

# Design and Experimental Realization of Adaptive Control Schemes for an Autonomous Underwater Vehicle

*Dissertation submitted in partial fulfillment  
of the requirements of the degree of*

***Doctor of Philosophy***

**in**

**Electrical Engineering**

*by*

***Raja Rout***

**(512EE104)**

*based on research carried out  
under the supervision of*

***Prof. Bidyadhar Subudhi***



DEPARTMENT OF ELECTRICAL ENGINEERING  
NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA  
JANUARY 2017



## CERTIFICATE OF EXAMINATION

09/01/2017

Roll Number : 512EE104

Name: *Raja Rout*

Title of Dissertation: *Design and Experimental Realization of Adaptive Control Schemes for an Autonomous Underwater Vehicle*

We the below signed, after checking the dissertation mentioned above and the official record book(s) of the student, hereby state our approval of the dissertation submitted in partial fulfillment of the requirements of the degree of *Doctor of Philosophy in Electrical Engineering at National Institute of Technology Rourkela*. We are satisfied with the volume, quality, correctness, and originality of the work.

---

Prof. D. R. K. Parhi  
(Member, DSC)

---

Prof. U. C. Pati  
(Member, DSC)

---

Prof. A. K. Naskar  
(Member, DSC)

---

Prof. A. K. Panda  
(Chairperson, DSC)

---

Prof. B. Subudhi  
(Supervisor)

---

Prof. A. Gupta  
(External Examiner)

---

Prof. J. K. Satapathy  
(Head of the Department)



## CERTIFICATE

This is to certify that the work presented in the dissertation entitled “**Design and Experimental Realization of Adaptive Control Schemes for an Autonomous Underwater Vehicle**” submitted by Raja Rout, Roll Number 512EE104, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of Doctor of Philosophy in Electrical Engineering. Neither this dissertation nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

---

Prof. Bidyadhar Subudhi  
(Supervisor)

## Declaration of Originality

I, Raja Rout, Roll Number 512EE104 hereby declare that this dissertation entitled **“Design and Experimental Realization of Adaptive Control Schemes for an Autonomous Underwater Vehicle”** presents my original work carried out as a doctoral student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections Reference or Bibliography. I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

RAJA ROUT

## ACKNOWLEDGEMENTS

First and foremost, I am truly indebted to my supervisor Prof. Bidyadhar Subudhi for his guidance and unwavering confidence through my study, without which this thesis would not be in its present form. I also thank him for gracious encouragement throughout the work.

I express my sincere gratitude to the Doctoral Scrutiny Committee chairman Prof. A. K. Panda and its members Prof. D. R. K. Parhi, Prof. U. C. Pati, Prof. A. K. Naskar for their suggestion to improve the work. I am also very much obliged to Head of the Department of Electrical Engineering, Prof. J. K. Satapathy for providing possible facilities. I also express my earnest thanks to Prof. Sandip Ghosh for his valuable suggestion and thanks to other faculty members in the department. I would also like to thank Sahadev Swain and Budu Oram for their assistance during my stay at Control and Robotics laboratory.

I would especially like to thank my colleague Subhasish, for his support in developing the AUV. I also thank to Biranchi, Chhavi and Shivam for their enjoyable moment during experimental tests at swimming pool. I thank to my seniors Dushmanta Ku. Dash, Shantanu Pradhan, Basant Ku. Sahu, Sathyam Bonala, Srinibas Bhuyan and Raseswari Pradhan for their inspiration and suggestion which helped me to complete this thesis.

My wholehearted gratitude to my beloved parents Smt. Promodini Rout and Sri. Ashok Kumar Rout for their encouragement and support. Thank you for believing in me.

RAJA ROUT

# Abstract

Research on Autonomous Underwater Vehicle(AUV) has attracted increased attention of control engineering community in the recent years due to its many interesting applications such as in Defense organisations for underwater mine detection, region surveillance, oceanography studies, oil/gas industries for inspection of underwater pipelines and other marine related industries. However, for the realization of these applications, effective motion control algorithms need to be developed. These motion control algorithms require mathematical representation of AUV which comprises of hydrodynamic damping, Coriolis terms, mass and inertia terms etc. To obtain dynamics of an AUV, different analytical and empirical methods are reported in the literature such as tow tank test, Computational Fluid Dynamics (CFD) analysis and on-line system identification. Among these methods, tow-tank test and CFD analysis provide white-box identified model of the AUV dynamics. Thus, the control design using these methods are found to be ineffective in situation of change in payloads of an AUV or parametric variations in AUV dynamics. On the other hand, control design using on-line identification, the dynamics of AUV can be obtained at every sampling time and thus the aforesaid parametric variations in AUV dynamics can be handled effectively.

In this thesis, adaptive control strategies are developed using the parameters of AUV obtained through on-line system identification. The proposed algorithms are verified first through simulation and then through experimentation on the prototype AUV. Among various motion control algorithms, waypoint tracking has more practical significance for oceanographic surveys and many other applications. In order to

implement, waypoint motion control schemes, Line-of-Sight (LoS) guidance law can be used which is computationally less expensive. In this thesis, adaptive control schemes are developed to implement LoS guidance for an AUV for practical realization of the control algorithm.

Further, in order to realize the proposed control algorithms, a prototype AUV is developed in the laboratory. The developed AUV is a torpedo-shaped in order to experience low drag force, underactuated AUV with a single thruster for forward motion and control planes for angular motion. Firstly, the AUV structure such as nose profile, tail profile, hull section and control planes are designed and developed. Secondly, the hardware configuration of the AUV such as sensors, actuators, computational unit, communication module etc. are appropriately selected. Finally, a software framework called Robot Operating System (ROS) is used for seamless integration of various sensors, actuators with the computational unit. ROS is a software platform which provides right platform for the implementation of the control algorithms using the sensor data to achieve autonomous capability of the AUV.

In order to develop adaptive control strategies, the unknown dynamics of the AUV is identified using polynomial-based Nonlinear Autoregressive Moving Average eXogenous (NARMAX) model structure. The parameters of this NARMAX model structure are identified online using Recursive Extended Least Square (RELS) method. Then an adaptive controller is developed for realization of the LoS guidance law for an AUV. Using the kinematic equation and the desired path parameters, a Lyapunov based backstepping controller is designed to obtain the reference velocities for the dynamics. Subsequently, a self-tuning PID controller is designed for the AUV to track these reference velocities. Using an inverse optimal control technique, the gains of the self-tuning PID controller are tuned on-line. Although, this algorithm is computationally less expensive but there lie issues such as actuator constraints and state constraints which need to be resolved in view of practical realization of the control law. It is also observed that the proposed NARMAX structure of the AUV consists of redundant

regressor terms.

To alleviate the aforesaid limitations of the Inverse optimal self-tuning control scheme, a constrained adaptive control scheme is proposed that employs a minimum representation of the NARMAX structure (MR-NARMAX) for capturing AUV dynamics. The regressors of the MR-NARMAX structure are identified using Forward Regressor Orthogonal Least Square algorithm. Further, the parameters of this MR-NARMAX model structure of the AUV are identified at every sampling time using RELS algorithm. Using the desired path parameters and the identified dynamics, an error objective function is defined which is to be minimized. The minimization problem where the objective function with the state and actuator constraints is formulated as a convex optimization problem. This optimization problem is solved using quadratic programming technique. The proposed MR-NARMAX based adaptive control is verified in the simulation and then on the prototype AUV. From the obtained results it is observed that this algorithm provides successful tracking of the desired heading. But, the proposed control algorithm is computational expensive, as an optimization problem is to be solved at each sampling instant.

In order to reduce the computational time, an explicit model predictive control strategy is developed using the concept of multi-parametric programming. A Lyapunov based backstepping controller is designed to generate desired yaw velocity in order to steer the AUV towards the desired path. This explicit model predictive controller is designed using the identified NARMAX model for tracking the desired yaw velocity. The proposed explicit MPC algorithm is implemented first in simulation and then in the prototype AUV. From the simulation and experimental results, it is found that this controller has less computation time and also it considers both the state and actuator constraints whilst exhibiting good tracking performance.

**Key words:** AUV, Adaptive control, NARMAX, Self-tuning control, Line-of-Sight, RELS, UUV.



# Contents

<b>Abstract</b>	<b>i</b>
<b>List of symbols and acronyms</b>	<b>viii</b>
<b>List of figures</b>	<b>xiii</b>
<b>List of tables</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Autonomous Underwater Vehicle and its Applications . . . . .	1
1.2 Guidance Algorithms . . . . .	4
1.3 Adaptive Control Structure . . . . .	9
1.4 Motivations of the present work . . . . .	12
1.5 Objectives of the thesis . . . . .	12
1.6 Organization of the thesis . . . . .	12
<b>2 Development of a Prototype AUV</b>	<b>14</b>
2.1 Mechanical structure of AUV . . . . .	14
2.2 Hardware Configuration . . . . .	20
2.3 Software framework . . . . .	24
2.4 Chapter Summary . . . . .	27

<b>3</b>	<b>LoS Guidance Law using Inverse Optimal Self-Tuning Adaptive Controller</b>	<b>28</b>
3.1	Problem Statement . . . . .	29
3.2	Identification of the AUV dynamics . . . . .	31
3.3	Development of the Adaptive Inverse-Optimal PID Controller . . . . .	34
3.3.1	Control design for kinematics . . . . .	35
3.3.2	Control design for dynamics . . . . .	39
3.4	Stability Analysis . . . . .	41
3.5	Results and Discussion . . . . .	43
3.5.1	Simulation Results . . . . .	43
3.5.2	Experimental Results . . . . .	46
3.6	Chapter Summary . . . . .	51
<b>4</b>	<b>Constrained Self-Tuning Adaptive Controller for an AUV with MR-NARMAX structure</b>	<b>52</b>
4.1	Problem Statement . . . . .	53
4.2	Identification of the AUV Dynamics . . . . .	54
4.3	NARMAX Self-Tuning Controller Design . . . . .	57
4.4	Results and Discussion . . . . .	62
4.4.1	Simulation Results . . . . .	62
4.4.2	Experimental Results . . . . .	65
4.5	Chapter Summary . . . . .	71
<b>5</b>	<b>Explicit model predictive control design for an AUV</b>	<b>72</b>
5.1	Problem Statement . . . . .	73
5.2	Explicit Control Design . . . . .	74
5.3	Results and Discussion . . . . .	81
5.3.1	Simulation Results . . . . .	81
5.3.2	Experimental Results . . . . .	84

---

5.4	Chapter Summary . . . . .	89
<b>6</b>	<b>Conclusion and Suggestion for Future Work</b>	<b>90</b>
6.1	Overall Summary of the thesis . . . . .	90
6.1.1	Contributions of the Thesis . . . . .	91
6.2	Suggestions for the future work . . . . .	92
<b>A</b>	<b>Kinematics and Dynamics of an AUV</b>	<b>94</b>
A.1	Kinematics . . . . .	94
A.2	Dynamics . . . . .	96
<b>B</b>	<b>Solution to Multiparameteric Quadratic programming</b>	<b>98</b>
	<b>References</b>	<b>101</b>
	<b>Publications from this thesis</b>	<b>107</b>



# List of symbols and acronyms

## List of symbols

$\{I\}$	:	Earth-fixed frame
$\{B\}$	:	Body-fixed frame
$x_k, y_k, z_k$	:	Linear position along x, y and z axis in $\{I\}$
$\phi_k, \theta_k, \psi_k$	:	Angular position along x, y and z axis in $\{I\}$
$u_c, v_k, w_k$	:	Linear velocities along x, y and z axis in $\{B\}$
$p_k, q_k, r_k$	:	Angular velocities along x, y and z axis in $\{B\}$
$\eta_1 \in \mathbb{R}^3$	:	Linear position vector $[x_k \ y_k \ z_k]^T$
$\eta_2 \in \mathbb{R}^3$	:	Angular position vector $[\theta_k \ \psi_k]^T$
$\eta \in \mathbb{R}^5$	:	Position vector $[\eta_1 \ \eta_2]^T$
$v_1 \in \mathbb{R}^2$	:	Linear velocity vector $[v_k \ w_k]^T$
$v_2 \in \mathbb{R}^2$	:	Angular velocity vector $[q_k \ r_k]^T$
$\nu \in \mathbb{R}^4$	:	Velocity vector $[v_1 \ v_2]^T$
$\delta_h$	:	Side slip angle in the heading motion
$\delta_d$	:	Angle of attack in the diving motion
$J \in \mathbb{R}^{5 \times 4}$	:	Transformation matrix from $\{I\}$ to $\{B\}$
$M \in \mathbb{R}^{4 \times 4}$	:	Mass Matrix
$C_r \in \mathbb{R}^{4 \times 4}$	:	Coriolis Matrix
$f_d \in \mathbb{R}^{4 \times 1}$	:	Damping force/moment
$r_s \in \mathbb{R}^{4 \times 1}$	:	Restoring force/moment
$\tau \in \mathbb{R}^{4 \times 1}$	:	Actuation inputs as thruster and/or control planes
$\delta_r$	:	Control Input for the Rudder plane
$\delta_s$	:	Control Input for the Stern plane

**List of acronyms**

AUV	:	Autonomous Underwater Vehicle
ROV	:	Remotely Operated Vehicle
LoS	:	Line-of-Sight
NARMAX	:	Nonlinear Autoregressive Moving Average eXogenous
RELS	:	Recursive Extended Least Square
FROLS	:	Forward Regressor Orthogonal Least Square
CSTC	:	Constrained Self-Tuning Control
MPC	:	Model Predictive Control
CROC	:	Constrained Robust Optimal Control
mp-QP	:	multiparametric Quadratic Programming
ROS	:	Robot Operating System

# List of Figures

1.1	Examples of Commercial AUV's . . . . .	2
1.2	Examples of Military application AUV's . . . . .	3
1.3	AUV's used for oceanography and marine studies . . . . .	3
1.4	Definition of AUV frames . . . . .	3
1.5	Trajectory tracking by an AUV . . . . .	5
1.6	Path following task by an AUV . . . . .	6
1.7	Way-point tracking by an AUV . . . . .	7
1.8	Line-of-Sight guidance by an AUV . . . . .	8
1.9	Combined Control and Learning architecture . . . . .	10
1.10	Separate Control and Learning architecture . . . . .	11
1.11	Augmented Control architecture . . . . .	11
2.1	Torpedo shaped AUVs . . . . .	15
2.2	Non-Torpedo shaped AUVs . . . . .	16
2.3	Design parameters of the Myring profile . . . . .	16
2.4	Nose section of the developed AUV . . . . .	17
2.5	Tail section of the developed AUV . . . . .	18
2.6	Hardware architecture of the AUV . . . . .	21
2.7	Sensor units . . . . .	22
2.8	Actuation units . . . . .	23
2.9	Power supply unit . . . . .	23

2.10	Communication unit . . . . .	24
2.11	Prototype AUV developed at National Institute of Technology Rourkela	24
2.12	Different parts of the developed AUV . . . . .	25
2.13	Example of ROS structure considering Computer, Sensor and Actuator as the Nodes . . . . .	27
3.1	Structure of the proposed NARMAX model based self-tuning controller	30
3.2	NARMAX model structure for system identification [1] . . . . .	32
3.3	Desired LoS path . . . . .	34
3.4	Tracking of desired heading by INFANTE AUV . . . . .	44
3.5	Heading error . . . . .	44
3.6	Estimated yaw velocity as compared to actual yaw velocity . . . . .	45
3.7	Updation of the NARMAX parameters for yaw motion . . . . .	45
3.8	Actuation signal while tracking a desired heading . . . . .	45
3.9	Communication between ROS nodes . . . . .	46
3.10	Implementation of the self-tuning controller . . . . .	48
3.11	Tracking of desired heading . . . . .	48
3.12	Heading error . . . . .	49
3.13	Pitch rate of the AUV during path follow . . . . .	49
3.14	Yaw rate of the AUV during path follow . . . . .	49
3.15	Updation of the NARMAX parameters . . . . .	50
3.16	Updation of the controller gain parameters . . . . .	50
3.17	Rudder plane orientation while following a desired path . . . . .	50
3.18	Computational time required to generate the actuation signal . . . . .	51
4.1	Controller Structure for the LOS Guidance law . . . . .	53
4.2	Tracking of Line of Sight path . . . . .	57
4.3	Tracking of heading reference by INFANTE AUV . . . . .	63
4.4	Heading error while tracking the desired reference . . . . .	64



4.5	Estimated yaw velocity as compared to actual yaw velocity . . . . .	64
4.6	Updatation of MR-NARMAX parameters . . . . .	64
4.7	Actuation signal while tracking a desired heading . . . . .	65
4.8	Implementation of constrained adaptive control strategy in ROS . . . .	66
4.9	Implementation of the developed algorithm in the prototype AUV . . .	67
4.10	Tracking of desired heading by prototype AUV . . . . .	68
4.11	Heading error . . . . .	68
4.12	Comparison of estimated velocity and actual yaw velocity . . . . .	69
4.13	Comparison of estimated velocity and actual pitch velocity . . . . .	69
4.14	Updatation of the NARMAX parameters . . . . .	69
4.15	Computational performance of MR-NARMAX identification . . . . .	70
4.16	Control signal for rudder plane . . . . .	70
4.17	Controller computational performance . . . . .	70
5.1	Controller Structure for the LOS Guidance law . . . . .	73
5.2	Explicit control design for AUV . . . . .	74
5.3	Tracking of Line of Sight path . . . . .	76
5.4	Solution of explicit MPC for equation (5.33) with horizon $N = 4$ . . .	82
5.5	Tracking of desired heading by the developed AUV . . . . .	82
5.6	Heading error while tracking LoS path . . . . .	83
5.7	Yaw velocity . . . . .	83
5.8	Sway velocity . . . . .	83
5.9	Control signal for rudder plane . . . . .	84
5.10	Implementation of explicit MPC in ROS . . . . .	85
5.11	Solution of explicit MPC for equation (5.35) with horizon $N = 4$ . . .	86
5.12	Implementation of the explicit MPC control algorithm . . . . .	86
5.13	Following a desired yaw orientation . . . . .	87
5.14	Orientation error along yaw motion . . . . .	87

---

5.15	Yaw velocity while tracking the desire path . . . . .	87
5.16	Rudder input required to steer the AUV along LOS path . . . . .	88
5.17	Time taken to generate the actuation signal . . . . .	88
A.1	Frames to represent AUV motion . . . . .	94

# List of Tables

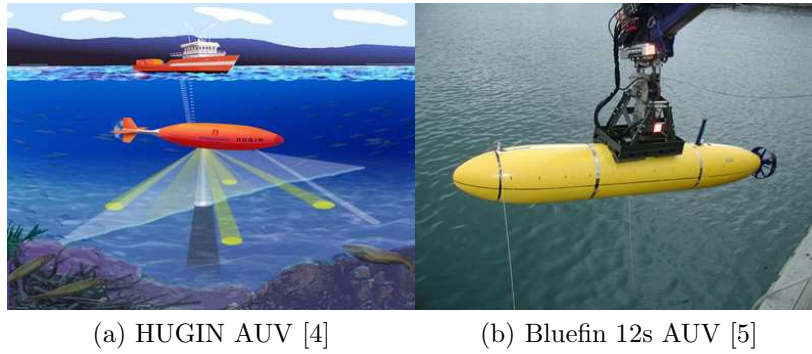
2.1	Nose profile of the developed AUV . . . . .	17
2.2	Tail profile of the developed AUV . . . . .	19
2.3	Design parameters of the control planes . . . . .	19
3.1	Simulation parameters . . . . .	44
3.2	Description of ROS nodes . . . . .	46
3.3	Description of ROS messages and its characteristics . . . . .	47
3.4	Parameters . . . . .	47
4.1	FROLS applied to INFANTE AUV heading motion . . . . .	62
4.2	Description of ROS nodes . . . . .	66
4.3	Description of ROS messages and its characteristics . . . . .	66
4.4	FROLS applied to AUV Heading motion . . . . .	66
5.1	Description of ROS nodes . . . . .	84
5.2	Description of ROS messages and its characteristics . . . . .	85
5.3	Comparison between various developed control algorithms . . . . .	89
A.1	AUV parameter definition . . . . .	97

# Chapter 1

## Introduction

### 1.1 Autonomous Underwater Vehicle and its Applications

As per National Oceanic and Atmospheric Administration (NOAA), it is known that the ocean covers about 71% of the earth surface. Only less than 5% of its ocean floor is explored and most of its regions are inaccessible for divers. In order to explore more about underwater environment and collection of information, underwater vehicles are deployed. Based on their operation, these vehicles can be broadly classified into two types Remotely Operated Vehicles (ROV) and Autonomous Underwater Vehicle (AUV). ROV is a remotely operated underwater robot which is connected to its base station through power cables and data cables. Through the tethered connection, the actuators and electronic equipments of the ROV is powered and a command signal is sent to the ROV from the base station. On the other hand, Autonomous Underwater Vehicle or Unmanned Underwater Vehicle (UUV) is an underwater robot which navigates autonomously in order to complete its assigned mission. These vehicles are equipped with sensors, actuators, power, communication and computational units which enable an AUV to attain autonomous capability. Unlike the Remotely Operated Vehicle (ROV), AUV is not tethered with the base station rather it collects the data during the mission execution and the data is retrieved once the mission is complete.



(a) HUGIN AUV [4]

(b) Bluefin 12s AUV [5]

Figure 1.1: Examples of Commercial AUV's

Not only in oceanographic studies, AUVs are also deployed for commercial and defence organizations. Some of its applications are discussed as follows.

- In commercial organization such as oil/gas industries, these AUVs are deployed for sea floor mapping and surveys which is necessary for the development of subsea infrastructure [2], [3]. Further, it can also be used for the leakage detection of pipeline or detection of cracks in underwater structure. These AUVs offer great benefits by replacing human operator thus avoiding the operation cost and risk in the extreme environment i.e. deep oceanic environment. Some of these AUVs which are generally used for commercial purpose are shown in Fig.1.1. HUGIN AUV Fig.1.1a has been developed by Kongsberg Maritime, Norway and Bluefin AUV Fig.1.1b by Bluefin Robotics, USA.
- For military applications, the AUVs such as Fig.1.2 are used for underwater mine countermeasure or search and salvage operations. It can also be employed in a protected area for surveillance of the region. Some of the AUVs which are known for their application in defense organization are AUV150 Fig.1.2a by CSIR-CMERI, India and ALISTER 100 Fig.1.2b by eca Robotics, USA.
- Apart from commercial and defense applications, research organizations related to oceanographic studies use these AUVs as a platform for the collection of data. In oceanographic environment, some of the places are inaccessible for human. Therefore, these AUVs are equipped with oceanographic sensors as payload, which acquire the desired information for researchers. Some of these

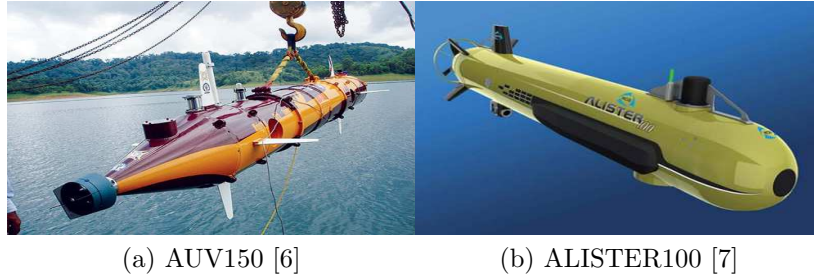


Figure 1.2: Examples of Military application AUV's

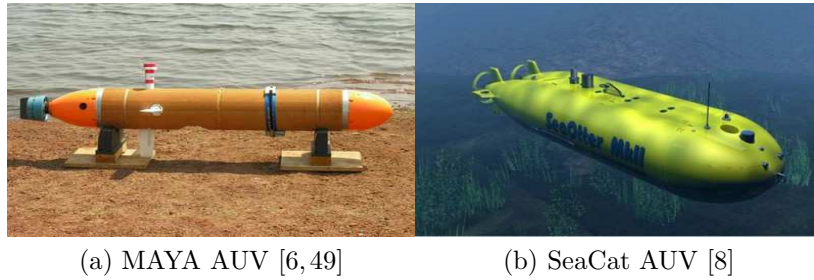


Figure 1.3: AUV's used for oceanography and marine studies

AUVs, which are used for marine research or environmental studies are MAYA AUV Fig.1.3a by NIO, India and SeaCat AUV Fig.1.3b by Atlas Elektronik, Germany.

AUVs also have immense applications in commercial, defense and oceanographic research organization and these applications motivate researchers to develop effective guidance algorithms. However, prior to develop a guidance algorithm, the knowledge of kinematics and dynamics of an AUV are necessary.

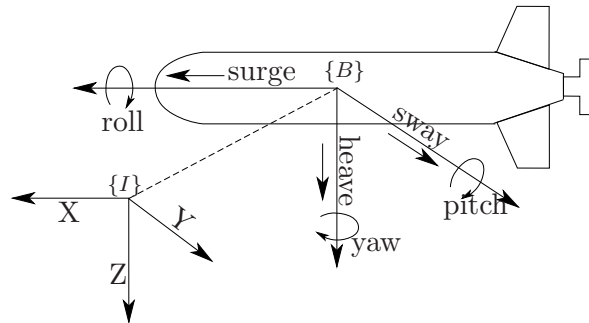


Figure 1.4: Definition of AUV frames

Referring to Fig.1.4, the velocities  $\nu = [v_1 \ v_2]^T \in \mathbb{R}^6$  are defined in body-fixed frame  $\{B\}$  along surge, sway, heave, roll, pitch and yaw motions, whereas the position of the AUV  $\eta = [\eta_1 \ \eta_2]^T \in \mathbb{R}^6$  is defined w.r.t earth-fixed frame  $\{I\}$ .  $\eta_1 \in \mathbb{R}^3$  and  $\eta_2 \in \mathbb{R}^3$  are the linear and angular position of the AUV in  $\{I\}$ . To observe the motion of the AUV from  $\{I\}$ , a transformation between  $\{B\}$  and  $\{I\}$  is necessary. So using the transformation matrix  $J \in \mathbb{R}^{6 \times 6}$  from [9], the expression for velocities in  $\{I\}$  is given by

$$\dot{\eta} = J(\eta) \nu \quad (1.1)$$

Equation (1.1) represents the kinematic description of the AUV, where  $\dot{\eta}$  is the velocity in  $\{I\}$ . Referring to [9], the dynamics of an AUV is given by

$$M\dot{\nu} + C_r(\nu) \nu + f_d(\nu) + r_s(\eta) = \tau, \quad (1.2)$$

where  $M$  is the mass matrix,  $C_r$  is the Coriolis matrix,  $f_d$  is the damping force and  $r_s$  is the restoring force.  $\tau$  represents the external input to the AUV. For detailed description of AUV kinematics and dynamics are provided in Appendix A. Considering the AUV kinematics (1.1) and dynamics (1.2), the guidance algorithm is developed.

Referring to various applications such as pipeline survey requires that AUV should follow a predefined path which is depicted as a pipeline. The region of surveillance in defense application requires that AUV should secure a region by moving along the perimeter of the specified region. Similarly in oceanographic studies, it is required that the AUV should collect the data at different waypoints. Likewise, most of the AUV applications can be addressed, if the AUV has the ability of following a path, tracking a trajectory or moving along a line-of-sight path.

## 1.2 Guidance Algorithms

From Section 1.1, it is described that guidance algorithms can be broadly categorized as (1) Trajectory tracking (2) Path following (3) Way-point tracking and (4) Line-of-Sight (LoS) path. In the literature, these guidance algorithms have been implemented for AUVs and the advantages and disadvantages of these algorithms are discussed as follows.

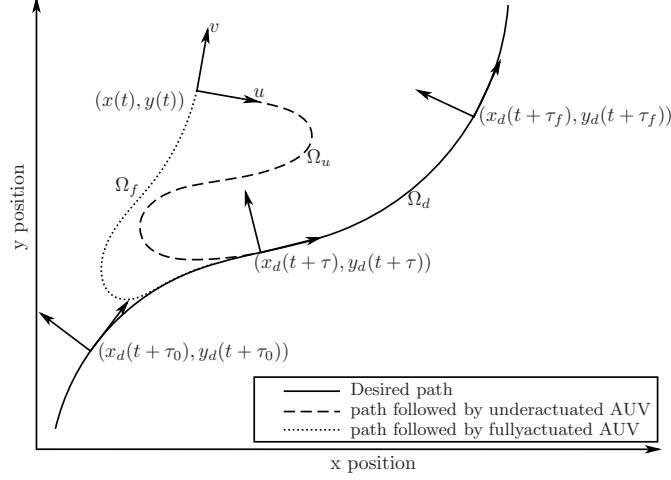


Figure 1.5: Trajectory tracking by an AUV

- **Trajectory Tracking:** In this guidance control problem, an AUV tracks a time-parameterized reference path. Referring to Fig. 1.5, the desired path  $\Omega_d(t)$  is parameterized as

$$x_d(t) = f_1(t) \quad (1.3)$$

$$y_d(t) = f_2(t) \quad (1.4)$$

where  $f_1(t)$  and  $f_2(t)$  are the desired path functions along  $x$  and  $y$  axes. An objective function can be chosen to minimize the distance error between the actual position of the AUV and the desired location at time  $\tau$  in the trajectory. Generally, the Lyapunov objective function is taken as

$$V = \|\eta(t) - \eta_d(t)\|_p, \quad (1.5)$$

where  $\eta(t)$  is the position of AUV and  $\eta_d(t)$  is the desired location in the trajectory  $\Omega_d(t)$ . However, effective tracking of a trajectory depends on whether the AUV is fully-actuated or under-actuated system. In the literature, trajectory tracking algorithms for fully-actuated AUVs as in [10], [11] are well established. But, in view of cost and weight of the actuator and energy requirement for long duration mission, fully-actuated AUVs are not desirable. In the other hand, de-



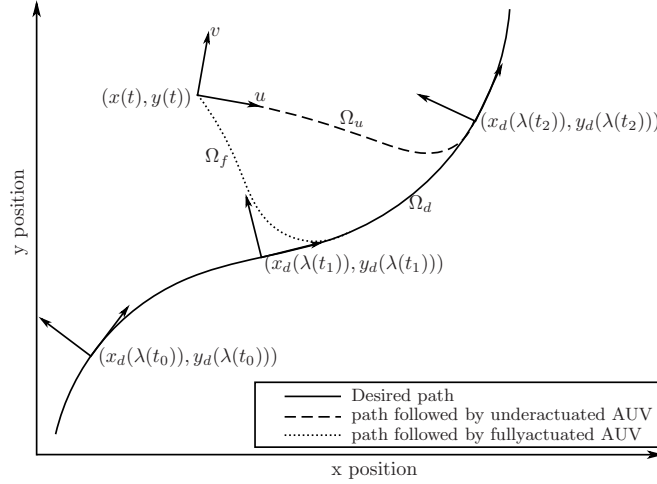


Figure 1.6: Path following task by an AUV

signing a tracking algorithm for an under-actuated AUV is difficult because most of the systems are not fully linearizable or exhibits non-holonomic constraints. Trajectory tracking algorithm for underactuated AUVs with initial position close to trajectory initial position is difficult to implement in practical scenario [12,13].

- **Path Following:** Like the trajectory tracking problem, the desired path  $\Omega_d(t)$  in the path following problem is not time parameterized. Rather, the desired path is parameterized using a variable  $\lambda$ . Referring to Fig.1.6, the desired path  $\Omega_d(t)$  is described as

$$x_d(t) = f_1(\lambda(t)) \quad (1.6)$$

$$y_d(t) = f_2(\lambda(t)) \quad (1.7)$$

where  $f_1(\lambda(t))$  and  $f_2(\lambda(t))$  denote the desired path functions. Referring to [14], [15], [16], [17], a virtual frame is designed which moves along the desired path  $\Omega_d(t)$  and the AUV is required to converge and follow the desired virtual frame  $S_1$ . Usually, a Serret-Frenet(S-F) frame is used as the virtual frame and referring to the literature the updatation of the S-F frame is given by

$$\dot{\lambda} = f(U, x_e, \psi_e), \quad (1.8)$$

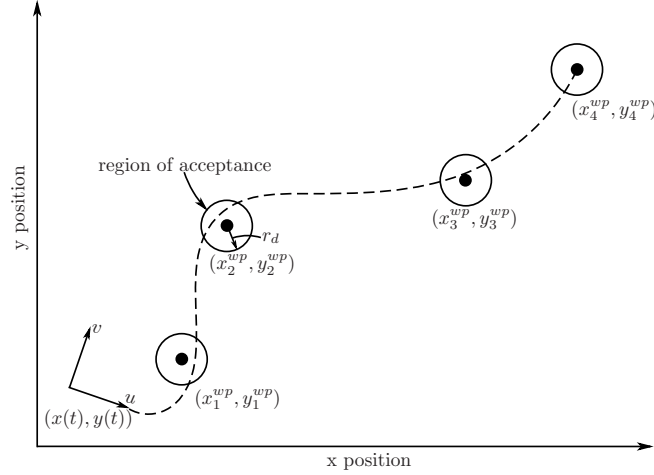


Figure 1.7: Way-point tracking by an AUV

where  $U = \sqrt{u^2 + v^2}$  is the net velocity of the AUV,  $\psi_e$  and  $x_e$  are the orientation and positional error between the S-F frame and AUV. Referring to Fig.1.6, a smoother convergence to the path is achieved as compared to trajectory tracking by fully-actuated as well as under-actuated AUVs. For the later case, actuation signals are less likely to achieve actuation saturation. Therefore, path-following algorithm is better suited for under-actuated AUVs. However, in view of practical realization of the algorithm in ocean environment another guidance algorithm i.e. way-point tracking can also be used in place of path following problem as discussed in [18].

- Way-point tracking: In this guidance system, the AUV tracks a given waypoint as shown in Fig.1.8. The present waypoint  $(x_{wp,i}, y_{wp,i})$  and previous waypoint  $(x_{wp,i-1}, y_{wp,i-1})$  is connected using a rectilinear path and the AUV has to follow the desired path  $\Omega_d$  as shown in the figure. The desired orientation while following the waypoints can be expressed as

$$\psi_d = \tan^{-1}(y_{wp,i} - y_{wp,i-1}, x_{wp,i} - x_{wp,i-1}). \quad (1.9)$$

The crosstrack heading error i.e.  $\psi_e$  as shown in Fig.1.8 is given as

$$\psi_e = \psi - \psi_d. \quad (1.10)$$

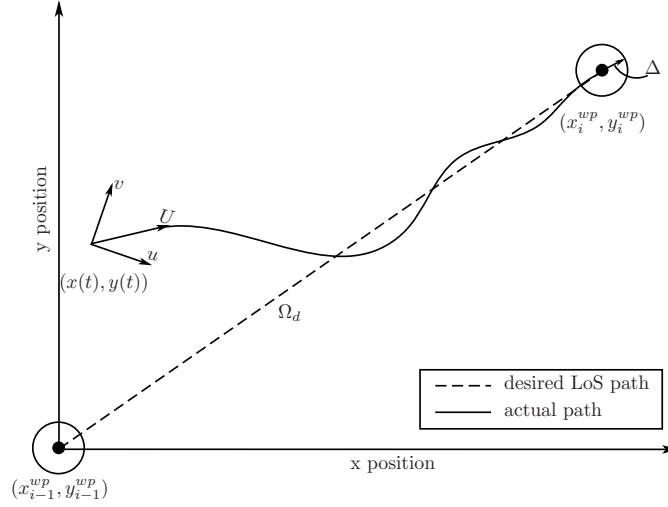


Figure 1.8: Line-of-Sight guidance by an AUV

A suitable Lyapunov candidate function can be chosen for minimizing this crosstrack error and thus required actuation signal can be obtained. In order to extend this LoS guidance system for multiple waypoints, the following condition can be imposed to switch the waypoints i.e.

$$\begin{aligned}
 \text{Step:1} \quad w_{p,e} &= \sqrt{(x - x_{wp,i})^2 + (y - y_{wp,i})^2} \\
 \text{Step:2} \quad &if(w_{p,e} \leq \Delta) \\
 &\quad then \\
 &\quad \quad i = i + 1 \\
 &\quad goto \text{Step:1.}
 \end{aligned} \tag{1.11}$$

Referring to literature [19], the LoS guidance or waypoint tracking algorithm is suitable for the practical scenario. For example in [20], these waypoints correspond to the location of plumes in the ocean environment.

For underactuated AUVs, trajectory tracking is difficult to realise because the non-linear AUV dynamics are not fully linearizable and the control signal reaches saturation very often [21–23], whereas path following and way-point guidance are parameterized irrespective of time. These motion control algorithms have much practical significance as compared to trajectory tracking. A LoS based guidance can be used for implemen-

tation of path following or waypoint guidance by simplifying as rectilinear paths as shown in Fig.1.8 or dubins path as in [19,24]. Thus various path planning algorithms can be developed to extend the LoS guidance for waypoint and path following implementation. Therefore, this work deals with the development of an control law for the implementation of LoS guidance algorithm.

## 1.3 Adaptive Control Structure

The dynamic equation (1.2) of an AUV comprises of mass, hydrodynamic damping, restoring and actuator terms. Amongst these terms, accurate measurement of damping terms is difficult to determine. However, it can be obtained analytically by approximating the AUV as an ellipsoid [25] or by using strip theory method [26] respectively. Amongst various shapes of AUV, if the designed AUV has standard torpedo shape then the theoretical drag coefficient can be determined referring to [27]. Thus, referring to [25], [26], [27], the hydrodynamic damping terms can be obtained using the derived drag coefficients. However, the design of the AUV is depends on its application or payload, therefore the AUV may not adhere to the design parameters as discussed in [27]. Other techniques such as computational fluid dynamics (CFD) analysis or planar mechanism motion (PMM) test [28–30] provides good approximation for the drag coefficient but at the cost of time and expensive experimental facility. Another method which is of much interest to the control community i.e. system identification (SI) technique. System identification method based on its implementation is categorized as off-line and on-line method. Referring to [31], an off-line technique is employed to identify the hydrodynamic damping terms using the prediction error method. But, with the change in payload of the AUV, the mass and the geometrical characteristics of the AUV will also change. Thus, the controller using the off-line identification technique becomes ineffective against these variations. On the other hand, on-line identification technique for an AUV dynamics (1.2) seems to be a satisfactory alternative. Therefore, this work will focus on development of adaptive control algorithms based on on-line-identification of the AUV dynamics.

The challenges of the parameter variation can be addressed by employing an adaptive control strategy. Referring to [32], adaptive control strategies in terms of control

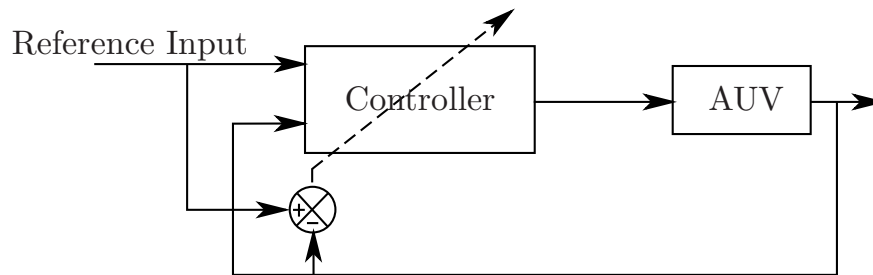


Figure 1.9: Combined Control and Learning architecture

and learning structure can be classified as (i) *combined control and learning*, (CCL), (ii) *separate control and learning*, (SCL) and (iii) *augmented control*, (AC). Referring the CCL architecture in [33, 34], the generation of the control law and identification of the model is carried out in subsequent time. It has simple control structure and no separate identification of the dynamics is required and also the generated control signal is based on the updated AUV dynamics. But apart from its advantages, this control scheme is computationally expensive because the learning and generation of control signal must be complete within a fixed sampling time. The constraint of restrictive learning can be alleviated by introducing a separate learning loop as in [35, 36] at the cost of complex architecture. Due to the separate learning loop, the parallel operation i.e. generation of control law and identification of the model within a fixed sampling time is achieved. Regardless of the complex control architecture, SCL is preferable over CCL because extensive and appropriate identification of the model is achieved due to parallel processing. The last control scheme AC [37–39] is the least expensive because rather than identifying complete dynamics as in CCL and SCL only the unknown term within the dynamics is to be identified. Although, its architecture is similar to SCL but the implementation of the AC is simpler and computationally less expensive. As discussed earlier, with change in geometrical characteristics not only damping but other terms such as mass, added mass, inertia, restoring terms are also affected. Thus, the situation where geometrical characteristics may change then the AC architecture will not be effective. Among three architecture, SCL is more preferable but computationally expensive.

Some of the surveying AUVs are equipped with robotic manipulator system or by adding extra payload such as camera or CTD sensor will affect the geometrical

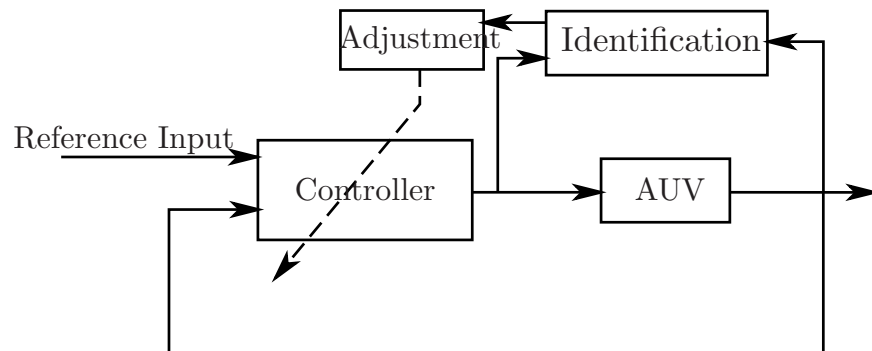


Figure 1.10: Separate Control and Learning architecture

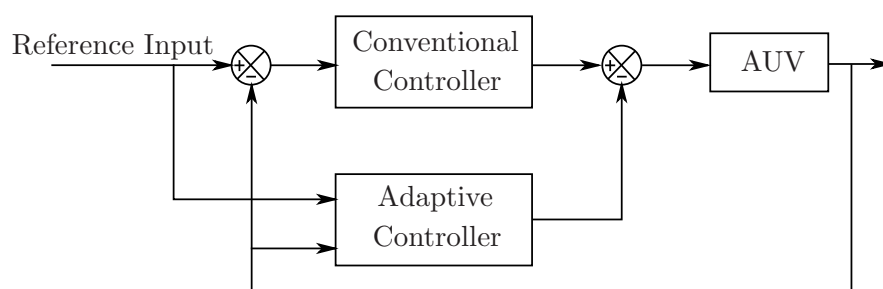


Figure 1.11: Augmented Control architecture

characteristics of the vehicle. These factors encourages the use SCL despite of its computationally expensiveness. SCL control structure in [35] implements a Self-Organizing Neural-Net Controller System (SONCS) structure. In SONCS, NN model is used for identifying the dynamics and another model is used as the feedforward controller. The SONCS network introduced in [35] has few problems i.e. the time complexity was more. To address this problem, a modified SONCS model is introduced in [36, 40], which consist of two parallel structure called Real-world part and imaginary-world part. Some of recent literature which implements NN or Neurofuzzy network for the realization of control algorithm are [41–44]. Another identification structure introduced in [45] i.e. polynomial-representation of NARMAX model, which is the general representation of any nonlinear system can also be exploited. As compared to NN model, polynomial-representation are simpler and these structures can correlated with actual dynamics. Due to its simpler design and real-time implementations in various fields [46–48], in this work polynomial-representation of the NARMAX model for SCL structure is chosen for the development of a motion control algorithm .

## 1.4 Motivations of the present work

From the available literature studied, it is observed that most of the research work reported on identification of the nonlinear dynamics of an AUV using soft-computing techniques such as neural-network or neuro-fuzzy techniques. However, a simpler model may exist which can identify the AUV dynamics with sufficient accuracy. Further, actuator limitations were not taken into account while designing the control laws for path-following task for an AUV. It is noted that very few work on control of AUV consider real-time implementation aspects pursued on a physical hardware. Thus, this thesis attempts to develop a prototype AUV and design adaptive controller.

## 1.5 Objectives of the thesis

- To develop a prototype AUV for practical implementation of the guidance algorithms for an AUV
- To derive minimal representation of the system identification algorithm for capturing the AUV dynamics.
- To develop an adaptive self-tuning PID control law scheme for an torpedo-shaped underactuated AUV for achieving guidance control design.
- To design a constrained adaptive control algorithm for implementing guidance algorithm considering the actuator constraints.
- In view of computational burden, a constrained explicit control algorithm is designed for implementing a guidance algorithm.

## 1.6 Organization of the thesis

The thesis is organized as follows,

- Chapter 2 describes the design and development of an AUV. Further, the hardware components required to achieve the autonomous capability is also discussed.

The software framework required to integrate various units and implementation of the control algorithm is then discussed.

- Chapter 3 presents development and implementation of an Inverse optimal adaptive PID controller for both diving motion and heading motion control of an AUV. This chapter first describes about the identification of AUV dynamics using a polynomial based NARMAX model of AUV followed by development of an gain adaptation algorithm for the PID controller. Further, the stability analysis of the controller is also studied.
- Chapter 4 derives a minimum representation of the polynomial based NARMAX model to identify the AUV dynamics. Further, a constrained self-tuning controller is developed for both heading and diving motion of the AUV. This control algorithm generates the actuation signal by solving a quadratic programming in the presence of actuator constraints.
- In Chapter 5 an explicit robust finite-time optimal controller is designed for an AUV considering the state and actuator constraints. The identified model obtained from Chapter 4 is used to design an explicit robust finite-time optimal controller.
- Chapter 6 provides the general conclusion of the thesis together with the contributions and scope of future work.
- Appendix A presents the dynamics and kinematics of the AUV. These equations are used for the development of control algorithm to implement LoS guidance law.
- Appendix B presents the solution of the multi-parametric Quadratic Programming (mp-QP) problem discussed in chapter 5, in order to generate the control signal.



# Chapter 2

## Development of a Prototype AUV

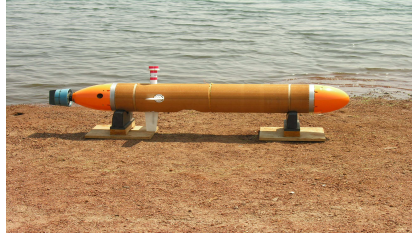
In view of experimentally verifying different control algorithms, an Autonomous Underwater Vehicle is developed in the laboratory. This chapter addresses the design and development aspect of the developed prototype AUV. The selection and design of the prototype AUV is discussed and further the hardware configuration such as sensors, actuators, computational unit used in the vehicle are also discussed. In addition to this, the software framework for the integration of various sensors and actuators with the computational unit is presented. This software framework provides a platform to implement the developed control algorithms presented in subsequent chapters.

The rest of the chapter is organized as follows. Section 2.1 describes about the design of the prototype AUV. The hardware components used in the prototype AUV is discussed in Section 2.2. Further, the software framework required to interface various sensors are presented in Section 2.3. Finally the chapter is concluded in Section 2.4.

### 2.1 Mechanical structure of AUV

Different applications of AUVs as discussed in chapter 1.1 necessitate to develop different type of AUVs, which can be broadly categorized in terms of design i.e. mono-hull structure and multi-hull structure.

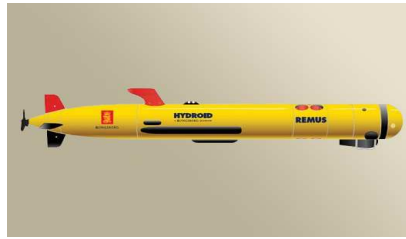
A mono-hull structure AUV is an underactuated system i.e. it has lesser number of actuators than the degree of freedom of the system. In order to control its six degree of freedom motion, mostly three actuators i.e. a rear-mounted thruster, rudder and



(a) MAYA AUV [49]



(b) MBARI AUV [50]



(c) REMUS AUV [51]

Figure 2.1: Torpedo shaped AUVs

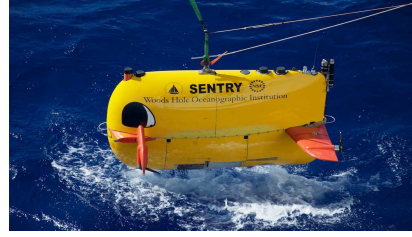
stern diving control planes are used. Due to its streamlined structure these vehicles experience less drag force, thus allows to achieving high speed and better endurance. Although the structures of these AUVs are simpler but as it is an underactuated system, the designing of control law is a challenging task. These AUVs are mostly preferred for long duration missions which require high endurance. Some of these AUVs are shown in Fig.2.1 which are deployed for various applications such as low-resolution surveys in large areas, surveillance and many other.

The multi-hull structure AUVs as shown in Fig.2.2 consists of multiple thrusters and its each degree of freedom is controllable. The design of these vehicles provides inherent stability against roll and pitch motions. Therefore, it has better maneuvering capability over mono-hull structure AUVs, however the endurance of these AUVs are less. Having better maneuvering and less endurance, these AUVs find application where close proximity to the environment is required. Some of its applications are photographic surveys or multibeam mapping tasks.

A mono-hull design is chosen for the prototype AUV due to its simple structure and challenges in the controller design. The design parameters of a mono-hull structure as shown in Fig.2.3 are necessary to describe the torpedo profile of an AUV. In [27], various torpedo profiles are studied and for the developed AUV Myring B profile with



(a) GIRONA AUV [52]



(b) SENTRY AUV [53]



(c) SEABED AUV [54]

Figure 2.2: Non-Torpedo shaped AUVs

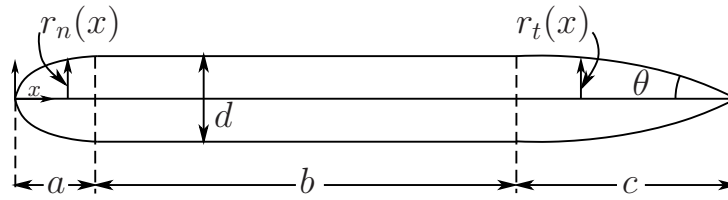


Figure 2.3: Design parameters of the Myring profile

design parameters is  $a/b/n/\theta/\frac{d}{2}=15/55/1.25/0.436/5$  is found to be suitable. However, the profile of the developed AUV structure is deviated from its ideal shape i.e. Myring B profile while manufacturing. In the subsequent section, these design profiles are discussed in detail.

- **Nose section:** The nose section is the frontal part of the AUV and it is used to place the additional payload according to different missions. According to [27], the expression for nose profile is given as

$$r_n(x) = \frac{d}{2} \left\{ 1 - \left( \frac{x-a}{a} \right) \right\}^{\frac{1}{n}}. \quad (2.1)$$

where  $r_n(x)$  is the radius of the nose profile along the x-axis,  $d$  is the diameter

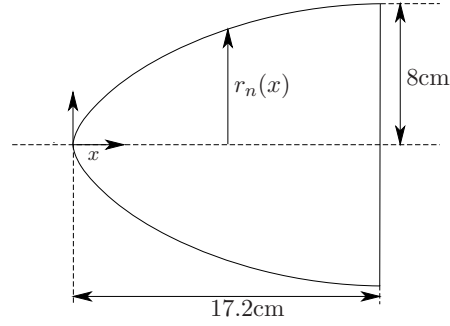


Figure 2.4: Nose section of the developed AUV

of the hull,  $a$  is the length of the nose section and  $n$  may be varied for different nose profile. These variables are also defined in Fig.2.3. The dimension of the developed nose profile is shown in Fig.2.4 and the data for its profile shape is presented in Table.2.1. The nose wall is of 5mm of thickness and it is made up of glass-fiber reinforced plastic (GFRP) material.

Table 2.1: Nose profile of the developed AUV

$x$ (in cm)	$r_n(x)$ (in cm)	$x$ (in cm)	$r_n(x)$ (in cm)
0	0	7.38	6.03
0.08	0.42	7.88	6.22
0.36	1.03	8.47	6.43
0.7	1.52	9.19	6.67
1.14	2.05	9.88	6.87
1.65	2.58	10.4	7.01
2.17	3.04	10.81	7.11
2.65	3.42	11.38	7.24
2.98	3.67	11.91	7.36
3.28	3.88	12.56	7.48
3.6	4.09	13.05	7.56
4.03	4.37	13.62	7.64
4.34	4.56	14.17	7.72
4.78	4.8	14.74	7.78
5.31	5.09	15.28	7.83
5.7	5.29	15.71	7.86
6.28	5.56	16.34	7.9
6.75	5.77	17.26	7.95

- **Tail section:** The tail section is the aft section of the AUV and it consists

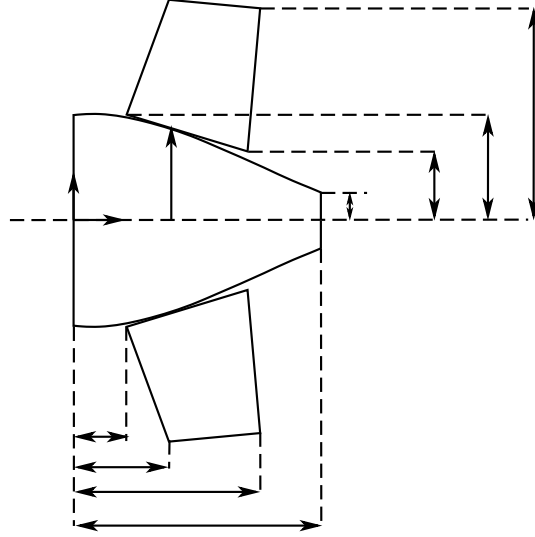


Figure 2.5: Tail section of the developed AUV

of actuators of the control planes and thruster as shown in Fig.2.5. According to [27], the expression for tail profile is given as

$$r_t(x) = \frac{d}{2} - \left\{ \frac{3d}{2(100 - a - b)^2} - \frac{\tan \alpha}{(100 - a - b)} \right\} \{x - a - b\}^2 + \left\{ \frac{d}{(100 - a - b)^3} - \frac{\tan \alpha}{(100 - a - b)^2} \right\} \{x - a - b\}^3 \quad (2.2)$$

where  $\alpha$  is the angle of the tail section and  $n$  is the variable which defines the curvature of the nose. The dimension of the designed tail section is presented in Fig.2.5 and its wall thickness is the same as the nose section. It is made up of GFRP material and its profile is presented in Table 2.2. The rudder and stern planes at the tail section enables the AUV for maneuvering in three dimensions. For the designing of these planes NACA-0020 air foil profile is used and it is made up of ABS plastic. The parameters of the control planes are presented in Table 2.3.

- **Hull section:** In the developed AUV, this section includes sensor unit, computational unit, power unit and battery bank. It is found that the hull section with length 68 cm and diameter 16 cm is sufficient to accommodate the components. The battery bank is placed at the lower half of the hull, so as to achieve passive

Table 2.2: Tail profile of the developed AUV

$x$ (in cm)	$r_t(x)$ (in cm)	$x$ (in cm)	$r_t(x)$ (in cm)	$x$ (in cm)	$r_t(x)$ (in cm)
85.86	7.95	94.56	6.44	99.78	4.2
86.4	8.01	95.02	6.26	99.98	4.09
86.94	8.04	95.4	6.12	100.23	3.99
87.48	8.04	95.96	5.89	100.64	3.8
87.89	8.02	96.48	5.67	100.97	3.65
88.31	8	96.89	5.49	101.32	3.49
88.91	7.94	97.18	5.37	101.61	3.36
89.72	7.82	97.44	5.25	101.94	3.22
90.23	7.73	97.63	5.17	102.29	3.05
90.55	7.66	97.9	5.04	102.45	2.99
91.11	7.53	98.14	4.94	102.76	2.86
91.57	7.41	98.37	4.84	102.99	2.76
92.17	7.24	98.63	4.72	103.22	2.66
92.73	7.07	98.86	4.62	103.49	2.54
93.34	6.88	99.11	4.51	104.08	2.3
93.78	6.72	99.38	4.38	104.77	2.02
94.19	6.58	99.58	4.29		

Table 2.3: Design parameters of the control planes

Parameter	Value	Unit	Definition
$S_f$	$8.63e - 3$	$m^2$	Planform Area
$t$	0.654	n/a	Taper Ratio
$b_f$	0.7246	m	Fin Span
$A_{re}$	1.766	n/a	Effective Aspect Ratio
$c_{L\alpha}$	2.8	n/a	Fin Lift Slope

stability against roll and pitch motion.

As mentioned earlier that the design of the prototype AUV is deviated from the Myring B profile while manufacturing the AUV structure. The scaled parameters which fit the Myring B profile are 16.84/61.7/1.25/0.436/5.61, whereas the parameters of the designed AUV are 17.26/60.6/ $n/\theta/8$ . Thus, due to the design difference, the theoretical drag coefficient provided in [27] cannot be used. Therefore, as discussed in the chapter 1, an identification technique can overcome the problem of identifying the AUV dynamics.

## 2.2 Hardware Configuration

Hardware architecture of the AUV includes sensor, computational, actuation, power and communication units as shown in Fig. 2.6. A detailed description of each unit is described as follows,

- **Sensor Unit:** It consists of Inertial Navigation System (INS), Doppler Velocity Log (DVL), Global Positioning System (GPS) and Pressure sensor as shown in Fig.2.7. INS is used to measure orientation  $(\phi, \theta, \psi)$  and angular velocities  $(p, q, r)$  while DVL provides translational velocities  $(u, v, w)$ . The positional information  $(x, y)$  with reference to  $\{I\}$  is obtained from differential GPS, whereas DVL and pressure sensor provides depth data  $(z)$ . Further, an extended Kalman filtering algorithm is employed to integrate various sensors in order to obtain states of AUV such as orientation and angular velocities. The outputs of the sensor unit are the states of the AUV i.e. linear and angular positions in  $\{I\}$  and linear and angular velocities in  $\{B\}$ . Fig.2.7 shows the interfacing between various sensors and the list sensors with its manufacturers which are used in the developed AUV.

- **INS:** Xsens MTi-30
- **DVL:** NavQuest 600 Micro
- **Temperature & Humidity:** Sensirion SHT-10
- **Pressure:** Measurement Specialities LM-31

- **Computational Unit:** An Intel Atom dual core processor of 1.6 GHz and 4 GB RAM and 30 GB hard disk with Linux operating system is used as the computational unit for the AUV. Further, packages such as Robot Operating System (ROS) [55] and MOSEK [56] are installed for the realization of the algorithms. ROS package is used to develop driver programs for different sensors as shown in Fig.2.7. It is also used to implement control or optimization algorithm written in C or python language.

- **Single Board Computer:** Advantech MIO-2261

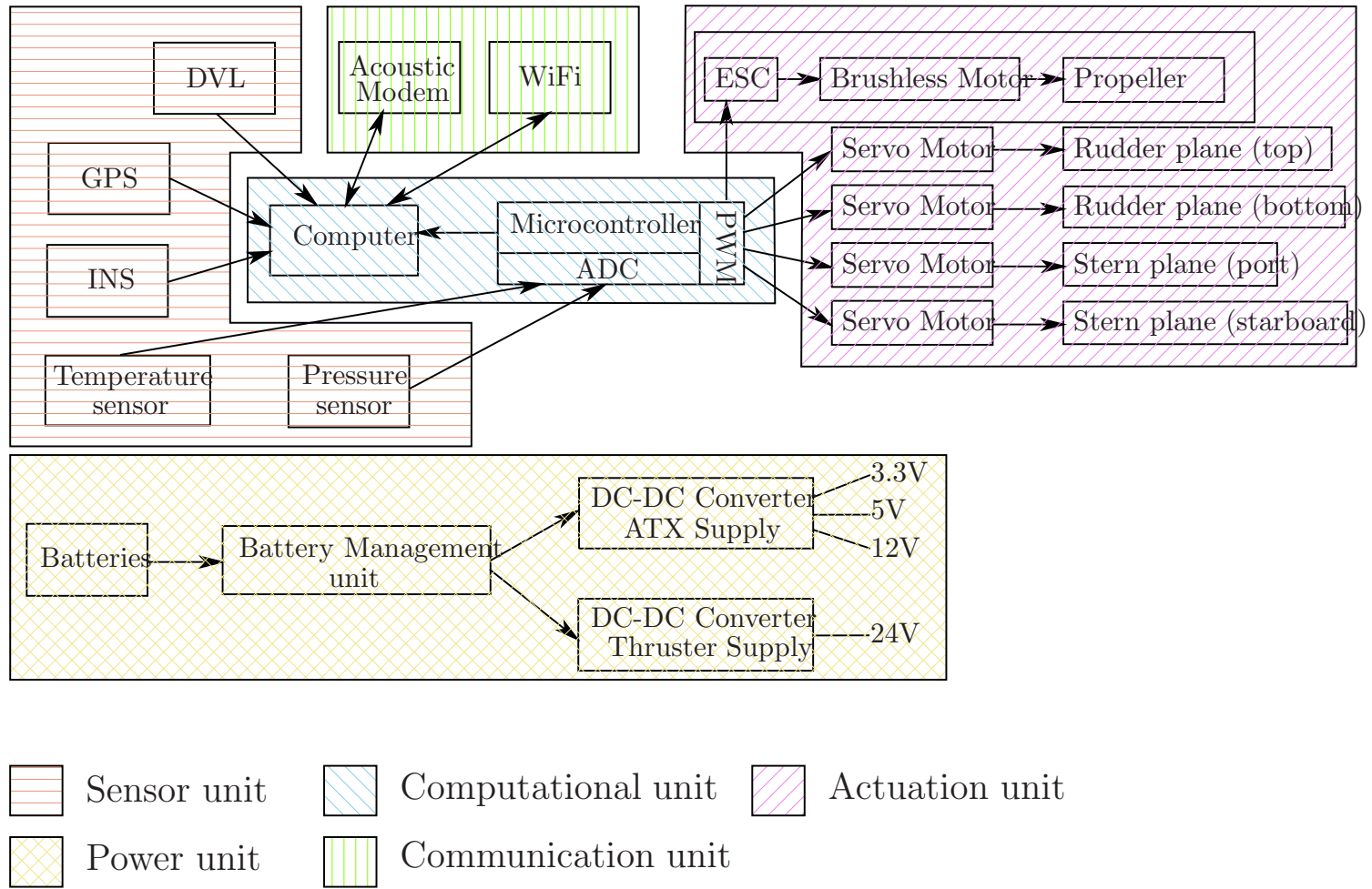


Figure 2.6: Hardware architecture of the AUV



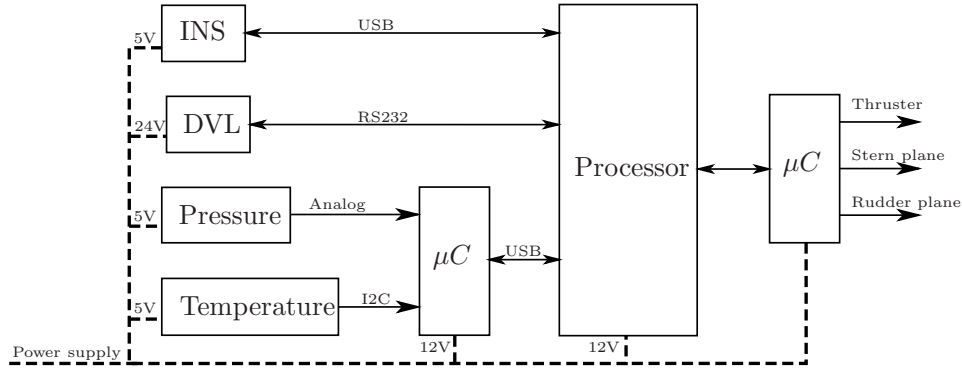


Figure 2.7: Sensor units

- **Actuation Unit:** It consists of single thruster for forward motion and four servo actuators to drive control planes. Pair of control planes are used to orient the AUV along yaw and pitch motion. These control planes are driven through high torque servo motors of maximum torque 11.3 kg/cm with 5 V power supply. In order to drive the servo motors, an Arduino microcontroller is used to generate equivalent PWM signal for the corresponding control signal. For surge motion, a 125 Watt thruster is used, which requires 12 V power supply for providing a forward thrust of 2.1 kg and 1.1 kg of backward thrust. It requires analog control signal between 0 V to 5 V for speed variation. Therefore, the 8 bit digital signal from the microcontroller is converted to an analog signal through a DAC IC.

- **Fins Motor:** Hitec HS-5646WP
- **Thruster:** Tecnadyne Model-150

- **Power Supply Unit:** It consists of battery bank, a battery management unit and two DC-DC converters. The amount of power required by the AUV is 180 watt approximately. The battery bank consisting of 6 Li-Ion batteries of 95W/hr each, thus it can power the AUV for 3 hours approximately. Battery management unit is used for charging, discharging and monitoring of these batteries and controlling the DC-DC converters. AUV is equipped with two DC-DC converters. One provides the ATX output (+12v, +5v, +3.3v) which is used to power the sensors, computational unit and communication modules. Another DC-DC

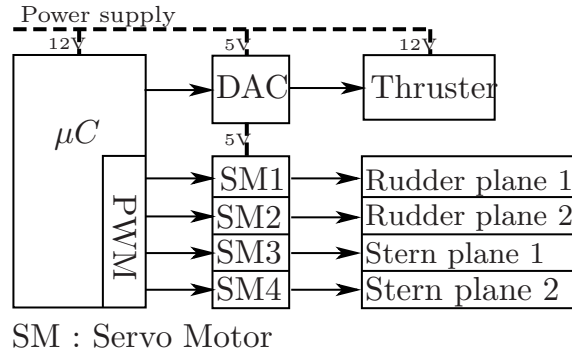


Figure 2.8: Actuation units

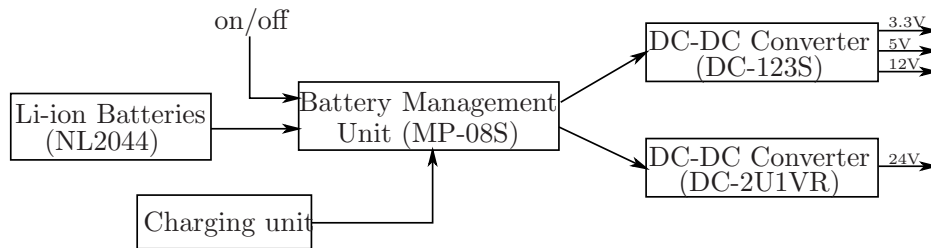


Figure 2.9: Power supply unit

converter which has high ampere rating is used to drive the thruster and other actuation unit.

- **Battery Management:** Oceanserver MP-08S
  - **DC-DC Converter:** Oceanserver DC-123S, DV-2U1VR
  - **Battery:** Inspiredenergy Li-Ion
- **Communication Unit:** It comprises of an acoustic modem, Wi-Fi and RF communication. An acoustic modem is used to communicate between AUV and the base station. As the data rate of the acoustic modem is very less approximately 30 bits per sec, it is only used for providing the way-points from the base station to the AUV. Wi-Fi is used to access the remote computer for retrieving the stored data or debugging the controller algorithm. A RF communication is also installed so as to generate the offline data by manually controlling the AUV
- **Acoustic Modem:** Desertstar SAM-1

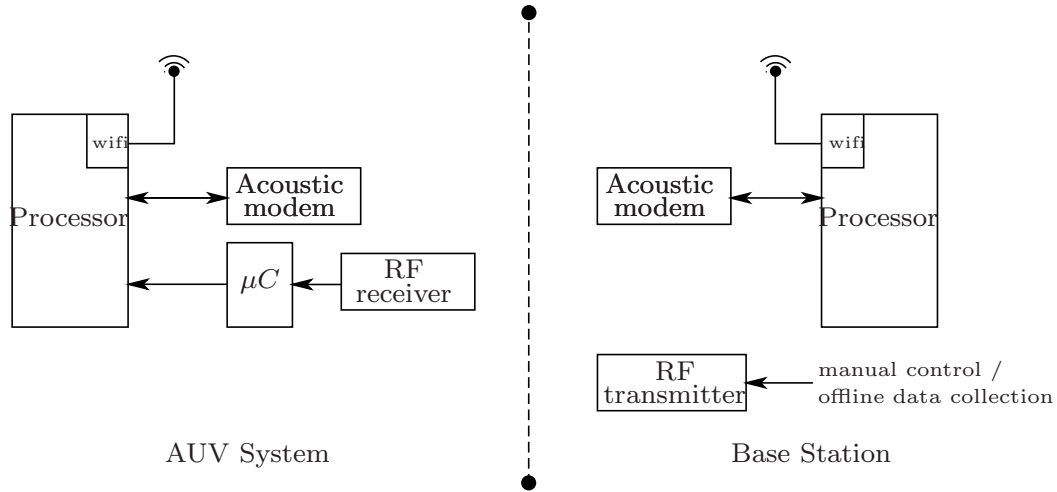


Figure 2.10: Communication unit

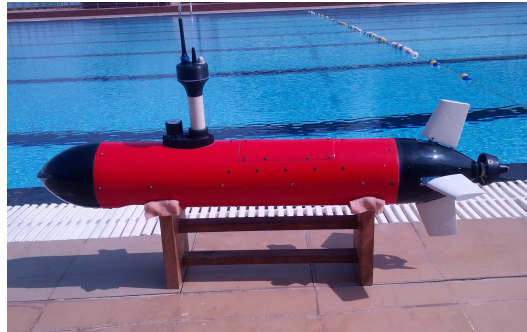


Figure 2.11: Prototype AUV developed at National Institute of Technology Rourkela

- **Wi-Fi:** Generic with Bulgin connector and antenna (IP68)
- **RF:** Fatuba joystick

Fig. 2.11 shows the developed prototype AUV which is developed at National Institute of Technology Rourkela and as discussed in this section the description of its structure is presented in Fig.2.12.

## 2.3 Software framework

Referring to Section 2.2, the computational unit of the AUV is an Intel dual core Atom processor with 30Gb hard disk and 2Gb RAM. Ubuntu 12.04 is installed along with

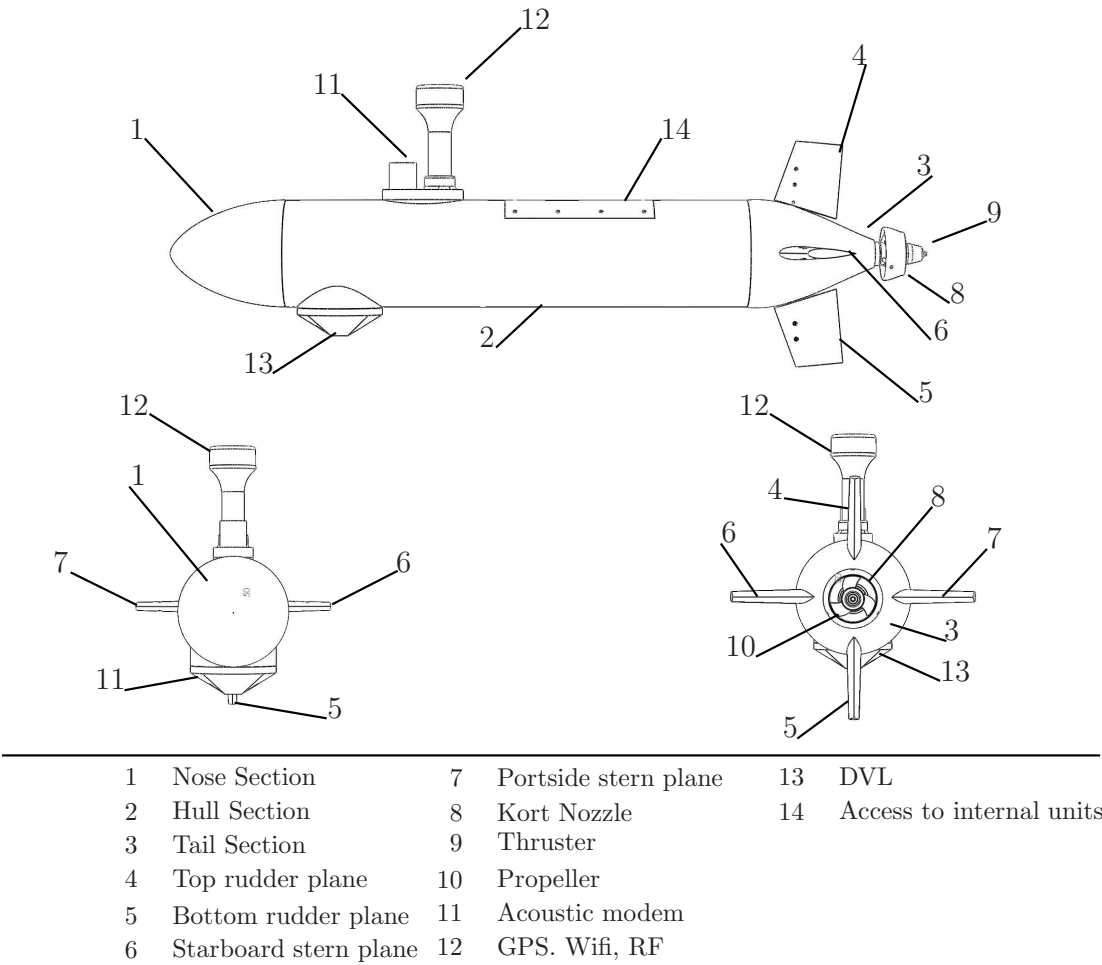


Figure 2.12: Different parts of the developed AUV

the following software packages for the execution of the AUV algorithms

- Robot Operating System (ROS) [55]: It is originated at Stanford Artificial Intelligence Lab and further developed and maintained by Willow Garage. ROS packages are used to integrate various sensor, actuator units and computational units. It is a set of libraries and tools which are used to write driver programs for sensors and control algorithms. Some of the terminologies used in ROS as follows
  - Nodes: Nodes are processes that perform computation. ROS is designed to be modular, it may consist of multiple nodes. For example, one node controls a laser range-finder, one node controls the wheel motors, one node performs localization, one node performs path planning, one Node provides a graphical view of the system, and so on.
  - Topics: A node sends out a message by publishing it to a given topic. The topic is a name that is used to identify the content of the message. It may be the information regarding velocity, speed, temperature etc.
  - Messages: Nodes communicate with each other by passing messages. A message is simply a data structure comprising typed fields.

In ROS intermediate nodes are created to acquire, process and transmit the data as shown in Fig.2.13. The sensor messages and actuator messages are the information required or generated from the computational unit. ROS is implemented in the developed AUV and each sensor and actuators are interfaced.

- MOSEK: MOSEK ApS provides free academic license to solve various convex optimization problem. For the developed AUV, it is used to solve constrained quadratic programming problem discussed in Chapter 4 and a constrained robust optimal control problem discussed in chapter 5 respectively.

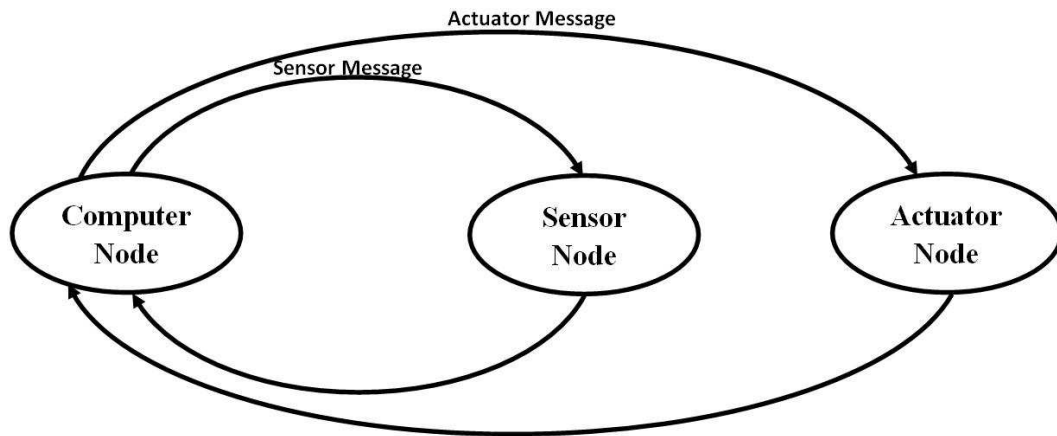


Figure 2.13: Example of ROS structure considering Computer, Sensor and Actuator as the Nodes

## 2.4 Chapter Summary

In this chapter, design and development of a prototype Autonomous Underwater Vehicle in the laboratory are presented. It discusses about the design specification of the nose and tail profile of the AUV. Further, the hardware configuration required to achieve autonomous capability is also discussed. The software framework which is required for interfacing of various sensors and also for the controller implementation is presented.

## Chapter 3

# LoS Guidance Law using Inverse Optimal Self-Tuning Adaptive Controller

In chapter 2, design and development of the prototype AUV is discussed. The developed AUV will be used for the experimental verification of the control algorithms. As stated in chapter 1, among various motion control schemes, path following and way-point tracking are suitable for underactuated AUVs. These algorithms can be implemented using LoS based guidance algorithms as presented in [19]. Further, an adaptive control strategy should be adopted to address the issue of payload variation or for resolving unknown AUV dynamics. Therefore, this chapter focusses on the development of a LoS based guidance control algorithm using Nonlinear Autoregressive Moving Average eXogenous (NARMAX) identified AUV dynamics for both heading as well as diving motion. Among various NARMAX structures as discussed in [1], polynomial-based NARMAX model structure is chosen because of its simplicity in control design. The parameters of the NARMAX model structure are updated on-line using Recursive Extended Least Square (RELS) algorithm in order to capture the unknown AUV dynamics. Using these parameters, an adaptive PID controller is designed for the implementation of the LoS guidance algorithm. The gain parameters of the PID controller are then tuned on-line at every  $k^{th}$  instant using an inverse optimal control technique [57], which alleviates the problem of solving a Hamilton-Jacobian

equation for generating a suitable control signal.

The chapter is organized as follows. Section 3.1 presents the problem statement addressed in this chapter. Further, the development of nonlinear identification technique for capturing AUV dynamics is described in Section 3.2. The obtained parameters are then used to develop an adaptive controller for both AUV kinematics and dynamics. The proposed control algorithm has been derived in two steps i.e. kinematic controller in Section 3.3.1 and dynamic controller in Section 3.3.2. The implementation of the control algorithm in a prototype AUV is discussed in Section 3.5, which envisages the effectiveness of the identification algorithm and LoS guidance algorithm. The chapter is concluded in Section 3.6.

## 3.1 Problem Statement

In order to develop a guidance algorithm for an torpedo shaped underactuated AUV, the roll motion is assumed to be zero and a constant surge velocity is considered throughout this work. Thus, considering these assumptions the kinematics and dynamics equation for an AUV is given as follows,

$$\dot{\eta} = J(\eta) \nu, \quad (3.1a)$$

$$M\dot{\nu} + C_r(\nu)\nu + f_d(\nu) + r_s(\eta) = \tau. \quad (3.1b)$$

In the kinematic expression (3.1a), the variable  $\eta = [x, y, z, \theta, \psi]^T \in \mathcal{R}^5$  denotes position vector in earth-fixed frame  $\{I\}$  and  $\nu = [v, w, q, r]^T \in \mathcal{R}^4$  is the velocity vector in the body-fixed frame  $\{B\}$ .  $J(\eta) \in \mathcal{R}^{5 \times 4}$  is the transformation matrix from  $\{B\}$  to  $\{I\}$ .

Separate AUV dynamics (3.1b) can be considered for heading motion  $[v, r]^T$  and diving motion  $[w, q]^T$  for the simplification of the control design. However the physical parameters of the AUV dynamics get affected when payload or/and physical structure is modified. Therefore, as discussed in chapter 1, a NARMAX identification technique is to be adopted for the identification of the AUV dynamics as shown in Fig.3.1. Further, a cascade control strategy is adopted for designing separate controllers for kinematics and dynamics. Using the path error, the controller for kinematics should



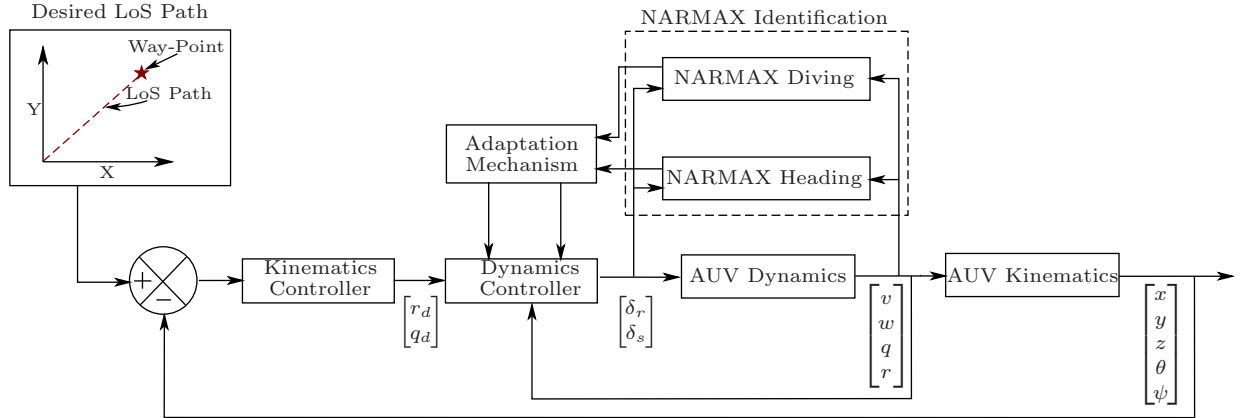


Figure 3.1: Structure of the proposed NARMAX model based self-tuning controller

generate desired velocities  $[r_d \ q_d]^T$  for the AUV dynamics. These velocities are to be followed by an AUV in order to track a desired path. Therefore, another controller for AUV dynamics should be designed which generates suitable actuation signal i.e.  $[\delta_r \ \delta_s]^T$  as shown in Fig.3.1. However, few assumptions are considered throughout this work i.e.

**Assumption 3.1.** *All the states of the kinematic equation (3.1a) and dynamic equation (3.1b) are measurable.*

**Remark 3.1.** *Considering the physical constraint or cost of the sensor system, the Assumption 3.1 is not always true. However, an observer can be designed as in [58], to estimate the unmeasured states of the AUV.*

**Assumption 3.2.** *The effect of rudder and stern plane on sway and heave motion is zero i.e.  $Y_{uu\delta_r} = Z_{uu\delta_s} = 0$ .*

**Remark 3.2.** *For an underactuated AUV, the inclusion of these terms complicates the controller design with no significant improvement in the tracking performance [17]. Unlike a fully-actuated vehicle, the effect of rudder and elevator fins along sway and heave motion is less significant, thus it can be neglected for the design of control law.*

**Assumption 3.3.** *Throughout this work, the surge velocity  $u_c$  of the AUV is assumed to be constant.*

**Remark 3.3.** *Considering an underactuated AUV, this assumption is generally adopted for path following problem because the desired path is independent of the time constraint. Further, an independent controller for surge motion can be designed to maintain a desired velocity.*

**Assumption 3.4.** *Roll angle and roll rate are assumed to be zero.*

**Remark 3.4.** *Although during some maneuvers, the roll oscillations may be significant. However, most of the AUVs maintain a vertical distance between center of gravity (CG) and center of buoyancy (CB) so as to decay the roll oscillation. Further, a decoupling method [59] or a separate roll-stabilization mechanism [60] can be employed to compensate the roll oscillations.*

**Assumption 3.5.** *It is assumed to have two separate identification schemes for heading and diving motions.*

**Limitation 1.** *In certain maneuvers the interaction of roll motion with the heading and diving dynamics is significant in nature, so during this instant Assumption 3.5 will not be effective.*

**Remark 3.5.** *Assumption 3.5 is desirable as it simplifies the controller design; the inherent characteristics of the controller to follow a desired path will eventually compensate the error accumulated due to Limitation 1.*

## 3.2 Identification of the AUV dynamics

System identification technique is suitable for capturing the AUV dynamics in real-time. Among the various system identification techniques, NARX model is a suitable paradigm for real-time implementation as mentioned in [61]. In spite of NARX model, a NARMAX model introduced in [1] can also be utilized for capturing the system dynamics more accurately. The general structure of the NARMAX model is given as

$$\hat{y}_p(k) = f(y_p(k-1), \dots, y_p(k-m), u_p(k-1), \dots, u_p(k-n)), \quad (3.2)$$

where  $f(\cdot)$  represents a nonlinear function consisting of delayed system output and control input,  $y_p(\cdot)$  is the system output and  $\hat{y}_p(\cdot)$  is the estimated output from the

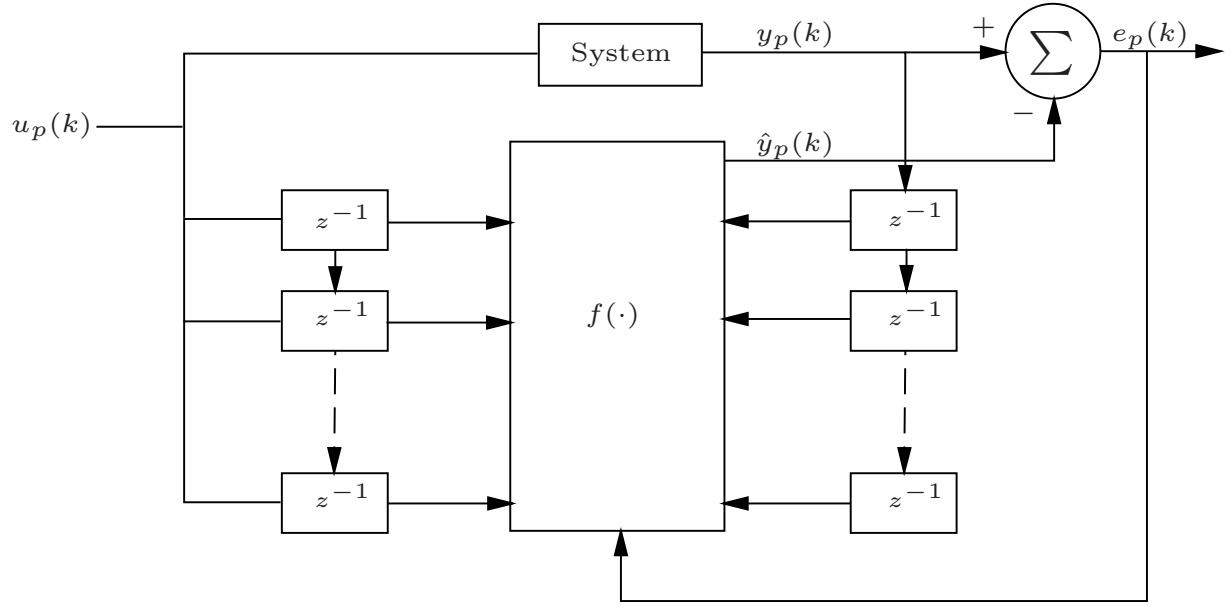


Figure 3.2: NARMAX model structure for system identification [1]

NARMAX structure. The general structure for the implementation of the NARMAX model in order to identify any dynamical system is shown in Fig.3.2. Referring to Fig.3.2, the output error of the NARMAX structure is used to tune the model parameters at every time instant.

In this work, polynomial based NARMAX model is used to identify the AUV dynamics which constituents of heading and diving motion. Heading motion includes sway and yaw motion  $[v_k \ r_k]^T$  whereas the diving motion includes heave and pitch motion i.e.  $[w_k \ q_k]^T$ . Two separate NARMAX structures are used to identify the heading and diving motion and for updatation of its parameter RELS algorithm is employed. Referring to [62], the NARMAX model for the heading motion is given by

$$\begin{bmatrix} v_k \\ r_k \end{bmatrix} = \begin{bmatrix} f_{11}(v_{k-1}, r_{k-1}) \\ f_{21}(v_{k-1}, r_{k-1}) \end{bmatrix} + \begin{bmatrix} g_{11} \\ g_{21} \end{bmatrix} \delta_{r,k-1} + \begin{bmatrix} d_{11} \\ d_{21} \end{bmatrix} e_{1,k-1}, \quad (3.3)$$

where

$$f_{11}(v_{k-1}, r_{k-1}) = \alpha_{01}v_{k-1} + \alpha_{11}r_{k-1} + \alpha_{21}v_{k-1}^2 + \alpha_{31}r_{k-1}^2 + \alpha_{41}v_{k-1}r_{k-1},$$

$$\begin{aligned}
g_{11} &= \alpha_{51}, \\
f_{21}(v_{k-1}, r_{k-1}) &= \beta_{01}v_{k-1} + \beta_{11}r_{k-1} + \beta_{21}v_{k-1}^2 + \beta_{31}r_{k-1}^2 + \beta_{41}v_{k-1}r_{k-1}, \\
g_{21} &= \beta_{51},
\end{aligned}$$

Similarly, the NARMAX model for diving motion is given by

$$\begin{bmatrix} w_k \\ q_k \end{bmatrix} = \begin{bmatrix} f_{12}(w_{k-1}, q_{k-1}) \\ f_{22}(w_{k-1}, q_{k-1}) \end{bmatrix} + \begin{bmatrix} g_{12} \\ g_{22} \end{bmatrix} \delta_{s,k-1} + \begin{bmatrix} d_{12} \\ d_{22} \end{bmatrix} e_{2,k-1}, \quad (3.4)$$

where

$$\begin{aligned}
f_{12}(w_{k-1}, q_{k-1}) &= \alpha_{02}w_{k-1} + \alpha_{12}q_{k-1} + \alpha_{22}w_{k-1}^2 + \alpha_{32}q_{k-1}^2 + \alpha_{42}w_{k-1}q_{k-1}, \\
g_{12} &= \alpha_{52} \\
f_{22}(w_{k-1}, q_{k-1}) &= \beta_{02}w_{k-1} + \beta_{12}q_{k-1} + \beta_{22}w_{k-1}^2 + \beta_{32}q_{k-1}^2 + \beta_{42}w_{k-1}q_{k-1}, \\
g_{22} &= \beta_{52},
\end{aligned}$$

In case of a torpedo shaped AUV with rear thruster for forward motion and control planes for orientation, there is no actuation along the sway and heave motion. Therefore,  $g_{11}$  and  $g_{12}$  in (3.3) and (3.4) can be termed as zero. These equations (3.3) and (3.4) can be represented as

$$Y_{i,k} = \phi_{i,k-1}^T \delta_{i,k-1}, \quad (3.5)$$

for  $i = \{1, 2\}$ . Referring to (3.5),  $\phi_{i,k-1}$  and  $\delta_{i,k-1}$  represents the regressor and parameter vector respectively. For  $i = 1$ ,  $Y_{1,k} = [v_k \ r_k]^T$  is considered for the identification of the heading dynamics. Similarly,  $Y_{2,k} = [w_k \ q_k]^T$  is used to identify the diving dynamics of the AUV. A Recursive Extended Least Square (RELS) algorithm [63] is employed due to unmeasurable noise terms  $[e_{1,k} \ e_{2,k}]^T$ . The expression for RELS algorithm for determining the estimated parameters are as follows,

$$\begin{aligned}
\hat{\delta}_{i,k} &= \hat{\delta}_{i,k-1} + \frac{S_{i,k-1}\phi_{i,k-1}}{\lambda_{k-1} + \phi_{i,k-1}^T S_{i,k-1} \phi_{i,k-1}} \varepsilon_{i,k-1}, \\
S_{i,k} &= \frac{1}{\lambda_{k-1}} \left\{ S_{i,k-1} - \frac{S_{i,k-1}\phi_{i,k-1}\phi_{i,k-1}^T S_{i,k-1}}{\lambda_{k-1} + \phi_{i,k-1}^T S_{i,k-1} \phi_{i,k-1}} \right\}, \\
\hat{Y}_{i,k} &= \phi_{i,k-1}^T \hat{\delta}_{i,k-1} + \varepsilon_{i,k-1},
\end{aligned}$$

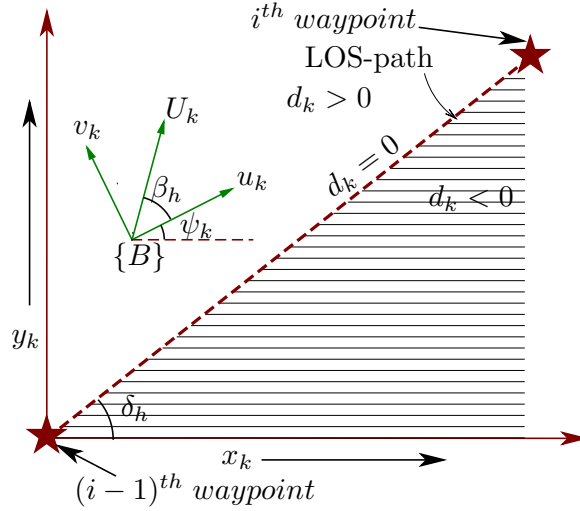


Figure 3.3: Desired LoS path

$$\lambda_k = \lambda_0 \lambda_{k-1} + (1 - \lambda_0), \quad (3.6)$$

where  $\lambda_k$ ,  $S_{i,k}$  and  $\varepsilon_{i,k-1}$  are the forgetting factor, covariance matrix and residual error output. In the subsequent section, a motion control scheme is developed using the identified model of the AUV.

### 3.3 Development of the Adaptive Inverse-Optimal PID Controller

The objective of path following control is achieved through designing separate controllers for kinematics and dynamics of the AUV. For kinematics, a Lyapunov based backstepping control is designed in section 3.3.1 to minimize the position and orientation error respectively. Further, a self-tuning PID controller is designed in section 3.3.2 for the AUV dynamics. This control law generates actuation signals for rudder and stern plane in order to steer the AUV along the LoS path. The detailed description of these controller development is presented in the subsequent sections.

### 3.3.1 Control design for kinematics

Referring to Fig. 3.3, let a LoS path with constant depth reference is to be followed by an AUV. From the figure, the expression for cross-track error can be deduced as

$$d_k = \langle A, X_k - X_0 \rangle - C \quad (3.7)$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & 0 & a_{23} \end{bmatrix}, \quad X_k - X_0 = \begin{bmatrix} x_k - x_0 \\ y_k - y_0 \\ z_k - z_0 \end{bmatrix}, \quad C = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}.$$

where  $A$  represents the path parameters and  $X_0, C$  represents the offset from the desired path. Further, the  $d_k = [y_{e,k} \ z_{e,k}]^T$  is defined as the cross-track error along the heading and diving motion respectively. In the subsequent section, the control input for AUV kinematics i.e. desired pitch and yaw velocities are derived for minimizing these cross-track errors to zero.

### Diving Control

Referring to [64], the kinematic equations for the diving motion is expressed as follows

$$z_k = z_{k-1} + T_s (-u_c \sin \theta_{k-1} + w_{k-1} \cos \theta_{k-1}), \quad (3.8a)$$

$$\theta_k = \theta_{k-1} + T_s q_{k-1}. \quad (3.8b)$$

Let the AUV is required to track a desired depth reference  $z_d$ , then the modified diving cross-track error for non-zero reference is given by

$$z_{e,k} = z_k - z_d. \quad (3.9)$$

$z_{e,k}$  should be minimized at every  $k^{th}$  instant. Therefore, a Lyapunov candidate function can be chosen as

$$V_{1,k} = \frac{1}{2} z_{e,k}^2 \quad (3.10)$$

and by showing that  $V_{1,k} - V_{1,k-1} \leq 0$ , the cross-track error  $z_{e,k}$  will always reduce to zero. Substituting  $z_k$  from (3.8a) into (3.10), the objective function  $V_{1,k}$  can be written as

$$V_{1,k} = V_{1,k-1} + T_s U_{cw} z_{e,k-1} \sin(\theta_{k-1} - \delta_d) + \frac{1}{2} T_s^2 \sin^2(\theta_{k-1} - \delta_d), \quad (3.11)$$

where  $\delta_d = \tan^{-1}(w_{k-1}/u_c)$  is the angle of attack and  $U_{cw} = \sqrt{(u_c^2 + w_{k-1}^2)}$  is the resultant velocity in the vertical axis. Assuming that sampling time  $T_s \ll 1$ , the equation (3.11) is expressed as

$$V_{1,k} - V_{1,k-1} = T_s U_{cw} z_{e,k} \sin(\theta_{k-1} - \delta_d). \quad (3.12)$$

In order to minimize the Lyapunov function  $V_{1,k}$  at each sampling instant, it is necessary that the following condition should be satisfied i.e.

$$V_{1,k} - V_{1,k-1} \leq 0. \quad (3.13)$$

Therefore, referring to (3.12), the condition (3.13) will be satisfied if

$$T_s U_{cw} z_{e,k} \sin(\theta_{k-1} - \delta_d) \leq 0. \quad (3.14)$$

Since  $T_s$  and  $U_{cw}$  are always positive, therefore a desired pitch angle can be chosen as

$$\theta_{k-1} = -\theta_a \frac{e^{2K_\delta z_{e,k-1}} - 1}{e^{2K_\delta z_{e,k-1}} + 1} + \delta_d, \quad (3.15)$$

where  $\theta_a$  is the maximum approaching angle and  $K_\delta$  is a positive term respectively. For  $k^{th}$  instant, equation (3.15) represents the desired pitch orientation ( $\theta_{d,k}$ ) and it is required to minimize the difference between the actual and desired pitch orientation. Therefore, an error in pitch orientation ( $e_{\theta,k}$ ) is defined as follows

$$e_{\theta,k} = \theta_k - \theta_{d,k}. \quad (3.16)$$

In order to minimize the pitch orientation error,  $e_{\theta,k}$  must satisfy the following condition i.e.

$$e_{\theta,k} = \xi_1 e_{\theta,k-1} \quad (3.17)$$

where  $\xi_1 \in (0, 1)$ . By substituting  $\theta_k$  from (3.8b) in (3.17), the desired pitch velocity for  $k^{th}$  instant is given by

$$q_{d,k} = \frac{1}{T_s} (\xi_1 \theta_{k-1} - \xi_1 \theta_{d,k-1} + \theta_{d,k} - \theta_{k-1}), \quad (3.18)$$

The desired pitch velocity  $q_{d,k}$  will be further used in Section 3.3.2 to derive a suitable actuation signal for AUV dynamics.

### Heading Control

Referring to [64], the heading motion is governed by the following kinematic equations i.e.

$$x_k = x_{k-1} + T_s \left( u'_{k-1} \cos \psi_{k-1} - v_{k-1} \sin \psi_{k-1} \right), \quad (3.19a)$$

$$y_k = y_{k-1} + T_s \left( u'_{k-1} \sin \psi_{k-1} + v_{k-1} \cos \psi_{k-1} \right), \quad (3.19b)$$

$$\psi_k = \psi_{k-1} + T_s \sec \theta_{k-1} r_{k-1}, \quad (3.19c)$$

where  $u'_{k-1} = (u_c \cos \theta_{k-1} + w_{k-1} \sin \theta_{k-1})$  is the projection of the net velocity along the surge motion of  $\{B\}$  and as specified earlier  $T_s$  is the sampling time. Using these kinematic equations, the desired yaw velocity should be derived such that the cross-track error along the heading motion should be minimized. Thus, referring to (3.7) the cross-track error is expressed as

$$y_{e,k} = a_{11}x_k + a_{12}y_k. \quad (3.20)$$

By substituting  $x_k$  and  $y_k$  from (3.19) into (3.20), the modified expression for cross-track error is given as

$$y_{e,k} = y_{e,k-1} + T_s U_k (a_{11} \cos(\psi_{k-1} + \beta) + a_{12} \sin(\psi_{k-1} + \beta)), \quad (3.21)$$

where  $\beta_k = \tan^{-1} \left( \frac{v_k}{u_c} \right)$  is the side-slip angle and  $U_k = \sqrt{u'^2_{k-1} + v^2_{k-1}}$  is the resultant velocity. Further, the expression (3.21) can be reduced to

$$y_{e,k} = y_{e,k-1} + T_s U_k a \sin(\psi_{k-1} + \beta_k - \delta_h), \quad (3.22)$$



where  $\delta_h = \tan^{-1} \frac{-a_{11}}{a_{12}}$  denotes the slope of the LoS path and  $a = \sqrt{a_{11}^2 + a_{12}^2}$  is the resultant value. In order to minimize the cross-track error  $y_{e,k}$ , a Lyapunov candidate function  $V_{2,k} \geq 0$  is chosen as

$$V_{2,k} = \frac{1}{2} y_{e,k}^2 \quad (3.23)$$

Assuming that  $T_s \ll 1$ , the candidate function  $V_{2,k}$  can be expressed as

$$V_{2,k} = V_{2,k-1} + T_s U_k a y_{e,k-1} \sin(\psi_{k-1} + \beta_k - \delta_h). \quad (3.24)$$

Thus, to minimize  $V_{2,k}$  at every  $k^{th}$  instant, the following condition should always be satisfied i.e.

$$U_k a y_{e,k-1} \sin(\psi_{d,k}) \leq 0, \quad (3.25)$$

where  $\psi_{d,k}$  is the desired yaw orientation. In (3.25),  $U_k$  and  $a$  are always positive therefore  $\psi_{d,k}$  can be chosen as

$$\psi_{d,k} = -\theta_a \frac{e^{2K_\delta y_{e,k-1}} - 1}{e^{2K_\delta y_{e,k-1}} + 1}. \quad (3.26)$$

Referring to Fig.3.3, considering the desired approaching angle  $\psi_{d,k}$  from (3.26), the orientation error between  $\{B\}$  and  $\{F\}$  is given by

$$\psi_{e,k} = \psi_k + \beta_k - \delta_h - \psi_{d,k}, \quad (3.27)$$

where  $\delta_h$  is the slope of the desired path and  $(\psi_k + \beta_k)$  is the orientation of the resultant velocity  $U_k$  w.r.t.  $\{I\}$ . Using (3.27) and substituting  $\psi_k$  from (3.19c), the desired yaw rate at  $k^{th}$  instant is

$$r_{d,k} = \frac{\cos \theta_{k-1}}{T_s} (\xi_2 \psi_{e,k} + \psi_{d,k} + \delta_h - \beta_k - \psi_{k-1}), \quad (3.28)$$

where  $\xi_2 \in (0, 1)$ . The desired velocities (3.18) and (3.28) are obtained using the AUV kinematic equations. In section 3.3.2, a suitable actuation signal is generated so that the actual velocities track the desired velocities. Thus, it enables the AUV to track the desired path.

### 3.3.2 Control design for dynamics

In this section, the suitable actuation signals for AUV dynamics have to be obtained such that the actual velocities  $q_k$  and  $r_k$  should track the desired velocities  $q_{d,k}$  and  $r_{d,k}$ . As discussed in section 3.3.1, the NARMAX model structure for heading motion and diving motion can be represented as

$$\nu_k = f(\nu_{k-1}) + g(\nu_{k-1})\mu_{k-1}, \quad (3.29)$$

where  $\nu_k \in \mathbb{R}^2$  and  $\mu_k \in \mathbb{R}$  are the states and input of the system.

**Theorem 3.1.** [65] *For an affine nonlinear function (3.29), let an output is defined as*

$$y_k = h(\nu_{k-1}, \nu_{\delta,k}) + J(\nu_{k-1})\mu_{k-1}. \quad (3.30)$$

*Considering a Lyapunov function  $V_{k-1} = \frac{1}{2}\nu_{e,k-1}^T \bar{P} \nu_{e,k-1}$ , where  $\nu_{e,k-1} = \nu_k - \nu_{\delta,k}$  is the difference between the actual and desired state. The passivity condition*

$$V_k - V_{k-1} \leq y_{k-1}^T \mu_{k-1}, \quad (3.31)$$

*is always satisfied if,*

$$\mu_{k-1} = -(\mathbf{I}_m + J(\nu_{k-1}))^{-1} h(\nu_{k-1}, \nu_{\delta,k}), \quad (3.32)$$

*where*

$$\begin{aligned} h(\nu_{k-1}, \nu_{\delta,k}) &= g^T(\nu_{k-1}) \bar{P} (f(\nu_{k-1}) - \nu_{\delta,k}), \\ J(\nu_{k-1}) &= \frac{1}{2} g^T(\nu_{k-1}) \bar{P} g(\nu_{k-1}). \end{aligned} \quad (3.33)$$

The control input expressed in (3.32) can also be expressed in terms of  $e_k$  as follows

$$\mu_{k-1} = -(I_m - J(\nu_{k-1}))^{-1} g^T(\nu_{k-1}) \bar{P} e_k. \quad (3.34)$$

By substituting (3.29) into (3.34), the control law becomes complex and it is not feasible for implementation as the control input  $\mu_{k-1}$  depends on future instant. Therefore,

a local linear model is identified in order to simplify the control structure. In view of feasibility of the controller, an adaptive self-tuned PID controller is adopted. Using [62], a linearized expression for (3.29) is presented as

$$\tilde{\nu}_k = A_{k-1}\tilde{\nu}_{k-1} + B_{k-1}\mu_{k-1} + L_{k-1}e_{k-1,lin}, \quad (3.35)$$

where  $e_{k,lin} = \tilde{\nu}_k - \nu_{\delta,k}$  and  $L_{k-1}$  is the observer gain. Considering the observer gain  $L_{k-1}$  as

$$L_{k-1} = [(P_{k-1} + R)^{-1}P_{k-1}A_{k-1}^T]^T, \quad (3.36)$$

ensures that the eigen values of the linearized model (3.35) always lie within the unit circle. In (3.36),  $R$  defines the covariance of the noise.  $P_{k-1}$  is the solution of the Riccati equation given in (3.37).

$$A_{k-1}P_{k-1}A_{k-1}^T - P_{k-1} - (A_{k-1}P_{k-1})(P_{k-1} + R_{k-1})^{-1}(P_{k-1}A_{k-1}^T) + Q_{k-1} = 0. \quad (3.37)$$

By replacing  $e_k$  with  $e_{k,lin}$  in (3.34), the control law can be written as

$$\mu_{k-1} = -(I_m - J(\tilde{\nu}_{k-1}))^{-1}g^T(\tilde{\nu}_{k-1})\bar{P}e_{k,lin}, \quad (3.38)$$

where  $e_{k,lin} = \tilde{\nu}_k - \nu_{\delta,k}$  and  $\bar{P} > 0$  is a weight variable. Substituting (3.35) into (3.38) yields

$$\mu_{k-1} = (I_m - J(\tilde{\nu}_{k-1}) + g^T(\tilde{\nu}_{k-1})B)^{-1}[-g^T(\tilde{\nu}_{k-1})\bar{P}\{(A + L_{k-1})e_{k-1,lin} + A\nu_{\delta,k-1} - \nu_{\delta,k}\}]. \quad (3.39)$$

The derived control law satisfies the requirements for the stability of the closed loop system. The control law is adaptive to the payload variation or modification to the physical structure due to the NARMAX identification of the AUV dynamics. Further, (3.39) is realized as an adaptive PID controller where the gains of the PID controller are updated at each instant in order to steer the AUV along the desired path. Let  $\bar{P} = 1 + P_{12}z^{-1} + P_{13}z^{-2}$ , then (3.39) can also be represented as

$$\mu_{k-1} = X_1 + X_2z^{-1} + X_3z^{-2}, \quad (3.40)$$

where

$$\begin{aligned} X_1 &= -(I_m - J(\tilde{\nu}_{k-1}) + g^T(\tilde{\nu}_{k-1})B)^{-1} g^T(\tilde{\nu}_{k-1})(A + L_{k-1})e_{k-1}, \\ X_2 &= -(I_m - J(\tilde{\nu}_{k-1}) + g^T(\tilde{\nu}_{k-1})B)^{-1} g^T(\tilde{\nu}_{k-1})\bar{P}_{12}(A + L_{k-1})e_{k-2}, \\ X_3 &= -(I_m - J(\tilde{\nu}_{k-1}) + g^T(\tilde{\nu}_{k-1})B)^{-1} g^T(\tilde{\nu}_{k-1})\bar{P}_{13}(A + L_{k-1})e_{k-3}. \end{aligned}$$

Referring to [66], the discrete-time PID control law is expressed as

$$\mu_{k-1} = \left( k_{ci} \left( 1 + \frac{T_s}{T_{Ii}} + \frac{T_{Di}}{T_s} \right) - k_{ci} \left( 1 + \frac{2T_{Di}}{T_s} \right) z^{-1} + k_{ci} \frac{T_{Di}}{T_s} z^{-2} \right) e_{k-1}, \quad (3.41)$$

and by comparing (3.41) with (3.40), the gains of the PID controller can be obtained as follows

$$k_{ci} = -(X_2 + 2X_3), \quad (3.42)$$

$$T_{Di} = -\frac{X_3}{(X_2 + 2X_3)} T_s, \quad (3.43)$$

$$T_{Ii} = -\frac{X_2 + 2X_3}{(X_1 + X_2 + X_3)} T_s. \quad (3.44)$$

However, the control law (3.39) and the adaptive PID controller (3.41) are equivalent but (3.41) is more suitable in view of implementation.

In view of controller design for heading motion the variables of (3.29) becomes  $\nu_k = [v_k \ r_k]^T$  and  $\mu_k = \delta_{r,k}$ , whereas for diving motion  $\nu_k = [w_k \ q_k]^T$  and  $\mu_k = \delta_{s,k}$  respectively.

## 3.4 Stability Analysis

Let a Discrete-Time Control-Lyapunov Function (DTCLF) for tracking the desired state  $x_{\delta,k}$ ,

$$V(\nu_k, \nu_{\delta,k}) = \frac{1}{2} (\nu_k - \nu_{\delta,k})^T P (\nu_k - \nu_{\delta,k}) \quad (3.45)$$

If the derived control input satisfies the passivity condition i.e.

$$V(\nu_k, \nu_{\delta,k}) - V(\nu_{k-1}, \nu_{\delta,k-1}) \leq \sigma_{k-1}^T \mu_{k-1} \quad (3.46)$$

then the system (3.29) is said to be globally asymptotic stable. Referring to [65],  $\sigma_{k-1}$  can be written as

$$\sigma_{k-1} = h(\nu_{k-1}, \nu_{\delta,k}) + J(\nu_{k-1})\mu_{k-1}. \quad (3.47)$$

Thus, rewriting the passivity condition (3.46) by replacing  $\sigma_{k-1}$  from (3.47) is given by

$$V(\nu_k, \nu_{\delta,k}) - V(\nu_{k-1}, \nu_{\delta,k-1}) \leq h(\nu_{k-1}, \nu_{\delta,k})^T \mu_{k-1} + \mu_{k-1}^T J(\nu_{k-1}) \mu_{k-1} \quad (3.48)$$

where  $h(\nu_{k-1}, \nu_{\delta,k})$  is given by

$$h(\nu_{k-1}, \nu_{\delta,k}) = g^T(\nu_{k-1})\bar{P}(e_{k+1} - g(\nu_{k-1})\mu_{k-1}) \quad (3.49)$$

the passivity condition of (3.46) becomes

$$\begin{aligned} V(\nu_k, \nu_{\delta,k}) - V(\nu_{k-1}, \nu_{\delta,k-1}) &\leq e_k^T \bar{P}g(\nu_{k-1})\mu_{k-1} - \mu_{k-1}^T \left( g(\nu_{k-1})^T \bar{P}g(\nu_{k-1}) - J(\nu_{k-1}) \right) \mu_{k-1} \\ &\leq e_{k+1,lin}^T \bar{P}g(\nu_{k-1})\mu_{k-1} - \mu_{k-1}^T g(\nu_{k-1})^T \bar{P}g(\nu_{k-1})\mu_{k-1} + \\ &\quad \mu_{k-1}^T J(\nu_{k-1}) \mu_{k-1} + \tilde{e}_{k+1}^T \bar{P}g(\nu_{k-1})\mu_{k-1} \end{aligned} \quad (3.50)$$

$\tilde{e}_{k+1}$  is the difference between actual error  $e_{k+1}$  and approximated error  $e_{k+1,lin}$  derived from the linearized model. Using (3.38), equation (3.50) is expressed as,

$$\begin{aligned} V(\nu_k, \nu_{\delta,k}) - V(\nu_{k-1}, \nu_{\delta,k-1}) &\leq \mu_{k-1}^T (-I_m + J(\nu_{k-1})) \mu_{k-1} - \mu_{k-1}^T g(\nu_{k-1})^T \bar{P}g(\nu_{k-1})\mu_{k-1} + \\ &\quad \mu_{k-1}^T J(\nu_{k-1}) \mu_{k-1} + \tilde{e}_{k+1}^T \bar{P}g(\nu_{k-1})\mu_{k-1} \\ &\leq -\mu_{k-1}^T \left( -I_m + J(\nu_{k-1}) - g(\nu_{k-1})^T \bar{P}g(\nu_{k-1}) + \right. \\ &\quad \left. J(\nu_{k-1}) \right) \mu_{k-1} + \tilde{e}_{k+1}^T \bar{P}g(\nu_{k-1})\mu_{k-1} \end{aligned} \quad (3.51)$$

$\|\tilde{e}_{k+1}^T \bar{P}g(\nu_{k-1})\mu_{k-1}\| < \rho$ , where  $\rho$  is a positive constant. The term  $\tilde{e}_{k+1}^T \bar{P}g(\nu_{k-1})\mu_{k-1}$  is always bounded because an observer gain is designed in (3.36) which reduces the  $\tilde{e}_{k+1}$  error. Thus, equation (3.51) can be expressed as

$$V(\nu_k, \nu_{\delta,k}) - V(\nu_{k-1}, \nu_{\delta,k-1}) \leq -\mu_{k-1}^T \mu_{k-1} + \rho, \quad (3.52)$$

which satisfies the passivity condition.

## 3.5 Results and Discussion

In this section, the effectiveness of the developed control algorithm is verified both in simulation as well as in experimental environment. Prior to implementing the control algorithm on the developed AUV whose parameters are unknown, simulation studies have been carried out using INFANTE AUV whose parameters are given in [67]. In the subsequent section the results obtained from simulation and experiment are discussed as follows.

### 3.5.1 Simulation Results

The effectiveness of the proposed controller is studied through simulation in the MATLAB environment. The performance of the controller developed in section 3.3 is verified using the parameters of INFANTE AUV given in [67]. The initial states of the AUV dynamics and kinematics are defined as  $[x \ y \ z \ \theta \ \psi \ v \ w \ q \ r] = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ . Other parameters which are necessary for the implementation of the control algorithm are listed in Table 3.1. In order to verify the control algorithm, the desired heading is provided as follows,

$$\psi_d = \begin{cases} 0, & 0 \leq t < 50 \\ 1.57, & 50 \leq t < 200 \\ 0, & 200 \leq t < 350 \\ -1.57, & 350 \leq t < 500 \\ 0, & 500 \leq t < 650 \\ 1.57, & 650 \leq t < 800 \end{cases} \quad (3.53)$$

and it is required that the actual heading of the AUV should track the desired heading.

The AUV is required to follow the desired heading as described in (4.32). Referring to Fig.3.4, the AUV tracks the desired heading and its corresponding heading error is presented in Fig.3.5. In order to design the controller, the unknown AUV dynamics is

Table 3.1: Simulation parameters

$T_s = 0.5$	$\theta_a = 0.52$	$K_\delta = 0.4$	$u_c = 0.8$
$\xi_1 = 0.8$	$\xi_2 = 0.8$		

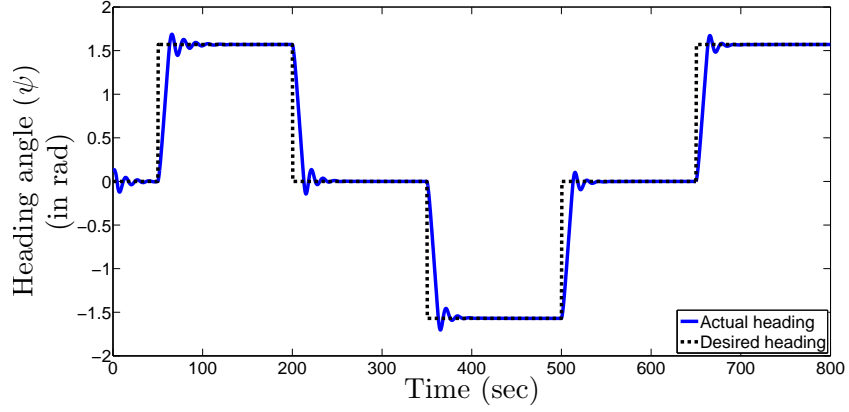


Figure 3.4: Tracking of desired heading by INFANTE AUV

identified using polynomial based NARMAX structure at every time instant. Fig.3.6 compares the identified yaw motion with the actual yaw motion and from the figure it is evident the polynomial-based NARMAX structure is suitable. Further, the updations of its parameters are given in Fig.3.7. Using these parameters the control law (3.41) is implemented and the generated actuation signal i.e.  $\delta_r$  to steer the AUV along the yaw motion is shown in Fig.3.8. In this work, it is assumed that the control planes varying from  $-0.785$  rad to  $0.785$  rad, therefore the actuation signal exceeding

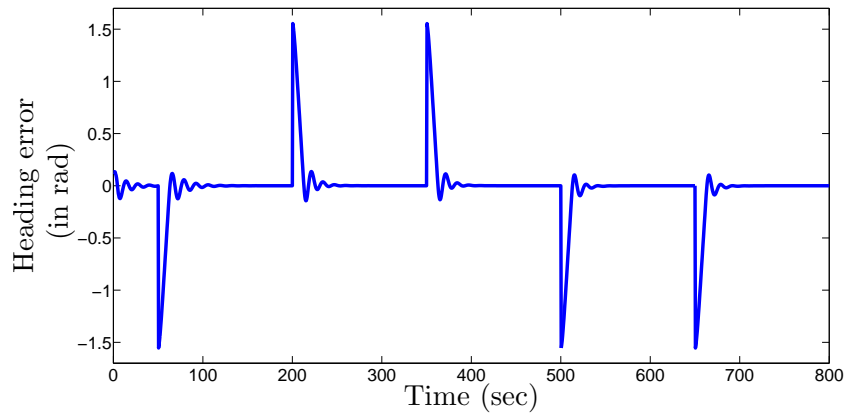


Figure 3.5: Heading error

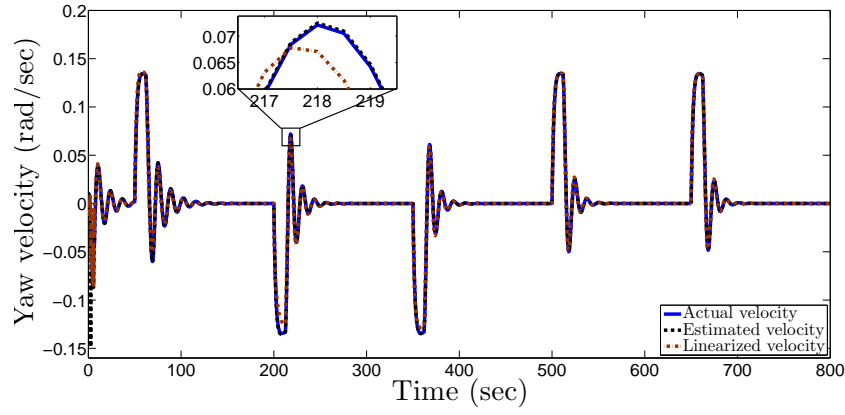


Figure 3.6: Estimated yaw velocity as compared to actual yaw velocity

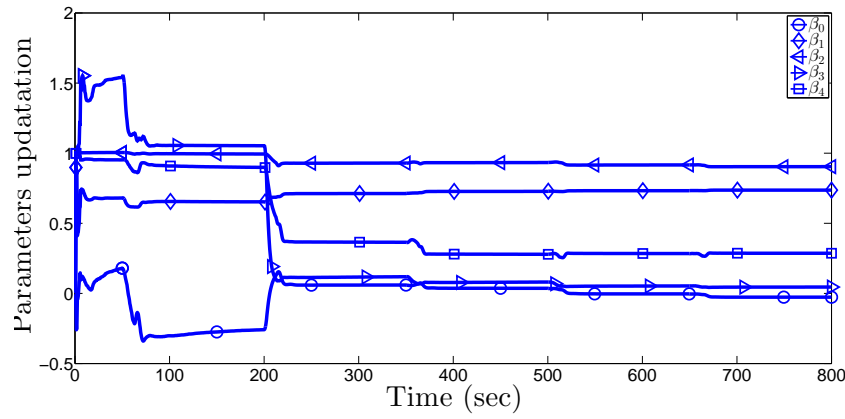


Figure 3.7: Updation of the NARMAX parameters for yaw motion

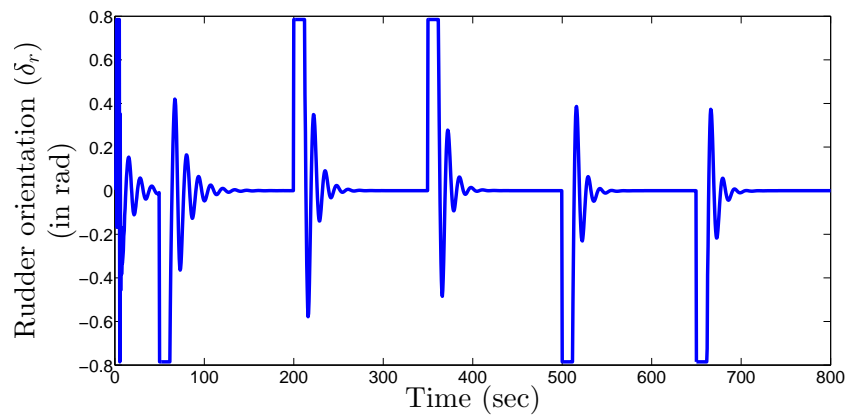


Figure 3.8: Actuation signal while tracking a desired heading



Table 3.2: Description of ROS nodes

Node	Description
/xsens_driver	Driver node to access Xsens MTI INS sensor
/navquest_node	Driver node to access Navquest DVL sensor
/odom_trans	Generates AUV states
/controller_self_tuning_PID	Controller node (3.41)
/narimax	NARMAX node for identification (3.2)
/narimax_linearized	Linearized AUV velocities (3.35)
/roserial_server	Arduino node for transmitting actuation signal

these extremum values is clipped using a limiter. From the obtained results, it is evident that the NARMAX model structure along with the self-tuning PID control algorithm is effective for controlling AUV motion.

### 3.5.2 Experimental Results

From the simulation results, it is confirmed that the system identification technique and the developed control algorithm is suitable for controlling the AUV. Further, in this section the control algorithm is verified in the prototype AUV. As discussed in chapter 1, the prototype AUV is installed with Ubuntu 12.04 and ROS Hydro [55]. Therefore, ROS nodes of the identification and control algorithm as shown in Fig.3.9 is designed which accepts the data from the AUV sensor nodes and generate the actuation signal to the AUV actuation nodes. The description of these nodes is presented in Table 3.2 and the characteristics of their messages such as bandwidth, frequency etc are shown in Table 3.3.

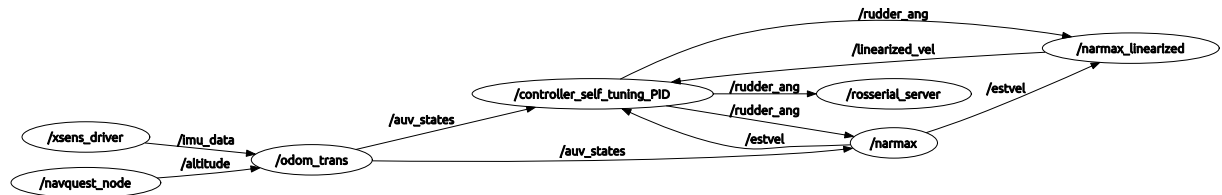


Figure 3.9: Communication between ROS nodes

The experimentation is conducted at National Institute of Technology Rourkela swimming pool having length 10m, breadth 5m and maximum depth 1.6m. Let, the

Table 3.3: Description of ROS messages and its characteristics

Message	Description	Bandwidth	Publishing rate
/imu_data	$\phi, \theta, \psi, p, q, r$	32.8 Kb/sec	100
/auv_states	$z, \phi, \theta, \psi, u, v, w, p, q, r$	523 B/sec	10
/linearized_vel	$\tilde{v}, \tilde{w}, \tilde{q}, \tilde{r}$	525 B/sec	10
/estvel	$\hat{v}, \hat{w}, \hat{q}, \hat{r}$	522 B/sec	10
/rudder_ang	$\delta_r$	20 B/sec	10
/altitude	$z$	20 B/sec	10

Table 3.4: Parameters

$T_s = 0.1$	$\theta_a = 0.52$	$K_\delta = 0.1$	$u_c = 1$
$\xi_1 = 0.1$	$\xi_2 = 0.8$		

desired heading reference similar to (4.32) is considered as follows

$$\psi_d = \begin{cases} 0, & 0 \leq t < 5 \\ 1.57, & 5 \leq t < 20 \\ 0, & 20 \leq t < 35 \\ -1.57, & 35 \leq t < 50 \\ 0, & 50 \leq t < 65 \\ 1.57, & 65 \leq t < 75 \end{cases} \quad (3.54)$$

and the initial states of the AUV is  $[x, y, z, \theta, \psi, v, w, q, r]^T = [0, 0, 0, 0, 0, 0, 0, 0, 0]^T$ . The implementation of the control algorithm in the prototype AUV is shown in Fig.3.10. Throughout the experimentation the surge velocity of the AUV is assumed to be constant i.e.  $0.8 \text{ m/sec}$ . Further, the design parameters considered during the experimental verification of the control algorithm is presented in Table 3.4.

Referring to Fig.3.11, the prototype AUV tracks the desired heading orientation (4.32) and the corresponding heading error is presented in Fig.3.12. Referring to Fig.3.14, the designed polynomial-based NARMAX structure is used to identify the AUV dynamics at each sampling time and the obtained parameters as shown in

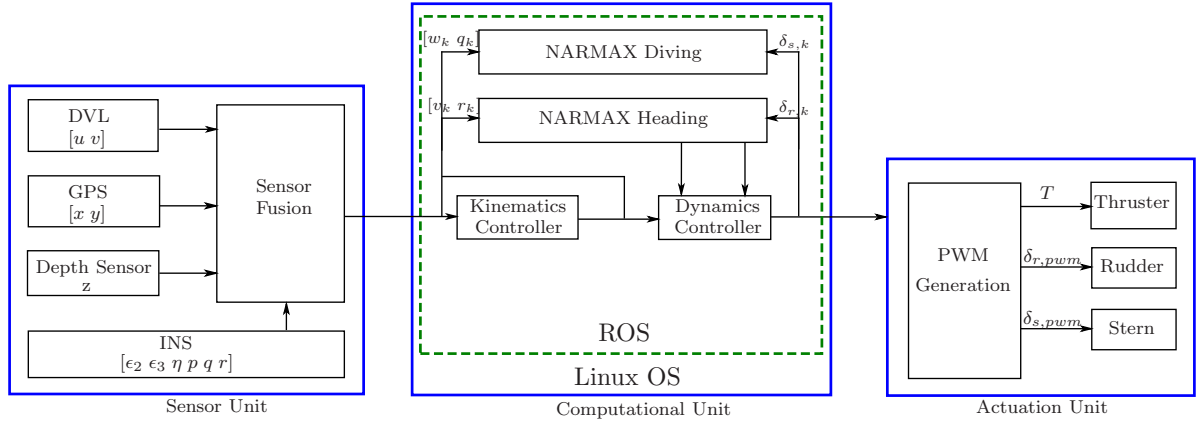


Figure 3.10: Implementation of the self-tuning controller

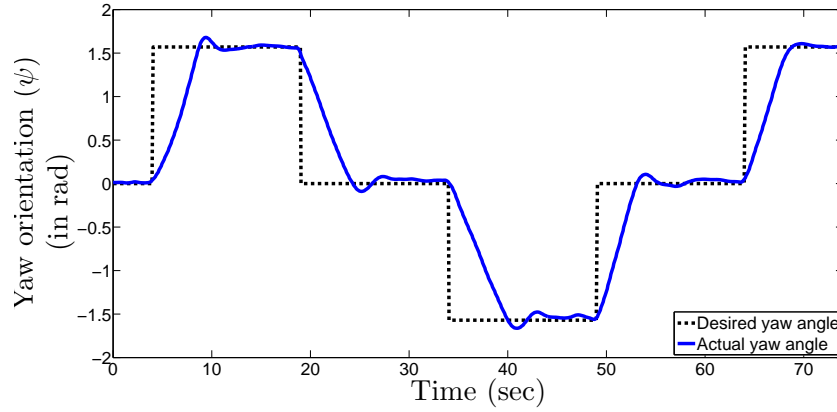


Figure 3.11: Tracking of desired heading

Fig.3.15 are then used to develop a control algorithm. Using these parameters the control law defined in (3.41) generates the actuation signal as shown in Fig.3.17 and the controller parameters are shown in Fig.3.16. Similar to the simulation studies, the control planes varies between  $-0.785$  rad to  $0.785$  rad, therefore the actuation signal exceeding these extremum values is clipped using a limiter. The computational time taken by the algorithm is shown in Fig. 3.18 and it is found that the maximum time taken by the algorithm to generate an actuation signal is 0.011 sec. From the obtained results, it is clear that the control algorithm is suitable for practical realization but the issues of actuator saturation which results windup needs to be addressed.

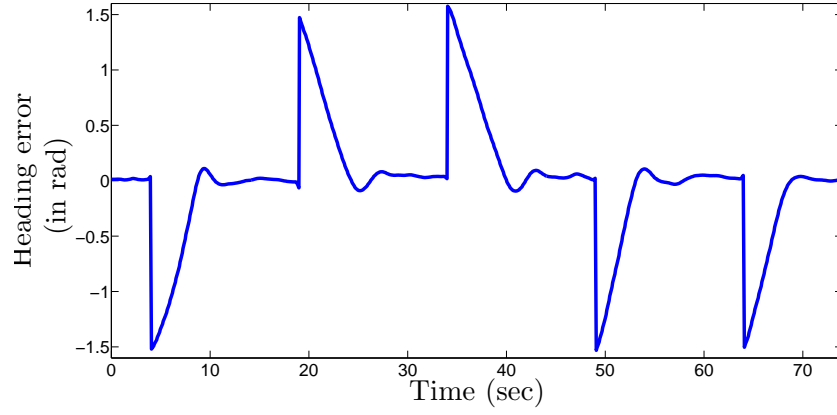


Figure 3.12: Heading error

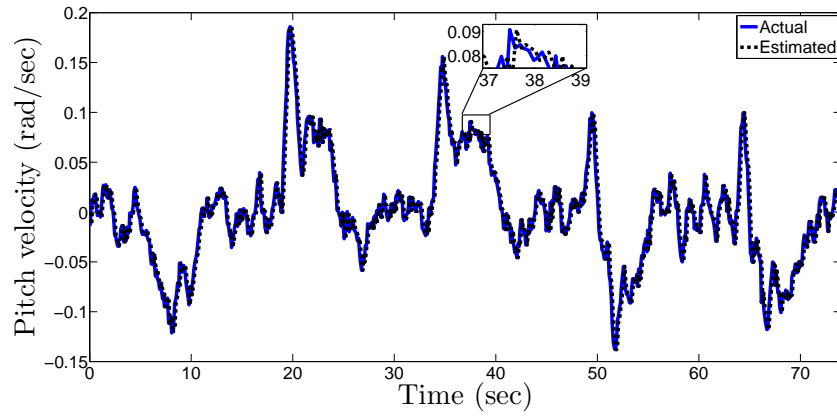


Figure 3.13: Pitch rate of the AUV during path follow

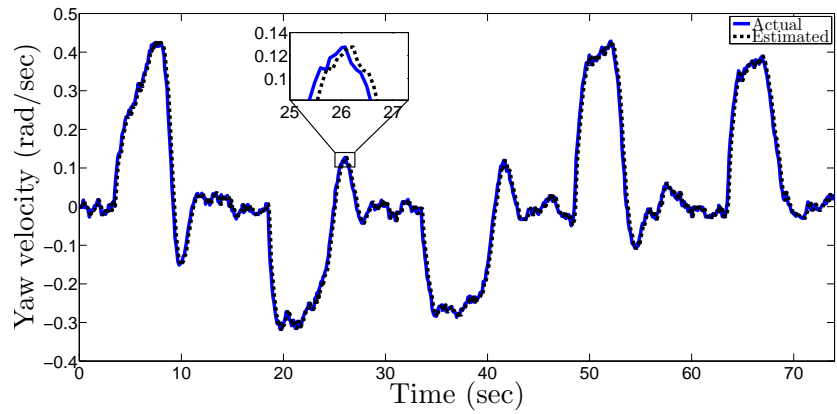


Figure 3.14: Yaw rate of the AUV during path follow

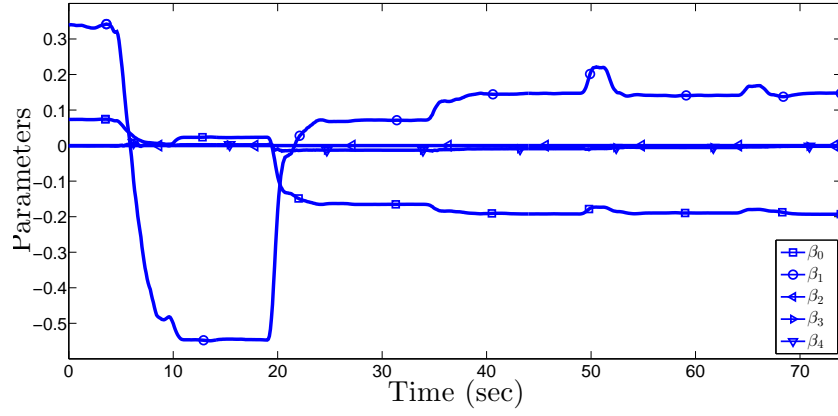


Figure 3.15: Updation of the NARMAX parameters

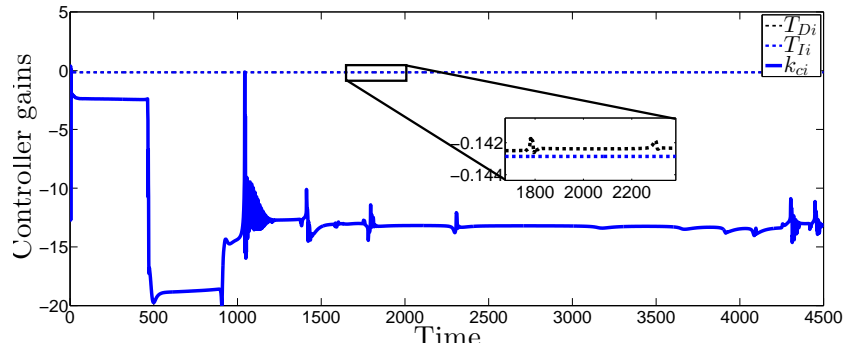


Figure 3.16: Updation of the controller gain parameters

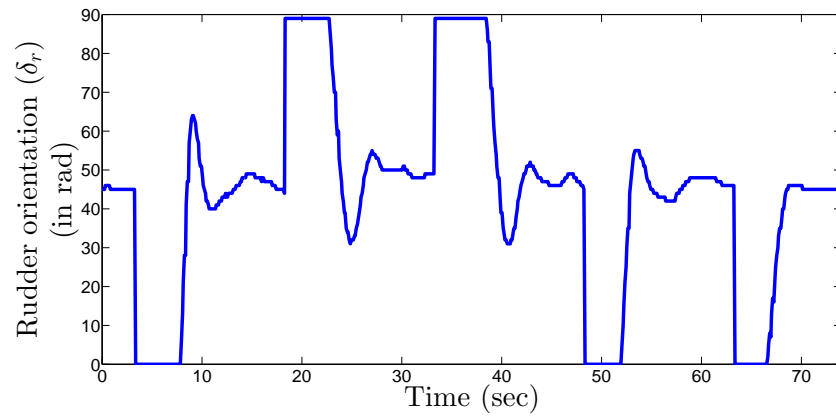


Figure 3.17: Rudder plane orientation while following a desired path

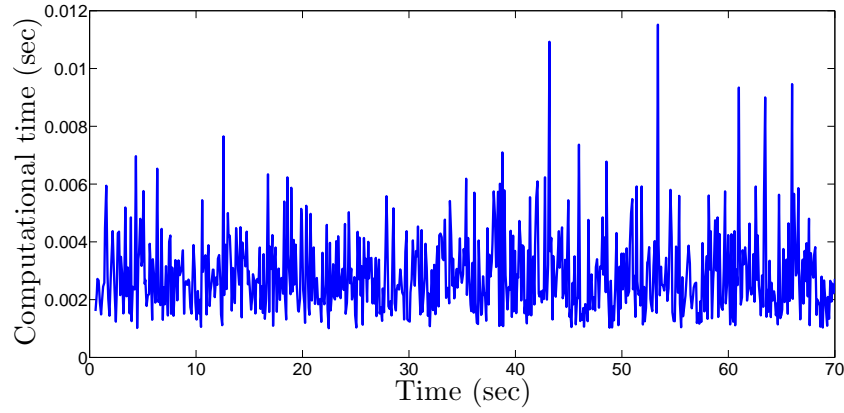


Figure 3.18: Computational time required to generate the actuation signal

## 3.6 Chapter Summary

In this chapter, a LoS guidance control algorithm is proposed which involves two steps. Firstly, a kinematic controller is designed which provides the reference for the dynamic controller. The kinematic controller is designed using backstepping controller and stability is proved using the Lyapunov theory. For the design of dynamic controller, first the dynamic model of the AUV is identified using NARMAX model and the parameters of the AUV are updated using RELS method. An inverse optimal controller is applied to design a Self-Tuning PID controller. This PID controller generates the control signal for the AUV to achieve path following task.

## Chapter 4

# Constrained Self-Tuning Adaptive Controller for an AUV with MR-NARMAX structure

Parameters of the AUV dynamics may vary due to change in payload or physical structure. In view of resolving this parameter variation and obtaining efficient heading and diving motion control algorithm, in this chapter a constrained self-tuning controller (CSTC) is developed. A Nonlinear Auto-Regressive Moving Average exogenous (NARMAX) model is designed using the significant regressors to identify the AUV heading and diving dynamics respectively. The parameters of the NARMAX model are updated using a Recursive Extended Least Square (RELS) algorithm at each time instant. Further, using the identified model a CSTC law is designed to track desired waypoints using Line of Sight (LoS) guidance law. The controller gains are updated at each instant satisfying the actuator constraint and computational efficiency is studied in view of achieving practical implementation of the developed algorithm. The efficacy of the developed NARMAX based CSTC algorithm to track a given reference is verified in simulation as well as in experimental environment. From the obtained results, it is concluded that the developed control algorithm is effective for an AUV to track desired references.

The rest of the chapter is organized as follows. The design of a MR-NARMAX structure for identifying AUV dynamics and updatation of its parameters are presented

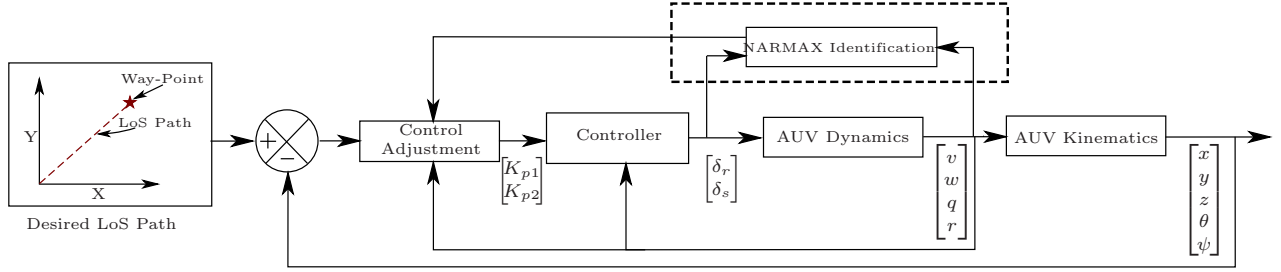


Figure 4.1: Controller Structure for the LOS Guidance law

in Section 4.2. Further, a constraint self-tuning adaptive control (CSTC) law for the realization of LoS guidance law is presented in Section 4.3. Both the algorithms are verified in an experimental environment which is discussed in Section 4.4. The chapter is concluded in Section 4.5.

## 4.1 Problem Statement

Referring to [64], the velocities  $\nu = [v_1 \ v_2]^T \in \mathbb{R}^4$  are defined in body-fixed frame  $\{B\}$  along surge, sway, heave, roll, pitch and yaw motions, whereas the position of the AUV  $\eta = [\eta_1 \ \eta_2]^T \in \mathbb{R}^5$  is defined w.r.t earth-fixed frame  $\{I\}$ .  $\eta_1 \in \mathbb{R}^3$  and  $\eta_2 \in \mathbb{R}^2$  are the linear and angular position of the AUV in  $\{I\}$ . To observe the motion of the AUV from  $\{I\}$ , a transformation between  $\{B\}$  and  $\{I\}$  is necessary. So using the transformation matrix  $J \in \mathbb{R}^{5 \times 4}$  from [9], the expression for velocities in  $\{I\}$  is given by

$$\dot{\eta} = J(\eta) \nu. \quad (4.1)$$

Equation (4.1) represents the kinematic description of the AUV, where  $\dot{\eta}$  is the velocity in  $\{I\}$ . Referring to [9], the dynamics of an AUV is given by

$$M\dot{\nu} + C_r(\nu) \nu + f_d(\nu) + r_s(\eta) = \tau. \quad (4.2)$$

As discussed in chapter 1, variation in the AUV structure or payload alters the AUV dynamics. Thus, a controller which depends on AUV parameters will not be effective as compared to an adaptive controller. Further, for practical implementation of this control strategy the physical limitations such as actuator constraints and computa-



tional efficiency should be considered. Therefore, an adaptive gain control strategy along with the constraints i.e. CSTC strategy as shown in Fig.4.1 is to be developed using the parameters obtained from a minimal-representation of an identified AUV dynamics.

## 4.2 Identification of the AUV Dynamics

A polynomial based NARMAX structure is designed in Chapter 3 for the identification of the AUV dynamics. But, NARMAX structure with redundant regressor terms affects the computational burden. Thus, a minimal representation of the polynomial-based NARMAX model i.e. MR-NARMAX using Forward Regression Orthogonal Least Square (FROLS) method is designed. Further, the parameters of the identified model are updated using a Recursive Extended Least Square (RELS) algorithm.

Referring to Assumption 3.5, common representation of the NARMAX structure for heading and diving motion is given by

$$\nu_k = f(\nu_{1,k-1}, \nu_{2,k-1}, u_{k-1}) + d_e e_{k-1}. \quad (4.3)$$

where  $\nu_k = [\nu_{1,k} \ \nu_{2,k}]^T \in \mathbb{R}^2$  is the estimated states,  $f = [f_1 \ f_2]^T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  is the non-linear polynomial function used to identify the dynamics. Let  $N$  is the number of regressor terms exist in a NARMAX structure then for  $i \in \{1, \dots, N\}$ ,  $\alpha_i$  and  $\beta_i$  are its parameters which are to be updated at each sampling time.  $d_e = [d_{e1} \ d_{e2}]^T \in \mathbb{R}^2$  are the disturbance parameters and  $e_{k-1}$  is the noise input. Eq (4.3) can be written as follows

$$\nu_{1,k} = f_1(\nu_{1,k-1}, \nu_{2,k-1}, u_{k-1}) + d_{e1} e_{k-1} \quad (4.4a)$$

$$\nu_{2,k} = f_2(\nu_{1,k-1}, \nu_{2,k-1}, u_{k-1}) + d_{e2} e_{k-1} \quad (4.4b)$$

where

$$\begin{aligned} f_1(\cdot) = & \alpha_0 \nu_{1,k-1} + \alpha_1 \nu_{2,k-1} + \alpha_2 \nu_{1,k-1}^2 + \alpha_3 \nu_{2,k-1}^2 \\ & + \alpha_4 \nu_{1,k-1} \nu_{2,k-1} + \alpha_5 u_{k-1}, \end{aligned}$$

$$f_2(\cdot) = \beta_0 \nu_{1,k-1} + \beta_1 \nu_{2,k-1} + \beta_2 \nu_{1,k-1}^2 + \beta_3 \nu_{2,k-1}^2 \\ + \beta_4 \nu_{1,k-1} \nu_{2,k-1} + \beta_5 u_{k-1}.$$

Using FROLS algorithm [1] significant state regressor terms of (4.4a) and (4.4b) are identified. Let  $K$  samples are obtained through experimentation, then the equation (4.4a) and (4.4b) can be represented as

$$V_i = P_i \omega_i, \text{ for } i \in \{1, 2\} \quad (4.5)$$

where  $V_i = [\nu_{i,1}, \dots, \nu_{i,K}]^T$  is the output vector,  $\omega_i \in \mathbb{R}^N$  is the parameter vector and  $P_i = [p_1, p_2, \dots, p_N] \in \mathbb{R}^{K \times N}$  is the regressor matrix respectively.

**Example 1.** For  $i = 1$ ,

$$\nu_1 = [\nu_{1,1}, \nu_{1,2}, \dots, \nu_{1,K}]^T, \omega_1 = [\alpha_0 \dots \alpha_4]^T, \\ P_1 = \begin{bmatrix} \nu_{1,0} & \nu_{2,0} & \nu_{1,0}^2 & \nu_{2,0}^2 & \nu_{1,0}\nu_{2,0} \\ \nu_{1,1} & \nu_{2,1} & \nu_{1,1}^2 & \nu_{2,1}^2 & \nu_{1,1}\nu_{2,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \nu_{1,K-1} & \nu_{2,K-1} & \nu_{1,K-1}^2 & \nu_{2,K-1}^2 & \nu_{1,K-1}\nu_{2,K-1} \end{bmatrix}$$

Similarly for  $i = 2$ ,  $\nu_2$ ,  $\omega_2$  and  $P_2$  can be derived.

For a matrix  $P_i$ , an orthogonal matrix  $L_i \in \mathbb{R}^{K \times N}$  and an upper triangular matrix  $\Upsilon_i \in \mathbb{R}^{N \times N}$  exist for which

$$P_i = L_i \Upsilon_i, \quad (4.6)$$

is always satisfied.  $L_i$  consists of orthogonal columns as  $(l_1, l_2, \dots, l_N)$  and referring to [1] each element of  $\Upsilon_i$  is defined as

$$\gamma_{j,m} = \frac{\langle p_m, l_j \rangle}{\langle l_r, l_j \rangle}, \quad (4.7)$$

for  $1 \leq j \leq m-1$  and  $m = 2, 3, \dots, N$ . Rewriting (4.5), by substituting  $P_i$  from (4.6) as

$$\nu_i = L_i G_i, \quad (4.8)$$

where  $G_i = [g_1, g_2, \dots, g_N]^T = \epsilon^{-1} L_i^T \nu_i$  and  $\epsilon = L_i^T L_i$ . The estimated parameters using FROLS is given by

$$\omega_i = \Upsilon_i^{-1} G_i. \quad (4.9)$$

To avoid overstated NARMAX structure (4.4), Error Reduction Ratio (ERR) can be used to identify the significant regressor terms. ERR is defined as

$$ERR_i = g_i^2 \frac{\langle l_i, l_i \rangle}{\nu_i, \nu_i}. \quad (4.10)$$

The regressor terms with significant ERR value are considered to design the minimal representation of (4.4) i.e. MR-NARMAX. The modified NARMAX structure is represented as

$$\begin{bmatrix} \nu_{1,k} \\ \nu_{2,k} \end{bmatrix} = \begin{bmatrix} \tilde{f}_1(\nu_{1,k-1}, \nu_{2,k-1}, u_{k-1}) \\ \tilde{f}_2(\nu_{1,k-1}, \nu_{2,k-1}, u_{k-1}) \end{bmatrix} + \begin{bmatrix} d_{e1} \\ d_{e2} \end{bmatrix} e_{k-1}, \quad (4.11)$$

where  $\tilde{f}_1$  and  $\tilde{f}_2$  consist of significant model terms.

**Remark 4.1.** For heading motion  $[\nu_{1,k} \ \nu_{2,k}]^T = [v_k \ r_k]^T$  is considered, whereas for diving motion  $[\nu_{1,k} \ \nu_{2,k}]^T = [w_k \ q_k]^T$  respectively. Since,  $\nu_{1,k}$  is the velocity along sway or heave motion, so for an underactuated AUV  $\alpha_5 = 0$ .

For  $s$  number of significant regressors, (4.11) can be modelled as

$$\nu_{i,k} = \tilde{\phi}_{i,k-1}^T \hat{\omega}_{i,k-1}, \text{ for } i \in \{1, 2\}, \quad (4.12)$$

where  $\tilde{\phi}_{i,k-1} \in \mathbb{R}^s$  and  $\hat{\omega}_{i,k-1} \in \mathbb{R}^s$  are the significant regressor vector and estimated parameter vector respectively. Then, a Recursive Extended Least Square (RELS) method is employed for parameter estimation as

$$\begin{aligned} \hat{\omega}_{i,k} &= \hat{\omega}_{i,k-1} + \frac{S_{i,k-1} \tilde{\phi}_{i,k-1}}{\lambda_k + \tilde{\phi}_{i,k-1}^T S_{i,k-1} \tilde{\phi}_{i,k-1}} \varepsilon_{i,k-1}, \\ S_{i,k} &= \frac{1}{\lambda_{k-1}} \left\{ S_{i,k-1} - \frac{S_{i,k-1} \tilde{\phi}_{i,k-1} \tilde{\phi}_{i,k-1}^T S_{i,k-1}}{\lambda_{k-1} + \tilde{\phi}_{i,k-1}^T S_{i,k-1} \tilde{\phi}_{i,k-1}} \right\}, \\ \nu_{i,k} &= \tilde{\phi}_{i,k-1}^T \hat{\omega}_{i,k-1} + \varepsilon_{i,k}. \end{aligned} \quad (4.13)$$

where  $\lambda_k$ ,  $S_{i,k}$  and  $\varepsilon_{i,k-1}$  are the forgetting factor, covariance matrix and residual error

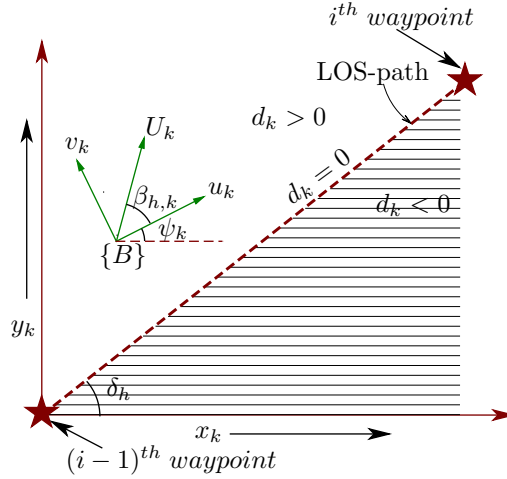


Figure 4.2: Tracking of Line of Sight path

output. The parameters obtained from (4.13) are used for developing an adaptive controller to achieve waypoint tracking for an AUV as discussed in the subsequent sections.

### 4.3 NARMAX Self-Tuning Controller Design

This section develops a control scheme for LoS guidance law by considering the identified AUV dynamics from Section 4.2. The line of sight guidance law can further be exploited for realizing way-point tracking algorithm.

Considering the Assumption 3.4, the kinematics of the AUV along the heading and diving motion is expressed as

$$x_k = x_{k-1} + T_s \left( u'_{k-1} \cos \psi_{k-1} - v_{k-1} \sin \psi_{k-1} \right), \quad (4.14a)$$

$$y_k = y_{k-1} + T_s \left( u'_{k-1} \sin \psi_{k-1} + v_{k-1} \cos \psi_{k-1} \right), \quad (4.14b)$$

$$z_k = z_{k-1} + T_s (-u_c \sin \theta_{k-1} + w_{k-1} \cos \theta_{k-1}), \quad (4.14c)$$

$$\theta_k = \theta_{k-1} + T_s q_{k-1}, \quad (4.14d)$$

$$\psi_k = \psi_{k-1} + T_s \sec \theta_{k-1} r_{k-1}, \quad (4.14e)$$

where  $u'_{k-1} = (u_c \cos \theta_{k-1} + w_{k-1} \sin \theta_{k-1})$  is the projection of the net velocity along the

surge motion. Using this kinematics expression (4.14), a controller is to be designed for tracking a rectilinear path connecting two way-points as shown in Fig.4.2. The variable  $d_k$  defined in Fig.4.2 is expressed as

$$d_k = \langle A, X_k - X_0 \rangle - C, \quad (4.15)$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & 0 & a_{23} \end{bmatrix}, \quad X_k - X_0 = \begin{bmatrix} x_k - x_0 \\ y_k - y_0 \\ z_k - z_0 \end{bmatrix},$$

$$C = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}.$$

To achieve LoS guidance by the AUV, the error between orientation angle  $(\psi_k + \beta_k)$  and slope of the LoS path should be reduced. Thus, the heading error  $(e_{h,k})$  is expressed as

$$e_{h,k} = \psi_k + \beta_k - \delta_h - \psi_{des,k}, \quad (4.16)$$

where  $\beta_k = \tan^{-1}\left(\frac{v_k}{u_c}\right)$  is the angle made by surge and sway velocities and  $\delta_h = \tan^{-1}\left(-\frac{a_{11}}{a_{22}}\right)$  is the slope of the desired path in x-y axis.  $\psi_{des,k}$  is the desired approaching angle of the AUV towards the desired path which is expressed as

$$\psi_{des,k} = \theta_A \tanh(k_\delta h_d), \quad (4.17)$$

where  $h_d = d_k(1)$  is the distance between the AUV and the desired path (4.15).  $\theta_A$  and  $k_\delta$  are the maximum approaching angle and constant parameters respectively. Substituting the  $\psi_k$  from (4.14e) into (4.16) results

$$e_{h,k} = c_1 + c_2 r_{k-1}, \quad (4.18)$$

where

$$c_1 = \psi_{k-1} + \beta_{h,k} - \psi_{des,k} - \delta_h,$$

$$c_2 = T_s \left( \frac{1}{\cos \theta_{k-1}} \right).$$

Similarly, the depth error between the AUV position and the desired path can be expressed as

$$e_{d,k} = \theta_k + \beta_{d,k} - \theta_{des,k} - \delta_d, \quad (4.19)$$

where  $\beta_{d,k} = \tan^{-1} \left( \frac{w_k}{u_c} \right)$  is the angle made by heave and sway velocities and  $\delta_d = \tan^{-1} \left( -\frac{a_{21}}{a_{23}} \right)$  is the slope of the desired path in x-z axis.  $\theta_{des,k}$  is the desired approaching angle of the AUV towards the desired path, which can be expressed as

$$\theta_{des,k} = \theta_A \tanh(k_\delta d_d), \quad (4.20)$$

where  $d_d = d_k(2)$  is the depth distance from the desired path (4.15) and  $\theta_A, k_\delta$  are the same as defined in (4.17). Substituting the expression of  $\theta_k$  from (4.14d) into (4.19) one obtains

$$e_{d,k} = c_3 + c_4 q_{k-1}, \quad (4.21)$$

where

$$\begin{aligned} c_3 &= \theta_{k-1} + \beta_{d,k} - \theta_{d,k} - \delta_d, \\ c_4 &= T_s. \end{aligned}$$

The orientation errors along heading and diving are modelled as given in (4.18) and (4.21). Thus, the objective here is to reduce these errors such that the AUV orients towards the desired path. Considering the error in heading( $e_{h,k}$ ), depth( $e_{d,k}$ ) and position error( $d_k$ ), an error objective function  $V_1$  for AUV kinematics is chosen as

$$V_1 = \frac{1}{2} e_k^T R e_k + \frac{1}{2} d_k^T Q d_k, \quad (4.22)$$

where  $e_k = \begin{bmatrix} e_{h,k} & e_{d,k} \end{bmatrix}^T$  and  $R \succ 0$  and  $Q \succ 0$  are the weight matrices for the orientation and distance error. Substituting (4.16) and (4.19) into (4.22), the error

objective function (4.22) can be expressed as

$$V_1 = \frac{1}{2} (c_1^2 + c_3^2 + d_k^T Q d_k) + c_1 c_2 r_{k-1} + c_3 c_4 q_{k-1}. \quad (4.23)$$

The desired angular velocities  $r_{d,k}$  and  $q_{d,k}$  are obtained by solving (4.23). Thus, the objective is to design a control law such that the error between actual and desired angular velocities becomes zero.

In (4.3), considering  $r_k \approx \tilde{r}_k$  and  $q_k \approx \tilde{q}_k$ , the NARMAX model of the heading and depth can be rewritten as

$$\begin{aligned} r_k &= f_{2,h}(v_{k-1}, r_{k-1}) + g_{2,h} \delta_{r,k-1}, \\ q_k &= f_{2,d}(w_{k-1}, q_{k-1}) + g_{2,d} \delta_{s,k-1}. \end{aligned} \quad (4.24)$$

The controller is considered as

$$\begin{bmatrix} \delta_{r,k-1} \\ \delta_{s,k-1} \end{bmatrix} = \begin{bmatrix} K_{p1} & 0 \\ 0 & K_{p2} \end{bmatrix} \begin{bmatrix} r_{e,k-1} \\ q_{e,k-1} \end{bmatrix}, \quad (4.25)$$

where  $K_{p1}$  and  $K_{p2}$  denote the controller gains for the rudder and stern planes respectively. Using (4.25), the error dynamics for the heading and depth motion can be represented as follows

$$\begin{aligned} r_{e,k} &= f_{2,h}(v_{k-1}, r_{k-1}) + g_{2,h} K_{p1} r_{e,k-1} - r_{d,k} \\ q_{e,k} &= f_{2,d}(w_{k-1}, q_{k-1}) + g_{2,d} K_{p2} q_{e,k-1} - q_{d,k}. \end{aligned} \quad (4.26)$$

Thus, considering the errors  $r_{e,k}$  and  $q_{e,k}$ , an error objective function  $V_2$  for AUV dynamics can be chosen as

$$V_2 = \frac{1}{2} r_{e,k}^2 + \frac{1}{2} q_{e,k}^2. \quad (4.27)$$

Using (4.26) in equation (4.27) yields

$$\begin{aligned} V_2 &= \frac{f_{2,h}^2}{2} - f_{2,h} r_{d,k} + f_{2,h} g_{2,h} K_{p1} r_{e,k-1} + \frac{r_{d,k}^2}{2} \\ &\quad + \frac{g_{2,h}^2}{2} K_{p1}^2 r_{e,k-1}^2 - g_{2,h} r_{d,k} K_{p1} r_{e,k-1} + \frac{f_{2,d}^2}{2} \end{aligned}$$

$$\begin{aligned}
& -f_{2,d}q_{d,k} + f_{2,d}g_{2,d}K_{p2}q_{e,k-1} + \frac{g_{2,d}^2}{2}K_{p2}^2q_{e,k-1}^2 \\
& + \frac{q_{d,k}^2}{2} - g_{2,d}q_{d,k}K_{p2}q_{e,k-1}.
\end{aligned} \tag{4.28}$$

An objective function which consists of both kinematics and dynamics error objective function is represented as

$$V = V_1 + V_2, \tag{4.29}$$

where  $V_1$  and  $V_2$  are defined in (4.22) and (4.27) respectively. Minimizing the objective function  $V$  using the suitable  $r_d$ ,  $q_d$ ,  $K_{p1}$  and  $K_{p2}$  will ensure that the kinematics error and dynamics error gradually reduce to zero. Eq (4.29) can be minimized in the presence of actuation constraint using a special class of convex optimization problem i.e. quadratic programming [68]. The modified equation is expressed as

$$\begin{aligned}
& \min \quad \frac{1}{2}X^T Q X + C^T X + C_f \\
& \text{subj. to } r_{d,k} \in \mathcal{X}_r, \quad q_{d,k} \in \mathcal{X}_q, \\
& \quad \quad K_{p1} \in \mathcal{X}_{p1}, \quad K_{p2} \in \mathcal{X}_{p2},
\end{aligned} \tag{4.30}$$

where

$$\begin{aligned}
X &= \begin{bmatrix} r_{d,k} & q_{d,k} & K_{p1} & K_{p2} \end{bmatrix}^T, \\
Q &= \begin{bmatrix} 1 & 0 & -g_{2,h}r_{e,k-1} & 0 \\ 0 & 1 & 0 & -g_{2,d}q_{e,k-1} \\ -g_{2,h}r_{e,k-1} & 0 & 2\frac{g_{2,h}^2}{2}q_{e,k-1}^2 & 0 \\ 0 & -g_{2,d}q_{e,k-1} & 0 & 2\frac{g_{2,d}^2}{2}r_{e,k-1}^2 \end{bmatrix}, \\
C^T &= \begin{bmatrix} c_1c_2 - f_{2,h} & c_3c_4 - f_{2,d} & f_{2,h}g_{2,h}r_{e,k-1} & f_{2,d}g_{2,d}q_{e,k-1} \end{bmatrix}, \\
C_f &= \frac{1}{2}(c_1^2 + c_3^2 + d^T Q d) + \frac{f_{2,h}^2}{2} + \frac{f_{2,d}^2}{2}.
\end{aligned}$$

The range of  $\mathcal{X}_r$  and  $\mathcal{X}_q$  can be selected by knowing the practical limitation of the yaw ( $r_k$ ) and pitch ( $q_k$ ) velocities of an AUV. Since,  $r_{d,k} \in \mathcal{X}_r$  and  $q_{d,k} \in \mathcal{X}_q$  the angular velocities error ( $r_{e,k-1}, q_{e,k-1}$ ) are always bounded, in view of the practical implementation of the developed controller it is necessary to incorporate the con-



Table 4.1: FROLS applied to INFANTE AUV heading motion

Regressor	Heave Motion		Pitch Motion	
	OLS Estimation	ERR	OLS Estimation	ERR
$v_{k-1}$	0.9560	0.9818	-0.185	$3.62e^{-4}$
$v_{k-1}^2$	0.0593	$1.28e^{-7}$	-13.489	$6.11e^{-5}$
$r_{k-1}$	0.0128	0.0136	0.7573	0.5748
$r_{k-1}^2$	$-2e^{-4}$	$6.87e^{-9}$	0.298	$1.8e^{-4}$
$v_{k-1}r_{k-1}$	0.002	$3.78e^{-9}$	-7.0228	$4.15e^{-4}$

straints for actuation signal. Thus, referring to (4.25) we define a range for controller gains  $(K_{p1}, K_{p2})$  such that the actuation signal  $(\delta_{r,k-1}, \delta_{s,k-1})$  does not exceed the limit of the control planes.  $\mathcal{X}_{p1}$  and  $\mathcal{X}_{p2}$  are the range of gains which ensures that  $\delta_{r,k-1} \in \mathcal{X}_{\delta_r} = \{\delta_{r,k-1} \in \mathbb{R} \mid \delta_{r,min} \leq \delta_{r,k-1} \leq \delta_{r,max}\}$  and  $\delta_{s,k-1} \in \mathcal{X}_{\delta_s} = \{\delta_{s,k-1} \in \mathbb{R} \mid \delta_{s,min} \leq \delta_{s,k-1} \leq \delta_{s,max}\}$ . The control law generated by solving quadratic constrained optimization problem (4.30) will be applied to verify the ability of the AUV to follow waypoints using LoS guidance law.

## 4.4 Results and Discussion

This section verifies the proposed MR-NARMAX based CSTC algorithm in simulation as well as in experimental environment. Firstly, a simulation study is performed using the parameters of an AUV to verify the algorithm. Later, an experimental study is conducted on the developed prototype AUV in order to test the algorithm for practical realization. The results and discussions are presented in the subsequent sections.

### 4.4.1 Simulation Results

Referring to Table 4.1, the threshold  $\gamma = 1e^{-5}$  is chosen for identifying significant regressor terms for designing the MR-NARMAX structure. For heading motion, MR-NARMAX structure for sway and yaw dynamics are chosen as

$$\begin{aligned}
 v_k &= \alpha_0 v_{k-1} + \alpha_1 r_{k-1} + d_{e,1} e_{1,k-1}, \\
 r_k &= \beta_0 v_{k-1} + \beta_1 r_{k-1} + \beta_2 v_{k-1}^2 + \beta_3 r_{k-1}^2 + \beta_4 v_{k-1} r_{k-1} \\
 &\quad + \beta_5 \delta_{r,k-1} + d_{e,2} e_{2,k-1}.
 \end{aligned} \tag{4.31}$$

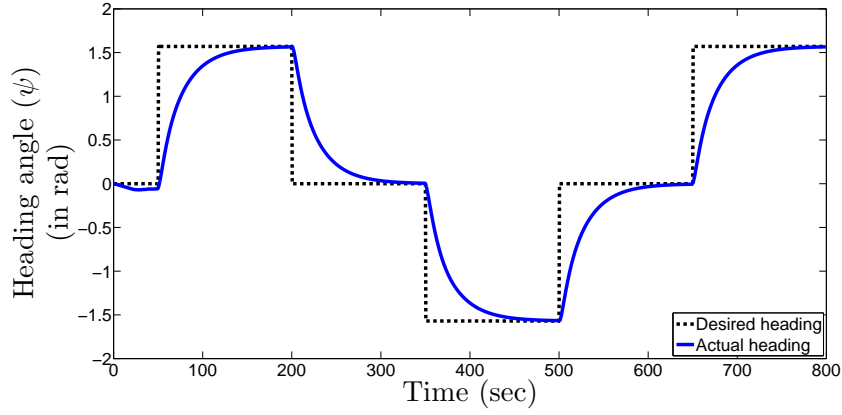


Figure 4.3: Tracking of heading reference by INFANTE AUV

The parameters of the INFANTE AUV [69] is considered and the control algorithm is verified in the MATLAB environment. The initial states of the AUV is taken as  $\begin{bmatrix} x & y & z & \theta & \psi & v & w & q & r \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$  and the parameters necessary for the implementation of the control algorithm are  $k_\delta = 0.1$ ,  $\theta_a = 0.785 \text{ rad}$  and  $u_c = 0.8 \text{ m/sec}$  respectively. Further, the constraint parameters are  $|r_{d,k}| \leq 1$ ,  $|q_{d,k}| \leq 1$ ,  $|K_{p1}| \leq 0.6$  and  $|K_{p2}| \leq 0.6$ .

In order to verify the control algorithm, a desired heading is provided as follows,

$$\psi_d = \begin{cases} 0, & 0 \leq t < 50 \\ 1.57, & 50 \leq t < 200 \\ 0, & 200 \leq t < 350 \\ -1.57, & 350 \leq t < 500 \\ 0, & 500 \leq t < 650 \\ 1.57, & 650 \leq t < 800 \end{cases} \quad (4.32)$$

and it is required that the actual heading of the AUV should track the desired heading. Referring to Fig. 4.3, the AUV successfully tracks the desired heading and the corresponding heading error is shown in Fig. 4.4. As discussed in section 4.2, a MR-NARMAX model is designed to capture the AUV dynamics and the parameters

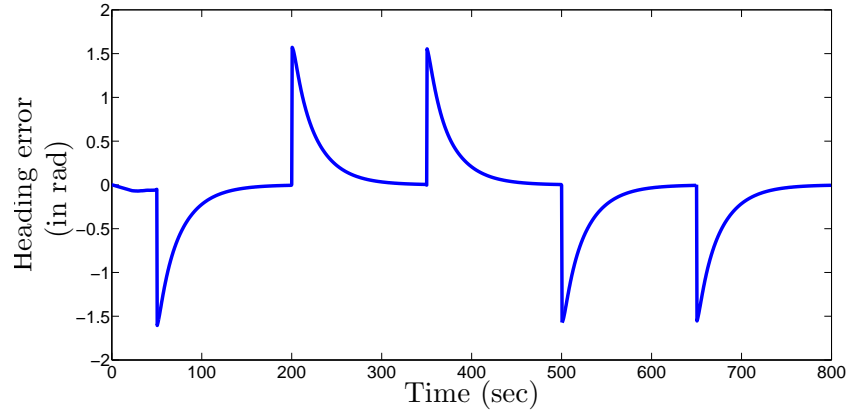


Figure 4.4: Heading error while tracking the desired reference

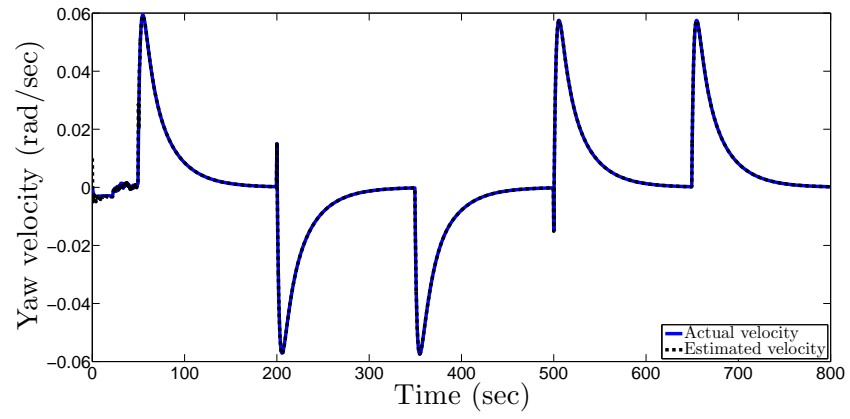


Figure 4.5: Estimated yaw velocity as compared to actual yaw velocity

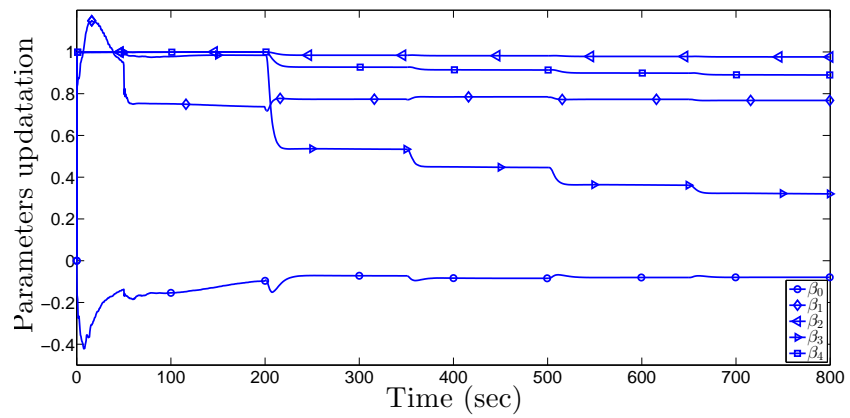


Figure 4.6: Updation of MR-NARMAX parameters

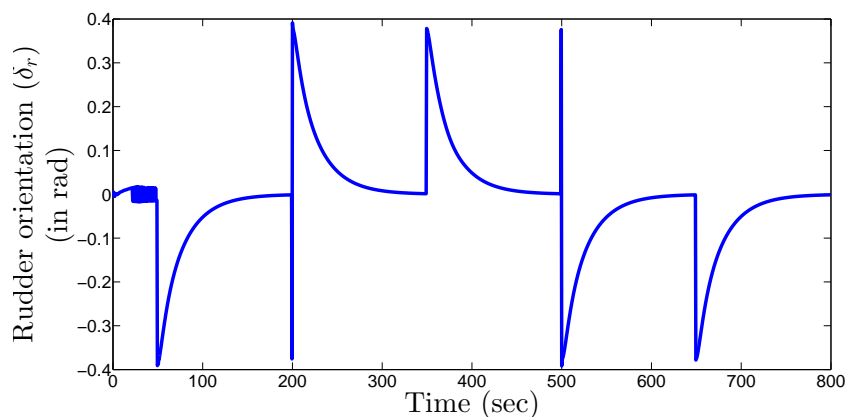


Figure 4.7: Actuation signal while tracking a desired heading

obtained can be used for the controller design. Referring to Fig. 4.5, the estimated yaw velocity successfully tracks the actual yaw velocity. Thus, it is evident that the MR-NARMAX structure successfully adapts to the AUV dynamics and the updation of the NARMAX parameters are shown in Fig. 4.6. These parameters are then further used solve the constrained quadratic programming problem (4.30), in order to generate the control law (4.25). The generated actuation signal i.e. rudder orientation is shown in Fig. 4.7 and it is clear that the actuation signal always lies within its constraint i.e.  $-0.785$  to  $0.785$ . From the above results, it is observed that the developed control algorithm is successful in steering the INFANTE AUV along the desired heading. Thus, in the subsequent section the MR-NARMAX based CSTC algorithm will be verified experimentally in the prototype AUV.

#### 4.4.2 Experimental Results

The developed control algorithm is verified on the developed prototype AUV discussed in chapter 1. The ROS nodes of the developed algorithm is shown in Fig.4.8 and the description of each node is given in Table 4.2. Various nodes communicate with each other using messages which comprises of state information, controller data etc. The characteristics of these messages such as bandwidth, frequency are presented in Table 4.3. Referring to Section 4.2, suitable MR-NARMAX model is derived for the prototype AUV. From an experimental trial, sensor and actuator data are captured with sampling time  $T_s = 0.1sec$  and Orthogonal Least Square is estimated for each

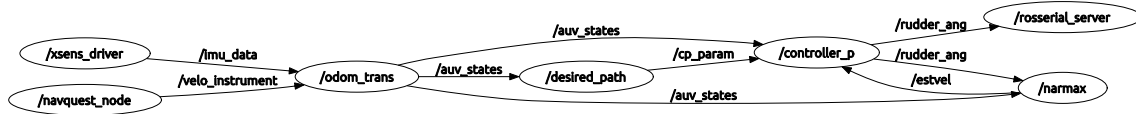


Figure 4.8: Implementation of constrained adaptive control strategy in ROS

Table 4.2: Description of ROS nodes

Node	Description
/xsens_driver	Driver node to access Xsens MTI INS sensor
/navquest_node	Driver node to access Navquest DVL sensor
/odom_trans	Generates AUV states
/desired_path_mppq	desired path node (4.15)
/narmax	NARMAX node for identification (4.3)
/controller_p	controller node (4.30)
/rosserial_server	Arduino node for transmitting actuation signal

Table 4.3: Description of ROS messages and its characteristics

Message	Description	Bandwidth	Publishing rate
/imu_data	$\phi, \theta, \psi, p, q, r$	32.8 Kb/sec	100
/auv_states	$z, \phi, \theta, \psi, u, v, w, p, q, r$	523 B/sec	10
/mppq_rudder	$K_{p1}$	525 B/sec	10
/mppq_data	$v_k, r_{e,k}, r_{d,k}$	522 B/sec	10
/rudder_ang	$\delta_r$	20 B/sec	10
/velo_instrument	$u, v, w$	20 B/sec	10

Table 4.4: FROLS applied to AUV Heading motion

Regressor	Sway Motion		Yaw Motion	
	OLS Estimation	ERR	OLS Estimation	ERR
$v_{k-1}$	0.9547	0.995	0.1810	$2.08e^{-4}$
$v_{k-1}^2$	0.016	$2.9e^{-9}$	6.41	$5.74e^{-5}$
$r_{k-1}$	0.0127	$3.6e^{-4}$	0.9433	0.766
$r_{k-1}^2$	-0.0002	$1.62e^{-10}$	0.2752	$2.16e^{-6}$
$v_{k-1}r_{k-1}$	0.0012	$2.26e^{-10}$	-2.773	$2.63e^{-5}$

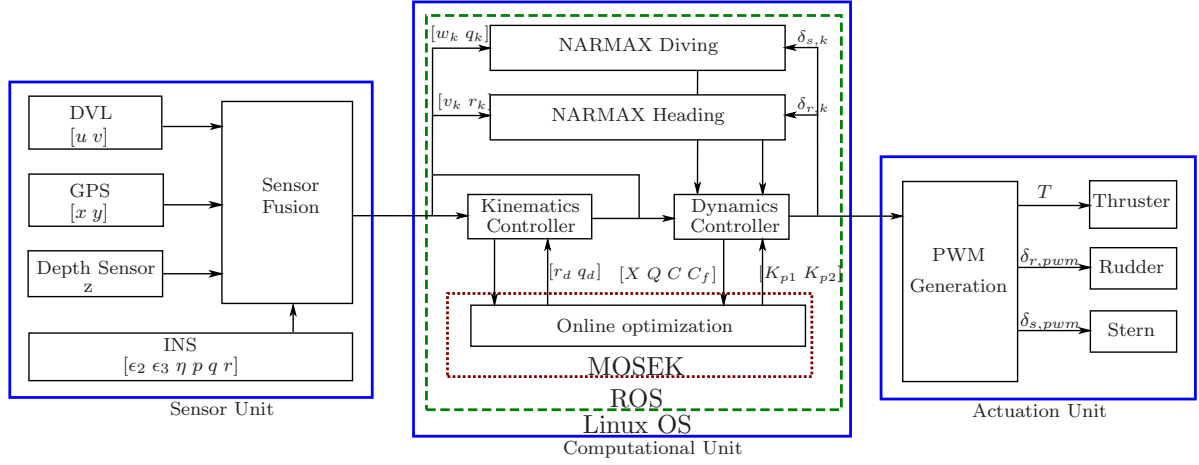


Figure 4.9: Implementation of the developed algorithm in the prototype AUV

regressor terms. Referring to Table 4.4, based on the high ERR i.e.  $ERR \geq \gamma$  where  $\gamma = 1e^{-5}$ , significant regressor terms are selected for designing MR-NARMAX structure. For heading motion, MR-NARMAX structure for sway and yaw dynamics are chosen as

$$\begin{aligned}
 v_k &= \alpha_0 v_{k-1} + \alpha_1 r_{k-1} + d_{e,1} e_{1,k-1}, \\
 r_k &= \beta_0 v_{k-1} + \beta_1 r_{k-1} + \beta_2 v_{k-1}^2 + \beta_3 r_{k-1}^2 + \beta_4 v_{k-1} r_{k-1} \\
 &\quad + \beta_5 \delta_{r,k-1} + d_{e,2} e_{2,k-1}.
 \end{aligned} \tag{4.33}$$

The RELS algorithm discussed in (4.12) is used to update the parameters of (4.33). The effectiveness of the MR-NARMAX based CSTC algorithm designed in Section 4.3 is verified using the heading reference shown in Fig.4.10 along with  $C = [0 \ 0]^T$  and  $X_0 = [0 \ 0 \ 0]^T$ . The initial condition for the decision variables are  $[r_{d,0} \ q_{d,0} \ K_{p1} \ K_{p2}] = [0 \ 0 \ 0 \ 0]$  and throughout the experimentation, a constant surge velocity  $u_c = 0.8 \text{ m/sec}$  is considered.

The developed algorithm is implemented on the prototype AUV with sampling time  $T_s = 0.1 \text{ sec}$ . A minimal representation of NARMAX model is derived using Table 4.4 and RLES is used to update its parameters. The implementation of the developed control algorithm is shown in Fig.4.9. The initial position and orientation of the AUV are  $[x \ y \ z \ \theta \ \psi \ v \ w \ q \ r]^T = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ , and the considered constraint

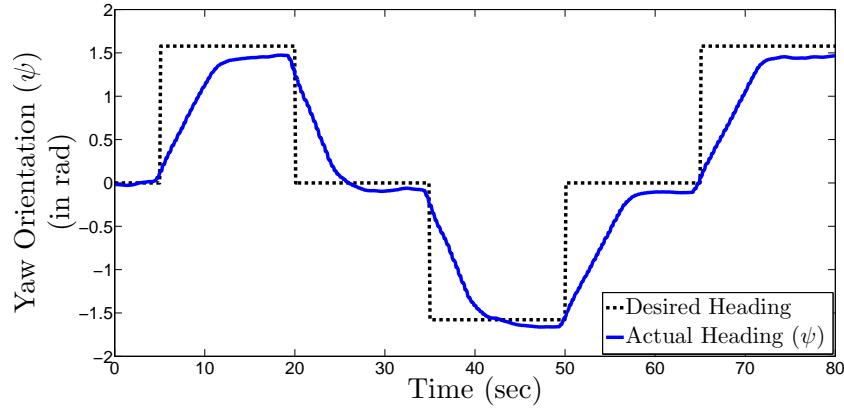


Figure 4.10: Tracking of desired heading by prototype AUV

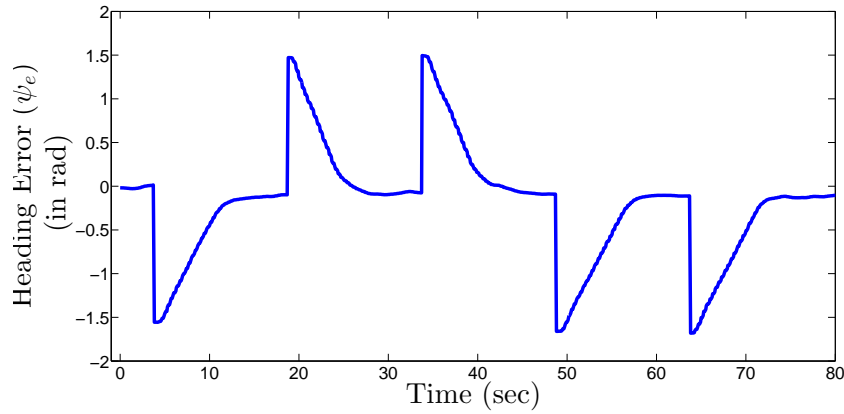


Figure 4.11: Heading error

parameters are  $|r_{d,k}| \leq 1$ ,  $|q_{d,k}| \leq 1$ ,  $|K_{p1}| \leq 0.6$  and  $|K_{p2}| \leq 0.6$ .

For experimental test, a lawn-mower type desired heading is considered and the result for tracking the desired heading is shown in Fig.4.10 and heading error is given in Fig.4.11. From the results it is verified that the controller is effective for real-time implementation. Referring to Fig.4.12 and Fig.4.13, the estimated velocities of the MR-NARMAX model are close to the actual velocities. Thus, the identified model closely matches with the AUV dynamics and the updatation of MR-NARMAX parameters are shown in Fig.4.14. Referring to Fig.4.15, the computational time taken by the MR-NARMAX model is  $0.4ms$  approx., it justifies that practical implementation of this identification technique. The parameters derived and the heading error are then used to generate the actuation signal by solving (4.25). The actuation signal which steers

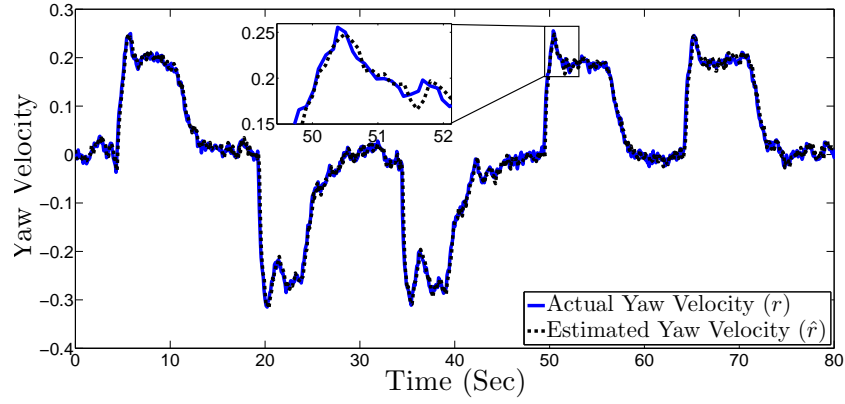


Figure 4.12: Comparison of estimated velocity and actual yaw velocity

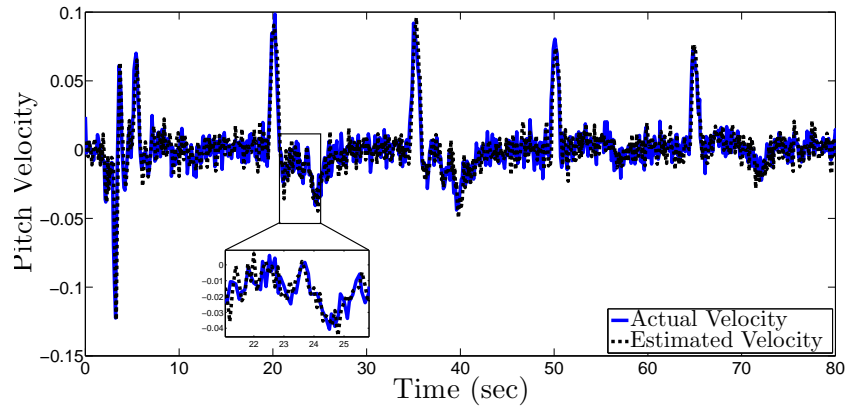


Figure 4.13: Comparison of estimated velocity and actual pitch velocity

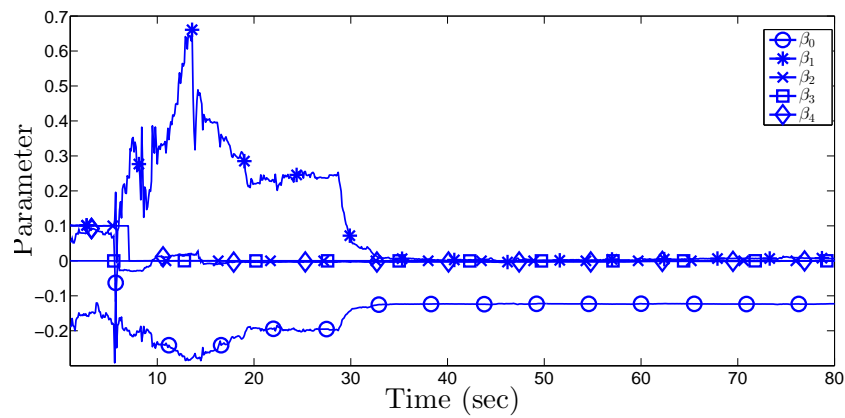


Figure 4.14: Updation of the NARMAX parameters



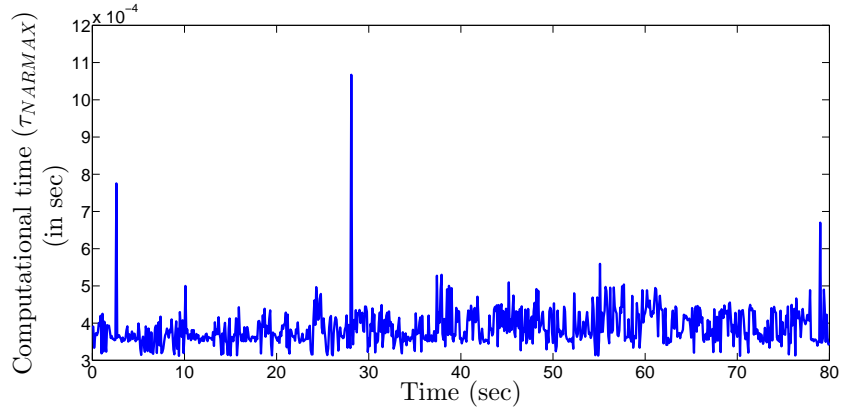


Figure 4.15: Computational performance of MR-NARMAX identification

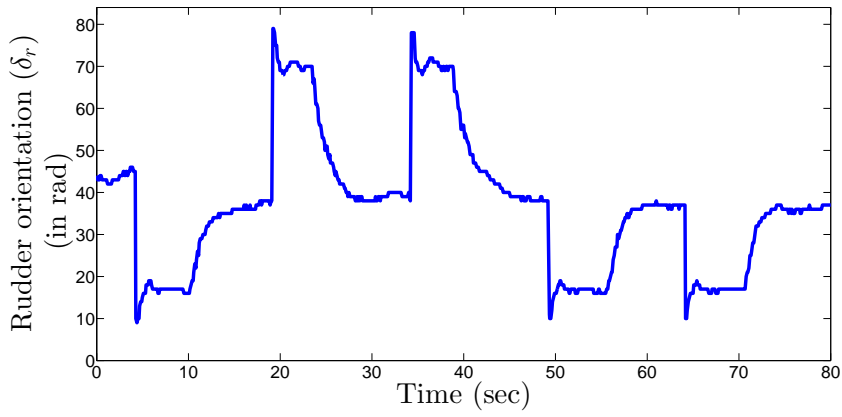


Figure 4.16: Control signal for rudder plane

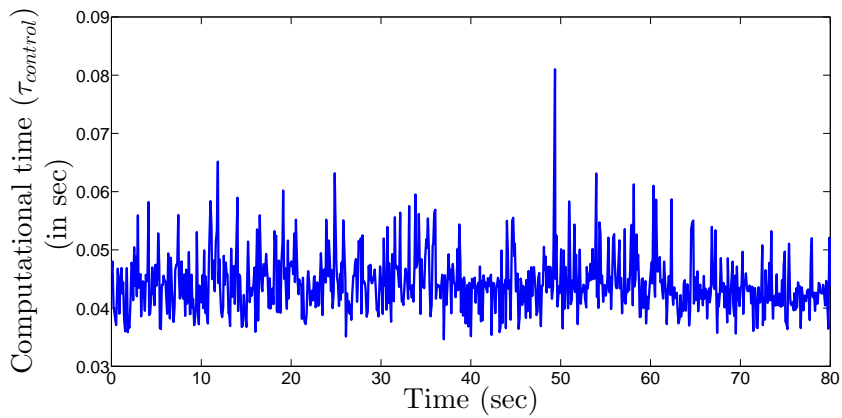


Figure 4.17: Controller computational performance

the AUV to track the desired heading is shown in Fig.4.16 and the computational time taken to solve the optimization problem is represented in Fig.4.17. From the results, it is seen that the maximum computational time is 0.08sec.

## 4.5 Chapter Summary

In this chapter, a CSTC algorithm is developed for an AUV to track waypoints in face of variation in the hydrodynamic parameters. To capture the unknown dynamics, a minimum representation of a NARMAX (MR-NARMAX) structure is identified which consists of significant regressors terms. The parameters of the MR-NARMAX model are updated using RELS algorithm at each time instant. Further, a constrained self-tuning controller is developed for a way-point tracking algorithm based on Line of Sight guidance law. The generated actuation signal complies with the actuator constraint for practical feasibility of the algorithm. The MR-NARMAX model with the waypoint tracking algorithm is successfully verified on the developed prototype AUV.

# Chapter 5

## Explicit model predictive control design for an AUV

In chapter 4, a constrained adaptive control has been designed using MR-NARMAX structure for heading motion as well as for diving motion. It solves a constrained quadratic optimization problem at every sampling instant in order to generate the optimal control gains. From the results obtained with MR-NARMAX based adaptive controller, it is evident that it is computationally expensive and parameters of the MR-NARMAX model are regularly updated even though the dynamics is not changing. In order to minimize this computational burden, an explicit controller is intended to be designed which can be implemented in parallel with the control algorithms developed in the previous chapters. Hence, in this chapter, design of an explicit Model Predictive (MPC) controller is developed to implement the LoS guidance law for an AUV to follow a desired path. In view of practical realization of the algorithm, this controller is designed using both the state and actuator constraints. Subsequently, the control algorithm is implemented on a prototype AUV developed in the laboratory as described in chapter 2.

The rest of the chapter is organized as follows. Section 5.1 describes the problem statement. Design of proposed explicit controller considering both kinematics and dynamics are presented in Section 5.2. In Section 5.3, both the simulation and experimental results using the proposed explicit controller for implementation of LoS guidance are presented. Then, this chapter is concluded in Section 5.4.

## 5.1 Problem Statement

Referring to [64], the motion of an AUV is defined using an earth-fixed frame  $\{I\}$  and a body-fixed frame  $\{B\}$ . The position vector of the AUV is measured with respect to  $\{I\}$  and the velocity vector is measured with respect to  $\{B\}$ . In order to observe the motion of the AUV w.r.t  $\{I\}$ , a transformation from  $\{B\}$  to  $\{I\}$  is necessary which is given by

$$\dot{\eta} = E(\alpha) \nu. \quad (5.1)$$

where  $E(\alpha)$  is the transformation matrix in terms of quaternion orientation  $\alpha$ ,  $\eta = \begin{bmatrix} \eta_1 & \alpha \end{bmatrix}^T$  is the position vector in  $\{I\}$  frame and  $\nu$  is the velocity vector in  $\{B\}$ . Further,  $\eta_1$  is the linear position of the AUV and  $\alpha$  is the quaternion orientation of the AUV. Equation (5.1) represents the kinematic description of the AUV for heading motion which is based on quaternion orientation in order to avoid the singularity problem at  $\theta = 0$  of the euler orientation.

The dynamics of the AUV can be described as

$$M\dot{\nu} + C(\nu)\nu + f_d(\nu) + g(\eta) = \tau \quad (5.2)$$

where  $M$  denotes the mass matrix,  $Cv$  is the Coriolis term,  $f_d(v)$  is the damping term

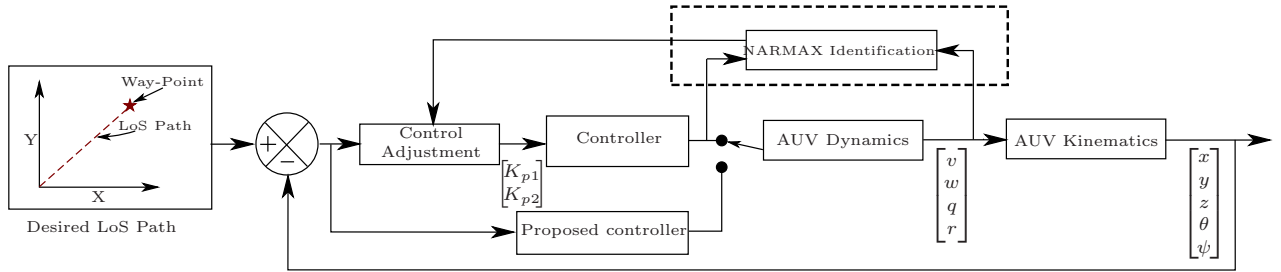


Figure 5.1: Controller Structure for the LOS Guidance law

and  $g(\eta)$  is the restoring term. The details of these terms are given in Appendix A. The external force  $\tau$  is the control input required to steer the AUV along the desired path. The detail description of the AUV dynamics is presented in appendix A.

In chapter 3, an adaptive controller is designed which identifies the parameters of the NARMAX model on-line and using those identified parameters, an adaptive control law is derived for the implementation of LoS guidance. However, it is to be noted that

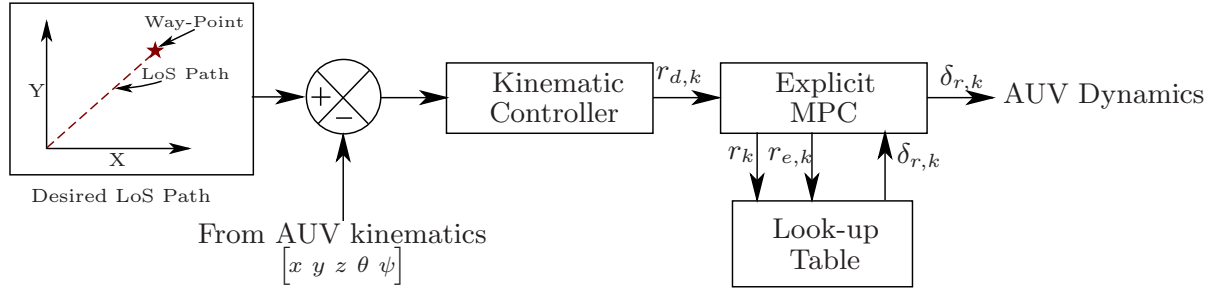


Figure 5.2: Explicit control design for AUV

the system identification technique continuously identifies the dynamics even though the AUV dynamics is not changing and an optimization problem is solved at every sampling instant which is computationally expensive. Therefore, in this chapter using the kinematics (5.1) and the identified AUV dynamics, an explicit model predictive control strategy is developed to reduce the computational burden of the control algorithm. The structure of the proposed controller is shown in Fig.5.1, where the explicit controller is to be implemented in parallel with the controller developed in chapter 4.

## 5.2 Explicit Control Design

This section describes the design of an explicit controller for an AUV using the kinematics and dynamics in order to realize a LoS guidance scheme as shown in Fig.5.2. Firstly, using the desired path parameters, a Lyapunov based backstepping controller is designed using discrete form of AUV kinematics. Using the Euler's first order method with sampling time  $T_s$ , the kinematic equation of the AUV (5.1) is discretized as follows,

$$\eta_k = \eta_{k-1} + T_s (E(\alpha_{k-1})) \nu_{k-1} \quad (5.3)$$

Referring to Appendix A, the transformation matrix  $E(\alpha_{k-1})$  is defined as

$$E(\alpha_{k-1}) = \begin{bmatrix} E_1(\alpha_{k-1}) & 0_{3 \times 3} \\ 0_{4 \times 3} & E_2(\alpha_{k-1}) \end{bmatrix}. \quad (5.4)$$

The details of the kinematic equation based on quaternion orientation (5.3) is discussed in Appendix A.

Considering the heading motion of an AUV,  $\eta_k = [\eta_{1,k} \ \alpha_k]^T$  which constituents  $\eta_{1,k} = [x_k \ y_k]^T$  is the translational position and  $\alpha_k = [\alpha_{2,k} \ \alpha_{3,k}]^T$  is the quaternion orientation. The transformation matrix for AUV heading motion is defined as follows

$$E(\alpha_{k-1}) = \begin{bmatrix} 1 - 2\alpha_{2,k-1}^2 & -2\alpha_{2,k-1}\alpha_{3,k-1} & 0 \\ 2\alpha_{2,k-1}\alpha_{3,k-1} & 1 - 2\alpha_{2,k-1}^2 & 0 \\ 0 & 0 & \frac{\alpha_{3,k-1}}{2} \\ 0 & 0 & \frac{-\alpha_{2,k-1}}{2} \end{bmatrix} \quad (5.5)$$

Thus, by substituting  $E(\alpha_{k-1})$  from (5.5), the kinematic equation (5.3) is expressed as

$$x_k = x_{k-1} + T_s \left( (1 - 2(\alpha_{2,k-1}^2)) u - 2\alpha_{2,k-1}\alpha_{3,k-1}v_{k-1} \right) \quad (5.6a)$$

$$y_k = y_{k-1} + T_s \left( 2\alpha_{2,k-1}\alpha_{3,k-1}u + (1 - 2\alpha_{2,k-1}^2)v_{k-1} \right) \quad (5.6b)$$

$$\alpha_{2,k} = \alpha_{2,k-1} + \frac{T_s}{2}\alpha_{3,k-1}r_{k-1} \quad (5.6c)$$

$$\alpha_{3,k} = \alpha_{3,k-1} - \frac{T_s}{2}\alpha_{2,k-1}r_{k-1}. \quad (5.6d)$$

The quaternion orientation for the desired LoS path is  $(\alpha_2^d, \alpha_3^d)$  and an earth-fixed frame  $\{I_m\}$  is defined as shown in Fig.5.3. The position of the AUV with reference to  $\{I_m\}$  is given by

$$\begin{bmatrix} dx_k \\ dy_k \end{bmatrix} = \begin{bmatrix} 1 - 2(\alpha_2^d)^2 & 2\alpha_2^d\alpha_3^d \\ -2\alpha_2^d\alpha_3^d & 1 - 2(\alpha_2^d)^2 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} \quad (5.7)$$

From Fig. 5.3, the cross track error is given by

$$dy_k = -2\alpha_2^d\alpha_3^dx_k + (1 - 2(\alpha_2^d)^2)y_k. \quad (5.8)$$

By substituting  $x_k$  and  $y_k$  from (5.6) in (5.8), the modified cross-track error is expressed as

$$dy_k = dy_{k-1} + T_s \left( (\alpha_3^d\alpha_{2,k-1} - \alpha_2^d\alpha_{3,k-1}) (\alpha_{3,k-1}\alpha_2^d - \alpha_{2,k-1}\alpha_2^d) \right). \quad (5.9)$$

To minimize the cross-track error  $dy_k$ , a Lyapunov function  $V_{1,k}$  is chosen as follows

$$V_{1,k} = \frac{1}{2}dy_k^2. \quad (5.10)$$



can be expressed as

$$\begin{aligned}\alpha_{2d,k-1} &= \sin\left(\frac{\lambda_d - 2\tan^{-1}(K_\delta dy_{k-1})}{2}\right) \\ &= \frac{\alpha_2^d - \alpha_3^d K_\delta dy_{k-1}}{\sqrt{1 + (K_\delta dy_{k-1})^2}},\end{aligned}\quad (5.14)$$

and

$$\begin{aligned}\alpha_{3d,k-1} &= \cos\left(\frac{\lambda_d - 2\tan^{-1}(K_\delta dy_{k-1})}{2}\right) \\ &= \frac{\alpha_3^d + \alpha_2^d K_\delta dy_{k-1}}{\sqrt{1 + (K_\delta dy_{k-1})^2}}.\end{aligned}\quad (5.15)$$

To minimize the difference between actual quaternion orientation  $[\alpha_{2,k} \ \alpha_{3,k}]^T$  and desired quaternion orientation  $[\alpha_{2d,k-1} \ \alpha_{3d,k-1}]^T$ , a Lyapunov function is chosen as follows

$$V_{2,k} = \frac{1}{2} (\alpha_{2,k} \alpha_{3d,k-1} - \alpha_{2d,k-1} \alpha_{3,k})^2. \quad (5.16)$$

Substituting  $\alpha_{2,k}$  and  $\alpha_{3d,k}$  from Eq. (5.6) and assuming  $T_s \ll 1$ , the Eq. (5.16) becomes

$$V_{2,k} = V_{2,k-1} + T_s u (\alpha_{3,k} \alpha_{3d,k-1} + \alpha_{2d,k-1} \alpha_{2,k-1}) (\alpha_{3d,k-1} \alpha_{2,k-1} - \alpha_{2d,k-1} \alpha_{3,k-1}) r_{k-1}. \quad (5.17)$$

In order to minimize the Lyapunov function  $V_{2,k}$  the following condition should satisfy

$$V_{2,k} - V_{2,k-1} \leq 0. \quad (5.18)$$

Therefore referring to (5.17), the following inequality condition should be satisfied i.e.

$$(\alpha_{3,k} \alpha_{3d,k-1} + \alpha_{2d,k-1} \alpha_{2,k-1}) (\alpha_{3d,k-1} \alpha_{2,k-1} - \alpha_{2d,k-1} \alpha_{3,k-1}) \leq 0. \quad (5.19)$$

From (5.19), a suitable yaw velocity  $r_{k-1}$  which satisfies the above inequality is ex-



pressed as

$$r_{k-1} = -K_\delta \tanh((\alpha_{3,k}\alpha_{3d,k-1} + \alpha_{2d,k-1}\alpha_{2,k-1})(\alpha_{3d,k-1}\alpha_{2,k-1} - \alpha_{2d,k-1}\alpha_{3,k-1})r_{k-1}). \quad (5.20)$$

The control input  $r_{k-1}$  for AUV kinematics (5.6c) and (5.6d) is obtained from (5.20). Henceforth, this term is referred as the desired yaw velocity i.e.  $r_{d,k}$  for designing a controller for AUV heading dynamics.

Prior to design an explicit controller for AUV heading motion, it is assumed that the parameters of the NARMAX structure reached its steady state. Therefore, the identified NARMAX structure for AUV heading motion can be expressed as,

$$\begin{aligned} v_k &= a_0 v_{k-1} + a_1 r_{k-1} + a_2 v_{k-1}^2 + a_3 r_{k-1}^2 + a_4 v_{k-1} r_{k-1} \\ r_k &= b_0 r_{k-1} + b_1 v_{k-1} + b_2 r_{k-1}^2 + b_3 v_{k-1}^2 + b_4 v_{k-1} r_{k-1} + b_5 \delta_{r,k-1}, \end{aligned} \quad (5.21)$$

where  $a_i$  and  $b_i$  are the identified parameters obtained using RELS algorithm as discussed in chapter 3 and chapter 4. The identified heading dynamics (5.21) can also be expressed as

$$\begin{aligned} v_k &= a_0 v_{k-1} + a_1 r_{k-1} + \Delta_{v,k-1} \\ r_k &= b_0 r_{k-1} + b_1 v_{k-1} + b_5 \delta_{r,k-1} + \Delta_{r,k-1} \end{aligned} \quad (5.22)$$

where

$$\begin{aligned} \Delta_{v,k-1} &= a_2 v_{k-1}^2 + a_3 r_{k-1}^2 + a_4 v_{k-1} r_{k-1} \\ \Delta_{r,k-1} &= b_2 r_{k-1}^2 + b_3 v_{k-1}^2 + b_4 v_{k-1} r_{k-1} \end{aligned}$$

constituents the nonlinearity of the AUV dynamics. The yaw velocity  $r_k$  and sway velocity  $v_k$  of the AUV are bounded as actuation input  $\delta_r$  is bounded, therefore the term  $\Delta_{v,k}$  and  $\Delta_{r,k}$  are always bounded i.e.

$$\Delta_k = [\Delta_{v,k} \quad \Delta_{r,k} \quad 0]^T \in \mathcal{W} \subset \mathbb{R}^3. \quad (5.23)$$

It is necessary that the actual yaw velocity  $r_k$  should track the desired yaw velocity

$r_{d,k}$  of (5.20). Therefore, the error dynamics of the heading motion can be expressed as

$$\begin{bmatrix} v_k \\ r_{e,k} \\ r_{d,k} \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_1 \\ b_1 & b_0 & 1 - b_0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{k-1} \\ r_{e,k-1} \\ r_{d,k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ b_5 \\ 0 \end{bmatrix} \delta_{r,k-1} + \begin{bmatrix} \Delta_{v,k-1} \\ \Delta_{r,k-1} \\ 0 \end{bmatrix} \quad (5.24)$$

where  $r_{e,k} = r_k - r_{d,k}$  is the error in yaw velocity at  $k^{th}$  instant. Let, a control law for  $\delta_{r,k-1}$  is chosen as follows,

$$\delta_{r,k-1} = K_p r_{e,k-1}, \quad (5.25)$$

where  $K_p$  is a control gain. The modified error dynamics of the heading motion can be represented as

$$\begin{bmatrix} v_k \\ r_{e,k} \\ r_{d,k} \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_1 \\ b_1 & b_0 & 1 - b_0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{k-1} \\ r_{e,k-1} \\ r_{d,k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ b_5 r_{e,k-1} \\ 0 \end{bmatrix} K_p + \begin{bmatrix} \Delta_{v,k-1} \\ \Delta_{r,k-1} \\ 0 \end{bmatrix} \quad (5.26)$$

The expression (5.26) can be expressed in the form of a linear uncertain system i.e.

$$\nu_k = A\nu_{k-1} + B(r_{e,k-1})K_p + \Delta_{k-1}, \quad (5.27)$$

where  $\nu_{k-1} = [v_{k-1} \ r_{e,k-1} \ r_{d,k-1}]^T \in \mathbb{R}^3$  and  $K_p \in \mathbb{R}$  are the states and the control input respectively. The vector  $\Delta_{k-1}$  is considered as an additive bounded disturbances as shown in (5.23). Additional constraints applied to the states and control inputs are given by

$$\nu_k \in \mathbb{X}, K_p \in \mathbb{U}. \quad (5.28)$$

A MPC strategy is used to generate an optimal control input for worst-case disturbance while satisfying the constraints (5.23) and (5.28). Let, an objective function for worst-case disturbance is defined as follows

$$J_0(\nu(0), U_0) = \max_{\Delta} \|P\nu_N\|_{\infty} + \Sigma \|Q\nu_k\|_{\infty} + \|R\delta_{r,k}\|_{\infty} \quad (5.29)$$

The minimization of this objective function (5.29) considering the constraints is ex-

pressed as

$$\begin{aligned}
J^*(x(0)) &= \min_{\pi_0(), \dots, \pi_{N-1}()} J(\nu(0), U_0) \\
\text{subj. to } &\nu_{k+1} = A\nu_k + B(r_{e,k-1})K_p + \Delta_k, \\
&\nu_k \in \mathcal{X}, K_p \in \mathcal{U}, \\
&K_p = \pi_k(x_k) \\
&\nu_N \in \mathcal{X}_f \\
&\nu_0 = \nu(0), \\
&\forall \Delta_k \in \mathcal{W}^a, k = 0, \dots, N-1
\end{aligned} \tag{5.30}$$

The above minimization problem (5.30) is also termed as the constrained robust optimal control problem over a closed-loop prediction (CROC-CL) [70]. In order to solve a CROC-CL problem with horizon length  $N$ , the following Theorem 5.1 is referred, which is stated below

**Theorem 5.1.** [70] *There exists a state feedback control law  $u_k^* = f_k(x(k))$ ,  $f_k : \mathcal{X}_k \subseteq \mathbb{R}^n \rightarrow \mathcal{U} \subseteq \mathbb{R}^m$ , solution of the CROC-CL (5.30) and  $k = 0, \dots, N-1$  which is time-varying, continuous and piecewise affine on polyhedra*

$$f_k(x) = F_k^i x + g_k^i \quad \text{if } x \in CR_k^i, \quad i = 1, \dots, N_k^r \tag{5.31}$$

where the polyhedron sets  $CR_k^i = \{x \in \mathbb{R}^n : H_k^i x \leq K_k^i\}$ ,  $i = 1, \dots, N_k^r$  are the partition of the feasible polyhedron  $\mathcal{X}_k$ . Moreover  $f_i$ ,  $i = 0, \dots, N-1$  can be found by solving  $N$  multi-parametric linear programming.

A detailed description for the solution of a multi-parametric programming is discussed in Appendix B. Once the CROC-CL problem is solved, then the solution of the MPC is obtained in the form of explicit piece-wise affine function. The control input for the AUV heading motion using explicit-MPC problem can be expressed as

$$K_p = u_k^* = F_k^i x + g_k^i, \quad i = 1, \dots, N_k^r \tag{5.32}$$

## 5.3 Results and Discussion

In this section the proposed explicit controller for the AUV heading motion is verified in both simulation as well as in experimental environment. In simulation environment the control algorithm is verified using the parameters of a known AUV. Later, in order to verify the algorithm for practical implementation, the algorithm is verified in the experimental environment. The results and discussion of these tests are presented in the subsequent sections.

### 5.3.1 Simulation Results

From simulation tests, heading dynamics of the INFANTE AUV after which the parameters are not changing is given as,

$$\begin{bmatrix} v_k \\ r_{e,k} \\ r_{d,k} \end{bmatrix} = \begin{bmatrix} 0.97 & 0.06 & 0.06 \\ -0.08 & 0.8 & -0.2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{k-1} \\ r_{e,k-1} \\ r_{d,k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.04r_{e,k-1} \\ 0 \end{bmatrix} K_p + \begin{bmatrix} \Delta_{v,k-1} \\ \Delta_{r,k-1} \\ 0 \end{bmatrix} \quad (5.33)$$

The weight matrix  $Q = \text{diag}(1, 10, 1)$  and  $R = 0.5$  are considered for the design of an explicit MPC with finite horizon length  $N = 4$ . While designing the kinematics controller, a positive constant  $K_\delta = 0.2$  is considered. To verify the efficacy of the proposed explicit control algorithm the initial states of the AUV is considered as  $[x_k \ y_k \ \alpha_{2,k} \ \alpha_{3,k} \ v \ r]^T = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$  with sampling time  $T_s = 0.5$  sec. The heading reference is given as

$$\psi_d = \begin{cases} 0, & 0 \leq t < 50 \\ 1.57, & 50 \leq t < 200 \\ 0, & 200 \leq t < 350 \\ -1.57, & 350 \leq t < 500 \\ 0, & 500 \leq t < 650 \\ 1.57, & 650 \leq t < 800 \end{cases} \quad (5.34)$$

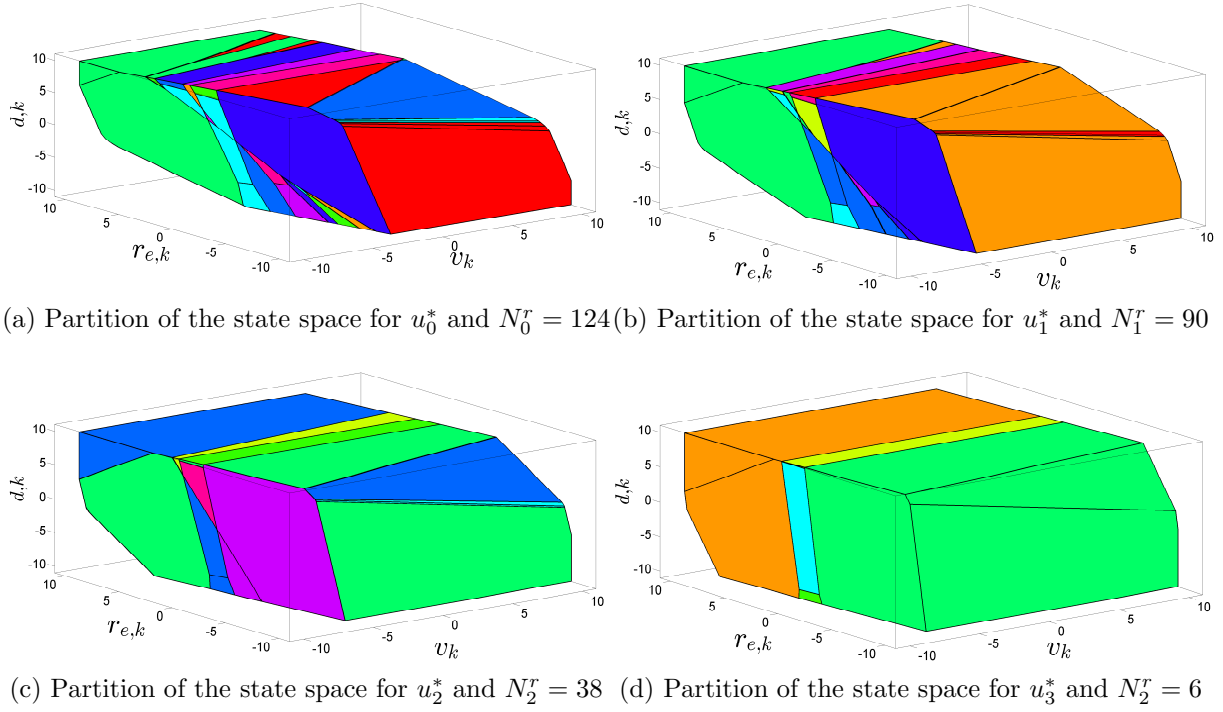


Figure 5.4: Solution of explicit MPC for equation (5.33) with horizon  $N = 4$

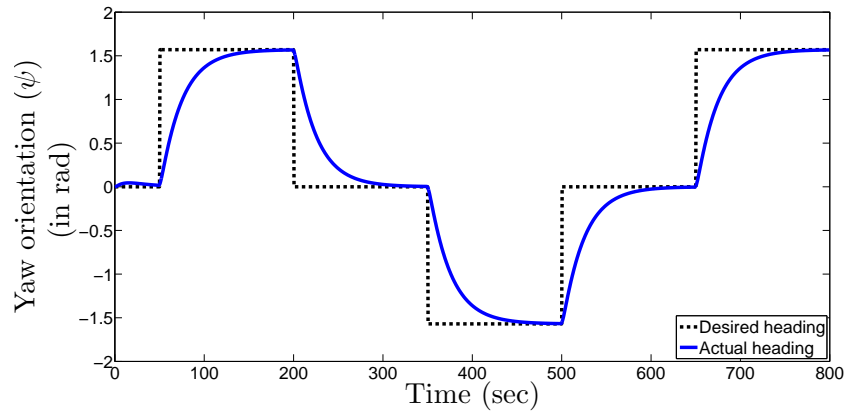


Figure 5.5: Tracking of desired heading by the developed AUV

It is required that the actual heading of the AUV should track the desired heading (5.34). Fig.5.5 shows that the AUV successfully tracks the desired heading and the corresponding heading error is shown in Fig.5.6. From the results Fig.5.5 and Fig.5.6, it

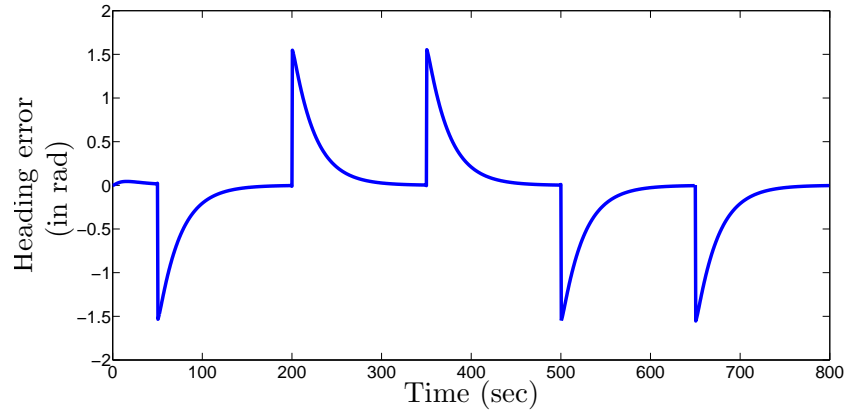


Figure 5.6: Heading error while tracking LoS path

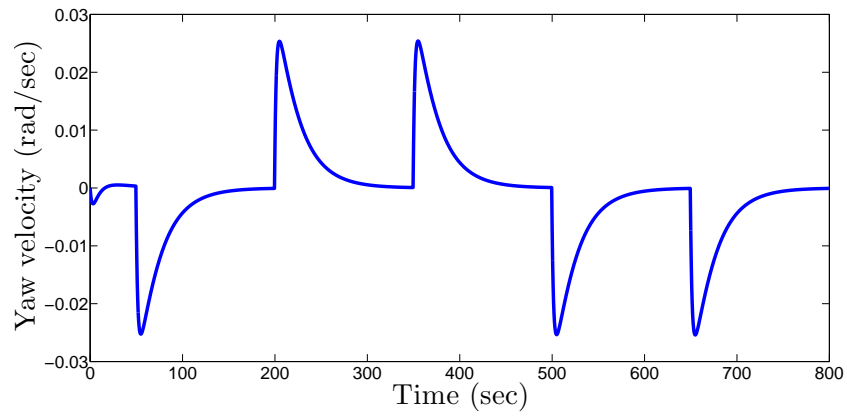


Figure 5.7: Yaw velocity

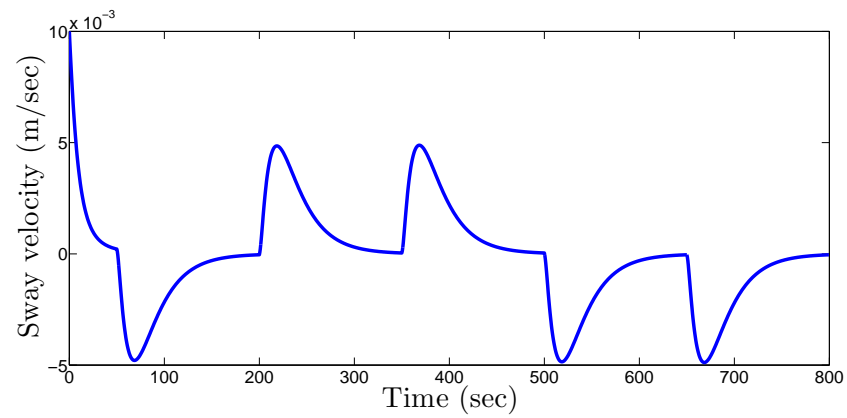


Figure 5.8: Sway velocity

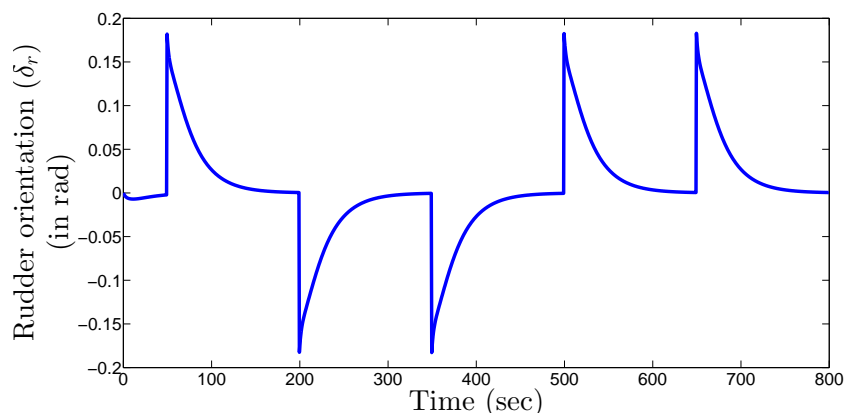


Figure 5.9: Control signal for rudder plane

Table 5.1: Description of ROS nodes

Node	Description
/xsens_driver	Driver node to access Xsens MTI INS sensor
/navquest_node	Driver node to access Navquest DVL sensor
/odom_trans	Generates AUV states
/desired_path_mppq	desired path node (5.14) & (5.15)
/mppq_mqp	controller node (5.31)
/rosterial_server	Arduino node for transmitting actuation signal

is observed that during the tracking of the heading reference, the heading error reduces to zero. Further, while tracking the desired heading the variation of yaw velocity and sway velocity are shown in Fig.5.7 and Fig.5.8. The actuation signal generated by the controller is given in Fig.5.9. From the simulation results, it is verified that the proposed explicit MPC algorithm is effective for AUV heading and further the practical realization of the algorithm is to be verified in the subsequent section.

### 5.3.2 Experimental Results

In this section, experimental results obtained through implementing the proposed control algorithm on the prototype AUV are presented. As discussed in chapter 2, this control algorithm is implemented in the ROS environment which is shown in Fig.5.10. The description of nodes and messages are presented in Table 5.1 and Table 5.2. From experimental trials, heading dynamics for the developed AUV is given as

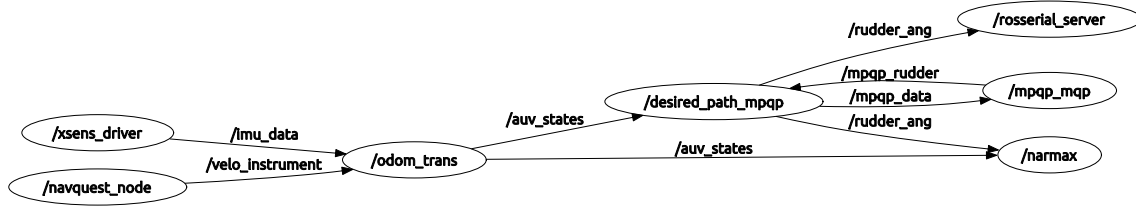


Figure 5.10: Implementation of explicit MPC in ROS

Table 5.2: Description of ROS messages and its characteristics

Message	Description	Bandwidth	Publishing rate
/imu_data	$\phi, \theta, \psi, p, q, r$	32.8 Kb/sec	100
/auv_states	$z, \phi, \theta, \psi, u, v, w, p, q, r$	523 B/sec	10
/mpqp_rudder	$K_{p1}$	525 B/sec	10
/mpqp_data	$v_k, r_{e,k}, r_{d,k}$	522 B/sec	10
/rudder_ang	$\delta_r$	20 B/sec	10
/velo_instrument	$u, v, w$	20 B/sec	10

$$\begin{bmatrix} v_k \\ r_{e,k} \\ r_{d,k} \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_1 \\ b_1 & b_0 & 1 - b_0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{k-1} \\ r_{e,k-1} \\ r_{d,k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ b_5 r_{e,k-1} \\ 0 \end{bmatrix} K_p + \begin{bmatrix} \Delta_{v,k-1} \\ \Delta_{r,k-1} \\ 0 \end{bmatrix} \quad (5.35)$$

and the considered constraint parameters are  $|r_{d,k}| \leq 1$  and  $|K_p| \leq 1$ . The initial states of the AUV is considered as  $[x_k \ y_k \ \alpha_{2,k} \ \alpha_{3,k} \ v \ r]^T = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$  with sampling time  $T_s = 0.1$  sec and the implementation of the developed explicit MPC control algorithm is shown in Fig.5.12. For experimentation the desired heading is taken as

$$\psi_d = \begin{cases} 0, & 0 \leq t < 5 \\ 1.57, & 5 \leq t < 20 \\ 0, & 20 \leq t < 35 \\ -1.57, & 35 \leq t < 50 \\ 0, & 50 \leq t < 65 \\ 1.57, & 65 \leq t < 80 \end{cases} \quad (5.36)$$



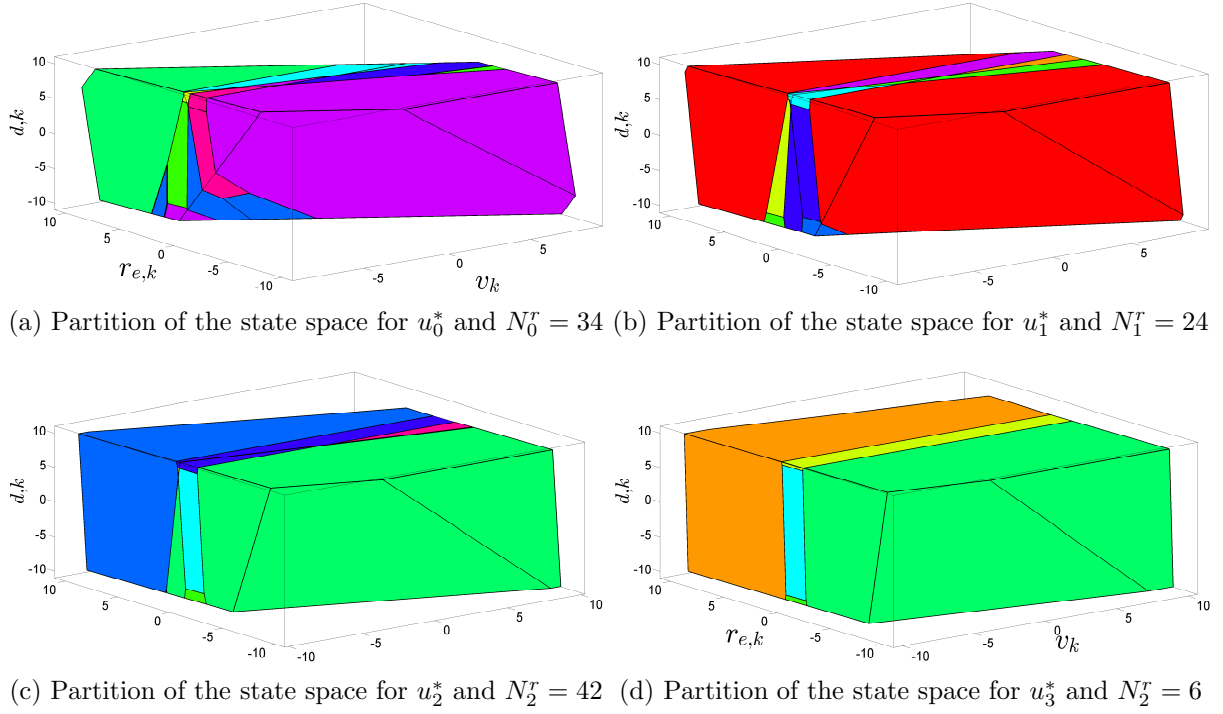
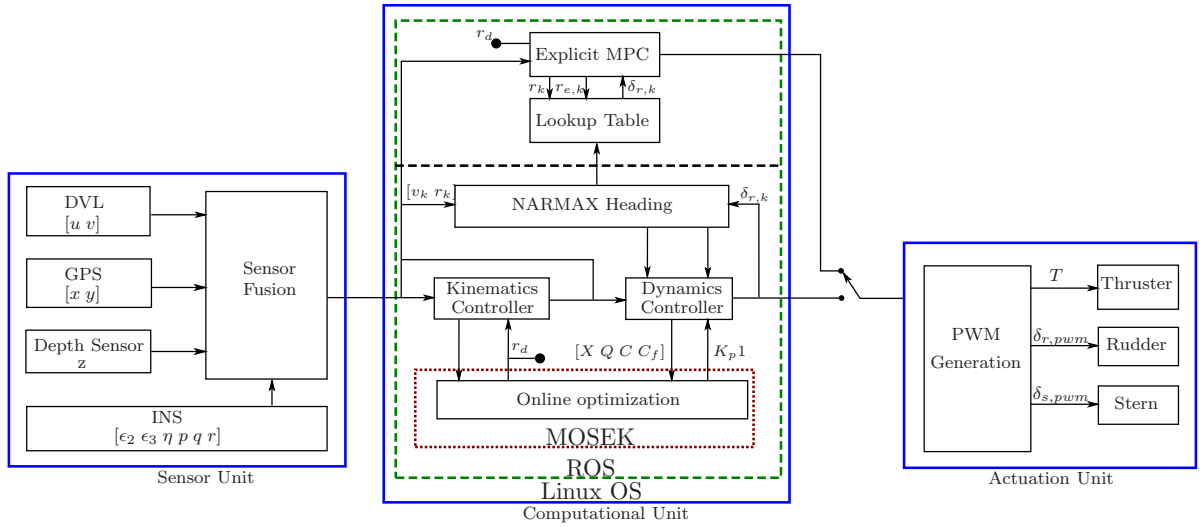
Figure 5.11: Solution of explicit MPC for equation (5.35) with horizon  $N = 4$ 

Figure 5.12: Implementation of the explicit MPC control algorithm

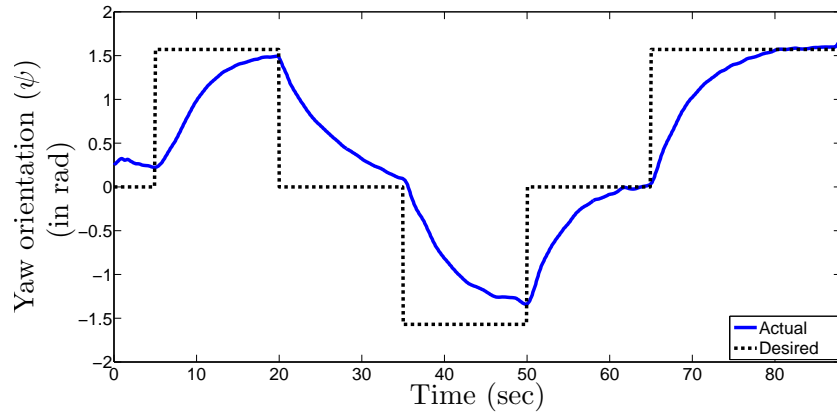


Figure 5.13: Following a desired yaw orientation

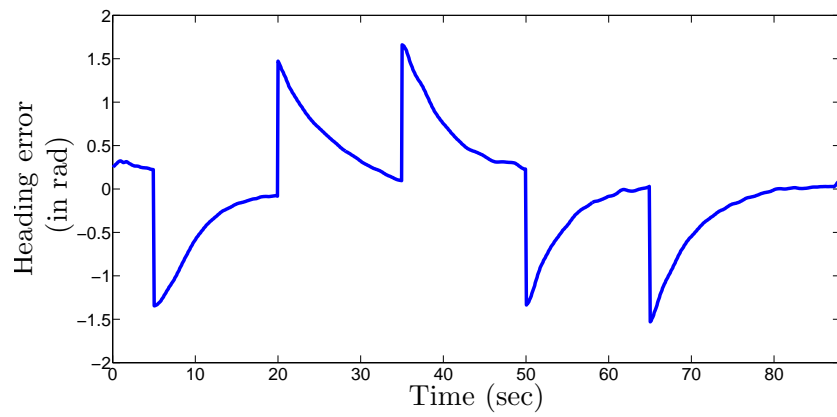


Figure 5.14: Orientation error along yaw motion

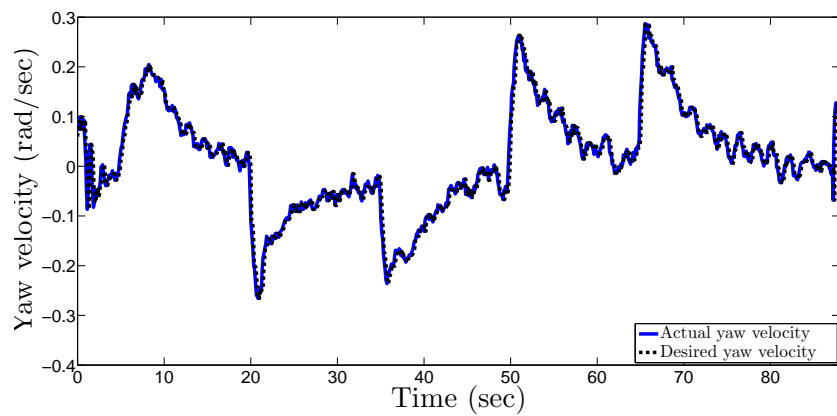


Figure 5.15: Yaw velocity while tracking the desire path

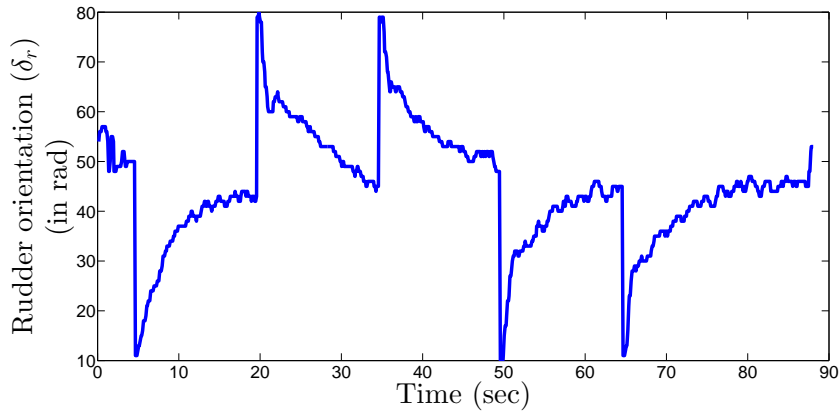


Figure 5.16: Rudder input required to steer the AUV along LOS path

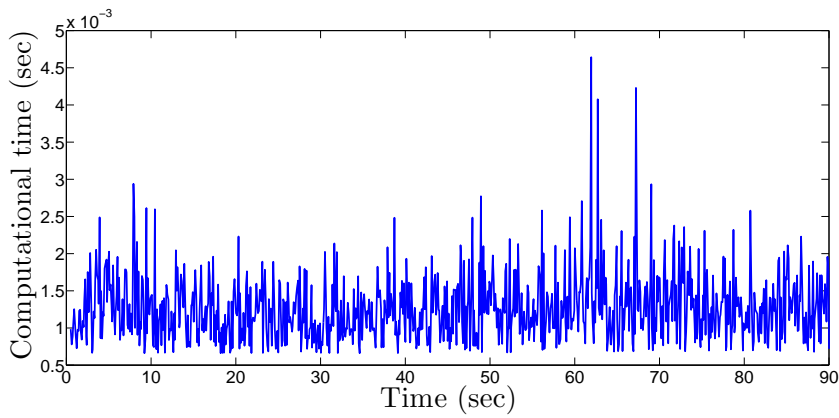


Figure 5.17: Time taken to generate the actuation signal

From the Fig.5.13, it is evident that the explicit MPC steers the AUV along the desired path and the error corresponding to the heading is shown in Fig. 5.14. While following the desired path the variation of yaw velocity is also observed which is shown in Fig. 5.15. The control signal generated by this explicit model predictive control algorithm to drive the AUV for heading tracking is presented in Fig. 5.16. In view of practical realization, the average computational time taken by the control algorithm is 1.25msec and is shown in Fig.5.17. Finally, a comparison is provided in Table 5.3 to show the effectiveness of the developed algorithm in chapter 3, chapter 4 and chapter 5.

Developed controller	Computational time	Remarks
Self-tuning adaptive controller (chapter-3)	0.011sec	Although the developed controller successfully tracks the reference path, the state and actuator constraints are not been addressed.
Constrained self-tuning adaptive controller (chapter-4)	0.08sec	The computational time is more as compared to the previous controller but it considers both the state and actuator constraints.
Explicit model predictive controller (chapter-5)	0.005sec	The developed controller considers both the state and actuator constraints and the computational time is also less. However, the online computation of the look-up table is required which further increases the computational complexity.

Table 5.3: Comparison between various developed control algorithms

## 5.4 Chapter Summary

In this chapter, the design of an explicit MPC control strategy is proposed for implementing a Line-of-Sight guidance law for an Autonomous Underwater Vehicle. The control law is derived by adopting a Lyapunov based backstepping controller for AUV kinematics. This kinematics controller provides a desired yaw velocity which is to be followed by the AUV dynamics. Further, using this desired yaw velocity an explicit model predictive controller is designed using the identified NARMAX structure of the AUV heading motion. The efficacy of the proposed control algorithm is first verified through simulation using MATLAB. Then, the proposed controller is implemented in an AUV developed in the laboratory for experimental verification.

# Chapter 6

## Conclusion and Suggestion for Future Work

### 6.1 Overall Summary of the thesis

This thesis addressed the Line-of-Sight control problem of an Autonomous Underwater Vehicle. As discussed in the chapter 1, the controllers for both heading motion and diving motion for an AUV have been developed and the effectiveness of the proposed control algorithm is verified through both simulation and experimentation.

In chapter 2, the design and development of a prototype Autonomous Underwater Vehicle in the laboratory are presented. It describes the design of nose, hull and tail profiles of the AUV. Further, the hardware configuration of AUV involving appropriate sensors, actuators, communication and computational components to achieve autonomous capability is also presented. The software framework needed for interfacing of various sensors and actuators for controller implementation in the prototype AUV is presented. Further, this prototype AUV is used for verifying the effectiveness of the proposed control algorithms experimentally.

Firstly, a self-tuning PID algorithm is proposed in chapter 3 which involves two steps. Firstly, a kinematic controller is designed which provides the reference for the dynamic controller. The kinematic controller is designed using backstepping controller and stability is proved using the Lyapunov theory. For the design of dynamic controller, the dynamic model of the AUV is identified using NARMAX model and

the parameters of the AUV are identified using RELS method. An inverse optimal controller is developed to design a Self-Tuning adaptive PID controller. This PID controller generates the control signal for the AUV, to achieve path following task e.g. rectilinear path connecting different waypoints. However, the controller suffers from the actuator saturation.

In chapter 4, a constrained self-tuning control (CSTC) algorithm is developed for an AUV to implement LoS guidance. To capture the unknown dynamics, a minimum representation of a NARMAX (MR-NARMAX) model structure of the AUV is identified which consists of significant regressors terms. The parameters of the MR-NARMAX model are identified using RELS algorithm at each sampling time instant. Further, a constrained self-tuning controller is developed to implement LoS guidance algorithm. The generated actuation signal complies with the actuator constraint for practical feasibility of implementing the algorithm. The heading tracking of the MR-NARMAX model with CSTC algorithm is successfully verified on the prototype AUV developed in the laboratory.

The CSTC algorithm in chapter 4, successfully tracks the Line-of-Sight path by considering the state and actuator constraints. But, this control algorithm implementation takes more computational time. The maximum computational time taken to generate the control signal is 0.8 sec. In view of reducing the computational time, in chapter 5, an explicit Model Predictive Control (MPC) strategy is developed for implementing the LoS guidance by the AUV. In this CSTC strategy, using the AUV kinematics, a Lyapunov based backstepping controller is designed to generate desired yaw velocity. This, explicit model predictive controller is designed using the identified NARMAX model as described in chapter 4 for tracking the desired yaw velocity. The efficacy of the proposed explicit MPC strategy is verified through both in simulation as well as in experimentation on a prototype AUV and tested in the institute swimming pool.

### 6.1.1 Contributions of the Thesis

- A prototype AUV with appropriate hardware configuration is developed in the laboratory for practical implementation of the proposed control algorithms.

- Unlike most of the previous work on control design based on known AUV dynamics. In view of capturing the unknown AUV dynamics, a nonlinear system identification i.e. polynomial based NARMAX model structure is used.
- An Inverse self-tuning PID control law for achieving LoS guidance is developed and experimentally verified.
  - R. Rout and B. Subudhi, “Inverse optimal self-tuning PID control design for an Autonomous Underwater Vehicle”, *International Journal of System Science*, vol. 48, no.2, pp.367-375, 2016 (Taylor & Francis).
- In order to resolve the state and actuator constraints, a new constrained adaptive control algorithm for LoS guidance has been developed in which a MR-NARMAX model is used for identifying the AUV dynamics.
  - R. Rout and B. Subudhi, “NARMAX Self-Tuning Controller for Line-of-Sight based Way-Point Tracking for an Autonomous Underwater Vehicle”, *IEEE Transaction on Control Systems Technology*, DOI: 10.1109/TCST.2016.2613969.
- In view of reducing computational burden of the control algorithm discussed in chapter 3 and chapter 4, an explicit model predictive control algorithm using AUV heading dynamics is developed for the LoS guidance algorithm.
  - R. Rout and B. Subudhi, “Design of a Line-of-Sight guidance law based on Explicit model predictive control design for an Autonomous Underwater Vehicle”, *IEEE Transaction on Control Systems Technology* (under preparation)

## 6.2 Suggestions for the future work

### ***Explicit MPC for heading and diving motion:***

The explicit MPC is designed for AUV heading motion is described in chapter 5 was found to be very efficient in achieving heading control with reduced computational overhead. Thus, this design can be extended for controlling both heading and diving motion of an AUV

***Roll motion in control design:***

In this work, the controllers are designed assuming no coupling between the heading and the diving motion exist. However, during complex maneuvering of the AUV this assumption may not be true. Therefore, this work can be extended by designing a new controller by including the effect of roll motion.

***Extension of LoS guidance:***

The developed algorithms are based on Line-of Sight guidance law. Further, it can be extended for the implementation of complex paths which will be the combination of LoS paths.



# Appendix A

## Kinematics and Dynamics of an AUV

### A.1 Kinematics

The kinematic expression of an AUV can be fully represented by using two coordinate frames i.e. earth-fixed frame  $\{I\}$  and body-fixed frame  $\{B\}$  as shown in Fig.A.1. The position parameters  $\eta_1 = [x, y, z, \phi, \theta, \psi]^T$  are obtained in  $\{I\}$ , whereas the velocity parameters  $\nu = [u, v, w, p, q, r]^T$  are obtained with reference to  $\{B\}$ . In order to observe the motion of the AUV from  $\{I\}$ , a transformation matrix  $J(\eta_2) = \text{diag}(J_1(\eta_2), J_2(\eta_2))$

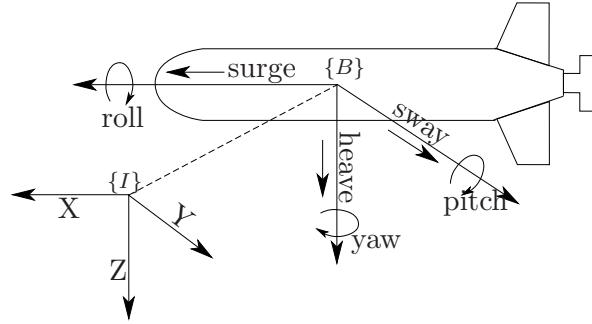


Figure A.1: Frames to represent AUV motion

from  $\{B\}$  to  $\{I\}$  is defined as follows,

$$J_1(\eta_2) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$J_2(\eta_2) = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix}$$

Using these transformation matrix  $J(\eta_2)$ , the expression for velocities in  $\{I\}$  is given by,

$$\begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} J_1(\eta_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & J_2(\eta_2) \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} \quad (\text{A.1})$$

where,  $\dot{\eta}_1 = [\dot{x}, \dot{y}, \dot{z}]^T$  and  $\dot{\eta}_2 = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$  represents the AUV velocities in the earth-fixed frame. The corresponding body-fixed velocities of the AUV are  $\nu_1 = [u, v, w]$  and  $\nu_2 = [p, q, r]$ . Referring to [64], the kinematic equation (A.1) in terms of quaternion orientation  $\alpha = [\alpha_0, \alpha_1, \alpha_2, \alpha_3]^T$  can be expressed as,

$$\begin{bmatrix} \dot{\eta}_1 \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} E_1(\alpha) & 0_{3 \times 3} \\ 0_{4 \times 3} & E_2(\alpha) \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix}, \quad (\text{A.2})$$

where

$$E_1(\alpha) = \begin{bmatrix} 1 - 2(\alpha_1^2 + \alpha_2^2) & 2(\alpha_0\alpha_1 - \alpha_2\alpha_3) & 2(\alpha_0\alpha_1 + \alpha_1\alpha_3) \\ 2(\alpha_0\alpha_1 + \alpha_2\alpha_3) & 1 - 2(\alpha_0^2 + \alpha_2^2) & 2(\alpha_1\alpha_2 - \alpha_0\alpha_3) \\ 2(\alpha_0\alpha_2 - \alpha_1\alpha_3) & 2(\alpha_1\alpha_2 + \alpha_0\alpha_3) & 1 - 2(\alpha_0^2 + \alpha_1^2) \end{bmatrix}$$

$$E_2(\alpha) = \frac{1}{2} \begin{bmatrix} \alpha_3 & -\alpha_2 & \alpha_1 \\ \alpha_2 & \alpha_3 & -\alpha_0 \\ -\alpha_1 & \alpha_1 & \alpha_3 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 \end{bmatrix}$$

## A.2 Dynamics

Dynamics of the AUV consists of nonlinearity and coupling between various terms. The AUV has 6DOF equation of motion along x,y and z axis, referring to [64] the following are the dynamic equation along its respective axis.

- Surge Motion:

$$m [\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})] = X \quad (\text{A.3})$$

- Sway Motion

$$m [\dot{v} - wp + ur - y_g(p^2 + r^2) + z_g(qr - \dot{p}) + x_g(pq + \dot{r})] = Y \quad (\text{A.4})$$

- Heave Motion

$$m [\dot{w} - uq + vp - z_g(q^2 + p^2) + x_g(pr - \dot{q}) + y_g(qr + \dot{p})] = Z \quad (\text{A.5})$$

- Roll Motion

$$\begin{aligned} I_x \dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} + \\ m [y_g(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)] = K \end{aligned} \quad (\text{A.6})$$

- Pitch Motion

$$\begin{aligned} I_y \dot{q} + (I_x - I_z)pr - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} + \\ m [z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] = M \end{aligned} \quad (\text{A.7})$$

- Yaw Motion

$$\begin{aligned} I_z \dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} + \\ m [x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)] = N \end{aligned} \quad (\text{A.8})$$

Table A.1: AUV parameter definition

Hydrostatic Force	$X_{HS}, Y_{HS}, Z_{HS}, K_{HS}, M_{HS}, N_{HS}$
Added Mass	$X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{w}}, Y_{\dot{r}}, Z_{\dot{q}}, M_{\dot{q}}, N_{\dot{v}}, N_{\dot{r}}$
Propeller Thrust	$X_{prop}$
Lift Force	$Z_{uu\delta_s}, Y_{uu\delta_r}, N_{uu\delta_r}$
Drag Force	$M_{ww}, M_{qq}, X_{uu}, Y_{vv}$

The parameter  $X, Y, Z, K, M, N$  are the external forces and moments, which includes Hydrostatic force, drag force, Lift force, Propeller Thrust, Added Mass and also the effect of stern plane and rudder planes. These external parameters are defined as follows,

$$\begin{aligned}
X &= X_{HS} + X_{u|u}|u| + X_{\dot{u}}\dot{u} + X_{wq}wq + X_{qq}qq + X_{vr}vr + X_{rr}rr + X_{prop} \\
Y &= Y_{HS} + Y_{v|v}|v| + Y_{r|r}|r| + Y_{\dot{v}}\dot{v} + Y_{\dot{r}}\dot{r} + Y_{ur}ur + Y_{wp}wp + Y_{pq}pq + Y_{uv}uv + \\
&\quad Y_{uu\delta_r}u^2\delta_r \\
Z &= Z_{HS} + Z_{w|w}|w| + Z_{q|q}|q| + Z_{\dot{w}}\dot{w} + Z_{\dot{q}}\dot{q} + Z_{uq}uq + Z_{vp}vp + Z_{rp}rp + Z_{uw}uw + \\
&\quad Z_{uu\delta_s}u^2\delta_s \\
K &= K_{HS} + K_{p|p}|p| + K_{\dot{p}}\dot{p} + K_{prop} \\
M &= M_{HS} + M_{w|w}|w| + M_{q|q}|q| + M_{\dot{w}}\dot{w} + M_{\dot{q}}\dot{q} + M_{uq}uq + M_{vp}vp + M_{rp}rp + \\
&\quad M_{uw}uw + M_{uu\delta_s}u^2\delta_s \\
N &= N_{HS} + N_{v|v}|v| + N_{r|r}|r| + N_{\dot{v}}\dot{v} + N_{\dot{r}}\dot{r} + N_{ur}ur + N_{wp}wp + N_{pq}pq + \\
&\quad N_{uv}uv + N_{uu\delta_r}u^2\delta_r
\end{aligned} \tag{A.9}$$

the parameter used in (A.9) are defined in table.A.1. These parameters are the external components which affect the overall dynamics of the Autonomous Underwater Vehicle.

# Appendix B

## Solution to Multiparametric Quadratic programming

Minimization of a 2-norm objective function in the presence of state and input constraint can be expressed as

$$\begin{aligned} J^*(x(0)) = & \min_{U_0} \quad J(x(0), U_0) = X^T \bar{Q} X + U_0^T \bar{R} U_0 \\ \text{subj. to } & x_{k+1} = Ax_k + Bu_k, \quad k = 0, \dots, N-1 \\ & x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad k = 0, \dots, N-1 \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(0), \end{aligned} \tag{B.1}$$

where  $X \in \mathbb{R}^N$ ,  $\bar{Q} \succeq 0 \in \mathbb{R}^{N \times N}$ ,  $\bar{R} \succeq 0 \in \mathbb{R}^{N \times N}$ ,  $U_0 \in \mathbb{R}^N$ . In the above equation (B.1),  $N$  is defined as the horizon length whereas  $X$  and  $U_0$  are the predicted states and predicted input. It is also required that the the final state  $x_N$  should lie in the terminal region  $\mathcal{X}_f \subseteq \mathbb{R}^2$ . Representing the predictive state vector  $X$  in terms of initial state  $x(0)$  as

$$X = \bar{A}x(0) + \overline{AB}U_0, \tag{B.2}$$

where

$$X = [x_0, x_1, \dots, x_N]^T,$$

$$\begin{aligned}
U_0 &= [u_0, u_1, \dots, u_{N-1}]^T, \\
\bar{A} &= [I \ A \cdots A^N]^T, \\
\overline{AB} &= \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ B & 0 & \cdots & 0 & 0 \\ AB & B & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & AB & B \end{bmatrix}.
\end{aligned}$$

By substituting  $X$  and  $U_0$  from (B.2), the modified objective function is defined as

$$J(x(0), U_0) = U_0^T H U_0 + 2x^T(0) F U_0 + x^T(0) Y x(0), \quad (\text{B.4})$$

where  $H = B_u^T \bar{Q} B_u + \bar{R}$ ,  $F = \bar{A}_x^T \bar{Q} B_u$ ,  $Y = \bar{A}_x^T \bar{Q} \bar{A}_x$ . Referring to [70], minimizing the modified objective function (B.4) with its constraints defined in (B.1) is equivalent to minimize the following objective function

$$\begin{aligned}
J^*(x(0)) = \min_z \quad & z^T H z \\
\text{subj. to} \quad & G_0 z \leq w_0 + S_0 x(0),
\end{aligned} \quad (\text{B.5})$$

where

$$\begin{aligned}
z &= U_0 + H^{-1} F^T x(0), \\
G_0 &= \begin{bmatrix} A_u & 0 & \cdots & 0 \\ 0 & A_u & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_u \\ 0 & 0 & \cdots & 0 \\ A_x B & 0 & \cdots & 0 \\ A_x A B & A_x B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_f A^{N-1} B & A_f A^{N-2} B & \cdots & A_f B \end{bmatrix}, \\
E_0 &= \begin{bmatrix} 0 & \cdots & 0 & -A_x & -A_x A & \cdots & -A_f A^N \end{bmatrix}^T,
\end{aligned}$$

$$w_0 = \begin{bmatrix} b_u & \cdots & b_u & b_x & b_x & \cdots & b_f \end{bmatrix}^T.$$

The objective function (B.5) can be minimized using multi-parametric quadratic programming (mp-QP) method, which utilizes the solution of Karush-Kuhn-Tucker (KKT) conditions

$$Hz^* + G^T u^* = 0 \quad (\text{B.6a})$$

$$\lambda_i^* (G_i z^* - w_i - S_i x) = 0 \quad (\text{B.6b})$$

$$u^* \geq 0 \quad (\text{B.6c})$$

$$Gz^* - w - Sx \leq 0. \quad (\text{B.6d})$$

Referring to [], the solution of the KKT conditions (B.6a)-(B.6d) is represented as

$$z^* = F_i x + g_i, \quad i = 1, \dots, m, \quad (\text{B.7})$$

where

$$\begin{aligned} F_i &= H^{-1} G_A^T (G_A H^{-1} G_A^T)^T S_a, \\ g_i &= H^{-1} G_A^T (G_A H^{-1} G_A^T)^T w_a. \end{aligned}$$

From (B.8), the optimal predictive control input ( $U_0^*$ ) is expressed as

$$U_0^* = F_i x + g_i - H^{-1} F^T x(0). \quad (\text{B.8})$$

First element of  $U_0^*$  is used as the control input i.e.  $u_{k-1} = U_0^*(1)$ . For the online implementation of the developed controller, the polyhedron region  $G_0 z \leq w_0 + S_0 x(0)$  is stored offline and a search technique [71] can be employed to identify the active region. The corresponding  $F_i$  and  $g_i$  of the active region is then used to derive the control law (B.7).

# References

- [1] S. A. Billings, *Nonlinear System Identification*. John Wiley & Sons, 2013.
- [2] R. Wernli, “AUV Commercialization-Who’s Leading the Pack?” in *Proc. MTS/IEEE OCEANS Conf.*, Providence, RI, 11-14 Sept. 2000, pp. 391–395.
- [3] R. L. Wernli, “AUVs-A Technology Whose Time Has Come,” in *Proc. International Symposium on Underwater Technology*, Tokyo, Japan, 19-19 April 2002, pp. 309–314.
- [4] P. E. Hagen, N. Storkersen, K. Vestgard, and P. Kartvedt, “The HUGIN 1000 Autonomous Underwater Vehicle for Military Applications,” in *Proc. MTS/IEEE OCEANS conf.*, San Diego, CA, USA, 22-26 Sept. 2003, pp. 1141–1145.
- [5] R. Panish, “Dynamic Control Capabilities and Developments of the Bluefin Robotics AUV Fleet,” in *Proc. 16<sup>th</sup> Int. Symp. Unmanned Untethered Submersible Technology*, Durham, NH, USA, 23-26 Aug. 2009, pp. 23–26.
- [6] S. K. Das, D. Pal, S. Nandy, V. Kumar, S. N. Shome, and B. Mahanti, *Trends in Intelligent Robotics*. Springer, 2010, vol. 103, ch. Control Architecture for AUV-150: A Systems Approach, pp. 41–48.
- [7] T. Copros and D. Scourzic, *Global Change: Mankind-Marine Environment Interactions*. Springer, 2011, ch. Alister-Rapid Environment Assessment AUV (Autonomous Underwater Vehicle), pp. 233–238.
- [8] J. Kalwa, “The RACUN-project: Robust acoustic communications in underwater networks-an overview,” in *Proc. MTS/IEEE OCEANS conf.*, Santander, Spain, 6-9 June 2011, pp. 1–6.
- [9] T. I. Fossen, “Nonlinear modelling and control of underwater vehicles,” Ph.D. dissertation, Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), June 1991.
- [10] G. Antonelli, S. Chiaverini, N. Sarkar, and M. West, “Adaptive control of an autonomous underwater vehicle: experimental results on odin,” *IEEE Transactions on Control Systems Technology*, vol. 9, no. 5, pp. 756–765, 2001.
- [11] B. K. Sahu and B. Subudhi, “Adaptive tracking control of an autonomous underwater vehicle,” *International Journal of Automation and Computing*, vol. 11, no. 3, pp. 299–307, 2014.



- [12] F. Y. Bi, Y. J. Wei, J. Z. Zhang, and W. Cao, "Position-tracking control of underactuated autonomous underwater vehicles in the presence of unknown ocean currents," *IET Control Theory Applications*, vol. 4, no. 11, pp. 2369–2380, 2010.
- [13] F. Repoulas and E. Papadopoulos, "Planar trajectory planning and tracking control design for underactuated AUVs," *Ocean Engineering*, vol. 34, no. 11-12, pp. 1650–1667, Aug. 2007.
- [14] L. Lapierre, D. Soetanto, and A. Pascoal, "Nonlinear path following with applications to the control of autonomous underwater vehicles," in *Proc. 42nd IEEE conference on Decision and Control*, Maui, HI, 9-12 Dec 2003, pp. 1256–1261.
- [15] L. Lapierre and D. Soetanto, "Nonlinear path-following control of an AUV," *Ocean Engineering*, vol. 34, no. 11-12, pp. 1734–1744, 2007.
- [16] J. Ghommam, F. Mnif, A. Benali, and N. Derbel, "Nonsingular Serret-Frenet Based Path Following Control for an Underactuated Surface Vessel," *Journal of Dynamic Systems, Measurement, and Control*, vol. 131, no. 2, pp. 1–8, 2009.
- [17] R. Rout, B. Subudhi, and S. Ghosh, "Adaptive Path Following Control of an Autonomous Underwater Vehicle," in *Proceedings of Advances in Control and Optimization of Dynamic Systems*, Bangalore, Feb. 16-18,, 2012, pp. 1–7.
- [18] E. Fredriksen and K. Pettersen, "Global k-exponential way-point maneuvering of ships: Theory and experiments," *Automatica*, vol. 42, no. 4, pp. 677–687, 2006.
- [19] T. Fossen, K. Y. Pettersen, R. Galeazzi *et al.*, "Line-of-sight path following for Dubins paths with adaptive sideslip compensation of drift forces," *IEEE Transactions on Control System Technology*, vol. 23, no. 2, pp. 820–827, 2015.
- [20] R. N. Smith, Y. Chao, P. P. Li, D. A. Caron, B. H. Jones, and G. S. Sukhatme, "Planning and Implementing Trajectories for Autonomous Underwater Vehicles to Track Evolving Ocean Processes Based on Predictions from a Regional Ocean Model," *International Journal of Robotics Research*, vol. 29, no. 12, pp. 1475–1497, 2010.
- [21] A. P. Aguiar and J. P. Hespanha, "Trajectory-Tracking and Path-Following of Underactuated Autonomous Vehicles With Parametric Modeling Uncertainty," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1362–1379, 2007.
- [22] A. Sanyal, N. Nordkvist, and M. Chyba, "An Almost Global Tracking Control Scheme for Maneuverable Autonomous Vehicles and its Discretization," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 457–462, 2011.
- [23] B. Subudhi, K. Mukherjee, and S. Ghosh, "A static output feedback control design for path following of autonomous underwater vehicle in vertical plane," *Ocean Engineering*, vol. 63, pp. 72–76, 2013.
- [24] T. Fossen, M. Breivik, and R. Skjetne, "Line-of-Sight Path Following of Underactuated Marine Craft," in *Proc. IFAC conf. on Manoeuvring and Control of Marine Craft*, Girona, Spain, 17-19 Sept. 2003, pp. 244–249.

- [25] G. D. Watt, "Estimates for the Added Mass of a Multi-Component, Deeply Submerged Vehicle. Part 1. Theory and Program Description," Defence Research Establishment, Tech. Rep., 1988.
- [26] J. H. Milgram, "Strip Theory for Underwater Vehicles in water of Finite Depth," *Journal of Engineering Mathematics*, vol. 58, no. 1-4, pp. 31–50, 2007.
- [27] D. F. Myring, "A theoretical study of body drag in sub-critical axisymmetric flow," *Aeronautical Quarterly*, vol. 27, no. 3, pp. 186–194, 1976.
- [28] E. A. de Barros, J. L. D. Dantas, A. M. Pascoal, and E. de Sa, "Investigation of Normal Force and Moment Coefficients for an AUV at Nonlinear Angle of Attack and Sideslip Range," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 538–549, 2008.
- [29] S. Tang, T. Ura, T. Nakatani, B. Thornton, and T. Jiang, "Estimation of Hydrodynamic Coefficients of the Complex-Shaped Autonomous Underwater Vehicle TUNA-SAND," *Journal of Marine Science and Technology*, vol. 14, pp. 373–386, 2009.
- [30] T. Prestero, "Verification of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle," Master's thesis, Woods Hole Oceanographic Institution, Departments of Ocean and Mechanical Engineering, MIT, MA, 2001.
- [31] P. Ridao, A. Tiano, A. El-Fakdi, M. Carreras, and A. Zirilli, "On the identification of non-linear models of unmanned underwater vehicles," *Control engineering practice*, vol. 12, no. 12, pp. 1483–1499, 2004.
- [32] P. W. J. V. de Ven, C. Flanagan, and D. Toal, "Neural Network control of Underwater Vehicles," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 5, pp. 533–547, 2005.
- [33] K. P. Venugopal, R. Sudhakar, and a. S. Pandya, "On-line Learning Control of Autonomous Underwater Vehicles using Feedforward Neural Networks," *IEEE Journal of Oceanic Engineering*, vol. 17, no. 4, pp. 308–319, 1992.
- [34] J.-S. Wang and C. S. G. Lee, "Self-Adaptive Recurrent Neuro-Fuzzy Control of an Autonomous Underwater Vehicle," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 283–295, 2003.
- [35] T. Fujii and T. Ura, "Development of motion control system for AUV using neural nets," in *Sympo. on Autonomous Underwater Vehicle Tech.*, Tokyo, Japan, 5-6 June 1990, pp. 81–86.
- [36] K. Ishii, T. Fujii, and T. Ura, "An On-line Adaptation Method in a Neural Network Based Control System for AUVs," *IEEE Journal of Oceanic Engineering*, vol. 20, no. 3, pp. 221–228, 1995.
- [37] J. H. Li, P. M. Lee, and S. J. Lee, "Neural Net based Nonlinear Adaptive Aontrol for Autonomous Underwater Vehicles," in *Proc. IEEE International Conference on Robotics & Automation*, vol. 2, 11-15 May 2002, pp. 1075–1080.

- [38] J. H. Li and P. M. Lee, "A neural network adaptive controller design for free-pitch-angle diving behavior of an autonomous underwater vehicle," *Robotics and Autonomous Systems*, vol. 52, no. 2-3, pp. 132–147, 2005.
- [39] P. W. Van de Ven, T. A. Johansen, A. J. Sørensen, C. Flanagan, and D. Toal, "Neural network augmented identification of underwater vehicle models," *Control Engineering Practice*, vol. 15, no. 6, pp. 715–725, 2007.
- [40] K. Ishii and T. Ura, "An adaptive neural-net controller system for an underwater vehicle," *Control Engineering Practice*, vol. 8, no. 2, pp. 177–184, Feb. 2000.
- [41] K. Shojaei and M. Arefi, "On the neuro-adaptive feedback linearising control of underactuated autonomous underwater vehicles in three-dimensional space," *IET Control Theory and Applications*, vol. 9, no. 8, pp. 1264–1273, 2015.
- [42] F. Ali, E. Kim, and Y.-G. Kim, "Type-2 fuzzy ontology-based semantic knowledge for collision avoidance of autonomous underwater vehicles," *Information Sciences*, vol. 295, pp. 441–464, 2015.
- [43] O. Hassanein, S. Anavatti, H. Shim, and T. Ray, "Model-based adaptive control system for autonomous underwater vehicles," *Ocean Engineering*, vol. 127, pp. 58–69, 2016.
- [44] J. Cervantes, W. Yu, S. Salazar, and I. Chairez, "Takagi-sugeno dynamic neuro-fuzzy controller of uncertain nonlinear systems," *IEEE Transactions on Fuzzy Systems*, vol. PP, no. 99, 2016.
- [45] S. Chen and S. Billings, "Representations of non-linear systems the NARMAX model," *International Journal of Control*, vol. 49, no. 3, pp. 1013–1032, 1989.
- [46] O. Akanyeti, T. Kyriacou, U. Nehmzow, R. Iglesias, and S. a. Billings, "Visual task identification and characterization using polynomial models," *Robotics and Autonomous Systems*, vol. 55, pp. 711–719, 2007.
- [47] O. Akanyeti, U. Nehmzow, and S. a. Billings, "Robot training using system identification," *Robotics and Autonomous Systems*, vol. 56, no. 12, pp. 1027–1041, 2008.
- [48] J. Mu, D. Rees, and G. P. Liu, "Advanced controller design for aircraft gas turbine engines," *Control Engineering Practice*, vol. 13, pp. 1001–1015, 2005.
- [49] P. Maurya, E. Desa, A. Pascoal, E. Barros, G. Navelkar, R. Madhan, A. Mascarenhas, S. Prabhudesai, S. Afzulpurkar, A. Gouveia *et al.*, "Control of the Maya AUV in the vertical and horizontal planes: theory and practical results," in *Proceedings of the 7<sup>th</sup> IFAC Conference on Manoeuvring and Control of Marine Craft*, Lisbon, Portugal, 20-22 Sept. 2006.
- [50] C. McGann, F. Py, K. Rajan, H. Thomas, R. Henthorn, and R. McEwen, "A Deliberative Architecture for AUV Control," in *Proc. of IEEE International Conference on Robotics and Automation*, 19-23 May 2008, pp. 1049–1054.

- [51] T. Prestero, "Development of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle," in *Proc. of MTS/IEEE OCEANS Conf.*, 5-8 Nov. 2001, pp. 450–455.
- [52] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 AUV: From Survey to Intervention," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 1, pp. 46–53, Feb 2012.
- [53] A. D. Bowen, D. R. Yoerger, L. L. Whitcomb, and D. J. Fornari, "Exploring the Deepest Depths: Preliminary Design of a Novel Light-Tethered Hybrid ROV for Global Science in Extreme Environments," *Marine Technology Society Journal*, vol. 38, no. 2, pp. 92–101, 2004.
- [54] H. Singh, A. Can, R. Eustice, S. Lerner, N. McPhee, O. Pizarro, and C. Roman, "Seabed AUV offers new platform for high-resolution imaging," *EOS, Transactions of the AGU*, vol. 85, no. 31, pp. 294–295, 2004.
- [55] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.
- [56] A. Mosek, "The mosek optimization software," *Online at <http://www.mosek.com>*, vol. 54, 2010.
- [57] R. Sepulchre, M. Jankovic, and P. V. Kokotovic, *Constructive Nonlinear Control*. Springer London, 1997.
- [58] J. E. Refsnes, A. J. Sorensen, and K. Y. Pettersen, "Model-Based Output Feedback Control of Slender-Body Underactuated AUVs: Theory and Experiments," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 5, pp. 930–946, 2008.
- [59] J. Petrich and D. J. Stilwell, "Robust control for an Autonomous Underwater Vehicle that suppresses pitch and yaw coupling," *Ocean Engineering*, vol. 38, no. 1, pp. 197–204, 2011.
- [60] E. Hong and M. Chitre, "Roll Control of an Autonomous Underwater Vehicle Using an Internal Rolling Mass." Springer International Publishing, 2015, vol. 105, pp. 229–242.
- [61] H. T. Siegelmann, B. G. Horne, and C. L. Giles, "Computational capabilities of recurrent NARX neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 27, no. 2, pp. 208–15, 1997.
- [62] J. S.-H. Tsai, C.-T. Wang, C.-C. Kuang, S.-M. Guo, L.-S. Shieh, and C.-W. Chen, "A narmax model-based state-space self-tuning control for nonlinear stochastic hybrid systems," *Applied Mathematical Modelling*, vol. 34, no. 10, pp. 3030–3054, 2010.
- [63] Y. Hu, B. Liu, Q. Zhou, and C. Yang, "Recursive Extended Least Squares Parameter Estimation for Wiener Nonlinear Systems with Moving Average Noises," *Circuits, Systems, and Signal Processing*, vol. 33, no. 2, pp. 655–664, 2014.

- [64] T. Fossen, *Guidance and Control of Ocean Vehicles*. New York: Wiley, 1994, ch. 2, pp. 6–55.
- [65] F. Ornelas, E. N. Sanchez, and A. G. Loukianov, “Discrete-time inverse optimal control for nonlinear systems trajectory tracking,” in *Proc. of 49th IEEE Conference on Decision and Control*, Atlanta, GA, 15-17 Dec. 2010, pp. 4813–4818.
- [66] T. Yamamoto and S. Shah, “Design and experimental evaluation of a multivariable self-tuning PID controller,” *IEE Proceedings-Control Theory and Applications*, vol. 151, no. 5, pp. 645–652, 2004.
- [67] C. Silvestre, “Multi-objective optimization theory with application to the integrated design of controllers/plants for autonomous vehicle,” Ph.D. dissertation, Robot. Dept., Instituto Superior Technico (IST), Lisbon, Portugal, Jun. 2000.
- [68] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [69] C. Silvestre and A. Pascoal, “Control of the INFANTE AUV using gain scheduled static output feedback,” *Control Engineering Practice*, vol. 12, no. 12, pp. 1501–1509, 2004.
- [70] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K: Cambridge University Press, 2014.
- [71] P. Tøndel, T. Johansen, and A. Bemporad, “Evaluation of piecewise affine control via binary search tree,” *Automatica*, vol. 39, no. 5, pp. 945–950, 2003.

# Dissemination

## Journals

1. R. Rout and B. Subudhi, "Inverse optimal self-tuning PID control design for an Autonomous Underwater Vehicle", *International Journal of System Science*, vol. 48, no.2, pp.367-375, 2016 (Taylor & Francis).
2. R. Rout and B. Subudhi, "A backstepping approach for the formation control of multiple autonomous underwater vehicles using a leaderfollower strategy", *Journal of Marine Engineering and Technology*, vol. 15, no. 1, pp. 38-46, 2016 (Taylor & Francis).
3. R. Rout and B. Subudhi, "NARMAX Self-Tuning Controller for Line-of-Sight based Way-Point Tracking for an Autonomous Underwater Vehicle", *IEEE Transaction on Control Systems Technology*, DOI: 10.1109/TCST.2016.2613969.

## Conference

1. R. Rout and B. Subudhi, "Development of a NARMAX based constraint-adaptive heading controller for an Autonomous Underwater Vehicle", *Oceans 2016*, Sanghai, pp 1-5, 10-13 April 2016.

## Article under preparation

1. R. Rout and B. Subudhi, "Design of a Line-of-Sight guidance law based on Explicit model predictive control design for an Autonomous Underwater Vehicle", *IEEE Transaction on Control Systems Technology*

# Author's biography

Raja Rout was born to Sri. Ashok Kumar Rout and Smt. Promodini Rout on 3rd June, 1987 at Kendrapara, Odisha, India. He obtained a Bachelor's degree in Electronics and Instrumentation Engineering from B.P.U.T, Rourkela, Odisha in 2008 and a Master in Control System Engineering from National Institute of Technology Rourkela, India in 2012. He joined as a Institute PhD Scholar in the Department of Electrical Engineering, National Institute of Technology Rourkela in July 2012. His areas of academic pursuit include Robotics, Autonomous Underwater Vehicle, Control System, Adaptive Control and System Identification.

## **Communications:**

Address: Department of Electrical Engineering, National Institute of Technology Rourkela, Orissa, PIN: 769008.

e-mail: raja.rout86@gmail.com