED UNDERWATER ROBOT FOR SURVEYING VESSELS AND SHALLOW WATER ENVIRONMENT

ABSTRACT

In recent decades, the importance of unmanned underwater operations concerning science, energy, maritime and military is increasing due to energy savings, occupational safety, operational accuracy and environmental challenges.

Within the scope of this thesis, the design, manufacturing, simulation, analysis and experimental tests have been carried out for an autonomous type underwater robot DEMarine AUV which can follow predetermined trajectories to operate under the floating line of marine vessels.

On an introductory basis, an overview of motivation, objective, original contribution and structuring of the subjects were provided.

Next comes a preliminaries and fundamentals section that includes the historical development, state of art, literature review and detailed theoretical approaches of AUV modelling, hydrodynamics and motion control.

As a continuation, the prototype was implemented by applying the design criteria determined in accordance with the mission descriptions. With underwater tests, the implementation was finalized with continuously developed mechanical hardware, electronic hardware, sensors, propulsion and software.

In order to apply control strategies and obtain controller inputs, determining the hydrodynamic coefficients constitutes an important part of the study. Therefore, two different orbital geometries were modelled in terms of coordinates and formulated with curve fitting method.

After the analysis carried out with the Computational Fluid Dynamics (CFD) method and the hydrodynamic data were obtained, three different controllers, namely PID, Linear Quadratic Regulator (LQR) and LQR + Sliding Mode Control (SMC), were designed and the results were compared by performing tests on the simulation model.

Keywords: Autonomous underwater vehicle, AUV, trajectory planning, motion control, hydrodynamic analysis

DEMARINE AUV: DENİZ TAŞITLARI VE SIĞ SU ÇEVRESİ DENETİMİ İÇİN DİNAMİK HAREKET KONTROLLÜ SUALTI ROBOTU

ÖZ

Son yıllarda enerji tasarrufu, iş güvenliği, operasyonel hassasiyet ve çevresel zorluklar nedeniyle bilim, enerji, denizcilik ve askeri alanlarda insansız sualtı operasyonlarının önemi artmaktadır.

Bu tez kapsamında, deniz araçlarının yüzer hat seviyesinin altında görev yapmak üzere önceden belirlenmiş yörüngeleri takip edebilen otonom tip bir sualtı robotu olan DEMarine AUV'nin tasarımı, üretimi, simülasyonu, analizi ve deneysel testleri gerçekleştirilmiştir.

İlk aşamada tezin motivasyonu, amacı, özgün değeri ve çalışmaların yapılanmasına dair genel bir perspektif sunulmuştur.

Sonrasında, AUV'lerin tarihsel gelişimi, güncel bilimsel ilerlemeler, literatür araştırması ve AUV'lerin modelleme, hidrodinamik hesaplamalar ve hareket kontrolünün teorik yaklaşımlarına dair temel bilgiler sunulmuştur.

Çalışmanın devamında, görev tanımlarına bağlı kalarak belirlenen tasarım kriterleri uygulanarak AUV prototipi gerçeklenmiştir. Sualtı testleri ile sürekli geliştirilen mekanik donanım, elektronik donanım, sevk sistemleri ve yazılımlar ile DEMarine AUV prototipi nihai formuna getirilmiştir.

Kontrol stratejilerini uygulamak ve kontrol girdilerini elde etmek amacıyla hidrodinamik katsayıların belirlenmesi çalışmanın önemli bir bölümünü oluşturmaktadır. Bu nedenle iki farklı yörüngenin geometrilerine ait koordinat verileri oluşturulmuş ve eğri uydurma yöntemi ile formüle edilmiştir.

Hesaplamalı akışkanlar dinamiği (CFD) yöntemi ile yapılan analizler sonucunda hidrodinamik parametrelerin elde edilmesinin ardından PID, Doğrusal Karesel Düzenleyici (LQR) ve LQR + Kayan Kipli Kontrol (SMC) olmak üzere üç farklı kontrolör tasarlanmış ve benzetim modelleri oluşturulup test edilerek hata performansları karşılaştırılmıştır.

Anahtar Kelimeler: Otonom sualtı aracı, AUV, yörünge planlama, hareket kontrolü, hidrodinamik Analiz

CONTENTS

Page

Ph.D. THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGEMENTS iii
ABSTRACT v
ÖZ vii
CONTENTS ix
LIST OF FIGURES xiii
LIST OF TABLES xvi
ABBREVIATIONS xvii
LIST OF SYMBOLS xix

CHAPTER ONE - INTRODUCTION 1

1.1 Overview 1
1.2 Motivation and Major Work 1
1.3 Objective of The Thesis 2
1.4 Novel Contribution 2
1.5 Structure of The Thesis 3

CHAPTER TWO - PRELIMINARIES AND FUNDAMENTALS 5

2.1 Historical Background 5
2.2 State of Art and Literature Review 12
2.3 AUV Modelling 18
2.3.1 AUV Kinematics 18
2.3.2 AUV Dynamics 21
2.3.3 AUV Hydrodynamics 21
2.4 AUV Motion Control 23

2.4.1 PID Controller Design 24
2.4.2 Linear Quadratic Regulator Design 29
2.4.3 Sliding Mode Controller Design 31
2.4.4 Lyapunov Controller Rule 32

CHAPTER THREE - DEMARINE AUV INFRASTRUCTURE 35

3.1 Mechanical Design 35
3.1.1 Frame Construction 35
3.1.2 Closed Enclosures and Sealing 36
3.2 Thrust and Propulsion System 39
3.3 Integrated Software and Electronics 41
3.3.1 Master-Slave Connection 43
3.3.2 Human Machine Interface Communication 46
3.3.3 Embedded Sensing System and Instrumentation 49
3.3.3.1 Hydrostatic Pressure and Depth Sensing 50
3.3.3.2 Distance measurement 51
3.3.3.3 Visual data 52
3.3.3.4 Inertial measurements 54
3.4 Experimental Setup and Underwater Tests 59

CHAPTER FOUR - HYDRODYNAMIC ANALYSIS 61

4.1 Determination of Hydrodynamic Characteristics: Scenario 1 61
4.1.1 Trajectory Planning of Scenario 1 63
4.1.2 Energy Modelling of Scenario 1 67
4.1.3 Computational Study of Scenario 1 68

4.1.3.1 Meshing and Boundary Conditions of Scenario 1 70
4.1.3.2 Numerical Validation of Scenario 1 72
4.1.3.3 Mesh Independency of Scenario 1 73
4.1.4 Results and Discussion of Scenario 1 74
4.2 Determination of Hydrodynamic Characteristics: Scenario 2 83
4.2.1 Trajectory Planning of Scenario 2 83
4.2.2 Energy Modelling of Scenario 2 85
4.2.3 Computational Study of Scenario 2 88
4.2.3.1 Meshing and Boundary Conditions of Scenario 2 88
4.2.3.2 Numerical Validation of Scenario 2 90
4.2.3.3 Mesh Independency of Scenario 2 90
4.2.4 Results and Discussion of Scenario 2 91

CHAPTER FIVE - CONTROL SYSTEM DESIGN 96

5.1 Scenario 1: Reference Trajectory Input and PID Control 96
5.1.1 Simulation Model 96
5.1.2 Control System Inputs and Boundary Conditions 97
5.1.3 Simulation Results and Performance Analysis 104
5.2 Scenario 2: Linear Quadratic Regulator Controller Design 106
5.2.1 Simulation Model 106
5.2.2 Control System Inputs and Boundary Conditions 115
5.2.3 Simulation Results and Performance Analyses 116
5.3 Scenario: 3 Sliding Mode and LQR Hybrid Control Architecture 119
5.3.1 Simulation Model 120
5.3.2 Control System Inputs and Boundary Conditions 125
5.3.3 Simulation Results and Performance Analyses 126

CHAPTER SIX - GENERAL DISCUSSION 128

CHAPTER SEVEN - CONCLUSIONS AND FUTURE WORKS 136

7.1 Conclusions 136

7.2 Future Works 137

REFERENCES 139

APPENDICES 151

Appendix 1 DEMarine AUV 3D Model 151

Appendix 2 DEMarine AUV Prototype Technical Drawing 152

Appendix 3 Source Codes 153

Appendix 4 Simulink Models 163

LIST OF FIGURES

Page

Figure 1.1 Schematic representation of thesis structure..... 4
Figure 2.1 Motion variables of AUV 19
Figure 2.2 Standard PID control 24
Figure 2.3 Comparison of P, PI, PD and PID control responses to a step input..... 25
Figure 2.4 Principal block diagram of control system 27
Figure 2.5 Equivalent transfer function based block diagram 28
Figure 3.1 Mechanical structure components of the DEMarine AUV prototype 36
Figure 3.2 Exploded view of closed enclosures and sealing..... 37
Figure 3.3 Sealing works on cable entry points 38
Figure 3.4 Sealing tests 38
Figure 3.5 Exploded view of T200 thruster 40
Figure 3.6 Front, left, top and perspective views of the DEMarine AUV prototype. 41
Figure 3.7 Block diagram abstraction of DEMARINE AUV electronic setup..... 42
Figure.3.8 PC and embedded computer connected to the common network..... 44
Figure 3.9 Master device configuration 44
Figure 3.10 Slave device configuration 45
Figure 3.11 Master-Slave connection test..... 45
Figure 3.12 Human machine interface communication and software..... 46
Figure 3.13 Brushless DC motors remote control diagram..... 47
Figure 3.14 Open loop control system electronics..... 47
Figure 3.15 Open loop control system electronics..... 48
Figure 3.16 Distances of the center of mass from thrust axes 49
Figure 3.17 Pressure and Depth Sensor System Circuit Diagram 50
Figure 3.18 Distance sensor system circuit diagram..... 51
Figure 3.19 Sonar testing and graphical user interface 52
Figure 3.20 Camera module instrumentation diagram..... 53
Figure 3.21 Camera module test setup..... 53
Figure 3.22 Inertia measurement system instrumentation diagram 54
Figure 3.23 Position vectors simultaneously visualized on RVIZ..... 58

Figure 3.24 Filtered and unfiltered IMU data 59
Figure 3.25 Underwater experiments in test pool 60
Figure 4.1 General view of research vessel DEU 1 62
Figure 4.2 Cubic spline curve fitted trajectory 65
Figure 4.3 Derivation and tangential intersection points along ideal trajectory 66
Figure 4.4 DEMarine AUV 69
Figure 4.5 Flow domain setup 70
Figure 4.6 Mesh structure side section 71
Figure 4.7 Mesh structure front section 71
Figure 4.8 Numerical validation results of Scenario 1 73
Figure 4.9 Mesh independency control for 1M, 2M and 3M mesh numbers 73
Figure 4.10 Velocity distribution for roll angles of 5°, 10°, 15°, 20°, 25°, 30° 75
Figure 4.11 Pressure distribution for roll angles of 5°, 10°, 15°, 20°, 25°, 30° 76
Figure 4.12 Velocity distribution for yaw angles of 5°, 10°, 15°, 20°, 25°, 30° 77
Figure 4.13 Pressure distribution for yaw angles of 5°, 10°, 15°, 20°, 25°, 30° 78
Figure 4.14 Hydrodynamic derivatives on time and roll position domain 79
Figure 4.15 Hydrodynamic derivatives on time and yaw position domain 80
Figure 4.16 Logarithmic trajectory model of the AUV 84
Figure 4.17 Orbital forces acting on the AUV 86
Figure 4.18 Flow domain setup of scenario 2 88
Figure 4.19 Mesh structure of the flow domain 89
Figure 4.20 Numerical validation results of Scenario 2 90
Figure 4.21 Mesh independency analysis results 91
Figure 4.22 Pressure distribution for 9 different angular positions 92
Figure 4.23 Velocity distribution for 9 different angular positions 93
Figure 4.24 Total power with respect to radius of curvature and orientation 94
Figure 5.1 AUV simulation model 97
Figure 5.2 Translational reference (a) X coordinates (b) Y coordinates (c) X step input (d) Y step input 98
Figure 5.3 Open-loop test setup of T200 thrusters (MathWorks Student Competitions Team, 2023) 99
Figure 5.4 System identification workflow of T200 thrusters 100

Figure 5.5 Feedback structure with PID controllers (MathWorks Student Competitions Team, 2022) 101
Figure 5.6 Step input responses; (a) and (b): X and Y axes with default coefficients, (d) and (e): X and Y axes with optimized coefficients 102
Figure 5.7 Reference tracking with default PID parameters 104
Figure 5.8 Reference tracking with default PID parameters 104
Figure 5.9 Reference tracking with optimized PID parameters 105
Figure 5.10 Stabilized Z position reference tracking 105
Figure 5.11 Eigenvalues of the system 116
Figure 5.12 LQR controller design simulation model 117
Figure 5.13 X, Y and Z responses of the system controlled by LQR 117
Figure 5.14 $\dot{X}$, $\dot{Y}$ *and* $\dot{Z}$ responses of the system controlled by LQR 118
Figure 5.15 Φ, θ *and* ψ responses of the system controlled by LQR 118
Figure 5.16 $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ responses of the system controlled by LQR 119
Figure 5.17 Sliding mode and LQR hybrid control architecture 120
Figure 5.18 Eigenvalues of LQR + SMC hybrid control system 126
Figure 5.19 X, Y and Z responses of the system controlled by LQR + SMC hybrid control system 126
Figure 5.20 3D plot of the AUV trajectory and the reference 127

LIST OF TABLES

Page

Table 2.1 6-DOF states of AUV 18
Table 3.1 Technical specifications of the 1st, 2nd and 3rd generation thrusters..... 39
Table 3.2 Performance and maneuver specifications of DEMarine AUV 60
Table 4.1 Technical specifications of research vessel DEU 1 62
Table 4.2 Half breadth table of research vessel DEU 1 63
Table 4.3 Technical specifications of DEMarine AUV 69
Table 4.4 Boundary conditions of CFD analysis 72
Table 4.5 Trajectory planning algorithm results 74
Table 4.6 Energy modelling algorithm results 81
Table 4.7 Performance evaluation table of DEMarine AUV 82
Table 4.8 Geometric parameters of AUV trajectory 85
Table 4.9 Numerical analysis boundary conditions 89
Table 4.10 AUV resultant velocity and corrected velocity data sets 92
Table 4.11 AUV Thrust and power parameters 93
Table 5.1. Results of system identification study 100
Table 5.2 PID controller parameters 102
Table 5.3 PID controller parameters for Z axis..... 103

ABBREVIATIONS

AUG :Autonomous Underwater Glider
AUSI :Autonomous Undersea System Institute
AUV :Autonomous Underwater Vehicle
CFD :Computational Fluid Dynamics
CNC :Computer Numerical Control
CURV :Cable-Controlled Underwater Research Vehicle
DOF :Degree of Freedom
ESC :Electronic Speed Controller
FVM :Finite Volume Method
GPS :Global Positioning System
HMI :Human Machine Interface
I2C :Inter-Integrated Circuit
IMTP :Institute of Marine Technology Problems
IMU :Inertial Measurement Unit
LQR :Linear Quadratic Regulator
LTI :Linear Time Invariant
MAST :Marine Science and Technology
MAUVE :Mini Autonomous Underwater Vehicle
MIMO :Multiple Input - Multiple Output
MPC :Model Predictive Controller
MUV :Multiple Unmanned Vehicle
P :Proportional
PD :Proportional Derivative
PI :Proportional Integral
PID :Proportional Integral Derivative
PLA :Polylactic Acid
PMMA :Polymethyl Methacrylate
PWM :Pulse Width Modulation
RANS :Reynolds Averaged Navier-Stokes
REMUS :Remote Environment Monitoring Units

RNG :Re-Normalisation Group

ROS :Robot Operating System

ROV :Remote Operated Vehicle

RPM :Rate Per Minute

RVIZ :ROS Visualization

RX :Receive X

SAUV :Solar Autonomous Underwater Vehicle

SCCB :Serial Camera Control Bus

SCL :Serial Clock

SDA :Serial Data

SMC :Sliding Mode Control

SNAME :Society of Naval Architects and Marine Engineers

SPURV :Self-Propelled Underwater Research Vehicle

SST :Shear Stress Transport

TX :Transmit X

UARS :Unmanned Arctic Research Submersible

UART :Universal Asynchronous Receiver Transmitter

UFFS :Unmanned Free swimming Submersible

UUV :Unmanned Underwater Vehicle

WL :Water Line

LIST OF SYMBOLS

ϕ	:Roll Angle
θ	:Pitch Angle
ψ	:Yaw Angle
x	:Surge Displacement
y	:Sway Displacement
z	:Heave Displacement
u	:Linear Velocity Around Xb axis
v	:Linear Velocity Around Y_b axis
w	:Linear Velocity Around Z_b axis
p	:Angular Velocity Around X_b axis
q	:Angular Velocity Around Y_b axis
r	:Angular Velocity Around Z_b axis
X	:Force Along X_b Axis
Y	:Force Along Y_b Axis
Z	:Force Along Z_b Axis
K	:Moment Along X_b Axis
M	:Moment Along Y_b Axis
N	:Moment Along Z_b Axis
X_b	:Body-Fixed Frame X Axis
Y_b	:Body-Fixed Frame Y Axis
Z_b	:Body-Fixed Frame Z Axis
O_b	:Body-Fixed Frame Center of Origin
X_e	:Earth-Fixed Frame X Axis
Y_e	:Earth-Fixed Frame Y Axis
Z_e	:Earth-Fixed Frame Z Axis
O_e	:Earth-Fixed Frame Center of Origin
η_1	:Position Vector
η_2	:Orientation Vector
ν_1	:Linear Velocity Vector
ν_2	:Angular Velocity Vector

τ_1 :External Force Vector

τ_2 :External Moment Vector

$J(\eta)$:Euler Transformation Matrix

M :Mass Matrix

M_{RB} :Generalized Constant Inertia Matrix

M_A :Added Inertia Matrix

$C(v)$:Coriolis and Centripetal Forces Matrix

$D(v)$:Hydrodynamic Damping Matrix

D :Linear Damping Matrix

$D_n(v)$:Nonlinear Damping MAtrix

$g(\eta)$:Buoyancy and Gravitational Effect Vector

$\tau(v,u)$:Resultant Force and Moment Vector

ρ :Flow Density

u :Velocity Vector Field

t :Time

p_f :Fluid Pressure

μ :Fluid Viscosity

f :Body Force

G_k :Turbulent Kinetic Energy

G_b :Buoyant Turbulent Kinetic Energy

Y_m :Dissipation Rate

ε :Turbulent Dissipation Rate

μ :Dynamic Viscosity

μ_t :Turbulent Dynamic Viscosity

C_μ :Viscosity Constant

$C_{1\varepsilon}$:Adjustable Constant Coefficient

$C_{2\varepsilon}$:Adjustable Constant Coefficient

$C_{3\varepsilon}$:Adjustable Constant Coefficient

σ_k :Turbulence Prandtl Number for k

σ_ε :Turbulence Prandtl Number for ε

$c(t)$: Control Signal Output

$C(s)$:Control Signal on s-Domain

$E(s)$:Error Signal on s-Domain

K_P :Proportional Gain

K_I :Integral Gain

K_D :Derivative Gain

T_i :Integral Time Constant

T_D :Derivative Time Constant

$s(t)$:Step Input

s_0 :Open-Loop Step Input

c_0 :Open-Loop Control Signal

s_∞ :Step Input Intensity

c_∞ :Control signal Intensity

K_0 :Open-Loop Gain

Δc :Change in Output Signal

Δs :Change in Input Signal

t_d :Dead Time Constant

τ_t :Time Constant

$H_p(s)$:Process Transfer Function

$H_u(s)$:Partial Transfer Function without Time Delay

$H_m(s)$:Sensor and Scaling Block Transfer Function

$y_{SP}(s)$:Setpoint Signal

$y_{mSP}(s)$:Measurement Setpoint Signal

$e_m(s)$:Laplace Transform of Control Error

$y_m(s)$:Process Measurement

$y(s)$:Process Output

$U_0(s)$:Nominal Control Variable

$u(s)$:Controller Signal

$v(s)$:Disturbance variable

$H_v(s)$:Disturbance Transfer Function

$H_{vm}(s)$:Equivalent Disturbance Transfer Function

$H_c(s)$:Equivalent Controller Transfer Function

$H_p(s)$:Process Transfer Function

T :Time constant

T_{vm} :Equivalent Time constant

K :Process Gain

T_{vm} :Disturbance Gain

$S(s)$:Sensitivity Function

$L(s)$:Loop Transfer Function

$T(s)$:Tracking Transfer Function

x :State Variable

$\dot{x}$:Derivative of State Variable

x_0 :Initial Condition

A :State Matrix

B :Input Matrix

J :Cost Function

Q :State Weight Matrix

R :Input Weight Matrix

P :Solution Matrix

$s(x)$:Sliding Surface

$h(x)$:Desired Response

$r(t)$:Reference Signal

$\dot{s}$:Derivative of Sliding Variable

$V(x)$:Lyapunov Function

$\dot{V}(x)$:Derivaitve of Lyapunov Function

A_c :Companion Matrix

x_m :Measured State

$\dot{e}$:Time Derivative of Error

u_f :External Input

x_k :Estimated Value

x_{k-1} :Previous Estimated Value

z_K :Measured Value

K_K :Kalman Gain

w_{k-1} :Process Noise

v_k :Measurement Noise

$A_{x_{k-1}}$:Predicted State of the System at Time k

A :State Transition Matrix

B :Control Input Matrix

H :Measurement Matrix

P'_k :Prior Error Covariance

d_k :Straight Distance

z_k :Intersection Nodes

ψ_k :Yaw Angle at Point k

β_k :Slope

$E_{klinear}$:Energy Requirement on Linear Paths

$E_{kangular}$:Energy Requirement on Yaw Motion

F_{ψ_k} :Drag Force Caused by Yaw Motion

F_x :Drag Force Caused by X-axis Motion

b_{AUV} :Width Size of AUV

E_{total} :Total Energy Requirement

P_{avg} :Average Power

Δt :Time of Travel

P_t :Total Power

V :Voltage

C_{batt} :Battery Capacity

I_{batt} :Amperage Draw

t_{batt} :Discharge Time of Battery

l_{nav} :Navigation Range

C_D :Drag Coefficient

C_L :Lift Coefficient

F_y :Lift Force

M_z :Moment Around Z-axis

K_i :Curvature Value at Each Point

d_i :Distance Vector

V_R :AUV Velocity due to Current

V_{AUV} :Speed due to Thrust Applied

V_C :Current Velocity

V_{COR} :Corrected Velocity

V_N :Normalised Velocity

F_C :Current Induced Drag Force

F_T :Heading Thrust

$P_{Heading}$:Heading Power

$P_{Lateral}$:Lateral Power

I :Inertia Tensor

m_g :Weight

$G_\varphi, G_{\theta,} G_\psi$:Gyroscopic Effects

$u_{x,y,z}$:Linear Actuation Terms

$\tau_{x,y,z}$:Rotational Actuation Terms

X_{LQR} :LQR State Vector

$\dot{X}$:Derivative of LQR State Vector

S_{des} :Desired States

u_{des} :Desired Linear States

τ_{des} :Desired Rotational States

Co :Controlability Matrix

Ob :Observability Matrix

F_{dist} :Disturbance Force

S_x, S_y :Sliding Variables

$\dot{S}_x$, $\dot{S}_y$:Derivatives of Sliding Variables

u_{eq} :Equilibirum Part of SMC

U_R :Robust Part of SMC

μ_x, μ_y :Gains of Sliding Mode Controller

σ :Sigmoid Function

CHAPTER ONE
INTRODUCTION

1.1 Overview

Under today's conditions, efforts to develop Unmanned Underwater Vehicles (UUVs) or Unmanned Robotic Vehicles (URVs) with low deployment cost, high durability, high range, high maneuverability and high sensing capacity focus on design, production, control architectures, navigation, intelligent task assignment, embedded sensor systems and data filtering (Kerim Deniz Kaya & Gören, 2015). As a result of these needs the number of reported scientific studies and R&D projects is increasing day by day. The purpose of this overview is to briefly describe the vehicles belonging to the underwater robotic class, structurally and functionally, and to provide details on the classification of these robotic systems. In addition, information will be given on how wide the scope of the application areas of underwater robotic systems, which are widely used in practice, and how strong the potential for future marine applications is.

The application areas and duties of these state-of-art systems, which are quite wide and continue to expand day by day, will be clarified in the following lines, first with general macro expressions including task summary and then specifically for military, academic, scientific, offshore-energy and inland operations.

1.2 Motivation and Major Work

Today, underwater robots designed to serve in scientific, commercial and military fields are generally controlled by open loop control systems, that is, under remote control. This situation requires the presence of an operator standing on a surface station and using the control device to perform the navigation and maneuvering movements of the underwater robot. These robots, which are named as Remote Operated Underwater Vehicles (ROV), may fall short for applications that require high cruise stability and precise trajectory tracking.

For this reason, studies on the realization of underwater robots called Autonomous Underwater Vehicles (AUVs), which include closed loop motion control systems, are gaining importance.

In autonomous underwater vehicles, in order to create systems such as autopilot, guided movement, trajectory tracking, and obstacle avoidance, it is necessary to obtain information of position, orientation, speed and acceleration in six axes in the underwater environment.

This study aims to implement an underwater robot that is capable of underwater exploration in shallow depth waters and surveying wet surfaces of marine vehicles moreover, it is planned to be an experimental instrument in automatic control theory education. The necessity of using domestic technologies mainly for fulfilling the requirements of Marine Science and Maritime Industry has been considered in selecting the Ph.D. subject.

1.3 Objective of The Thesis

The aim of the thesis is to develop an underwater robot prototype with both remote and autonomous features, capable of navigating and maneuvering in accordance with certain task descriptions. In order to achieve motion, observation and tracking precision, many advanced problems must be addressed. Within the scope of the thesis, it is aimed to define these problems and specially to design a correct hydrodynamic model and a properly working control system.

1.4 Novel Contribution

The original scientific value that this thesis study adds to the literature can be expressed in theoretical and practical terms. First of all, structurally mini-sized and open-frame hulls are rarely seen in the historical development of autonomous underwater robots. However, many specific underwater operations today require precision and high maneuverability in tight spaces. The prototypes developed within

the scope of the DEMarine AUV project have a body frame structure and propulsion arrangement that can meet this need. Thus, detailed examination of the wet body parts of surface vehicles and intervention if necessary, and the potentiality of performing missions in shallow waters will increase. This unique structure brought along original studies that contributed to the literature on mathematical modelling, determination of hydrodynamic coefficients and motion control.

1.5 Structure of The Thesis

The general structure of the thesis document showing the main titles and their contents is given in Figure 1.1.

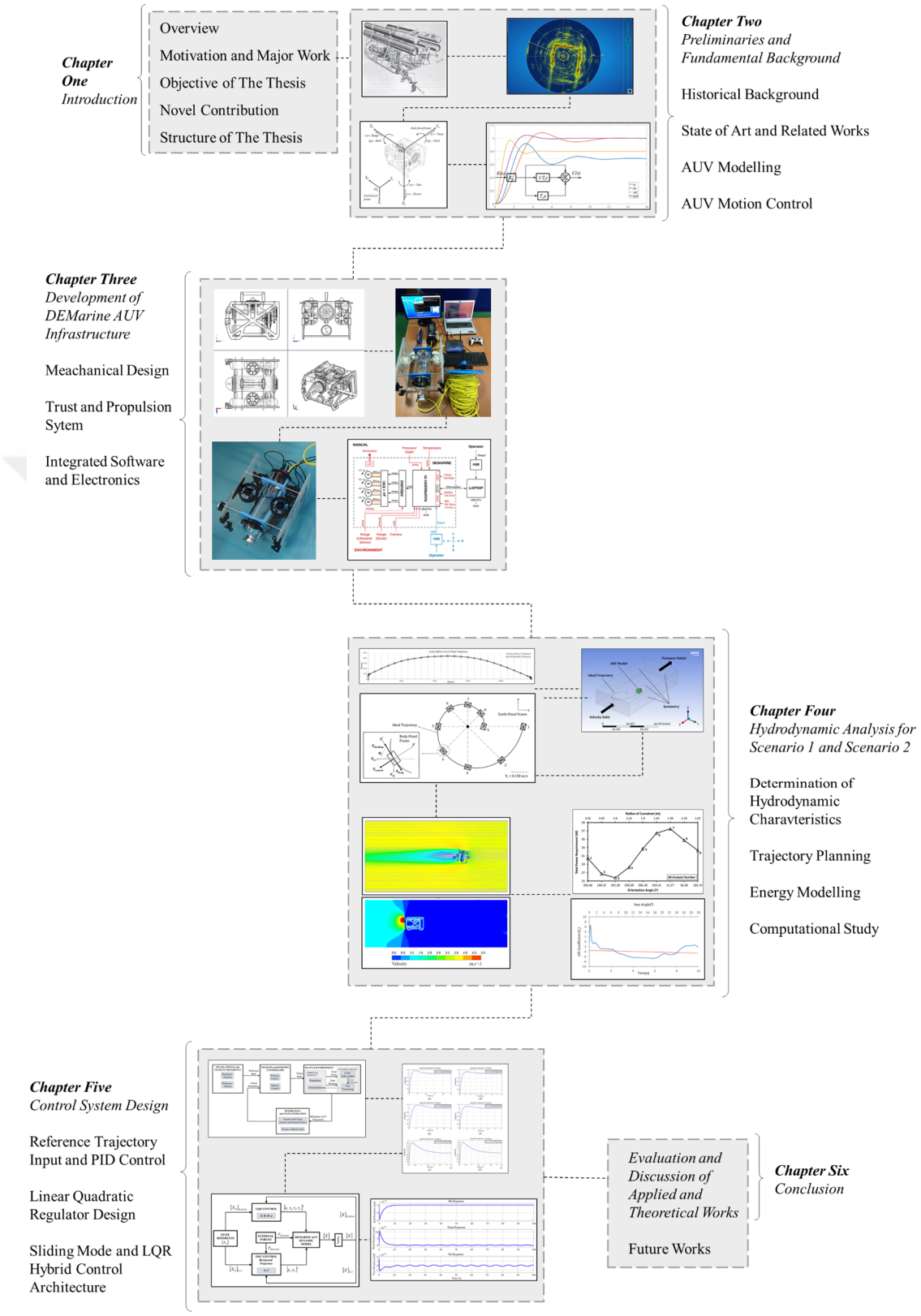


Figure 1.1 Schematic representation of thesis structure

CHAPTER TWO
PRELIMINARIES AND FUNDAMENTALS

2.1 Historical Background

Although the idea-driven design of submarines began in the 16th century, the first unmanned underwater vehicle (UUV) was developed for search and rescue purposes by the US Navy in 1958. One of the first systems implemented was developed at the US Navy Laboratories and was named the Cable-Controlled Underwater Research Vehicle (CURVS) (Michel & Wernli, 1998). The cable-controlled vehicle was used to locate military submarines and equipment lost in different geographic areas in 1963 and 1966 (Craven et al., 1998)..Based on this early stage technology, the first commercial use of UUVs began with the discovery of areas in the North Sea that host offshore oil and gas resources (Vervoort, 2008). Technology continued to evolve with hydraulic and later electrically driven versions of CURV type unmanned underwater systems.

As of 1974, it can be reported that more than 20 systems with various equipment and units have been implemented. The CURV system, which continued to be developed throughout this process, became able to serve in emergency rescue operations with its 3rd version (CURV III). Between the mid-1970s and the early 1980s, the development curve showed an exponential character, and by the end of 1982 there were 500 unmanned underwater systems around the world. After this date, it started to be produced in a more compact size, carrying its own battery for power, with high maneuverability and low cost, in line with specific needs. In particular, apart from the military sector, the activities of the overseas oil and gas industry in the North Sea have been instrumental in both dimensional reduction and developments in the electronics field. Thus, improvements were seen on the reliability problem in the systems that were tried to be developed until that time. With the involvement of marine technology engineering companies established in European countries and Japan, the advancement of technology allowed to reach deeper working depths.

Between 1982 and 1989, 27 different companies were operating, highlighting the concept of Remote Operated Vehicle (ROV). The industrial progress experienced in this period brought with it the organization of ROV themed scientific conferences. With the high funding of R&D studies from the mid-80s, unmanned underwater systems began to become portable and affordable enough to be accessible to academic institutes and civil organizations. This development led to the emergence of variations such as Mini Rover and Phantom, allowing the application areas to expand to include new markets such as structure engineering, structure inspection, safety operations, fish swarm locating, oceanographic science and power plant survey. The small-size ROV manufactured in Japan, named Kaiko, was submerged at the deepest point at 10909 meters in the Mariana Trench. Today, it can be estimated that there are approximately over 100 system manufacturers around the world and 3000 ROVs of various sizes, capacities and capabilities, controlled by over 100 operators.

Although the historical development of unmanned underwater systems has been shaped around ROVs, after a certain point and within the framework of needs, the focus of developers and especially academic institutes shifted to eliminating the cable connection with the surface station by preserving features such as endurance, bandwidth and reliability. In parallel, the experimental Autonomous Underwater Vehicle (AUV) studies were initiated and carried out. The first operational AUV prototypes were developed at the Applied Physics Laboratory at the University of Washington and were tested under the designation Self-Propelled Underwater Research Vehicles (SPURVs). The successor to SPURV 1, the torpedo-shaped AUV, and similar form Unmanned Arctic Research Submersible (UARS) and SPURV 2 were launched and operated under ice packs. Simultaneously, other countries and companies continued to develop the technology and implement a variety of research tools from the mid to late 1970s. The US Navy also continued to develop the Unmanned Free Swimming Submersible (UFSS) vehicle in collaboration with the Naval Research Laboratory and the University of Hampshire.

The 1980s were an important decade for AUVs. The L'Epaulard AUV became operational from 1981, scanning an area of 805 kilometers for 5 years and was able to

send over 200,000 images to the surface. By 1984, there were 17 underdeveloped AUV projects, with the exception of the L'Epaulard and SPURV systems, which were battery operated and worked on following pre-programmed paths. In this period, system developments continued in three different areas of expertise as commercial, academic and military organizations. The challenging factors in advancing AUV systems were the need for higher-capacity energy sources for onboard power, lighter systems with higher efficiency, more computing capacity, and more precise navigation. One of the important operational systems produced in the 1980s is the MT '88 vehicle developed at the Institute of Marine Technology Problems (IMTP) in Russia and used for the survey of sunken submarines. One of the important operational systems produced in the 1980s is the MT '88 vehicle, which was developed at the Institute of Marine Technology Problems (IMTP) in Russia and used for the examination of submerged submarines in Norway and the Atlantic.

In the decade of the 1990s, implementation and development work was mostly led by academic research institutes. From a literature standpoint, at the beginning of this decade, more than 56 different AUVs appeared to have been published, but these systems were prototypes that were unable to fully complete field missions (Bellingham & Rajan, 2007; Wynn et al., 2014). Through the efforts of organizations such as Florida Atlantic University and MIT of the US, the Institute of Oceanographic Science Deacon Laboratory of the UK and the Marine Science and Technology (MAST) program of the European Consortium (Italy, Portugal, Belgium, Denmark), vehicles such as Ocean Explorer, Ocean Voyager, Sea Squirt, Odyssey, Doggie, Dolphin, Mini Autonomous Underwater Vehicle (MAUVE), Marius and Martin have been launched. To fully compile historical information on the emergence and development of UUVs, structurally dependent systems such as crawling-towed, towed, and hybrid should also be addressed.

AUVs are developed at the design stage to withstand various pressure values. Depending on these pressure values, they can navigate in waters up to 6000 m in depth without being subject to deformation. Scientific studies such as profiling, mapping and obtaining high-resolution three-dimensional images on the seafloor are possible when

AUVs with deep water resistance can travel in a vertical position close to the bottom. These AUVs can navigate with higher accuracy than surface boats and towed hulls, thanks to the equipment deployed on them such as sidescan sonar and camera systems (D. Jones et al., 2008; Scheirer et al., 2000; Wynn et al., 2014). As of 1995, ideas and designs on the glider form have multiplied. The fact that AUVs in the form of glider, of which highly developed types are used today, have the capacity to glide underwater by making use of surface control, allowing them to stay underwater for long periods of time, up to months. An important sub-type of AUVs, these buoyancy-driven underwater gliders make use of their fin-profile structures in order to have a stable cruise (C. Jones et al., 2007). These gliders, which later started to serve as deep water gliders, have a cruising range of 4600 miles and increased maneuvering capacity. Underwater gliders in the sector under the name SLOCUM follow the saw profile by making use of variable buoyancy in the vertical plane and can transfer information to the surface station (Simonetti & Webb, 1997). The first type of underwater glider mentioned, SLOCUM Electric navigates based on basic glider principles. It has a hydraulic equipment that regulates the lifting force. The buoyancy control technique in the second type, SLOCUM Thermal, is based on the thermocline effect. In this technique, a temperature change is created with ethylene glycol to give the glider buoyancy (Nidagundi, 2017).

The first marine geoscience studies were published in 1995-1996 with the information obtained from the Autonomous Benthic Explorer (ABE) AUV of the Woods Hole Oceanographic Institution located in the Juan de Fuca Ridge region in the USA (A. Tivey & Paul, 1998; M. A. Tivey et al., 1997; Yoerger et al., 1998). In 1999, the iRobot company developed an aluminum hull sea glider that generates lift by pump.

As the next decade, the 2000s, begins, it has become clear that AUVs are available for use for a wide variety of commercial missions. As the next decade, the 2000s, begins, it has become clear that AUVs are available for use for a wide variety of commercial missions. This is a period in which the predictions that the technology of autonomous underwater vehicles will be transferred to the maritime and ocean

industries in line with the needs, in addition to the presence of the technology in the academic-based study area, are strengthened. In 2001, two major AUVs were launched by Scandinavian region companies. One of them was developed by Maridan A/S with name in Denmark with the name of M600 and acquired by De Beers Marine to investigate the presence of minerals at various depths (110-150 m) in the South African sea. Another was the AUV developed by the Norwegian company Kongsberg A/S under the name Hugin 300. The AUV is used for the exploration of gas and oil resources in the Gulf of Mexico for the US-based ocean exploration firm C&C Technologies.

One of the technologies that have evolved in different directions this decade has been the collaborative control of multiple AUVs. In this context, the development of controller structures for joint action in cases where a group of AUVs has to work in a task that needs to be fulfilled has been provided. In addition to the problems of self-localization and navigation, which are the main issues in the control of AUVs, information sharing between vehicles, acoustic-based and light-based underwater communication and related problems such as small bandwidth, high latency and unreliability have been solved and applications have been made. (González-García et al., 2020; Paull et al., 2014).

In the same period, especially the Russian Institute of Marine Technology Problems started the SAUV (Solar Autonomous Underwater Vehicle) project, focusing on meeting the energy needs of AUVs with solar panels. AUSI (Autonomous Undersea System Institute), the organization that collaborated with within the scope of the project, developed their experience further and realized their own system as SAUV II in 2003. The battery pack was regularly charged with onboard solar panels, enabling the SAUV II to operate for up to a month. The SAUV II also had a charge-free operating life of 8 hours and was capable of GPS navigation with a high payload in the process. With the improvement of screw-driven AUVs, the REMUS-100 (Remote Environmental Monitoring Units) and REMUS-600 projects were carried out in 2001 and then 2003, respectively. These systems, which are designed for investigation, exploration and mining in shallow waters, can obtain environmental information with

a large number of sensors and measuring devices on them and send them to the main controller via acoustic communication. The controller can process this data and program the AUV according to the desired trajectory. What made the REMUS-100 unique was its small volume, light weight and compact structure. Developed on the infrastructure of the REMUS-100, the REMUS-600 has increased its size, weight, load carrying capacity and ability to operate in deep waters. In addition to all these, the features that make the REMUS-600 unique are its modular structure and universal adaptability to different tasks. It is possible to make the AUV ready for different missions by updating the hardware in a practical and rapid manner. The AUV, which can reach a range of up to 280 miles with the activation of additional batteries, has a high yaw motion balance with three bow and fodder rudders.

The aluminum body Seaglider, developed by iRobot, which was mentioned in the previous parts of the history, was later modified with a composite body structure made of carbon fiber material. Thanks to this, the diving depth reached 6000 m. This new version made its first dive under the name Deepglider in 2006.

By 2009, it is seen that AUV's were developed in 14 different countries around the world. Again, if the quantitative distribution of the AUV projects in the aforementioned year is analyzed according to the countries; USA comes first with 53.19%, followed by Canada, France and Japan with 8.51%, 7.45%, 5.32% and 4.26% respectively. It is seen that international projects have a significant share with a rate of 4.26% (Bocharov, 2009).

Towards the end of 2010, a new categorization for AUVs was mentioned. In this size-oriented categorization, AUVs weighing less than 100 kg are defined as mini-AUVs and ones up to 20 kg as micro-AUVs. With their small size and weight, these systems have found a place in a wide range of military, commercial and scientific fields. The ULISAR (National Autonomous Underwater Vehicle) project, which was financially supported by The Scientific and Technological Research Council of Turkey in 2006-2010, was completed by the founders of the Turkish company Desistek A.Ş. and became the first AUV realized in Turkey. Founded in 2011 at the Middle East

Technical University KOSGEB Technology Development Center, Desistek A.Ş. develops portable ROVs capable of surveying, monitoring, data acquisition, sample collection and scanning underwater in poor visibility areas. HOYTEK company, established in Ankara in 2015, put into service the expertise of Unmanned Marine Systems, Embedded Systems and Offshore Operations with the ROV 100, DEMRE, MYRA, LYRA and LYRA S ROVs it developed. HOYTEK ROV 100 received by ITU POLREC was used in Antarctica to conduct polar underwater research. In 2018, the company became the first user in the world by integrating Gemini720im multibeam sonar, produced by Unique Group, into DEMRE ROV. The light-duty class LYRA S, produced in 2019 for Dokuz Eylul University Marine Sciences and Technology Institute, can dive up to 1000 m and is used in observation, survey, underwater archeology, search-rescue and wreck removal missions with its two manipulator arms.

In 2020, within the scope of the TÜBİTAK Shipwreck Inventory Project, a warship, which is considered to belong to the 18th century, was discovered off the coast of İzmir Foça, in cooperation with the DEU Koca Piri Reis Research Ship and LYRA S. DEMRE, on the other hand, was equipped with cameras, led units and sensors to be used in observation and search and rescue missions of the maritime police in 2020-2021. Again in 2021, the AHRS, depth and temperature sensor package developed and produced by the company was exported to Canada. In 2019, another Turkish company, Maren Robotics, was established in Ankara and Antalya. SAGA (Unmanned Underwater Observation Vehicle), SAGAPRO (Unmanned Underwater Search&Rescue Vehicle) and DORIS (Sporting class unmanned underwater observation vehicle) developed by the company have been put into the service of both military and private institutions.

ITU Robotics AUV Team, which started working with ROV projects under Istanbul Technical University Naval Architecture and Marine Sciences Faculty in 2016, developed the first AUV prototype in 2018 with a team of 30 people. The designed Turquoise AUV is equipped with ROS operating system, Nvidia Jetson Xavier super-fast processing power, high capacity Li-ion battery, 8 BlueRobotics T100 thrusters and high resolution 4K-stereo cameras. The AUV performed tasks such as autonomous

navigation, image processing, acoustic localization and robotic manipulation for the ROBOSUB competition in 2021 (Yazici et al., 2021).

2.2 State of Art and Literature Review

In the last decades, technological developments in artificial intelligence, artificial neural networks, deep learning, control and automation have reflected on maritime operations. These developments aimed at eliminating the human factor during maritime operations can be divided into surface and underwater technologies. For underwater robots, it is referred to as unmanned underwater vehicle (UUV) in the literature.

The term UUV is a generic concept that includes ROVs and AUVs and is used to describe robotic systems that are separated from manned submarines. As such, their design and engineering are carried out with a similar approach with mobile robots and electric vehicles. When the usage areas of UUVs are examined, it is revealed that there is a very high usage rate in the scientific and military fields, energy and maritime sectors (K. D. Kaya & Gören, 2017).

Within the scope of scientific studies, activities such as collecting biological data, measuring the amount of existing particles, pollution, salinity and radioactivity of the underwater environment, scanning / mapping the sea floor, bathymetric surveys on subsea terrains and monitoring seismic activities can be carried out. Underwater robots with robotic manipulators and grippers can take samples from the sea floor.

Considering the energy field, underwater robots are used in the control, observation and inspection of energy transmission lines passing through the sea and ocean floor today. In addition, underwater robots are used to locate and extract energy resources such as oil and natural gas under the seabed. In addition, underwater robots take part in the observation and inspection of the underwater parts of the turbine towers in offshore wind power plants consisting of wind turbines installed on the sea. Separate from the marine or ocean environment, it is used in hydroelectric power plants, in the

detection of cracks in dam constructions and walls, and in operations to detect weak or strong areas of dam structures.

When military applications are investigated, it is seen that UUVs are used in operations related to detecting and eliminating external threat elements of military ships. Parallel to this, UUVs are also used in applications for detecting and neutralizing submarine mines.

From the point of view of the maritime sector, conventionally, two different methods are used for the inspection of the hull surfaces of ships under the waterline. For this, the necessary inspection, maintenance and repair operations are carried out either by dipping a diver or by taking the ship into the dry dock at the shipyard. There is a cost loss and occupational safety risk brought about by these methods. In order to eliminate cost loss and risks, underwater robots are used at a high rate, especially in ship hull inspection processes. In addition, classification societies like Lloyd's Register that provide classification, compliance and consultation services for ships set standards for AUVs and ROVs used in ship surveillance (Germanischer Lloyd, 2016). Also in the maritime industry, underwater robots are used to determine the situation and location of the damage on the ship's hull that occur as a result of collusion in environments with heavy sea traffic. In addition, ROVs and AUVs are used to locate and bring out shipwrecks due to their deep diving and object detection features. These features are used both in the control of net structures in fish farms and in monitoring fish shoals in the open sea. In today's conditions, design, production, control architectures, navigation, intelligent task assignment, embedded sensor systems and data filtering issues for the development of low deployment cost, high durability, high range, high maneuverability and high sensor capacity ROVs and AUVs. The number of scientific and R&D studies reported is increasing day by day.

In AUVs, since radio frequencies cannot be transmitted in water and the acoustic transmission does not provide a sufficient bandwidth for direct control, advanced autonomic functions must be relied upon. This requires running highly complex electronic hardware and software on AUVs.

When the task output of multiple unmanned vehicles (MUVs) functioning in collaboration is compared with that of each robot, it can be seen that the time duration required, system effectiveness and mission accomplishment rate have improved thanks to the task allocation in the MUV (Youmin Zhang & Mehrjerdi, 2013). The reflection of this approach on AUVs was manifested in the proposal of collaborative control methods of multiple AUVs (Multi-AUVs) that enable tasks such as adaptive formation control (Fiorelli et al., 2006). Especially in target recognition applications, Multi-AUVs with a collaborative sensing model using convolutional neural network and feature fusion can be successful (Cai et al., 2020).

The scientific literature surveyed and used as a source throughout the study is thematically categorized as “AUV Modelling and System Identification”, “AUV Hydrodynamics” and “AUV Control Architecture” and divided into sub-groups.

Underwater robots are referred to as unmanned underwater vehicles (UUVs) as a general concept in the literature. AUVs along with remotely operated vehicles (ROVs) are a subtype of UUVs. Unlike ROVs, AUVs can perform their programmed tasks autonomously without the need to be guided by an operator (Sahoo et al., 2020). Ensuring the stability of AUVs and designing their control systems requires the construction of a mathematical model containing kinematic and dynamic equations for 6 degrees of freedom (Fossen, 1994, 2002). Additionally, Thor I. Fossen (2011), provided nonlinear and linearized maneuvering models and state-space transformations for AUVs and ROVs. In other respects, empirical modeling studies have also been carried out on this subject. Aras et al. (2013), developed a thruster model by system identification in a numeric computing environment. Using onboard sensors Natarajan et al. (2012) identified the dynamic model with unknown physical parameters on surge, sway and yaw motions. Eng et al. (2016), developed an online identification method to obtain AUV dynamics by experiments, where the AUV is guided to execute a set of maneuvers. Rustrian (2016), conducted an open-loop testing using an observation class mini underwater vehicle to develop a dynamic simulation model.

The identification of underwater vehicle hydrodynamic characteristics is essential for developing a compliant actuation system to perform a high precision maneuvering motion. Developing appropriate motion control strategies for underwater robots that have inherent complex dynamics requires clearly revealing their hydrodynamic characteristics. For the purpose of developing guidance and motion control strategies for AUVs, (Jagadeesh et al., 2009a) performed an experimental study for the axisymmetric underwater body which generates the results of hydrodynamic forces and moments at varying flow speeds and pitch angles.

Recently, Computational Fluid Dynamics (CFD) tools are used to calculate the hydrodynamic parameters of bodies moving underwater with models created by FEM (Finite Element Method) or FVM (Finite Volume Method) (Du et al., 2014; Wu et al., 2014). When identifying the flow field and pressure distribution to find out the hydrodynamic parameters, resulting forces and moments, applying CFD provides an advantage of reducing time and cost in comparison with other approaches such as tank tests with constrained models and tests with free-running models (De Barros & Dantas, 2012). Javanmard (2013); Mansoorzadeh & Javanmard, (2014) have investigated the free surface effect on drag and lift coefficients of AUV at varying depth values using k-ε and shear stress transport (SST) model and linked the hydrodynamic coefficients to the submergence depth and AUV speed. Salari & Rava (2017) claimed that the hydrodynamic behavior of the AUV when moving near the water surface is different from that when moving at higher depths. Besides that, the authors numerically studied the hydrodynamic coefficients over AUV near the water surface with various turbulence models. Congruently, Amiri, Sphaier, Vitola, & Esperança (2019) studied to estimate the function of the leeward vortical flow structure in the hydrodynamic characteristics of an underwater vehicle which is positioned shallow levels at a moderate drift angle.

Li, Pan, Zhao, Ma, & Wang (2018) introduced a design of an Autonomous Underwater Glider (AUG) equipped with hydro-wings generating wave motion-induced thrust and the hydrodynamic behavior of AUG was investigated numerically by solving incompressible viscous Navier-Stokes equations. da Silva Costa, Almeida,

de Melo Filho, da Cunha Lima, & da Cunha Lima (2019) analyzed the torque effects of hull-wake interaction on the maneuverability of an AUV which is flatfish-shaped. Mitra, Panda, & Warrior (2019) experimentally and numerically studied the effect of free-stream turbulence on the hydrodynamics of AUV hull form for different depths. Moreover, Mitra et al, (2020) investigated hydrodynamic characteristics for various Reynold numbers over sloped sea beds. Bao-Ji, Xin-Di, & Wen-Xuan (2019) investigated the resistance performance and movement stability of open-frame remotely operated vehicle with the propeller in motion using the steady-state Reynolds-averaged Navier-Stokes method for simulating the forces.

On another study of open-frame AUV hydrodynamics performed by Tang, Ura, Nakatani, Thornton, & Jiang (2009) the importance level of each hydrodynamic coefficient was investigated by analyzing the relationship between hydrodynamic loads and motion variables. Besides all these numerical studies, the proposed CFD method should be validated with existing experimental data Tian, Song, & Ding (2019).

Together with hydrodynamic analysis, in the evaluation of performance parameters of AUV, the usage of data derived from CFD results constitutes an important point (Evans & Nahon, 2004). Besides that, hydrodynamic analysis results may provide an additional contribution to the studies which investigate power consumption and energy efficiency of AUVs (Carneiro et al., 2019; Sarkar et al., 2016; Yang et al., 2019; Zavari et al., 2019).

At the end of the literature review, a comprehensive information will be provided about the requirements adherence to trajectory, controller structures and control performances in AUVs. Looking at the early periods, it is seen that the first controller used was PID (Hammad et al., 2018; Ye Li, Jiang, et al., 2015; Ye Li, Wei, et al., 2015). This control system, which has a very simple structure, gave insufficient performance when dealing with the change of target coordinates. However, setting the parameters of the PID controller was complicated and could not give the desired result.

For this reason, using it in an integrated manner with other control methods improves system response.

Sliding Mode Control (SMC), which works very efficiently under external perturbation and time-varying effects, has been applied in AUVs as another control approach (Cheng et al., 2007; Londhe et al., 2017). SMC has also been used in conjunction with other control methods as specified for PID (Jiang et al., 2018; T. Li et al., 2018). However, in the SMC method, the chattering effect caused by the natural structure of the switching function must be eliminated by certain methods.

Another method that works with high control accuracy is Backstepping control. It is frequently preferred in trajectory tracking of AUVs due to its performance in nonlinear systems (Xu et al., 2014). However, in higher order systems the computational load will inevitably become more difficult.

Adaptive algorithms can be run in harmony with various methods and can resist parameter perturbation and disturbance in the system. Therefore, it is frequently used in the design of path following controllers (Antonelli et al., 2001; Cui et al., 2017; Khodayari & Balochian, 2015).

Model Predictive Controller (MPC), which is an optimal controller, is effective in nonlinear systems as well as in linear systems. With this method, when designing the motion controller of AUVs, state constraints and control constraints are defined in the optimization function (Gan et al., 2020; J. H. Zhang & Yao, 2003). Yongding Zhang, Liu, Luo, & Yang (2019) proposed a 3D path following controller design utilizing MPC. Yan, Gong, Zhang, & Wu (2020) developed a trajectory tracking system with two feedback loops. In this design, the input of the dynamic controller in the inner loop, which is the output velocity, comes from the outer loop that contains the kinematic controller. Thus, this dual feedback controller controls torque and thrust. Rashidi, Karimi, & Khodaparast (2020) presented an MPC approach that runs the Laguerre function inside, which greatly relieves the computational load. Another optimal controller, the linear quadratic regulator (LQR), was applied by Tchilian,

Rafikova, Gafurov, & Rafikov (2017) to an autonomous underwater glider. In this approach, the strategy of linearizing the non-linear model is applied so that the underwater glider can be controlled to follow a reference.

2.3 AUV Modelling

The mathematical model is created using various physical laws to describe how the AUV is affected by effects such as gravity, friction, internal / external forces, and internal / external moments. To design of feedback control structures for ships, high-speed crafts, floating structures and underwater vehicles, reduced-order models (decoupling the motions according to 1 DOF, 3 DOF, 4 DOF, 6 DOF) are often used since most marine vehicle do not have actuation in all DOF (Fossen, 2011). Before describing an underwater vehicle in 6 DOF, surge, sway, heave, roll, pitch, and yaw motions of marine vehicles should be specified referring to Society of Naval Architects and Marine Engineers as given in Table 2.1 (SNAME, 1950).

Table 2.1 6-DOF states of AUV

DOF	**Motion**	**Force and Moment**	**Velocity and Angular Rate**	**Position and Euler Angle**
1	Linear motion along the x-axis	X	u	x
2	Linear motion along the y-axis	Y	v	y
3	Linear motion along the z-axis	Z	w	z
4	Rotational motion around the x-axis	K	p	ϕ
5	Rotational motion around the y-axis	M	q	θ
6	Rotational motion around the z-axis	N	r	ψ

2.3.1 AUV Kinematics

It is important to assign and locate appropriate coordinate systems in defining the dynamic behavior of the AUV. The first of the two different coordinate systems deployed is the body-fixed coordinate system and acts in integration with the AUV. In

this system, subject to the right-hand rule, the x-axis is placed forward along the length of the AUV. The y-axis is directed from the center of origin to starboard direction. The z-axis is located in a downward direction from the AUV body. The body-fixed frame on the AUV makes it easy to define sensor data including position, speed and orientation information. The body-fixed frame on the AUV makes it easy to identify the sensor data moving with the AUV and the forces and moments resulting from the thrusters. On the other side, position and orientation data are described according to the earth-fixed frame (Caccia, 2007). The placement of body-fixed frame and earth-fixed frame on the AUV is given in Figure 2.1.

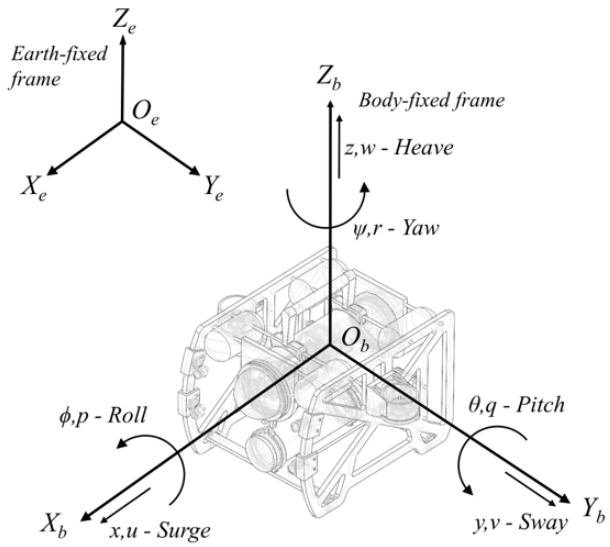


Figure 2.1 Motion variables of AUV

The vectors that express the motion through six axes are indicated in Equations 2.1 to 2.6.

$$\eta_1 = [x \ y \ z]^T \tag{2.1}$$

$$\eta_2 = [\phi \ \theta \ \psi]^T \tag{2.2}$$

$$\nu_1 = [u \ v \ w]^T \tag{2.3}$$

$$\nu_2 = [p \ q \ r]^T \tag{2.4}$$

$$\tau_1 = [X \ Y \ Z]^T \tag{2.5}$$

$$\tau_2 = [K \ M \ N]^T \tag{2.6}$$

Here, vector η is position and orientation coordinate of AUV with respect to earth fixed coordinate system, whereas vector v is the velocity vector, and each τ respectively represents external force vector and external moment vector that acts on AUV body. To obtain the kinematic expression of AUV, the formulation must be transformed from body fixed coordinate system to earth fixed coordinate system. In particular, Euler angles (ϕ,θ,ψ) are significant due to their function of describing the orientation and defining the control parameters. Linear velocity transformation with respect to earth fixed frame is given in Equation 2.7 (Ramesh et al., 2013).

$$\dot{\eta}_1 = J_1(\eta_2)\nu_1 \tag{2.7}$$

Here $J_1(\eta_2)$, which is the transformation matrix related with Euler angles is written in Equation 2.8.

$$J_1(\eta_2) = \begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi s\phi + c\psi s\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta c\phi & c\theta c\phi \end{bmatrix} \tag{2.8}$$

Angular velocity vector with respect to the body fixed frame is associated with Euler vector through $J_2(\eta_2)$ transformation matrix as written in Equation 2.9.

$$\dot{\eta}_2 = J_2(\eta_2)\nu_2 \tag{2.9}$$

$J_2(\eta_2)$ transformation matrix can be written in terms of Euler angles as in Equation 2.10.

$$J_2(\eta_2) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi / c\theta & c\phi / c\theta \end{bmatrix} \tag{2.10}$$

2.3.2 AUV Dynamics

The 6 Degree of Freedom model of dynamics for AUVs in body fixed frame can be written as in Equation 2.11 to 2.14 (K. D. Kaya & Gören, 2017).

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau(\nu, u) \tag{2.11}$$

$$M = M_{RB} + M_A \tag{2.12}$$

$$C(\nu) = C_{RB}(\nu) + C_A(\nu) \tag{2.13}$$

$$D(\nu) = D + D_n(\nu) \tag{2.14}$$

Here, M denotes the mass matrix, which is the 6×6 inertia matrix containing M_{RB} which is the generalized constant inertia matrix and M_A which is the added mass inertia matrix. $C(\nu)$ is the 6×6 Coriolis and centripetal forces matrix including added mass. Nonlinear and linear hydrodynamic damping terms are located inside the 6×6 matrix $D(\nu)$, D includes the linear damping expressions, and $D_n(\nu)$ comprises the nonlinear damping expressions. $g(\eta)$ is a 6×1 vector and denotes buoyancy and gravitational effects. $\tau(\nu,u)$ denotes the resultant force and moment vector exerting on the AUV, the control inputs are denoted by u which includes the angular velocity of the motors due to the generated thrusts, the heading angle data transmitted to the steering actuators.

2.3.3 AUV Hydrodynamics

First of all, the details of the numerical model used in common for hydrodynamic analysis studies are given below.

Conservation equations include continuity, momentum and energy expressions for certain boundary conditions of three-dimensional, time-dependent, incompressible fluids. In underwater flow problems, the continuity equation, which ensures that the amount of material entering the flow volume is equal to the amount of material at the outlet, is written as in Equation 2.15 (Kim & Park, 2017).

$$\frac{\partial \rho}{\partial t} + \nabla(\rho u) = 0 \tag{2.15}$$

Here ρ and u represent the flow density and velocity vector field, respectively. Considering Newton's Second Law, the momentum relations for Naiver Stokes or the control volume can be written as in Equation 2.16 (Tu et al., 2013; Versteeg & Malalasekera, 2007).

$$\rho \frac{Du}{Dt} = f - \nabla p_f + \mu \nabla^2 u \tag{2.16}$$

In the above equations, p_f, μ and f represent fluid pressure, fluid viscosity and body force respectively.

Different turbulence models are used in numerical analysis studies depending on flow type, boundary conditions and physical properties. The generally preferred models for turbulent flow conditions are Reynolds averaged Navier-Stokes (RANS) Equations based on k-ε standard, k-ε realizable and k-ε RNG. In the study, the k-ε standard model is used as the turbulence model. The main differences between the mentioned models are the sets of equations used to define the turbulent viscosity calculation method and turbulent diffusion terms such as k and ε (ANSYS Fluent, 2015).

The two-equation k-ε standard turbulence model allows both turbulence length and time scale functions to be solved with two separate transport equations. The k-ε standard turbulence model has significant advantages in terms of time accuracy for flow calculations in engineering problems compared to the other models. The model

is also a semi-empirical method derived from exact equations and using physical reasoning. The *k-ε* standard turbulence model is expressed as in Equations 2.17 and 2.18 (Farhadi et al., 2018; Shirzadi et al., 2017).

$$\frac{\partial}{\partial t}(\rho k)+\frac{\partial}{\partial x_i}(\rho k u_i)=\frac{\partial}{\partial x_j}\left[\left(\mu\frac{\mu_t}{\mu_k}\right)\frac{\partial k}{\partial x_j}\right]+G_k+G_b-\rho\varepsilon-Y_m \tag{2.17}$$

$$\frac{\partial}{\partial t}(\rho \varepsilon)+\frac{\partial}{\partial x_i}(\rho \varepsilon u_i)=\frac{\partial}{\partial x_j}\left[\left(\mu\frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right]+C_{1\varepsilon}\frac{\varepsilon}{k}(G_k+C_{3\varepsilon}G_b)-C_{2\varepsilon}\rho\frac{\varepsilon^2}{k} \tag{2.18}$$

G_k, G_b and Y_m refer to turbulent kinetic energy, buoyant turbulent kinetic energy and dissipation rate. $C_{1\varepsilon}$, $C_{2\varepsilon}$ and $C_{3\varepsilon}$ are the constant coefficients and their values are taken as 1.44, 1.92 and 0.09, respectively. σ_k and σ_ε are the turbulence Prandtl numbers for k and ε, and their values are taken as 1 and 1.3, respectively. The k-ε model involves coefficients C_μ, $C_{1\varepsilon}$, $C_{1\varepsilon}$, σ_k, and σ_ε. The nature of these coefficients leads to ambiguity regarding their values. These coefficients have been calibrated to simulate the flows over the domain(Ngo & Lim, 2020). μ_t, turbulence viscosity is expressed as Equation 2.19.

$$\mu_t=\rho C_\mu\frac{k^2}{\varepsilon} \tag{2.19}$$

2.4 AUV Motion Control

In the control applications explained in detail in Chapter 5 of this thesis, various controller structures have been established and run in the simulation environment. Three different control scenarios PID, Linear Quadratic Regulator (LQR) and LQR + Sliding Mode Control (SMC) decoupled structures were modeled and tested separately. In this section, the basic theoretical infrastructures of PID, LQR and SMC controllers, which have been applied in practice within the scope of the thesis, are mentioned.

2.4.1 PID Controller Design

Although it is not suitable for a linear controller due to the nonlinear nature of underwater robots, PID control is also included in order to compare performance with Linear Quadratic Regulator (LQR) and Sliding Mode Control (SMC) applications.

PID control rule is a continuous process that utilizes proportional, integral and derivative (P, I and D) control parameters. In other words, as long as there is an error in the system, the control command is also generated. PID control is the most basic controlling method and implements adequate, robust and stable control in many industrial application areas. On the other hand, besides this simplicity and usefulness, there are also weaknesses. First of all, PID control cannot be applied to every complex system today. PID control can be easily applied to single-loop systems with a mostly linear and simple structure. On the other hand, it is difficult to implement in non-linear systems, especially in systems where the dead time delay is very high compared to the system time delay. Also, it is almost impossible to stabilize it by means of PID control, especially in the presence of nonlinearities due to frictional resistance (Yüksel, 2014).

Depending on the dynamic nature of the system to be controlled, the simplest possible combination of the three basic control effects involved in PID control is used. These combinations can be in the form of P, PI, PD and PID. The block diagram of the standard PID control structure is as given in Figure 2.2.

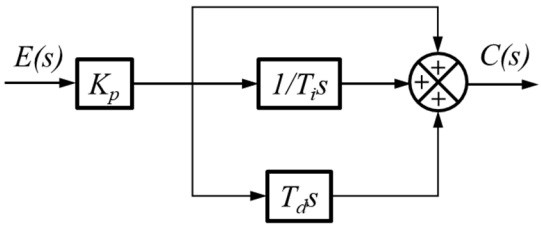


Figure 2.2 Standard PID control

The control signal output *c(t)* depending on the standard PID control law can be expressed as the sum of three basic control effects and the transfer function can be written as in Equation 2.20.

$$\frac{C(s)}{E(s)} = K_P\left(1+\frac{1}{T_i s}+T_d s\right) \tag{2.20}$$

As it can be seen from the Equation 2.20, it generates a signal which provides the advantages of several controlling effects in one system. While the integral term eliminates the steady state error occurring in the system, the derivative effect speeds up the response for the same relative stability of the system compared to utilization of the PI controller. Obviously, using the PID provides a faster response behavior with zero error in the steady state system. The response curves of the closed loop systems operating with P, PI, PD and PID control, shown by the error signal in response to the step input, are given in Figure 2.3.

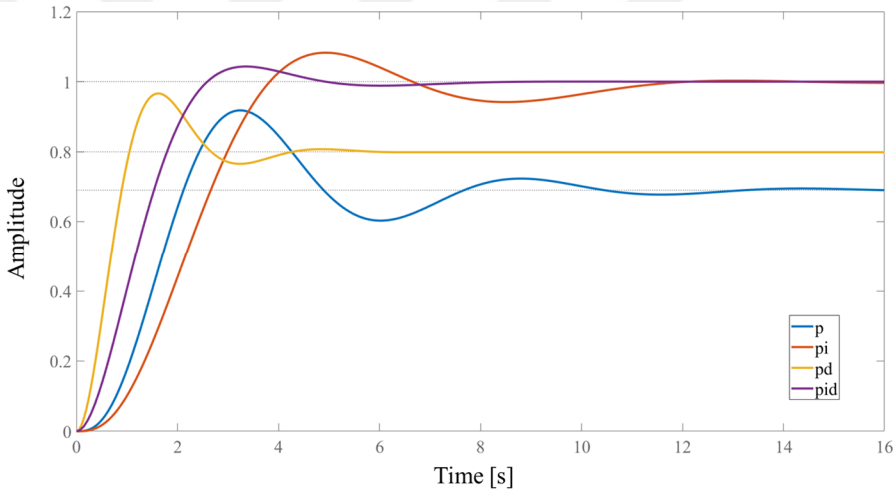


Figure 2.3 Comparison of P, PI, PD and PID control responses to a step input

It can be seen from Figure 2.3 that only in the controlled state P, the response curve deviates from the point where it should be in the steady-state state after a few oscillations. In the case of the PI control, a response curve is similarly obtained, but no deviation occurs in the steady-state condition. However, the addition of integral effect to the proportional effect increases the susceptibility of the system to instability. To compensate for this situation, the proportional gain should be chosen lower than the proportional effect control alone. However, in this case, the correction of the error will not be as fast as in the proportional effect only, and the initial error will occur at higher values. In PD control, as soon as the error occurs, both the proportional effect

and the derivative effect reduce the error, so the first peak value of the error remains smaller. After the resulting maximum value, the effect still tries to reduce the error. However, the rate of change acts in the opposite direction, trying to prevent change. For this reason, derivative control creates a stabilizing effect on the system and provides the opportunity to use a stronger proportional effect gain. As it can be seen from the curve, less initial deviation and less steady state error occur with PD control than with P control. In PID control, on the other hand, although a large initial error occurs, it is seen that the steady-state error goes to zero over time.

Adjusting the PID control means optimally determining the proportional, integral and derivative parameters K_P, T_i and T_d. The optimal setting of the proportional gain K_P is sufficient in the controller that only works with proportional effect. On the other hand, in dual parameter controllers such as PI or PD, the optimal setting of both parameters is required. In PID control, the most appropriate setting of the three parameters should be determined. Here, what is meant by the most appropriate setting is the feedback system with minimum or zero error, eliminating the error in a short time and the most stable operation. In other words, it should be aimed to eliminate the minimum error in minimum time and with minimum oscillation. There are two different ways of setting control parameters, experimental and analytical. If a step function is given to the input of the closed-loop control system during the experimental setting, the parameters that allow the system response to reach the new steady-state value in the shortest time and in a stable manner are accepted as the optimum setting.

For this purpose, some criteria have been developed by taking the vibration response amplitudes of the system as a criterion. The most widely used of these was developed by Ziegler Nichols. Oscillation/Vibration Method and System Response method are applied in order to make fine adjustments within the scope of experimental tuning. The Oscillation/Vibration Method is based on testing the system in a closed loop continuous oscillation. In System Response method, when the system is in open loop, step input $s(t)=s_0$ is recorded which causes steady state $c(t)=c_0$. Then its intensity is adjusted towards s_∞ and c_∞ in this case is determined. The terms of open-loop gain K_0, change in output signal Δc, change in input signal Δs, dead time constant t_d and

time constant τ_t depending on these terms are used to determine the parameters of the PID controller.

The analytical tuning method is based on calculating the optimum values of the parameters K_P, T_i and T_d, if the dynamic behavior of the plant and the sensing element is known. The complex optimization methods used in these calculations can be applied in computerized PID tuning programs. In the 5.1.1 part of this thesis, the Transfer Function Based PID Tuning method in the MATLAB/Simulink PID block is used. For this reason, the theoretical background of the Transfer Function Based PID Tuning method will be presented (Haugen, 2004).

The basic block diagram of the PID control system is as shown in Figure 2.4.

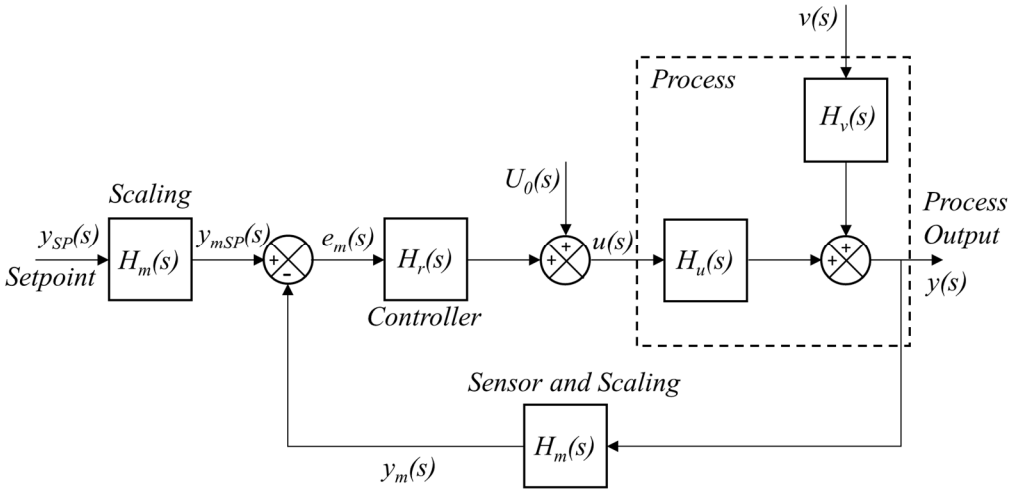


Figure 2.4 Principal block diagram of control system

If the sensor and scaling blocks are integrated into the process blocks, the combined process transfer functions can be written as in Equations 2.21 and 2.22.

$$\frac{y_m(s)}{u(s)} = H_u(s) \cdot H_m(s) = H_p(s) \tag{2.21}$$

$$\frac{y_m(s)}{v(s)} = H_v(s) \cdot H_m(s) = H_{vm}(s) \tag{2.22}$$

The equivalent of the transfer function based block diagram is given in Figure 2.5.

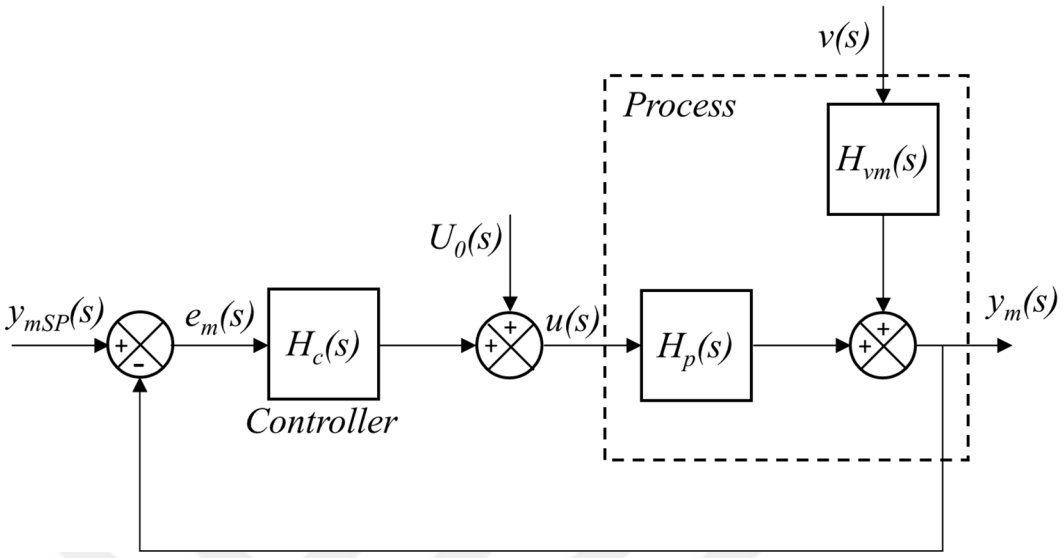


Figure 2.5 Equivalent transfer function based block diagram

As it can be seen in the block diagram above, the parameter to be controlled is the measurement of the process. Dominant disturbance, expressed as input disturbance in most systems, affects the process dynamically in a similar way to the controller signal. In such systems, the dynamic properties of $H_p(s)$ and $H_{vm}(s)$ are similar. For example, if the time constants T and T_{vm} are equal, the transfer functions will have the same dynamics as in Equations 2.23 and 2.24.

$$H_p(s) = \frac{K}{Ts+1} \tag{2.23}$$

$$H_{vm}(s) = \frac{K_{vm}}{T_{vm}s+1} = \frac{K_{vm}}{Ts+1} \tag{2.24}$$

The error transfer function of the control system given in Figure 2.5 can be written as in Equation 2.25.

$$e_m(s) = \underbrace{S(s)y_{mSP}(s)}_{e_{ymSP}(s)} - \underbrace{S(s)H_{vm}(s)v(s)}_{e_{vm}(s)} \tag{2.25}$$

Here *S(s)* denotes the sensitivity function which is given in Equation 2.26.

$$S(s) = \frac{1}{1 + L(s)} \tag{2.26}$$

Where the loop transfer function is denoted by *L(s)*. The loop transfer function can be written as in Equation 2.27.

$$L(s) = H_c(s) H_p(s) \tag{2.27}$$

The tracking transfer function is given as in Equation 2.28.

$$T(s) = \frac{1}{1 + L(s)} \tag{2.28}$$

2.4.2 Linear Quadratic Regulator Design

Linear Quadratic Regulator (LQR) is an optimal control algorithm used for the control of linear dynamic systems. In order to apply the LQR controller, the system dynamics must be defined as a set of linear differential equations in state-space form, as given in Equation 2.29 (Kirk, 2004).

$$\dot{x} = Ax + Bu \qquad x \in R^n,\ u \in R^n \tag{2.29}$$

Here *x* represents the state vector and *u* is the control input. System dynamics are defined with *A and B* matrices. The LQR algorithm is based on minimizing the cost function, which is a quadratic function of states and inputs. The cost function is written as in Equation 2.30.

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \tag{2.30}$$

Where Q and R denote symmetric positive semi-definite matrices. Q weights the states and R weights the inputs of the dynamic system. x^T and u^T are transposes of x and u. The performance of the feedback control system is measured by the cost function. Here the goal is to calculate the u's that minimize the cost function. The Riccati equation is used to determine the optimal control strategy in the LQR controller. This equation, which consists of nonlinear matrices and gives the relationship between cost function and control strategy, is written as in Equation 2.31.

$$-\dot{P} = A^T P + PA - PBR^{-1}B^T P + Q \tag{2.31}$$

Where A is the state matrix and B is the input matrix. From this equation, the P solution can be reached and the K gain matrix can be obtained if it is selected as $K=R^{-1}B^T P$.

When designing an LQR controller, the optimization problem involves selecting the weighting matrices Q and R, which determine the trade-off between minimizing the control effort (energy spent) and minimizing the deviation from the reference trajectory over time. If $Q = R$ is selected, it means that the importance of minimizing the control effort and the deviation from the reference trajectory are equal. If $Q > R$, it is more important for the system to go to the reference as fast as possible. In other words, the high amount of energy to be spent in order to get a quick response from the system is ignored. In the case of $Q < R$, it is desired that the amount of energy consumed is low. The status of the control signal changes according to the R parameter. The higher the R parameter is selected, the lower the control signal to be generated will be. The choice of R provides practicality in adjusting the control signal amplitude. The Q matrix contains the weight coefficient of each of the system state variables. The higher the weight coefficient of a state variable, the more energy is required to control that variable. In other words, more weight is given to the control of the variable with a higher weight coefficient.

2.4.3 Sliding Mode Controller Design

One of the methods used in the control of systems with uncertain and rapidly changing dynamics is Sliding Mode Control (SMC). In the SMC, which is a nonlinear controller, a control criterion is configured that forces the system to slide along a surface in the state space defined as the sliding surface (Christopher & Spurgeon, 1998). The sliding surface can be written as in Equation 2.32.

$$s(x) = h(x) - r(t) = 0 \tag{2.32}$$

Here, x denotes the state of the system. Scalar function of the state space $h(x)$ represents the desired response of the plant. The reference signal is expressed by r(t). The control criterion sets $s(x)$ to zero so that the system can capture and follow the reference signal. The control rule of SMC is written as in Equation 2.33.

$$u(t) = K\,\mathrm{sgn}(s(x)) \tag{2.33}$$

Here K denotes a constant gain. The sign function $sgn(s(x))$ can be +1 or -1 depending on whether $s(x)$ is positive or negative. Thus, the Sliding Mode Controller can be written as in Equation 2.34.

$$\begin{aligned} \dot{x} &= f(x,u) \\ \dot{s} &= h(x) - r(t) \\ u &= K\,sgn(s) \end{aligned} \tag{2.34}$$

The prominent strength of SMC is its robustness in the face of disturbance inputs and uncertainties. SMC whose theoretical details are given above, is that it behaves highly robust in the face of disturbing inputs and uncertainties. It does this by making the system slide along the sliding surface, free from uncertainties and disturbances. With these features, it is preferred in the design of trajectory tracking controllers of mobile robots.

2.4.4 Lyapunov Controller Rule

Since the stability evaluations of the controller structures to be implemented in Chapter 5 are based on Lyapunov's Rule, under this subheading, the theoretical background on the Lyapunov Stability Theorem, the application of this theorem to Linear Time Invariant (LTI) systems and the design of nonlinear controllers by using this method has been given (Isidori, 1995).

If a system like Equation 2.35 is considered.

$$\dot{x} = f(x); \quad x(t_0) = t_0 \tag{2.35}$$

Here x and $\dot{x}$ denote the state variable and the derivative of the state variable, respectively. The system is asymptotically stable according to the Lyapunov criterion if the time derivative of a positive $V(x)$ function can be calculated as negative $V(x)$. That is, for every x_0 initial condition $t \rightarrow \infty$ becomes $x(t) \rightarrow 0$. However, if $\dot{V}(x)$ is definite, it can be said that the system is unstable. In the application of the Lyapunov control rule to LTI systems, a system expressed as in Equation 2.36 is considered first.

$$\dot{x} = A_c x \tag{2.36}$$

Equation 2.37 is chosen as the Lyapunov function.

$$V = x^T P x \tag{2.37}$$

And the derivative expression, which must be negative for stability, is written as in Equation 2.38.

$$\dot{V} = \dot{x}^T P x + x^T P \dot{x} = x^T A_c^T P x + x^T P A_c x \tag{2.38}$$

Equation (2.38) is arranged and brought to the form in Equation 2.39.

$$\dot{V} = x^T (A_c^T P + PA_c)x \tag{2.39}$$

Let Equation 2.40 give a certain matrix Q known to be symmetrically definite positive.

$$-Q = A_c^T P + PA_c \tag{2.40}$$

The system is stable if P, which is solved by the Q matrix, is symmetrical definite positive. If it is desired to design a nonlinear controller using the Lyapunov method, the system input is not zero. When examining the derivative of any definite positive Lyapunov function $V(x)$ with respect to time, the u term will also appear due to the resulting $\dot{x}$ terms. A stable control is obtained if u can be assigned so that $V(x)$ is definite negative. For this, an error dynamic is written as in Equation 2.41.

$$\dot{e} = Ae + f(x, x_m) + B(r - u) \tag{2.41}$$

Here the error is $e=x_m-x$ and x_m does not have to be constant, it can come from a reference model or be a given function. Equation 2.42 is written because the error is desired to go to zero.

$$u = r + Ke + u_f \tag{2.42}$$

Here, the gain matrix K is assigned by adjusting the eigenvalues of the $A_c=A-BK$ matrix as desired and stable, as in linear systems. Thus, P solved from the symmetric equation with a certain matrix Q, which is known to be symmetric definite positive, will be symmetrically definite positive. This time the Lyapunov function is chosen as in Equation 2.43.

$$V = e^T Pe \tag{2.43}$$

The derivative of the Lyapunov expression is written as in Equation 2.44 and arranged as in Equation 2.45.

$$\dot{V} = \dot{e}^T Pe + e^T P\dot{e} = e^T A_c^T Pe + x^T PA_c e + 2e^T Pf(x, x_m) - 2e^T PBu_f \quad (2.44)$$

$$\dot{V} = -e^T Qe + 2e^T Pf(x, x_m) - 2e^T PBu_f \quad (2.45)$$

The derivative of V is set to zero as in Equation 2.46.

$$e^T Qe + 2e^T Pf(x, x_m) - e^T PBu_f = 0 \quad (2.46)$$

Assignment of a suitable u_f provides stable control with the input as given in Equation 2.47.

$$u = r + Ke + u_f \quad (2.47)$$

CHAPTER THREE
DEMARINE AUV INFRASTRUCTURE

In this part of the thesis, details about the design and manufacturing phases of the first two prototypes of DEMarine AUV will be provided. First of all, the underwater missions, which are the basis for the clarification of the design criteria, were determined and the hull type, size, thruster arrangement and critical centers (geometric center, center of gravity and swimming center) were determined. In general, it was decided that the DEMarine AUV should have a functional structure capable of performing the observation and inspection tasks of ships and underwater structures by following a reference underwater. In this sense, the development of DEMarine AUV's infrastructure will be explained as mechanical design, thrust and propulsion, system and electronic hardware.

3.1 Mechanical Design

The designed mechanical parts are handled in three groups as carrier frame and rods, closed enclosures and sealing components. In addition, floaters and lead ballast weights are used to fine-tune the buoyancy and immersion degrees. This equipment, which is used for the underwater robot to hover in the desired vertical position, is also a part of the mechanical system. All mechanical and electronic equipment of the underwater robot is positioned on the carrier frames. Embedded systems and battery are housed in watertight cylinders. Port and starboard side frames are designed and sized to fit over all remaining equipment.

3.1.1 Frame Construction

Port and starboard frames were manufactured by Computer Numerical Control (CNC) laser cutting method using Polymethyl Methacrylate (PMMA) material after the necessary unloading for weight optimization during the design phase. In the same way, undercarriage was manufactured using PMMA material with CNC laser cutting

method. Three-dimensional view of Mechanical structure components of the DEMarine AUV prototype is presented in Figure 3.1

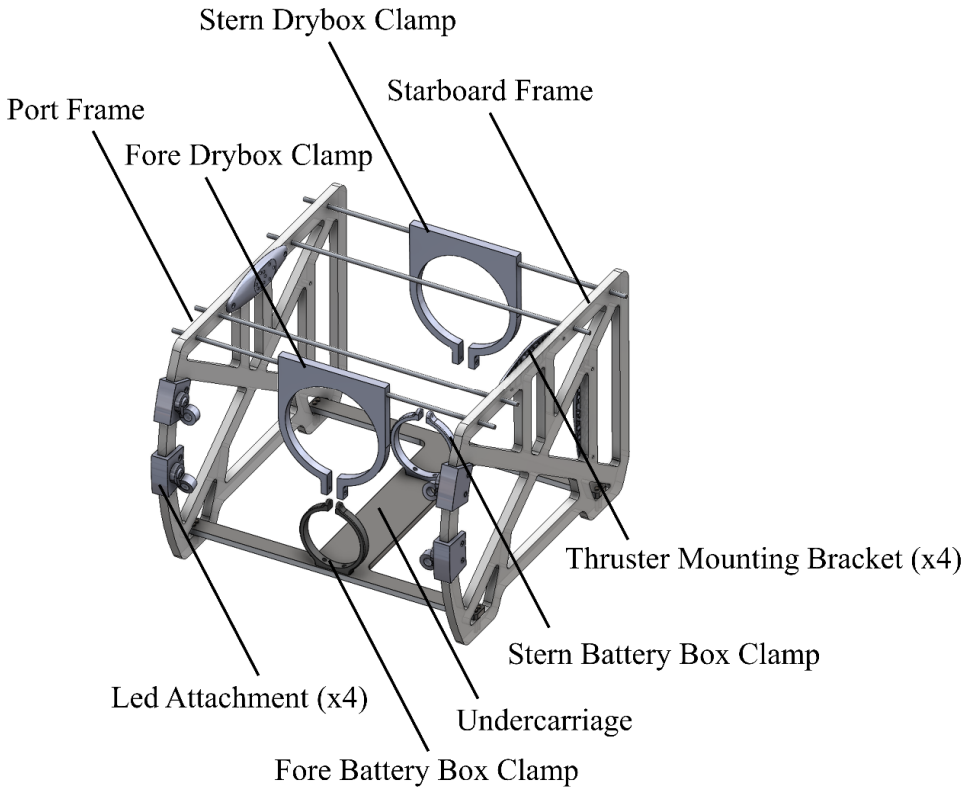


Figure 3.1 Mechanical structure components of the DEMarine AUV prototype

Fore drybox clamp, aft drybox clamp, fore battery box clamp and stern battery box clamp were produced from Polylactic Acid - Biopolymer (PLA) material using a 3D prototype production device. Rods carrying drybox clamps, floaters and thruster cases are made of stainless steel and attached to the frames with a bolt-nut connection. The four led attachments on the front of the AUV are also made of PLA material with a 3D prototype production device.

3.1.2 Closed Enclosures and Sealing

Watertight spaces are needed to contain the battery and electronic equipment protected from contact with water. While the initial prototype was being manufactured,

these enclosures, called drybox and battery box, were produced entirely with laboratory facilities. Acrylic tubes from the parts constituting the assembly were supplied as 100 mm for the drybox and 70 mm for the battery box and were cut into suitable lengths. O-ring sealing flanges are produced from aluminum material on a lathe. O-rings made of Buna-N (nitrile rubber) material were supplied in sizes suitable for the channels opening on the flanges. Acrylic endcaps were manufactured on a laser cutting machine and drilled using a vertical drill to attach cable penetrators. Cable penetrators are manufactured and threaded from aluminum cylinders on a lathe. Acrylic dome was procured from the market in suitable sizes. Stainless steel material and corrosion resistant socket head cap screws were used for flange-endcap and flange-dome connections. The exploded three-dimensional view of the drybox and the battery box is as seen in Figure 3.2.

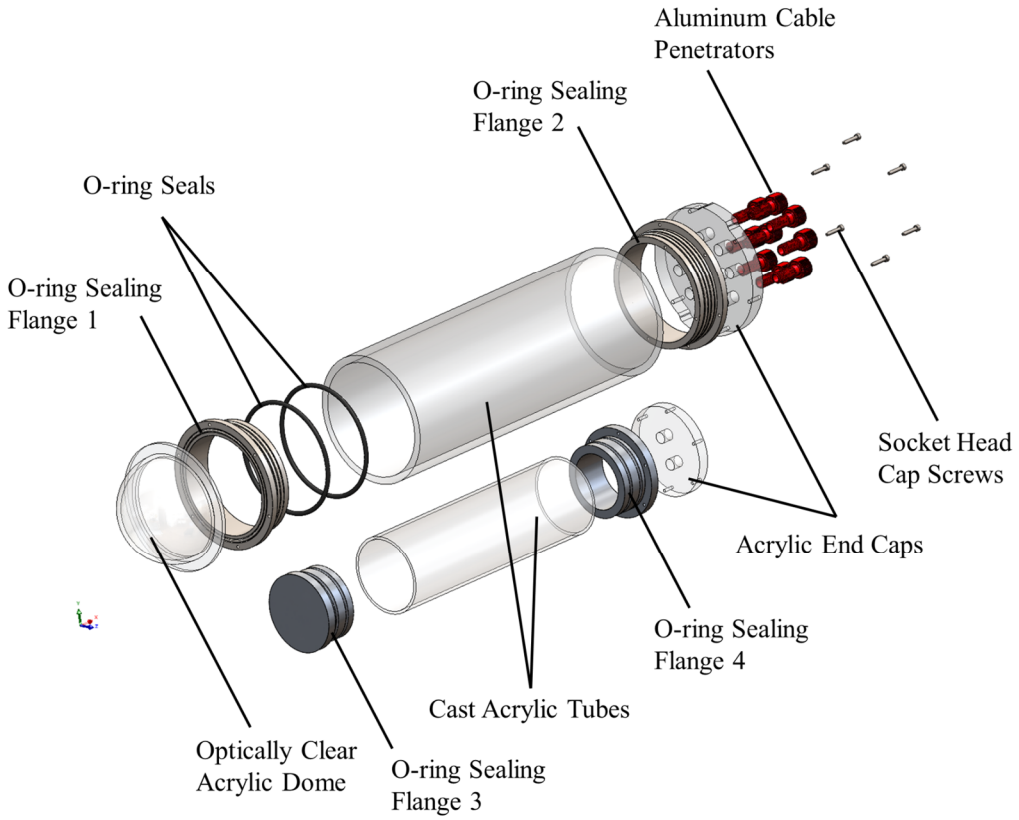


Figure 3.2 Exploded view of closed enclosures and sealing

After all parts of the waterproof housings were manufactured or supplied, the assembly process was carried out. After this stage, the enclosures had to be tested by applying sealing solutions in order to be fully sealed. A two-component epoxy adhesive with an initial setting time of 30 minutes and completely solidified within 24 hours was used to seal the cable entry points. Photographs of the sealing works are shown in Figure 3.3.

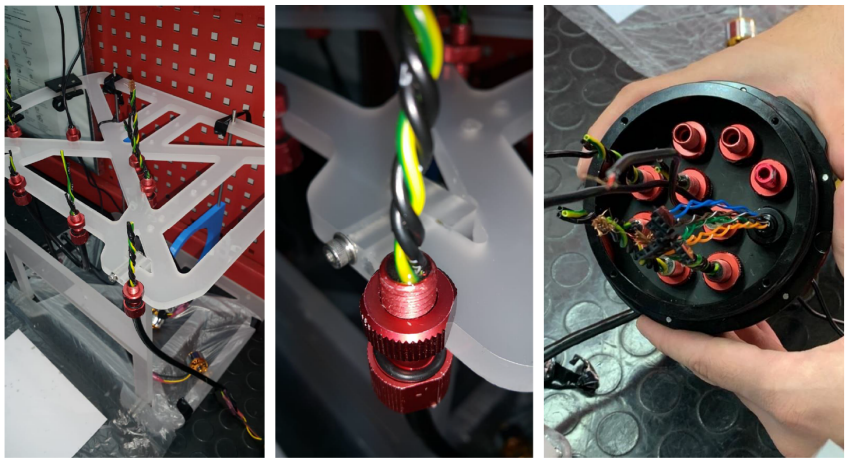


Figure 3.3 Sealing works on cable entry points

As it can be seen in Figure 3.4, sealing tests were carried out at different depths and times at various intervals to ensure the functionality of the O-ring and epoxy adhesives, and the results were observed.

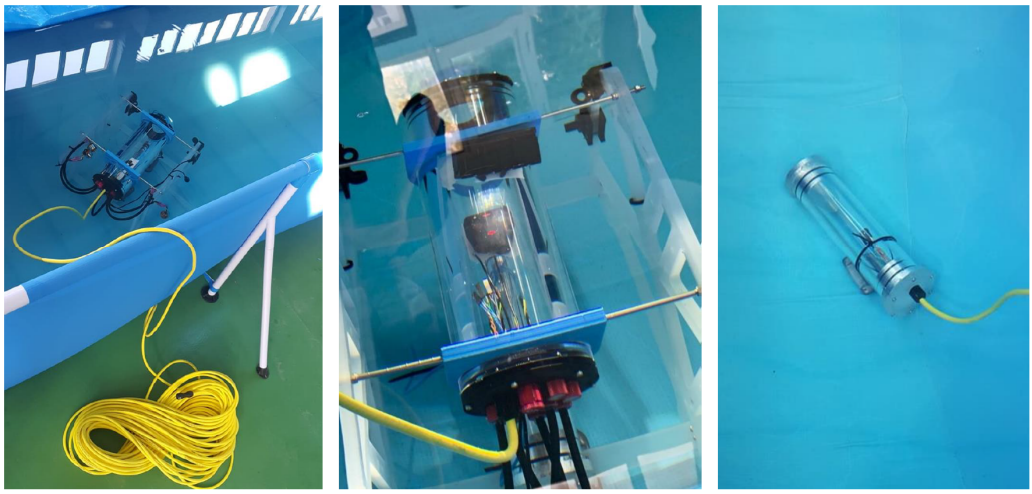


Figure 3.4 Sealing tests

3.2 Thrust and Propulsion System

During the implementation of the DEMarine AUV prototype, three different thrusters were used, from the initial to the finalized version. Although the axis locations and directions are almost close to each other, these three different models of thrusters vary considerably in terms of technical specifications. The technical specifications of the first, second and third generation thrusters are given in Table 3.1.

Table 3.1 Technical specifications of the 1st, 2nd and 3rd generation thrusters

Feature	**1st Gen**	**2nd Gen**	**3rd Gen**
Voltage [V]	11.1	12 V	7-20 V
Load Current [A]	12	15 A	24 A (16 V)
KV [RPM/V]	1000	800 KV	-
Thrust [kgf]	0.885	2	6.7
Power [W]	-	-	390 W (16 V)
Diameter [mm]	27.8	63 mm	100 mm
Length [mm]	27	80 mm	113 mm
Weight [g]	52	-	344 g

The first generation dc brushless motors used in the prototype are originally produced for aerial drones were replaced by higher performance thrusters in the previous operating period. After the second generation thrusters were used for a while, the more professional BlueRobotics T200 thrusters were used and an important step was taken for the underwater robot to perform real open sea missions. T200 model thrusters have a high thrust capacity of 14.8 lbf. The RPM, current (A), voltage (V), power (W) and efficiency (g/W) values provided by the manufacturer can be read depending on the PWM (μs). The three-dimensional exploded view of the final thruster system is given in Figure 3.5.

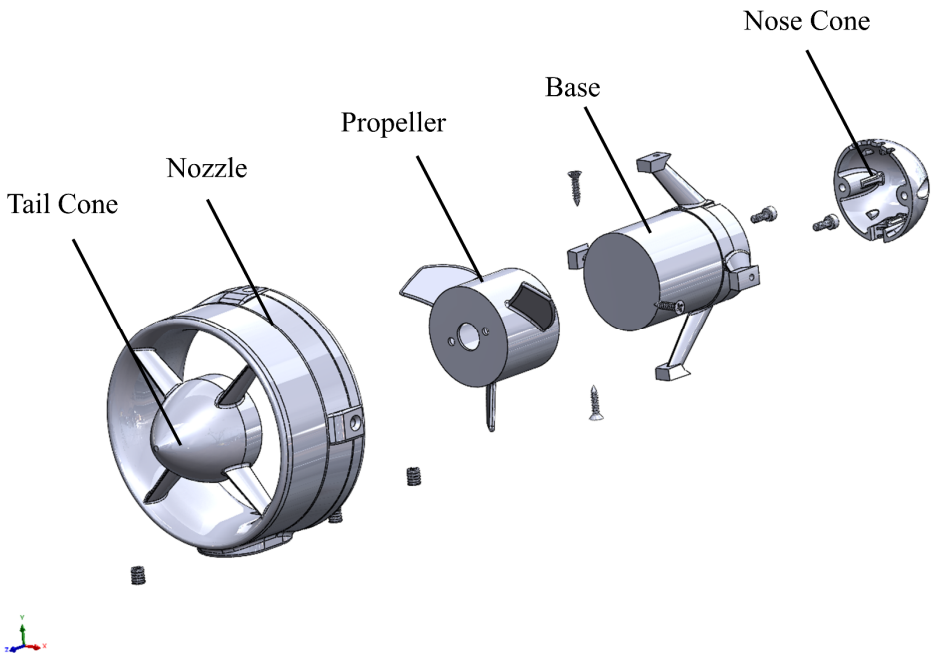


Figure 3.5 Exploded view of T200 thruster

The thrusters consist of brushless DC motor, three-bladed propeller, nozzles and motor holder flanges. While it is not possible to operate the second generation thrusters in two directions, reverse maneuvers can also be made thanks to the T200s. Thus, mobility in the negative surge axis is gained. The third generation thrusters were mounted on the prototype as seen in Figure 3.6 after being tested considering the propeller rotation directions.

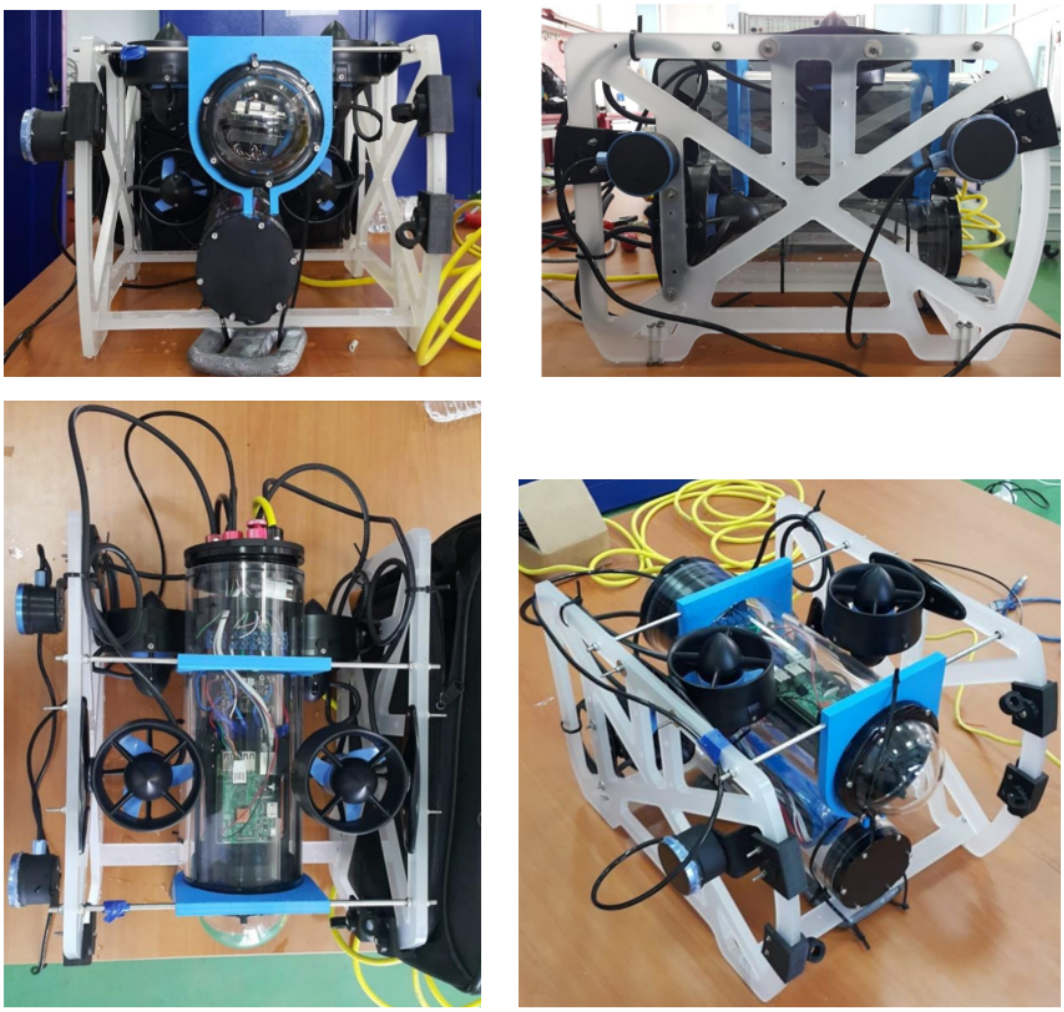


Figure 3.6 Front, left, top and perspective views of the DEMarine AUV prototype

The propulsion system of the underwater robot consists of four thrusters, two horizontal and two vertical axes.

3.3 Integrated Software and Electronics

The development of AUV's electronic hardware was carried out with the facilities of Dokuz Eylül University Robotics Laboratory and with the support of D.E.Ü BAP 2019.KB.FEN.011 and TÜBİTAK 118E689 projects. In laboratory studies, computers and embedded computers (Raspberry Pi 3 and compatible expansion cards), sensors (Inertial measurement unit, Camera), electronic speed controllers (ESC), batteries and electronic circuit components (Button, Terminal) provided within the scope of project

supports used. Embedded software and electronic system planning includes the gradual commissioning of different types of control structures that complement each other, as illustrated in Figure 3.7.

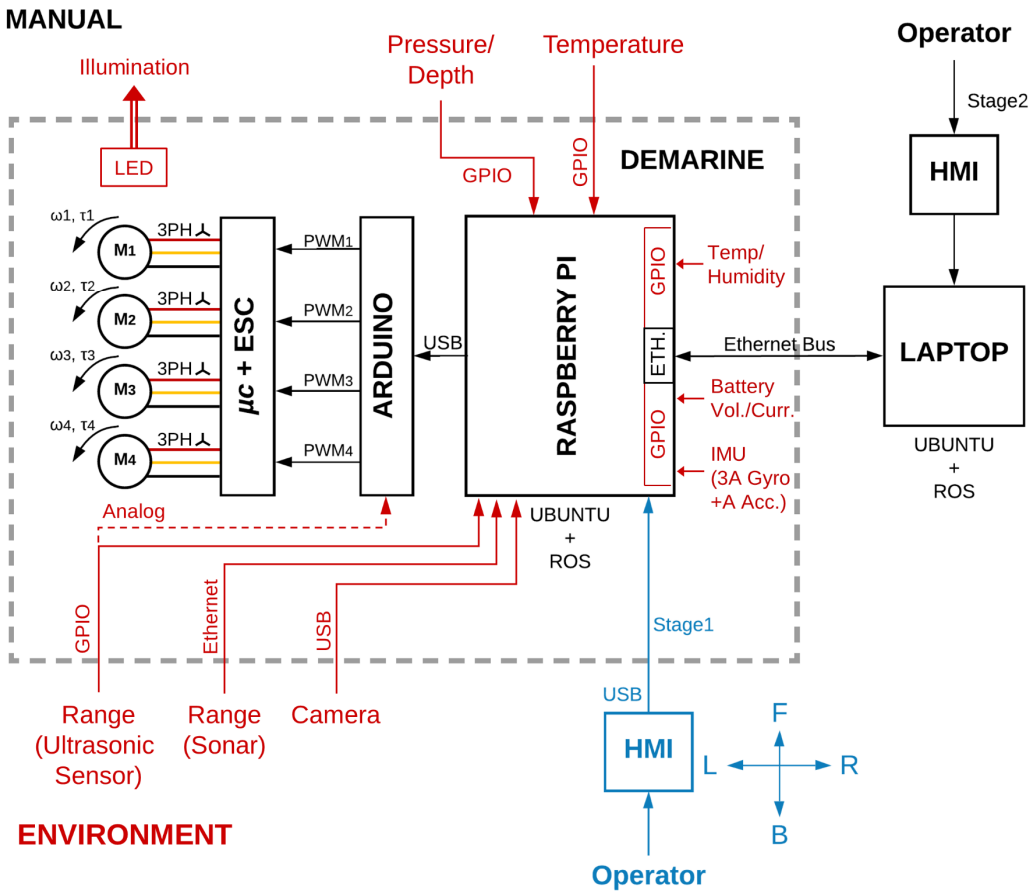


Figure 3.7 Block diagram abstraction of DEMARINE AUV electronic setup

As it can be seen from the figure, applied research studies were handled as human machine interface, sensor data logging/mapping, semi-autonomy and autonomy procedures. In the first stage, open loop control system has been implemented with the purpose of remote controlling of thrusters with dc motors. An operator generates the control signal via human machine interface device that transmits the analog signal through an embedded pc (Raspberry Pi + Arduino) which provides pulse width modulation signals for electronic speed controllers to drive the motors. Due to the ability of logging and evaluation of physical magnitudes from environment such as horizontal distance from objects, depth, velocity, acceleration and orientation it is

possible to be able to have the capability of semi-autonomy. From this stage a computer on surface station is used to process the data which requires high processing power. When switching to autonomy mode the operator with human machine interface deactivated and sensor data has been utilized as the only input for controller. Both Raspberry Pi and surface station computer linked to each other with Ethernet bus and runs Robot Operating System (ROS). ROS is a meta operating and open source operating system developed for robot control. Using ROS provides access to the facilities such as abstraction of hardware, communication and control of processes and devices, and necessary packages. In addition, ROS has libraries and toolboxes for configuring, developing, writing, and executing the codes between several computers. Thus, the control commands generated in line with the sensor signals and the navigation plan can be transmitted to the embedded computer and the propulsion system via the computer. In the following sections, the studies for the commissioning of the ROS system will be explained.

3.3.1 Master-Slave Connection

In order to establish a connection between multiple devices over ROS and to provide Host-Slave (Master-Slave) communication, all devices must be connected to the same network. Therefore, the computer and Raspberry Pi connected to the same router. PC and embedded computer connected to the common network presented in Figure 3.8.

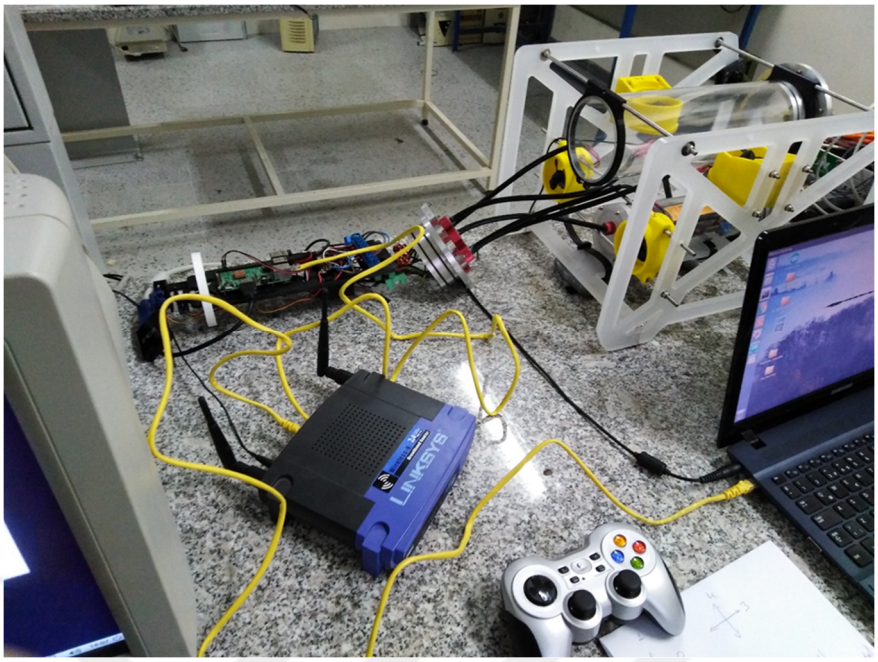


Figure.3.8 PC and embedded computer connected to the common network

To establish a connection between ROS and peripheral devices, master and slave devices must be defined. Currently, the computer is selected as the master device and the Raspberry Pi as the slave device. Linux-based Ubuntu operating system is installed on the devices and IP addresses are determined by entering the interface configuration (Interface Configuration, ifconfig) command on the terminal screen of the devices. Configuration steps applied in master and slave devices are given in Figure 3.9 and Figure 3.10.

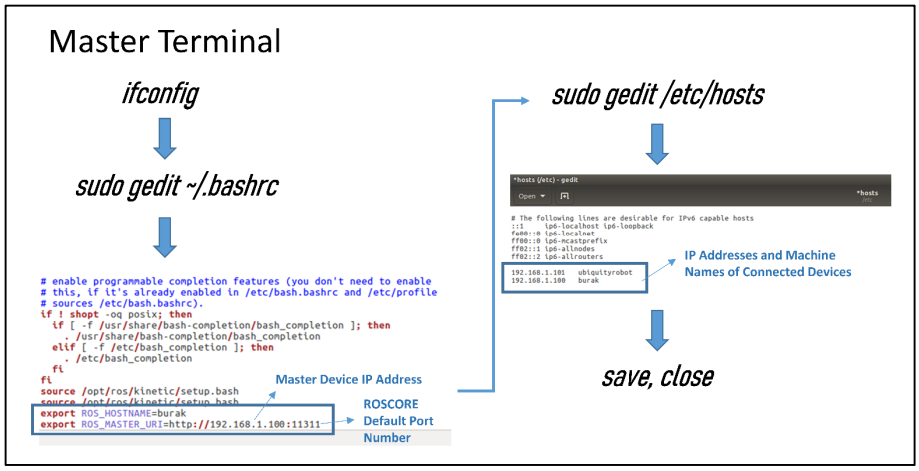


Figure 3.9 Master device configuration

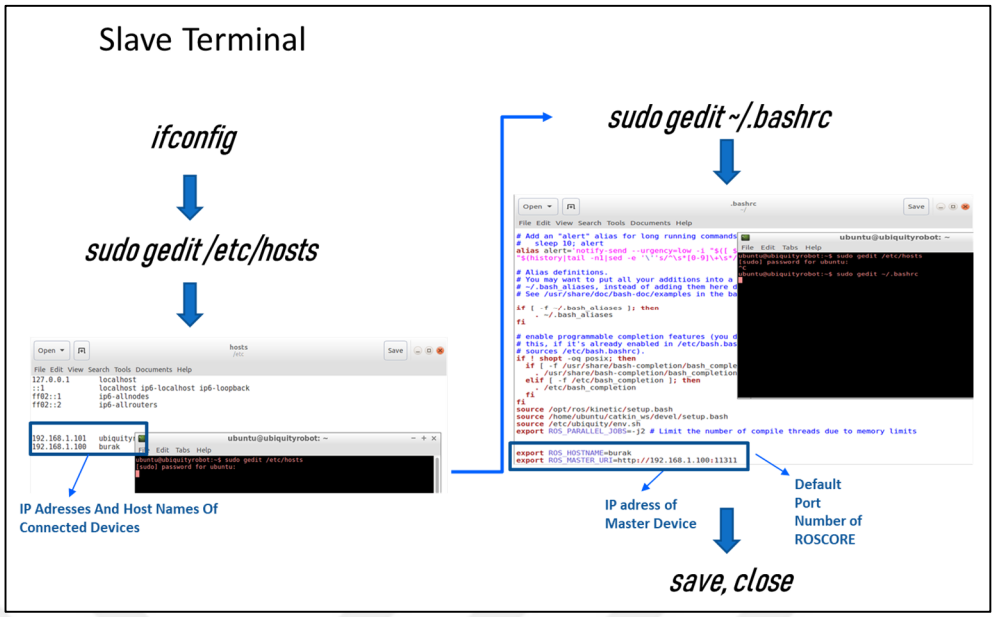


Figure 3.10 Slave device configuration

After configuring the master and slave devices separately, the connection was tested by running the "joy_node.py" sub-function of the "Joy" package running on ROS as in Figure 3.11.

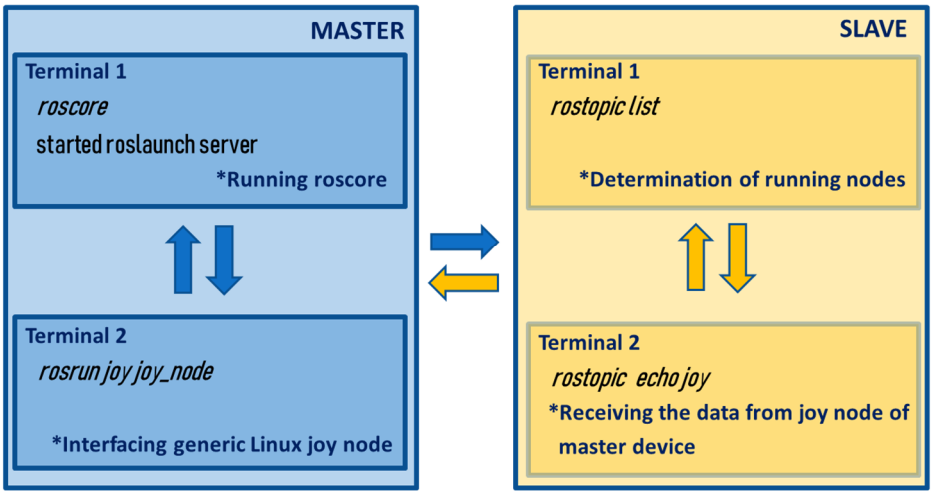


Figure 3.11 Master-Slave connection test

Thus, by running the "rostopic echo joy" command on the slave device, the status of each button and axis of the control device can be followed on the terminal screen.

3.3.2 Human Machine Interface Communication

The control signals from the remote control device are displayed by the ROS as digital values for the button and analog values in the range 0-1 for the axes. The produced button and axis values are mapped in Arduino to create PWM inputs to the ESCs. In Figure 3.12, the communication between the control device, embedded computers and the propulsion system can be seen.

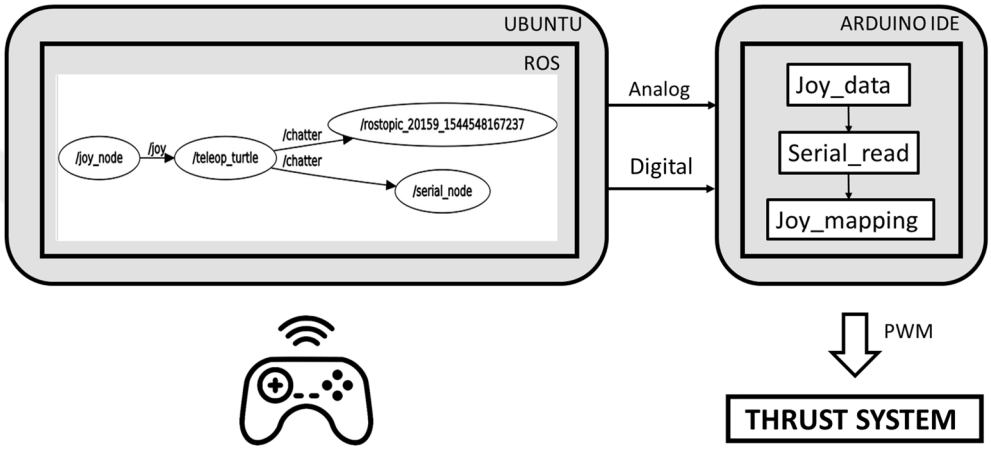


Figure 3.12 Human machine interface communication and software

The triggering scheme applied by the adjustable resistors on the control device to each DC motor is planned as shown in Figure 3.13 and the embedded computer is programmed to match the driver signals suitable for these tasks.

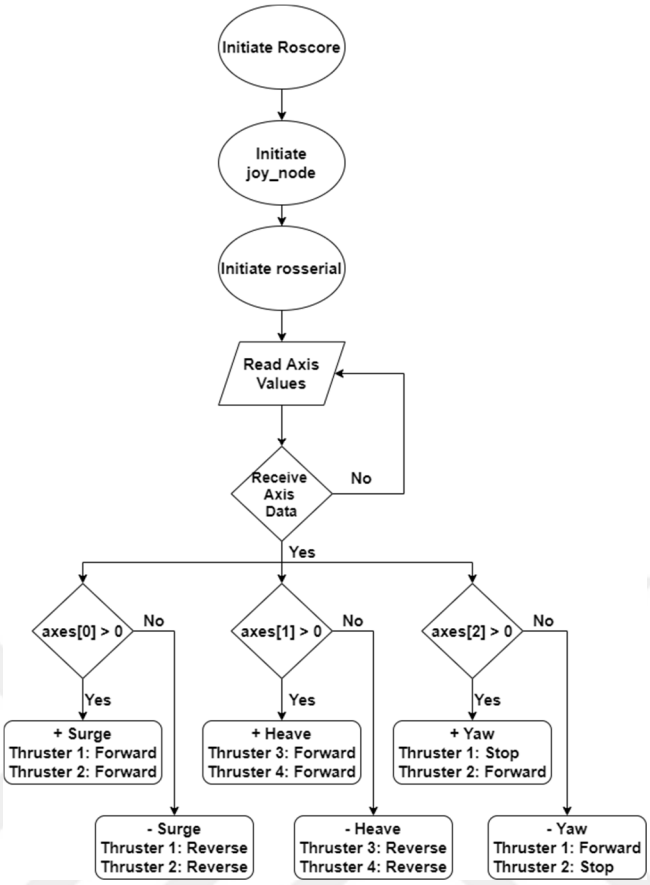


Figure 3.13 Brushless DC motors remote control diagram

The electronic equipment through which the signal flow is carried out in the control system is shown in Figure 3.14.

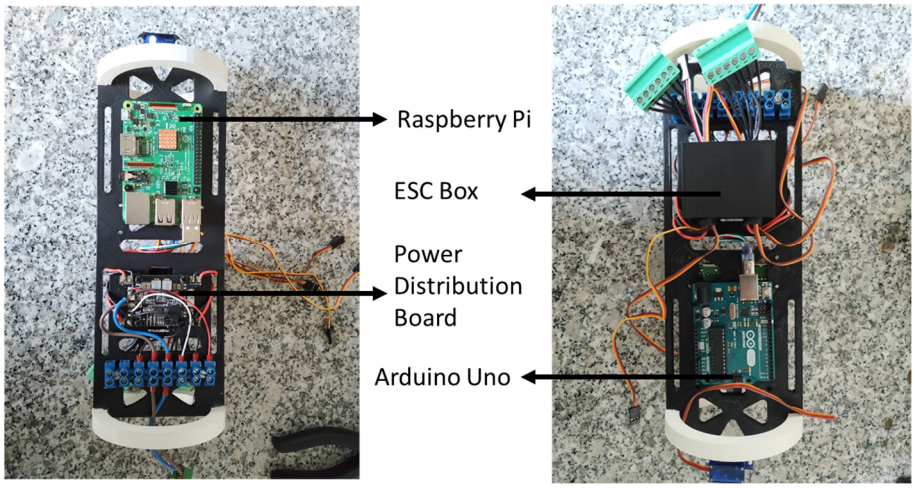


Figure 3.14 Open loop control system electronics

The general view of the open-loop control system setup is as in Figure 3.15.



Figure 3.15 Open loop control system electronics

In order to determine the forces and moments that will occur as a result of thrust on the prototype, the distances of the center of mass and thrust axes relative to each other have been determined. Thus, both the creation of the dynamical model has been contributed and some of the inputs to be entered into the control simulation model have been determined. The distances of the center of mass from thrust axes is given as in Figure 3.16.

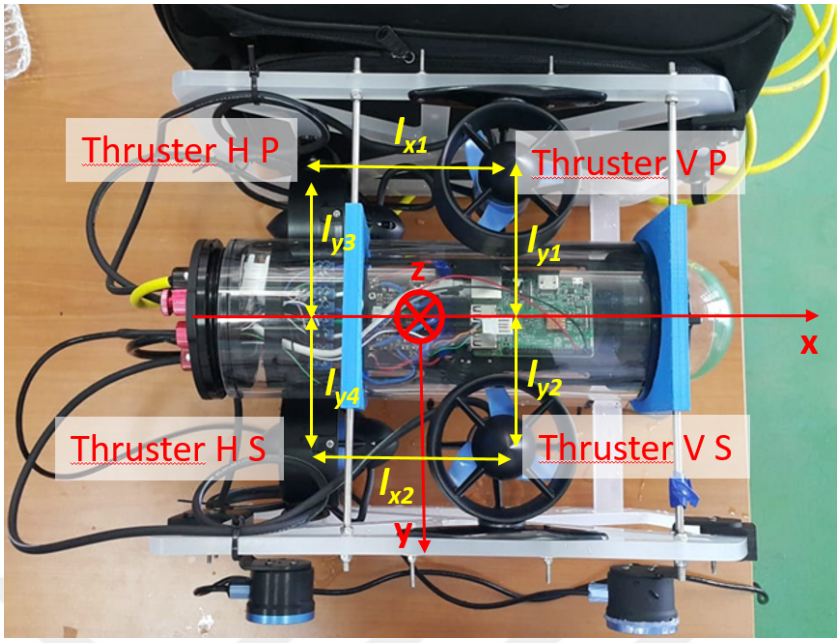


Figure 3.16 Distances of the center of mass from thrust axes

Top view of the prototype has been given above. The body fixed coordinate system which is fixed on the AUV prototype is red. Moment distances of the thrusters are colored yellow. Each thruster is named separately.

3.3.3 Embedded Sensing System and Instrumentation

After the open loop control system development and testing phases, sensing systems were configured and test setups were developed in order to upgrade the control structure to semi-autonomous and autonomous. A three-stage development plan was implemented for the measurement of each physical quantity:

- Establishment of an electronic circuit that energizes the sensor and generates a signal at the sensor output.
- Obtaining raw sensor data from embedded computer serial output channel
- Processing of sensor output data to obtain real physical measurement information

3.3.3.1 Hydrostatic Pressure and Depth Sensing

In order to obtain pressure and depth values of underwater environment, MS 5837-30BA pressure/depth sensor has been used. This sensor can measure up to 30 Bar (300 m depth) with a depth resolution of 2 mm, communicates over I2C and operates on 3.3 V I2C voltage but can accept power input up to 5.5 V. Instrumentation details of pressure/depth sensing system has been illustrated as in Figure 3.17.

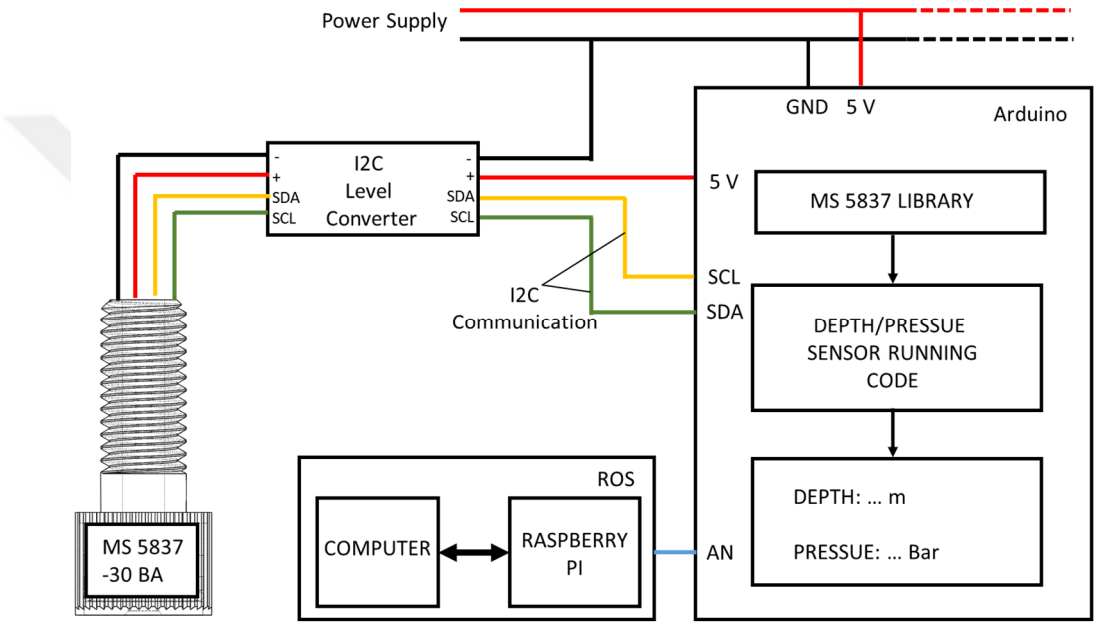


Figure 3.17 Pressure and Depth Sensor System Circuit Diagram

As seen in the figure above, power need of MS 5837-30BA has been provided by Arduino through I2C level converter. Sensor data has been transmitted via SCL which is the clock line used to synchronize all data transfers over the I2C bus and SDA which is the data line. As given in the Appendices section, Depth/pressure data has been obtained via Arduino code through MS 5837 library.

3.3.3.2 Distance measurement

Distance measurement values, has been obtained by BlueRobotics® ping sonar altimeter which is a single-beam echo sounder that measures distance up to 30 meters underwater. This sonar is planned to use for motion control algorithms such as obstacle avoidance and ship hull following scenarios. It has a beam width of 30 degrees and depth limit of 300meter. The sonar reserves a piezoelectric transducer for transmitting ultrasonic acoustic pulses in underwater and then captures the echoes. This data is used for to calculate the distance to the echoes, which is generally the seabed or an object. Instrumentation for echo sounding system has been illustrated as in Figure 3.18.

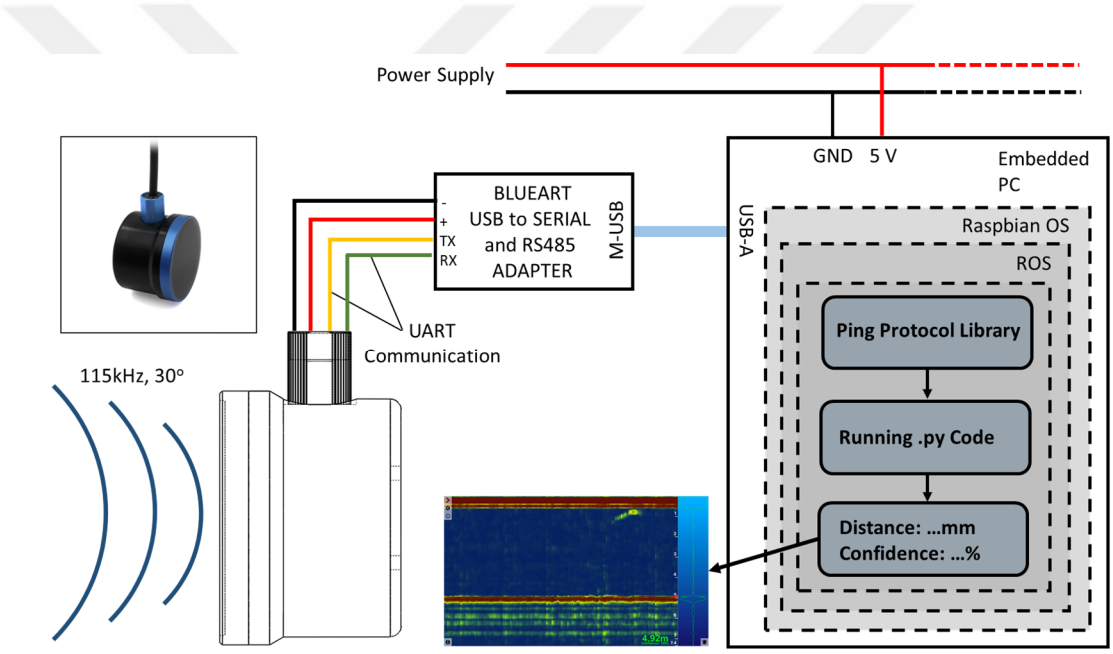


Figure 3.18 Distance sensor system circuit diagram

As seen in the diagram above, adopted ping sonar uses a 115 kHz transducer frequency. The connection to microcontroller device has been established computer through a BLUART USB to Serial adapter. The ping sonar uses UART communication which utilizes separate transmit (TX) and receive (RX) signal lines, which allows full-duplex communication. It communicates with a binary message format called the Ping-Protocol. Ping sonar activated and tested by Python coding (given in the Appendices section) running over Robot Operating System thus distance and measurement data could be acquired. To validate the acquisition setup and test the instrumentation the

ping sonar directly connected a computer running Linux operating system. Test results have been showed that the ping sonar provides 99% accurate results for $d > 0.7$ m distances. A total of three sonars of two different types were purchased as part of the project support. One of them is Ping 360 Scan Imaging type, which can scan 360°, and the other is Ping Sonar and Echosounder type with 30° beam width. Before starting practical work with sonar, they were all tested using the manufacturer's Pingviewer software as shown in Figure 3.19.



Figure 3.19 Sonar testing and graphical user interface

3.3.3.3 Visual data

Raspberry Pi wide-angle camera module has been planned to verify the data received from the ping sonar with visual data. Moreover, it is planned to be adopted to monitorize underwater environment both for exploring ship hull surfaces and establishing various autonomous applications such as obstacle avoidance and wall following. The camera module which has a visual angle of 200°, provides 5-megapixel 1080 p resolution with OV5647 sensor. The camera module uses Serial Camera Control Bus (SCCB) protocol to communicate with the embedded computer. Data received from the camera module has been processed to get real time image by raspicam_node inside Roscore. Then, the visual data has been transferred to master PC to monitorize the flow of real time images. The instrumentation diagram of the camera module is given in the Figure 3.20.

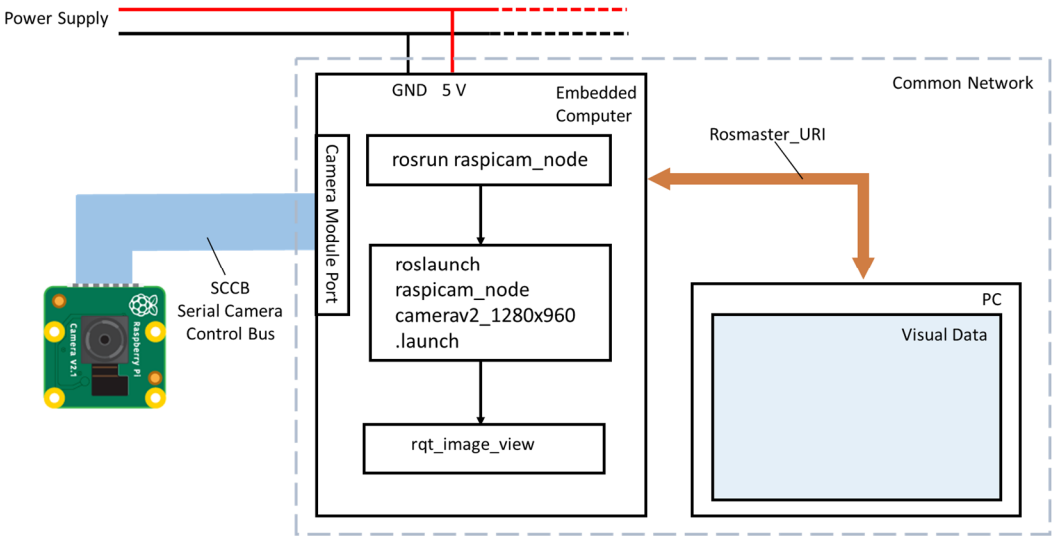


Figure 3.20 Camera module instrumentation diagram

The test whether the camera module is operational or not is performed as in the Figure 3.21.

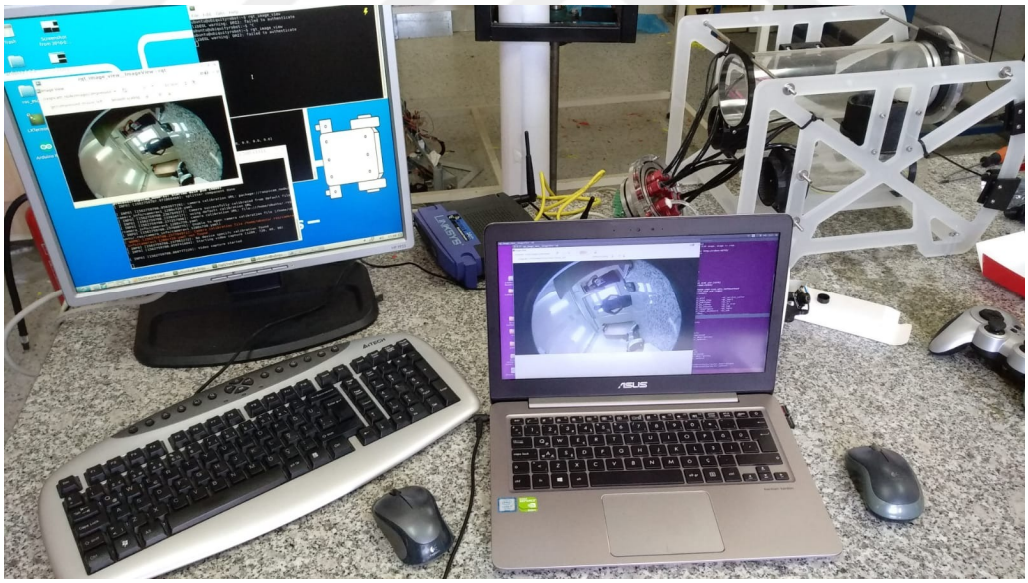


Figure 3.21 Camera module test setup

3.3.3.4 Inertial measurements

With purpose of acquiring forces, angular rates and the orientation of the AUV, The SparkFun 9DoF Razor IMU which constitutes of an accelerometer, gyroscope and magnetometer has been adopted. It combines a SAMD21 microprocessor with an MPU-9250 sensor. In the study, 5V, ground and I2C connections are provided and the Arduino code has been run to check whether there is data flow on serial screen. It is then intended to add the IMU Sensor as a node to Roscore, to sense the incoming data and display this data in the RVIZ (Simulation Screen). The instrumentation diagram of Inertial Measurement Unit has been given as in Figure 3.22.

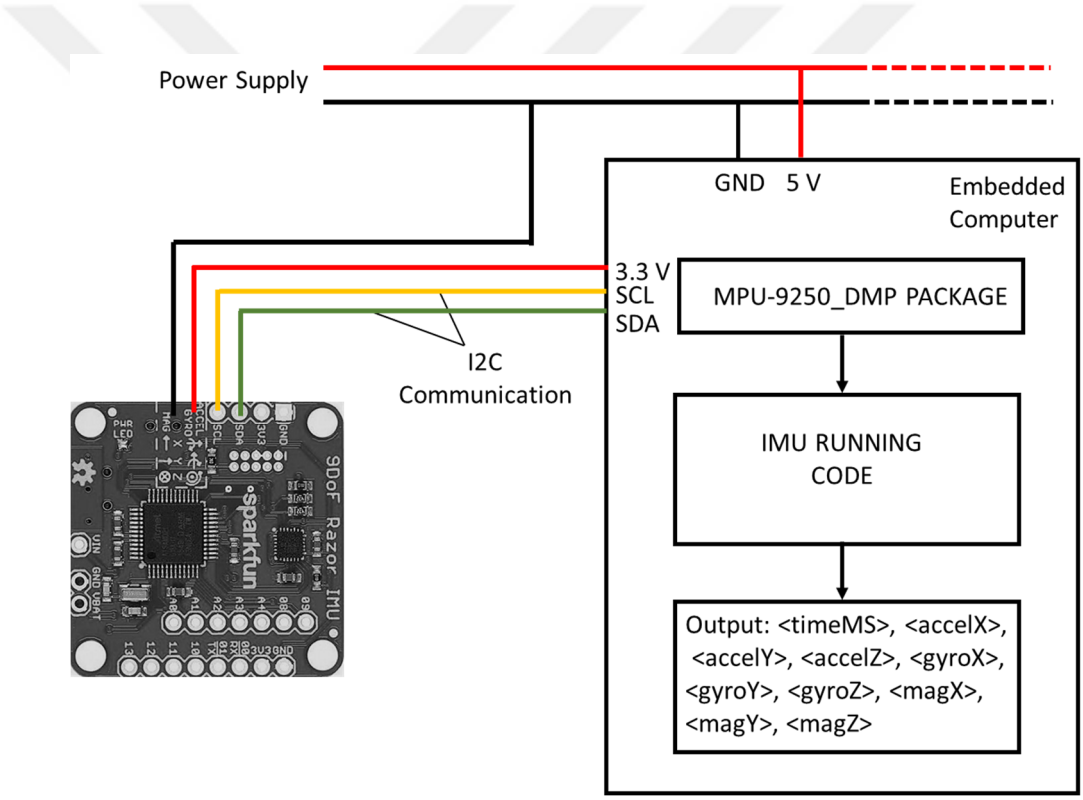


Figure 3.22 Inertia measurement system instrumentation diagram

The code for activating the Inertial Measurement Unit has been given in the appendices. In addition, for enhancement of IMU output reliability, studies on Kalman Filtering were carried out within the scope of inertial measurement applications. The Kalman Filter includes two basic steps: the use of the system dynamic model to predict the change of state between measurement times, and the use of the system

measurement model and measurements to optimally correct the predicted state. It is well known that the Kalman Filter is a recursive, computationally efficient and optimal in terms of the minimum average of squared errors. In the application of the Kalman Filter, the measurement and estimation are compared and the difference is multiplied and scaled with a parameter called Kalman gain. Kalman gain value should be selected as low to follow filter model estimates and high to follow filter output observations. Then this value is reflected to the system as feedback in order to predict the next step of the system more accurately. The filter tries to produce predictions that are closer to the unknown data than the predictions that can be obtained based on the predictions. Kalman Filter, which is an improved version of Bayesian Filter, works very efficiently in dynamic systems as it is applied continuously over time. The filter produces predictions of noisy or unknown data with their uncertainty. When the next predicted data are analyzed, the mean value is updated with less use of high uncertainty estimates in the data.

The mathematical modelling of Kalman Filter begins with subtracting state matrices from the predictive value term as in Equation 3.1.

$$x_k = K_k z_k + (1 - K_k).x_{k-1} \tag{3.1}$$

This equation can also be called the probabilistic starting point of the filter. Because the only unknown value in the specified equation is the x_k value, that is, the filtered value of the analyzed data. The value k, which is used as a subscript in the equation, is, for example, a time unit in milliseconds. At the end of the processes, it is expected to have filtered values that change depending on time.

While doing this, z_k, the measured values, and x_{k-1}, the estimated values in the previous case, are used. The last variable in the equation, K_k, is the Kalman gain. This value is assigned a value before the filtering operation, then the Kalman gain value is changed with various operations and the real values of the filtered measurements are converged.

The Kalman filter algorithm solves the general problem of predicting the x state of a discrete time controlled process governed by the linear random difference equation represented in Equations 3.2 and 3.3.

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \tag{3.2}$$

$$z_K = Hx_k + v_k \tag{3.3}$$

Each filtered data can be found with the help of Equation 3.3. The variable *w* used in this equation is the process noise, the *u* variable is the control signal, and the index *k* is a time expression. For example, X_{k-1} represents the predicted value of *X* before *k* units. *A*, *B* and *H* are matrices containing only numerical values.

The variable *v* in Equation 2 expresses the noise of the measurement and this value is completely independent of the *w* variable. The Kalman filter predicts a process using a feedback control form: the filter estimates the process state for *k* units of time, for example, and receives feedback in the form of noisy measurements.

Kalman Filter equations can be divided into two separate groups as time update equations and measurement update equations. Time update equations are responsible for projecting the current state and error covariance estimates forward to obtain predictions at the next time stage. Measurement update equations are responsible for feedback. Time update equations are given in Equations 3.4 and 3.5.

$$x'_k = Ax'_{k-1} + Bu_k \tag{3.4}$$

$$P'_k = AP_{k-1}A^{\mathrm{T}} + Q \tag{3.5}$$

Equation 3.3 estimates the previous data, while Equation 3.5 finds error covariance. Measurement updates are given in Equations 3.6 to 3.8.

$$K_k = P'_k H^T (H P'_k H^T + R)^{-1} \quad (3.6)$$

$$x'_k = x'_{k'} + K_{k'}(z_k - H x'_{k'}) \quad (3.7)$$

$$P_k = (1 - K_k H) P'_k \quad (3.8)$$

Kalman gain can be calculated with Equation 6. With the feedback of this calculated value, the estimated data in Equation 7 is updated. With Equation 8, the error covariance is updated in the light of the previous data. With the above equations, the target is to find Q measurement error covariance and R to find process noise. Kalman Filter will work more efficiently when these variables take an optimum value. In most cases it is possible to estimate the noise, so R can be found. The approximate correct value of the Q value can only be found by applying the Kalman Filter and interpreting it.

Practical study of Kalman Filter applied on IMU. The filter algorithm programmed on the embedded computer was run for the 9 DoF IMU data. In the first place, the necessary ROS packages are defined in the Ubuntu operating system environment. The Visual Python package was used to display the data. The data received from the sensor is usually processed in Arduino. However, in actual IMU model, data flow can also be provided with a micro USB 3.0 cable. In order to achieve this, the relevant boards are loaded in the Arduino IDE program in ROS. The path to be followed in order to install the boards: Tools > Board > Boards Manager, then search "Arduino SAMD Boards (32-bits ARM Cortex-M0+)".

In addition, if the board specially developed by the manufacturer of the IMU is loaded, it will be possible to view the IMU used. In the next step, the IMU was connected to the Raspberry Pi via micro USB and the correct port was selected. With this step, the data (roll, pitch and yaw) received from the sensor with various commands in ROS will be added and displayed. In order to detect whether the data has been read or not, the data flow was viewed from the serial monitor on the integrated development environment of the embedded computer. After making sure that the data

was read correctly, imaging was performed on the RVIZ (ROS Visualization) screen running as a node in the ROS operating system. As a final step, Body Fixed Frame is set to “base_link” in RVIZ program and position vectors are shown in Figure 3.23 on a unit cube.

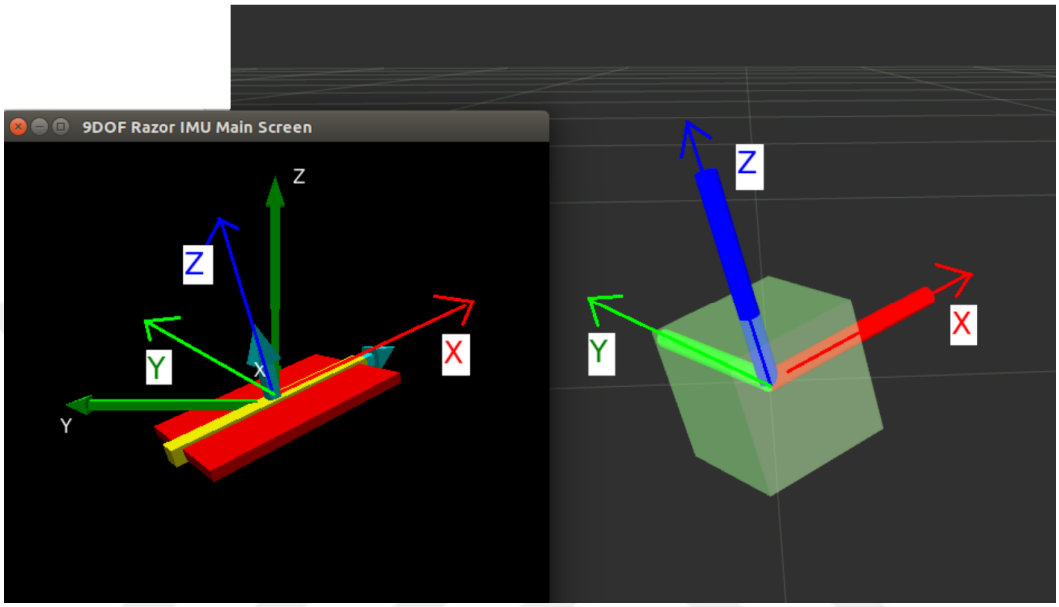


Figure 3.23 Position vectors simultaneously visualized on RVIZ

In order to monitor the movements of a virtual robot, Turtlebot_simulator package is used in Gazebo simulation program. Time-dependent locations must be added as a different node to compare filtered and unfiltered trajectories. Using the odometry data, the odometry_to_trajectory package creates two different trajectories. The comparison of filtered and unfiltered position data obtain from IMU is given in Figure 3.24.

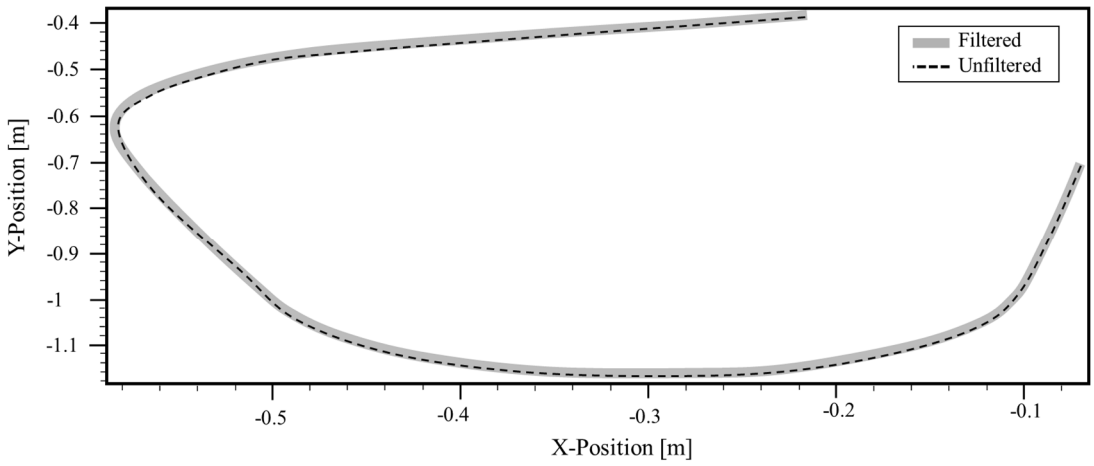


Figure 3.24 Filtered and unfiltered IMU data

3.4 Experimental Setup and Underwater Tests

A test pool was set up in the laboratory environment to conduct underwater experimental studies of the DEMarine AUV prototype. The prefabricated test pool installed has dimensions of 400 cm in length, 211 cm in width and 81 mm in depth. With these dimensions, it has a volumetric capacity of 6.84 m^3. Each design, hardware and methodical development of AUV's mechanical, electronic, propulsion and sensor systems were tested in the underwater environment of the test pool. Underwater tests played an important role in the development of different thrusters, different batteries, various electrical equipment and software. Images of the experimental study and testing process are shown in Figure 3.25.

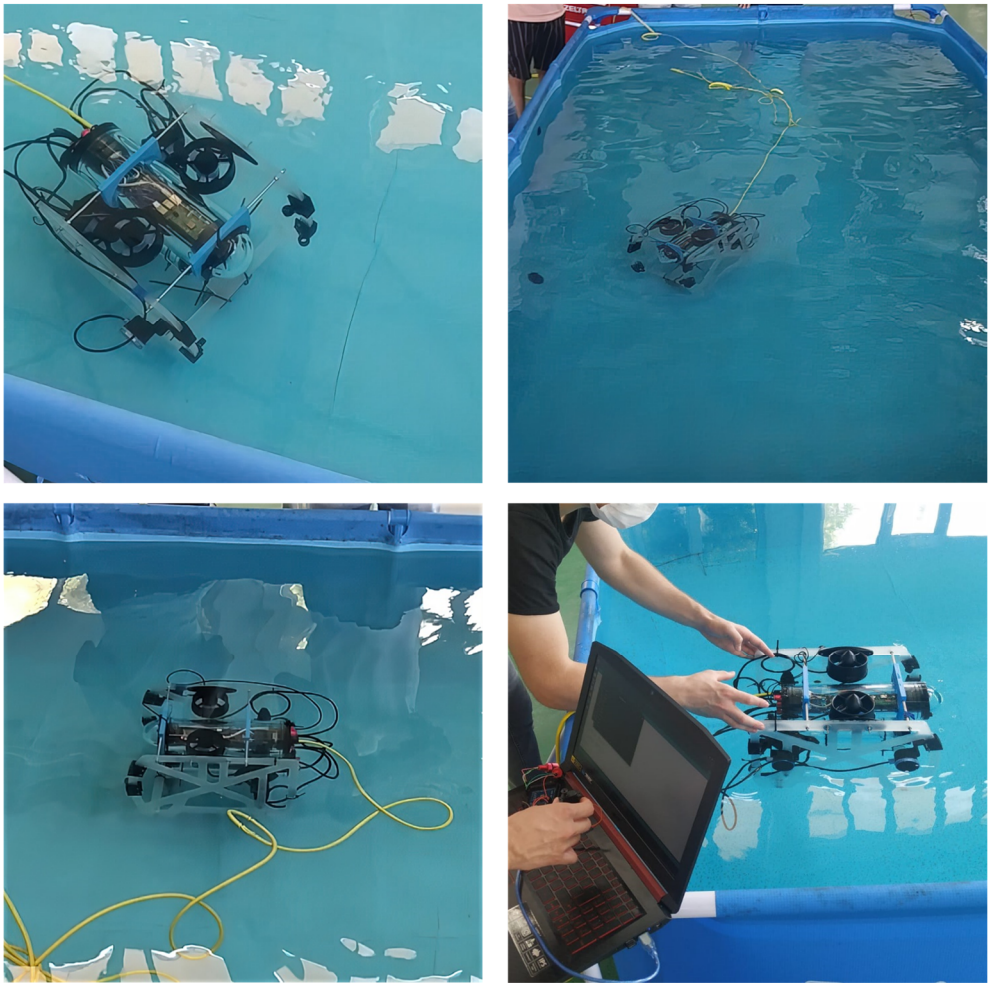


Figure 3.25 Underwater experiments in test pool

As the tests were carried out, the performance and maneuver specifications of the DEMarine AUV prototype were revealed as given in Table 3.2.

Table 3.2 Performance and maneuver specifications of DEMarine AUV

Performance Parameter	**Unit**	**Value**
Max. Horizontal Speed	m/s	2-2.5
Max. Vertical Speed	m/s	1.5-2
Nominal Speed	m/s	1
Navigation Time	min	>30
Min. Steering Radius	m	≈1.5 (at 1m/s)
Motion Axes	-	Surge, Heave, Yaw

CHAPTER FOUR
HYDRODYNAMIC ANALYSIS

In this section, the hydrodynamic analyzes of the DEMarine AUV under certain boundary conditions, traveling along separate trajectories designed for two different mission scenarios, were conducted. Orbital geometries have been formulated and waypoints determined specifically for the structure of each trajectory. The first of the determined trajectories was designed to circumferentially follow a ship's hull below the waterline (Kerim Deniz Kaya et al., 2020). The second was designed on the assumption that the AUV would follow a logarithmic trajectory. In the following sub-headings, the orbital energy model, flow model, boundary conditions and findings will be given independently for each mission scenario.

The hydrodynamic analysis part of the thesis consists of two different studies for orbit in two different geometries. The first study organized for obtaining thrust capacity, power and energy parameters of DEMarine AUV, which follows a logarithmic trajectory segment. In the first instance, the curve equations are identified parametrically to conceive the pertinent trajectory characteristics along which the AUV performs its navigating motion. Afterwards, thanks to the algorithm developed and programmed, curvature and radius of curvature data sets for each point on the trajectory were acquired. Subsequently, in compliance with the trajectory, a fixed speed free streamflow domain was simulated. Within the flow domain, CFD analysis which applies FVM method was carried out for various points on the trajectory. The utilized feature of the logarithmic trajectory is that the radius of curvature changes at each point. The heading and lateral thrust components and the total power requirement in the course of the navigation were obtained utilizing hydrodynamic force values due to the radius of curvature received from the CFD analysis.

4.1 Determination of Hydrodynamic Characteristics: Scenario 1

The navigational scenario 1 for the AUV is based on the environmental geometry of the research boat DEU 1, which is used in scientific studies of Dokuz Eylül

University Institute of Marine Science and Technology. In the scope of the scenario, AUV is requested to follow the water line 3 on the wetted surface of the floating marine vessel, keeping 1 m distance within a margin of ±150 mm. General view and technical specifications of DEU 1 are presented in Figure 4.1 and Table 4.1



Figure 4.1 General view of research vessel DEU 1

Table 4.1 Technical specifications of research vessel DEU 1

Specification	Symbol	Unit	Value
Length	L_{oa}	m	26.4 m
Breadth	B	m	8 m
Depth (from keel)	D	m	2.5 m
Draught	T	m	2 m
Main Machine	P	hp	400 hp
Personnel Capacity	-	-	5 vessel staff and 8 academic staff

XY coordinates of water line 3 was obtained from the half breadth distances of DEU 1 as shown in Table 4.2.

Table 4.2 Half breadth table of research vessel DEU 1

Number of Station	WL_1 mm	WL_2 mm	WL_3 mm	WL_4 mm	Main Deck mm
Transom	-	-	400	2840	3100
0	-	-	1080	3040	3280
1	80	230	2450	3300	3500
2	240	800	3240	3540	3700
3	340	1660	3530	3730	3850
4	410	2530	3700	3840	3950
5	440	2800	3750	3890	4000
6	400	2700	3600	3800	3950
7	340	1970	3090	3440	3750
8	20	1210	2120	2630	3320
9	10	680	1030	1420	2310
10	-	-	-	-	750

WL_j indicates the set of lines on marine vessels which are parallel to water surface when the vessel is afloat in a normal position.

4.1.1 Trajectory Planning of Scenario 1

For the WL_3 obtained from Table 4.2, the intervals of the XY coordinates are precisely divided by auxiliary stations and the curve fitting algorithm is aimed to give a more accurate result, especially in critical regions where the radius of curve is low. Augmented XY coordinates were offset by 1 m to determine the points that AUV should reach during the mission. Following this step, cubic spline curve fitting method was applied to determine the ideal trajectory of AUV. The reason of adopting cubic spline curve fitting is to handle slope changes smoothly which is not possible with lower order splines. Therewithal, higher order splines are not preferred because of instabilities inherent in higher order polynomials (Chapra, 2011). Cubic spline curve fitting method is based on cubic spline interpolation which is a way of fitting respective

polynomials in between each two points. A third degree polynomial can be written as in Equation 4.1.

$$f_i(x)=a_i x^3+b_i x^2+c_i x+d_i \qquad i=1,2,\ldots,n \tag{4.1}$$

Here, for n+1 data points and n intervals, there are 4n unknown coefficients need to evaluate by a set of 4n equations. It is possible to reduce the number of equations in the solution set to n-1 by applying an approach which is based on observation that the second derivative within each interval is a straight line (Cheney & Kincaid, 2013). This observation can be proved by differentiating twice of Equation (4.1) as in Equations 4.2 and 4.3.

$$f_i'(x) = 3a_i x^2 + 2b_i x + c_i \qquad i = 1,2,\ldots,n \tag{4.2}$$

$$f_i''(x) = 6a_i x + 2b_i \qquad i = 1,2,\ldots,n \tag{4.3}$$

The derivation from the mentioned approach gives the cubic equation for each interval as in Equation 4.4.

$$\begin{aligned} f_i(x) &= \frac{f_i''(x_{i-1})}{6(x_i - x_{i-1})}(x_i - x)^3 + \frac{f_i''(x_i)}{6(x_i - x_{i-1})}(x - x_{i-1})^3 \\ &+ \left[\frac{f(x_{i-1})}{x_i - x_{i-1}} - \frac{f''(x_{i-1})(x_i - x_{i-1})}{6}\right](x_i - x) \\ &+ \left[\frac{f(x_i)}{x_i - x_{i-1}} - \frac{f''(x_i)(x_i - x_{i-1})}{6}\right](x - x_{i-1}) \end{aligned} \tag{4.4}$$

The unknowns which are the second derivatives at the end of each interval can be calculated using Equation 4.5.

$$\begin{aligned} &(x_i - x_{i-1})f''(x_{i-1}) + 2(x_{i+1} - x_{i-1})f''(x_i) + (x_{i+1} - x_i)f''(x_{i+1}) \\ &= \frac{6}{x_{i+1} - x_i}\left[f(x_{i+1}) - f(x_i)\right] + \frac{6}{x_i - x_{i-1}}\left[f(x_{i-1}) - f(x_i)\right] \end{aligned} \tag{4.5}$$

Considering the second derivatives at the end nodes equals zero, if Equation 4.5. is written for all interior nodes, a solution set of n-1 equations with n-1 unknowns is obtained. Furthermore, the system of equation set is tridiagonal thus the solution is simplified. The system of equation forms as in Equation 4.6.

$$\begin{bmatrix} 1 & & & & \\ h_1 & 2(h_1+h_2) & h_2 & & \\ & \cdots & \cdots & \cdots & \\ & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ & & & & 1 \end{bmatrix} \begin{Bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{Bmatrix} = \begin{Bmatrix} 0 \\ 3(f[x_3,x_2]-f[x_2,x_1]) \\ \vdots \\ 3(f[x_n,x_{n-1}]-f[x_{n-1},x_{n-2}]) \\ 0 \end{Bmatrix} \quad (4.6)$$

Here, h_i, c_i and $f[x_i\ x_j]$ are identified as in Equations 4.7, 4.8 and 4.9 respectively.

$$h_i(x) = x_{i+1} - x_i \qquad i = 1,2,\ldots,n \quad (4.7)$$

$$c_i = c_{i+1} - 3d_i h_i \qquad i = 1,2,\ldots,n \quad (4.8)$$

$$f[x_i, x_j] = \frac{f_i - f_j}{x_i - x_j} \qquad i = 1,2,\ldots,n \quad (4.9)$$

Cubic spline curve fitting algorithm was computerized and run by JAVA® programming language. The output of the cubic spline curve fitting algorithm is plotted as shown in Figure 4.2.

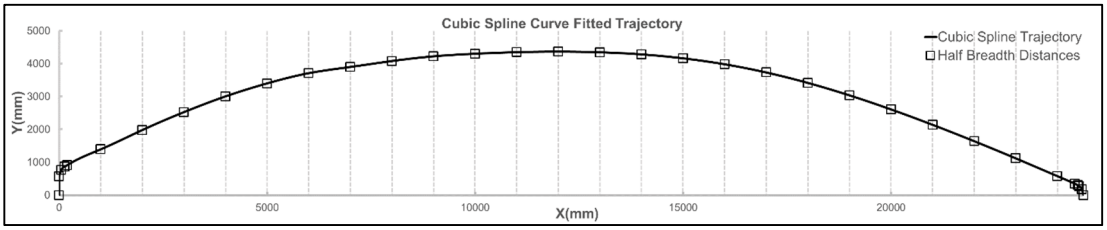


Figure 4.2 Cubic spline curve fitted trajectory

With the purpose of generating straight distance d_k and yaw angle ψ_k data sets which will be used as an input of motion induced energy requirement evaluation algorithm,

the ideal trajectory was degraded into broken line form. Degraded trajectory was designed by plotting tangent lines through predetermined derivation points. AUV changes its angular position around Z_b axis at certain points where each tangent line passing through derivative point $p(x_k,y_k)$ intersects with the tangent line passing through the next derivative point $p(x_{k+1},y_{k+1})$. As it is seen in Figure 4.3, the location and number of derivative points was optimized ensuring that the intersection points remain within the ±150 mm trajectory margin.

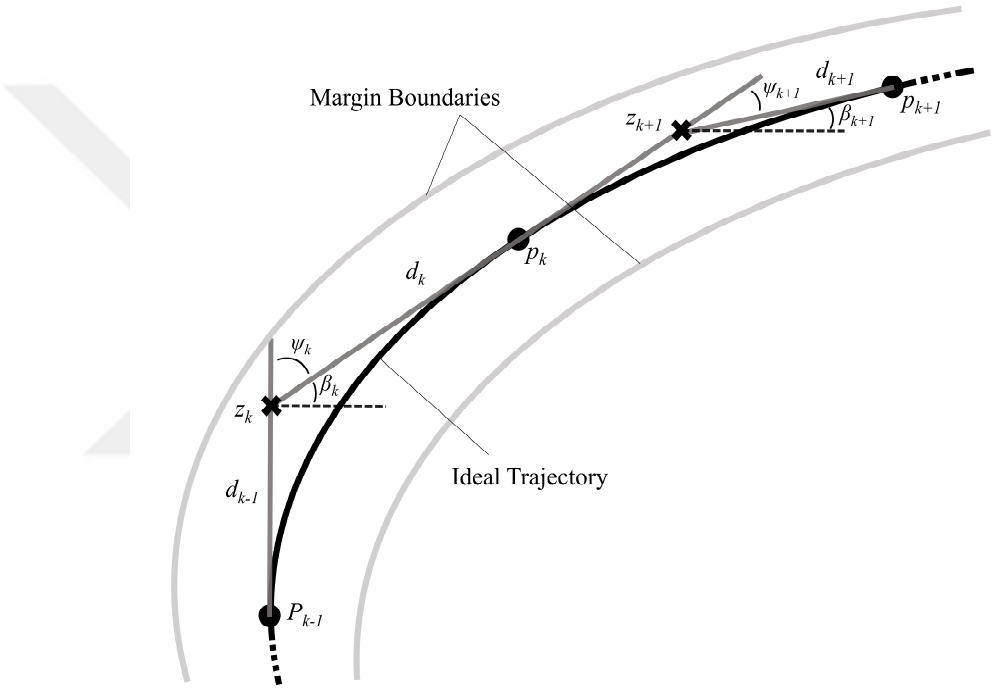


Figure 4.3 Derivation and tangential intersection points along ideal trajectory

From the designated tangent points $p(x_k,y_k)$, each slope β_k was obtained by differentiating the cubic spline polynomial defined for the relevant interval. The yaw angle ψ_k data set that AUV should perform at each intersection point was evaluated based on the difference of successive slopes formulated as in Equations 4.10 to 4.12.

$$\beta_k = \arctan\left(\frac{df_k}{dx_k}\right) \qquad k = 1,2,\ldots,m \tag{4.10}$$

$$\psi_k = \beta_{k-1} - \beta_k \qquad i = 1,2,\ldots,m \tag{4.11}$$

$$\psi_k = \arctan\left(\frac{df_{k-1}}{dx_{k-1}}\right) - \arctan\left(\frac{df_k}{dx_k}\right) \qquad k = 1,2,\ldots,m \tag{4.12}$$

The set of each straight distance d_k along the trajectory was evaluated by Equation 4.13.

$$d_k = \frac{x_k - x_{k-1}}{\cos\beta_k} \qquad k = 1,2,\ldots,m \tag{4.13}$$

4.1.2 Energy Modelling of Scenario 1

Data sets d_k and ψ_k which are obtained from Equations 4.12 and 4.13, will be used as an input of motion induced energy requirement evaluation algorithm. The amount of energy that the AUV will consume throughout the trajectory was expressed as the sum of the energies it consumes while performing the straight path navigation $E_{k_{linear}}$ and yaw movements around the Z_b axis $E_{k_{angular}}$. For ψ equals zero and corresponding drag force $F_{\psi=0}$ obtained from hydrodynamic analysis result which can be seen in Figure 4.15, $E_{k_{linear}}$ is evaluated as in Equation 4.14.

$$E_{k_{linear}} = F_{\psi=0} d_k \qquad k = 1,2,\ldots,m \tag{4.14}$$

For ψ_k data and corresponding drag force F_{ψ_k} obtained from hydrodynamic analysis result which can be seen in Figure 4.15, $E_{k_{angular}}$ is evaluated as in Equation 4.15.

$$E_{k_{angular}} = F_{\psi_k} b_{AUV} \psi_k \, \pi/180 \qquad k = 1,2,\ldots,m \tag{4.15}$$

Here ψ_k transformed into radian unit and F_{ψ_k} have an impact on AUV as a force couple in which width size of AUV b_{AUV} is the perpendicular distance between the

forces. The total energy consumption throughout the trajectory E_{total} is evaluated as in Equation 4.16.

$$E_{total} = \sum_{k=1}^{m}\left(E_{k_{linear}} + E_{k_{angular}}\right) = \sum_{k=1}^{m}\left(F_{\psi=0} d_k + F_{\psi_k} b_{AUV} \psi_k \, \pi/180\right) \tag{4.16}$$

Average power need of horizontal thrusters P_{avg} can be reached using E_{total} and time of travel Δt as seen in Equation 4.17.

$$P_{avg} = \frac{E_{total}}{\Delta t} = \frac{E_{total} u}{\sum_{k=1}^{m} d_k} \tag{4.17}$$

Here linear velocity of AUV is expressed by u. Since AUV has two horizontal thrusters, the total power requirement is halved for each thrusters and expressed with P_t. For identified P_t, if the operating voltage V and battery capacity C_{batt} in terms of ampere hour is known, total corresponding amperage draw I_{batt} gives the discharge time of battery t_{batt} as in Equation 4.18.

$$t_{batt} = \frac{C_{batt}}{I_{batt}} \tag{4.18}$$

Navigation range l_{nav} with respect to t_{batt} and u is formulated as in Equation 4.19.

$$l_{\max} = t_{batt} u \tag{4.19}$$

4.1.3 Computational Study of Scenario 1

The CFD simulation was run on the first version of the DeMarine observer type AUV prototype which is implemented in Dokuz Eylul University Robotics Laboratory for research in shallow water and observations on ship wetted surfaces. DEMarine

AUV comprises of main frame, two vertical thrusters, two horizontal thrusters, dry box with electronic hardware inside and battery box as shown in Figure 4.4.

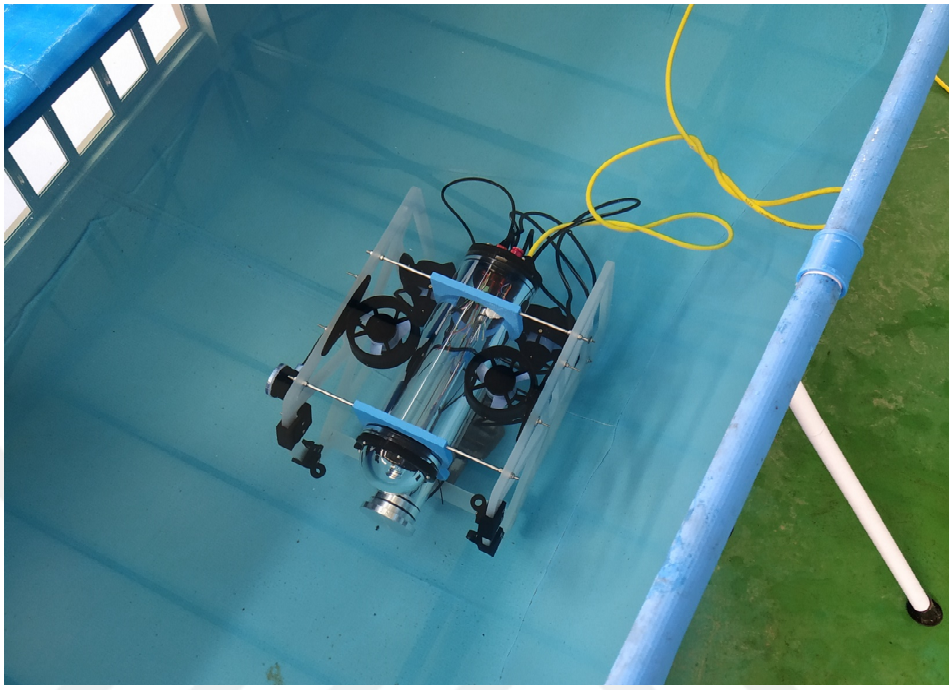


Figure 4.4 DEMarine AUV

Technical specifications of DEMarine AUV is given in Table 4.3.

Table 4.3 Technical specifications of DEMarine AUV

Size	300 mm (Width) x 280 (Height) mm x 400 mm (Depth)
Weight	5 kg
Maximum Working Depth	Linear motion along the y-axis
Thrusters	2 Vertical, 2 Horizontal
Maximum Speed	1.5 m/s
Camera	Raspberry fish eye cam, 180° field of vision
Sensors	Ping sonar, IMU, depth, temperature, humidity
Cable	30 m, 4 pairs
Battery	22.2 V, 10 Ah
Lights	2 x 200 lumen

Numerical results were presented with ANSYS FLUENT 19.1 commercial CFD software. The following sections will include criteria such as the settings of the selected solver, governing equations, boundary conditions and mesh independence.

4.1.3.1 Meshing and Boundary Conditions of Scenario 1

The flow domain was created as a symmetrical geometry which has a velocity input and atmospheric pressure output as shown in Figure 4.5.

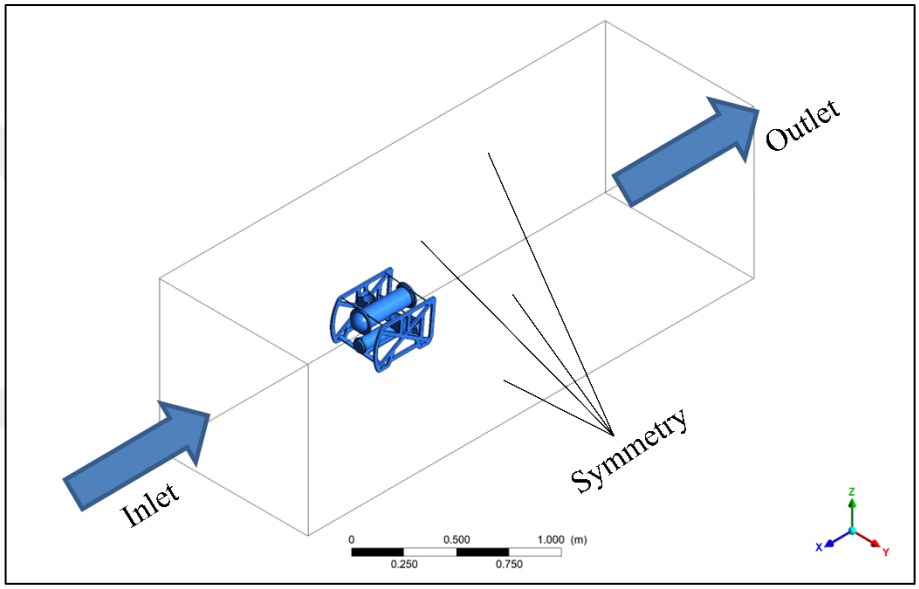


Figure 4.5 Flow domain setup

Mesh size and mesh form was adjusted to be finer in the areas close to critical points as shown in Figure 4.6 and Figure 4.7.

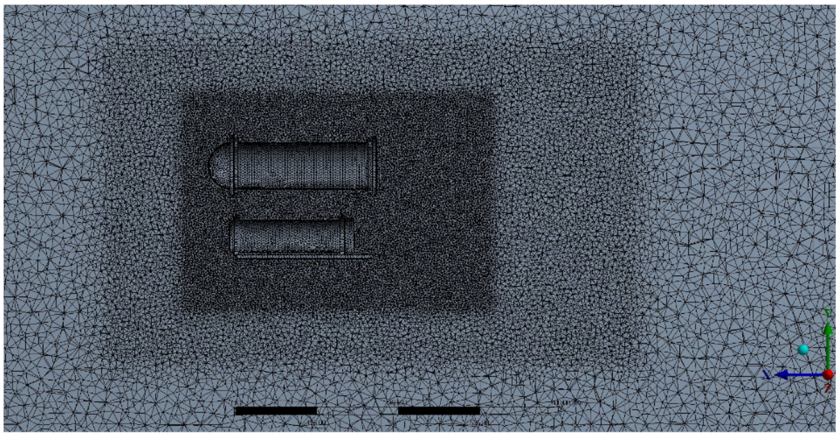


Figure 4.6 Mesh structure side section

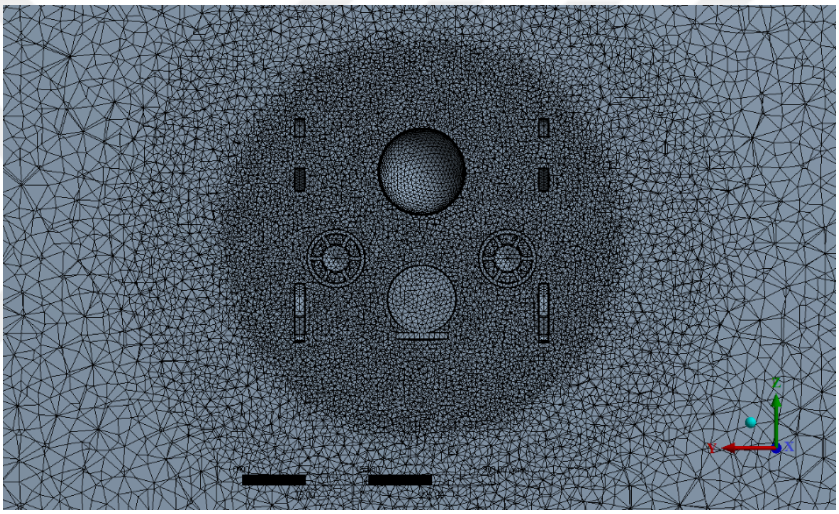


Figure 4.7 Mesh structure front section

Boundary values configured before the initialization of the solver are shown in Table 4.4.

Table 4.4 Boundary conditions of CFD analysis

Boundary Condition	Unit	Boundary Value
Turbulence Model	-	*k-ε*
Flow Domain Dimensions	mm	*1000x1000x4000*
Flow Velocity	m/s	*1.5*
Flow Direction	-	*-z*
AUV Dimensions	mm	*300x280x400*
Mesh Number	M	*2*
Mesh Form	-	*Hexahedral+Quadriteral*
Inlet	-	*Velocity Inlet*
Outlet	-	*Pressure Outlet*
Exterior Domain Surface	-	*Symmetry*

4.1.3.2 Numerical Validation of Scenario 1

Validation is used to determine how accurately the modelled physical process works. The confident use of numerical results is supplied by validation of the numerical models. Considering the numerical accuracy of the applied turbulence method, validation is implemented in the underwater axisymmetric body named as Afterbody 1 (Jagadeesh et al., 2009a). Afterbody 1 is an axisymmetric low drag body with a diameter of 0.14 m and the overall length of this body is 1.4 m. In validation process, the realizable k-ε model is applied against experimental results with similar model domain setup of the current numerical analysis. The numerical results are in good agreement with the experimental values with acceptable errors. Validation results for the flow speed range of 0.4–1.4 m/s are shown in 4.8.

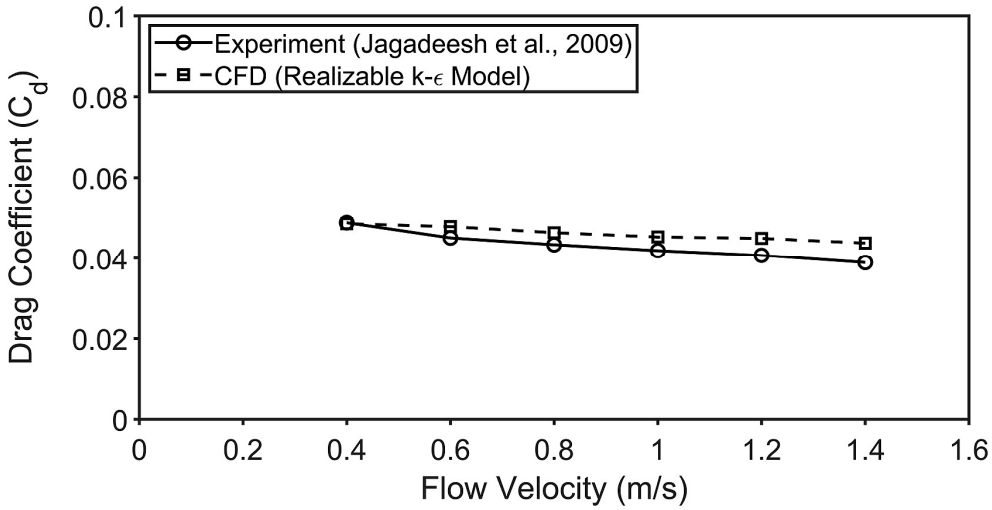


Figure 4.8 Numerical validation results of Scenario 1

4.1.3.3 Mesh Independency of Scenario 1

Within the scope of the analysis, the independence from mesh was calculated within the limits of different sufficient accuracy by taking into account the calculation period. The independence from mesh was realized in three different mesh numbers in case of the 5° roll angle for the steady state phase of the model. The number of mesh was determined as 1M, 2M and 3M (integer) and the structural mesh was generated. The drag coefficient was calculated for three different mesh structure values and the maximum error was found e_3=0.019 which is tolerable as shown in Figure 4.9.

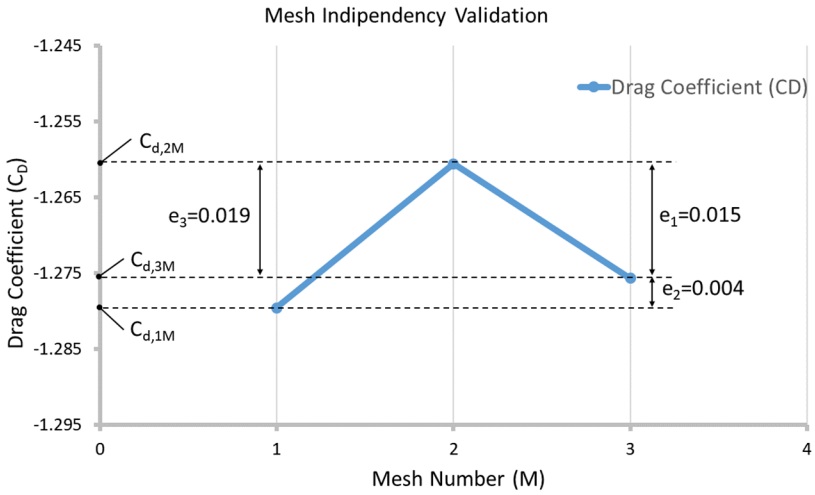


Figure 4.9 Mesh independency control for 1M, 2M and 3M mesh numbers

4.1.4 Results and Discussion of Scenario 1

Corresponding slopes $f'_k(x_k)$, slope angles β_k, location of intersection nodes, linear path distances d_k and yaw angles ψ_k with respect to derivation points p_k were obtained as the output vectors of cubic spline curve fitting algorithm which was computerized and run with JAVA. As seen in Table 4.5, at all intersection nodes z_k, maximum yaw angle value in ψ_k data set resulting from trajectory planning reached up to ψ_{max}=29.56°. For this reason, the hydrodynamic analysis settings were adjusted so that the ϕ and ψ angles were in the range of 0°-30°. Trajectory planning algorithm results are presented in Table 4.5.

Table 4.5 Trajectory planning algorithm results

Derivation Point Coordinates (mm)	Slope (-)	Slope Angle (°)	Intersection Node Coordinates (mm)	Linear Path (mm)	Yaw Angle (°)
$p_k\,(x_{p_k},y_{p_k})$	$f'_k(x_k)$	β_k	$z_k\,(x_{z_k},y_{z_k})$	d_k	ψ_k
p_1 (0,570)	+∞	90	z_1 (0,676)	676	29.54
p_2 (52,767)	1.76	60.46	z_2 (104,859)	211	29.56
p_3 (1000,1396)	0.59	30.9	z_3 (3938,3153)	4465	15.8
p_4 (6000,3710)	0.26	15.1	z_4 (8056,4265)	4266	13.46
p_5 (11000,4349)	0.03	1.64	z_5 (13293,4415)	5236	10.03
p_6 (15000,4163)	-0.14	171.6	z_6 (16937,3877)	3684	13.74
p_7 (19000,3038)	-0.4	157.87	z_7 (20959,2242)	4342	6.5
p_8 (24423,351)	-0.55	151.37	z_8 (24478,321)	4010	17.84
p_9 (24521,275)	-1.05	133.53	z_9 (24563,232)	123	17.2
p_{10} (24591,174)	-2.02	116.33	z_{10} (24631,94)	154	26.3
p_{11} (24631,0)	-∞	270	z_{11} (24631,0)	94	0

Time dependent analysis had been run for angular rate around X_b axis p=0.3 deg/s and around Z_b axis r=0.3 deg/s separately for roll and yaw rotations. Step times had been adjusted to last at 10 seconds on the purpose of acquiring the angle range of 0°-

30º on time domain. Velocity distributions for roll angles of 5º, 10º, 15º, 20º, 25º, 30º around the AUV inside the flow domain are illustrated as shown in Figure 4.10.

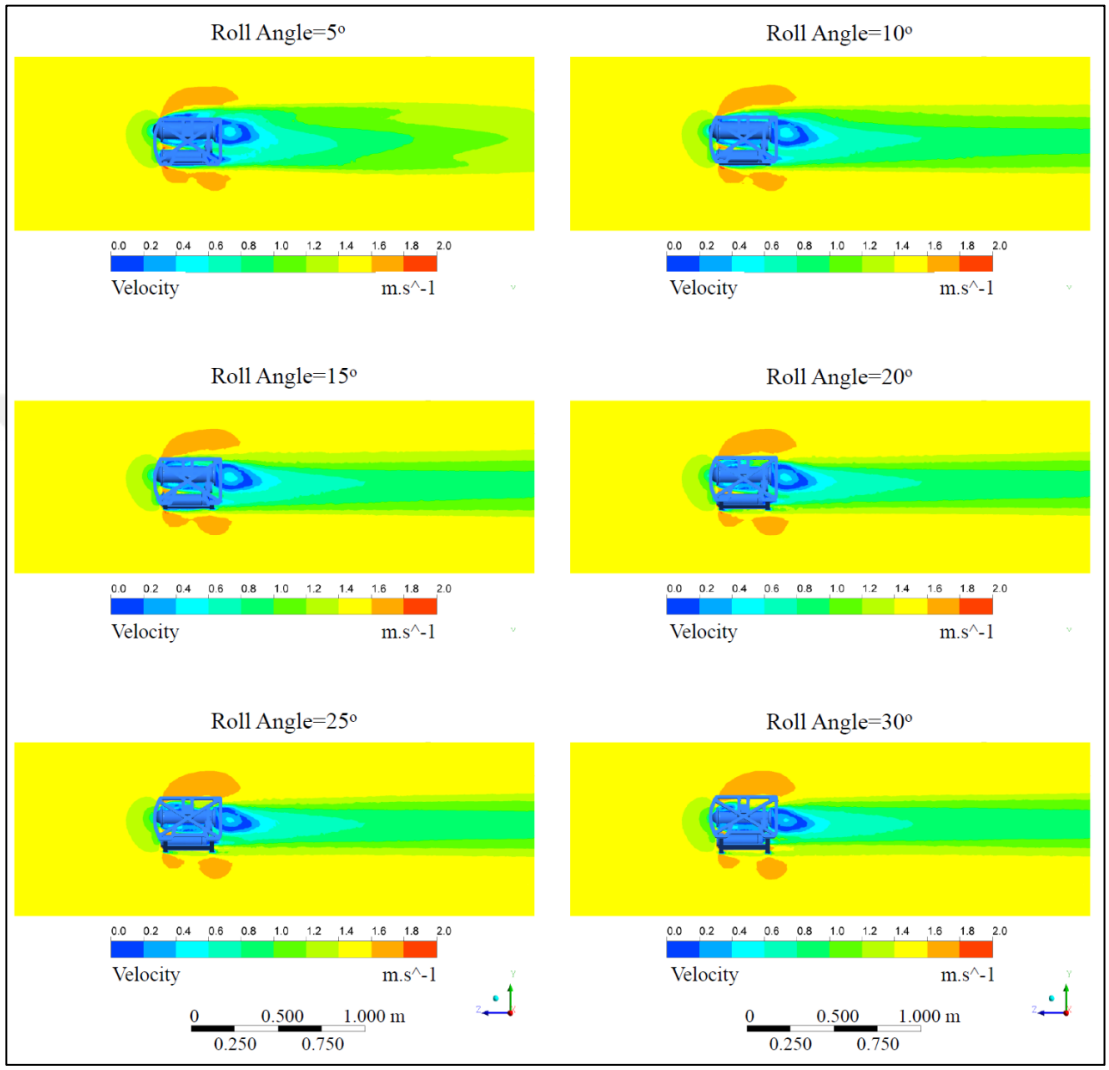


Figure 4.10 Velocity distribution for roll angles of 5º, 10º, 15º, 20º, 25º, 30º

Pressure distributions for roll angles of 5º, 10º, 15º, 20º, 25º, 30º around the AUV inside flow domain are illustrated as shown in Figure 4.11.

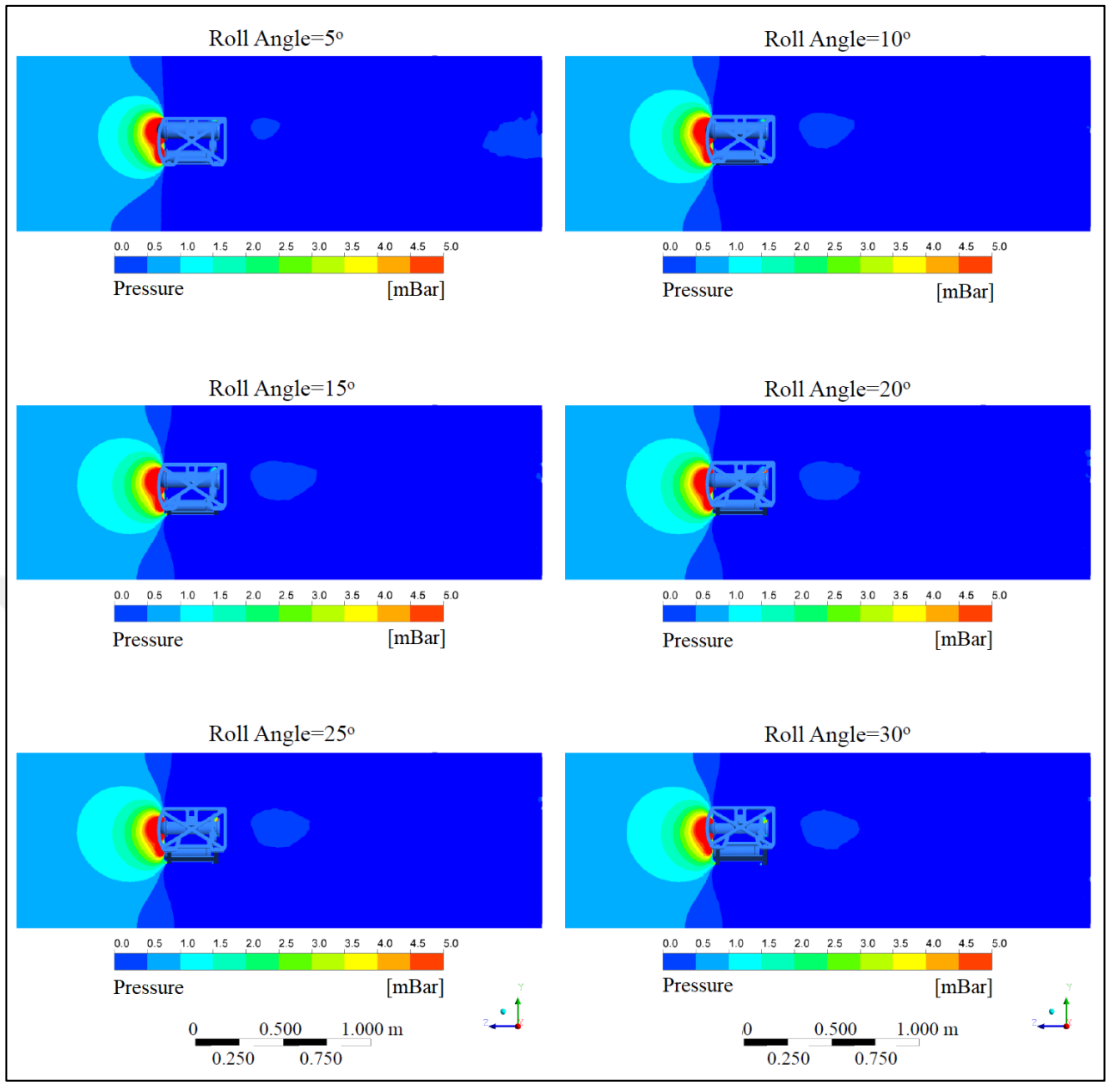


Figure 4.11 Pressure distribution for roll angles of 5°, 10°, 15°, 20°, 25°, 30°

Velocity distributions for yaw angles of 5°, 10°, 15°, 20°, 25°, 30° around the AUV inside flow domain are illustrated as shown in Figure 4.12 below.

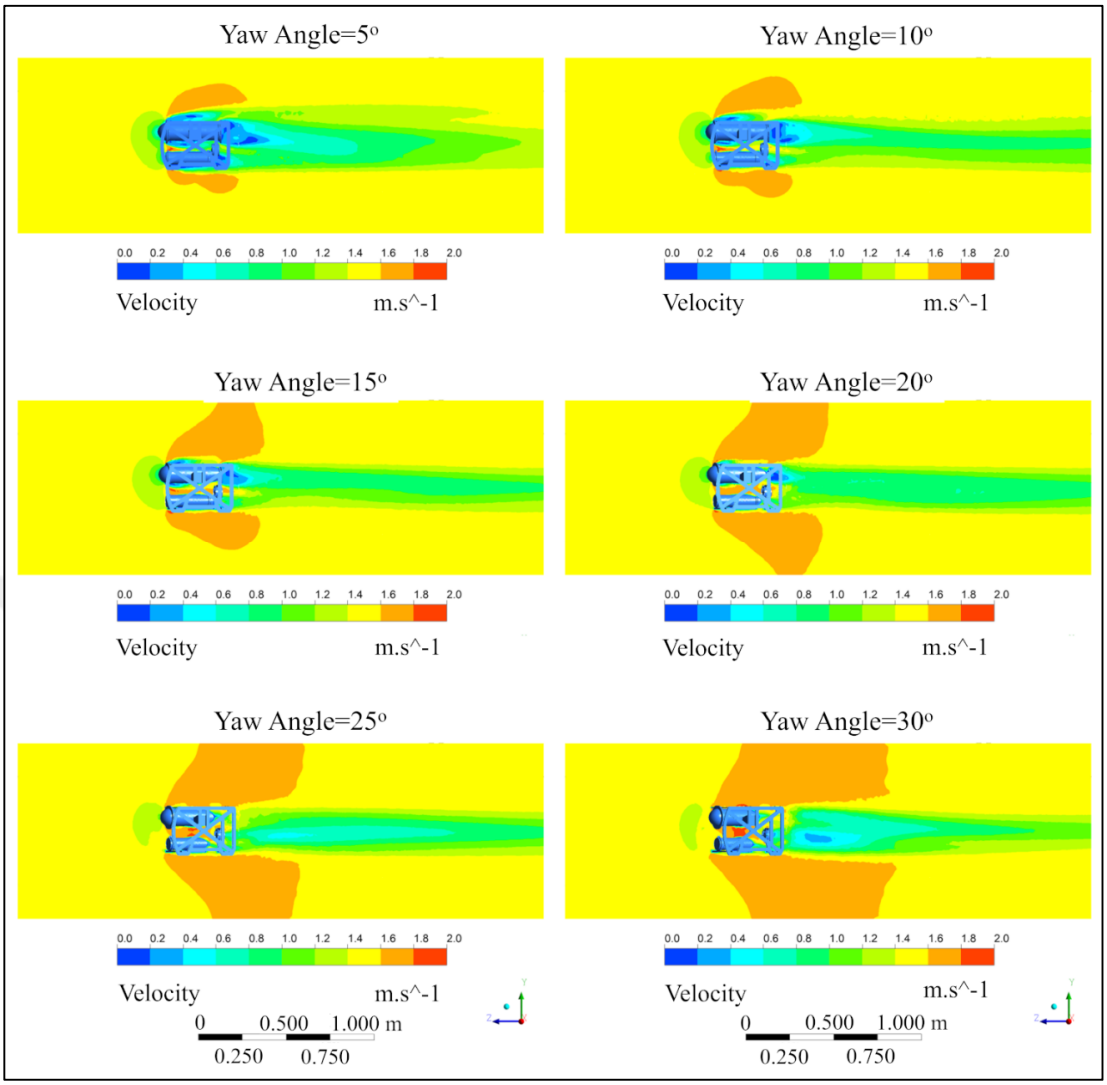


Figure 4.12 Velocity distribution for yaw angles of 5°, 10°, 15°, 20°, 25°, 30°

Pressure distributions for yaw angles of 5°, 10°, 15°, 20°, 25°, 30° around the AUV inside flow domain are illustrated as shown in Figure 4.13. below:

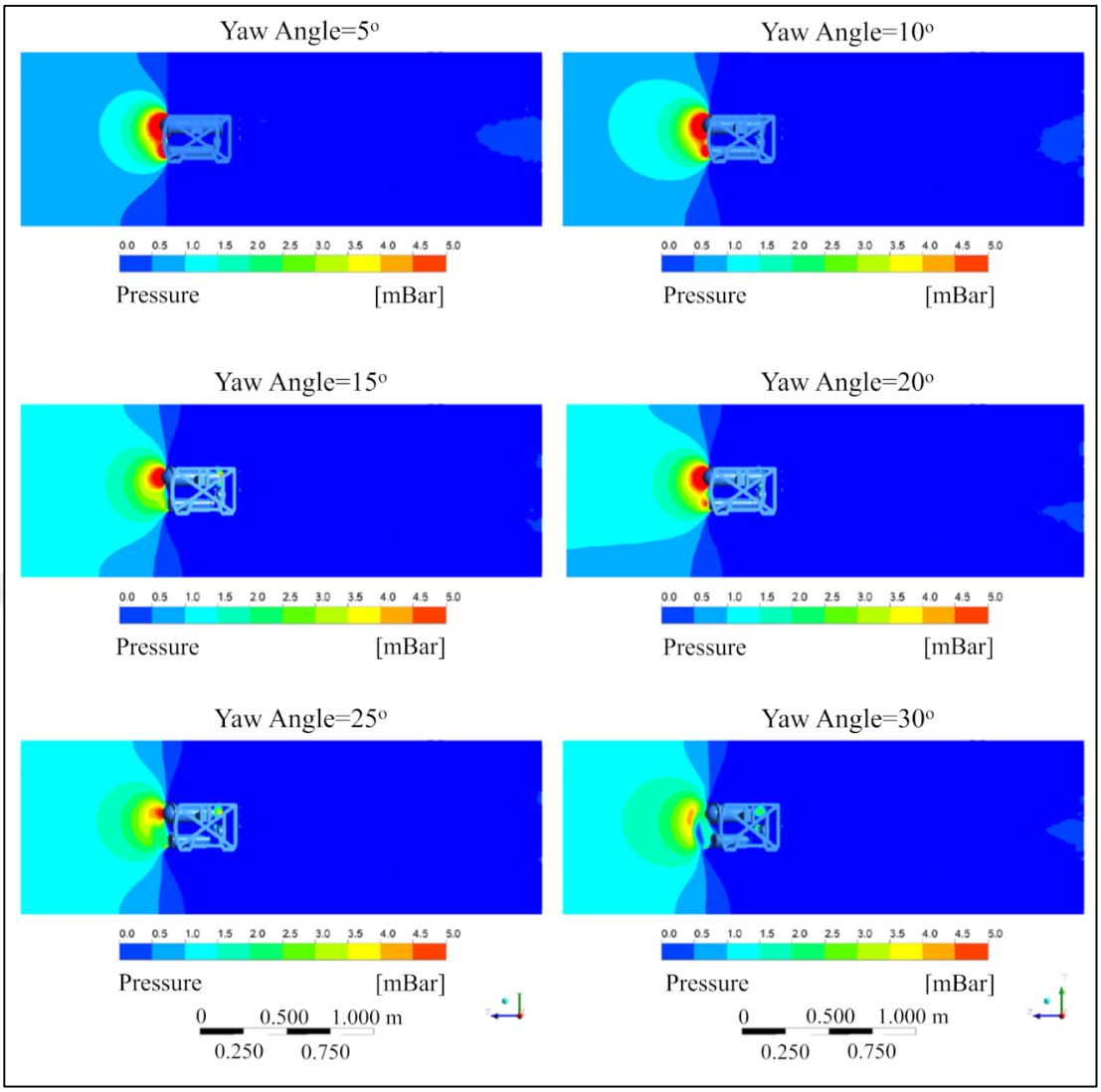


Figure 4.13 Pressure distribution for yaw angles of 5°, 10°, 15°, 20°, 25°, 30°

For roll rotation, time and angular position dependent behavior of hydrodynamic derivatives (drag coefficients, lift coefficients, resulting forces and moments) are given in Figure 4.14.

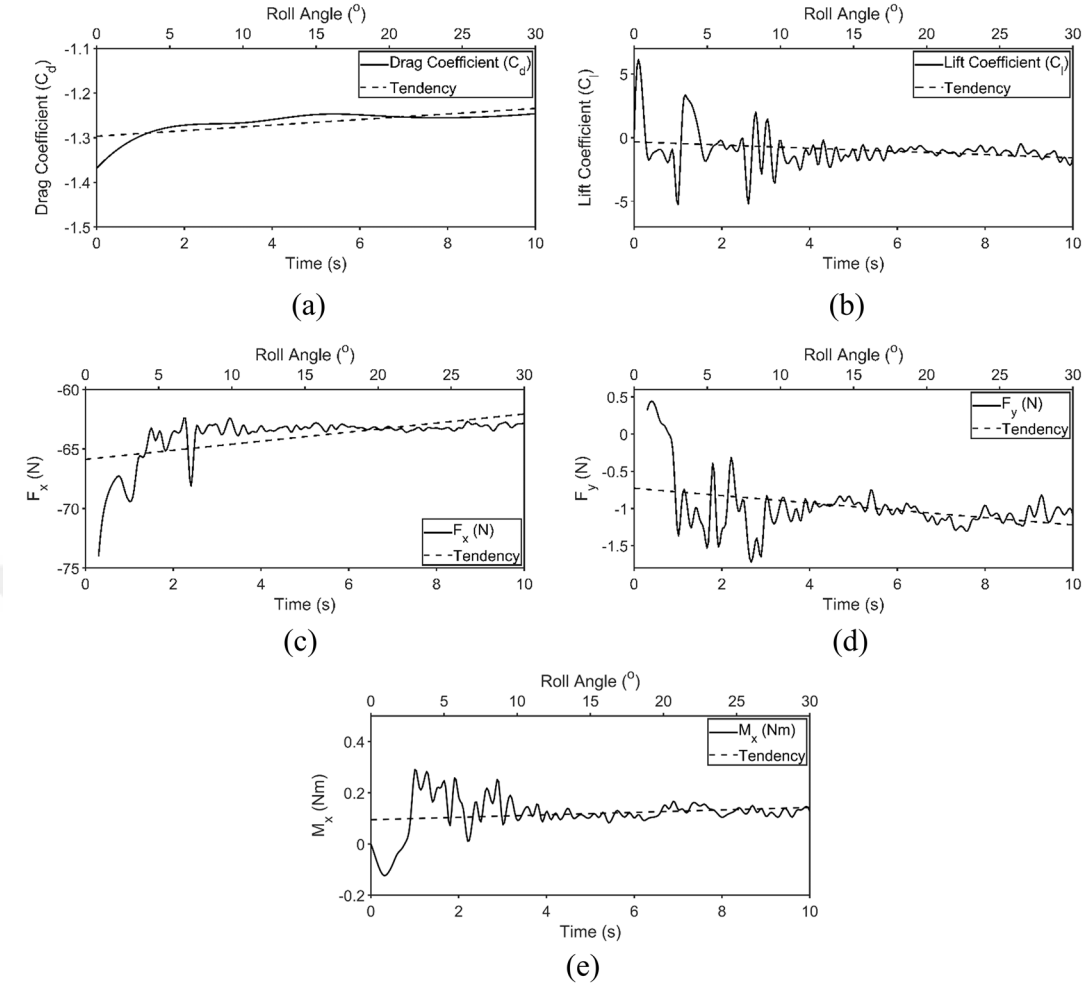


Figure 4.14 Hydrodynamic derivatives on time and roll position domain

For yaw rotation, time and angular position dependent behavior of hydrodynamic derivatives (drag coefficients, lift coefficients, resulting forces and moments) are given in Figure 4.15.

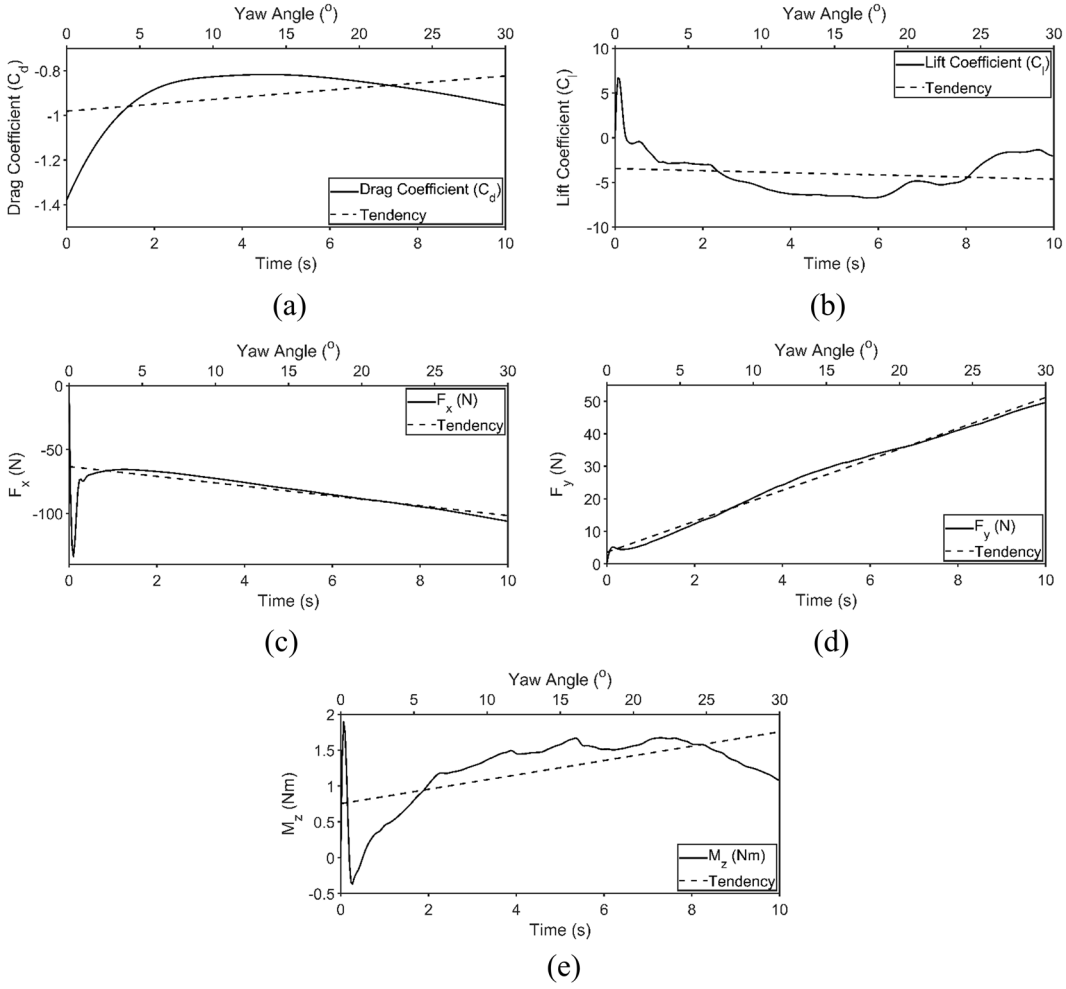


Figure 4.15 Hydrodynamic derivatives on time and yaw position domain

As a result of the flow analysis carried out by the finite volume method, it was observed that drag force F_x, roll moment M_x and C_D values had a slight upward tendency, F_y and C_L value had a slight downward tendency in the change of roll angle between ϕ=0°-30°. When the results are evaluated, the alterations on the hydrodynamic parameters brought about by roll angle variations are negligibly low. This is because the size of the projection area exposed to the flow of the underwater robot is not dependent of the roll angle. While yaw angle was increasing from 0° to 30°, more distinct and significant changes were observed in the hydrodynamic derivatives. With the increase in the yaw angle, C_D value had a slight downward tendency, F_x and F_y rise and C_L changes in the 0-6 bandwidth. M_z increases during the yaw angle is between ψ=0°-15°, remains around 1.5 Nm during ψ=15°-23° and decreases during the during ψ=23°-30°.

Noticeable fluctuations in the early phases of the solution arises from the time dependent operation of the numerical solver. The convergence time to the correct result lasts around 2 seconds. CFD simulation was carried out using a 16 core dual processor unit with a processing capacity of 2.53 GHz. The use of high capacity processors is known to enable more precise solutions by allowing to configure lower step size in the CFD solver setup. Therefore, by using high processing capacity, the initial state values of the time dependent solution can be obtained more accurately. In addition, the simulation can be run in the range of -10° - +30° to eliminate inaccuracies that occur at the beginning phase of the 0°-30° range. In the second half of the study, $F_x(\psi)$ data set obtained from hydrodynamic analysis and d_k-ψ_k angle sets obtained from trajectory planning phase which can be seen in Table 4.5 constitute the input parameters for motion induced energy requirement evaluation algorithm. Energy modelling of the AUV was computerized and run with MATLAB® software. Output results of $F_x(\psi)$, $E_{k_{linear}}$ and $E_{k_{angular}}$ is listed in Table 4.6.

Table 4.6 Energy modelling algorithm results

d_k (mm)	$F_x(\psi=0)$ (N)	$E_{k_{linear}}$ (J)	ψ_k (^{o})	$F_x(\psi)$ (N)	$E_{k_{angular}}$ (J)
d_{k_1}	62.3	*42.11*	ψ_{k_1}	*105*	*16.23*
d_{k_2}	62.3	13.14	ψ_{k_2}	105	16.24
d_{k_3}	62.3	278.16	ψ_{k_3}	81.9	6.77
d_{k_4}	62.3	265.77	ψ_{k_4}	78	5.49
d_k5	62.3	326.20	ψ_{k_5}	72	3.77
d_{k_6}	62.3	229.51	ψ_{k_6}	78.8	5.66
d_{k_7}	62.3	270.50	ψ_{k_7}	67.8	2.30
d_{k_8}	62.3	249.82	ψ_{k_8}	85	7.93
d_{k_9}	62.3	7.66	ψ_{k_9}	84	7.56
$d_{k_{10}}$	62.3	9.59	$\psi_{k_{10}}$	98.8	13.59
$d_{k_{11}}$	62.3	5.85	$\psi_{k_{11}}$	0	0

Here, to avoid picking incorrect value of $F_x(\psi=0)$ inflicted from the initial phase instabilities of the computational solver, the point where the trending line intersects

the vertical axis was used as it can be seen in Figure 4.15. Using the data in Table 4.6, the total energy requirement for navigation from the beginning to the end of the trajectory calculated as E_{total} =1.78 kJ. Considering the time of travel throughout the trajectory Δt=18.17 s, the need of average power is calculated as P_{avg}=98.18 W which is halved by both horizontal thruster as P_t=49.09 W. In the process of development of DEMarine AUV drive system, it is planned to adopt BlueRobotics® T200 thrusters for performing of horizontal plane motions. DEMarine AUV currently supplies its power from 22.2V 10 Ah Lithium Polymer battery. For identified P_t and various operating voltages *V,* corresponding rotation speed, PWM, amperage draw, thrust force and efficiency values are received from BlueRobotics® T200 thruster's performance charts. Using mentioned performance data, calculated discharge time of battery t_{batt} and navigation range l_{nav} is given in Table 4.7.

Table 4.7 Performance evaluation table of DEMarine AUV

Operating Voltage (V)	*PWM* (µs)	*Motor Speed* (rpm)	*Current Draw (A)*	*Power (W)*	*Thrust Force (kgf)*	*Efficiency (g/W)*	*Battery Discharge Time (h)*	*Navigation Range (m)*
10	1244	*1957*	*4.84*	*48.4*	*-1.19*	*24.6*	*1.03*	*5578*
12	1280	1955	*4.1*	49.2	-1.19	24.2	1.21	6585
14	1308	1930	*3.5*	49.0	-1.17	23.8	1.42	7714
16	1328	1903	*3*	48.0	-1.12	23.3	1.67	9000
18	1340	1871	*2.73*	49.1	-1.08	22.1	1.83	9890
20	1344	1816	*2.46*	49.2	-1.02	20.7	2.03	10975

As it can be seen in Table 4.7, according to operating voltages ranging from 10V to 20V, the discharge time of battery t_{batt} and implicitly the duty period of AUV varies from 1.03 h to 2.03 h, the navigation range l_{nav} 5.57 km to 10.97 km and efficiency 20.27 g/W to 24.6 g/W.

4.2 Determination of Hydrodynamic Characteristics: Scenario 2

The main focus of scenario 2 is obtaining thrust capacity, power and energy parameters of an Autonomous Underwater Vehicle (AUV), which follows a logarithmic trajectory segment. In the first phase, the parametric forms of curve equations are identified to represent the relevant trajectory characteristics along which the AUV performs its maneuvering motion. Following this, an algorithm was developed and computerized to acquire curvature and radius of curvature data sets for each point on the trajectory. In the second phase, a flow domain, in which a fixed speed free streamflow of the seawater was simulated in compliance with the AUV trajectory. For various locations of the AUV inside the flow domain, Computational Fluid Dynamics (CFD) analysis, which uses the Finite Volume Method (FVM), was carried out. The hydrodynamic force values due to the radius of curvature obtained from the CFD analysis were used as an input to find the heading and lateral thrust and power requirement of the AUV during the maneuvering motion.

4.2.1 Trajectory Planning of Scenario 2

The designed trajectory geometry is intended to determine the motion control inputs of the AUV following a course with a varying radius of curvature. For this purpose, the approach maneuver to a smart float in Izmir Bay, under constant flow velocity, was modelled using the curves defined by the logarithmic function. The logarithmic model is expressed parametrically as in Equations 4.20 and 4.21.

$$x = e^{0.2t}\cos(t) \tag{4.20}$$

$$y = e^{0.2t}\sin(t) \tag{4.21}$$

In this model, the t parameter, which expresses the time variable, was limited between 0-2π and the ideal trajectory of the underwater robot was determined as in Figure 4.16.

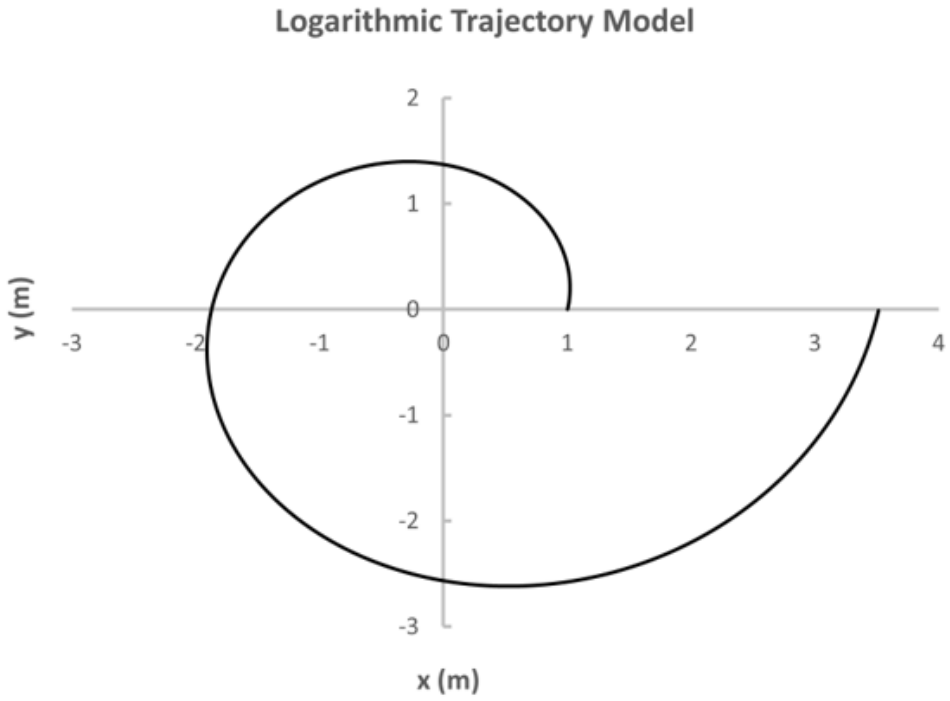


Figure 4.16 Logarithmic trajectory model of the AUV

The vector distances to the center point d_i for n different positions of the AUV were calculated on the generated trajectory model as in Equation 4.22.

$$d_i = \sqrt{(x_i - 1)^2 + y_i^{\,2}} \qquad i = 1...n \tag{4.22}$$

If the average curvature value for any $A_i(x_i, y_i)A_{i+1}(x_{i+1}, y_{i+1})$ arc on the trajectory is expressed as $\Delta\alpha/\Delta s$, the local curvature value at point A_i is calculated as in Equation 4.23 by applying the limit of $\Delta s \rightarrow 0$ in the average curvature expression written for the A_i-A_{i+1} arc.

$$K = \lim_{\Delta s \to 0} \left| \frac{\Delta \alpha}{\Delta s} \right| \tag{4.23}$$

Here $\Delta\alpha$ is the angle between the tangents and Δs is the distance of two consecutive points. Curvature values K_i at each $A_i(x_i, y_i)$ point for the logarithmic trajectory expressed in the parametric form were calculated as in Equation 4.24 (Svirin, 2009).

$$\sum_{i=1}^{n} K = \frac{\left| \frac{dx(t)}{dx_i} \frac{d^2 y(t)}{dy_i^2} - \frac{dy(t)}{dy_i} \frac{d^2 x(t)}{dx_i^2} \right|}{\left[\left(\frac{dx(t)}{dx_i} \right)^2 + \left(\frac{dy(t)}{dy_i} \right)^2 \right]^{3/2}} \tag{4.24}$$

On the trajectory, the radius of curvature value at each point is the inverse of the curvature value based on multiplication and was calculated as in Equation 4.25.

$$R_i = \frac{1}{K_i} \qquad i = 1...n \tag{4.25}$$

By using the formulation specified above, the geometric parameters for the nine different locations determined in $\pi/4$ intervals on the logarithmic trajectory were obtained and presented in Table 4.8.

Table 4.8 Geometric parameters of AUV trajectory

Location Number	**Coordinate** $p_i(x_i, y_i)$	**Vector Distance.** d_i **(m)**	**Curvature.** K_i **(m^{-1})**	**Radius of Curvature.** R_i **(m)**
1	p_1(3.51, 0)	3.51	0.28	3.56
2	P_2(2.12, -2.12)	3	0.33	3.04
3	p_3(0, 2.56)	2.57	0.38	2.60
4	p_4(-1.55, -1.55)	2.19	0.45	2.22
5	p_5(-1.87, 0)	1.87	0.52	1.90
6	p_6(-1.13, 1.13)	1.6	0.61	1.62
7	p_7(0, 1.36)	1.37	0.71	1.39
8	p_8(0.83, 0.83)	1.17	0.84	1.18
9	P_9 (1, 0)	1	0.98	1.01

4.2.2 Energy Modelling of Scenario 2

At this stage of the study, in order to obtain the performance data of the AUV, the thrust and power requirements were determined according to the geometric manoeuvre parameters that appeared at different points on the logarithmic trajectory. As the velocity of current for the analysis, the highest value detected in the Izmir Bay in the

summer period, v_c= 0.158 m/s was chosen (Sayin, 2003). The forces that occur depending on the current and the speed of the AUV are shown in Figure 4.17.

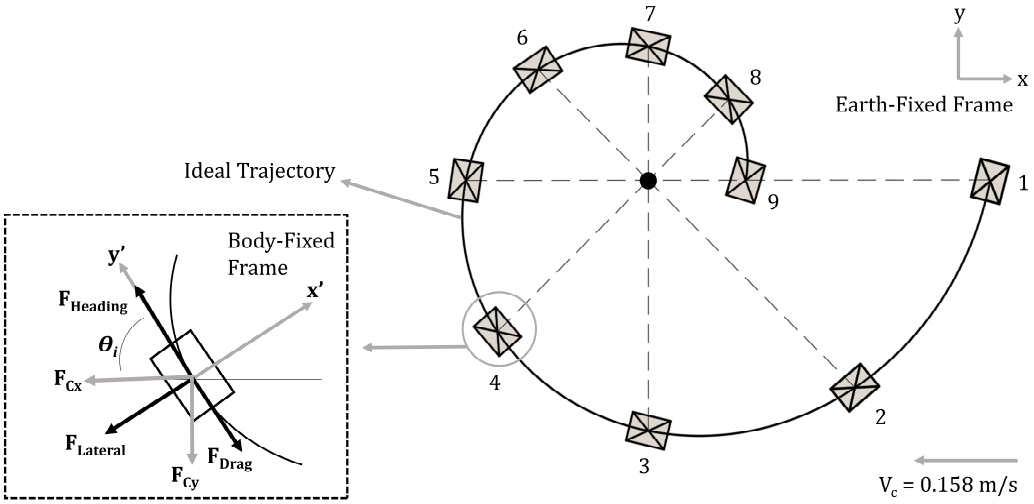


Figure 4.17 Orbital forces acting on the AUV

As seen in figure above, during the course of the AUV, the effect of the current on the movement dynamics varies depending on the orbital geometry. A numerical analysis study was carried out to determine the speed, thrust and power requirement of the AUV along the determined trajectory. The AUV velocity components can be expressed by Equations 4.26 and 4.27 depending on the angle between the current vector and the constantly changing heading vector.

$$V_{R_i(y')} = V_{AUV} \pm V_C \left|\cos\theta_i\right| \tag{4.26}$$

$$V_{R_i(x')} = \pm V_C \left|\sin\theta_i\right| \tag{4.27}$$

Here, V_R refers to the velocity the AUV has due to the current, V_{AUV} is the speed due to the thrust applied, and V_C is the current velocity. As it can be seen, the resultant velocity varies depending on the directions of the AUV and the current. However, to minimise the energy requirement of the AUV, an optimisation study was made depending on the resultant speed. The resultant velocity of the AUV is normalised to

1 m/s, and this velocity is kept constant by the energy delivered from the battery unit. Therefore, if AUV and current are in the same direction, the energy delivered is reduced; otherwise, it is increased. In order to achieve this optimisation, the corrected AUV velocity is calculated in Equation 4.28.

$$V_{COR_i} = V_{AUV} - (V_{R_{i(y')}} - V_{N_i}) \tag{4.28}$$

Here V_{COR_i} and V_{N_i} denote the corrected and normalised velocities, respectively. Thus, in the last case, the total resultant velocity of the AUV without being affected by the current is always maintained as 1 m/s, and the energy drawn from the battery unit is based on the corrected speed of the AUV. Determining the forces on the trajectory expressed in Figure 4.17 requires the force components to be moved from the earth fixed frame to the body fixed frame. The results obtained from the numerical analysis study were determined according to the earth fixed frame. Current-induced drag force components $F_{C_i(x)}$ and $F_{C_i(y)}$ on the AUV were obtained according to the earth fixed frame. However, the resistance components caused by AUV's velocity are identified over the body fixed frame. According to the body fixed frame, the *x'* and *y'* components of the total drag force exposed by the AUV are calculated by Equations 4.29 and 4.30.

$$F_{T_{i(x')}} = F_{C_{i(x)}} \sin\theta_i \pm F_{C_{i(y)}} \cos\theta_i \tag{4.29}$$

$$F_{T_{i(y')}} = F_{AUV} \pm (F_{C_{i(x)}} \cos\theta_i \pm F_{C_{i(y)}} \sin\theta_i) \tag{4.30}$$

The force and velocity components that come out due to numerical and vector analysis make it possible to calculate the power requirement. The need for power consists of two components. The first one meets the navigation in the orbital direction, and the other one is the component necessary to resist the lateral side-slip due to the current. Total power requirement is calculated as in Equation 4.31 using total drag force and corrected velocities.

$$P_{TOTAL\ i} = F_{T_i(y')} V_{COR\ i} + F_{T_i(x')} V_{R_i(x')} \tag{4.31}$$

4.2.3 Computational Study of Scenario 2

CFD simulation was performed for DEMarine AUV, the technical specifications of which were given in the previous scenario, and the numerical results were calculated with ANSYS FLUENT 19.1 commercial software in the same way.

4.2.3.1 Meshing and Boundary Conditions of Scenario 2

The uniform flow domain was generated with a velocity inlet and atmospheric pressure outlet as demonstrated in Figure 4.18.

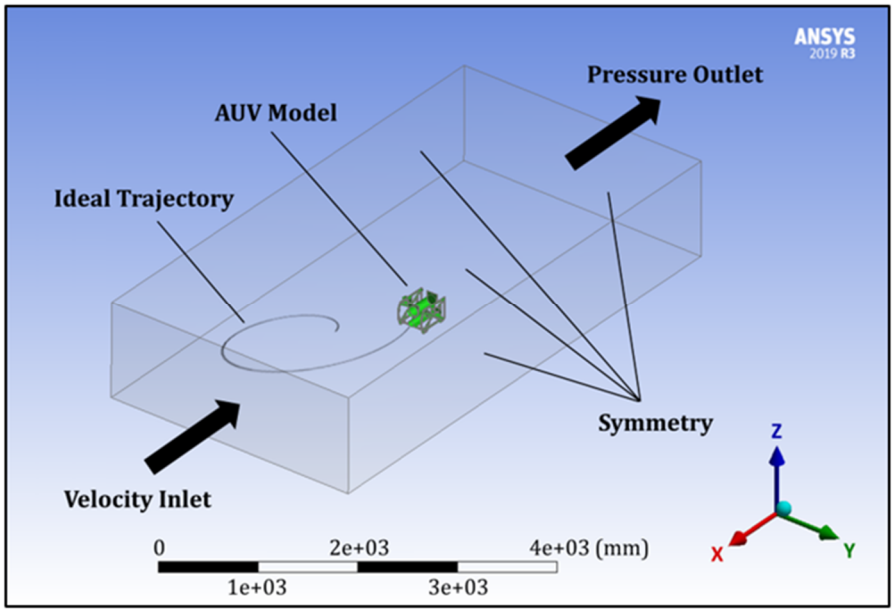


Figure 4.18 Flow domain setup of scenario 2

Structural and unstructured meshes were applied to the computational domain. The mesh structure is shown in detail in Figure 4.19. As it can be seen, while mesh density is more refined in the distant regions, it is more concentrated near the model.

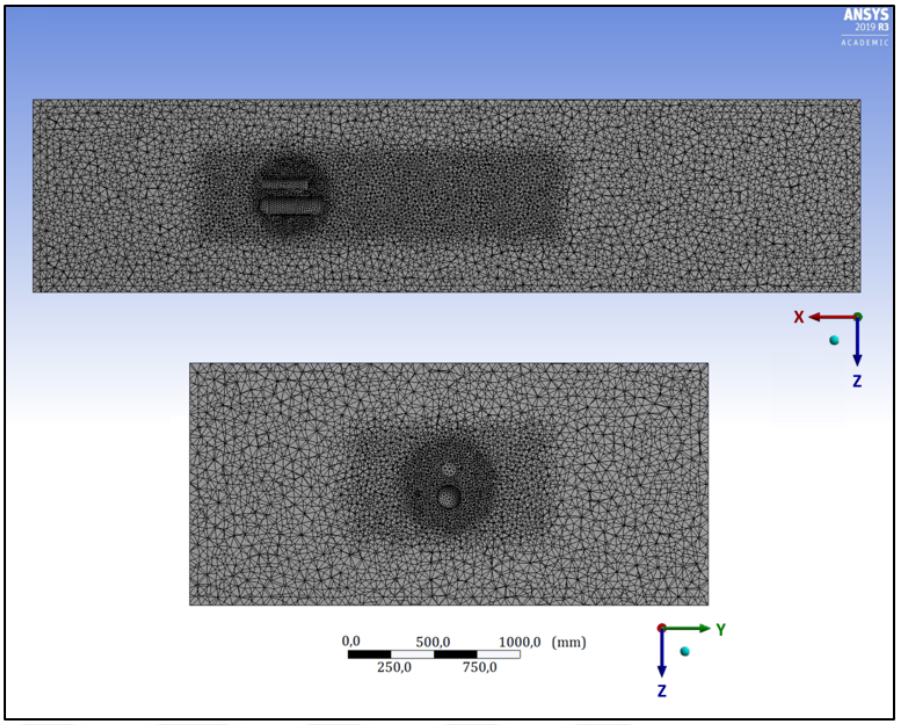


Figure 4.19 Mesh structure of the flow domain

The CFD simulation model was created and executed for each position within the boundary conditions by designating hydrodynamic analysis points at $\pi/4$ intervals on the orbit. Numerical analysis boundary conditions are presented in Table 4.9

Table 4.9 Numerical analysis boundary conditions

Boundary Condition	**Unit**	**Value/Type**
Turbulence Model	-	k-ε standard
AUV Velocity	m/s	1
Velocity of Current	m/s	0.158
AUV Dimensions	mm	300x280x400
Flow Domain Dimensions	mm	4800x1120x2400
Number of Mesh	M	1.4
Mesh Geometry	-	Quadrilateral
Inlet	-	Velocity Inlet
Outlet	-	Pressure Outlet
External Domain Surface	-	Symmetry

4.2.3.2 Numerical Validation of Scenario 2

In the boundary conditions described above, numerical analysis was performed for the underwater vehicle, which has experimental data from the literature, to verify that the k- ε standard turbulence model is suitable for our model. There are various types of reference hull forms providing experimental results available for numerical validation (Jagadeesh et al., 2009b). The model is a parametrically derived series modelled by Hou, Liang, & Mu (2018) and called the Myring-type body. The axisymmetric Myring-type body with 0.28 m diameter and 1.521 m overall length has a nose shape of No.1 and tail shape of No.3. The modelled geometry was compared with the results of the experimental force analysis conducted in the range of 0.3-1.5 m/s from the study of Gao, Wang, Pang, & Cao (2016). It is observed that the results obtained from the analysis studies conducted with the k-ε standard model are compatible in the acceptable error range. Numerical validation results for 0.3- 1.5 m/s are given in Figure 4.20.

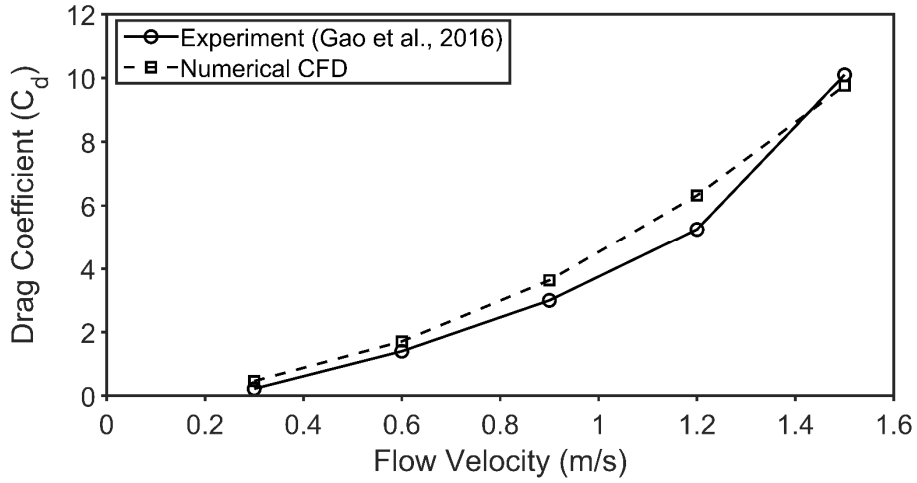


Figure 4.20 Numerical validation results of Scenario 2

4.2.3.3 Mesh Independency of Scenario 2

One of the methods used to ensure the accuracy of the model in numerical analysis studies is the analysis of mesh independence. These analyses are carried out under the

same boundary conditions with different numbers of elements. In the study, drag coefficient C_D values for the AUV moving at 1 m/s velocity without current were compared with the analyses made for the five different element numbers. In the analysis carried out for 532 k, 826 k, 1 M, 1.4 M and 2 M element numbers, the difference between the C_D values obtained for the 1 M, 1.4 M and 2 M element numbers are found to be at acceptable levels. Therefore, the parametric analyses were carried out for the 1.4 M number of elements. The result of mesh independency analysis is shown in Figure 4.21.

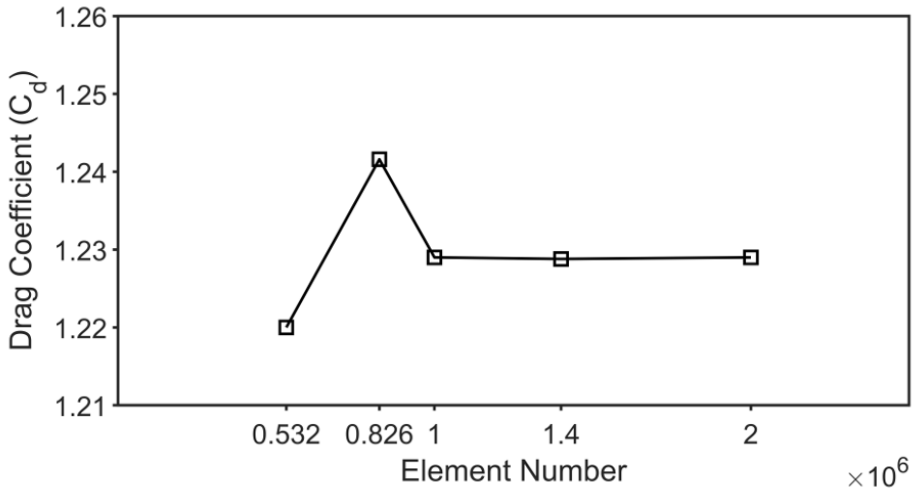


Figure 4.21 Mesh independency analysis results

4.2.4 Results and Discussion of Scenario 2

On the logarithmic trajectory shown in Figure 4.17, the position and velocity values obtained for nine different locations determined in $\pi/4$ radian intervals are calculated in the vector domain as presented in Table 4.10.

Table 4.10 AUV resultant velocity and corrected velocity data sets

Analysis Number	Orientation Angle, θ (°)	AUV Velocity, v_{auv} (m/s)	Flow Stream Velocity, v_c (m/s)	Resultant Velocity, v_b (m/s)	Corrected Velocity, v_c (m/s)
1	101.66	1	0.158	1.032	0.968
2	146.15	1	0.158	1.131	0.868
3	191.29	1	0.158	1.155	0.845
4	236.3	1	0.158	1.088	0.912
5	281.2	1	0.158	0.969	1.030
6	326.31	1	0.158	0.868	1.131
7	11.27	1	0.158	0.845	1.154
8	56.26	1	0.158	0.912	1.087
9	101.14	1	0.158	0.969	1.030

As the result of the hydrodynamic analysis, pressure/velocity contours and flow streamlines were obtained for each position of the AUV on the trajectory and plotted as in Figure 4.22 and Figure 4.23.

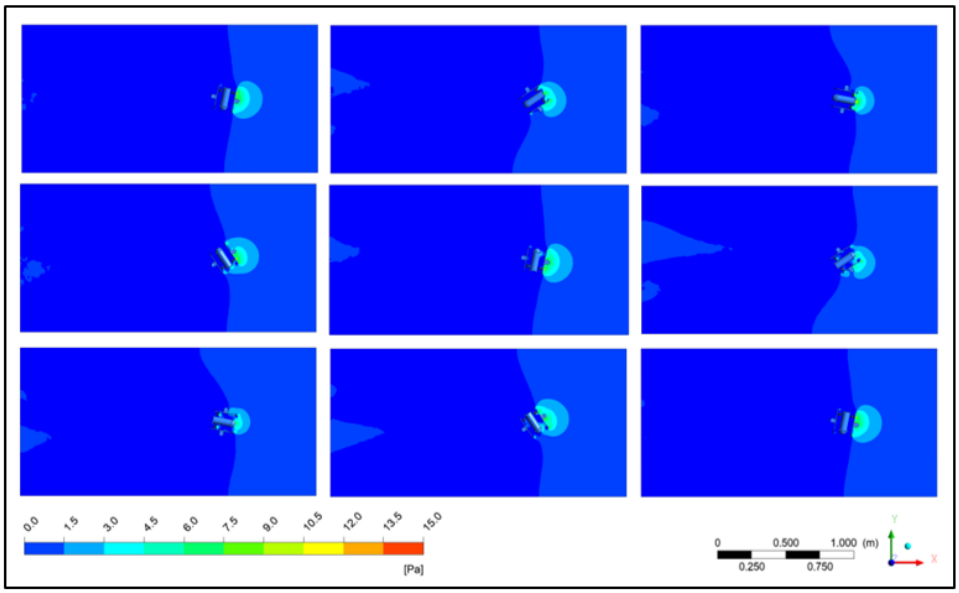


Figure 4.22 Pressure distribution for 9 different angular positions

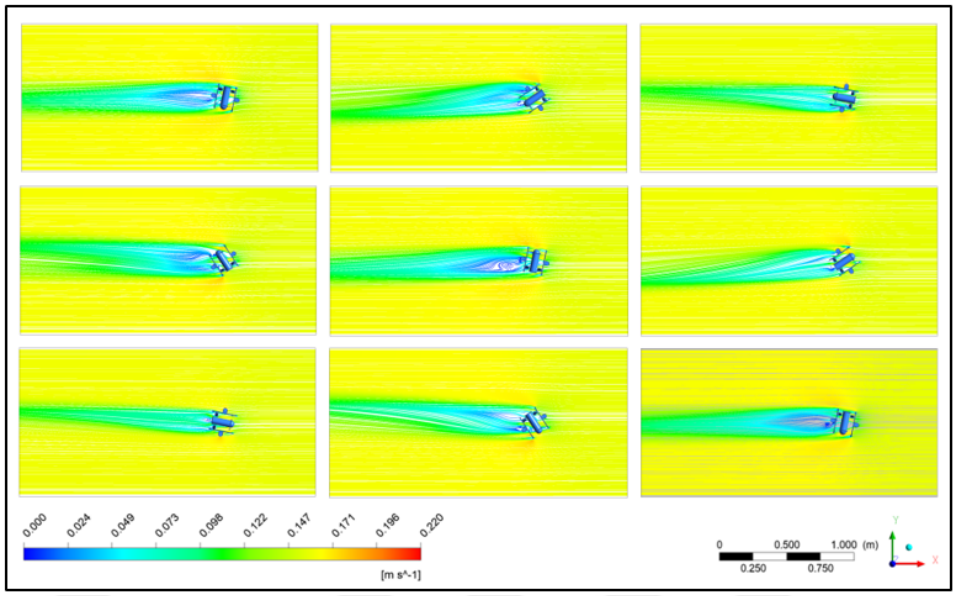


Figure 4.23 Velocity distribution for 9 different angular positions

The CFD analysis was carried out by adhering to the boundary conditions set in Table 4.9 and the forces resulting from each analysis were obtained. In this way, the heading and lateral thrust capacity that the AUV should have to not deviate from its path is given in Table 4.11.

Table 4.11 AUV Thrust and power parameters

Analysis Number	**Radius of Curvature R_i (m)**	**Current-Induced Drag Force, F_{c_i} (N)**	**Heading Thrust, $F_{T_{i(y')}}$ (N)**	**Heading Power, $P_{Heading_i}$ (W)**	**Lateral Thrust, $F_{T_{i(x')}}$ (N)**	**Lateral Power, $P_{Lateral_i}$ (W)**	**Total Power Demand, P_{Total_i} (W)**
1	3.56	0.215	31.218	30.221	1.554	0.221	30.443
2	3.04	0.898	30.535	26.529	1.128	0.177	26.706
3	2.60	0.955	30.478	25.755	-0.412	0.0221	25.777
4	2.22	0.589	30.844	28.140	-1.487	0.147	28.288
5	1.90	-0.091	31.525	32.493	-1.555	0.245	32.739
6	1.62	-0.739	32.173	36.403	-1.203	0.076	36.479
7	1.39	-0.962	32.396	37.416	-0.471	0.071	37.488
8	1.18	-0.474	31.908	34.7086	-1.566	0.070	34.779
9	1.01	0.201	31.232	32.185	-1.555	0.140	32.326

Table 4.11 also presents the heading, lateral and total power requirements while the AUV follows the logarithmic trajectory at a constant speed. Total power requirement,

P_{Total} with respect to the radius of curvature, R and orientation angle, Θ has been demonstrated in Figure 4.24.

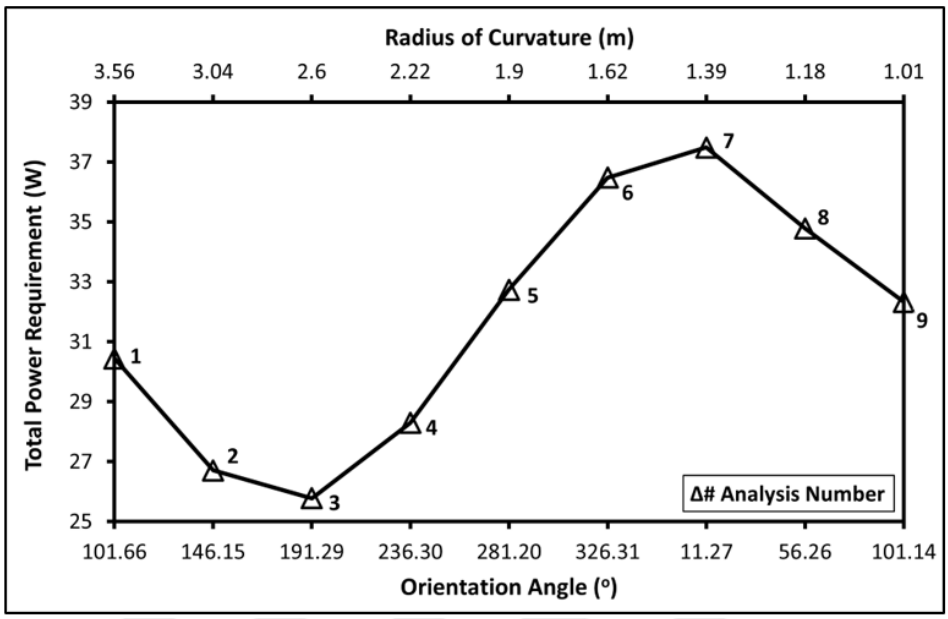


Figure 4.24 Total power with respect to radius of curvature and orientation

When the results are examined, the lowest power requirement appeared in analysis number 3 as 25.77 W. This analysis was carried out with the radius of curvature of 2.6 m and the orientation angle of 191.29°. In analysis number 7 carried out for the position where the radius of curvature is 1.39 m and the angle of orientation is 11.27°, the power requirement has emerged as 37.48 W as the maximum value. These results confirm the hypothesis that the energy demand defined and formulated in section 4.2.2.2 depends on the angular proximity between the AUV orientation and the current direction. The results obtained from the study can be interpreted as follows.

The numerical analysis study was verified by experimental studies obtained from the literature and the results are found to be consistent. In addition, the study was validated by the mesh independence analysis.

In this study, angles were determined according to the current velocity vector for nine different situations taken along the trajectory followed by the AUV, and thus the

power requirement to navigate at the desired speed depending on its velocity and the forces caused by the current was revealed.

The thrust and power requirement of the AUV depends on two different conditions; the first one is the need for the AUV to maintain its course due to the change of the radius of curvature of the trajectory and the other one is that the current has a positive or negative effect on the movement relative to the angular position of the AUV heading vector with the current vector.

The decrease in the radius of curvature has increased the energy requirement. If positions 1 and 9 with the same orientation angle are examined, it can be seen that the total power requirement increases with the decrease of the radius of curvature.

Just before position 3 of the AUV on the orbit, the local tangent of the logarithmic curve coincides with the direction of the stream vector and is in the same direction. So here AUV has a minimum power requirement. Near the 7th position, the local tangent of the logarithmic curve is coincident with the direction of the current vector and is opposite. Therefore, in this position, the AUV needs maximum power to navigate at the desired speed.

CHAPTER FIVE
CONTROL SYSTEM DESIGN

While developing the DEMarine prototype, a remote-controlled system operating with open-loop logic was developed first. This physically implemented system has been updated and finalized many times as mechanical, propulsion, electronic, energy and sensor capacity. During the process of finalizing the physical system, data such as hydrodynamic coefficients, center of mass, buoyancy center, center of volume, final position of thrust axes, trajectory to be covered, energy requirement, thrust capacity, maximum cruise time and average mission speed were recorded. All these obtained data were transferred to a feedback AUV simulation, which includes a model whose inputs and outputs are determined, and various controllers that control this model.

5.1 Scenario 1: Reference Trajectory Input and PID Control

5.1.1 Simulation Model

The main frame of the simulation was built on a structure provided by Mathworks (MathWorks Student Competitions Team, 2022), and DEMarine's pre-recorded physical (moment arm distances, body size measurements), hydrodynamic and orbital parameters were used to transform this model. The ultimate aim of the simulation model application was to determine the most suitable controller for the underwater robot, which is expected to operate under nonlinear effects. The simulation model has modes that can be switched in line with various mission requirements. These modes can be set in different combinations such as position or speed control, low or high fidelity sensing, and operation with or without noise conditions, and can be applied to 6 DoF AUV models. The AUV simulation model has been given in Figure 5.1.

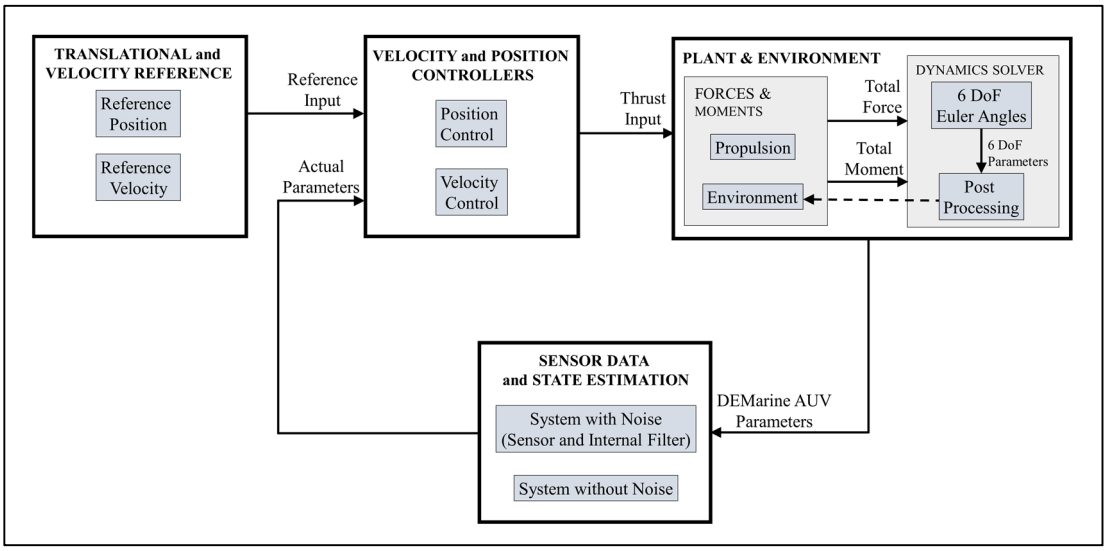


Figure 5.1 AUV simulation model

The model whose flow chart is shown above requires Aerospace and Control System Toolboxes and Aerospace Blockset to operate. Vehicle dynamics, thruster dynamics and propulsion of the DEMarine prototype were transferred to the model using the Aerospace Blockset. Motion, mission and environment analyzes were performed by using the mathematical operations, coordinate systems, spatial transformations and environment models provided by Aerospace Toolbox. Control System Toolbox tools were used to implement and systematically analyze the linear controller design carried out at the beginning of the control study. By using transfer function and frequency response based tuning methods, compensator parameters were adjusted considering criteria such as reference tracking, rise time and stability margin. The first stage of the simulation-based control architecture development process was the commissioning of the PID, one of the principal controllers, for the X, Y and Z axes, and fine-tuning the initial state control parameters, taking into account the criteria for achieving the desired system responses.

5.1.2 Control System Inputs and Boundary Conditions

In order to make the simulation model operational in line with the mission and parameters of the DEMarine AUV, the first step was to turn the translational reference information into an input source in the Simulink and to obtain the reference position

in the spatial environment. For this purpose, the X and Y coordinate data of the curvilinear trajectory planned for the hydrodynamic analysis studies in Chapter 4 were converted into a repeating sequence stair, a series of step inputs were created and given to the controllers as it can be seen in Figure 5.2.

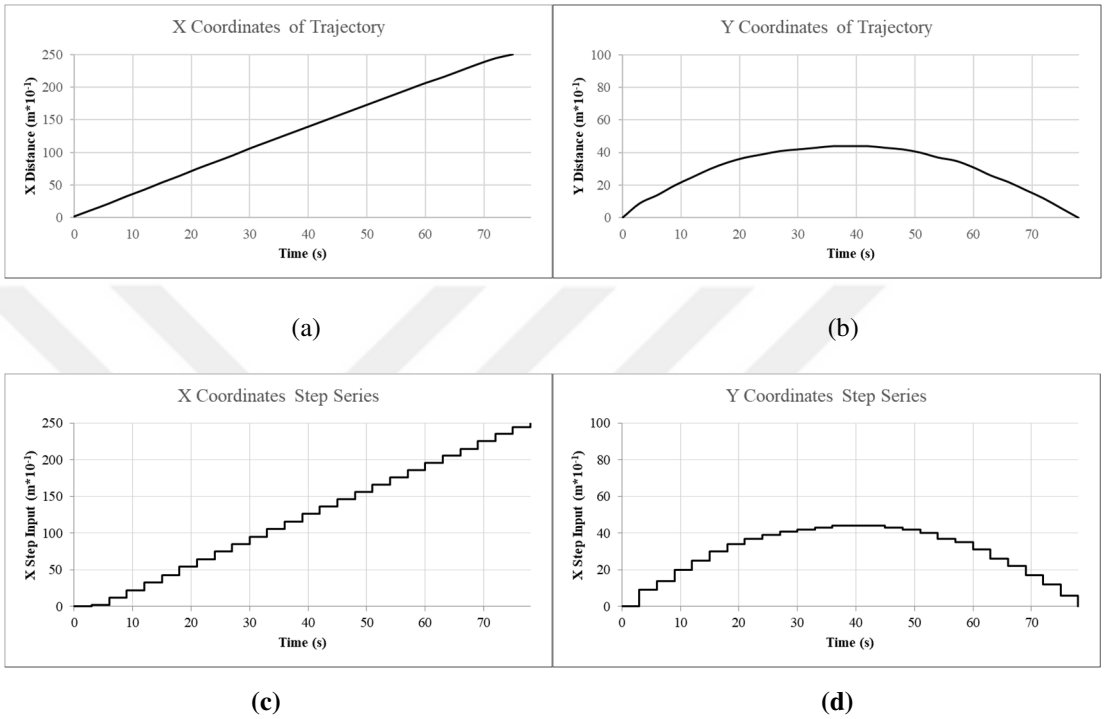


Figure 5.2 Translational reference (a) X coordinates (b) Y coordinates (c) X step input (d) Y step input

These step input series, which are defined separately for the X and Y axes, are mapped to form the reference position data in the horizontal plane. The displacement in the Z axis was planned as a 1-meter heave movement after traversing the horizontal trajectory at each level. For this purpose, the translational reference on the Z axis was set as a 1-meter single step input. The generated reference input is subtracted with the position information coming from the plant output, and the error signal to be transmitted to the controller is obtained. Position errors calculated separately for the X, Y and Z axes are processed by the dedicated PID controller blocks for the regulation of each position component in these three axes and converted into thrust inputs that trigger the actuators.

With a focus of developing the proper control architecture, system identification process on real-life thruster data has been initialized. Having an accurate model enables to design and test a controller as well as model a larger system. The Matlab System Identification Toolbox enables to perform all stages of modeling such as importing and preprocessing the data, trying out different model structures, and evaluating the resulting models. With the project support of the BAP unit, the latest prototype of DEMarine AUV was equipped with T200 model thrusters of Blue-Robotics. The input and output datasets for a T200 Thruster are provided by the manufacturer (*T200 Thrusters*, n.d.). The open-loop test setup of thrusters is shown in Figure 5.3.

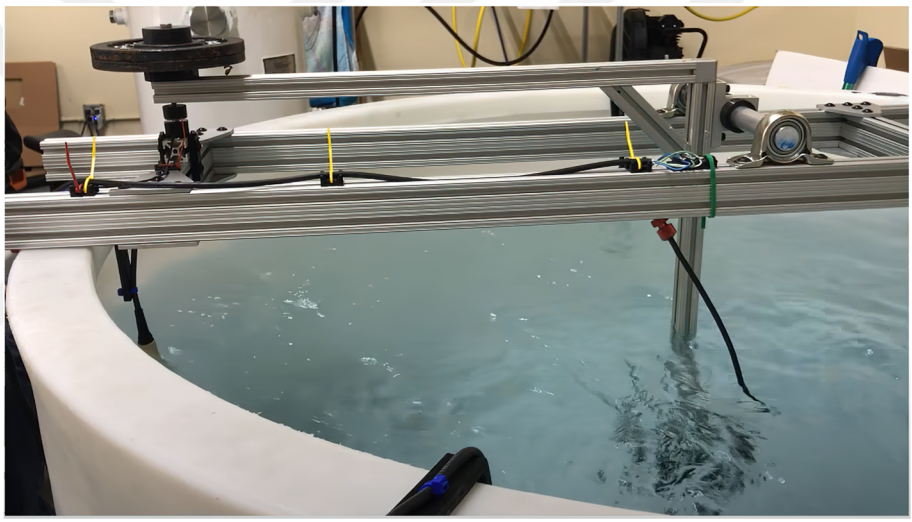


Figure 5.3 Open-loop test setup of T200 thrusters (MathWorks Student Competitions Team, 2023)

With the results of open-loop tests, it is possible to use the input-output dataset in modeling process. In order to obtain a suitable model for Thruster, two separate data sets are required. One is used for creating the model and the other is used for the validation. T200 manufacturer provides two inputs as sine-sweep and square-sweep. System identification toolbox needs to define the input-output data and model structure to be capable of creating the model. While defining the system, first data is collected, then preprocessing is done according to the data. These are two inputs, validation data and prediction data. One of the inputs is used as a sine and the other as an input square, the first is to build the model, and the second is to verify how well the model works.

Using the estimation data and the model, it can be seen how well the model works in the system identification toolbox environment. It can be observed how well it works by increasing the compatibility. System identification workflow model is given in Figure 5.4.

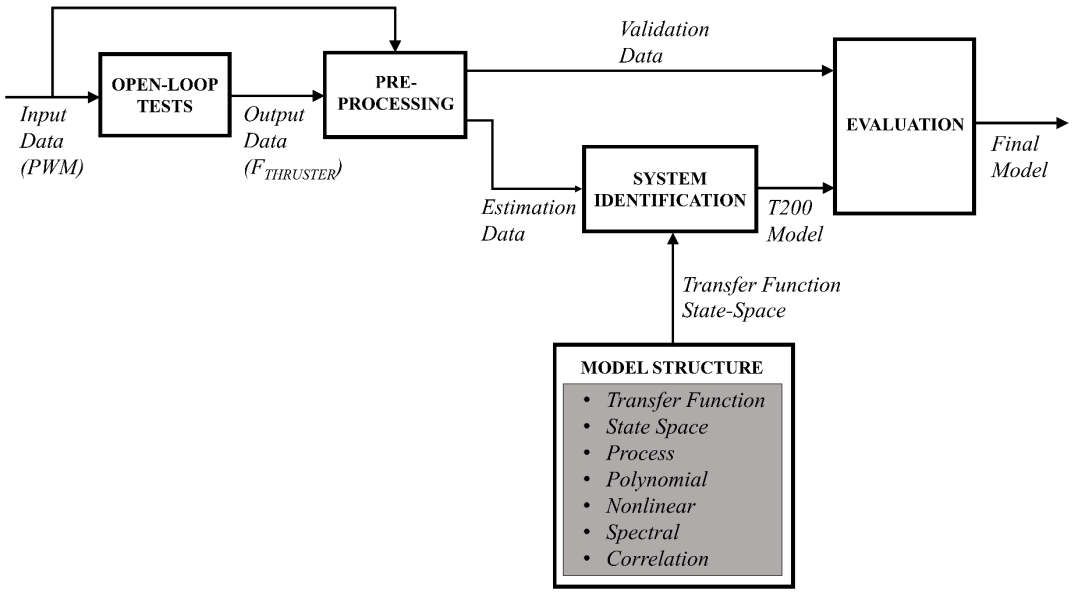


Figure 5.4 System identification workflow of T200 thrusters

In line with the given workflow, analysis was performed in the input range of 0-10 Hz and 1600-1900 μs. Mean filter is used to remove noise in square and sine data. For a model with four poles and two zeros, the results were obtained and compared with the transfer function and state-space model options. The obtained fitting ratio values are given in Table 5.1

Table 5.1. Results of system identification study

Estimation Model	Estimation Method	Fitting Ratio [%]
State Space	Subspace N4SID	91.3
Transfer Function	Continuous Time	91.44

As a result, the transfer function for T200 thrusters, whose characteristic polynomial is of the fourth order, is determined as in Equation 5.1.

$$T(s)=\frac{-0.107s^2+8.358\times10^4 s+3.989\times10^4}{s^4+152.9s^3+3.036\times10^4 s^2+2.93\times10^6 s+1.157\times10^6} \tag{5.1}$$

The block diagram abstraction of PID controller structure is represented by Figure 5.5.

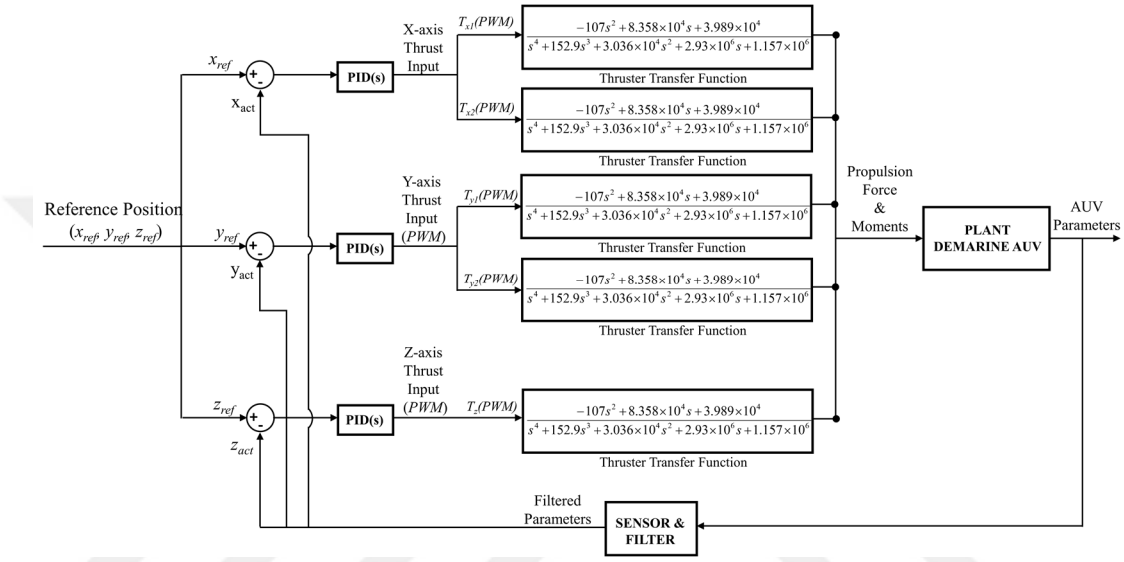


Figure 5.5 Feedback structure with PID controllers (MathWorks Student Competitions Team, 2022)

While examining the control performance, the simulation model was run for t=78 seconds by applying the default and optimized (based on transfer function PID tuning) K_P, K_I and K_D coefficients for every PID tuner given in Table 5.2.

Table 5.2 PID controller parameters

	Axis	Filter Coefficient [N]	Desired Response Time [s]	Transient Behavior Coefficient	K_P	K_I	K_D
Default Gain Values	X	260.06	1.818	0.6	0.035	4.67e-4	0.59
	Y	438.62	1.359	0.6	0.172	3.9e-3	1.72
	Z	205.90	-	-	-0.044	-4.63e-4	-0.93
Optimized Gain Values	X	397.61	0.5748	0.738	0.415	0.038	0.90
	Y	517.94	0.4413	0.765	1.182	0.134	2.01
	Z	205.90	-	-	-1.32	-4.63e-4	-0.93×5

During the adjustment of the PID controller gains, the desired step input system responses are shown in the time and amplitude axes as in Figure 5.6.

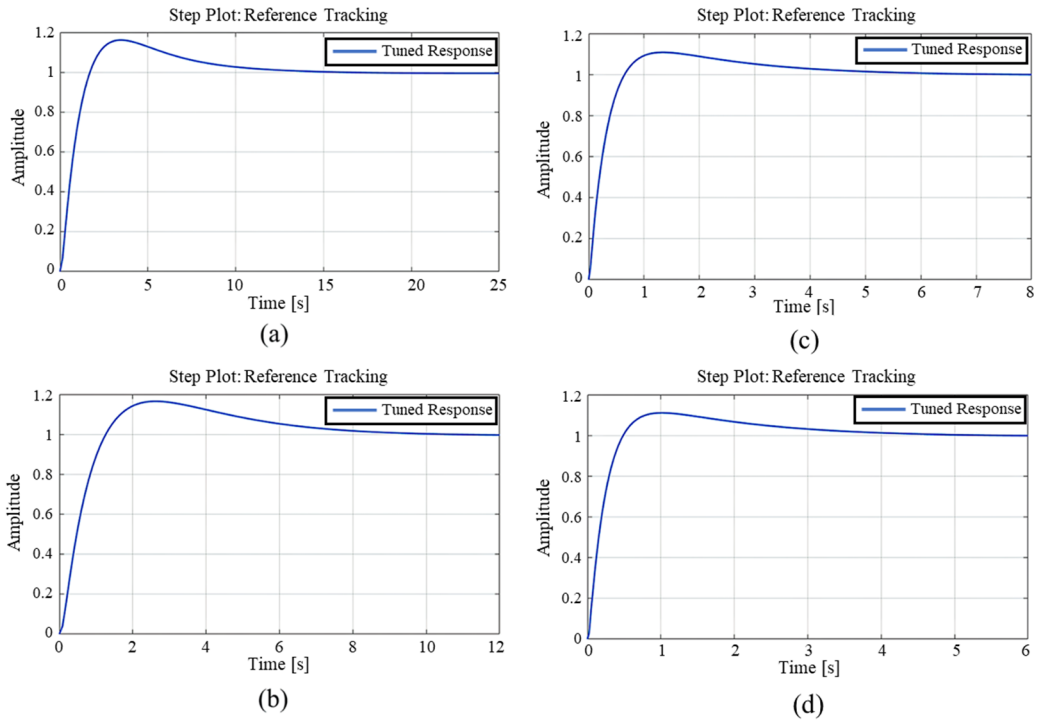


Figure 5.6 Step input responses; (a) and (b): X and Y axes with default coefficients, (d) and (e): X and Y axes with optimized coefficients

At this stage, it has been determined that the phase margin for the Z-axis controller is below 45° and the gain margin is below 0 dB, so the controller is unstable. For this reason, in the PID tuning algorithm, the phase margin was set as 90° and the bandwidth as 4 rad/s to stabilize the controller, and the PID parameters were redefined. For this reason, in the PID tuning algorithm, the phase margin was set as 84.2^o and the bandwidth as 4 rad/s to stabilize the controller, and the PID parameters were redefined. The redetermined PID parameters for the Z axis are given in Table 5.3.

Table 5.3 PID controller parameters for Z axis

Performance and Robustness		**Controller Parameters**	
Rise Time [s]	0.464	P	-0.21904
Settling Time [s]	4.23	I	-0.0051114
Overshoot [%]	3.41	D	-2.0856
Peak [-]	1.03	N	457.1312
Gain Margin [dB]	26.7		
Gain Margin Bandwidth [rad/s]	117		
Phase Margin [dB]	84.2		
Phase Margin Bandwidth [rad/s]	4		

Phase and gain characteristics for the Z axis controller are given in Figure 5.7 as a bode plot, including peak response and stability margins.

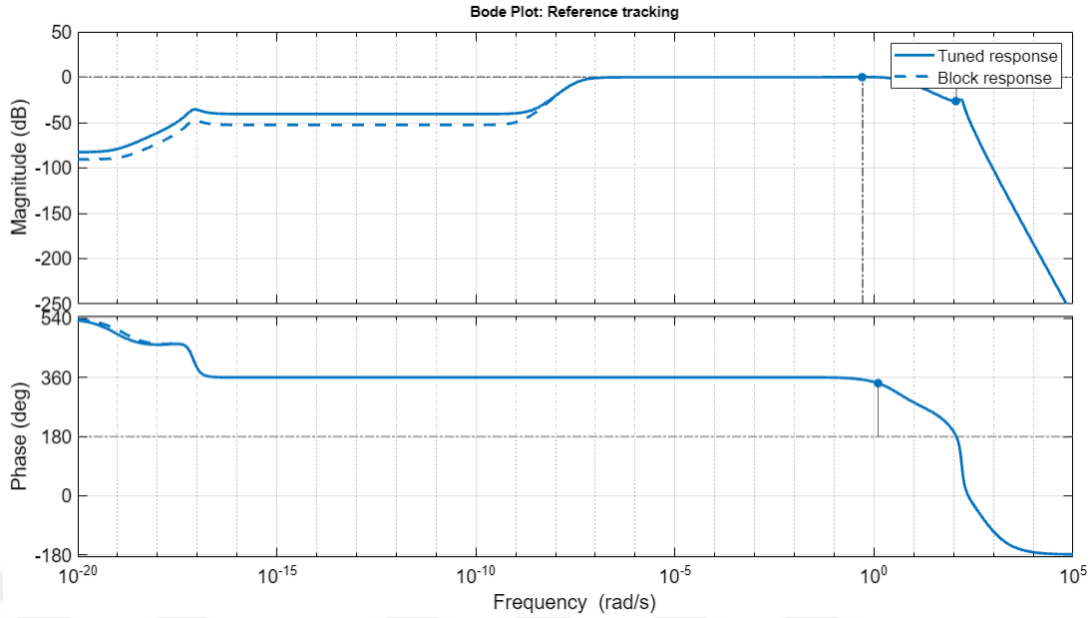


Figure 5.7 Reference tracking with default PID parameters

5.1.3 Simulation Results and Performance Analysis

After determining the PID controller gains with the "Transfer Function Based PID Tuning" application, which can be run in the PID block in the Simulink control library, the simulation was run and the performance of the system to follow the reference trajectory was observed as in Figure 5.8 and Figure 5.9.

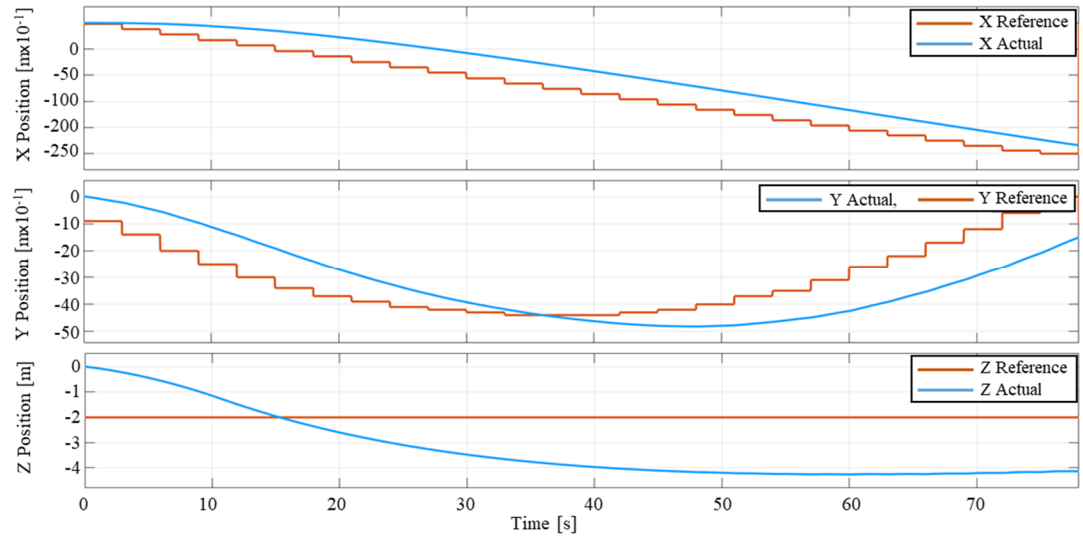


Figure 5.8 Reference tracking with default PID parameters

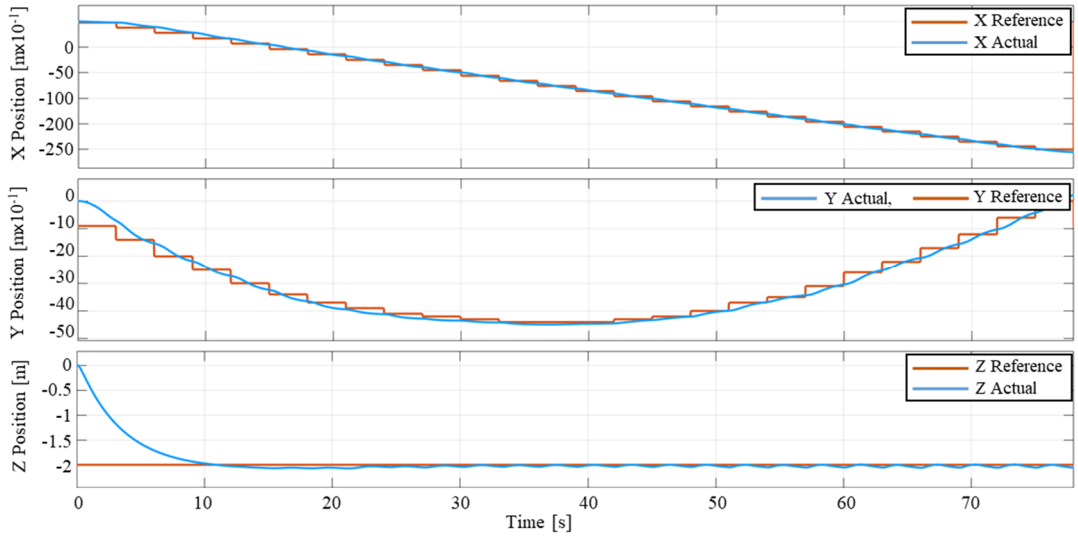


Figure 5.9 Reference tracking with optimized PID parameters

When the tracking performances given in Figure 5 and Figure 6 are analyzed, it is seen that the position error in all axes is reduced when the K_P, K_D and K_I gains are adjusted with the "Transfer Function Based PID Tuning" application in the second state, compared to the control applied with the default values in the first state.

The simulation was re-run by defining the phase margin and bandwidth that ensure stability for the Z axis and tuning the PID parameters accordingly. For this case, the response to the reference on the Z-axis is given in Figure 5.10.

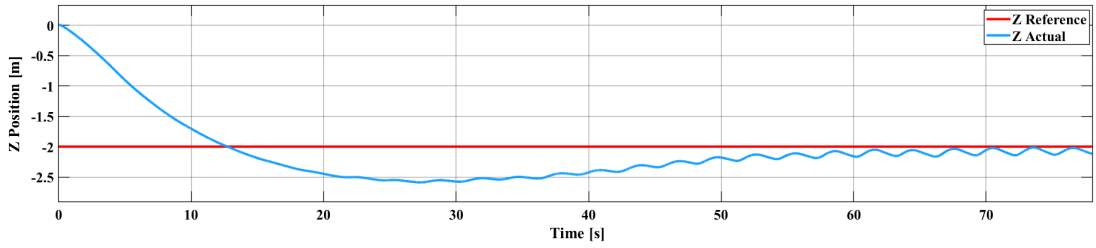


Figure 5.10 Stabilized Z position reference tracking

However, since it is known that the instability of the PID controller will increase under the influence of disturbing input, an LQR controller architecture is designed in the next scenario.

5.2 Scenario 2: Linear Quadratic Regulator Controller Design

Although the use of a linear controller in the first scenario provided information about the dynamic response of the system, the LQR control application, which is one of the optimal control methods known to be more successful than the PID controller, was studied in the second phase. In the previous scenario, PID controllers assigned separately for each axis were producing single output according to single input, but in this scenario, a multiple input-multiple output (MIMO) approach was adopted. Considering this situation, the simulation model of the system was restructured.

5.2.1 Simulation Model

Initially, the combined translational and rotational rigid body dynamics of the AUV should be obtained in accordance with the Newton-Euler formalism notation. Linear forces on Earth-Fixed Frame can be written in a sum operator as in Equation 5.2.

$$\sum F_{Earth-Fixed\ Frame} = m\dot{V}_{Earth-Fixed\ Frame} \tag{5.2}$$

Here $\dot{V}_{Earth\text{-}Fixed\ Frame}$ denotes the derivatives of linear speeds as in Equation 5.3.

$$\dot{\upsilon}_{Earth-Fixed\ Frame} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{5.3}$$

Angular moments on Body-Fixed Frame can be written in a sum operator as in Equation 5.4.

$$\sum M_{Body-Fixed Frame} = I\dot{\upsilon} + \upsilon \times I\upsilon \tag{5.4}$$

Here, I represents the inertia tensor and v represents the angular velocities and they are written as in Equations 5.5 and 5.6.

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{5.5}$$

$$\upsilon = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\theta & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{5.6}$$

Here, p, q, and r give the change in roll, pitch and yaw angles, that is, angular velocities. Forces and moments can be explicitly written as in Equations 5.7 and 5.8.

$$\sum F_{Earth-Fixed\ Frame} = \begin{pmatrix} 0 \\ 0 \\ m_g \end{pmatrix} + R \begin{pmatrix} u_z \\ u_y \\ u_z \end{pmatrix} + \begin{pmatrix} F_z \\ F_y \\ F_z \end{pmatrix}_{External} \tag{5.7}$$

$$\sum M_{Earth-Fixed\ Frame} = \begin{pmatrix} G_\phi \\ G_\theta \\ G_\psi \end{pmatrix} + \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} + \begin{pmatrix} \tau_x \\ \tau_y \\ \tau_z \end{pmatrix}_{External} \tag{5.8}$$

The term m_g in the explicit expression refers to the weight along the Z-axis. G_Φ, G_θ and G_ψ express gyroscopic effects, while $u_{x,y,z}$ and $\tau_{x,y,z}$ are linear and rotational actuation terms, respectively. For slow moving bodies, gyroscopic effects can be neglected. Another assumption is that p=q=0 can be taken within the scope of small angle approximation. This eliminates centrifugal terms. For the current situation, there is no torque for the angular dynamics.

The finalize dynamic model of AUV can be written as in Equations 5.9 to 5.14.

$$\frac{u_x c_\psi c_\theta + u_y (c_\psi s_\theta s_\phi - s_\psi c_\theta) + u_z (c_\psi c_\phi s_\phi + s_\psi s_\phi) + F_{External_x}}{m} = \ddot{x} \tag{5.9}$$

$$\frac{u_x s_\psi c_\theta + u_y (s_\psi s_\theta s_\phi - c_\psi c_\phi) + u_z (s_\psi c_\phi s_\theta - c_\psi s_\phi) + F_{External_y}}{m} = \ddot{y} \quad (5.10)$$

$$\frac{-u_x s_\theta + u_y c_\theta s_\phi + u_z c_\theta c_\phi + F_{External_z} + g}{m} = \ddot{z} \quad (5.11)$$

$$\frac{\tau_x}{I_x} = \ddot{\phi} \quad (5.12)$$

$$\frac{\tau_y}{I_y} = \ddot{\theta} \quad (5.13)$$

$$\frac{\tau_z}{I_z} = \ddot{\psi} \quad (5.14)$$

Without disturbance, the state-space form of the dynamic system can be written as in Equation 5.15.

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \quad (5.15)$$

The state vector, the time derivative vector of the states, and the input vectors can be written as in Equations 5.16 and 5.17.

$$\begin{aligned} X_{LQR} &= \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \\ &\Downarrow \\ \dot{X}_{LQR} &= \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \ddot{x} & \ddot{y} & \ddot{z} & \dot{\phi} & \dot{\theta} & \dot{\psi} & \ddot{\phi} & \ddot{\theta} & \ddot{\psi} \end{bmatrix}^T \end{aligned} \quad (5.16)$$

$$u = \begin{bmatrix} u_x & u_y & u_z & \tau_\phi & \tau_\theta & \tau_\psi \end{bmatrix}^T \quad (5.18)$$

The vectors f(x) and g(x) in the state-space form are written as in Equations 5.18 and 5.19.

$$f(x)=\begin{bmatrix}\dot{x} & \dot{y} & \dot{z} & 0 & 0 & 0 & \dot{\phi} & \dot{\theta} & \dot{\psi} & 0 & 0 & 0\end{bmatrix}^T \tag{5.18}$$

$$G(x)=\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{c_\psi c_\theta}{m} & \frac{c_\psi s_\theta s_\phi - s_\psi c_\phi}{m} & \frac{c_\phi s_\theta c_\psi + s_\psi s_\phi}{m} & 0 & 0 & 0 \\
\frac{s_\psi c_\theta}{m} & \frac{s_\psi s_\theta s_\phi + c_\psi c_\phi}{m} & \frac{c_\phi s_\theta s_\psi - c_\psi s_\phi}{m} & 0 & 0 & 0 \\
\frac{-s_\theta}{m} & \frac{c_\theta s_\phi}{m} & \frac{c_\theta c_\phi}{m} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{I_x} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{I_y} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{I_z} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \tag{5.19}$$

For linearization at S_{des}, Θ and ϕ are taken to zero, and u and τ to u_{des} and τ_{des} as desired values. With small angle approximation, the terms with sin and cos will converge to zero and one. Partial derivative operations required for Jacobian Linearization are given in Equations 5.20 to 5.28.

$$\frac{\partial \ddot{x}}{\partial \phi}=\frac{u_y c_\psi s_\theta c_\phi + u_y s_\psi s_\phi - u_z c_\psi s_\phi s_\theta + u_z s_\psi c_\phi}{m} \tag{5.20}$$

$$\frac{\partial \ddot{y}}{\partial \phi} = \frac{-u_y s_\psi s_\theta c_\phi - u_y c_\psi s_\phi - u_z s_\phi s_\theta s_\psi - u_z c_\psi c_\phi}{m} \tag{5.21}$$

$$\frac{\partial \ddot{z}}{\partial \phi} = \frac{u_y c_\theta c_\phi - u_z c_\theta s_\phi}{m} \tag{5.22}$$

$$\frac{\partial \ddot{x}}{\partial \theta} = \frac{-u_x c_\psi s_\theta + u_y c_\psi c_\theta s_\phi + u_z c_\phi c_\theta c_\psi}{m} \tag{5.23}$$

$$\frac{\partial \ddot{y}}{\partial \theta} = \frac{-u_x s_\psi s_\theta + u_y s_\psi c_\theta s_\phi + u_z c_\phi c_\theta s_\psi}{m} \tag{5.24}$$

$$\frac{\partial \ddot{z}}{\partial \theta} = \frac{-u_x c_\theta - u_y s_\theta s_\phi - u_z s_\theta c_\phi}{m} \tag{5.25}$$

$$\frac{\partial \ddot{x}}{\partial \psi} = \frac{-u_x s_\psi c_\theta - u_y (s_\psi s_\theta s_\phi - c_\psi c_\phi) + u_z (-c_\phi s_\theta s_\psi + c_\psi s_\phi)}{m} \tag{5.26}$$

$$\frac{\partial \ddot{y}}{\partial \psi} = \frac{u_x c_\psi c_\theta + u_y (c_\psi s_\theta s_\phi - s_\psi c_\phi) + u_z (c_\phi s_\theta c_\psi - s_\psi s_\phi)}{m} \tag{5.27}$$

$$\frac{\partial \ddot{z}}{\partial \psi} = 0 \tag{5.28}$$

Assuming that the roll and pitch movements are stabilized at zero, with small angle approximation, the terms with sinus and cosinus will converge to zero and one as in Equations 5.29 to 5.37.

$$\frac{\partial \ddot{x}}{\partial \phi} = \frac{u_{z_{des}} s_{\psi_{des}}}{m} \tag{5.29}$$

$$\frac{\partial \ddot{y}}{\partial \phi} = \frac{-u_{z_{des}} c_{\psi_{des}}}{m} \tag{5.30}$$

$$\frac{\partial \ddot{z}}{\partial \phi} = \frac{-u_{y_{des}}}{m} \tag{5.31}$$

$$\frac{\partial \ddot{x}}{\partial \theta} = \frac{u_{z_{des}} c_{\psi_{des}}}{m} \tag{5.32}$$

$$\frac{\partial \ddot{y}}{\partial \theta} = \frac{u_{z_{des}} s_{\psi_{des}}}{m} \tag{5.33}$$

$$\frac{\partial \ddot{z}}{\partial \theta} = \frac{-u_{x_{des}}}{m} \tag{5.34}$$

$$\frac{\partial \ddot{x}}{\partial \psi} = \frac{-u_{x_{des}} c_{\psi_{des}} - u_{y_{des}} c_{\psi_{des}}}{m} \tag{5.35}$$

$$\frac{\partial \ddot{y}}{\partial \psi} = \frac{-u_{x_{des}} c_{\psi_{des}} - u_{y_{des}} s_{\psi_{des}}}{m} \tag{5.36}$$

$$\frac{\partial \ddot{z}}{\partial \psi} = 0 \tag{5.37}$$

In Equations 5.38 to 5.46, partial derivatives are given according to the system inputs (u_x, u_y, u_z) that will form the B matrix.

$$\frac{\partial \ddot{x}}{\partial u_x} = \frac{c_\psi c_\theta}{m} = \frac{c_{\psi_{des}}}{m} \tag{5.38}$$

$$\frac{\partial \ddot{y}}{\partial u_x} = \frac{s_\psi c_\theta}{m} = \frac{s_{\psi_{des}}}{m} \tag{5.39}$$

$$\frac{\partial \ddot{z}}{\partial u_x} = \frac{-s_\theta}{m} = 0 \tag{5.40}$$

$$\frac{\partial \ddot{x}}{\partial u_y} = \frac{c_\psi s_\theta s_\phi - s_\psi c_\phi}{m} = \frac{-s_{\psi_{des}}}{m} \tag{5.41}$$

$$\frac{\partial \ddot{y}}{\partial u_y} = \frac{s_\psi s_\theta s_\phi - c_\psi c_\phi}{m} = \frac{c_{\psi_{des}}}{m} \tag{5.42}$$

$$\frac{\partial \ddot{z}}{\partial u_y} = \frac{c_\theta s_\phi}{m} = 0 \tag{5.43}$$

$$\frac{\partial \ddot{x}}{\partial u_z} = \frac{c_\phi s_\theta c_\psi + s_\psi s_\phi}{m} = 0 \tag{5.44}$$

$$\frac{\partial \ddot{y}}{\partial u_z} = \frac{c_\phi s_\theta c_\psi + s_\psi s_\phi}{m} = 0 \tag{5.45}$$

$$\frac{\partial \ddot{z}}{\partial u_z} = \frac{c_\theta c_\psi}{m} = \frac{1}{m} \tag{5.46}$$

LQR requires Linear Time Invariant (LTI) state-space representation for operations. For this case, the state-space form is written as a function as in Equation 5.47.

$$
\begin{aligned}
&\dot{x}(t) = f(x,u,t),\ y(t) = h(x) \\
&h(x) = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T \\
&A = \frac{\partial f}{\partial x}(X_0, U_0) \\
&B = \frac{\partial f}{\partial u}(X_0, U_0) \\
&C = \frac{\partial h}{\partial x}(X_0, U_0)
\end{aligned}
\tag{5.47}
$$

In the light of this information, system matrices X_0, U_0, A, B and C are written as in Equation 5.48 to 5.52.

$$
X_0 = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & \psi_{des} = 0^\circ & 0 & 0 & 0 \end{bmatrix}^T \tag{5.48}
$$

$$
u_0 = \begin{bmatrix} u_{x_{des}} & u_{y_{des}} & u_{z_{des}} & 0 & 0 & 0 \end{bmatrix}^T \tag{5.49}
$$

$$
A = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{u_z s_\psi}{m} & \frac{u_z c_\psi}{m} & \frac{-u_x s_\psi - u_y c_\psi}{m} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{-u_z c_\psi}{m} & \frac{u_z s_\psi}{m} & \frac{u_x c_\psi - u_y s_\psi}{m} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{u_y}{m} & \frac{-u_x}{m} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}_{des}
\tag{5.50}
$$

$u_{x_{des}}$, $u_{y_{des}}$ and $u_{z_{des}}$ will resist wave induced forces and weight force. The B matrix is written as in Equation 5.51.

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{c_\psi}{m} & -\frac{s_\psi}{m} & 0 & 0 & 0 & 0 \\ \frac{s_\psi}{m} & \frac{c_\psi}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix}_{des} \tag{5.51}$$

Since the angles Φ_{des} and Θ_{des} are stabilized, the terms τ_Φ and τ_Θ will also be zero. The C matrix is written as in Equation 5.52, with all terms that are not derivatives accepted as system output.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.52}$$

When the observability and controllability matrices are written as in Equations 5.53 to 5.55 and rank is checked, it is seen that they are full rank and their degrees are 12. In this case, the linearized system is observable and controllable.

$$Co = \left[B \vdots AB \vdots A^2B \vdots \cdots \vdots A^{n-1}B \right] \tag{5.53}$$

$$Ob = \left[C \vdots CA \vdots CA^2 \vdots \cdots \vdots CA^{n-1} \right]^T \tag{5.54}$$

$$\begin{aligned} rank(Co) &= 12 \\ rank(Ob) &= 12 \end{aligned} \tag{5.55}$$

5.2.2 Control System Inputs and Boundary Conditions

While defining disturbance, the force analysis for position 1 in the hydrodynamic analysis study presented under sub-heading 4.2.2 was taken as a basis. Lateral perturbation of 1.5 N, which emerged in the force analysis, was determined as the disturbance amplitude and was given to the system. A sinusoidal waveform was determined as the disturbance input model as in Equation 5.56 and integrated into the system.

$$F_{dist} = 1.5\sin(\omega t) \tag{5.56}$$

In case the disturbance input amplitude is given as the displacement perpendicular to the movement direction of the AUV in the Simulink environment, it is expressed by Equations 5.57 and 5.58 to be derived from the $\dot{x}=v$ relationship.

$$v = 0.158\cos(\omega t) \tag{5.57}$$

$$x = \int v dt = \frac{0.158}{\omega} \sin(\omega t) \tag{5.58}$$

The state reference vector containing the reference position which was defined as z=2 connected to the LQR controller.

The weights of the states that make up the Q matrix were adjusted mainly to control the Z coordinate. The input weights that make up the R matrix were adjusted

to regulate the state of the control signal. The most suitable Q and R matrices were determined by performing tests on the system model.

When the algorithm is run under these boundary conditions, the positions of the eigenvalues in the complex plane are shown in Figure 5.11.

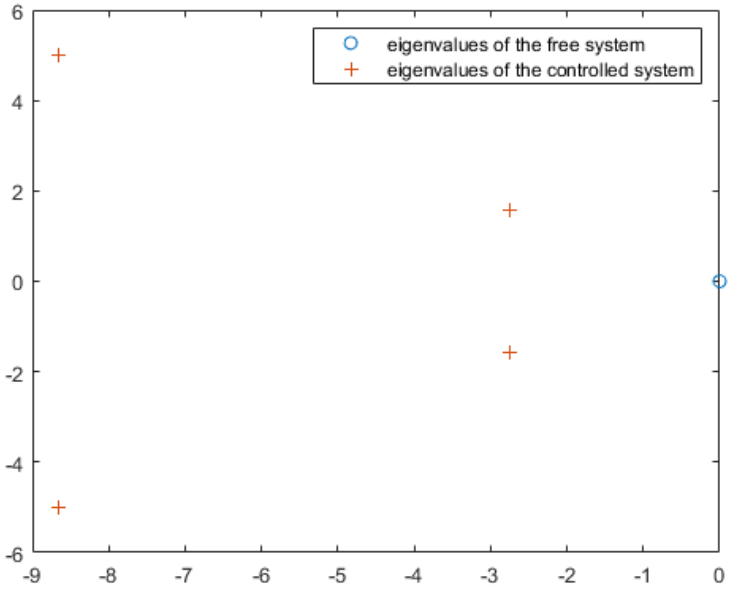


Figure 5.11 Eigenvalues of the system

As it can be seen from the location of eigenvalues, the poles of the original system were moved to the stable region as a result of the control.

5.2.3 Simulation Results and Performance Analyses

The block diagram of the simulation created in the MATLAB/Simulink environment by using the state space model, LQR algorithm, reference input model and disturbance input model, which includes the real behavior of the system is given in Figure 5.12.

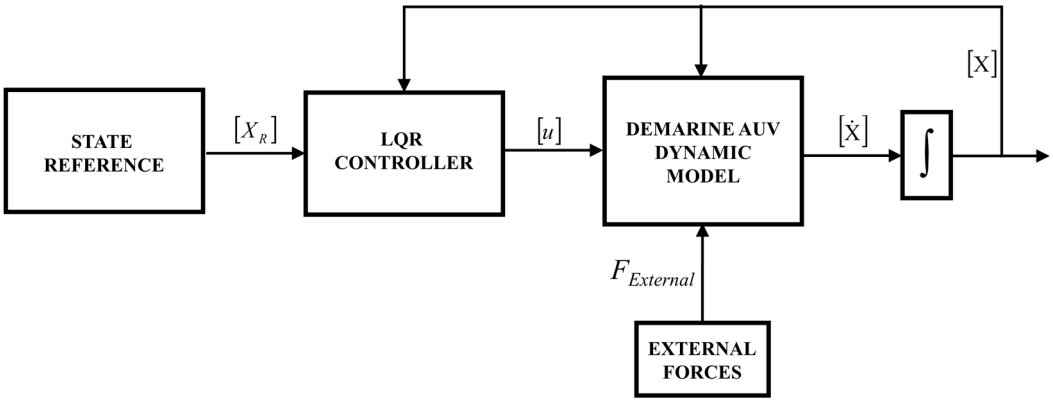


Figure 5.12 LQR controller design simulation model

When the Simulink model is run with the *X* and *Y* references of 0 meters and the *Z* references of 2 meters for the AUV, the time dependent system response for the *X*, *Y* and *Z* displacements is obtained as in Figure 5.13 and Figure 5.14.

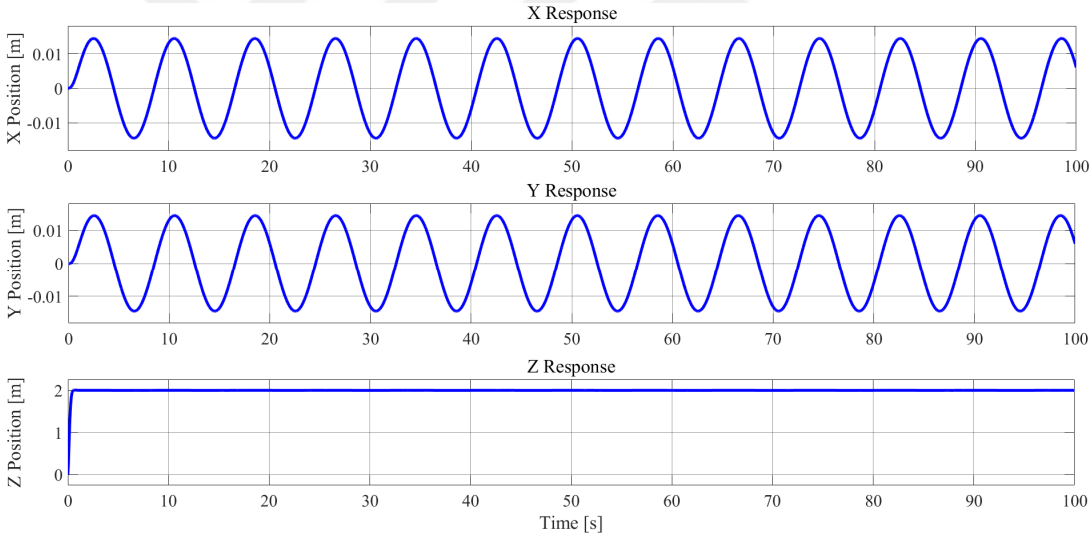


Figure 5.13 *X*, *Y* and *Z* responses of the system controlled by LQR

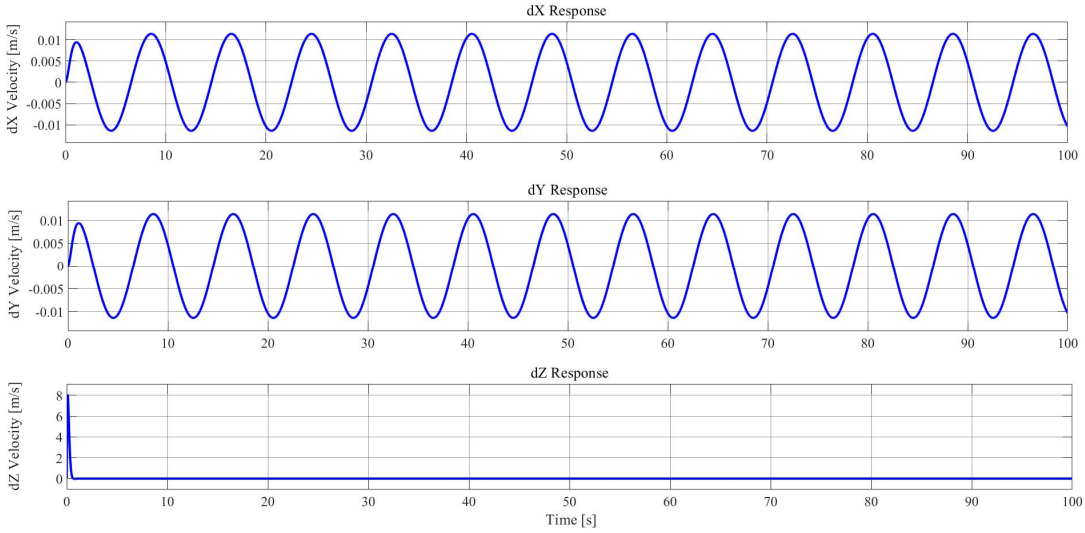


Figure 5.14 $\dot{X}$, $\dot{Y}$ *and* $\dot{Z}$ responses of the system controlled by LQR

At this point, while a reference was given for Z, it was desired to examine the behaviors under the influence of disturbance without defining reference to the angles. The time dependent system response for the ϕ, θ *and* ψ angular displacements is obtained as in Figure 5.15 and Figure 5.16.

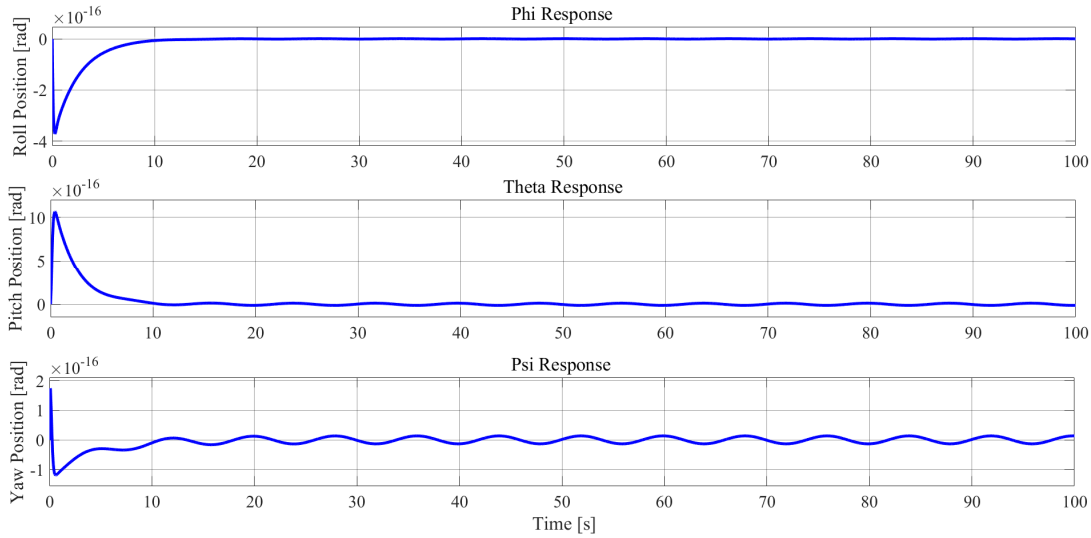


Figure 5.15 ϕ, θ *and* ψ responses of the system controlled by LQR

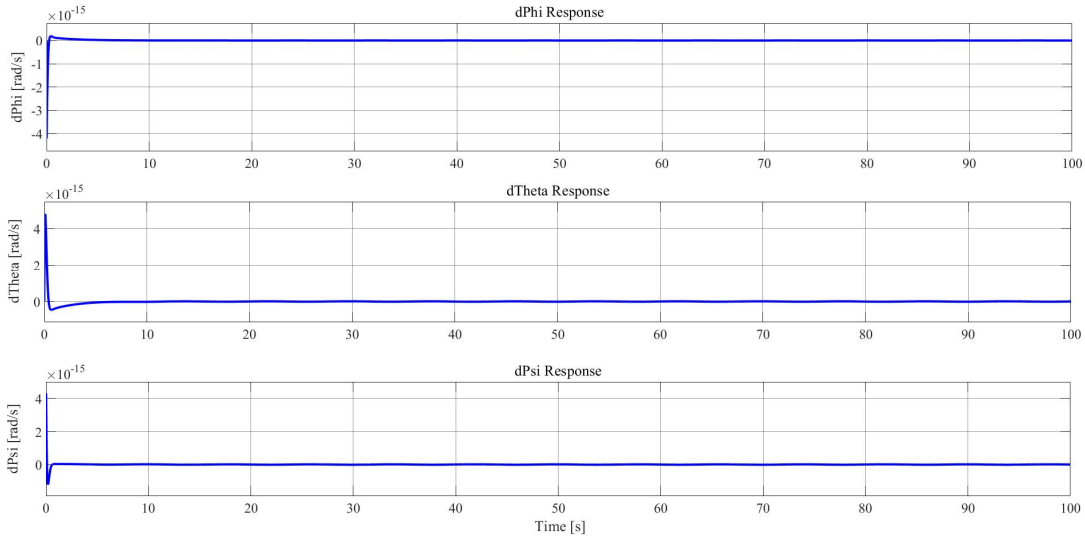


Figure 5.16 $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ responses of the system controlled by LQR

When the results are examined, it is seen that the AUV state-space model controlled by LQR responds in a high-speed, stable and error-eliminating manner in the *Z* axis despite the disturbing input. On the *X* and *Y* axes, it was observed that the position was maintained within a tolerance band with low amplitude movements. On the other hand, considering that the states will be split for the SMC in the next scenario, it can be ensured that the angular responses are also kept stable. The weakness of the LQR controller is that it can only accept reference input as single targets. In this case, it is not possible to perform a trajectory tracking performance analysis. For this reason, in the next scenario, a hybrid controller structure will be created by integrating the Sliding Mode Control (SMC) block into the system.

5.3 Scenario: 3 Sliding Mode and LQR Hybrid Control Architecture

SMC is known to be highly effective in controlling mobile robots due to its robustness, and that is why it was chosen in Scenario 3. SMC block has been added in order to eliminate some deficiencies of the Linear Quadratic Regulator-based controller structure, such as not being able to define the waypoints sequence as a reference input. It is aimed to obtain a response with stronger performance by adapting SMC, which is a variable structure system and is considered a robust control method.

The SMC method generalizes more than one system to a single one, takes parts from each of them when necessary and integrates them to obtain the desired behavior. Although the main problem of this control method is the chattering effect, different methods that can eliminate this undesirable situation have been presented in the literature. When the chattering effect is eliminated by using sigmoid function, sufficient stability can be obtained despite the presence of internal and external disturbances.

5.3.1 Simulation Model

It has been observed in Scenario 2 that the state errors of z,ϕ, θ *and* ψ can be reduced to zero when LQR gains are adjusted appropriately. For this reason, SMC is assigned for trajectory tracking missions to control *X* and *Y* degrees of freedom. The designed control architecture is illustrated in Figure 5.17

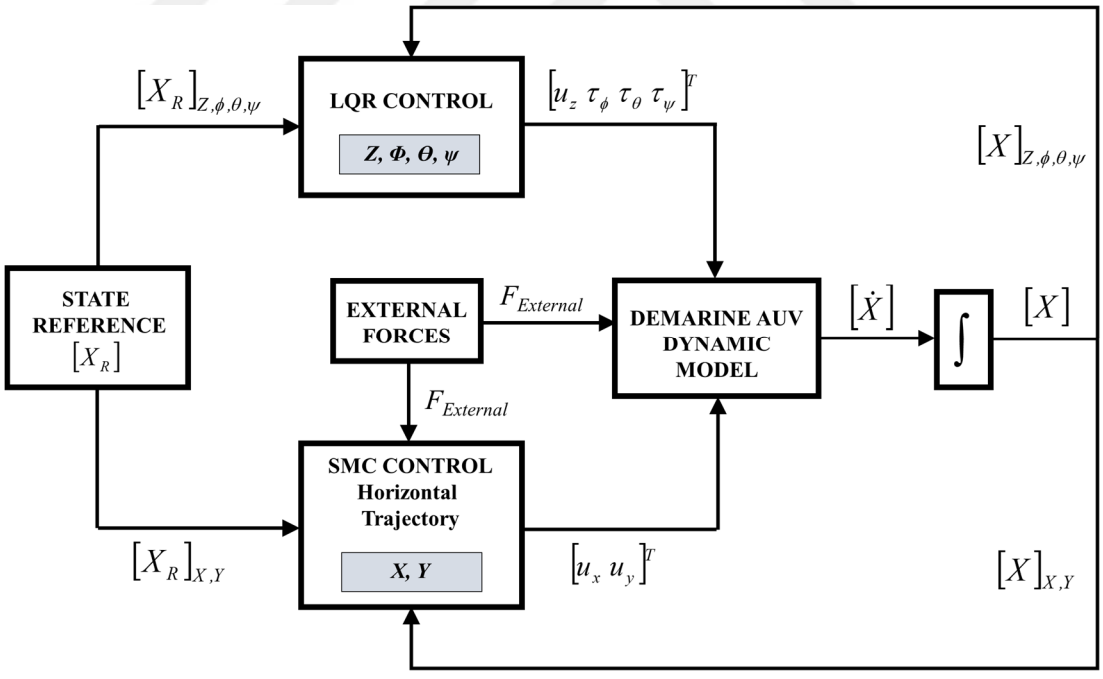


Figure 5.17 Sliding mode and LQR hybrid control architecture

As it can be seen in the figure, the desired state variables are separated and transmitted to the LQR and SMC blocks. At this stage, the SMC operates as a planar

trajectory controller. That is, control of X_{des} and Y_{des} of the desired states is handled by the SMC block.

Error definitions for the SMC controller are given in Equation 5.60.

$$\begin{aligned} e_x(t) &= x(t) - x_{des}(t) \\ e_y(t) &= y(t) - y_{des}(t) \end{aligned} \tag{5.60}$$

Sliding variables are given as in Equation 5.61.

$$\begin{aligned} S_x &= \dot{e}_x + \lambda_x e_x \\ S_y &= \dot{e}_y + \lambda_y e_y \end{aligned} \tag{5.61}$$

The derivative of the sliding variables with respect to time is given as in Equation 5.62.

$$\begin{aligned} S_x &= \ddot{e}_x + \lambda_x \dot{e}_x = \ddot{x} - \ddot{x}_{des} + \lambda_x \dot{e}_x \\ S_y &= \ddot{e}_y + \lambda_y \dot{e}_y = \ddot{y} - \ddot{y}_{des} + \lambda_y \dot{e}_y \end{aligned} \tag{5.62}$$

Where λ_x and λ_y are the coefficients that represents the slope in the error dynamics. LQR controller can keep θ and ϕ angles very close to zero. Therefore, s_θ and s_Φ can be taken as zero. Linear acceleration expressions can be written as in Equations 5.63 and 5.64.

$$\ddot{x} = \frac{u_x c_\psi c_\theta + F_{external\,x}}{m} \tag{5.63}$$

$$\ddot{y} = \frac{u_y c_\psi c_\theta + F_{external\,y}}{m} \tag{5.64}$$

Thus, $\dot{x}$, $\dot{S}_x$ and $\dot{S}_y$ can be written as in Equations 5.65 to 5.67.

$$\dot{x} = f(x) + g(x)u + d(t) \tag{5.65}$$

$$\dot{S}_x = \frac{u_x c_\psi c_\theta + F_{external\,x}}{m} + \lambda_x \dot{e}_x - \ddot{x}_{des} \tag{5.66}$$

$$\dot{S}_y = \frac{u_y c_\psi c_\theta + F_{external\,y}}{m} + \lambda_y \dot{e}_y - \ddot{y}_{des} \tag{5.67}$$

Sliding mode controller form consisting of equilibrium and robust parts is written as in Equation 5.68.

$$u = u_{eq} + u_R \tag{5.68}$$

For given disturbance u_{eq} is designed for $\dot{S}$=-ks and $d(t)=0$ as in Equations 5.69 and 5.70.

$$u_{x_{eq}} = \frac{m}{c_\psi c_\theta}(-\lambda_x \dot{e}_x + \ddot{x}_{des} - k_x s_x) \tag{5.69}$$

$$u_{y_{eq}} = \frac{m}{c_\psi c_\theta}(-\lambda_y \dot{e}_y + \ddot{y}_{des} - k_y s_y) \tag{5.70}$$

When Equations 5.69 and 5.70 are replaced in Equations 5.66-5.67, the expressions -$k_x S_x$ and -$k_y S_y$, which are considered as linear controllers, remain after the simplification of the respective expressions. When designing the second part of the controller form, the u_R, it is necessary to insert a $\mu sign(s)$ greater than $F_{external}$. Accordingly, u_R can be designed as in Equations 5.71 and 5.72.

$$u_{x_R} = -\frac{m}{c_\psi c_\theta}\mu_x sign(s_x) \tag{5.71}$$

$$u_{y_R} = -\frac{m}{c_\psi c_\theta}\mu_y sign(s_y) \tag{5.72}$$

High frequency switching control in sliding mode causes chattering effect. Ideally, the amplitude should tend to zero and the frequency to infinity. In discrete-time simulation, the sign function causes this defect. This situation can be overcome with the smooth sigmoid function as in Equation 5.73.

$$sign(\sigma) \approx \frac{\sigma}{|\sigma| + \varepsilon} \tag{5.73}$$

Stability control of each sliding variable is required before applying the controller. For this purpose, the storage functions given in Equations 5.74-5.77 are defined for two slip surfaces and they are required to be greater than or equal to zero according to the Lyapunov Stability Theorem, whose theoretical background is given in Chapter 2. However, the derivative of the storage function must be less than or equal to zero. When a sufficiently large coefficient k is inserted, the first term becomes less than zero. In the second term, it becomes less than zero when the μ factor is large enough according to $F_{external}$.

$$V_x(s) = \frac{1}{2} s_x^{\,2} \tag{5.74}$$

$$\begin{aligned} \dot{V}_x(s,\dot{s}) = s_x \dot{s}_x &= s_x(-k_x s_x - \mu_x sign(s_x) - d(t)) \\ &= \underbrace{-k_x s_x^{\,2}}_{\leq 0 \, for \, k_x > 0} - |s_x| \Big(\underbrace{\mu_x - F_{external_x}}_{\leq 0 \, for \, \mu_x \geq \left\| F_{external_x}(t) \right\|_\infty} \Big) \end{aligned} \tag{5.75}$$

$$V_y(s) = \frac{1}{2} s_y^{\,2} \tag{5.76}$$

$$\begin{aligned} \dot{V}_y(s,\dot{s}) = s_y \dot{s}_y &= s_y(-k_y s_y - \mu_y sign(s_y) - d(t)) \\ &= \underbrace{-k_y s_y^{\,2}}_{\leq 0 \, for \, k_y > 0} - |s_y| \Big(\underbrace{\mu_y - F_{external_y}}_{\leq 0 \, for \, \mu_y \geq \left\| F_{external_y}(t) \right\|_\infty} \Big) \end{aligned} \tag{5.77}$$

Since a decoupled cascaded system is currently being built, the states and the inputs created in the previous scenario needs to be updated as corresponding submatrices as given in Equations 5.78 and 5.79 for Scenario-3.

$$X_{LQR} = \begin{bmatrix} z \\ \dot{z} \\ \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix} \Rightarrow \dot{X}_{LQR} = \begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} \tag{5.78}$$

$$u_{LQR} = \begin{bmatrix} u_z \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{5.79}$$

Re-differentiation operations are performed in accordance with the given new states and A_{LQR}, B_{LQR} and C_{LQR} matrices are obtained as in Equations 5.80-5.82.

$$A_{LQR} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{u_y}{m} & 0 & -\frac{u_x}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.80}$$

$$B_{LQR} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} \tag{5.81}$$

$$C_{LQR} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.82}$$

Assumption: If operating point of desired dynamics is constrained to get small accelerations, then $u_x=u_y\approx 0$ can be said.

5.3.2 Control System Inputs and Boundary Conditions

A circle with a radius of 0.2 m is defined as the position reference for the AUV to follow in the horizontal plane. As in the previous scenario, a disturbance input of $F_{dist}=1.5\sin(\omega t)$ is given to the plant and the simulation is run.

When the algorithm is run under these boundary conditions, the positions of the eigenvalues in the complex plane are determined as in Figure 5.18.

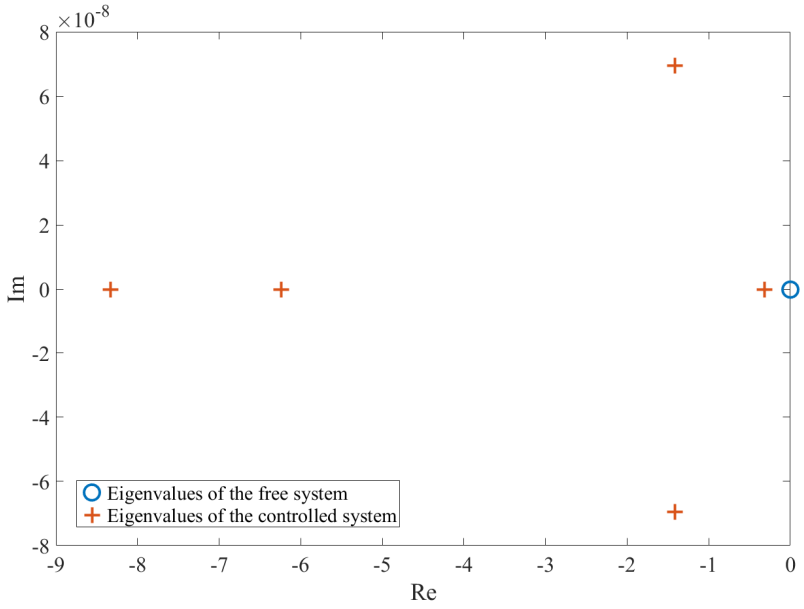


Figure 5.18 Eigenvalues of LQR + SMC hybrid control system

5.3.3 Simulation Results and Performance Analyses

The time dependent system response for the *X*, *Y* and *Z* displacements is obtained as in Figure 5.19.

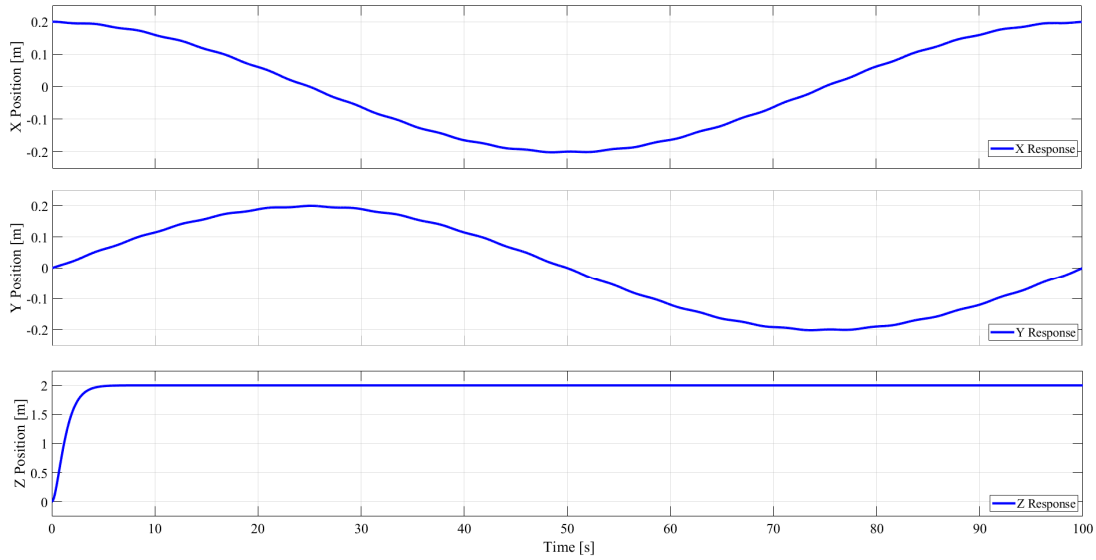


Figure 5.19 *X*, *Y* and *Z* responses of the system controlled by LQR + SMC hybrid control system

3D plot of the AUV trajectory and reference is given as in Figure 5.20.

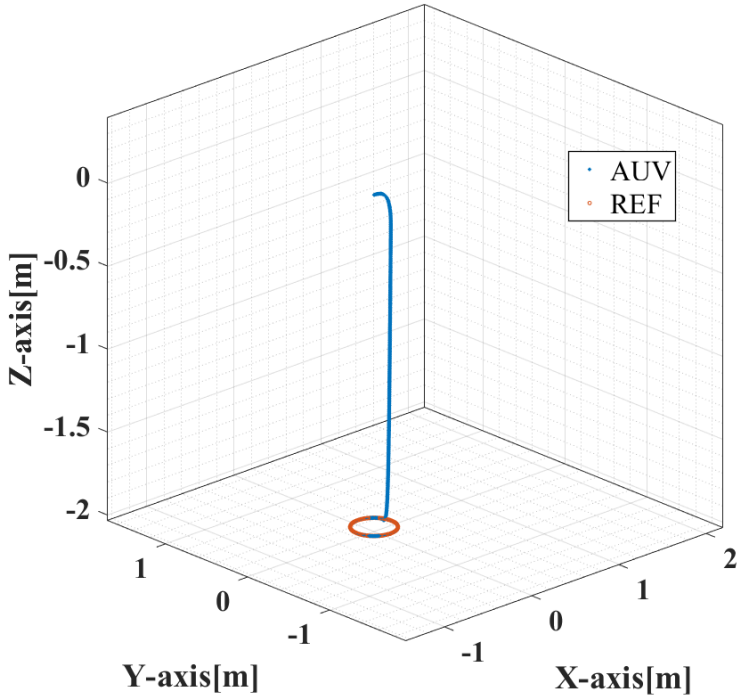


Figure 5.20 3D plot of the AUV trajectory and the reference

When the results are analyzed, it is seen that the hybrid controller created by adapting the SMC can follow the circular reference given in the horizontal plane with a very low error. In addition, when the time-dependent *X* and *Y* position response is examined, it carries the effect of disturbance given to the system, albeit vaguely. When looking at the *Z* position, it is seen that the controller enables to reach the target coordinate in about 3 seconds and while doing this, it has a response character with almost zero error, zero overshoot and zero oscillation.

CHAPTER SIX
GENERAL DISCUSSION

This PhD thesis is organized for the mechanical, electronic and software implementation of an AUV capable of investigating wet hull surfaces of boats and ships. For this purpose, in the first stage, the production of an initial prototype was made in the laboratory environment. As the process progressed, the initial prototype was optimized thanks to the scientific and technical experience gained from the research, and the advanced devices and equipment obtained from the project supports. The developed AUV prototype was equipped with thrusters, sensor systems, power supplies, sealing solutions and software that were made more efficient at every stage throughout the process, and the data obtained were reported and published as scientific articles and papers.

In particular, the information obtained from detailed literature research on hydrodynamic analysis and motion control of underwater robots has been effective in determining the design criteria of the DEMarine AUV prototype. In this context, the distant and recent history of AUVs has been examined and it has been determined which features they should have for certain missions. Studies on the determination of hydrodynamic properties of underwater robots with experimental and numerical methods have been scanned, and the planning of analysis studies for the DEMarine AUV prototype to determine the resistances and forces that it will be exposed to during navigation and maneuvering has been clarified. Thus, the boundary conditions of the CFD analyzes to be made within the scope of the thesis were determined and a road map was drawn.

Again in the literature review, system modeling and feedback control methods for AUVs were investigated in detail, and the organizational structure of the control applications chapter was determined by researching the most efficient and useful feedback controllers for the DEMarine AUV, which has an open-frame body. In this context, it was decided to establish and test PID, LQR and LQR+SMC structures in the simulation model environment, respectively, for control applications. The

literature survey on controller structures also revealed that the studies carried out within the scope of the DEMarine AUV project can contribute to fill a gap in the literature and applied technique due to the mini-sized and open-frame form of the developed prototype.

Along with the literature review and the gains it brings, the theoretical infrastructures of the methodologies of mathematical modelling, flow analysis, system identification and control system applications that will be used in the thesis have been examined and presented in detail with their formulations.

The design phase was carried out by considering many criteria, especially the mission descriptions determined for the AUV, the desired maneuvering capacity, the mission duration, the trajectory characteristics to be covered and the available budget. All the components that make up the design were modeled and assembled in 3D in a computer environment one by one. While designing, dimensions and standard fasteners were selected according to manufacturability and ergonomics conditions. Most of the structural parts that make up the resulting design prototype could be manufactured with widely accessible production machines such as a 3D prototype production device, lathe, milling machine, CNC and laser cutting machine. As the underwater tests and improvement works continued, the mechanical and electronic hardware was updated and these updates were also reflected in the final design. As a result, the current DEMarine AUV prototype has been implemented by combining standard machine elements with custom-made parts produced with separate laboratory and industrial facilities.

An important part of this work is the design, installation and testing of electronic circuits for many different purposes. While developing the prototypes, different battery packs and different thrusters were used, and the energy transmission and distribution circuits were redeveloped for each battery-thruster combination. In the first stages of the project, a brushless DC motor with a maximum thrust of 0.885 kgf used in quadcopters was used. Propeller and nozzle suitable for this DC motor have been designed and produced. As the second version thruster, a more professional

brushless DC motor and ESC combination is used, which can produce 2 kgf of thrust. In the third prototype, T200 thrusters with thrust parameters up to a maximum of 6.7 kgf were used. Different power distribution circuits have been designed and used by considering the maximum navigation range depending on the operating voltages, current drops and powers at which all these different thrusters can operate at maximum efficiency. In addition, the energy needs of sensor systems, cameras and led lighting are also provided through the common battery. For this reason, integrating each component that needs power input required the reorganization of the power network.

Yet another phase of the design, installation, testing and commissioning of electronic circuits includes signal generation, signal transmission and communication. The first of the applications made for this purpose was the Ethernet circuit that runs the communication protocol between the surface station computer and the internal embedded computers in DEMarine. PC and embedded computers were connected to the same network and a master-slave relationship could be established thanks to this communication circuit. In the next step, the control circuit that transmits the control signals from the remote control device from one embedded computer to the other and then to the thrust system has been established. In this designed circuit, software that converts and transmits control signals as analog and digital, and also performs mapping has been developed and run.

Another important part of the design, installation, testing and commissioning of electronic circuits is the installation of embedded sensor system circuits, including data logging/mapping. The stages of these works, which can also be defined as instrumentation, are energizing the sensor and obtaining the output signal, obtaining raw sensor data from embedded computer serial output channel and processing of sensor output data to obtain real physical measurement information. In this process, in which individual configurations were made for each sensor system, a 30 Bar pressure and depth sensor was first instrumented for depth measurement. I2C communication protocol level converter and two embedded computers were used to receive data from the sensor. Hydrostatic pressure and depth information has been made readable via the serial port with the Arduino library of the pressure sensor and the written code. The

measured data were transferred to the ROS environment running on another embedded computer, Raspberry PI, and monitored on the surface station computer via Ethernet connection.

Subsequently, instrumentation was performed for two different sonars for distance measurement based navigation and imaging. The first of these sonars was a single-beam echo sounder that could measure up to 30 meters. The other sonar is a mechanical acoustic transducer that can scan 360º. UART communication protocol was used in the circuit setup of these four sonars in total. By converting the TX and RX signals to the USB standard, the raw sonar data was transmitted to the embedded computer. Distance information and confidence percentage were obtained by running the relevant ROS libraries and the developed code in the embedded computer. In addition, it is possible to observe and record sonar data visually by making use of the graphical user interface provided by the manufacturer. If all sonars can be properly positioned on the AUV and operated simultaneously, it is thought that inspection, obstacle avoidance, target location and tracking operations can be performed as well as navigation.

Another sensor system instrumentation has been made for inertial measurements. In this way, the acquisition of position, speed, acceleration, force, angular position and orientation information is aimed. In the circuit using a 9 DoF IMU containing a microprocessor as well as the sensor, the raw data was transferred to the embedded computer via the I2C communication protocol, the inertial measurements on the X, Y and Z axes were displayed with the sensor package and the developed code. In addition, Kalman Filtering algorithm has been applied and tested in order to increase the reliability of the data received from the IMU.

Finally, within the scope of the sensor system instrumentation, an electronic circuit is configured for commissioning the camera module. The camera module, which was planned to be used for both visual data collection and sonar data verification, was connected to the embedded computer via the serial camera control bus. The images obtained by running the required nodes in the embedded computer were transferred to the surface computer with a uniform resource identifier.

Numerical analysis was carried out to determine the hydrodynamic parameters for two different scenarios within the scope of the thesis. The two scenarios differ from each other in terms of trajectory geometry and boundary conditions. The results obtained from the CFD analyzes carried out were used in the calculations of technical limits and performance parameters such as power, thrust capacity, current draw, efficiency, range and battery life.

The stages of the hydrodynamic analysis study conducted within the scope of Scenario 1 and the results obtained can be listed as follows:

- The study organized for DEMarine AUV prototype following a predetermined trajectory on the basis of hydrodynamic characteristics depending on angular positions

- A mission scenario was set for the AUV to circumnavigate a marine vessel along its waterline within a particular margin. The trajectory geometry was generated from the water line dimensions of marine vessel DEU-1 by cubic spline curve fitting method.

- Based on the determined trajectory, a navigation plan consisting of linear distances and rotation angles was created for energy calculation. Derivation points along the trajectory and navigation data consisting of linear segments and yaw angles were identified.

- In compliance with angular position information received from the navigation plan, hydrodynamic impacts on the AUV were obtained by Computational Fluid Dynamics (CFD) analysis based on Finite Volume Method (FVM).

- Motion induced energy requirement model was established and computerized as an executable algorithm.

- CFD analysis was organized for calculating the hydrodynamic coefficients, total resistance and moments acting on AUV based on roll and yaw angle variations. For roll/yaw angles in the range of 0°-30° and velocity of 1.5 m/s, CFD analysis designating k-ε turbulence model was run.

- The resulting hydrodynamic data was used for evaluating energy requirement which provides performance parameters of AUV navigating along its trajectory.

The stages of the hydrodynamic analysis study conducted within the scope of the Scenario 2 and the results obtained can be listed as follows:

- The study organized for obtaining thrust capacity, power and energy parameters of DEMarine AUV following a logarithmic trajectory segment.

- The parametric forms of curve equations are identified to represent the relevant trajectory characteristics along which the AUV performs its maneuvering motion.

- An algorithm was developed and computerized to acquire curvature and radius of curvature data sets for each point on the trajectory.

- A flow domain, in which a fixed speed free streamflow of the seawater was simulated in compliance with the AUV trajectory.

- For various locations of the AUV inside the flow domain, Computational Fluid Dynamics (CFD) analysis, which uses the Finite Volume Method (FVM), was carried out.

- The hydrodynamic force values due to the radius of curvature obtained from the CFD analysis were used as an input to find the heading and lateral thrust and power requirement of the AUV during the maneuvering motion.

The control system design part of the thesis is structured separately for 3 different scenarios. The dynamic behavior data and hydrodynamic analysis data obtained from the open-loop controller underwater tests were used as boundary conditions for the simulation models of the feedback controllers, taking into account the task definitions determined for the DEMarine AUV. In 3 different scenarios where both various controllers are applied and different references are defined, first the simulation models are created mathematically, then these models are transferred to MATLAB functions and then reconstructed as block diagrams in the Simulink environment. In the first scenario, a simulation model from the Mathworks platform was adapted for DEMarine AUV.

In the first scenario, a simulation model from the Mathworks platform was adapted for DEMarine AUV. This simulation model includes translational reference, translation controller, plant & environment model, and sensor data & state estimation subsystems. The position reference generated in the Translational reference subsystem has been converted to the trajectory used in the hydrodynamic analysis scenario 1. The reference sent to the translation controller in a sequence of step inputs is transferred to PIDs that control the position on each axis separately. The reference sent to the translation controller in a sequence of step inputs is transferred to PIDs that control the position on each axis separately. At this stage, the process of setting the PID controller parameters has been started. The transfer function based tuning algorithm in the PID block is used for this work. The desired stability behavior was adjusted in the algorithm, and the PID parameters were determined, and the error performance and tracking ability were compared with the PID controller operating with the default parameters. In addition, transfer functions were obtained from the s domain with a fitting ratio of 91.44% using the MATLAB System Identification Toolbox of the thruster models included in the simulation model. Again, the axis distances forming the moment arms are defined in the moment computation block in accordance with the physical prototype dimensions. Finally, the forces caused by the drag are defined in the environmental forces subsystem using the C_D coefficient obtained from the hydrodynamic analysis. When the results obtained are examined, it is seen that the

AUV performs the reference tracking at the desired efficiency in the trials where there is no disturbing input and the PID gain parameters are finely adjusted.

In the second scenario, it is aimed that the dynamic system can follow the reference under the influence of disturbing input and an LQR controller is designed. For the LQR controller, which is intended to be operated as MIMO system, all state matrices were determined during the construction of the mathematical model of the simulation model. As a disturbance, the lateral perturbation vector, which is taken as an external effect within the scope of hydrodynamic analysis, due to the maximum current in Izmir Bay, is given to the dynamic system. By determining the position of the eigen-values, it was ensured that the system was stable and the response of the LQR controller was obtained for the given reference. When the results are examined, it is seen that the reference depth position in the Z axis can be reached in a short time and it oscillates in a tolerable bandwidth in a way that carries the character of the disturbing input at this position. When the results are examined, it is seen that the horizontal position of the system in the X and Y axes can be reached in a short time and the system oscillates in a tolerable bandwidth in a way that carries the character of the disturbing input in this position.

In the third and final scenario, state references are shared to LQR and SMC. The purpose here is for the AUV to go to the depth target with LQR and follow the horizontal trajectory with the SMC controller. Thus, LQR's lack of trajectory tracking has been eliminated. The mathematical model was rearranged for the new LQR and SMC, and after coding the related functions in MATLAB, the block diagram was set up in the Simulink environment. The simulation was run for a circular reference and the disturbance input model applied in the previous scenario and the results were examined. As a result, the system performed with a very low error in both capturing the depth reference and following the circular trajectory.

CHAPTER SEVEN
CONCLUSIONS AND FUTURE WORKS

7.1 Conclusions

In this thesis, the stages of design, manufacturing, analysis and control applications of an AUV are presented in a way that includes scientific, theoretical, technical and practical applications.

In Chapter One, the background of the AUVs and motivation of the work, research objectives and questions, the significance of the study and an overview of the thesis are presented. In Chapter Two, AUVs and their applications are reviewed in terms of trajectory tracking control strategies and hydrodynamic analysis. Recent advances reported have identified the limitations of existing controller designs, revealing gaps in current research. At this point, it has been determined that there are disconnections between the acquisition of hydrodynamic parameters and the design of suitable nonlinear controllers, and it is understood that the contribution of the study is concentrated in this framework. In addition to these, theoretical descriptions of the AUV system model were made and information about the control methods used mostly was given.

In the studies carried out within the scope of Chapter Three, the infrastructure of the AUV prototype was designed and implemented with all its mechanical and electronic equipment. All algorithms running thrust and embedded sensor systems are designed and coded in this chapter. In Chapter Four, hydrodynamic analysis studies were carried out and the behavior of the AUV prototype under different flow conditions was reported. The hydrodynamic parameters obtained as a result of these studies constituted inputs within the control system structures. In Chapter 5, where three different controller structures are introduced, a PID controller that follows the reference trajectory, an LQR controller that enables reaching the given coordinate targets, and a decoupled LQR+SMC controller that can follow a given circular orbital target with low error are designed.

When the response of the controller structure created by dividing the states into LQR and SMC is examined, it can be deduced that compared to single PID and single LQR controller applications, it gives a more robust and near-zero steady state error performance in both planar trajectory tracking and control of depth and rotational states.

When the study content is examined, it can be considered as an important scientific contribution that the hydrodynamic properties of an open-frame, compact-sized AUV are determined by CFD method and used as input data for control structures.

7.2 Future Works

The parts of this study, which has a very wide scope, that are open to development and the topics that will add value to the study in the future are listed below:

- The production and testing of different body types as a design will be beneficial in terms of revealing the most functional form.

- Re-adjusting the positions and axis arrangements of the T200 thrusters, which can produce sufficient power and work efficiently, will allow to improve the maneuvering performance.

- The deployment of sensors, which have been made operational and data can be received, on the AUV is important for the fulfillment of autonomous tasks in real working conditions.

- Moving the experimental studies to larger pools and the real marine environment will play an important role in making the prototype functional.

- During the hydrodynamic analyzes carried out within the scope of the thesis study, it was seen that there is a need for the existence of reference geometries, which have been previously tested and the results of which are known, in order

to measure the accuracy of the numerical calculations of the behavior of unmanned underwater robots such as ROVs and AUVs under flow.

- The hydrodynamic analysis study is planned to be expanded for different AUV body forms, different current velocities and different orbital geometries in the following period. The optimum combination of these parameters can be determined to obtain the most suitable navigation scenario for minimum thrust and power needs. Moreover, the battery and thruster capacities required for the operation of AUV can be determined by evaluating the study results. Thus, the autonomous controller can be configured that adjusts the amount of energy to be delivered to the actuators.

- Transferring the controller structures developed and tested on the simulation model to electronic equipment, testing on the real physical prototype and under real disturbance effects and comparing the results with the simulation data is one of the main studies planned to be carried out in the future for control theory and applications.

- It is planned to integrate the Adaptive Backstepping algorithm, which requires less parameters to be tuned, in the light of the data obtained by running the PID, LQR and LQR+SMC controller structures developed for DEMarine AUV.

- In addition, it is planned to implement the Dynamic Feedback Linearization method with Non-Interactive Control, which has the ability to control complex nonlinear systems, the ability to independently design controllers for each subsystem, and the ability to achieve high control performance.

REFERENCES

Amiri, M. M., Sphaier, S. H., Vitola, M. A., & Esperança, P. T. (2019). URANS investigation of the interaction between the free surface and a shallowly submerged underwater vehicle at steady drift. *Applied Ocean Research*, *84*, 192–205. https://doi.org/10.1016/J.APOR.2019.01.012

ANSYS Fluent. (2015). *Ansys Fluent Theory Guide* (16.2, Vol. 15317, Issue July). ANSYS Inc. http://www.pmt.usp.br/academic/martoran/notasmodelosgrad/ANSYS Fluent Theory Guide 15.pdf

Antonelli, G., Chiaverini, S., Sarkar, N., & West, M. (2001). Adaptive control of an autonomous underwater vehicle: Experimental results on ODIN. *IEEE Transactions on Control Systems Technology*, *9*(5), 756–765. https://doi.org/10.1109/87.944470

Aras, M. S. M., Abdullah, S. S., Rahman, A. A., & Aziz, M. A. A. (2013). Thruster modelling for underwater vehicle using system identification method. *International Journal of Advanced Robotic Systems*, *10*. https://doi.org/10.5772/56432

Bao-Ji, Z., Xin-Di, L., & Wen-Xuan, S. (2019). Resistance performance simulation of remotely operated vehicle in deep sea considering propeller rotation. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 1475090219867604. https://doi.org/10.1177/1475090219867604

Bellingham, J. G., & Rajan, K. (2007). Robotics in remote and hostile environments. *Science*, *318*(5853), 1098–1102. https://doi.org/10.1126/science.1146230

Bocharov, L. (2009). Underwater vehicle: current status and overall trends (part 1). *Electronics: Science, Technology, Business*, *7*.

Caccia, M. (2007). Vision-based ROV horizontal motion control: Near-seafloor

experimental results. *Control Engineering Practice*, *15*(6), 703–714. https://doi.org/10.1016/j.conengprac.2006.05.008

Cai, L., Sun, Q., Xu, T., Ma, Y., & Chen, Z. (2020). Multi-AUV Collaborative Target Recognition Based on Transfer-Reinforcement Learning. *IEEE Access*, *8*, 39273–39284. https://doi.org/10.1109/ACCESS.2020.2976121

Carneiro, J. F., Bravo Pinto, J., De Almeida, F. G., & Cruz, N. (2019). Using a variable buoyancy system for energy savings in an AUV. *Proceedings of the 2019 5th Experiment at International Conference, Exp.at 2019*, 305–309. https://doi.org/10.1109/EXPAT.2019.8876514

Chapra, S. C. (2011). *Applied Numerical Methods with MATLAB for Engineers and Scientists* (third). McGraw Hill.

Cheney, E. W., & Kincaid, D. R. (2013). *Numerical Mathematics and Computing* (seventh). Brooks/Cole, Cengage Learning. https://books.google.com.tr/books?id=p6xAAAAACAAJ

Cheng, J., Yi, J., & Zhao, D. (2007). Design of a sliding mode controller for trajectory tracking problem of marine vessels. *Iet Control Theory and Applications*, *1*, 233–237.

Christopher, E., & Spurgeon, S. (1998). *Sliding mode control: theory and applications*. Crc Press.

Craven, P. J., Sutton, R., & Burns, R. S. (1998). Control strategies for unmanned underwater vehicles. *Journal of Navigation*, *51*(1), 79–105. https://doi.org/10.1017/S0373463397007601

Cui, R., Yang, C., Li, Y., & Sharma, S. (2017). Adaptive Neural Network Control of AUVs with Control Input Nonlinearities Using Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*(6), 1019–1029. https://doi.org/10.1109/TSMC.2016.2645699

da Silva Costa, G., Almeida, M. P., de Melo Filho, A. R. V., da Cunha Lima, A. T., &

da Cunha Lima, I. C. (2019). 3-D realistic simulations over a Flatfish shaped AUV submitted to ocean currents. *Applied Ocean Research*, *90*, 101849. https://doi.org/10.1016/J.APOR.2019.06.006

De Barros, E. A., & Dantas, J. L. D. (2012). Effect of a propeller duct on AUV maneuverability. *Ocean Engineering*, *42*, 61–70. https://doi.org/10.1016/j.oceaneng.2012.01.014

Eng, Y. H., Teo, K. M., Chitre, M., & Ng, K. M. (2016). Online System Identification of an Autonomous Underwater Vehicle Via In-Field Experiments. *IEEE Journal of Oceanic Engineering*, *41*(1), 5–17. https://doi.org/10.1109/JOE.2015.2403576

Evans, J., & Nahon, M. (2004). Dynamics modeling and performance evaluation of an autonomous underwater vehicle. *Ocean Engineering*, *31*(14–15), 1835–1858. https://doi.org/10.1016/j.oceaneng.2004.02.006

Farhadi, A., Mayrhofer, A., Tritthart, M., Glas, M., & Habersack, H. (2018). Accuracy and comparison of standard k-ε with two variants of k-ω turbulence models in fluvial applications. *Engineering Applications of Computational Fluid Mechanics*, *12*(1), 216–235. https://doi.org/10.1080/19942060.2017.1393006

Fiorelli, E., Leonard, N. E., Bhatta, P., Paley, D. A., Bachmayer, R., & Fratantoni, D. M. (2006). Multi-AUV control and adaptive sampling in Monterey Bay. *IEEE Journal of Oceanic Engineering*, *31*(4), 935–948. https://doi.org/10.1109/JOE.2006.880429

Fossen, T. I. (1994). *Guidance and Control of Ocean Vehicles*. Wiley. https://books.google.com.tr/books?id=cwJUAAAAMAAJ

Fossen, T. I. (2002). Marine control systems. *Lecture Notes*. http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Marine+Control+Systems#1

Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control* (1st ed.). John Wiley & Sons.

Gan, W., Zhu, D., Hu, Z., Shi, X., Yang, L., & Chen, Y. (2020). Model Predictive Adaptive Constraint Tracking Control for Underwater Vehicles. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, *67*(9), 7829–7840.

Gao, T., Wang, Y., Pang, Y., & Cao, J. (2016). Hull shape optimization for autonomous underwater vehicles using CFD. *Engineering Applications of Computational Fluid Mechanics*, *10*(1Gao, T., Wang, Y., Pang, Y., Cao, J., 2016. Hull shape optimization for autonomous underwater vehicles using CFD. Engineering Applications of Computational Fluid Mechanics 10, 599–607. doi:10.1080/19942060.2016.1224735), 599–607. https://doi.org/10.1080/19942060.2016.1224735

Germanischer Lloyd. (2016). *(I-5-3) Unmanned Submersibles (ROV,AUV) and Underwater Working Machines*.

González-García, J., Gómez-Espinosa, A., Cuan-Urquizo, E., García-Valdovinos, L. G., Salgado-Jiménez, T., & Escobedo Cabello, J. A. (2020). Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions. *Applied Sciences (Switzerland)*, *10*(4). https://doi.org/10.3390/app10041256

Hammad, M. M., Elshenawy, A. K., & El Singaby, M. I. (2018). Position Control and Stabilization of Fully Actuated AUV using PID Controller. *Lecture Notes in Networks and Systems*, *16*, 517–536. https://doi.org/10.1007/978-3-319-56991-8_39

Haugen, F. (2004). *PID Control*. Fagbokforlaget.

Hou, Y. hang, Liang, X., & Mu, X. yang. (2018). AUV hull lines optimization with uncertainty parameters based on six sigma reliability design. *International Journal of Naval Architecture and Ocean Engineering*, *10*(4), 499–507. https://doi.org/10.1016/j.ijnaoe.2017.10.001

Isidori, A. (1995). *Nonlinear Control Systems* (3rd ed). Springer-Verlag.

Jagadeesh, P., Murali, K., & Idichandy, V. G. (2009a). Experimental investigation of

hydrodynamic force coefficients over AUV hull form. *Ocean Engineering*, *36*(1), 113–118. https://doi.org/10.1016/j.oceaneng.2008.11.008

Jagadeesh, P., Murali, K., & Idichandy, V. G. (2009b). Experimental investigation of hydrodynamic force coefficients over AUV hull form. *Ocean Engineering*, *36*(1), 113–118. https://doi.org/10.1016/J.OCEANENG.2008.11.008

Javanmard, E. (2013). *Determination of Hydrodynamic Coefficients of an AUV with Computational Fluid Dynamics and Experimental Fluid Dynamics Methods, Master's Thesis*. Isfahan University of Technology.

Jiang, C. meng, Wan, L., Sun, Y. shan, & Li, Y. ming. (2018). Design of novel sliding-mode controller for high-velocity AUV with consideration of residual dead load. *Journal of Central South University*, *25*(1), 121–130. https://doi.org/10.1007/s11771-018-3722-y

Jones, C., Webb, D., Kohut, J., Glenn, S., Kerfoor, J., Roarty, H., Aragon, D., Haldeman, C., Graver, J., & Yan, A. (2007). Slocum Gliders Advancing Oceanography. *Proceedings of the 15th International Symposium on Unmaned Untethered Submersible Technology Conference (UUST'07)*, *March 2017*. http://auvac.com/uploads/publication_pdf/jones_slocum_gliders_advancing_oceanography.pdf

Jones, D., Bett, B. J., Wynn, R. B., & Masson, D. G. (2008). The use of towed camera platforms in deep-water science. *Underwater Technology*, *28*(2), 41–50. https://doi.org/10.3723/ut.28.041

Kaya, K. D., & Gören, A. (2017). Linear Load Transfer Based Horizontal Motion Control of a ROV. *Proceedings of the World Electro Mobility Conferenece*, 102–105.

Kaya, Kerim Deniz, & Gören, A. (2015). Yarı Otonom Bir Sualtı Robotu Tasarımı. *Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı*, 789–794.

Kaya, Kerim Deniz, Goren, A., Yilmaz, S., & Bayramoğlu, K. (2020). Determination of operating parameters of an AUV following a preplanned trajectory using

hydrodynamic analysis data. *Ocean Engineering*, *217*, 107708. https://doi.org/10.1016/j.oceaneng.2020.107708

Khodayari, M. H., & Balochian, S. (2015). Modeling and control of autonomous underwater vehicle (AUV) in heading and depth attitude via self-adaptive fuzzy PID controller. *Journal of Marine Science and Technology (Japan)*, *20*(3), 559–578. https://doi.org/10.1007/s00773-015-0312-7

Kim, G. H., & Park, S. (2017). Development of a numerical simulation tool for efficient and robust prediction of ship resistance. *International Journal of Naval Architecture and Ocean Engineering*, *9*(5), 537–551. https://doi.org/10.1016/J.IJNAOE.2017.01.003

Kirk, D. E. (2004). *Optimal Control Theory: An Introduction*. Dover Publications.

Li, T., Zhao, R., Chen, C. L. P., Fang, L., & Liu, C. (2018). Finite-time formation control of under-actuated ships using nonlinear sliding mode control. *IEEE Transactions on Cybernetics*, *48*(11), 3243–3253. https://doi.org/10.1109/TCYB.2018.2794968

Li, Ye, Jiang, Y. qing, Wang, L. feng, Cao, J., & Zhang, G. cheng. (2015). Intelligent PID guidance control for AUV path tracking. *Journal of Central South University*, *22*(9), 3440–3449. https://doi.org/10.1007/s11771-015-2884-0

Li, Ye, Wei, C., Wu, Q., Chen, P., Jiang, Y., & Li, Y. (2015). Study of 3 dimension trajectory tracking of underactuated autonomous underwater vehicle. *Ocean Engineering*, *105*, 270–274. https://doi.org/10.1016/j.oceaneng.2015.06.034

Li, Yongcheng, Pan, D., Zhao, Q., Ma, Z., & Wang, X. (2018). Hydrodynamic performance of an autonomous underwater glider with a pair of bioinspired hydro wings–A numerical investigation. *Ocean Engineering*, *163*, 51–57. https://doi.org/10.1016/J.OCEANENG.2018.05.052

Londhe, P. S., Dhadekar, D. D., Patre, B. M., & Waghmare, L. M. (2017). Uncertainty and disturbance estimator based sliding mode control of an autonomous underwater vehicle. *International Journal of Dynamics and Control*, *5*(4), 1122–

1138. https://doi.org/10.1007/s40435-016-0260-z

Mansoorzadeh, S., & Javanmard, E. (2014). An investigation of free surface effects on drag and lift coefficients of an autonomous underwater vehicle (AUV) using computational and experimental fluid dynamics methods. *Journal of Fluids and Structures*, *51*, 161–171. https://doi.org/10.1016/J.JFLUIDSTRUCTS.2014.09.001

MathWorks Student Competitions Team. (2022). *Modeling and Simulation of an Autonomous Underwater Vehicle*. The MathWorks, Inc. https://github.com/mathworks/AUV-modeling-and-sim

MathWorks Student Competitions Team. (2023). *MATLAB and Simulink Robotics Arena : From Data to Model*. https://www.mathworks.com/matlabcentral/fileexchange/65919-matlab-and-simulink-robotics-arena-from-data-to-model

Michel, D., & Wernli, R. L. (1998). *Operational Effectivenes of Unmanned Underwater Systems* (R. L. Wernli (Ed.)). Marine Technology Society Remotely Operated Vehicle Committee.

Mitra, A., Panda, J. P., & Warrior, H. V. (2019). The effects of free stream turbulence on the hydrodynamic characteristics of an AUV hull form. *Ocean Engineering*, *174*, 148–158. https://doi.org/10.1016/J.OCEANENG.2019.01.039

Mitra, A., Panda, J. P., & Warrior, H. V. (2020). Experimental and numerical investigation of the hydrodynamic characteristics of Autonomous Underwater Vehicles over sea-beds with complex topography. *Ocean Engineering*, *198*, 106978. https://doi.org/10.1016/J.OCEANENG.2020.106978

Natarajan, S., Gaudig, C., & Hildebrandt, M. (2012). Offline experimental parameter identification using on-board sensors for an Autonomous Underwater Vehicle. *Ieee*, 1–8. https://doi.org/10.1109/OCEANS.2012.6404905

Ngo, S. I., & Lim, Y. Il. (2020). Multiscale eulerian CFD of chemical processes: A review. *ChemEngineering*, *4*(2), 1–27.

https://doi.org/10.3390/chemengineering4020023

Nidagundi, D. S. (2017). Chronology of underwater vehicle. *International Journal of Latest Engineering and Management Research (IJLEMR)*, *02*(02), 89–91.

Paull, L., Saeedi, S., Seto, M., & Li, H. (2014). AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, *39*(1), 131–149. https://doi.org/10.1109/JOE.2013.2278891

Ramesh, R., Ramadass, N., Sathianarayanan, D., Vedachalam, N., & Ramadass, G. A. (2013). Heading control of ROV ROSUB6000 using non-linear model-aided PD approach. *International Journal of Emerging Technology and Advanced Engineering*, *3*(4), 382–393.

Rashidi, A. J., Karimi, B., & Khodaparast, A. (2020). A Constrained Predictive Controller for AUV and Computational Optimization Using Laguerre Functions in Unknown Environments. *International Journal of Control, Automation and Systems*, *18*(3), 753–767. https://doi.org/10.1007/s12555-018-0946-4

Rustrian, W. (2016). MODELING OF A SMALL REMOTELY OPERATED UNDERWATER VEHICLE FOR AUTONOMOUS NAVIGATION AND CONTROL [California State University]. In *ProQuest LLC*. https://www.researchgate.net/publication/269107473_What_is_governance/link/548173090cf22525dcb61443/download%0Ahttp://www.econ.upf.edu/~reynal/Civil wars_12December2010.pdf%0Ahttps://think-asia.org/handle/11540/8282%0Ahttps://www.jstor.org/stable/41857625

Sahoo, A., Dwivedy, S. K., & Robi, P. S. (2020). Dynamic Modelling and Control of a Compact Autonomous Underwater Vehicle. In A. P. P. Abdul Majeed, J. A. Mat-Jizat, M. H. A. Hassan, Z. Taha, H. L. Choi, & J. Kim (Eds.), *RITA 2018* (pp. 303–321). Springer Singapore.

Salari, M., & Rava, A. (2017). Numerical investigation of hydrodynamic flow over an AUV moving in the water-surface vicinity considering the laminar-turbulent transition. *Journal of Marine Science and Application*, *16*(3), 298–304.

https://doi.org/10.1007/s11804-017-1422-x

Sarkar, M., Nandy, S., Vadali, S. R. K., Roy, S., & Shome, S. N. (2016). Modelling and simulation of a robust energy efficient AUV controller. *Mathematics and Computers in Simulation*, *121*, 34–47. https://doi.org/10.1016/j.matcom.2015.08.021

Sayin, E. (2003). Physical features of the Izmir Bay. *Continental Shelf Research*, *23*(10), 957–970. https://doi.org/10.1016/S0278-4343(03)00083-9

Scheirer, D. S., Fornari, D. J., Humphris, S. E., & Lerner, S. (2000). High-resolution seafloor mapping using the DSL-120 sonar system: Quantitative assessment of sidescan and phase-bathymetry data from the Lucky Strike segment of the Mid-Atlantic Ridge. *Marine Geophysical Researches*, *21*(1–2), 121–142. https://doi.org/10.1023/a:1004701429848

Shirzadi, M., Mirzaei, P. A., & Naghashzadegan, M. (2017). Improvement of k-epsilon turbulence model for CFD simulation of atmospheric boundary layer around a high-rise building using stochastic optimization and Monte Carlo Sampling technique. *Journal of Wind Engineering and Industrial Aerodynamics*, *171*(May 2018), 366–379. https://doi.org/10.1016/j.jweia.2017.10.005

Simonetti, P., & Webb, D. C. (1997). A Simplified Approach to the Prediction and Optimization of Performance of Underwater Gliders. *Proceedings of the 10th International Symposium on Unmanned Untethered Submersible Technology*.

SNAME. (1950). *Nomenclature for Treating the Motion of a Submerged Body through a Fluid*.

Svirin, A. (2009). *MATH24*. https://math24.net/curvature-radius.html

T200 Thrusters. (n.d.). BlueRobotics Company. https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/

Tang, S., Ura, T., Nakatani, T., Thornton, B., & Jiang, T. (2009). Estimation of the hydrodynamic coefficients of the complex-shaped autonomous underwater

vehicle TUNA-SAND. *Journal of Marine Science and Technology*, *14*(3), 373–386. https://doi.org/10.1007/s00773-009-0055-4

Tchilian, R. D. S., Rafikova, E., Gafurov, S. A., & Rafikov, M. (2017). Optimal Control of an Underwater Glider Vehicle. *Procedia Engineering*, *176*, 732–740. https://doi.org/10.1016/j.proeng.2017.02.322

Tian, W., Song, B., & Ding, H. (2019). Numerical research on the influence of surface waves on the hydrodynamic performance of an AUV. *Ocean Engineering*, *183*, 40–56. https://doi.org/10.1016/J.OCEANENG.2019.04.007

Tivey, A., & Paul, H. (1998). *magnetic field mapping by autonomous underwater vehicle Maurice Dana Yoerger*. *25*(6), 805–808.

Tivey, M. A., Bradley, A., Yoerger, D., Catanach, R., Duester, A., Liberatore, S., & Singh, H. (1997). Autonomous underwater vehicle maps seafloor. *Eos*, *78*(22). https://doi.org/10.1029/97EO00151

Tu, J., Yeoh, G. H., & Liu, C. (2013). *Computational fluid dynamics : a practical approach*. Elsevier/Butterworth-Heinemann. http://public.eblib.com/choice/publicfullrecord.aspx?p=1012531

Versteeg, H. K., & Malalasekera, W. (2007). An Introduction to Computational Fluid Dynamics. In *Pearson Inc.* (second). Pearson Education Limited. https://doi.org/10.2514/1.22547

Vervoort, J. H. A. . (2008). *Modeling and Control of an Unmanned Underwater Vehicle*. University of Canterbury.

Wynn, R. B., Huvenne, V. A. I., Le Bas, T. P., Murton, B. J., Connelly, D. P., Bett, B. J., Ruhl, H. A., Morris, K. J., Peakall, J., Parsons, D. R., Sumner, E. J., Darby, S. E., Dorrell, R. M., & Hunt, J. E. (2014). Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, *352*, 451–468. https://doi.org/10.1016/j.margeo.2014.03.012

Xu, J., Wang, M., & Qiao, L. (2014). Backstepping-based controller for three-dimensional trajectory tracking of underactuated unmanned underwater vehicles. *Kongzhi Lilun Yu Yinyong/Control Theory and Applications*, *31*, 1589–1596. https://doi.org/10.7641/CTA.2014.30835

Yan, Z., Gong, P., Zhang, W., & Wu, W. (2020). Model predictive control of autonomous underwater vehicles for trajectory tracking with external disturbances. *Ocean Engineering*, *217*(July), 107884. https://doi.org/10.1016/j.oceaneng.2020.107884

Yang, N., Chang, D., Amini, M. R., Johnson-Robersor, M., & Sun, J. (2019). Energy management for autonomous underwater vehicles using economic model predictive control. *Proceedings of the American Control Conference*, *2019-July*, 2639–2644. https://doi.org/10.23919/acc.2019.8815106

Yazici, S., Arslan, T., Uyar, Z., Bahadir, I., Sarac, I., Kartal, E., Ozdenoglu, M. T., Cirakman, E., Karadag, A. E., Yildizay, C. R., Kirgoz, S. C., Polat, Y., & Bal, D. S. (2021). *Istanbul Technical University RoboSub 2021 Technical Design Report*.

Yoerger, D. R., Bradley, A. M., Walden, B. B., Singh, H., & Bachmaver, R. (1998). Surveying a subsea lava flow using the Autonomous Benthic Explorer (ABE). *International Journal of Systems Science*, *29*(10), 1031–1044. https://doi.org/10.1080/00207729808929596

Yüksel, İ. (2014). *Otomatik Kontrol Sistem Dinamiği ve Denetim Sistemleri* (9th ed.). Dora Basım-Yayın Dağıtım Ltd. Şti. www.dorayayincilik.com.tr

Zavari, S., Usenius, O., Villa, J., Heininen, A., Salomaa, T., Laitinen, J., Aaltonen, J., & Tkoskinen, K. (2019). Propulsion system development and power consumption in an autonomous underwater vehicle. *2019 23rd International Conference on System Theory, Control and Computing, ICSTCC 2019 - Proceedings*, 521–524. https://doi.org/10.1109/ICSTCC.2019.8885503

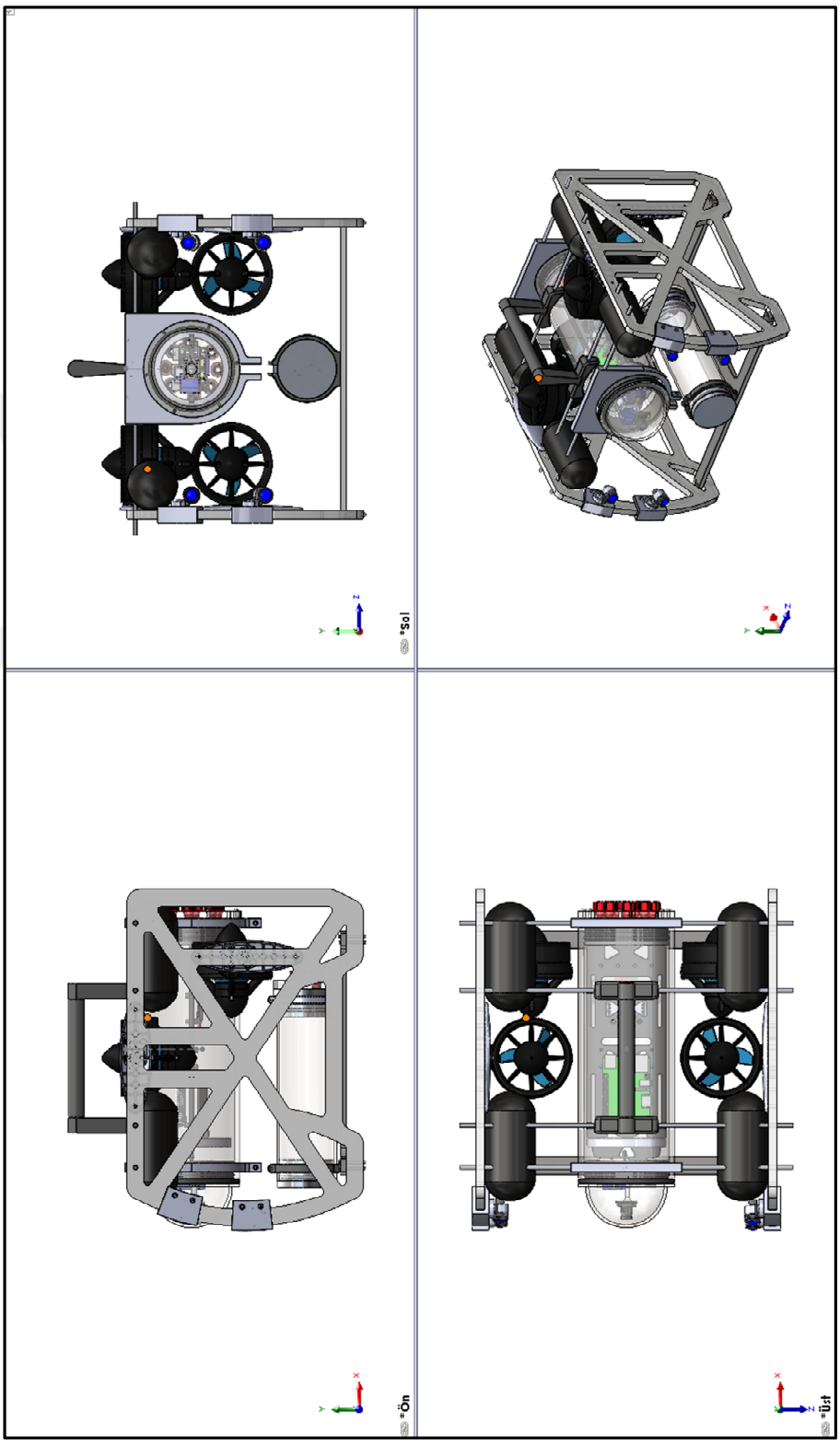
Zhang, J. H., & Yao, X. L. (2003). *Ship Control Theory* (2nd ed.). Harbin Engineering University Press.

Zhang, Yongding, Liu, X., Luo, M., & Yang, C. (2019). MPC-based 3-D trajectory tracking for an autonomous underwater vehicle with constraints in complex ocean environments. *Ocean Engineering*, *189*(March), 106309. https://doi.org/10.1016/j.oceaneng.2019.106309

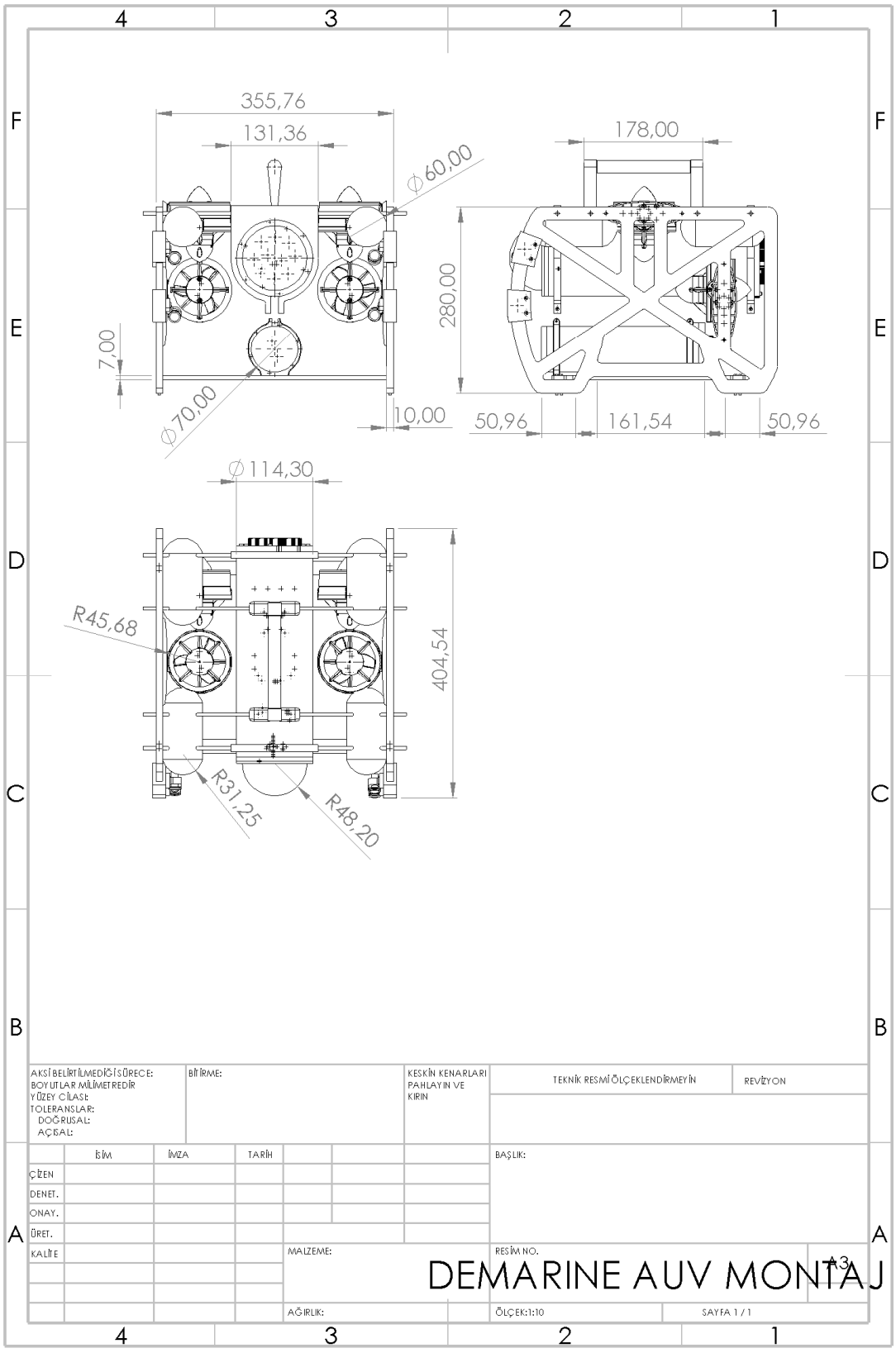
Zhang, Youmin, & Mehrjerdi, H. (2013). A survey on multiple unmanned vehicles formation control and coordination: Normal and fault situations. *2013 International Conference on Unmanned Aircraft Systems, ICUAS 2013 - Conference Proceedings*, 1087–1096. https://doi.org/10.1109/ICUAS.2013.6564798

APPENDICES

Appendix 1 DEMarine AUV 3D Model



Appendix 2 DEMarine AUV Prototype Technical Drawing



Appendix 3 Source Codes

Pressure/Depth Sensing Code (Arduino)

```
#include <Wire.h>
#include "MS5837.h"
MS5837 sensor;

void setup() {
  Serial.begin(9600);
  Serial.println("Starting");
  Wire.begin();
  // Initialize pressure sensor
  // Returns true if initialization was successful
  // We can't continue with the rest of the program unless we can initialize
the sensor
  while (!sensor.init()) {
    Serial.println("Init failed!");
    Serial.println("Are SDA/SCL connected correctly?");
    Serial.println("Blue Robotics Bar30: White=SDA, Green=SCL");
    Serial.println("\n\n\n");
    delay(5000);
  }

  sensor.setModel(MS5837::MS5837_30BA);
  sensor.setFluidDensity(997); // kg/m^3 (freshwater, 1029 for seawater)
}

void loop() {
  // Update pressure and temperature readings
  sensor.read();

  Serial.print("Pressure: ");
  Serial.print(sensor.pressure());
  Serial.println(" mbar");

  Serial.print("Temperature: ");
  Serial.print(sensor.temperature());
  Serial.println(" deg C");

  Serial.print("Depth: ");
  Serial.print(sensor.depth());
  Serial.println(" m");

  Serial.print("Altitude: ");
  Serial.print(sensor.altitude());
  Serial.println(" m above mean sea level");

  delay(1000);
}
```

Distance Measurement Code (Phyton):

```
!/usr/bin/env python

from brping import Ping1D
import time
import argparse

from builtins import input

##Parse Command line options
```

Distance Measurement Code (continued)

```
############################

parser = argparse.ArgumentParser(description="Ping python library example.")
parser.add_argument('--device',  action="store",  required=True,  type=str,
help="Ping device port.")
parser.add_argument('--baudrate', action="store", type=int, default=115200,
help="Ping device baudrate.")
args = parser.parse_args()

#Make a new Ping
myPing = Ping1D(args.device, args.baudrate)
if myPing.initialize() is False:
    print("Failed to initialize Ping!")
    exit(1)

print("----------------------------------")
print("Starting Ping..")
print("Press CTRL+C to exit")
print("----------------------------------")

input("Press Enter to continue...")

# Read and print distance measurements with confidence
while True:
    data = myPing.get_distance()
    if data:
        print("Distance:   %s\tConfidence:   %s%%"   %   (data["distance"],
data["confidence"]))
    else:
        print("Failed to get distance data")
    time.sleep(0.1)
```

Inertial Measurement Unit Running Code (Phyton):

```
#include <ros.h>
#include <std_msgs/String.h>
#include <Wire.h>
const int MPU_addr=0x68; // I2C address of the MPU
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
//Set up the ros node and publisher
std_msgs::String imu_msg;
ros::Publisher imu("imu", &imu_msg);
ros::NodeHandle nh;
void setup()
{
nh.initNode();
nh.advertise(imu);
Wire.begin();
Wire.beginTransmission(MPU_addr);
Wire.write(0x6B); // PWR_MGMT_1 register
Wire.write(0); // set to zero (wakes up the MPU)
Wire.endTransmission(true);
Serial.begin(9600);
}
long publisher_timer;
void loop()
{
Wire.beginTransmission(MPU_addr);
Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
Wire.endTransmission(false);
```

```
Inertial Measurement Unit Running Code (Continued)

Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers String
AX =
String(mpu6050.getAccX());
AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
String AX = String(AcX);
String AY = String(AcY);
String AZ = String(AcZ);
String GX = String(GyX);
String GY = String(GyY);
String GZ = String(GyZ);
String tmp = String(Tmp);
String data = "A" + AX + "B"+ AY + "C" + AZ + "D" + GX + "E" + GY + "F" + GZ
+ "G" ;
Serial.println(data);
int length = data.indexOf("G") +2;
char data_final[length+1];
data.toCharArray(data_final, length+1);
if (millis() > publisher_timer) {
// step 1: request reading from sensor
imu_msg.data = data_final;
imu.publish(&imu_msg);
publisher_timer = millis() + 100; //publish ten times a second
nh.spinOnce();
}
}
```

Motion-Induced Energy Requirement Algorithm Code (MATLAB):

```
clear all; close all; clc;
Angle=[0 0.3 0.6 0.9 1.2 1.5 1.8 2.1 2.4 2.7 3 3.3 3.6 3.9 4.2 4.5 4.8 5.1
5.4 5.7 6 6.3 6.6 6.9 7.2 7.5...
        7.8 8.1 8.4 8.7 9 9.3 9.6 9.9   10.2    10.5    10.8    11.1    11.4
11.7    12  12.3    12.6...
        12.9    13.2    13.5    13.8    14.1    14.4    14.7    15  15.3
15.6    15.9    16.2    16.5...
        16.8    17.1    17.4    17.7    18  18.3    18.6    18.9    19.2
19.5    19.8    20.1    20.4... %yaw açısı
        20.7    21      21.3    21.6    21.9    22.2    22.5    22.8    23.1
23.4    23.7    24  24.3...
        24.6    24.9    25.2    25.5    25.8    26.1    26.4    26.7    27
27.3    27.6    27.9    28.2...
        28.5    28.8    29.1    29.4    29.7    30];
Fz=[0       133.7589351   82.74294026   74.64174154   71.26484437   69.52549217
68.42523905 67.63377456 66.9783241...
    66.48248249 66.08991312 65.79834717 65.61934745 65.59829345 65.73491809
65.92304688 66.13476103...
    66.37253714 66.62300917 66.90602254 67.19769096 67.52143762 67.82191537
68.13851428 68.51188783...
    68.92523565 69.3697002   69.78864116 70.25231834 70.70913787 71.18242814
71.63270061 72.0570112...
    72.48523408 72.92925213 73.39887623 73.84611556 74.30590959 74.81410041
75.28810211 75.78195927...
    76.28068376 76.78292234 77.30228893 77.82154964 78.31998059 78.82151915
79.30196859 79.76325116...  %Direnç Değerleri
```

Motion-Induced Energy Requirement Algorithm Code (Continued)

```
    80.24518785 80.69375653 81.11672463 81.53358458 81.97404283 82.43217311
82.90432767 83.417629...
    83.94211071 84.44339627 84.9310108  85.41165472 85.87862245 86.34031916
86.81091455 87.29444267...
    87.76473118 88.25450114 88.74941174 89.20313551 89.59999304 89.97825349
90.36594782 90.79612164...
    91.25922181 91.73939335 92.23221007 92.77579457 93.34487083 93.83968085
94.34980266 94.77666435...
    95.22506035 95.64269267 96.09421827 96.57291292 97.1437758  97.68900056
98.29684444 98.86693446...
    99.4682939  100.0923338 100.7018382 101.3123568 101.9196549 102.5244387
103.1471377 103.7619397...
    104.3660113 104.9810744 105.5846869 106.1966374 ];

psi=0:1:30;
FF=70.12-1.116.*psi+0.09941.*psi.^2+0.008201.*psi.^3-
0.0006116.*psi.^4+1.05e-05.*psi.^5;
F=@(psi) 70.12-1.116*psi+0.09941*psi^2+0.008201*psi^3-0.0006116*psi^4+1.05e-
05*psi^5; % direncin yaw açısı ile değişimi denklemi
d=[676  211 4465    4266    5236    3684    4342    4010    123 154 94]; %
yörünge uzunlukları
yaw=[29.54  29.56   15.8    13.46   10.03   13.73   6.5 17.84   17.2    26.3
0]; % yaw açısı
E=0;
V=1.5; %rov hızı
t=0;
plot(Angle,Fz);
hold on
plot(psi,FF)
l=0.3; %rov yarıgenişlik
    for i=1:11
        a=yaw(1,i);
        t=t+d/V+30;
        E=E+F(0)*(d(1,i)/1000)+F(i)*l; %Toplam enerji [J]
    end

    E %toplam enerji [J]
```

LQR Initialization Code

```
clc
clear
close all

spd = sin(0);        % Desired yaw angle is decided to be 0 deg = 0 rad s and
c represend sine and cosine functions
cpd = cos(0);
m = 10;              % Mass of the AUV
g = 9.81;            % Gravitational acceleration
uxd = 0;            % AUV is decided to be on hovering condition so no input
on x and y directions
uyd = 0;
uzd = 0;           % Vertical input is designed with respect to the balanced
lifting force and weight
Ix = 0.12;           % Inertia around x-axis
Iy = 0.16;           % Inertia around y-axis
Iz = 0.12;           % Inertia around z-axis

A = [0 0 0 1 0 0 0 0 0 0 0 0;
     0 0 0 0 1 0 0 0 0 0 0 0;
     0 0 0 0 0 1 0 0 0 0 0 0;
     0 0 0 0 0 0 uzd*spd/m uzd*cpd/m (-uxd*spd-uyd*cpd)/m 0 0 0;
```

LQR Initialization Code(Continued)

```
     0 0 0 0 0 0 -uzd*cpd/m uzd*spd/m (uxd*cpd-uyd*spd)/m 0 0 0;
     0 0 0 0 0 0 uyd/m -uxd/m 0 0 0 0;
     0 0 0 0 0 0 0 0 0 1 0 0;
     0 0 0 0 0 0 0 0 0 0 1 0;
     0 0 0 0 0 0 0 0 0 0 0 1;
     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];      % State-space matrices of the system for
decided operating points

B = [0 0 0 0 0 0;
     0 0 0 0 0 0;
     0 0 0 0 0 0;
     cpd/m -spd/m 0 0 0 0;
     spd/m cpd/m 0 0 0 0;
     0 0 1/m 0 0 0;
     0 0 0 0 0 0;
     0 0 0 0 0 0;
     0 0 0 0 0 0;
     0 0 0 Ix/m 0 0;
     0 0 0 0 Iy/m 0;
     0 0 0 0 0 Iz/m];

C = [1 0 0 0 0 0 0 0 0 0 0 0;
     0 1 0 0 0 0 0 0 0 0 0 0;
     0 0 1 0 0 0 0 0 0 0 0 0;
     0 0 0 0 0 0 1 0 0 0 0 0;
     0 0 0 0 0 0 0 1 0 0 0 0;
     0 0 0 0 0 0 0 0 1 0 0 0];

Co = ctrb(A,B);     % Controllability matrix
Rc = rank(Co);

if Rc == 12
    disp 'System is controllable'
else
    disp 'System is not controllable'
end

O = obsv(A,C);      % Observability matrix
Ro = rank(O);
if Ro == 12
    disp 'System is observable'
else
    disp 'System is not observable'
end

Q = eye(12);       % Weights of the states designed for mainly regulating z
coordinate
Q(1,1) = 100;
Q(2,2) = 100;
Q(3,3) = 10000;
Q(4,4) = 10;
Q(5,5) = 10;
Q(6,6) = 100;
Q(7,7) = 100;
Q(8,8) = 100;
Q(9,9) = 10;
Q(10,10) = 100;
Q(11,11) = 100;
Q(12,12) = 10;
```

```
R = eye(6);      % Weights of the inputs are designed mainly on penalizing
linear inputs
R(1,1) = 0.01;
R(2,2) = 0.01;
R(3,3) = 0.01;
R(4,4) = 0.1;
R(5,5) = 0.1;
R(6,6) = 0.1;
K = lqr(A,B,Q,R); % Gain matrix for LQR for u = -K*x

e1 = eigs(A);     % Eigenvales of free system
e2 = eigs(A-B*K); % Eigenvalues of controlled system

real_e1 = real(e1);
imag_e1 = imag(e1);

real_e2 = real(e2);
imag_e2 = imag(e2);

plot(real_e1,imag_e1,'o',real_e2,imag_e2,'+')   % Eigenvalue Plot
legend('eigenvalues of the free system','eigenvalues of the controlled
system')
```

LQR AUV Model Function

```
function dstate = fcn(state, U, m, Ix, Iy, Iz, S)

x = state(1);
y = state(2);
z = state(3);
dx = state(4);
dy = state(5);
dz = state(6);
phi = state(7);
theta = state(8);
psi = state(9);
dphi = state(10);
dtheta = state(11);
dpsi = state(12);

ux = U(1);
uy = U(2);
uz = U(3);
T_phi = U(4);
T_theta= U(5);
T_psi = U(6);

F = [S;S;S];

ddx = (F(1)/m)+((ux)*cos(psi)*cos(theta)+(uy)*(cos(psi)*sin(theta)*sin(phi)-
sin(psi)*cos(phi))+(uz)*(cos(psi)*cos(phi)*sin(theta)+sin(psi)*sin(phi)))/m;
% Dynamic equations of the system with external force of two sinusoidal waves
represented in global referernce frame
ddy                                                                    =
(F(2)/m)+((ux)*sin(psi)*cos(theta)+(uy)*(sin(psi)*sin(theta)*sin(phi)+cos(ps
i)*cos(phi))+(uz)*(sin(psi)*cos(phi)*sin(theta)-cos(psi)*sin(phi)))/m;
ddz                                  =                                 (F(3)/m)+(-
(ux)*sin(theta)+(uy)*cos(theta)*sin(phi)+(uz)*cos(theta)*cos(phi))/m;
ddphi = T_phi/Ix;
ddtheta = T_theta/Iy;
ddpsi = T_psi/Iz;
```

```
LQR AUV Model Function (Continued)

dstate   =   [dx;dy;dz;ddx;ddy;ddz;dphi;dtheta;dpsi;ddphi;ddtheta;ddpsi];   %
Derivative of the state as output of the block

end
```

LQR Controller Function

```
function u = fcn(R, x_meas, K)

u = K*(R-x_meas); % Control action by designed LQR

end
```

LQR Reference Function

```
function R = fcn(t)

R = [0;0;2;0;0;0;0;0;0;0;0;0];      % Reference for this study is a constant
position of 2 meters in depth and all other states are desired to be zero

End
```

LQR + SMC Reference Generator

```
function state_d  = fcn(t)

x_d = 0.2*cos(pi*t/50);
dx_d = -(0.2*pi/50)*sin(pi*t/50);

y_d = 0.2*sin(pi*t/50);
dy_d = (0.2*pi/50)*cos(pi*t/50);

z_d = 2;
dz_d = 0;

phi_d = 0;
dphi_d = 0;

theta_d = 0;
dtheta_d = 0;

psi_d = 0;
dpsi_d = 0;

state_d                                                                    =
[x_d;dx_d;y_d;dy_d;z_d;dz_d;phi_d;dphi_d;theta_d;dtheta_d;psi_d;dpsi_d];
end
```

LQR + SMC External Forces

```
function F = fcn(t)

Fx = 1.5*sin(1.5*t);
```

```
LQR + SMC External Forces (Continued)

Fy = 1.5*sin(1.5*t);
Fz = 0;

F = [Fx;Fy;Fz];

end
```

LQR + SMC Sliding Surfaces

```
function [dS, ux, uy] = fcn(S, E, F, t, state, m, Ly, Lx, Kx, Ky, Mx, My)

ddx_d = -(0.2*pi/50)*(pi/50)*cos(pi*t/50);
ddy_d = -(0.2*pi/50)*(pi/50)*sin(pi*t/50);

phi = state(7);
theta = state(9);
psi = state(11);

ex = E(1);
ey = E(3);

dex = E(2);
dey = E(4);

sx = S(1);
sy = S(2);

fx = F(1);
fy = F(2);

% ux = (m*(-Lx*dex)+ddx_d-Kx*sx-Mx*sign(sx))/(cos(psi)*cos(theta));
% uy = (m*(-Ly*dey)+ddy_d-Ky*sy-My*sign(sy))/(cos(psi)*cos(phi));

ux = (m*(-Lx*dex)+ddx_d-Kx*sx-Mx*(sx/(abs(sx)+0.3)))/(cos(psi)*cos(theta));
uy = (m*(-Ly*dey)+ddy_d-Ky*sy-My*(sy/(abs(sy)+0.3)))/(cos(psi)*cos(phi));

dSx = ((ux*(cos(psi)*cos(theta))+fx)/m)+Lx*dex-ddx_d;
dSy = ((uy*(cos(psi)*cos(phi))+fy)/m)+Ly*dey-ddy_d;

dS = [dSx;dSy];

end
```

LQR + SMC Initialization Code

```
clc
clear
close all

m = 5;        % Mass of the AUV
g = 9.81;     % Gravitational acceleration
Ix = 0.12;    % Inertia around x-axis
Iy = 0.16;    % Inertia around y-axis
Iz = 0.12;    % Inertia around z-axis

%% LQR For [z;phi;theta;psi]
```

LQR + SMC Initialization Code(Continued)

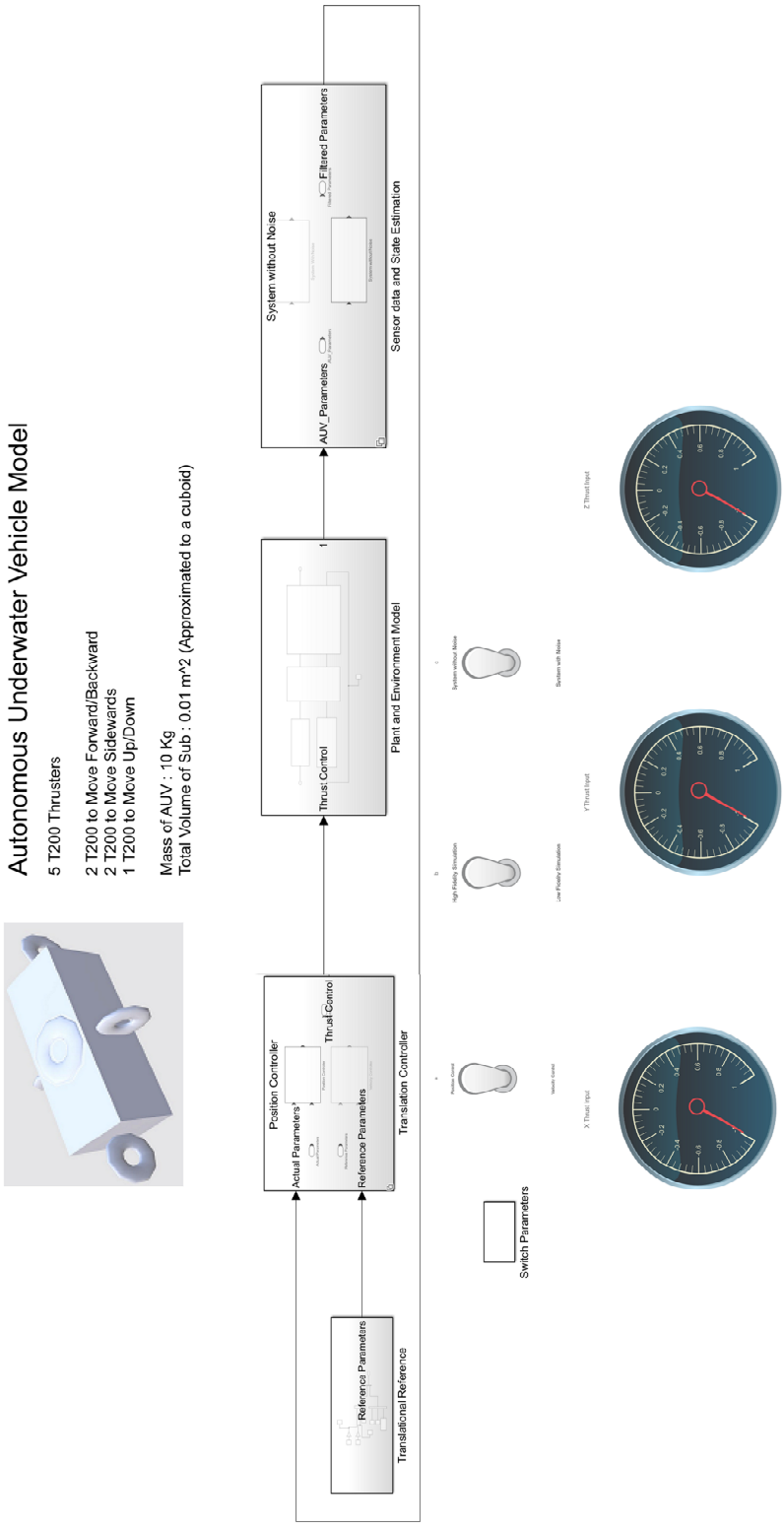
```
A = [0 1 0 0 0 0 0 0;
     0 0 0 0 0 0 0 0;
     0 0 0 1 0 0 0 0;
     0 0 0 0 0 0 0 0;
     0 0 0 0 0 1 0 0;
     0 0 0 0 0 0 0 0;
     0 0 0 0 0 0 0 1;
     0 0 0 0 0 0 0 0];  % ux = uy = 0 is done for operating point approximation

B = [0 0 0 0;
     1/m 0 0 0;
     0 0 0 0;
     0 1/Ix 0 0;
     0 0 0 0;
     0 0 1/Iy 0;
     0 0 0 0
     0 0 0 1/Iz];

C = [1 0 0 0 0 0 0 0;
     0 0 1 0 0 0 0 0;
     0 0 0 0 1 0 0 0;
     0 0 0 0 0 0 1 0];

Co = ctrb(A,B);     % Controllability matrix
Rc = rank(Co);
O = obsv(A,C);      % Observability matrix
Ro = rank(O);

Q = eye(8);
Q(1,1) = 100;
Q(2,2) = 100;
Q(3,3) = 0.1;
Q(5,5) = 0.1;
Q(7,7) = 0.1;

R = eye(4);

K = lqr(A,B,Q,R); % Gain matrix for LQR for u = -K*x

e1 = eigs(A);     % Eigenvales of free system
e2 = eigs(A-B*K); % Eigenvalues of controlled system

real_e1 = real(e1);
imag_e1 = imag(e1);

real_e2 = real(e2);
imag_e2 = imag(e2);

plot(real_e1,imag_e1,'o',real_e2,imag_e2,'+')   % Eigenvalue Plot
legend('eigenvalues of the free system','eigenvalues of the controlled
system')

%% SMC for [x;y]

Lx = 100;    % Lambda velue of sliding variable Sx
Ly = 100;    % Lambda velue of sliding variable Sy

Kx = 50;     % K velue of control on sliding variable Sx
Ky = 5;     % K velue of control on sliding variable Sx

Mx = 700;    % Disturbance attenuation value of Mu on sliding variable Sx
My = 700;    % Disturbance attenuation value of Mu on sliding variable Sy
```

LQR + SMC AUV Model

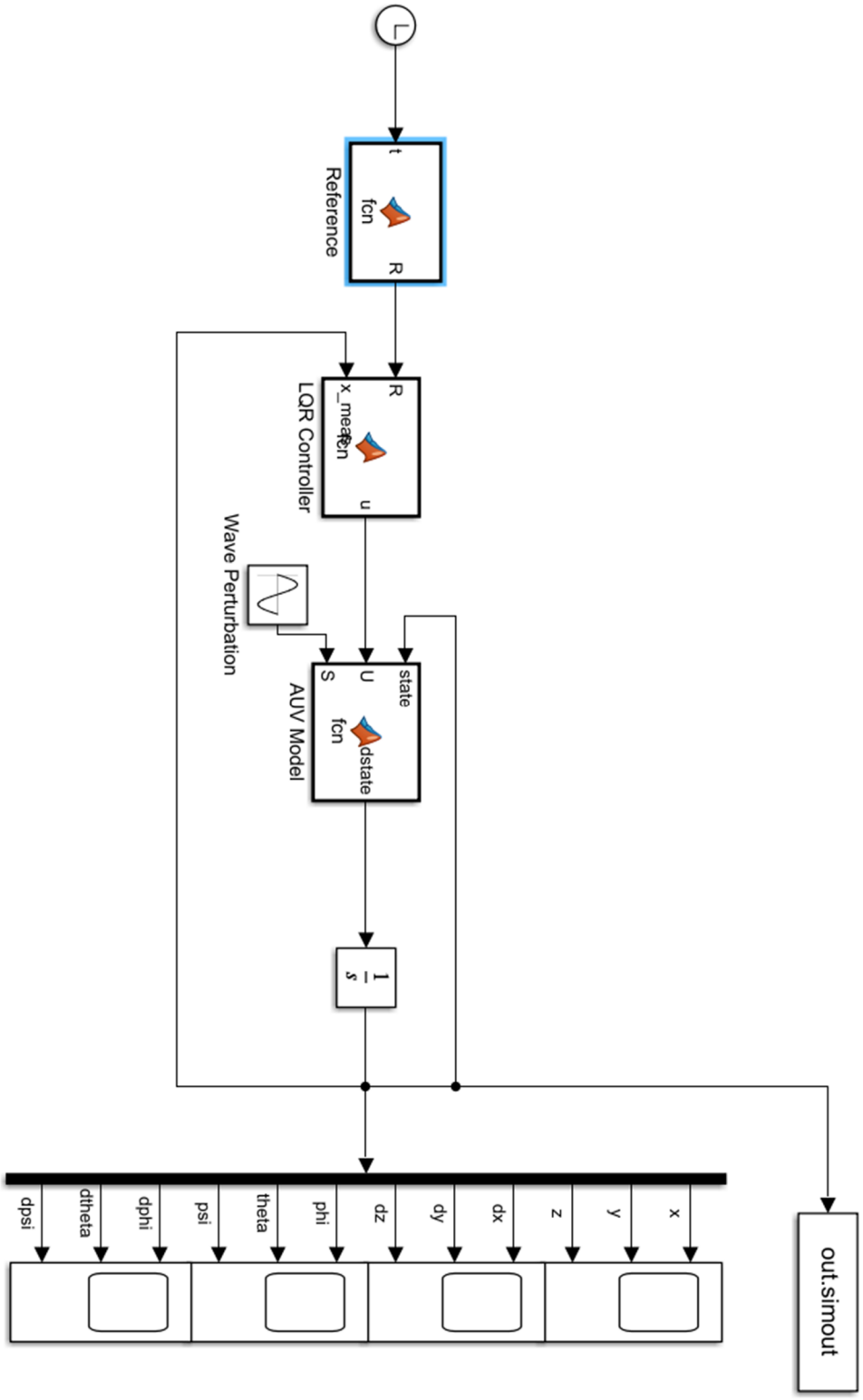
```
function dstate = fcn(state, U, F, m, Ix, Iy, Iz, g)

x = state(1);
dx = state(2);
y = state(3);
dy = state(4);
z = state(5);
dz = state(6);
phi = state(7);
dphi = state(8);
theta = state(9);
dtheta = state(10);
psi = state(11);
dpsi = state(12);

ux = U(1);
uy = U(2);
uz = U(3);
T_phi = U(4);
T_theta = U(5);
T_psi = U(6);

ddx = (F(1)/m)+(ux)*cos(psi)*cos(theta)+(uy)*(cos(psi)*sin(theta)*sin(phi)-
sin(psi)*cos(phi))+(uz)*(cos(psi)*cos(phi)*sin(theta)+sin(psi)*sin(phi))/m;
ddy                                                                    =
(F(2)/m)+(ux)*sin(psi)*cos(theta)+(uy)*(sin(psi)*sin(theta)*sin(phi)+cos(psi
)*cos(phi))+(uz)*(sin(psi)*cos(phi)*sin(theta)-cos(psi)*sin(phi))/m;
ddz                                  =                                 (F(3)/m)+(-
(ux)*sin(theta)+(uy)*cos(theta)*sin(phi)+(uz)*cos(theta)*cos(phi))/m;
ddphi = T_phi/Ix;
ddtheta = T_theta/Iy;
ddpsi = T_psi/Iz;

dstate   =   [dx;ddx;dy;ddy;dz;ddz;dphi;ddphi;dtheta;ddtheta;dpsi;ddpsi];   %
Derivative of the state as output of the block

end
```

Appendix 4 Simulink Models

SCENARIO 1: REFERENCE TRAJECTORY INPUT AND PID CONTROL



SCENARIO 2: LINEAR QUADRATIC REGULATOR CONTROLLER DESIGN



SCENARIO: 3 SLIDING MODE AND LQR HYBRID CONTROL ARCHITECTURE

