

**ANOMALY DETECTION BASED ON THE ESTIMATION OF SPEED AND  
FLOW MAPPING FOR CONTROLLED LAGRANGIAN PARTICLES**

A Dissertation  
Presented to  
The Academic Faculty

By

Sungjin Cho

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in the  
School of Electrical and Computer Engineering

Georgia Institute of Technology

December 2017

Copyright © Sungjin Cho 2017

**ANOMALY DETECTION BASED ON THE ESTIMATION OF SPEED AND  
FLOW MAPPING FOR CONTROLLED LAGRANGIAN PARTICLES**

Approved by:

Dr. Fumin Zhang, Advisor  
Professor, School of Electrical and  
Computer Engineering  
*Georgia Institute of Technology*

Dr. Catherine Edwards, Co-advisor  
Assistant Professor, Department of  
Marine Science  
*University of Georgia*

Dr. Patricio Vela  
Associate Professor, School of Elec-  
trical and Computer Engineering  
*Georgia Institute of Technology*

Dr. Xiaoli Ma  
Professor, School of Electrical and  
Computer Engineering  
*Georgia Institute of Technology*

Dr. Michael West  
Senior Research Engineer, GTRI  
Robolab  
*Georgia Tech Research Institute*

Dr. Jonathan Rogers  
Assistant Professor, School of  
Mechanical Engineering  
*Georgia Institute of Technology*

Date Approved: November 9, 2017



This dissertation is dedicated to Ms. Suk-hee Choi in heaven

## ACKNOWLEDGEMENTS

This thesis would not have been possible without the help and support of many people who know me and love me. I would like to take an opportunity to thank them even though it would be impossible to list them all by name.

First and Foremost, I would like to thank to my advisor Dr. Fumin Zhang with all my heart. His deep and wide knowledge on underwater robotics always help solve difficult problems that I encounter. I cannot see more than his support, guidance, and encouragement during my Ph.D life at Georgia Tech. His great insights and unlimited passions on research inspire me. I am always greatly thankful to Dr. Zhang for sharing his breakthrough ideas with me and motivating me to achieve higher goals.

I would like to sincerely thank to my co-advisor Dr. Catherine Edwards. She introduced me to Oceanography and the Modena (underwater glider) that I had never experienced, and her teaching and patience encourage me to move one step forward to Ocean Science field.

I would like to thank my thesis committee members: Dr. Patricio Vela, Dr. Xiaoli Ma, Dr. Michael West, Dr. Jonathan Rogers, Dr. Catherine Edwards, for their precious time and valuable suggestions for the work done in this dissertation.

I would like to thank my fellow lab members in Georgia Tech Systems Research (GTSR) lab directed by Dr. Fumin Zhang. I have learned a lot by interacting with GTSR members; specially thanks to the GT-MAB team (Qiuyang, Ningshi, Vivek) and the GT-WMR team (Landan, Kevin, Andrew) and peer review members (Sean, Said, Paul). I also thankful to all my collaborators at Skidaway Institute of Oceanography, and Gray's Reef National Marine Sanctuary.

Without their love and belief, this thesis would not be achieved. I would like to dedicate this thesis to my father, my wife's parents, and my family.

## TABLE OF CONTENTS

<b>Acknowledgments</b> . . . . .	v
<b>List of Tables</b> . . . . .	x
<b>List of Figures</b> . . . . .	xi
<b>Chapter 1: Introduction</b> . . . . .	1
1.1 Background . . . . .	3
1.1.1 Controlled Lagrangian Particle Tracking . . . . .	3
1.1.2 Adaptive Inverse Dynamic Control . . . . .	5
1.1.3 Anomaly Detection of Autonomous Underwater Vehicles . . . . .	6
1.1.4 Acoustic Localization . . . . .	9
1.2 Summary of Contributions . . . . .	10
1.2.1 Anomaly Detection of Controlled Lagrangian Particles . . . . .	11
1.2.2 Anomaly Detection under Adaptive Control and Learning . . . . .	12
1.2.3 Acoustic Localization using a Passive Receiver . . . . .	13
1.2.4 Development of an Indoor Test Bed and Experimental Results . . . . .	14
<b>Chapter 2: Anomaly Detection of Controlled Lagrangian Particles</b> . . . . .	15
2.1 Vehicle Motion Model . . . . .	17

2.2	Controlled Lagrangian Localization Error . . . . .	21
2.3	Adaptive Learning Algorithm . . . . .	22
2.3.1	Input constraints . . . . .	27
2.3.2	Inaccuracy in flow modeling . . . . .	29
2.4	Anomaly Detection Algorithm . . . . .	30
2.5	Mathematical Simulations of Anomaly Detection Algorithms . . . . .	33
<b>Chapter 3: Anomaly Detection under Adaptive Control and Learning . . . . .</b>		<b>36</b>
3.1	Vehicle Motion Model . . . . .	36
3.1.1	Integrating Adaptive Control and Learning for Anomaly Detection .	37
3.2	Flow Canceling Control for Maintaining Heading . . . . .	38
3.2.1	Controlled Lagrangian prediction and localization errors . . . . .	41
3.2.2	Adaptive control and learning algorithms . . . . .	42
3.3	Flow Canceling Control for Trajectory Tracing . . . . .	50
3.3.1	Adaptive Control Algorithm . . . . .	50
3.3.2	Input constraints for adaptive control algorithm . . . . .	56
3.3.3	Inaccuracy in flow modeling for adaptive control algorithm . . . . .	57
3.3.4	Controlled Lagrangian Localization Error . . . . .	58
3.3.5	Adaptive Learning Algorithm . . . . .	59
3.3.6	Input constraints for adaptive learning algorithm . . . . .	63
3.3.7	Inaccuracy in flow modeling for adaptive learning algorithm . . . . .	65
3.4	Mathematical Simulations of Incorporating Adaptive Learning and Control for Anomaly Detection . . . . .	67
3.5	Mathematical Simulations of Adaptive Learning and Control Algorithms . .	70

3.5.1	Adaptive learning algorithm . . . . .	70
3.5.2	Adaptive control algorithm . . . . .	71
<b>Chapter 4: Acoustic Localization using a Passive Receiver . . . . .</b>		<b>76</b>
4.1	Localization Incorporating Flow Model and Acoustic Detection . . . . .	77
4.1.1	Acoustic measurement model . . . . .	78
4.1.2	Vehicle motion model . . . . .	79
4.1.3	Odometry using flow estimation . . . . .	80
4.1.4	Division of sensing regions . . . . .	86
4.1.5	Graph-based particle filter . . . . .	88
4.1.6	Mathematical Simulation of Acoustic Localization Algorithm . . .	91
4.2	Acoustic Detection Rate . . . . .	93
4.2.1	Configuration of acoustic array . . . . .	93
4.2.2	Detection rate . . . . .	95
4.3	Tidal Analysis . . . . .	97
4.4	Flow Direction . . . . .	99
4.5	Stratification . . . . .	105
<b>Chapter 5: Development of an Indoor Test Bed and Experimental Results . . . .</b>		<b>110</b>
5.1	Georgia Tech Miniature Autonomous Blimp (GT-MAB) . . . . .	110
5.1.1	Physical design . . . . .	112
5.2	GT-MAB Field Experiments with GT-WMR . . . . .	113
5.2.1	Georgia Tech wind measuring robot (GT-WMR) . . . . .	115
5.2.2	Adaptive learning algorithm . . . . .	117

5.2.3	Anomaly detection algorithm . . . . .	126
5.2.4	Adaptive control algorithm . . . . .	128
<b>Chapter 6: Conclusion and Future Work . . . . .</b>		<b>132</b>
<b>Appendix A: Publications . . . . .</b>		<b>135</b>
<b>Appendix B: Autopilot design for GT-MAB . . . . .</b>		<b>137</b>
B.1	Dynamics and Control . . . . .	137
B.2	Dynamics with 6-DOF . . . . .	137
B.3	Motion Primitives . . . . .	139
B.4	System Identification . . . . .	141
B.5	Speed and Heading Controller Designs . . . . .	144
B.5.1	Altitude controller with a scheduling algorithm . . . . .	145
<b>References . . . . .</b>		<b>155</b>

## LIST OF TABLES

4.1	Signal to noise ratio of tidal constituents <sup>†</sup> . . . . .	98
B.1	Speed and Yaw PID Controller Gains . . . . .	144
B.2	Altitude PID Controller Gains . . . . .	146

## LIST OF FIGURES

1.1	Two estimated paths of an AUV are identical, but they happen in different ways; the thruster fault of the AUV (left) and no thruster fault of the AUV (right) . . . . .	2
1.2	Controlled Lagrangian prediction error (CLPE) is the difference between the predicted trajectory and the estimated trajectory (left). CLPE increases over time at the experiment in Feb. 2013, shown by significant deviation of the estimated trajectory from the predicted trajectory (right). . . . .	5
1.3	An acoustic localization algorithm estimates vehicle path. The estimated path is the input to the adaptive learning algorithm. The anomaly detection algorithm uses the output of the learning algorithm, which is identified flow and controlled speed to detect abnormal vehicle motion. . . . .	11
2.1	The diagram of the adaptive learning algorithm . . . . .	16
2.2	Every 20 sec, the heading angle command is changed with this order $0^\circ$ , $90^\circ$ , $180^\circ$ , $270^\circ$ . repeatedly, When there is no flow, the simulated true trajectory has one square, but it is not because of flow. . . . .	34
2.3	CLLE converges to zero after 7 intervals, but CLLE is abruptly increasing when vehicle speed reduce to 0.5m/s because of abnormal motion. After that, CLLE maintains closing zero (1 cycle=10 intervals=200 sec). . . . .	34
2.4	True through-water speed (upper) and flag (bottom) . . . . .	35
2.5	Estimation error of vehicle speed (1 cycle =10 intervals = 200 sec) . . . . .	35
2.6	Convergence of flow parameters: Six flow parameters converges to true values after 20 sec. . . . .	35
2.7	Convergence of identification error: Identification error of flow parameters converges to zero after 20 sec. . . . .	35



3.1	When $\ \mathbf{e}_{L_1}\ $ is larger than $\ \mathbf{e}_P\ $ at time $t$ , the identified trajectory is relatively far away from the estimated trajectory . . . . .	38
3.2	When $\ \mathbf{e}_{L_2}\ $ is smaller than $\ \mathbf{e}_P\ $ at time $t$ , the identified trajectory is relatively close to the estimated trajectory. . . . .	38
3.3	Predicted (black), estimated (blue), and identified (red) trajectories . . . . .	68
3.4	Controlled Lagrangian localization and prediction Error (CLLE, CLPE) . . .	68
3.5	Identified through-water speed (top) and flag (bottom) . . . . .	69
3.6	Identification error of through-water speed . . . . .	69
3.7	Identified trajectory and simulated true trajectory . . . . .	71
3.8	Convergence of CLLE: CLLE converges to zero over one cycle (1 cycle=10 intervals=200 sec). . . . .	71
3.9	Convergence of horizontal flow parameters: Parameters for horizontal flow converge to the true values over 200 secs . . . . .	71
3.10	Identification error of horizontal flow parameters: Identification error converge to zero over 200 secs . . . . .	71
3.11	Convergence of vertical flow parameters: Parameters for vertical flow converge to the true values over 200 sec . . . . .	72
3.12	Identification error of vertical flow parameters: Identification error converge to zero over 200 sec . . . . .	72
3.13	Convergence of controller gains: Controller gains converge to the true values over 200 sec . . . . .	72
3.14	Identification error of controller gains: Identification error converge to zero over 200 sec . . . . .	72
3.15	Simulated true trajectory and predicted trajectory . . . . .	73
3.16	Convergence of CLPE: CLPE converges to zero over one and half cycle (1.5 cycle=15 intervals=300 sec). . . . .	73
3.17	Convergence of horizontal flow parameters: Parameters for horizontal flow converge to the true values over 300 sec . . . . .	74

3.18	Estimation of horizontal flow parameters: Estimation error converge to zero over 300 sec . . . . .	74
3.19	Convergence of vertical flow parameters: Parameters for vertical flow converge to the true values over 300 sec . . . . .	74
3.20	Estimation of vertical flow parameters: Estimation error converge to zero over 300 sec . . . . .	74
3.21	Convergence of controller gains: Parameters for controller gains converge to the true values over 300 sec . . . . .	75
3.22	Estimation error of controller gains: Estimation error converge to zero over 200 secs . . . . .	75
4.1	The schematic of the waypoint controller: $\mathbf{sn}$ represents the velocity vector of the vehicle with constant speed $s$ . $u$ is a desired heading angle called a control input. $\mathbf{T}$ is the unit vector of $\overrightarrow{O\hat{\mathbf{x}}}$ , $\mathbf{N}$ is the rotation vector of $\mathbf{T}$ by 90 degrees counterclockwise, and $X_s$ - $Y_s$ is the $X$ - $Y$ axis in an inertial frame. . .	81
4.2	Three transmitters and regions of detection . . . . .	88
4.3	The graph transformed from Figure 4.2 . . . . .	88
4.4	Generation of initial particles; $\hat{r}_0$ and $\hat{\theta}_0$ represent the initial position of the AUV. $\phi_n$ and $R_m$ are the orientation and the size of the half circle, respectively. . .	89
4.5	True trajectories of the AUV . . . . .	92
4.6	Estimation error . . . . .	92
4.7	The two dimensional placement of receivers and tags: The origin of the figure is the center of the array, located 3.09-km ENE of the National Data Buoy Center (NDBC) buoy at Gray's Reef (31.400N, 80.868W). The circle about each receiver reflects a nominal 200-m radius of detection during summer as observed by [34]. . . . .	94
4.8	Mean hourly detections: The horizontal and vertical axes in the figure shows the index number of receivers and tags, respectively. The color bar represents mean values of hourly detections. . . . .	96
4.9	Mean hourly detections (on the left) and standard deviation of hourly detections (on the right) . . . . .	97

4.10	SNR on the configuration: The direction of each arrow represents the direction of the tag toward the receiver. The length of each arrow represent the magnitude of SNR. Alongshore pairs have much stronger SNR than cross-shore pairs . . . . .	99
4.11	The reference direction at hourly detections . . . . .	100
4.12	Maximum (left) and minimum (right) hourly detections, taken over each successive $M_2$ tidal cycle, for three receiver-tag pairs oriented alongshore. .	101
4.13	Phase of flow with respect to the direction from Tag 3M to Receiver 6 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (negative alongshore, NNE); phases of 90 and 270 degrees correspond to offshore/positive and onshore/negative cross-shore flow, respectively. . . . .	102
4.14	Flow direction at minimum detection rates of Tag 3M at Receiver 6 . . . . .	103
4.15	Phase of flow with respect to the direction from Tag 1M to Receiver 4 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (negative alongshore, NNE); phases of 90 and 270 degrees correspond to offshore/positive and onshore/negative cross-shore flow, respectively. . . . .	103
4.16	Phase of flow with respect to the direction from Tag 4M to Receiver 1 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (positive alongshore, SSW); phases of 90 and 270 correspond to onshore/negative and offshore/positive cross-shore flow, respectively. . . . .	104
4.17	Phase of flow with respect to the direction from Tag 5M to Receiver 6 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (positive cross-shore, ESE); phases of 0 and 180 degrees correspond to offshore/positive and onshore/negative cross-shore flow, respectively. . . . .	105
4.18	Phase of flow with respect to the direction from Tag 5M to Receiver 4 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (negative cross-shore, WNW); phases of 0 and 180 degrees correspond to onshore/negative and offshore/positive cross-shore flow, respectively. . . . .	105

4.19	Phase of flow with respect to the direction from Tag 2M to Receiver 1 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (negative cross-shore, WNW); phases of 0 and 180 correspond to onshore/negative and offshore/positive cross-shore flow, respectively. . . . .	106
4.20	Collected density data (top), and post processed $N^2$ stratification index (bottom) . . . . .	107
4.21	Hourly detections (top), $N^2$ (middle), and depth-averaged ADCP currents (bottom) . . . . .	108
5.1	Children play with the GT-MAB . . . . .	111
5.2	The GT-MAB's gondola includes four motor thrusters. Two motor thruster are vertically installed to control up-down motion. The other two motor thrusters are horizontally installed to control forward-backward and spinning motions, which are presented in [72]. . . . .	111
5.3	This gondola has one more motor than that of Figure 5.2. The motor installed along the side-way direction provides thrust for side-way motion of the GT-MAB. . . . .	113
5.4	Indoor test bed: The yellow bulbs represent infrared motion capture cameras. The blue square represents the Dyson fan. The star represents the starting point of the GT-MAB. The red line represents the trajectory of the GT-MAB. When the GT-MAB is flying at the starting point, the GT-MAB motion is disturbed by flow generated from the Dyson fan. Then, the motion capture cameras collect the attitudes and trajectory of the GT-MAB. . .	114
5.5	GT-WMR: The GT-WMR contains two main components: an omnidirectional robot called omnibot and three wind sensors. The three wind sensors on an horizontal black frame are connected to the omnibot. The black frame can be moved vertically to measure wind speed at different heights. The omnibot has two Arduino board; one is an embedded Arduino board that receives wheel command from an ground station, and drive the wheels of the omnibot. The other is an additional Arduino board that collects wind measurements and send them to the ground station via the embedded Arduino board. The four gray spheres represent makers that the motion capture cameras recognize. The motion capture cameras collect attitudes and trajectories of the omnibot. . . . .	116
5.6	Two types of wind sensors: directional and non-directional . . . . .	117

5.7	Three wheels of the omnidirectional robot . . . . .	117
5.8	The GT-MAB trajectory without the wind source (the black) and with the wind source (the blue) . . . . .	118
5.9	Location of the wind source (black circle), and multiple starting points for the GT-MAB deployment . . . . .	118
5.10	Estimated (blue) and identified (red) trajectories starting at No. 1 (black) . .	119
5.11	Estimated (blue) and identified (red) trajectories starting at No. 2 (black) . .	119
5.12	Estimated (blue) and identified (red) trajectories starting at No. 3 (black) . .	119
5.13	Estimated (blue) and identified (red) trajectories starting at No. 4 (black) . .	119
5.14	Estimated (blue) and identified (red) trajectories starting at No. 5 (black) . .	119
5.15	Estimated (blue) and identified (red) trajectories starting at No. 6 (black) . .	119
5.16	Estimated (blue) and identified (red) trajectories starting at No. 7 (black) . .	120
5.17	Estimated (blue) and identified (red) trajectories starting at No. 8 (black) . .	120
5.18	Estimated (blue) and identified (red) trajectories starting at No. 9 (black) . .	120
5.19	Estimated (blue) and identified (red) trajectories starting at No. 10 (black) .	120
5.20	Estimated (blue) and identified (red) trajectories starting at No. 11 (black) .	120
5.21	Estimated (blue) and identified (red) trajectories starting at No. 12 (black) .	120
5.22	Estimated (blue) and identified (red) trajectories starting at No. 12 (black) .	121
5.23	Estimated (blue) and identified (red) trajectories starting at No. 13 (black) .	121
5.24	Estimated (blue) and identified (red) trajectories starting at No. 14 (black) .	121
5.25	Estimated (blue) and identified (red) trajectories starting at No. 15 (black) .	121
5.26	Estimated (blue) and identified (red) trajectories starting at No. 15 (black) .	121
5.27	Estimated (blue) and identified (red) trajectories starting at No. 16 (black) .	121
5.28	The path (blue) and waypoints (red) of the GT-WMR . . . . .	122

5.29	Measured flow velocity at each waypoint . . . . .	122
5.30	Identified wind velocities along the GT-MAB trajectories . . . . .	123
5.31	Reconstructed wind speed map using identified velocities of Figure 5.30 . .	125
5.32	Measured wind speed map using measured velocities of Figure 5.29 . . . .	125
5.33	Reconstructed wind direction map using identified velocities of Figure 5.30	125
5.34	Measured wind direction map using measured velocities of Figure 5.29 . . .	125
5.35	Error map of wind speed . . . . .	126
5.36	Error map of wind direction . . . . .	126
5.37	Identified wind velocities along the GT-MAB trajectories . . . . .	127
5.38	CLLE along the X-axis (black), and CLLE along the Y-axis (red) . . . . .	128
5.39	Estimated (blue) and identified (red) trajectories with identified flow (black)	128
5.40	The adaptive control algorithm makes the GT-MAB trajectory (red) follow the predicted trajectory (black); without the adaptive control algorithm, the GT-MAB trajectory (blue) is away from the predicted trajectory. . . . .	130
5.41	CLPE along the X-axis keeps increasing until 1m without the adaptive con- trol algorithm (blue); but, CLPE along the X-axis is bounded by 0.3m with the adaptive control algorithm (red). . . . .	130
5.42	The similar trend of the increasing rate is shown in both CLPE along the Y-axis without the adaptive control algorithm (blue) and CLPE along the Y-axis with the adaptive control algorithm (red). . . . .	130
B.1	Inertial and body coordinate frames. $O_B - X_B Y_B Z_B$ represents the body- fixed axis of the blimp. $O_I - X_I Y_I Z_I$ represents the inertial axis of the blimp. The origin $O_B$ is the center of the buoyancy (CB) of the blimp. The center of buoyancy (CB) is the same as the center of volume of the balloon. The height $H$ of the blimp is 0.46 meters and the diameter $D$ is 0.72 meters. . . .	138

B.2	Poles and zeros of the open loop transfer function $P_1(s)$ and the closed-loop transfer function $G_1(s)$ for forward speed control. Blue markers represent poles and zeros of the open-loop transfer function. Red markers represent poles and zeros of the closed-loop transfer function. One open-loop pole located at $(-5.88, 0)$ and one closed-loop pole located at $(-4.88, 0)$ are omitted in the figure. . . . .	145
B.3	Simulated step response (red) and measured step response (blue) for two motion primitives of the GT-MAB . . . . .	146
B.4	Measured step response for changing altitude (top) and switching modes (bottom) . . . . .	148

## SUMMARY

Anomaly detection for autonomous underwater vehicles (AUVs) is especially challenging because AUVs explore distant and hostile environments with unpredictable disturbances. Mechanical and electrical subsystems in AUVs can be easily exposed to harsh environments such as the deep sea and the polar oceans. During long-range or long-period missions of AUVs, marine creatures and biofouling may harm AUV sensors and thrusters. Thus, anomaly detection in AUVs is of great importance for vehicle survivability.

The main contribution of this dissertation is a set of algorithms that detect anomaly of autonomous underwater vehicles (AUVs) without relying on sensors monitoring vehicle components. Only using trajectory information, the proposed strategy detects abnormal vehicle motion under unknown ocean flow. It has the potential for mitigating abnormal vehicle motion with path-planning and controller design of AUVs. In the Controlled Lagrangian particle framework, the adaptive learning algorithm identifies vehicle motion while producing the estimated flow velocity and vehicle speed. The estimated vehicle speed determines whether or not the vehicle motion is abnormal. A false alarm prevention scheme is proposed to avoid false alarm caused by estimation error of vehicle speed. Moreover, incorporating adaptive control and learning algorithms is proposed to enhance the accuracy of anomaly detection. The experimental results of the Georgia Tech Miniature Autonomous Blimp (GT-MAB) and Georgia Tech Wind Measuring Robot (GT WMR) in an indoor test bed verify the proposed algorithms. To estimate vehicle trajectory underwater, the AUVs are equipped with acoustic passive receivers primarily used for monitoring tagged fish. We propose a localization algorithm that integrates odometry using flow estimation and acoustic detection in order to accurately estimate vehicle position. Acoustic detection that reduces odometry error can be poor due to underwater environmental factors. We identify environmental factors using the specially designed array of passive acoustic receivers and fish tags in Gray's Reef National Marine Sanctuary.



# **CHAPTER 1**

## **INTRODUCTION**

Anomaly detection is critical to autonomous vehicles. Faulty software and hardware in unmanned aerial vehicles, autonomous ground vehicles, and autonomous underwater vehicles can lead to unsuccessful missions. Anomaly detection allows vehicles to overcome software and hardware problems by activating redundant systems or reconfiguring the malfunctioning systems.

Anomaly detection for autonomous underwater vehicles (AUVs) is especially challenging because AUVs explore distant and hostile environments with unpredictable disturbances. Mechanical and electrical subsystems in AUVs can be easily exposed to harsh environments such as the deep sea and the polar oceans. During long-range or long-period missions of AUVs, marine creatures and biofouling may harm AUV sensors and thrusters [1]. Thus, anomaly detection in AUVs is of great importance for vehicle survivability.

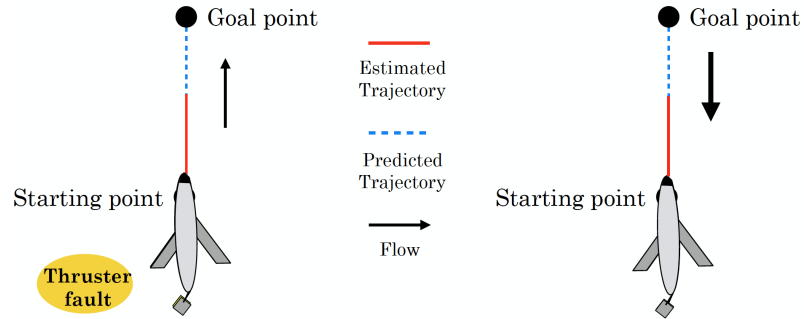
Many studies have addressed abnormal motion through monitoring sensors installed in AUVs. Components vulnerable to faults (i.e., thrusters and actuators) can be responsible for abnormal motion of AUVs. For example, damaged propellers impair propulsive efficiency to control vehicle speed. These faults could be detected with rotational speed sensors of propellers; however, this approach requires increased hardware complexity and cost, and it may not detect unexpected external disturbances (e.g., white shark attack).

We propose novel anomaly detection algorithms for AUVs. Instead of using measurements from sensors that monitor hardware components, in this dissertation, we incorporate trajectory data to detect abnormal vehicle motion. Given a trajectory, adaptive learning identifies vehicle motion while producing the estimates of vehicle speed and flow velocity. The vehicle speed estimate is used to determine whether or not vehicle motion is abnormal. Anomaly occurs when the vehicle speed estimate is out of the range of AUV speed

in normal operation; otherwise, no anomaly is identified. Compared to existing algorithms that detect faults of individual components, trajectory-based anomaly detection of AUVs has the potential to mitigating abnormal vehicle motion with path planning and controller design of AUVs.

We use vehicle speed to determine if the vehicle's motion is abnormal. For the estimation of vehicle speed, we develop an on-line adaptive learning algorithm based on the framework of controlled Lagrangian particle tracking. The adaptive learning algorithm uses the vehicle trajectory, the net motion of the AUV due to vehicle propulsion, advection by flow, and external forces, to estimate vehicle speed.

Estimating vehicle speed from an AUV's trajectory is substantially difficult in that ocean flow affecting AUV motion is unknown. Here and after, we call vehicle speed *controlled* speed to distinguish vehicle speed relative to water, which is controlled by the vehicle, and vehicle speed produced by the time derivative of trajectory data. In presence of flow, controlled speed is different from time derivative of trajectory data. Consequently, it is impossible to estimate controlled speed from trajectory data without identifying ocean flow. It is difficult for us to detect abnormal vehicle motion under unknown flow. One class of AUVs called underwater gliders typically move at controlled speeds between 25-35 cm/s. Figure 1.1 depicts a glider with a thruster fault and shows estimated paths of an AUV in two cases: faulty case on the left, and non-faulty case on the right. Consider one



**Figure 1.1:** Two estimated paths of an AUV are identical, but they happen in different ways; the thruster fault of the AUV (left) and no thruster fault of the AUV (right)

starting point where an AUV is deployed, and one goal point to be reached. The predicted

trajectory is the straight line between the starting and the goal point before deployment. However, the estimated trajectory do not reach the goal point after deployment because of two different reasons on the left and the right. On the left case, the faulty thruster of the AUV stops vehicle motion and the vehicle cannot reach the goal. On the right case, the AUV has no thruster fault; however, the direction of strong flow, from top to bottom, disturbs vehicle motion, leading to a trajectory that cannot reach the goal point. Therefore, identification of flow serves to prevent false alarms by identifying unexpected motion of AUVs that can be explained by flow. Controlled Lagrangian particle tracking serves as a theoretical tool that analyzes interaction between AUV motion and ocean flow [2]. In this framework describing the motion of AUVs, adaptive learning identifies vehicle motion under unknown ocean flow while estimating controlled speed and ocean flow from trajectory data.

## **1.1 Background**

Over the past decades, autonomous underwater vehicles have proven to be valuable sensing platforms in a variety of scientific and practical missions [3]. One class of AUVs, called underwater gliders, achieve long endurance or ranges by taking advantage of an energy efficient method of propulsion [4]. Gliders use fixed wings to change their buoyancy and center of gravity to translate vertical descent/ascent to forward and turning motion in the horizontal plane without motor thrusters such as propellers [5]. The motion is particularly energy efficient; RU-27, a Slocum glider operated by Rutgers University, traveled 7400 km for 221 days from the US to Spain on a single battery charge [6].

### 1.1.1 Controlled Lagrangian Particle Tracking

High navigational performance of AUVs is essential for persistent and efficient collection of information-rich data [7], and serious performance degradation can result when flow speed is comparable to or exceeds controlled speed of AUVs, as is the case for underwater

gliders [8]. Controlled Lagrangian particle tracking (CLPT) is a theoretical framework to analyze the interaction between ocean flow and AUV control [9]. In contrast to *passive* Lagrangian methods, an AUV is viewed as a *controlled* Lagrangian particle in the sense that AUVs are not freely advected by ocean flow. The trajectory of the controlled Lagrangian particles can be generated from the vehicle motion model as follows:

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}_R(\mathbf{x}, t) + \mathbf{v}_R(\psi_c(t)), \quad (1.1)$$

where  $\mathbf{x}$  is true position of the AUV that is assumed to be known. This assumption will be removed in Chapter 4 where we discuss how to estimate  $\mathbf{x}$ .  $\mathbf{F}_R$  is an actual flow,  $\mathbf{v}_R$  is the through-water velocity, and  $\psi_c(t)$  is heading angle command. To track the controlled Lagrangian particle, we generate the predicted trajectory of the AUV by simulating a vehicle motion model composed of modeled flow velocity and through-water velocity. That is,

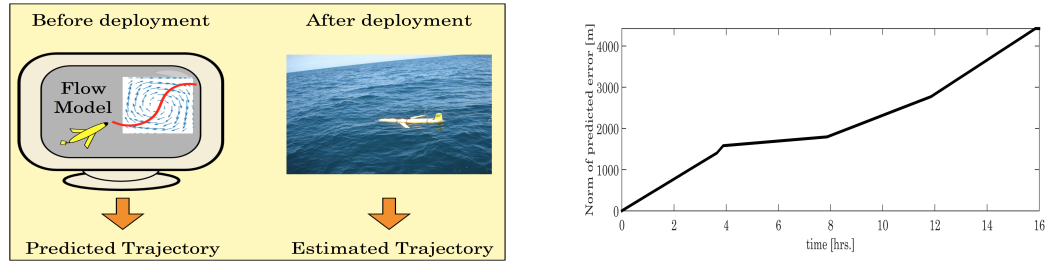
$$\frac{d\mathbf{y}}{dt} = \mathbf{F}_M(\mathbf{y}, t) + \mathbf{v}_M(\psi_c(t)), \quad (1.2)$$

where  $\mathbf{y}$  is predicted position,  $\mathbf{F}_M$  is a known modeled flow,  $\mathbf{v}_M$  is through-water velocity. Then, we compare this predicted trajectory with the estimated trajectory of the AUV. The discrepancy between the two trajectories shows the tracking performance of the controlled Lagrangian particle, called controlled Lagrangian prediction error (CLPE). CLPE,  $\mathbf{e}$ , is given by

$$\mathbf{e} = \mathbf{x} - \mathbf{y}. \quad (1.3)$$

CLPE is a crucial measure that can be interpreted as the degree to which AUV navigational performance is degraded by ocean model inaccuracy. In [9], the resolution of flow model is shown to determine the upper bound of CLPE growth. Station-keeping and transect following controllers can cause affect CLPE growth. Increasing CLPE implies that the estimated trajectory is significantly deviated from the predicted trajectory. Figure 1.2 shows

growing CLPE computed by the estimated trajectory and the predicted trajectory during February 2013 field experiment in Long Bay, SC.



**Figure 1.2:** Controlled Lagrangian prediction error (CLPE) is the difference between the predicted trajectory and the estimated trajectory (left). CLPE increases over time at the experiment in Feb. 2013, shown by significant deviation of the estimated trajectory from the predicted trajectory (right).

Our work uses models of AUV motion under ocean flow to detect abnormal motion of AUVs. Controlled Lagrangian particle tracking (CLPT) addresses interactions between AUV motion and ocean flow. Our problem is related to the framework of CLPT because our work deals with the combination of flow motion and controlled AUV motion; however, our work is different from CLPT in the sense that controlled speed is extracted from the estimated trajectory, while CLPT is a theoretical tool that evaluates the accuracy of the predicted trajectory. In the next section, we describe adaptive inverse dynamic control that identifies unknown parameters from observed output data.

### 1.1.2 Adaptive Inverse Dynamic Control

Adaptive control systems can be tuned using parameters from observed output data. For linear time invariant systems, model reference adaptive control [10] and self tuning controllers [11] were developed to identify parameters of linear models. The identified parameters are used to change linear controller gains so that plant output follows model output. For nonlinear systems, adaptive inverse dynamic control was developed to identify parameters of nonlinear controller in manipulator applications [12, 13, 14]. The nonlinear, model-based controller uses the identified parameters so that the joint positions of a manipulator follows

its desired positions. Our work is similar to previous work in that we use flow canceling controller based on multiple nonlinear time-varying functions. However, previous work did not use identified parameters for anomaly detection, but improve plant performance by compensating the weakness of the nonlinear controller. Our work identifies unknown parameters to determine abnormal vehicle motion. In addition, the previous work deals with manipulator applications, and but our work treats marine robots in flow fields as controlled Lagrangian particles.

### 1.1.3 Anomaly Detection of Autonomous Underwater Vehicles

Anomaly detection in data helps us identify abnormal behaviors and reconfigure malfunctioning vehicle components. Recent ground robot and marine vehicle studies have detected anomalous signals of sensors to identify abnormal motion of robots and vehicles [15, 16]. A fault is defined as an unpermitted deviation of at least one characteristic property of a variable from an acceptable behavior [17]. A traditional approach to detecting a fault in the system is to use multiple sensors that measure the same physical quantity and detects the occurrence of a fault by a voting technique [18].

Fault detection data analysis techniques can use labeled or unlabeled data. Both methods have associated difficulties for anomaly detection. The labels of data points denote if an individual data point is normal or abnormal. It is expensive to acquire accurate and representative data points that cover all types of normal and abnormal behaviors in systems. A number of techniques have been developed for modeling normal behaviors in systems with unlabeled data. However, the techniques focus on modeling normal behaviors in systems assuming that normal behaviors occur far more frequently than abnormal behaviors in unlabeled data. If this assumption is not true, then such techniques can produce incorrect detection results, or false alarms [19].

Using unlabeled data, dynamical system models can be used for model-based fault detection. The installation of the extra physical components in the system increases cost

and complexity of the system [20]. Model-based schemes offer improvements in cost and complexity, and employ dynamical system models to detect component faults in the system. A linear dynamical system model is represented by

$$x(t+1) = Ax(t) + Bu(t) + Ed(t) + Bf_a(t), \quad y(t) = Cx(t), \quad (1.4)$$

where  $x \in \mathbb{R}^n$  is the state vector,  $u \in \mathbb{R}^q$  is the plant input vector,  $y \in \mathbb{R}^m$  is the output vector,  $d \in \mathbb{R}^n$  is the noise vector, and  $f_a \in \mathbb{R}^n$  is the fault vector. A great number of theoretical methods have been developed for detecting faults from equation (1.4) [21]. For example, the unknown input observer approach was developed to identify  $f_a(t)$ . The basic idea of this approach is to decouple unknown noise vector  $d(t)$  from estimation error. By using new variables transformed from  $x(t)$ , a stable observer was designed for the identification of  $f_a(t)$ . However, most theoretical methods that use the dynamical system models containing the additive fault model can not detect unmodeled faults. Although fault is modeled in dynamical systems with known scenarios, unknown fault scenarios can occur in critical subsystems composed of sensors and actuators. Other noncritical subsystems are able to cause some fault of the system components. For example in 2015, a software configuration error in an underwater glider disabled an internal mass shifter that adjusts the vehicle's trim, leading the vehicle to sink to the sea floor and resulting in temporary loss [16].

For AUVs, most fault detection algorithms have dealt with abnormal behaviors of the system components that are the most vulnerable to faults [22, 23, 24]. Blocked propellers, leaking thrusters, and rotor failure are documented as frequently occurring faults [25]. In order to detect thruster faults and to identify model parameter changes after fault in [22], an approximate probability distribution of a motion variable such as surge velocity, sway velocity, and yaw rate is iteratively computed. After the locally weighted projection regression yields multiple trained models for given data sets, the motion variable is estimated by a filtering algorithm. In [23], thruster and actuator faults are detected by observing abrupt

change in control-force estimates within a certain time window.

Instead of detecting faults in an individual component, vehicle motion can be used for anomaly detection. The yo-yo trajectory of underwater gliders in the vertical plane of 3-D space is achieved by changing the glider's pitch angle and total mass. Authors in [26] use a threshold technique to prevent a glider from hitting the sea floor. The deviation from expected AUV motion in the vertical plane is detected by monitoring stern plane angle, pitch angle, and depth rate. However, due to the cost and slow speed of Iridium satellite communications, it may not be feasible to telemeter the required sensor fields back to shore in order to monitor for faults.

Our work addresses anomaly detection of AUVs by using dynamical system models and estimated trajectory data. The dynamical system models we use enable simultaneous estimation of controlled speed and flow velocity from an estimated trajectory. Our models may be viewed as the dynamical system models used for model-based fault detection in the literature. However, the system models in the literature contain an additive fault model that represents a fault signal with known fault scenarios. Instead of the additive fault model, our model has unknown parameters developed in the framework of controlled Lagrangian particles. The unknown parameters that represent flow velocity and controlled speed can change according to abnormal motion. The use of trajectory data in our work follows similar motivations as surveillance applications (e.g., [26]). However, our work uses underwater trajectory data of autonomous underwater vehicles while previous work [27, 28] uses car trajectory data and surface trajectory data of marine vessels, both systems in which trajectory information is available in real time. When trajectory data is used in surveillance applications, many schemes such as clustering techniques and optimization techniques have been developed to identify path models that represent normal motion of the ground and marine vehicles [26, 27, 28]; however, we identify unknown parameters of dynamical system models from trajectory data; identifying unknown parameters means that flow velocity and controlled speed composed of the unknown parameters are identi-



fied from trajectory data. Then we check whether extracted controlled speed is within the predetermined range of vehicle speed under normal vehicle operation to detect abnormal motion of the vehicles. In the next section, we describe acoustic localization algorithms that estimate trajectory information.

#### 1.1.4 Acoustic Localization

AUV localization is a challenging issue in marine robotics. The global positioning system (GPS) and radio-frequency signals are not available for positional estimation of AUVs underwater. Many researchers have proposed various methods to solve the localization problem of AUVs (e.g., [29, 30]). Most methods focus on decreasing the dead-reckoning or odometry error of AUVs by using external acoustic sensors that provide range measurements. A basic kinematic model that represents vehicle motion ignoring flow is used to predict dead-reckoning or odometry error. In order to reduce predicted odometry error, active acoustic localization employs information on time of arrival in order to accurately estimate distances between transmitters and receivers.

Here we develop a localization algorithm combining an odometry model and a passive acoustic receiver [31] to detect transmitters at known fixed locations. Unlike basic odometry models that do not account for ocean currents, our odometry uses flow estimation under the CLPT framework so that we can more accurately predict odometry error. The passive acoustic receiver listens for signals from transmitters and record only receiver time stamps and the identity of transmitters. Since acoustic signals can be significantly disturbed by environmental factors, the accuracy of the localization algorithm depending on the receiver measurements can be low.

Over several decades, acoustic telemetry has been developed and used to monitor fish and marine mammals in freshwater and oceanic environments [32]. Acoustic telemetry uses such passive acoustic receivers to monitor the presence of fish. Ecological and behavioral analysis of telemetric data have revealed important new information about life cycles

of marine species [33]. However, the accuracy of telemetry is dependent on detection range, or the maximum distance over which receivers are able to detect transmitters/tags. If detection range is unknown, telemetry does not inform scientists about where the detected fish are located. Moreover, detection range with spatial and temporal variability is affected by a great number of environmental factors such as density, density stratification, and mixing, which are mediated by wind, buoyancy input, and other effects [32, 34].

Previous studies have examined tidal influence on acoustic detection patterns. Authors in [35] use Fourier analysis of detection rate to show that fish have a tidal pattern in their movements. When more than one receiver detects the transmitter in fish within a tidal cycle, authors insist that the fish's position has changed because of tidal influence. However, their results assume that detection range remains constant over a tidal cycle. More recent work [34] using data from a static array of multiple receivers deployed near an acoustic Doppler current profiler (ADCP) that measured currents over multiple years suggests that detection rate and range may be dominated by environmental processes. While the annual cycle has the strongest control on detection rate, patterns in detection probability were noted at seasonal cycles, tidal cycles, and synoptic-scale weather events.

Our work addresses identifying acoustic detection patterns influenced by tidal flow. Previous work [34] describes acoustic detections related to tidal cycles. Our problem is related to previous work in finding the relationship between tidal flow and acoustic detection; however, our work is different from previous work in that we identify environmental factors by using the specially designed acoustic array. The vertical and horizontal axis of the acoustic array in the 2D plane is aligned with the semi-major axis and semi-minor axis of tidal ellipse, respectively.

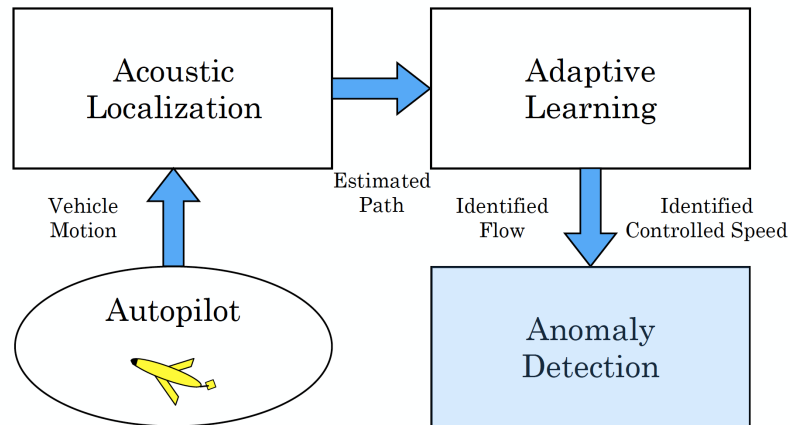
## **1.2 Summary of Contributions**

The rest of the dissertation is organized as follows. Chapter 2 describes a new method developed for anomaly detection using controlled Lagrangian particle methods. Chapter

3 extends the method with adaptive control and learning, Chapter 4 develops a technique of localization that incorporates a flow model and passive acoustics, and describes results from a field experiment. Chapter 5 verifies the algorithms developed in Chapters 3 and 4 in an indoor test bed using a miniature autonomous blimp and a wind measuring robot. Summaries of these chapters are given below.

### 1.2.1 Anomaly Detection of Controlled Lagrangian Particles

We formulate an anomaly detection problem of AUVs using trajectory information in the framework of controlled Lagrangian particles. First, AUV motion is learned from estimated trajectory. The on-line adaptive learning algorithm simultaneously estimates flow velocity and controlled speed from estimated trajectory while guaranteeing error convergence, parameter convergence, and robustness. Then, estimated controlled speed is used as a decision variable in that we obtain the known normal range of controlled speed of AUVs as an indicator of normal motion. Estimated controlled speed may be inaccurate at a certain time interval in spite of parameter convergence. The comparison of modeled flow velocity to estimated flow velocity can improve the reliability of estimated controlled speed. The block diagram for anomaly detection is shown in Figure 1.3.



**Figure 1.3:** An acoustic localization algorithm estimates vehicle path. The estimated path is the input to the adaptive learning algorithm. The anomaly detection algorithm uses the output of the learning algorithm, which is identified flow and controlled speed to detect abnormal vehicle motion.

Estimated controlled speed may not be accurate due to the inaccuracy of estimated trajectory. Acoustic sensor error and localization algorithm error can be included in the estimated trajectory. The use of controlled speed extracted from the estimated trajectory may induce false alarms. We utilize a predicted trajectory as extra information to reduce the false alarms.

Estimated trajectory may be significantly different from the trajectory we predicted before deploying the AUV due to unknown ocean flow that affects AUV motion. Consequently, the estimated trajectory may not be accurate enough to detect anomalies using only the controlled speed that is extracted using the on-line adaptive learning algorithm. To improve accuracy, an adaptive control algorithm is designed. The adaptive control algorithm implemented on AUVs enables the estimated trajectory to follow the predicted trajectory so that the accuracy of anomaly detection from the estimated trajectory is enhanced.

### 1.2.2 Anomaly Detection under Adaptive Control and Learning

Incorporating adaptive control and learning algorithms can improve the estimate of controlled speed. When the adaptive learning algorithm only uses the estimated trajectory, the learning algorithm has no knowledge on how accurate the estimated trajectory is. If both the predicted trajectory and the estimated trajectory are provided to the adaptive learning algorithm, the adaptive learning algorithm can compare the estimated trajectory with the predicted trajectory in order to precisely estimate controlled speed.

The adaptive learning algorithm generates a trajectory called an identified trajectory based on the estimation of controlled speed and flow velocity. Controlled Lagrangian localization error (CLLE) is used to evaluate the difference between the estimated trajectory and the identified trajectory. In the case of large CLLE, the controlled speed generated by the learning algorithm should not be trusted. The adaptive control algorithm is developed to make the estimated trajectory follow the predicted trajectory. Controlled Lagrangian prediction error (CLPE) is used to evaluate the difference between the estimated trajec-

tory and the predicted trajectory. If CLLE is larger than CLPE, estimated controlled speed from the learning algorithm can be inaccurate in that the identified trajectory is relatively far away from the estimated trajectory when we compare the identified trajectory and the predicted trajectory with respect to the estimated trajectory. Thus, we derive the conditions theoretically to find when CLLE is larger than CLPE to avoid false alarms, improving the reliability of estimated controlled speed.

### 1.2.3 Acoustic Localization using a Passive Receiver

In order to improve the trajectory information required by the adaptive learning algorithm, we develop a new localization algorithm to more accurately estimate the trajectory of AUVs. The AUVs are equipped with acoustic receivers primarily used for monitoring tagged fish. This new acoustic method using the acoustic receivers can be a binary acoustic method in that the receivers only provide binary information. We developed odometry using flow estimation under the framework of controlled Lagrangian particle tracking (CLPT), and derived equations that govern the error growth when the vehicle is subject to flow in the framework of CLPT. We integrate acoustic detection and the odometry model to reduce localization error. We estimate vehicle position by a maximum a posteriori estimator.

Using only the binary acoustic sensor still presents substantial difficulty for localization. The binary acoustic sensor provides true measurements inside detection range or false measurements out of detection range. Detection range is the maximum distance at which the binary acoustic sensor can detect one signal at least. Because the proposed localization algorithm uses sensing regions determined by detection range to correct the inaccuracy of the odometry model, the estimated trajectory generated from the localization algorithm depends on the accuracy of detection range. Although detection range provided by manufacturers is given as a fixed value, in practice, the detection range is not fixed. Instead, it varies according to underwater environmental factors.

We identify certain underwater environmental factors to improve the accuracy of detection range based on a bioacoustic glider experiment at Gray’s Reef National Marine Sanctuary (GRNMS), located 40 nm SE of Savannah, GA. The acoustic array is specially designed to find the relationship between acoustic detections and environmental factors such as the tidal flow. Data from an acoustic Doppler current profiler (ADCP) and an underwater glider deployed in the array enables us to find which underwater environmental factors influence the correlation between acoustic detections and tidal flow. Improvement of detection range accuracy through identifying the environmental factor can increase the accuracy of the acoustic localization algorithm.

#### 1.2.4 Development of an Indoor Test Bed and Experimental Results

We verify the proposed algorithms with experimental results in an indoor test bed. The use of real AUVs potentially have a risk of vehicle loss, consuming great amounts of money and time. We developed a flying robot called the Georgia Tech Miniature Autonomous Blimp (GT-MAB), that has many similarities of AUV motion in order to repeatedly test the proposed algorithms. The GT-MAB is deployed in the indoor test bed where a wind source generates artificial flow, and motion capture cameras collect the GT-MAB trajectory in a confined space; the input to the adaptive learning algorithm is the GT-MAB trajectory. The output of the adaptive learning algorithm is the estimated flow velocity. For the verification of the adaptive learning algorithm, we evaluate the accuracy of the estimated flow velocity. One way that evaluates the accuracy of the estimated flow velocity is to compare ground truth data and estimated flow velocity. We deploy the Georgia Tech Wind Measuring Robot (GT-WMR) that measures actual flow velocity in the indoor test bed so that we compare measured and identified flows.

## CHAPTER 2

### ANOMALY DETECTION OF CONTROLLED LAGRANGIAN PARTICLES

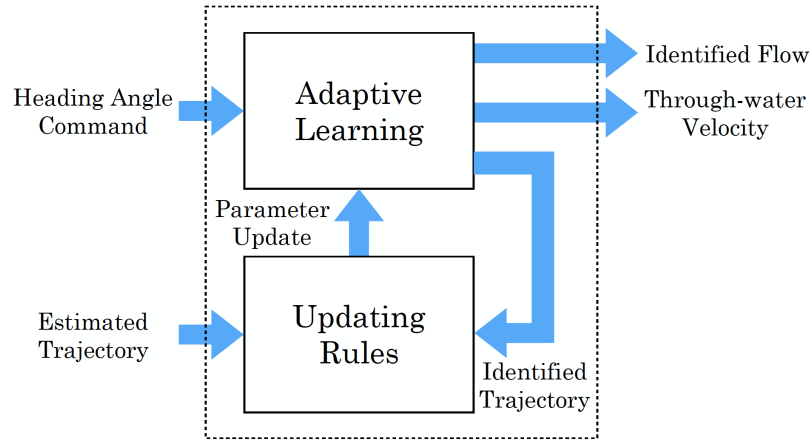
In the framework of controlled Lagrangian particle tracking, the net motion of controlled Lagrangian particles is determined by flow velocity and controlled speed. Through-water speed can be generated by thrusters and other methods of propulsion. Unlike flow velocity which cannot be controlled, the through-water speed can be controlled and hence the name of “controlled speed” arises.

Through-water speed is a critical variable that represents horizontal navigation performance of AUVs moving in a dynamic ocean environment. Positional accuracy and arrival time to target points are greatly affected by through-water speed of AUVs under ocean flow. When through-water speed is much lower than flow speed because of thruster faults, AUV motion is significantly disturbed by flow, and navigation error is increased, then arrival time is more unpredictable.

We developed an on-line adaptive learning algorithm of controlled Lagrangian particles in order to estimate the through-water speed of an AUV. Previous work [9] shows that controlled Lagrangian prediction error (CLPE) that represents the accuracy of the simulated motion models can increase over time in simulated and field experiments. Because actual flows differ from flows generated from flow models in simulated motion models, a learning algorithm is proposed to identify actual flow instead of the use of flow models. The learning algorithm, which is extended from the framework of controlled Lagrangian particles that describes the motion of AUVs partially advected by flow, simultaneously estimates both through-water speed and flow velocity.

Updating rules of the learning algorithm depends on controlled Lagrangian localization error. An estimated trajectory is one kind of path information of AUVs. Since the estimated trajectory reflects vehicle motion, we use the estimated trajectory as the input of

the updating rules in order to evaluate how adaptive learning identifies true vehicle motion, then output of adaptive learning is an identified trajectory. Controlled Lagrangian localization error (CLLE) is the difference between the estimated trajectory and the identified trajectory. When CLLE is zero, the identified trajectory generated from adaptive learning fits the estimated trajectory. Adaptive learning is updated by the updating rules that make not only estimation error of flow velocity and through-water speed converge to zero, but also CLLE converges to zero as time goes to infinity. Further, the learning algorithm guarantees bounded CLLE under uncertain disturbances. In spite of uncertainty of actual flow, the learning algorithm does not make CLLE diverge. This boundedness shows that the learning algorithm is robust to disturbances. Figure 2.1 shows the diagram of the learning algorithm composed of adaptive learning and the updating rules.



**Figure 2.1:** The diagram of the adaptive learning algorithm

Measuring through-water speed of AUVs is substantially difficult because of limited hardware capability. The inertial measurement unit (IMU) and the Doppler velocity log (DVL) can be combined to measure through-water speed. Positional accuracy of the only IMU is too low due to the double integral of the accelerometer measurements that contain various errors such as scale factor, bias, and noise [36]. IMU error can be corrected by the DVL, which measures both vehicle velocity with respect to ground (bottom-tracking), and flow velocity with respect to the vehicle (water-tracking) [37]. However, over several hun-



dred meters of water depth, the DVL's acoustic beam cannot reach to the bottom because of limited power; thus, the DVL is not operated in deep depth. Moreover, DVL accuracy can be poor when the AUV does not maintain constant altitude and heading angle. Alternatively, acoustic positioning systems can estimate AUV speed using a short baseline (SBL) or a long base line (LBL) method, which utilizes multiple beacons located at the sea floor or the hull of a ship [7]. However, this measured velocity is not through-water velocity, but it is ground velocity that combines both through-water velocity and flow velocity. Our goal is to simultaneously extract through-water velocity and flow velocity from trajectory information without any speed sensor.

We define a range of normal operation of AUV specified by maximum and minimum through-water speed when AUVs are in normal operation. If the through-water speed estimate is outside the range of normal operation, then anomalies are detected. The inaccuracy of the through-water speed estimate can occur; for example, model uncertainty used for the adaptive learning algorithm can induce estimation error of through-water speed, and then it leads to a false alarm. Because we have knowledge on flow velocity from flow models, comparing modeled flow velocity and estimated flow velocity enables avoiding the false alarm.

We describe a vehicle motion model based on the framework of controlled Lagrangian particles in the next section. The vehicle motion model incorporates a flow model and a particle model of the vehicle.

## 2.1 Vehicle Motion Model

Let  $\mathbf{F}: D \times [0, \infty] \rightarrow \mathbb{R}^2$  be a deterministic ambient flow velocity, where  $D \in \mathbb{R}^2$  is the domain of interest. Furthermore, let  $\mathbf{v}$  be the through-water velocity. Then, the vehicle motion model is approximated by

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}_R(\mathbf{x}, t) + \mathbf{v}_R(\psi_c(t)), \quad (2.1)$$

where subscript  $R$  for the flow  $\mathbf{F}$  denotes an actual flow.  $\mathbf{F}_R$  and  $\mathbf{v}_R$  are assumed to be locally Lipschitz in  $\mathbf{x} = [x_1, x_2]^\top \in D$ .  $\mathbf{x}$  is the true position of the AUV that is assumed to be known. This assumption will be removed in Chapter 5 where we discuss how to estimate  $\mathbf{x}$ . In equation (2.1),  $\psi_c(t) \in \mathbb{R}$  that represents the heading angle command of the AUV is known. We define  $\mathbf{y}(t)$  as the predicted position of the vehicle before deploying the vehicle. The set of predicted positions, or the predicted trajectory of the vehicle has following simulated motion model.

$$\frac{d\mathbf{y}}{dt} = \mathbf{F}_M(\mathbf{y}, t) + \mathbf{v}_M(\psi_c(t)), \quad (2.2)$$

where subscript  $M$  of the flow  $\mathbf{F}$  denotes a known modeled flow. After we deployed the AUVs, in the learning algorithm, adaptive learning that incorporates the heading angle command as input identifies the motion of the vehicle. Let us define  $\mathbf{z}(t)$  as the output of adaptive learning, then the identified trajectory of the vehicle is modeled by:

$$\frac{d\mathbf{z}}{dt} = \mathbf{F}_L(\mathbf{z}, t) + \mathbf{v}_L(\psi_c(t)), \quad (2.3)$$

where subscript  $L$  of the flow  $\mathbf{F}$  is to denote an identified flow.

Flow fields can be represented by spatial and temporal basis functions [38]. We consider that spatial and temporal basis functions are to be the combination of Gaussian radial and tidal basis functions, respectively. Let  $N$  be a positive integer, and  $\theta, \alpha \in \mathbb{R}^{2 \times N}$  be unknown and known parameters, respectively. Let  $\phi: D \times [0, \infty] \rightarrow \mathbb{R}^N$  be  $[\phi^1(\mathbf{x}, t), \dots, \phi^N(\mathbf{x}, t)]^\top$ .

$$\mathbf{F}_R(\mathbf{x}, t) = \theta \phi(\mathbf{x}, t) \quad (2.4)$$

$$\mathbf{F}_M(\mathbf{x}, t) = \alpha \phi(\mathbf{x}, t), \quad (2.5)$$

where

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \theta_1^1 & \cdots & \theta_1^N \\ \theta_2^1 & \cdots & \theta_2^N \end{bmatrix}, \quad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \alpha_1^1 & \cdots & \alpha_1^N \\ \alpha_2^1 & \cdots & \alpha_2^N \end{bmatrix}. \quad (2.6)$$

The combined basis functions are

$$\phi^i(\mathbf{x}, t) = \exp^{-\frac{\|\mathbf{x} - \mathbf{c}_i\|}{2\sigma_i}} \cos(\omega_i t + v_i), \quad i = 1, \dots, N, \quad (2.7)$$

where  $\mathbf{c}_i$  is the centers,  $\sigma_i$  is the widths,  $\omega_i$  tidal frequencies, and  $v_i$  tidal phases. Here we assume that the flow only contains tidal flow and biased flow, which represents the combination of high and low frequency components of flow. If  $\omega_i$  equals zero, flow only has spatial variability. Similarly,  $\mathbf{F}_L(\mathbf{z}, t)$  is defined as follows.

$$\mathbf{F}_L(\mathbf{z}, t) = \xi(t)\phi(\mathbf{z}, t), \quad (2.8)$$

$$\text{where } \xi(t) = \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix} = \begin{bmatrix} \xi_1^1(t) & \cdots & \xi_1^N(t) \\ \xi_2^1(t) & \cdots & \xi_2^N(t) \end{bmatrix}.$$

The through-water velocity of the AUV can be represented by the combination of AUV through-water speed and heading angle command as the general particle model, which is known as the unicycle model of unmanned ground robots. Let actual through-water speed  $V_R$  be a constant value, and let identified through-water speed  $V_L$  be a time-varying parameter. Let  $\Psi_c = [\cos \psi_c(t), \sin \psi_c(t)]^\top$  be the vector of heading angle command. Let  $\beta(t) \in \mathbb{R}^2$  be a learning injection parameter. Then,

$$\mathbf{v}_R(\psi_c(t)) = V_R \Psi_c \quad (2.9)$$

$$\mathbf{v}_L(\psi_c(t)) = V_L(t) \Psi_c + \beta(t). \quad (2.10)$$

For normal operation,  $V_R$  in equation (2.9) is a constant. However, the predicted and estimated trajectories diverge when  $V_R$  has a different abnormal constant value because of

unexpected AUV motion. To detect such abnormal constant values, the learning algorithm identifies through-water velocity represented by equation (2.10) that contains time-varying speed term  $V_L(t)$ , and learning injection term  $\beta(t)$ . By plugging equations (2.10) and (2.8) into equation (2.3), the closed-loop dynamics for the generation of the identified trajectory is

$$\dot{\mathbf{z}} = \xi(t)\phi(\mathbf{z}, t) + V_L(t)\Psi_c + \beta(t). \quad (2.11)$$

Meanwhile, the closed-loop dynamics for the generation of the estimated trajectory is

$$\dot{\mathbf{x}} = \theta\phi(\mathbf{x}, t) + V_R\Psi_c, \quad (2.12)$$

when we plug equations (2.4) and (2.9) into equation (2.1). If  $\mathbf{x}$  and  $\mathbf{z}$  have the same initial condition and  $V_L = V_R$ , and  $\beta(t) = \theta\phi(\mathbf{x}, t) - \xi(t)\phi(\mathbf{z}, t)$ , the identified trajectory is identical to the estimated trajectory. Our goal is to design updating rules and the learning injection parameter so that the two closed-loop dynamics can be similar. Note that the parameter  $\theta$  is unknown so we can not use  $\beta(t) = \theta\phi(\mathbf{x}, t) - \xi(t)\phi(\mathbf{z}, t)$  directly.

On the other hand, through-water velocity can be saturated because of control power constraints of AUVs. We modify equations (2.1) as follows:

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}_R(\mathbf{x}, t) + \mathbf{u}(\psi_c(t)), \quad (2.13)$$

where

$$\mathbf{u}(\psi_c(t)) = \begin{cases} \mathbf{v}_R(\psi_c(t)) & \text{if } \|\mathbf{v}_R(\psi_c(t))\|_\infty \leq u_0 \\ u_0 \text{sgn}(\mathbf{v}_R(\psi_c(t))) & \text{if } \|\mathbf{v}_R(\psi_c(t))\|_\infty > u_0. \end{cases} \quad (2.14)$$

The maximum through-water speed  $u_0$  is determined by the hardware configuration of AUVs. Our goal is to design updating rules of the time-varying parameters in equation (2.10) for AUVs with saturated through-water velocity so that the updating rules make the identified trajectory follows the estimated trajectory in the ocean flow field.

## 2.2 Controlled Lagrangian Localization Error

We first derive controlled Lagrangian localization error (CLLE) dynamics that models how much the estimated trajectory is deviated from the identified trajectory. By subtracting equation (2.11) from equation (2.12), CLLE dynamics is represented by

$$\dot{\mathbf{e}} = \dot{\mathbf{x}} - \dot{\mathbf{z}} = \theta\phi(\mathbf{x}, t) - \xi(t)\phi(\mathbf{z}, t) + (V_R - V_L(t))\Psi_c - \beta(t). \quad (2.15)$$

For example, if we let  $\beta(t) = \xi(t)\phi(\mathbf{x}, t) - \xi(t)\phi(\mathbf{z}, t) + K\mathbf{e}$ , where  $K$  is a diagonal matrix with positive components,  $V_L(t)$  converges to  $V_R$ , and  $\xi(t)$  converges to  $\theta$ , CLLE goes to zero as time goes by, which implies that the identified trajectory follows the estimated trajectory. Then we design the learning parameter injection as follows:

$$\beta(t) = \xi(t)\phi(\mathbf{x}, t) - \xi(t)\phi(\mathbf{z}, t) + K\mathbf{e}. \quad (2.16)$$

When we plug equation (2.16) into equation (2.15), CLLE dynamics is

$$\dot{\mathbf{e}} = (\theta - \xi(t))\phi(\mathbf{x}, t) + (V_R - V_L(t))\Psi_c - K\mathbf{e}. \quad (2.17)$$

We derive CLLE dynamics under control input constraints. Let  $\delta\mathbf{u} = \mathbf{u}(\psi_c(t)) - \mathbf{v}_R(\psi_c(t))$  be the saturation term of through-water velocity. We combine equations (2.13) and (2.14) using (2.9), and subtract (2.11) from the combination of (2.13) and (2.14). CLLE dynamics becomes

$$\dot{\mathbf{e}} = (\theta - \xi(t))\phi(\mathbf{x}, t) + (V_R - V_L(t))\Psi_c - K\mathbf{e} + \delta\mathbf{u}, \quad (2.18)$$

where  $\delta\mathbf{u}$  can be viewed as an additional disturbance to CLLE dynamics. We assume that  $\|\delta\mathbf{u}\|$  is bounded by  $\delta u_{max}$ . Equation (2.18) is used for the proposed updating rules that ensure CLLE to be ultimately bounded in spite of saturation of through-water velocity.

### 2.3 Adaptive Learning Algorithm

Our goal is to design the learning algorithm that updates parameters  $\xi(t)$  and  $V_L(t)$  with the updating rules by using CLLE dynamics so that CLLE converges to zero. Let  $\bar{\xi}$ ,  $\bar{\theta}$ , and  $\mathbf{e} \otimes \phi \in \mathbb{R}^{2N}$  be row vectors. That is,  $\bar{\xi}(t) = [\xi_1^1(t), \dots, \xi_1^N(t), \xi_2^1(t), \dots, \xi_2^N(t)]^\top$ ,  $\bar{\theta}(t) = [\theta_1^1(t), \dots, \theta_1^N(t), \theta_2^1(t), \dots, \theta_2^N(t)]^\top$ , and  $\mathbf{e} \otimes \phi = [e_1\phi^1, \dots, e_1\phi^N, e_2\phi^1, \dots, e_2\phi^N]^\top$ , where  $\otimes$  is the Kronecker product. We design the updating rules for time-varying parameters as follows:

$$\dot{\bar{\xi}}(t) = \bar{\gamma} \mathbf{e} \otimes \phi(\mathbf{x}, t) \quad (2.19)$$

$$\dot{V}_L(t) = \bar{\gamma} \mathbf{e}^\top \Psi_c. \quad (2.20)$$

To prove error convergence, parameter convergence, and robustness of equations (2.19) and (2.20), we need Theorems and Lemmas in [39] as follows: Let  $A(t) \in \mathbb{R}^{n \times n}$ ,  $C(t), L(t) \in \mathbb{R}^{n \times l}$ ,  $X(t) \in \mathbb{R}^{n \times 1}$ , and  $Y(t) \in \mathbb{R}^{l \times 1}$  be matrices that satisfy the following equation.

$$\dot{X}(t) = A(t)X(t), \quad Y(t) = C^\top(t)X(t). \quad (2.21)$$

*Theorem 2.1* A necessary and sufficient condition for the uniformly asymptotically stability of the equilibrium of  $\dot{X}(t) = A(t)X(t)$  is that there exists a symmetric matrix  $P(t)$  such that both  $c_1 I \leq P(t) \leq c_2 I$  and  $A(t)^\top P(t) + P(t)A(t) + \dot{P}(t) + \nu C(t)^\top C(t) \leq 0$  are satisfied  $\forall t$  and some constant  $\nu > 0$ , where  $c_1 > 0$ , and  $c_2 > 0$  and  $C(t)$  is such that  $(C(t), A(t))$  is uniformly completely observable.

*Definition 2.1* [40, 41] A vector signal  $u$  is persistently exciting if there exist positive constants  $\kappa_1$ ,  $\kappa_2$ , and  $T$  such that  $\kappa_2 I \geq \int_t^{t+T} u(\tau)u^\top(\tau)d\tau \geq \kappa_1 I \forall t$ .

*Lemma 2.1* Assume that there exists constants  $\nu > 0, k_\nu \geq 0$  such that for all  $t_0 \geq 0$ ,  $L(t)$  satisfies the inequality  $\int_{t_0}^{t_0+\nu} \|L(\tau)\|^2 d\tau \leq k_\nu$ . Then system  $(C(t), A(t))$  is a uniformly completely observable if and only if system  $(C(t), A(t) + L(t)C(t)^\top)$  is a uniformly completely

observable.

*Lemma 2.2* If  $u : \mathbb{R}^+ \mapsto \mathbb{R}^n$  is persistently exciting,  $u \in L_\infty$ ,  $\dot{u} \in L_\infty$ , and  $H(s)$  is a stable, minimum phase, proper rational transfer function, then  $u' = H(s)u$  is persistently exciting.

*Lemma 2.3* Consider system  $\dot{Y}_1 = A_c Y_1 - B_c \phi^\top Y_2$ ,  $\dot{Y}_2 = 0$ , and  $y_0 = C_c^\top Y_1$ , where  $A_c$  is a stable matrix,  $(C_c, A_c)$  is observable, and  $\phi \in L_\infty$ . If  $\phi_f$  defined as  $\phi_f \triangleq C_c^\top (sI - A_c)^{-1} B_c \phi$  satisfies  $\alpha_1 I \leq \frac{1}{T_0} \int_t^{t+T_0} \phi_f(\tau) \phi_f^\top(\tau) d\tau \leq \alpha_2 I, \forall t \geq 0$  for constants  $\alpha_1, \alpha_2, T_0 > 0$ , then the system is uniformly completely observable.

*Lemma 2.4* [42, 43] If  $g$  is a real function of real variable  $t$ , defined and uniformly continuous for  $t \geq 0$ , and if the limit of the integral  $\int_0^t g(s) ds$  as  $t$  tends to infinity exists and is a finite number, then  $\lim_{t \rightarrow \infty} g(t) = 0$ .

Let  $\tilde{\phi}_1 = \begin{bmatrix} \phi_1^1 & \cdots & \phi_1^N \\ 0 & \cdots & 0 \end{bmatrix}$  and  $\tilde{\phi}_2 = \begin{bmatrix} 0 & \cdots & 0 \\ \phi_2^1 & \cdots & \phi_2^N \end{bmatrix}$  be in  $\mathbb{R}^{2 \times N}$ . Let  $w = [\tilde{\phi}_1, \tilde{\phi}_2, \Psi_c]^\top \in \mathbb{R}^{(2N+1) \times 2}$ . For parameter convergence, we need an assumption on  $w$  as follows:

*Assumption 2.1*  $w$  is persistently exciting. By Definition 2.1, there exists positive definite matrix  $W(t) \in \mathbb{R}^{(2N+1) \times (2N+1)}$  such that

$$W(t) = \int_t^{t+T} \begin{bmatrix} \phi_1^1 \phi_1^1 & \phi_1^1 \phi_1^2 & \cdots & \phi_1^1 \phi_1^N & 0 & 0 & \cdots & 0 & \phi_1^1 \cos \psi_c \\ \phi_1^2 \phi_1^1 & \phi_1^2 \phi_1^2 & \cdots & \phi_1^2 \phi_1^N & 0 & 0 & \cdots & 0 & \phi_1^2 \cos \psi_c \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_1^N \phi_1^1 & \phi_1^N \phi_1^2 & \cdots & \phi_1^N \phi_1^N & 0 & 0 & \cdots & 0 & \phi_1^N \cos \psi_c \\ 0 & 0 & \cdots & 0 & \phi_2^1 \phi_2^1 & \phi_2^1 \phi_2^2 & \cdots & \phi_2^1 \phi_2^N & \phi_2^1 \sin \psi_c \\ 0 & 0 & \cdots & 0 & \phi_2^2 \phi_2^1 & \phi_2^2 \phi_2^2 & \cdots & \phi_2^2 \phi_2^N & \phi_2^2 \sin \psi_c \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \phi_2^N \phi_2^1 & \phi_2^N \phi_2^2 & \cdots & \phi_2^N \phi_2^N & \phi_2^N \sin \psi_c \\ \cos \psi_c \phi_1^1 & \cos \psi_c \phi_1^2 & \cdots & \cos \psi_c \phi_1^N & \sin \psi_c \phi_2^1 & \sin \psi_c \phi_2^2 & \cdots & \sin \psi_c \phi_2^N & 1 \end{bmatrix} d\tau, \quad (2.22)$$

where  $\phi_j^i(\mathbf{x}, \tau) = \exp^{-\frac{\|\mathbf{x} - \mathbf{c}_j\|}{2\sigma_i}} \cos(\omega_i \tau + v_i), i = 1 \cdots N, j = 1, 2$ , and  $T > 0$ . This assumption is critical to prove the convergence of parameters in that estimation error of parameters cannot converge to zero when the persistent excitation condition is not satisfied [44]. For

example, when  $\phi_j^i$  is constant and  $\psi_c$  equals 90 degree,  $W(t)$  equals

$$T \begin{bmatrix} \phi_1^1 \phi_1^1 & \phi_1^1 \phi_1^2 & \cdots & \phi_1^1 \phi_1^N & 0 & 0 & \cdots & 0 & 0 \\ \phi_1^2 \phi_1^1 & \phi_1^2 \phi_1^2 & \cdots & \phi_1^2 \phi_1^N & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_1^N \phi_1^1 & \phi_1^N \phi_1^2 & \cdots & \phi_1^N \phi_1^N & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & \phi_2^1 \phi_2^1 & \phi_2^1 \phi_2^2 & \cdots & \phi_2^1 \phi_2^N & \phi_2^1 \\ 0 & 0 & \cdots & 0 & \phi_2^2 \phi_2^1 & \phi_2^2 \phi_2^2 & \cdots & \phi_2^2 \phi_2^N & \phi_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \phi_2^N \phi_2^1 & \phi_2^N \phi_2^2 & \cdots & \phi_2^N \phi_2^N & \phi_2^N \\ 0 & 0 & \cdots & 0 & \phi_2^1 & \phi_2^2 & \cdots & \phi_2^N & 1 \end{bmatrix}. \quad (2.23)$$

Then, this  $W(t)$  is singular due to the last row depending on the other rows with elements composed of  $\phi_2^i$ ; hence  $w$  is not persistently exciting. Because constant  $\phi_j^i$  and  $\psi_c$  are not enough to excite system modes so that unknown parameters are identified, estimation errors of parameters cannot converge to zero.

We prove error and parameter convergence to show that the learning algorithm accurately identifies a vehicle motion under flow with the proposed updating rules. Error convergence indicates that the identified trajectory converges to the estimated trajectory, which implies that the learning algorithm identifies the vehicle motion. The convergence of CLLE using Lemma 2.4 is proved as follows.

*Theorem 2.2* Using equations (2.19) and (2.20), CLLE converges to zero when time goes to infinity; that is,  $\mathbf{e}(t) \rightarrow \vec{0}$  as  $t \rightarrow \infty$ .

*Proof.* Consider a candidate Lyapunov function:

$$V(\mathbf{e}, \xi, V_L) = \frac{1}{2} \left\{ \mathbf{e}^\top \mathbf{e} + \frac{1}{\bar{\gamma}} (\bar{\theta} - \bar{\xi}(t))^\top (\bar{\theta} - \bar{\xi}(t)) + \frac{1}{\bar{\gamma}} (V_R - V_L(t))^2 \right\}. \quad (2.24)$$



The derivative of  $V$  is

$$\dot{V} = -\mathbf{e}^T K \mathbf{e} + \mathbf{e}^T (\theta - \xi(t)) \phi(\mathbf{x}, t) + (V_R - V_L(t)) \left( \mathbf{e}^T \Psi_c - \frac{1}{\gamma} \dot{V}_L(t) \right) - \frac{1}{\gamma} (\bar{\theta} - \bar{\xi}(t)) \dot{\xi}(t). \quad (2.25)$$

We know  $\mathbf{e}^T (\theta - \xi(t)) \phi(\mathbf{x}, t) = (\bar{\theta} - \bar{\xi}(t)) \mathbf{e} \otimes \phi(\mathbf{x}, t)$ . Then, using equation (2.19) and (2.20),

$$\dot{V} = -\mathbf{e}^T K \mathbf{e} \leq 0. \quad (2.26)$$

$\dot{V}$  is negative semi-definite and this implies  $\mathbf{e}$ ,  $\xi(t)$ , and  $V_L(t)$  are bounded. In addition, the second order time derivative of  $V$  satisfies

$$\ddot{V} = -2\mathbf{e}^T K \dot{\mathbf{e}} = -2\mathbf{e}^T K \{ (\theta - \xi(t)) \phi(\mathbf{x}, t) + (V_R - V_L(t)) \Psi_c - K \mathbf{e} \}. \quad (2.27)$$

Because  $\Psi_c$  is bounded,  $\ddot{V}$  is bounded, and hence  $\dot{V}$  is uniformly continuous. By Lemma 2.4,  $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$ . Since  $K$  is the diagonal matrix,  $\mathbf{e}(t) \rightarrow \vec{0}$  as  $t \rightarrow \infty$ .  $\square$

Even if CLLE convergence is shown, the learning algorithm may not identify actual flow because multiple parameters that represent flow are identified from one type of information, which is the estimated trajectory. Thus, we prove parameter convergence to declare that the vehicle motion is accurately identified.

*Theorem 2.3* Under the same setting of Theorem 2.2,  $\bar{\xi}(t)$  and  $V_L(t)$  converges to  $\bar{\theta}$  and  $V_R$ , respectively; that is,  $\bar{\xi}(t) \rightarrow \bar{\theta}$ , and  $V_L(t) \rightarrow V_R$  as  $t \rightarrow \infty$ .

*Proof.* Let  $\eta_1$ ,  $\eta_2$  and  $\eta_3$  be  $(\theta_1 - \xi_1(t))$ ,  $(\theta_2 - \xi_2(t))$ , and  $(V_R - V_L(t))$ , respectively. We rewrite equation (2.17) using equation  $\eta_1$ ,  $\eta_2$ , and  $\eta_3$  as follows:

$$\dot{\mathbf{e}} = \tilde{\phi}_1(\mathbf{x}, t) \eta_1 + \tilde{\phi}_2(\mathbf{x}, t) \eta_2 + \Psi_c \eta_3 - K \mathbf{e}. \quad (2.28)$$

We augment  $\mathbf{e}$ ,  $\eta_1$ ,  $\eta_2$ , and  $\eta_3$  to new state variable  $X$ . Then

$$\dot{X} = A(t)X, \quad Y = CX, \quad A(t) = \begin{bmatrix} -K & \tilde{\phi}_1 & \tilde{\phi}_2 & \Psi_c \\ -\tilde{\gamma}\tilde{\phi}_1 & 0 & 0 & 0 \\ -\tilde{\gamma}\tilde{\phi}_2 & 0 & 0 & 0 \\ -\tilde{\gamma}\Psi_c^\top & 0 & 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.29)$$

where 0 is the zero matrix with proper dimensions according to the components of the first row matrix of  $A$ . Our goal is to show that the origin of  $\dot{X} = A(t)X$  is uniformly asymptotically stable, which implies that  $\tilde{\xi}(t)$  converges to  $\bar{\theta}$ , and  $V_L(t)$  converges to  $V_R$  when time goes to infinity. By Theorem 2.1, we need to show that  $P$  exists and  $(C, A)$  is uniformly completely observable. Let

$$P = \begin{bmatrix} \frac{1}{2}K^{-1} & 0 & 0 & 0 \\ 0 & \frac{1}{2\tilde{\gamma}}K^{-1} & 0 & 0 \\ 0 & 0 & \frac{1}{2\tilde{\gamma}}K^{-1} & 0 \\ 0 & 0 & 0 & \frac{1}{2\tilde{\gamma}}K^{-1} \end{bmatrix}. \quad (2.30)$$

Let  $V'$  be  $X^\top PX$ . Then,  $\dot{V}' = X^\top (A^\top P + P^\top A + \dot{P})X \leq -\nu X^\top C^\top CX = -\nu \|Y\|^2$ , where  $\dot{P} = 0$ . Thus, there exists a symmetric matrix  $P(t)$  such that  $c_1 I \leq P(t) \leq c_2 I$  and  $A(t)^\top P(t) + P(t)A(t) + \dot{P}(t) + \nu C(t)^\top C(t) \leq 0$ . Now we will prove  $(C, A)$  is a uniformly completely observable. Because it is hard to prove the observability of time varying system matrix  $A$ , we will instead show  $(C, A + LC)$  is uniformly completely observable with some bounded

matrix  $L$ , called output injection by Lemma 3.1. Let  $L = \begin{bmatrix} K & 0 & 0 & 0 \\ \tilde{\gamma}\tilde{\phi}_1 & 0 & 0 & 0 \\ \tilde{\gamma}\tilde{\phi}_2 & 0 & 0 & 0 \\ \tilde{\gamma}\Psi_c^\top & 0 & 0 & 0 \end{bmatrix}$ . Since  $\Psi_c$  is

bounded, and  $\tilde{\phi}$  is a sinusoidal function with exponential magnitude,  $L$  is bounded. Then,

$$A + LC = \begin{bmatrix} 0 & \tilde{\phi}_1 & \tilde{\phi}_2 & \Psi_c \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \text{ Thus,}$$

$$\begin{aligned} \dot{X} &= AX = (A + LC)X - LY \\ Y &= CX. \end{aligned} \tag{2.31}$$

Let  $\eta = [\eta_1, \eta_2, \eta_3]^\top$ . We have the following equation corresponding to equation (2.31).

$$\begin{aligned} \dot{\mathbf{e}} &= -K\mathbf{e} + w^\top \eta \\ \dot{\eta} &= 0 \\ Y &= \mathbf{e}. \end{aligned} \tag{2.32}$$

By Assumption 2.1,  $w$  is persistently exciting. Let  $\Phi(\tau) = \int_t^\tau \exp^{-K(\tau-\sigma)} w(\sigma) d\sigma$  be output of equation (2.32) given input  $w$ . By Lemma 2.2,  $\Phi(\tau)$  satisfies persistently exciting conditions because  $w(\sigma)$  is persistently exciting, and the transfer function of of equation (2.32),  $(sI_{2 \times 2} + K)^{-1}$ , is a stable, minimum phase, proper rational transfer function. Therefore, there exists constant  $\rho_1, \rho_2, T_0 > 0$  such that  $\rho_2 I \geq \frac{1}{T_0} \int_t^{t+T_0} \Phi(\tau) \Phi^\top(\tau) d\tau \geq \rho_1 I \forall t \geq 0$ . By applying Lemma 2.1 to the system of equation (2.32),  $(C, A + LC)$  is uniformly completely observable; hence, the system of equation (2.29) is uniformly completely observable. Therefore, the origin of  $\dot{X} = A(t)X$  is uniformly asymptotically stable; that is  $X \rightarrow \vec{0}$  as  $t \rightarrow \infty$ . This means that  $\eta_1, \eta_2$ , and  $\eta_3$  go to zeros, individually. Thus,  $\bar{\xi}(t)$  and  $V_L(t)$  converge to  $\bar{\theta}$  and  $V_R$ , respectively.  $\square$

### 2.3.1 Input constraints

The AUVs have limited power to control their motions. The control power is saturated by the maximum capacity of hardware such as motors and thrusters. This induces constraint

to controlling AUVs in the ocean.

Since equation (2.18) includes one saturated term represented by  $\delta \mathbf{u} = \mathbf{u} - \mathbf{v}_R$ , which shows the discrepancy between input and output of the saturator, we reject additional disturbance  $\delta \mathbf{u}$  from saturation in equation (2.18) by using a scheme in [45]. We generate additional signal  $\mathbf{e}_\delta$  governed by a differential equation as follows:

$$\dot{\mathbf{e}}_\delta = -K\mathbf{e}_\delta + \Lambda\delta\mathbf{u}, \quad (2.33)$$

where  $\Lambda = \text{diag}\{\Lambda_1, \Lambda_2\}$  is the matrix with parameters that we design. Those parameters are determined to reject the disturbance. Let  $\boldsymbol{\varepsilon} = \mathbf{e} - \mathbf{e}_\delta$  be the difference between CLLE and the additional signal. When subtracting (2.33) from (2.18), we have

$$\dot{\boldsymbol{\varepsilon}} = \dot{\mathbf{e}} - \dot{\mathbf{e}}_\delta = (\boldsymbol{\theta} - \boldsymbol{\xi}(t))\boldsymbol{\phi}(\mathbf{x}, t) + (V_R - V_L(t))\boldsymbol{\Psi}_c - K\mathbf{e} + (I - \Lambda)\delta\mathbf{u}. \quad (2.34)$$

$\Lambda$  is designed to be  $I$ . Let  $\delta\mathbf{u} = [\delta u_1, \delta u_2]^\top$  be two dimensional vectors. To make  $\boldsymbol{\varepsilon}$  go to zero, we design the updating rules for time-varying parameters  $\bar{\boldsymbol{\xi}}$  and  $V_L$  by the following equations.

$$\dot{\bar{\boldsymbol{\xi}}} = \bar{\gamma}\boldsymbol{\varepsilon} \otimes \boldsymbol{\phi}(\mathbf{x}, t) \quad (2.35)$$

$$\dot{V}_L = \bar{\gamma}\tilde{\boldsymbol{\Psi}}_c\boldsymbol{\varepsilon} \quad (2.36)$$

*Theorem 2.4* Under the update rules (2.35) and (2.36), CLLE is ultimately bounded.

$$\|\mathbf{e}\| \leq \frac{\lambda_{\max}(\Lambda)\|\delta\mathbf{u}\|}{\bar{\xi}}, \quad (2.37)$$

where the positive constant  $\bar{\xi} < 1$ .

*Proof.* Let  $V_{\mathbf{e}_\delta} = \frac{1}{2}\mathbf{e}_\delta^\top K^{-1}\mathbf{e}_\delta$ . The derivative of  $V_{\mathbf{e}_\delta}$  is  $\dot{V}_{\mathbf{e}_\delta} = -\mathbf{e}_\delta^\top \mathbf{e}_\delta + \mathbf{e}_\delta^\top \Lambda \delta \mathbf{u}$ . Then,  $\dot{V}_{\mathbf{e}_\delta} \leq -(1-\xi)\|\mathbf{e}_\delta\|^2 - \xi\|\mathbf{e}_\delta\|^2 + \|\mathbf{e}_\delta\|\lambda_{\max}(\Lambda)\|\delta \mathbf{u}\|$ . When  $\|\mathbf{e}_\delta\| \geq \frac{\lambda_{\max}(\Lambda)\|\delta \mathbf{u}\|}{\xi}$  given positive constant  $\xi < 1$ ,  $\dot{V} \leq -(1-\xi)\|\mathbf{e}_\delta\|^2$ . This means  $\dot{V}$  is not positive. Thus,  $\|\mathbf{e}_\delta\| \leq \frac{\lambda_{\max}(\Lambda)\|\delta \mathbf{u}\|}{\xi}$ .

We will show  $\varepsilon$  goes to zero when time goes infinity by the following candidate Lyapunov function.

$$V(\varepsilon, \bar{\xi}, V_L) = \frac{1}{2} \left\{ \varepsilon^\top \varepsilon + \frac{1}{\gamma} (\bar{\theta} - \bar{\xi})^\top (\bar{\theta} - \bar{\xi}) + \frac{1}{\gamma} (V_R - V_L)^2 \right\}, \quad (2.38)$$

By using equations (2.35) and (2.36),  $\dot{V} = -\varepsilon^\top K \varepsilon \leq 0$ .  $\dot{V}$  is negative semi-definite and this implies  $\varepsilon$ ,  $\bar{\xi}$ , and  $V_L$  are bounded. In addition,  $\dot{V} = -2\varepsilon^\top K \dot{\varepsilon} = -2\varepsilon^\top K \{(\theta - \xi)\phi(\mathbf{x}, t) + (V_R - V_L)\Psi_c\}$ . Since  $\mathbf{e} = \mathbf{e}_\delta + \varepsilon$ ,  $\mathbf{e}$  is bounded. This implies that  $\mathbf{x}$  is bounded. In addition,  $\xi$  and  $V_L$  are bounded. Thus,  $\dot{V}$  is bounded, and then  $\dot{V}$  is uniformly continuous. By Lemma 3.4,  $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$ . Since  $K$  is the diagonal matrix,  $\varepsilon \rightarrow \vec{0}$  when  $t \rightarrow \infty$ ;  $\mathbf{e} \rightarrow \mathbf{e}_\delta$  when  $t \rightarrow \infty$ . Thus, CLLE is ultimately bounded.  $\square$

### 2.3.2 Inaccuracy in flow modeling

Although the basis functions well capture the spatial variability of actual flows in a specific region, the functions still include deterministic errors induced by the variability out of the region. In this section, we address the robustness of the proposed adaptive learning algorithm.

To show that the proposed algorithm is robust to disturbance in the flows, we prove the boundedness of CLLE when the actual flow model has unknown disturbances. We assume  $\mathbf{F}_R(\mathbf{x}, t) = \theta \phi(\mathbf{x}, t) + \Delta$ , where  $\|\Delta\|$  is bounded by  $\Delta_{\max} \in \mathbb{R}$ . Then,

$$\dot{\mathbf{e}} = (\theta - \xi(t))\phi(\mathbf{x}, t) + (V_R - V_L(t))\Psi_c - K\mathbf{e} + \Delta \quad (2.39)$$

The Theorem of robustness is proved below.

*Theorem 2.5* Under the same setting of Theorem 2.2 and  $\mathbf{F}_R(\mathbf{x}, t) = \theta\phi(\mathbf{x}, t) + \Delta$ , CLLE is ultimately bounded.

$$\|\mathbf{e}\| \leq \frac{1}{v}\|\Delta\|, \quad (2.40)$$

where the positive constant  $v < \lambda_{\min}(K)$  and  $\|\Delta\|$  is bounded by  $\Delta_{\max}$ .

*Proof.* Let  $V$  be the Lyapunov function represented by equation (2.24). By using equation (2.39), the derivate of  $V$  is

$$\begin{aligned} \dot{V} = & -\mathbf{e}^T K \mathbf{e} + \mathbf{e}^T \Delta + (V_R - V_L(t))^T \left( \frac{1}{\bar{\gamma}} \dot{V}_L(t) - \mathbf{e}^T \Psi_c \right) \\ & + \frac{1}{\bar{\gamma}} (\bar{\theta} - \bar{\xi}(t)) \left( \frac{1}{\bar{\gamma}} \dot{\bar{\xi}}(t) - \mathbf{e} \otimes \phi(\mathbf{x}, t) \right) \end{aligned}$$

Then, we plug the updating rules represented by equations (2.19) and (2.20) into equation (2.41). Then,

$$\begin{aligned} \dot{V} = & -\mathbf{e}^T K \mathbf{e} + \mathbf{e}^T \Delta \leq -\lambda_{\min}(K) \mathbf{e}^T \mathbf{e} + \mathbf{e}^T \Delta \leq -\lambda_{\min}(K) \|\mathbf{e}\|^2 + \|\mathbf{e}\| \|\Delta\| \\ \leq & -(\lambda_{\min}(K) - v) \|\mathbf{e}\|^2 + \|\mathbf{e}\| \|\Delta\| - v \|\mathbf{e}\|^2 \end{aligned} \quad (2.41)$$

When  $\|\mathbf{e}\| \geq \frac{1}{v}\|\Delta\|$  given positive constant  $v < \lambda_{\min}(K)$ ,  $\dot{V} \leq -(\lambda_{\min}(K) - v) \|\mathbf{e}\|^2$ , which means  $\dot{V}$  is negative definite. Thus, CLLE is ultimately bounded. The bound of CLLE is  $\|\mathbf{e}\| \leq \frac{1}{v}\|\Delta\|$ .  $\square$

## 2.4 Anomaly Detection Algorithm

From Theorems 2.2-2.5, we prove error and parameter convergence of the learning algorithm. With the updating rules represented by equations (2.19) and (2.20), we estimate flow velocity and through-water speed simultaneously. The through-water speed estimate is used for a critical measure that decides whether or not abnormal vehicle motion occurs. Generally, we know maximum and minimum through-water speed when AUVs are in nor-

mal operation. If through-water speed estimate is within the range between maximum and minimum through-water speed, we determine that the AUV is normally operated without abnormal motion; however, we determine that the abnormal motion of the AUV happens when AUV through-water speed estimate is out of the normal range. The binary decision rule based on through-water speed estimate can wrongly inform that anomaly occurs. Even if actual through-water speed is within the normal range, through-water speed estimate can be out of the normal range due to large estimation error of through-water speed; this is called a false alarm. Such a false alarm may stop the vehicle mission.

In order to prevent the false alarm, we propose that flow velocity estimate is used as an extra information that can validate through-water speed estimate. Since the adaptive learning algorithm makes the identified trajectory match the estimated trajectory while estimating both flow velocity and through-water speed, large error of through-water speed estimate implies large error of flow velocity estimate, and vice versa. Thus, we define flow estimation error as the difference between estimated flow velocity and modeled flow velocity generated from flow models (ex. [46, 47]) available. To determine that flow estimation error is large or small by a threshold technique, we use Euclidean norm of normalized flow estimation error, which is the norm of the ratio of flow estimation error to modeled flow. Since the ratio describes how much estimated flow is different from modeled flow, we are able to compare the ratio and a predetermined threshold. When the ratio is below the threshold, flow estimation error is evaluated to be small; otherwise, flow estimation error is evaluated to be large. If the ratio is above the threshold, it is highly possible that the binary decision rule is wrong; on the other hand, the binary decision is reliable when the ratio is below the threshold. However, using the norm of the ratio has one disadvantage. Flow model with spatial and time variabilities can have flow speed close to zero. This low flow speed can cause the large value of the norm of the ratio; we cannot use a fixed threshold to determine that flow estimation error is small or large. Instead of modeled flow used for the denominator of the ratio, we compare maximum values of estimated and modeled flows.

Let  $\hat{F}_{L_{max}} = \max(\|\mathbf{F}_L(\tau)\|_{\tau \in [0,t]})$  be the maximum value of estimated flow speed until time  $t$ . Let  $\hat{F}_{M_{max}} = \max(\|\mathbf{F}_M(\tau)\|_{\tau \in [0,t]})$  be the the maximum value of modeled flow speed until time  $t$ . The measure is proposed by

$$p_E = \frac{\|\mathbf{F}_M(t) - \mathbf{F}_L(t)\|}{2\max(\hat{F}_{L_{max}}, \hat{F}_{M_{max}})}. \quad (2.42)$$

The numerator of the measure in equation (2.42) represents the difference between estimated and modeled flows. The denominator of the measure normalizes the numerator by a type of maximum flow. When we compare the maximum value of estimated flow speed until time  $t$  to that of modeled flow speed until time  $t$ , we select the larger value between the two maximum values in order to avoid numerator near zero. The value 2 in the denominator is a scale factor that make measure be 1 when the difference between estimated and modeled flows is maximum. The prevention scheme including the proposed measure  $p_E$  is combined with the binary decision rule, which describes in the form of pseudocodes in Algorithm 1.

---

**Algorithm 1:** Anomaly Detection Algorithm

---

**Input:** Flow velocity estimate  $F_L(t)$ , modeled flow estimate  $F_M(t)$ , false alarm factor  $\gamma_f$ , controlled speed estimate  $V_L(t)$ , maximum speed  $V_{\max}$ , minimum speed  $V_{\min}$

**Output:** Anomaly detection flag

- 1  $\hat{F}_{L_{max}} = \max(F_L(\tau)_{\tau \in [0,t]})$
- 2  $\hat{F}_{M_{max}} = \max(F_M(\tau)_{\tau \in [0,t]})$
- 3  $p_E = \frac{\|F_L(t) - F_M(t)\|}{2\max(\hat{F}_{L_{max}}, \hat{F}_{M_{max}})}$
- 4 **if**  $p_E > \gamma_f$  **then**
- 5     | flag = 2  $\triangleright$  False Alarm
- 6 **else if**  $V_L(t) > V_{\max}$  **or**  $V_L(t) < V_{\min}$  **then**
- 7     | flag = 1  $\triangleright$  Anomaly Detected
- 8 **else**
- 9     | flag = 0  $\triangleright$  No Anomaly Detected
- 10 **end**

---



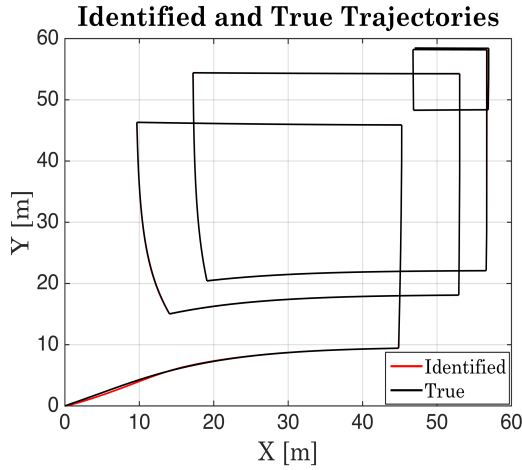
## 2.5 Mathematical Simulations of Anomaly Detection Algorithms

This section describes simulation results for the anomaly detection algorithms in Section 2.4. For the presentation of 2D ocean flow,  $\theta_1 = [0.9 \ 0.5 \ 0.7]$  are selected as the true flow parameter along the horizontal direction;  $\theta_2 = [0.8 \ 0.5 \ 0.9]$  the true flow parameter along the vertical direction.  $\alpha_1 = [1.08 \ 0.6 \ 0.84]$  represents modeled flow along the horizontal direction;  $\alpha_2 = [0.96 \ 0.6 \ 1.08]$  represents modeled flow along the vertical direction. The three combined basis functions are composed of center  $c_i$ , width  $\sigma_i$ , harmonic frequency  $\omega_i$ , and harmonic phase  $v_i$ , where  $i = 1, 2, 3$ .  $c_1, c_2$ , and  $c_3$  are  $[0, 0]^\top$ ,  $[10, 10]^\top$ , and  $[5, 5]^\top$ , respectively.  $\sigma_1, \sigma_2$ , and  $\sigma_3$  are all equal to 5.  $\omega_1, \omega_2$ , and  $\omega_3$  are represented by periods 600 sec, 300 sec, and 800 sec, respectively. Those harmonic periods are arbitrary chosen. Harmonic phases  $v_1, v_2$ , and  $v_3$  are zeros.

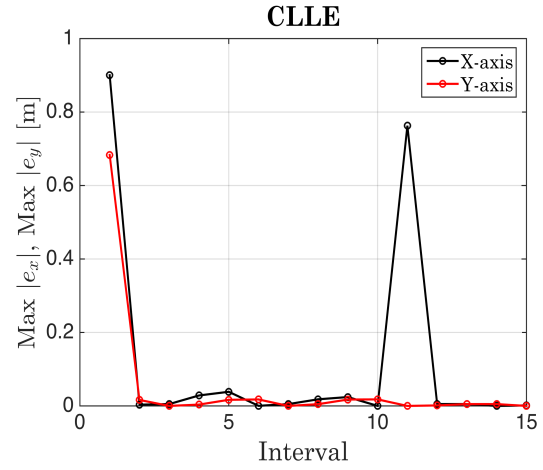
For the anomaly detection algorithm, positive constant  $K$  in the learning parameter injection term is the identity matrix. Adaptation speed  $\bar{\gamma}$  is 0.8. In the prevention scheme of false alarm, false alarm factor  $\gamma_f$  is 0.07. Fig 2.2 represents trajectories of an AUV when the direction of the AUV in the horizontal plane is controlled by heading angle command  $\Psi_c = \frac{\pi}{2} \lfloor \frac{t}{20} \rfloor$ . In Figure 2.2, the simulated true trajectory represented by the black line would have one square if there is no flow. However, because flow with spatial and temporal variabilities affects vehicle motion, the true trajectory has multiple squares.

Figure 2.4 shows simulation results of through-water speed and anomaly detection. In the upper panel, two green lines represent upper and lower bound of normal through-water speed, respectively. When actual through-water speed is 0.5m/s after 200 sec due to abnormal motion, the learning algorithm keeps tracking actual through-water speed until 300 sec. The anomaly detection algorithm shows changing flag in the bottom panel. Flag change from 0 to 2 within 10 sec shows that a false alarm happens due to the inaccuracy of identified flow in a transient period. Flag 0 to 1 is occurred at 200 sec because identified through-water speed is out of the normal range of through-water speed. Figure 2.5 shows

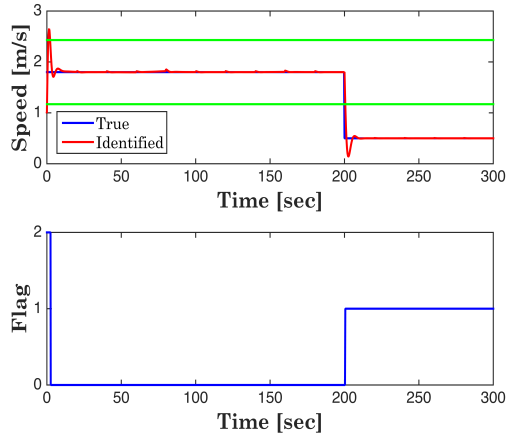
estimation error of vehicle speed converging to zero after 140 sec, but error is abruptly increasing when vehicle speed reduces to 0.5m/s because of abnormal motion. This increasing error shows that the learning algorithm is able to identify inaccurate vehicle speed in a transient period. However, error converges to zero shortly. Figure 2.3 shows the convergence of CLLE; CLLE converges to zero after 140 sec. When vehicle speed reduces to 0.5 m/s at 200 sec, CLLE is increasing abruptly, but converging to zero shortly. Figures 2.6 shows that identified flow parameters correspond to true flow parameters, and Figure 2.7 show identification error of flow parameters. In Figures 2.6, three parameters that represent flow along the X-axis in the upper panel converge to true parameters until 300 sec. Three parameters that represent flow along the Y-axis in the bottom panel converge to true parameters until 300 sec. In Figure 2.7, identification error converges to zero after 100 sec. When through-water speed reduces to 0.5 m/s after 200 sec, identification error is still converging to zero; These results support our theoretical analysis of Chapter 3.



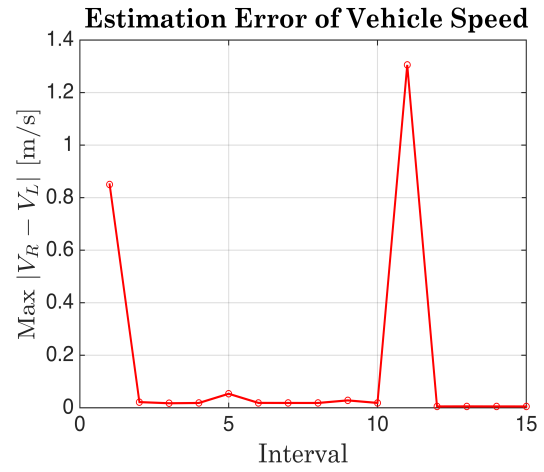
**Figure 2.2:** Every 20 sec, the heading angle command is changed with this order  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ . repeatedly, When there is no flow, the simulated true trajectory has one square, but it is not because of flow.



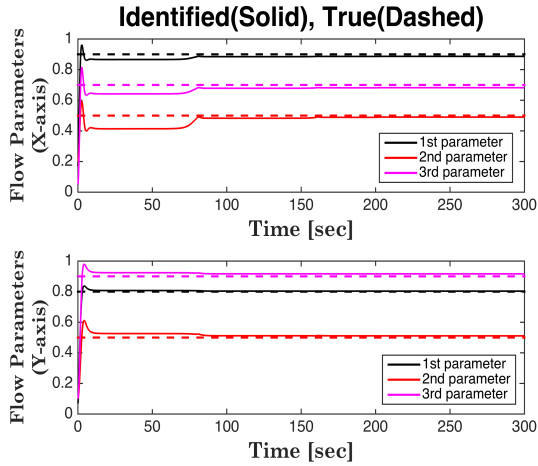
**Figure 2.3:** CLLE converges to zero after 7 intervals, but CLLE is abruptly increasing when vehicle speed reduce to 0.5m/s because of abnormal motion. After that, CLLE maintains closing zero (1 cycle=10 intervals=200 sec).



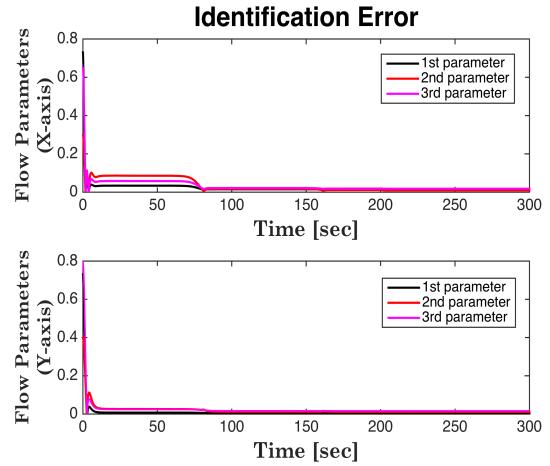
**Figure 2.4:** True through-water speed (upper) and flag (bottom)



**Figure 2.5:** Estimation error of vehicle speed (1 cycle = 10 intervals = 200 sec)



**Figure 2.6:** Convergence of flow parameters: Six flow parameters converges to true values after 20 sec.



**Figure 2.7:** Convergence of identification error: Identification error of flow parameters converges to zero after 20 sec.

## CHAPTER 3

### ANOMALY DETECTION UNDER ADAPTIVE CONTROL AND LEARNING

The previous chapter describes anomaly detection algorithms based on the estimation of through-water speed and flow velocity. When through-water speed of the vehicle is assumed to be constant, both estimates from the adaptive learning algorithm are used to detect abnormal vehicle motion, reducing false alarms. However, through-water speed can be changed by the propulsion system of AUVs (ex. [48],[49]), and control strategies. This chapter addresses the anomaly detection problem when the through-water speed is controlled by an adaptive control algorithm. A predicted trajectory is generated before vehicle deployment. Given the predicted trajectory, the adaptive control algorithm changes the through-water speed of the AUV to reduce controlled Lagrangian prediction error, which is the difference between the estimated and predicted positions. Then, estimated trajectory is acquired after vehicle deployment. Adaptive learning that identifies vehicle motion controlled by the adaptive control algorithm generates through-water and flow velocity estimates. We will show that integration of adaptive control and learning algorithms is able to create the criteria that detects anomaly and reduces false alarms.

#### 3.1 Vehicle Motion Model

Let  $\Psi_c(t)$  be heading angle commands, which are orientation angles of the vehicle to be achieved. Let  $\mathbf{y}_c(t) = [y_{1c}(t), y_{2c}(t)]^\top$  be the set of waypoints according to time in the 2D plane, which are target locations to be reached. Let  $\Gamma_c(t)$  be commands for achieving a goal. If the goal is to make a vehicle maintain the orientation angle, then we let  $\Gamma_c(t)$  equals  $\Psi_c(t)$ . If the goal is to make vehicle reach waypoints, then we let  $\Gamma_c(t)$  equals  $\mathbf{y}_c(t)$ ;

thus,  $\Gamma_c(t) \in \mathbb{R}$  or  $\Gamma_c(t) \in \mathbb{R}^2$ . The vehicle motion model is approximated by

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}_R(\mathbf{x}, t) + \mathbf{v}_R(\mathbf{x}, t, \Gamma_c(t)), \quad (3.1)$$

where  $\mathbf{x}$  is the true position of the AUV that is assumed to be known. This assumption will be removed in Chapter 5 where we discuss how to estimate  $\mathbf{x}$ . We define  $\mathbf{y}(t)$  as the predicted position of the vehicle before deploying the vehicle. The set of predicted positions, or the predicted trajectory of the vehicle is generated from the following simulated motion model.

$$\frac{d\mathbf{y}}{dt} = \mathbf{F}_M(\mathbf{y}, t) + \mathbf{v}_M(\mathbf{y}, t, \Gamma_c(t)). \quad (3.2)$$

After we deployed the AUVs, in the learning algorithm, adaptive learning that incorporates command  $\Gamma_c(t)$  as input identifies the motion of the vehicle. Let us define  $\mathbf{z}(t)$  as the output of adaptive learning, then the identified trajectory of the vehicle is modeled by:

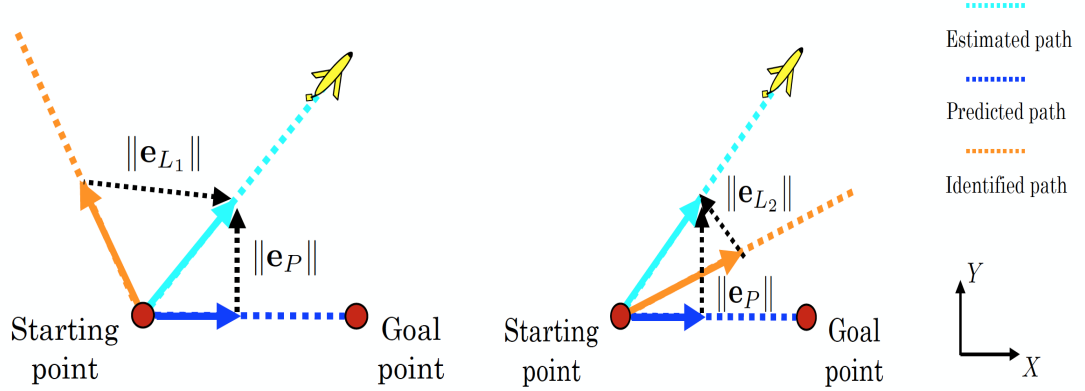
$$\frac{d\mathbf{z}}{dt} = \mathbf{F}_L(\mathbf{z}, t) + \mathbf{v}_L(\mathbf{z}, t, \Gamma_c(t)). \quad (3.3)$$

### 3.1.1 Integrating Adaptive Control and Learning for Anomaly Detection

If we have localization service for the vehicle, and if the vehicle is under adaptive control to reduce the controlled Lagrangian prediction error (CLPE), then we can use CLPE as an indicator to reduce false alarm.

A predicted trajectory can be used for anomaly detection, reducing the rate of false alarms induced by estimation error of through-water speed. Let controlled Lagrangian localization error (CLLE) denote  $\mathbf{e}_L$ , and controlled Lagrangian prediction error (CLPE)  $\mathbf{e}_P$  for notationally differentiating between CLLE and CLPE. To explain the usefulness of the predicted trajectory, Figures 3.1 and 3.2 show examples of an estimated trajectory, a predicted trajectory, and an identified trajectory together with CLLE  $\mathbf{e}_L$  and CLPE  $\mathbf{e}_P$ . Figures 3.1 and 3.2 are two different identified trajectories represented by Cases 1 and 2, respec-

tively; subscripts 1 and 2 for  $\mathbf{e}_L$  represent Cases 1 and 2, respectively. Case 1 shows that the identified trajectory is far from the estimated trajectory when we compare the identified trajectory and the predicted trajectory with respect to the estimated trajectory since CLLE is larger than CLPE. If the through-water speed estimates from the identified trajectory in Case 1 are used for anomaly detection, false detection results are likely obtained. On the other hand, Case 2 shows that CLLE is smaller than CLPE. Because the identified trajectory is closer to the estimated trajectory in Case 2 than in Case 1, through-water speed estimates from the identified trajectory are reliable. It leads to using both  $\mathbf{e}_L$  and  $\mathbf{e}_P$  enables anomaly detection, which is robust to false alarms.



**Figure 3.1:** When  $\|\mathbf{e}_{L_1}\|$  is larger than  $\|\mathbf{e}_P\|$  at time  $t$ , the identified trajectory is relatively far away from the estimated trajectory

**Figure 3.2:** When  $\|\mathbf{e}_{L_2}\|$  is smaller than  $\|\mathbf{e}_P\|$  at time  $t$ , the identified trajectory is relatively close to the estimated trajectory.

### 3.2 Flow Canceling Control for Maintaining Heading

The vehicle is using adaptive flow canceling control, which is an assumption that will make the use of CLPE possible. The adaptive controller plays a key role in controlling the AUV to follow the predicted trajectory generated in the stage of pre-deployment. Let  $\vartheta(t) = \begin{bmatrix} \vartheta_1^1(t) & \cdots & \vartheta_1^N(t) \\ \vartheta_2^1(t) & \cdots & \vartheta_2^N(t) \end{bmatrix}$  be a  $2 \times N$  matrix with time varying parameters. Let  $V_I(t) \in \mathbb{R}$  be

time-varying speed. Let  $\delta(t) \in \mathbb{R}^2$  be a control injection parameter. Let  $\Psi_c(t)$  be heading angle command as external input. When  $\Gamma_c(t) = \Psi_c(t)$ , we let the through-water velocity of the vehicle be

$$\mathbf{v}_R(\mathbf{x}, t, \psi_c(t)) = -\vartheta(t)\phi(\mathbf{x}, t) + V_I(t)\Psi_c(t) + \delta(t). \quad (3.4)$$

Equation (3.4) represents the control law with flow canceling. Flow canceling term  $-\vartheta(t)\phi(\mathbf{x}, t)$ , velocity control term  $V_I(t)\Psi_c$ , and stabilizing term  $\delta(t)$  are combined for vehicle control. The velocity control term makes the vehicle move with time-varying speed  $V_I$  and heading angle command  $\psi_c$  after canceling out estimated flow.

For the predicted trajectory, we assume that vehicle speed is constant after canceling modeled flow. This assumption is different from the constant speed assumption used in Chapter 2. In this Chapter, the feedback controller on the vehicle is responsible to keep a constant speed in the direction of heading angle after the flow is canceled. This assumption will be removed in Section 4.3 where we discuss the structure of feedback and feedforward controllers. Let  $V_M \in \mathbb{R}$  be fixed speed,  $V_L(t) \in \mathbb{R}$  time-varying speed. Let  $\beta(t) \in \mathbb{R}^2$  be a learning injection parameter. Then, through-water velocity  $\mathbf{v}_M$  for the predicted trajectory and through-water velocity  $\mathbf{v}_L$  for the identified trajectory are

$$\mathbf{v}_M(\mathbf{y}, t, \psi_c(t)) = -\mathbf{F}_M(\mathbf{y}, t) + V_M\Psi_c \quad (3.5)$$

$$\mathbf{v}_L(\mathbf{z}, t, \psi_c(t)) = -\mathbf{F}_M(\mathbf{z}, t) + V_L(t)\Psi_c + \beta(t), \quad (3.6)$$

respectively.  $V_L(t)$  is different from  $V_I(t)$  in that  $V_L(t)$  is designed to reduce CLLE; however,  $V_I(t)$  is designed to reduce CLPE.

By plugging equations (3.6) and (2.8) into equation (2.3), the closed-loop dynamics for the identified trajectory is

$$\dot{\mathbf{z}} = (\xi(t) - \alpha)\phi(\mathbf{z}, t) + V_L(t)\Psi_c + \beta(t). \quad (3.7)$$

Meanwhile, the closed-loop dynamics for the generation of the estimated trajectory is

$$\dot{\mathbf{x}} = (\boldsymbol{\theta} - \boldsymbol{\vartheta}(t))\phi(\mathbf{x}, t) + V_I(t)\Psi_c + \boldsymbol{\delta}(t), \quad (3.8)$$

when we plug equations (3.4) and (2.4) into equation (2.1). Both of closed-loop dynamics, represented by equations (3.7) and (3.8), are different from closed dynamics we described in equations (2.11) and (2.12); flow canceling terms  $-\alpha\phi(\mathbf{z}, t)$  and  $-\boldsymbol{\vartheta}(t)\phi(\mathbf{x}, t)$  are added into closed-loop dynamics. If  $\mathbf{x}$  and  $\mathbf{z}$  have the same initial condition and  $V_L(t) = V_I(t)$ ,  $\beta(t) = (\boldsymbol{\theta} - \boldsymbol{\vartheta}(t))\phi(\mathbf{z}, t) - (\xi(t) - \alpha)\phi(\mathbf{z}, t) + \boldsymbol{\delta}(t)$ , the identified trajectory is identical to the estimated trajectory. Our goal is to design updating rules and the learning injection parameter  $\beta(t)$  so that the two types of closed-loop dynamics can be similar. Note that the parameter  $\boldsymbol{\theta}$  is unknown so we can not use  $\beta(t) = (\boldsymbol{\theta} - \boldsymbol{\vartheta}(t))\phi(\mathbf{z}, t) - (\xi(t) - \alpha)\phi(\mathbf{z}, t) + \boldsymbol{\delta}(t)$ , directly.

By plugging equations (3.5) and (2.5) into equation (3.2), the closed-loop dynamics for the predicted trajectory is

$$\begin{aligned} \dot{\mathbf{y}} &= \mathbf{F}_M(\mathbf{y}, t) + \mathbf{v}_M(\mathbf{y}, t, \Psi_c(t)), \\ &= \mathbf{F}_M(\mathbf{y}, t) - \mathbf{F}_M(\mathbf{y}, t) + V_M\Psi_c \\ &= V_M\Psi_c. \end{aligned} \quad (3.9)$$

If  $\mathbf{x}$  and  $\mathbf{y}$  have the same initial condition and  $V_I(t) = V_M$ , and suppose  $\boldsymbol{\delta}(t) = -(\boldsymbol{\theta} - \boldsymbol{\vartheta}(t))\phi(\mathbf{x}, t)$ , then the estimated trajectory is identical to the predicted trajectory. Our goal is to design adaptation laws and the control injection parameter so that the closed-loop dynamics for the estimated trajectory and closed-loop dynamics for the predicted trajectory can be similar. Note that the parameter  $\boldsymbol{\theta}$  is unknown so we can not use  $\boldsymbol{\delta}(t) = -(\boldsymbol{\theta} - \boldsymbol{\vartheta}(t))\phi(\mathbf{x}, t)$ , directly.



### 3.2.1 Controlled Lagrangian prediction and localization errors

We first derive controlled Lagrangian prediction error (CLPE) dynamics that shows how much the estimated trajectory deviates from the predicted trajectory. On the subtraction of equation (3.9) from equation (3.8), CLPE dynamics represents

$$\dot{\mathbf{e}}_P = \dot{\mathbf{x}} - \dot{\mathbf{y}} = (\theta - \vartheta(t))\phi(\mathbf{x}, t) + (V_I(t) - V_M)\Psi_c + \delta(t). \quad (3.10)$$

Suppose  $\vartheta(t)$  converges to  $\theta$ ,  $V_I(t)$  converges to  $V_M$ , and  $\delta(t) = -K_1 \mathbf{e}_P$  where  $K_1$  is a diagonal matrix with positive components, then CLPE goes to zero over time, which implies the estimated trajectory follows the predicted trajectory. Our first goal is to design adaptation laws for updating parameters  $\vartheta$  and  $V_I$  by using CLPE dynamics so that CLPE converges to zero.

For the development of the adaptive learning algorithm, we derive controlled Lagrangian localization error (CLLE) dynamics. The difference between the estimated trajectory and the identified trajectory, or CLLE, is a measure presenting the deviation of the estimated trajectory from the identified trajectory. By subtracting equations (3.7) from equation (3.8), CLLE dynamics represents

$$\dot{\mathbf{e}}_L = \dot{\mathbf{x}} - \dot{\mathbf{z}} = (\theta - \vartheta(t))\phi(\mathbf{x}, t) - (\xi(t) - \alpha)\phi(\mathbf{z}, t) + (V_I(t) - V_L(t))\Psi_c + \delta(t) - \beta(t). \quad (3.11)$$

Unlike Chapter 2 that shows the proof of convergence of the learning algorithm for constant vehicle speed  $V_R$ , here time varying speed  $V_I(t)$  is included in CLLE dynamics; this leads to difficulties on the proof of convergence. Thus, we incorporate both adaptation laws for the adaptive control algorithm and updating rules for the adaptive learning algorithm. Our second goal is to design updating rules for updating parameters  $\vartheta$ ,  $\xi$ , and  $V_L$  by using CLLE dynamics so that CLLE converges to zero under the adaptation laws.

### 3.2.2 Adaptive control and learning algorithms

The adaptive control algorithm is developed to make the estimated trajectory follow the predicted trajectory. Because the estimated trajectory is disturbed by ocean flow, the proposed algorithm keeps updating flow parameters and time-varying speed with on-line adaptive control laws so that the estimated trajectory follows the predicted trajectory. We design adaptation laws for time-varying parameters as follows:

$$\dot{\bar{\vartheta}}(t) = \gamma \mathbf{e}_P \otimes \phi(\mathbf{x}, t) \quad (3.12)$$

$$\dot{V}_I(t) = -\gamma \mathbf{e}_P^\top \Psi_c, \quad (3.13)$$

where  $\otimes$  is the Kronecker product, and  $\gamma$  is any positive constant. Let

$$\bar{\vartheta}(t) = [\vartheta_1^1(t), \dots, \vartheta_1^N(t), \vartheta_2^1(t), \dots, \vartheta_2^N(t)]^\top \quad (3.14)$$

$$\mathbf{e}_P \otimes \phi = [e_{P_1} \phi^1, \dots, e_{P_1} \phi^N, e_{P_2} \phi^1, \dots, e_{P_2} \phi^N]^\top \quad (3.15)$$

be row vectors in  $\mathbb{R}^{2N}$ . We design a control injection parameter as follows

$$\delta(t) = -K_1 \mathbf{e}_P, \quad (3.16)$$

where  $K_1$  is a diagonal positive definite matrix. With the adaptation laws, we prove error and parameter convergence to show that the adaptive control algorithm makes the estimated trajectory follow the predicted trajectory in unknown ocean flow fields. Error convergence shows that the adaptive control algorithm accurately estimates flow velocity and cancels estimated flow velocity, which implies that the estimated trajectory follows the predicted trajectory. We prove that CLPE converges to zero in Theorem 3.1.

*Theorem 3.1* Using equations (3.12) and (3.13), CLPE converges to zero when time goes

to infinity; that is,  $\mathbf{e}_P(t) \rightarrow \vec{0}$  as  $t \rightarrow \infty$ .

*Proof.* Consider a candidate Lyapunov function:

$$V(\mathbf{e}_P, \bar{\vartheta}, V_R) = \frac{1}{2} \left\{ \mathbf{e}_P^\top \mathbf{e}_P + \frac{1}{\gamma} (\bar{\theta} - \bar{\vartheta})^\top (\bar{\theta} - \bar{\vartheta}) + \frac{1}{\gamma} (V_I(t) - V_M)^2 \right\}. \quad (3.17)$$

The derivative of  $V$  is

$$\begin{aligned} \dot{V} = & -\mathbf{e}_P^\top K_1 \mathbf{e}_P + \mathbf{e}_P^\top (\theta - \vartheta(t)) \phi(\mathbf{x}, t) + (V_I(t) - V_M) \left( \mathbf{e}_P^\top \Psi_c - \frac{1}{\gamma} \dot{V}_I(t) \right) \\ & - \frac{1}{\gamma} (\bar{\theta} - \bar{\vartheta}(t)) \dot{\bar{\vartheta}}(t). \end{aligned}$$

We know that  $\mathbf{e}_P^\top (\theta - \vartheta(t)) \phi(\mathbf{x}, t) = (\bar{\theta} - \bar{\vartheta}(t)) \mathbf{e}_P \otimes \phi(\mathbf{x}, t)$ . Then, using equation (3.12) and (3.13),

$$\dot{V} = -\mathbf{e}_P^\top K_1 \mathbf{e}_P \leq 0. \quad (3.18)$$

$\dot{V}$  is negative semi-definite and this implies  $\mathbf{e}_P$ ,  $\vartheta(t)$ ,  $V_I(t)$  are bounded. In addition, the second order time derivative of  $V$  satisfies

$$\ddot{V} = -2\mathbf{e}_P^\top K_1 \dot{\mathbf{e}}_P = -2\mathbf{e}_P^\top K_1 \{ (\theta - \vartheta(t)) \phi(\mathbf{x}, t) + (V_I(t) - V_M) \Psi_c(t) - K_1 \mathbf{e}_P \}. \quad (3.19)$$

Because  $\Psi_c$  is bounded,  $\ddot{V}$  is bounded, and hence  $\dot{V}$  is uniformly continuous. By Lemma 2.4,  $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$ . Since  $K_1$  is the positive definite diagonal matrix,  $\mathbf{e}_P(t) \rightarrow \vec{0}$  as  $t \rightarrow \infty$ . □

Despite error convergence, flow velocity may not be precisely estimated in that multiple parameters that represent flow are identified from two types of trajectory information, which are the estimated trajectory and the predicted trajectory. Because the control algorithm uses estimated flow velocity to cancel actual flow and follow the predicted trajectory, inaccurate flow estimates may prevent the estimated trajectory from following the predicted trajectory. Thus, we prove the convergence of parameters  $\bar{\vartheta}$  and  $V_I$  in Theorem 3.2.

*Theorem 3.2*  $\bar{\vartheta}(t)$  and  $V_I(t)$  converges to  $\bar{\theta}$  and  $V_M$ , respectively; that is,  $\bar{\vartheta}(t) \rightarrow \bar{\theta}$ , and  $V_I(t) \rightarrow V_M$  as  $t \rightarrow \infty$

*Proof.* Let  $\eta_1$ ,  $\eta_2$  and  $\eta_3$  be  $(\theta_1 - \vartheta_1(t))$ ,  $(\theta_2 - \vartheta_2(t))$ , and  $(V_I(t) - V_M)$ , respectively. Let  $\tilde{\phi}_1 = \begin{bmatrix} \phi_1^1 & \cdots & \phi_1^N \\ 0 & \cdots & 0 \end{bmatrix}$  and  $\tilde{\phi}_2 = \begin{bmatrix} 0 & \cdots & 0 \\ \phi_2^1 & \cdots & \phi_2^N \end{bmatrix}$  be in  $\mathbb{R}^{2 \times N}$ . We rewrite equation (3.10) using  $\eta_1$ ,  $\eta_2$ , and  $\eta_3$  as follows:

$$\dot{\mathbf{e}}_P = \tilde{\phi}_1(\mathbf{x}, t)\eta_1 + \tilde{\phi}_2(\mathbf{x}, t)\eta_2 + \Psi_c\eta_3 - K_1\mathbf{e}_P. \quad (3.20)$$

Because equation (3.20) has the same form of equation (2.28), and  $w$  is persistently exciting by Assumption 3.5, we use the same proof of Theorem 2.3. Hence,  $\bar{\vartheta}$  and  $V_I$  converge to  $\bar{\theta}$  and  $V_M$ , respectively.  $\square$

The adaptive learning algorithm in Chapter 2 uses the estimated trajectory propagated by constant through-water speed; however, the estimated trajectory in this chapter is generated by time-varying through-water speed. Because the adaptation laws represented by equations (3.12) and (3.13) govern the through-water speed, we need to incorporate the adaptation laws into the proposed learning algorithm. We design the updating rules as follows:

$$\dot{\bar{\xi}}(t) = \tilde{\gamma}\mathbf{e}_L \otimes \phi(\mathbf{x}, t) - \gamma\mathbf{e}_P \otimes \phi(\mathbf{x}, t) \quad (3.21)$$

$$\dot{V}_L(t) = \tilde{\gamma}\mathbf{e}_L^\top \Psi_c - \gamma\mathbf{e}_P^\top \Psi_c, \quad (3.22)$$

where  $\tilde{\gamma}$  is a design parameter, which is positive constant. Let

$$\bar{\xi}(t) = [\xi_1^1(t), \dots, \xi_1^N(t), \xi_2^1(t), \dots, \xi_2^N(t)]^\top \quad (3.23)$$

$$\bar{\alpha}(t) = [\alpha_1^1(t), \dots, \alpha_1^N(t), \alpha_2^1(t), \dots, \alpha_2^N(t)]^\top \quad (3.24)$$

$$\mathbf{e}_L \otimes \phi = [e_{L_1}\phi^1, \dots, e_{L_1}\phi^N, e_{L_2}\phi^1, \dots, e_{L_2}\phi^N]^\top \quad (3.25)$$

be row vectors in  $\mathbb{R}^{2N}$ . Then, we design learning parameter injection as follows:

$$\beta(t) = (\xi(t) - \alpha) \phi(\mathbf{x}, t) - (\xi(t) - \alpha) \phi(\mathbf{z}, t) + K_2 \mathbf{e}_L - K_1 \mathbf{e}_P, \quad (3.26)$$

where  $K_2$  is a positive definite diagonal matrix with positive components. In order to identify vehicle motion, we first prove that CLLE converges to zero with the designed updating rules in Theorem 3.3.

*Theorem 3.3* Using equations (3.12), (3.13), (3.21) and (3.22), CLLE converges to zero when time goes to infinity; that is,  $\mathbf{e}_L(t) \rightarrow \vec{0}$  as  $t \rightarrow \infty$ .

*Proof.* Consider a candidate Lyapunov function:

$$\begin{aligned} V(\mathbf{e}_L, \vartheta, \xi, V_I, V_L) \\ = \frac{1}{2} \left\{ \mathbf{e}_L^\top \mathbf{e}_L + \frac{1}{\tilde{\gamma}} (\bar{\theta} - \bar{\vartheta}(t) - (\bar{\xi}(t) - \bar{\alpha}))^\top (\bar{\theta} - \bar{\vartheta}(t) - (\bar{\xi}(t) - \bar{\alpha})) + \frac{1}{\tilde{\gamma}} (V_I(t) - V_L(t))^2 \right\}. \end{aligned} \quad (3.27)$$

The derivative of  $V$  is

$$\begin{aligned} \dot{V} = & -\mathbf{e}_L^\top K_2 \mathbf{e}_L + \mathbf{e}_L^\top (\theta - \vartheta(t) - (\xi(t) - \alpha)) \phi(\mathbf{x}, t) \\ & + (V_I(t) - V_L(t)) \left( \mathbf{e}_L^\top \Psi_c + \frac{1}{\tilde{\gamma}} (\dot{V}_I(t) - \dot{V}_L(t)) \right) \\ & - \frac{1}{\tilde{\gamma}} (\bar{\theta} - \bar{\vartheta}(t) - (\bar{\xi}(t) - \bar{\alpha}))^\top (\dot{\bar{\vartheta}}(t) + \dot{\bar{\xi}}(t)). \end{aligned}$$

We know that  $\mathbf{e}_L^\top (\theta - \vartheta(t) - (\xi(t) - \alpha)) \phi(\mathbf{x}, t)$  equals to  $(\bar{\theta} - \bar{\vartheta}(t) - (\bar{\xi}(t) - \bar{\alpha}))^\top \mathbf{e}_L \otimes \phi(\mathbf{x}, t)$ . Then,

$$\begin{aligned} \dot{V} = & -\mathbf{e}_L^\top K_2 \mathbf{e}_L + (\bar{\theta} - \bar{\vartheta}(t) - (\bar{\xi}(t) - \bar{\alpha}))^\top \left( \mathbf{e}_L \otimes \phi(\mathbf{x}, t) - \frac{1}{\tilde{\gamma}} (\dot{\bar{\vartheta}}(t) + \dot{\bar{\xi}}(t)) \right) \\ & + (V_I(t) - V_L(t)) \left( \mathbf{e}_L^\top \Psi_c + \frac{1}{\tilde{\gamma}} (\dot{V}_I(t) - \dot{V}_L(t)) \right). \end{aligned}$$

In addition, we know that  $\dot{\vartheta}(t) = \gamma \mathbf{e}_P \otimes \phi(\mathbf{x}, t)$ , and  $\dot{V}_I(t) = -\gamma \mathbf{e}_P^\top \Psi_c$  from the adaptation laws represented by equations (3.12) and (3.13). Then,

$$\begin{aligned} \dot{V} = & -\mathbf{e}_L^\top K_2 \mathbf{e}_L + (\bar{\theta} - \vartheta(t) - (\bar{\xi}(t) - \bar{\alpha}))^\top \left( \mathbf{e}_L \otimes \phi(\mathbf{x}, t) - \frac{1}{\bar{\gamma}} \left( \gamma \mathbf{e}_P \otimes \phi(\mathbf{x}, t) + \dot{\bar{\xi}}(t) \right) \right) \\ & + (V_I(t) - V_L(t)) \left( \mathbf{e}_L^\top \Psi_c + \frac{1}{\bar{\gamma}} \left( -\gamma \mathbf{e}_P^\top \Psi_c - \dot{V}_L(t) \right) \right). \end{aligned}$$

Using equation (3.21) and (3.22),

$$\dot{V} = -\mathbf{e}_L^\top K_2 \mathbf{e}_L \leq 0. \quad (3.28)$$

$\dot{V}$  is negative semi-definite and this implies that  $\mathbf{e}_L$ ,  $\xi(t)$ , and  $V_L(t)$  are bounded. In addition, the second order time derivative of  $V$  satisfies

$$\ddot{V} = -2\mathbf{e}_L^\top K_2 \{ (\theta - \vartheta(t) - (\xi(t) - \alpha)) \phi(\mathbf{x}, t) + (V_I(t) - V_L(t)) \Psi_c - K_2 \mathbf{e}_L \}. \quad (3.29)$$

Because  $\vartheta$ ,  $V_I$ , and  $\Psi_c$  are bounded,  $\ddot{V}$  is bounded, and hence  $\dot{V}$  is uniformly continuous. By Lemma 2.4,  $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$ . Since  $K_2$  is the diagonal matrix,  $\mathbf{e}_L(t) \rightarrow \vec{0}$  when  $t \rightarrow \infty$ .  $\square$

Even if vehicle motion identified from CLLE convergence, the ocean flow field and through-water speed may not be accurately identified from the learning algorithm in that multiple parameters that represent flow are extracted from one type of information, which is the estimated trajectory. For the accurate identification of vehicle motion, we prove that identified parameters converge to true parameters with the designed updating rules represented by equations (3.21) and (3.22) in Theorem 3.4.

*Theorem 3.4* Under the same setting of Theorem 3.3,  $\bar{\xi}(t)$  and  $V_L(t)$  converges to  $\bar{\alpha}$  and  $V_I(t)$ , respectively; that is,  $\bar{\xi}(t) \rightarrow \bar{\alpha}$ , and  $V_L(t) \rightarrow V_I(t)$  as  $t \rightarrow \infty$

*Proof.* Let  $\zeta_1$ ,  $\zeta_2$  and  $\zeta_3$  be  $(\theta_1 - \vartheta_1(t) - (\xi_1(t) - \alpha_1))$ ,  $(\theta_2 - \vartheta_2(t) - (\xi_2(t) - \alpha_2))$ , and

$(V_I(t) - V_L(t))$ , respectively. Let  $\tilde{\phi}_1 = \begin{bmatrix} \phi_1^1 & \cdots & \phi_1^N \\ 0 & \cdots & 0 \end{bmatrix}$  and  $\tilde{\phi}_2 = \begin{bmatrix} 0 & \cdots & 0 \\ \phi_2^1 & \cdots & \phi_2^N \end{bmatrix}$  be in  $\mathbb{R}^{2 \times N}$ . Let  $\tilde{\Psi} = \text{diag}\{\cos(\psi_c), \sin(\psi_c)\}$ . We rewrite equation (3.11) using equation  $\zeta_1, \zeta_2$ , and  $\zeta_3$  as follows:

$$\dot{\mathbf{e}}_L = \tilde{\phi}_1(\mathbf{x}, t)\zeta_1 + \tilde{\phi}_2(\mathbf{x}, t)\zeta_2 + \tilde{\Psi}\zeta_3 - K_2\mathbf{e}_L. \quad (3.30)$$

Because equation (3.30) has the same form of equation (2.28), we use the same proof of Theorem 2.3. Hence,  $\tilde{\xi}(t)$  and  $V_L(t)$  converge to  $\bar{\alpha}$  and  $V_I(t)$ , respectively.  $\square$

CLLE interpreted by the accuracy of time-varying speed estimates may be large in a transient period although we prove the convergence of time-varying vehicle speed. In this case, it is difficult to use time-varying speed estimates for anomaly detection. We propose a criteria when time-varying speed estimates are reliable, incorporating the predicted trajectory. Let  $\tilde{\mathbf{e}} = \mathbf{e}_L - \mathbf{e}_P$  be the difference between CLLE and CLPE. Let  $\bar{\mathbf{e}} = \mathbf{e}_L + \mathbf{e}_P$  be the summation of CLLE and CLPE. Let  $K = K_1 = K_2$  be a positive definite diagonal matrix with positive components. Let  $\tilde{\mathbf{e}}_i$  be each component of  $\tilde{\mathbf{e}}$ , let  $\bar{\mathbf{e}}_i$  be each component of  $\bar{\mathbf{e}}$ , let  $k_i$  be each component of  $K$ , and let  $\Psi_{c_i}$  be each component of  $\Psi_c$ , where  $i = 1, 2$ .

*Theorem 3.5* Given the adaptation laws described by equations (3.12) and (3.13), and the updating rules (3.21) and (3.22),  $\|\tilde{\mathbf{e}}\|$  is bounded, and  $\|\bar{\mathbf{e}}\|$  is bounded. If  $\tilde{\mathbf{e}}_i(t_0) < 0$ ,  $\bar{\mathbf{e}}_i(t_0) > 0$ ,  $\tilde{\mathbf{e}}_i \geq \frac{g_i(t)}{k_i}$ , and  $\bar{\mathbf{e}}_i \leq \frac{1}{k_i}(g_i(t) + h_i(t))$ , where  $g_i(t) = \sum_{j=1}^N (\alpha_i^j - \xi_i^j(t)) \phi^j(\mathbf{x}, t) + (V_M - V_L(t))\Psi_{c_i}$ , and  $h_i(t) = 2 \left( \sum_{j=1}^N (\theta_i^j - \vartheta_i^j(t)) \phi^j(\mathbf{x}, t) + (V_I(t) - V_M)\Psi_{c_i} \right)$ , then  $\|\mathbf{e}_L\| < \|\mathbf{e}_P\|$  at  $t \geq t_0$ .

*Proof.*  $\|\mathbf{e}_L\| < \|\mathbf{e}_P\|$  is equivalent to  $\|\mathbf{e}_L\|^2 < \|\mathbf{e}_P\|^2$ . We will prove  $\|\mathbf{e}_L\| < \|\mathbf{e}_P\|$  by showing that  $\sum_{i=1}^2 \tilde{\mathbf{e}}_i \bar{\mathbf{e}}_i < 0$ . By subtracting equation (3.10) from equation (3.11), we obtain  $\tilde{\mathbf{e}}$  dynamics below.

$$\dot{\tilde{\mathbf{e}}}_i = \sum_{j=1}^N (\alpha_i^j - \xi_i^j(t)) \phi^j(\mathbf{x}, t) + (V_M - V_L(t))\Psi_{c_i} - k_i \tilde{\mathbf{e}}_i. \quad (3.31)$$

$\xi(t)$ ,  $V_L(t)$  are bounded by Theorem 3.3. The derivative of equation (3.31) is

$$\begin{aligned}\ddot{\mathbf{e}}_i = & -k_i \dot{\mathbf{e}}_i - \sum_{j=1}^N \dot{\xi}_i^j(t) \phi^j(\mathbf{x}, t) + \sum_{j=1}^N \left( \alpha_i^j - \xi_i^j(t) \right) \dot{\phi}^j(\mathbf{x}, t) \\ & - \dot{V}_L(t) \Psi_{c_i} + (V_M - V_L(t)) \dot{\Psi}_{c_i}.\end{aligned}\quad (3.32)$$

Since  $\dot{\xi}(t)$  and  $\dot{V}_L(t)$  are bounded by the updating rules,  $\ddot{\mathbf{e}}_i$  is bounded; hence,  $\dot{\mathbf{e}}_i$  is uniformly continuous. By Lemma 2.4,  $\dot{\mathbf{e}}_i$  goes to zero when time goes to infinity. Therefore,  $\tilde{\mathbf{e}}_i$  is bounded; hence,  $\|\tilde{\mathbf{e}}\|$  is bounded. By summing equations (3.10) and (3.11), we obtain  $\bar{\mathbf{e}}$  dynamics below.

$$\dot{\bar{\mathbf{e}}} = \sum_{j=1}^N \left( 2(\theta_i^j - \vartheta_i^j(t)) + (\alpha_i^j - \xi_i^j(t)) \right) \phi^j(\mathbf{x}, t) + (2V_I(t) - V_L(t) - V_M) \Psi_{c_i} - k_i \bar{\mathbf{e}}_i. \quad (3.33)$$

$\vartheta(t)$ ,  $V_I(t)$  are bounded by Theorem 3.1.  $\xi(t)$ ,  $V_L(t)$  are bounded by Theorem 3.3. The derivative of equation (3.33) is

$$\begin{aligned}\ddot{\bar{\mathbf{e}}} = & -k_i \dot{\bar{\mathbf{e}}} - \sum_{j=1}^N -2(-\dot{\vartheta}_i^j(t) - \dot{\xi}_i^j(t)) \phi^j(\mathbf{x}, t) + \sum_{j=1}^N 2 \left( \dot{\theta}_i^j - \dot{\vartheta}_i^j(t) - \dot{\xi}_i^j(t) + \dot{\alpha}_i^j \right) \dot{\phi}^j(\mathbf{x}, t) \\ & 2(\dot{V}_I(t) - \dot{V}_L(t)) \Psi_{c_i} + (2V_I(t) - V_L(t) - V_M) \dot{\Psi}_{c_i}.\end{aligned}\quad (3.34)$$

Since  $\dot{\xi}(t)$  and  $\dot{V}_L(t)$  are bounded by the updating rules, and  $\dot{\vartheta}(t)$  and  $\dot{V}_I(t)$  are bounded by the adaptation laws,  $\ddot{\bar{\mathbf{e}}}_i$  is bounded; hence,  $\dot{\bar{\mathbf{e}}}_i$  is uniformly continuous. By Lemma 2.4,  $\dot{\bar{\mathbf{e}}}_i$  goes to zero when time goes to infinity. Therefore,  $\bar{\mathbf{e}}_i$  is bounded; hence,  $\|\bar{\mathbf{e}}\|$  is bounded. If  $\sum_{j=1}^N \left( \alpha_i^j - \xi_i^j(t) \right) \phi^j(\mathbf{x}, t) + (V_M - V_L(t)) \Psi_{c_i} \leq k_i \bar{\mathbf{e}}_i$ , then  $\dot{\bar{\mathbf{e}}}_i \leq 0$ . Thus,  $\bar{\mathbf{e}}_i < 0$  when  $\bar{\mathbf{e}}_i(t_0) < 0$  where  $t_0$  is initial time. In addition, If  $\sum_{j=1}^N \left( \alpha_i^j - \xi_i^j(t) \right) \phi^j(\mathbf{x}, t) + (V_M - V_L(t)) \Psi_{c_i} + 2 \left( \sum_{j=1}^N \left( \theta_i^j - \vartheta_i^j(t) \right) \phi^j(\mathbf{x}, t) + (V_I(t) - V_M) \Psi_{c_i} \right) \geq k_i \bar{\mathbf{e}}_i$ , then  $\dot{\bar{\mathbf{e}}}_i \geq 0$ . Thus,  $\bar{\mathbf{e}}_i > 0$  when  $\bar{\mathbf{e}}_i(t_0) > 0$ . Therefore,  $\sum_{i=1}^2 \bar{\mathbf{e}}_i \dot{\bar{\mathbf{e}}}_i < 0$ ;  $\|\mathbf{e}_L\| < \|\mathbf{e}_P\|$  at  $t \geq t_0$ .  $\square$

Theorem 3.5 provides conditions when the magnitude of CLLE is smaller than the magnitude of CLPE. This indicates how large CLLE is allowed with the predicted trajectory



to identify when abnormal motion is detectable. In Theorem 3.5,  $h_i$  represents estimation errors of vehicle speed and flow velocity used for the adaptive control algorithm, and  $g_i$  represents identification errors of vehicle speed and flow velocity used for the anomaly detection algorithm. Let us consider that CLPE has a certain value. This implies that  $h_i$  has a certain value. One way that keeps CLLE smaller than CLPE is that we increase gain  $k_i$ . Increasing  $k_i$  makes a smaller lower bound for  $\tilde{e}_i$ ; that lower bound allows the large value of  $\tilde{e}_i$ . In addition, changing the adaptation speed of the learning algorithm enables the smaller lower bound. Because  $g_i$  contains identification errors of flow velocity and vehicle speed, fast adaptation speed decreases estimation errors in a short time; then, the value of  $g_i$  is small.

The anomaly detection algorithm including the detectable criteria is described by Algorithm 2 extended from Algorithm 1 of the previous Chapter. When the magnitude of CLLE is larger than or equal to the magnitude of CLPE, Algorithm 2 considers abnormal motion is not detectable due to large estimation error of controlled speed; then, the flag number is set to value 3, which represents a false alarm.

---

**Algorithm 2:** Anomaly Detection Algorithm Integrating Adaptive Control and Learning

---

**Input:** Flow velocity estimate  $F_L(t)$ , modeled flow estimate  $F_M(t)$ , false alarm factor  $\gamma_f$ , time-varying speed estimate  $V_L(t)$ , maximum speed  $V_{\max}$ , minimum speed  $V_{\min}$ , CLPE  $\mathbf{e}_P$ , CLLE  $\mathbf{e}_L$

**Output:** Anomaly detection flag

```

1  $\hat{F}_{L_{\max}} = \max(F_L(\tau)_{\tau \in [0, t]})$ 
2  $\hat{F}_{M_{\max}} = \max(F_M(\tau)_{\tau \in [0, t]})$ 
3  $p_E = \frac{\|F_L(t) - F_M(t)\|}{2\max(\hat{F}_{L_{\max}}, \hat{F}_{M_{\max}})}$ 
4 if  $\|\mathbf{e}_L\| \geq \|\mathbf{e}_P\|$  then
5   | flag = 3  $\triangleright$  False Alarm
6 else if  $p_E > \gamma_f$  then
7   | flag = 2  $\triangleright$  False Alarm
8 else if  $V_L(t) > V_{\max}$  or  $V_L(t) < V_{\min}$  then
9   | flag = 1  $\triangleright$  Anomaly Detected
10 else
11   | flag = 0  $\triangleright$  No Anomaly Detected
12 end
```

---

### 3.3 Flow Canceling Control for Trajectory Tracing

We focus on a type of state feedback controller with flow canceling for trajectory tracing missions. Trajectory tracing is to estimate unknown vehicle trajectory with estimated flow velocity and heading angle [50]. After deploying an autonomous underwater vehicle, we encounter estimating vehicle trajectory without any localization service; this mission is called trajectory tracing mission. For the trajectory tracing missions, we design a flow canceling controller. Previous work [2] cancel out flow velocity with respect to the straight line between current position and goal position of the AUV. In our work, we cancel flow at the current position as much as the vehicle can and reduce positional error between current position and goal position of the vehicle. We do this through a vehicle controller designed by a combination of feedback and feedforward control laws.

On the other hand, the vehicle controller can be saturated because of control power constraints of AUVs. We modify equations (3.1) as follows:

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}_R(\mathbf{x}, t) + \mathbf{u}(\mathbf{x}, t, \mathbf{y}_c(t)), \quad (3.35)$$

$$\mathbf{u}(\mathbf{x}, t, \mathbf{y}_c(t)) = \begin{cases} \mathbf{v}_R(\mathbf{x}, t, \mathbf{y}_c(t)) & \text{if } \|\mathbf{v}_R(\mathbf{x}, t, \mathbf{y}_c(t))\|_\infty \leq u_0 \\ u_0 \text{sgn}(\mathbf{v}_R(\mathbf{x}, t, \mathbf{y}_c(t))) & \text{if } \|\mathbf{v}_R(\mathbf{x}, t, \mathbf{y}_c(t))\|_\infty > u_0. \end{cases} \quad (3.36)$$

The maximum through-water speed  $u_0$  is determined by the hardware configuration of AUVs. Our goal is to design the controller that identifies vehicle motion for AUVs with saturated through-water velocity so that the controller makes the identified trajectory follow the estimated trajectory in the ocean flow field.

#### 3.3.1 Adaptive Control Algorithm

An adaptive controller plays a key role in controlling the AUV to follow the predicted trajectory generated in the stage of pre-deployment. We design a desired controller for

the predicted trajectory and the adaptive controller for the estimated trajectory. Let  $K_M = \text{diag}\{K_{M_1}, K_{M_2}\}$  be diagonal matrices with known parameters. Let  $K_R = \text{diag}\{K_{R_1}(t), K_{R_2}(t)\}$  be diagonal matrices with time-varying parameters. Let

$$\vartheta(t) = \begin{bmatrix} \vartheta_1(t) \\ \vartheta_2(t) \end{bmatrix} = \begin{bmatrix} \vartheta_1^1(t) & \cdots & \vartheta_1^N(t) \\ \vartheta_2^1(t) & \cdots & \vartheta_2^N(t) \end{bmatrix} \quad (3.37)$$

be a  $2 \times N$  matrix with time varying parameters. Then,

$$\mathbf{v}_R(\mathbf{x}, t, \mathbf{y}_c(t)) = -\vartheta(t)\phi(\mathbf{x}, t) - K_R(t)(\mathbf{x} - \mathbf{y}_c(t)) \quad (3.38)$$

$$\mathbf{v}_M(\mathbf{y}, t, \mathbf{y}_c(t)) = -\mathbf{F}_M(\mathbf{z}, t) - K_M(\mathbf{y} - \mathbf{y}_c(t)). \quad (3.39)$$

The actual controller represented by equation (3.38) is a path following controller with the feedforward-feedback structure. The actual controller contains a feedforward term and feedback terms. The feedforward term is to cancel out flow estimates  $(-\vartheta(t)\phi(\mathbf{x}, t))$ , and the feedback term is a positional negative feedback term including a positional command with time varying parameters  $(-K_R(t)(\mathbf{x} - \mathbf{y}_c(t)))$ ; however, the desired controller represented by equation (3.39) is to cancel out modeled flow term  $(-\mathbf{F}_M(\mathbf{y}, t))$  in addition to fixed gains of feedback term  $(-K_M(t)(\mathbf{y} - \mathbf{y}_c(t)))$ . By plugging equations (3.39) and (2.5) into equation (3.2), the closed loop dynamics of the desired controller is

$$\dot{\mathbf{y}} = -K_M(\mathbf{y} - \mathbf{y}_c(t)). \quad (3.40)$$

The feedback terms enable vehicles to go toward waypoints; the waypoints  $\mathbf{y}_c(t)$  is a time-varying signal that are changed according to missions of the vehicles. Meanwhile, the adaptive controller represented by equation (3.38) includes adaptive parameters despite having the same structure of the desired controller. The closed loop dynamics of the adap-

tive controller is

$$\dot{\mathbf{x}} = (\boldsymbol{\theta} - \boldsymbol{\vartheta}(t))\boldsymbol{\phi}(\mathbf{x}, t) - K_R(t)(\mathbf{x} - \mathbf{y}_c(t)), \quad (3.41)$$

when we plug equations (3.38) and (2.4) into equation (3.1). If  $\mathbf{x}$  and  $\mathbf{y}$  have the same initial condition,  $\boldsymbol{\vartheta}(t)$  converges to  $\boldsymbol{\theta}$ , and  $K_R(t)$  converges to  $K_M$ , the predicted trajectory is followed by the estimated trajectory as seen from closed-loop dynamics. Our goal is to design adaptation laws so that the closed loop dynamics of the adaptive controller can be similar to that of the desired controller. To control the AUV that follows the predicted trajectory in the actual flow field, we first derive controlled Lagrangian prediction error (CLPE) dynamics that models how much the estimated trajectory is deviated from the prediction trajectory. By subtracting equation (3.40) from equation (3.41), CLPE dynamics is represented by

$$\begin{aligned} \dot{\mathbf{e}} &= \dot{\mathbf{x}} - \dot{\mathbf{y}} = (\boldsymbol{\theta} - \boldsymbol{\vartheta}(t))\boldsymbol{\phi}(\mathbf{x}, t) - K_R(t)(\mathbf{x} - \mathbf{y}_c(t)) + K_M(\mathbf{y} - \mathbf{y}_c(t)) \\ &= (\boldsymbol{\theta} - \boldsymbol{\vartheta}(t))\boldsymbol{\phi}(\mathbf{x}, t) + (K_M - K_R(t))(\mathbf{x} - \mathbf{y}_c(t)) - K_M\mathbf{e}. \end{aligned} \quad (3.42)$$

If  $\boldsymbol{\vartheta}(t)$  converges to  $\boldsymbol{\theta}$ , and  $K_R(t)$  converges to  $K_M$ , CLPE goes to zero as time goes by because of positive  $K_M$ , which implies that the estimated trajectory follows the predicted trajectory. In addition, the AUV identifies the actual flow field. Our goal is to design adaptation laws for parameters  $\boldsymbol{\vartheta}$  and  $K_R$  by using CLPE dynamics so that CLPE converges to zero. On the other hand, We derive CLPE dynamics under control input constraints. When we combine equations (3.35) and (3.36), and subtract equation (3.40) from the combination of (3.35) and (3.36), CLPE dynamics becomes

$$\dot{\mathbf{e}} = (\boldsymbol{\theta} - \boldsymbol{\vartheta}(t))\boldsymbol{\phi}(\mathbf{x}, t) + (K_M - K_R(t))(\mathbf{x} - \mathbf{y}_c(t)) - K_M\mathbf{e} + \boldsymbol{\delta}\mathbf{u}. \quad (3.43)$$

$\boldsymbol{\delta}\mathbf{u}$  is viewed as an additional disturbance to the error dynamics of controlled Lagrangian prediction. Equation (3.43) will be used for the proposed adaptation law that enables CLPE is ultimately bounded in spite of the saturation of controllers. To design adaptation laws

for controlled Lagrangian particle tracking, we need some definitions and assumptions.

*Assumption 3.1* The trajectory  $\mathbf{y}_c(t)$  is a persistently exciting signal.

*Remark 3.1*  $\mathbf{y}_c(t)$  is bounded because of persistent excitation.

*Assumption 3.2*  $K_M$ , or the feedback gain matrix of equation (3.40), is positive definite.

*Remark 3.2* The predicted trajectory of AUVs is usually generated from stable closed loop dynamics of equation (3.40). It requires gain matrix  $K_M$  is positive definite.

*Assumption 3.3* We suppose that through-water velocity of the simulated model is not saturated.

*Remark 3.3* We can design controller gain matrix  $K_M$  and sinusoidal function  $\mathbf{y}_c$  to avoid the saturation of control input given modeled flows for the generation of the predicted trajectory of the vehicle. However, through-water velocity of AUVs in the true flow field can be saturated because of unknown true flows.

*Assumption 3.4*  $\|\delta \mathbf{u}\|$  is bounded by  $\delta u_{\max}$ .

*Remark 3.4* When the vehicle controller is not saturated,  $\delta \mathbf{u}$  is zero. However, if the vehicle controller is saturated,  $\delta \mathbf{u}$  is bounded.

Let  $\tilde{\phi}_1 = \begin{bmatrix} \phi_1^1 & \cdots & \phi_1^N \\ 0 & \cdots & 0 \end{bmatrix}$  and  $\tilde{\phi}_2 = \begin{bmatrix} 0 & \cdots & 0 \\ \phi_2^1 & \cdots & \phi_2^N \end{bmatrix}$  be in  $\mathbb{R}^{2 \times N}$ . Let  $\tilde{\mathbf{x}} = \text{diag}\{x_1, x_2\}$ , and  $\tilde{\mathbf{y}}_c = \text{diag}\{y_{1c}, y_{2c}\}$  be diagonal matrices. Let  $w = [\tilde{\phi}_1, \tilde{\phi}_2, \tilde{\mathbf{x}} - \tilde{\mathbf{y}}_c]^\top \in \mathbb{R}^{(2N+2) \times 2}$ . For parameter convergence, we need an assumption on  $w$  as follows:

*Assumption 3.5*  $w$  is persistently exciting. By Definition 2.1, there exists positive definite

matrix  $W(t) \in \mathbb{R}^{(2N+2) \times (2N+2)}$  such that

$$W(t) = \int_t^{t+T} \begin{bmatrix} \phi_1^1 \phi_1^1 & \dots & \phi_1^1 \phi_1^N & 0 & \dots & 0 & \phi_1^1(x_1-y_{c_1}) & 0 \\ \phi_1^2 \phi_1^1 & \dots & \phi_1^2 \phi_1^N & 0 & \dots & 0 & \phi_1^2(x_1-y_{c_1}) & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_1^N \phi_1^1 & \dots & \phi_1^N \phi_1^N & 0 & \dots & 0 & \phi_1^N(x_1-y_{c_1}) & 0 \\ 0 & \dots & 0 & \phi_2^1 \phi_2^1 & \dots & \phi_2^1 \phi_2^N & 0 & \phi_2^1(x_2-y_{c_2}) \\ 0 & \dots & 0 & \phi_2^2 \phi_2^1 & \dots & \phi_2^2 \phi_2^N & 0 & \phi_2^2(x_2-y_{c_2}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & \phi_2^N \phi_2^1 & \dots & \phi_2^N \phi_2^N & 0 & \phi_2^N(x_2-y_{c_2}) \\ (x_1-y_{c_1})\phi_1^1 & \dots & (x_1-y_{c_1})\phi_1^N & 0 & \dots & 0 & (x_1-y_{c_1})^2 & 0 \\ 0 & \dots & 0 & (x_2-y_{c_2})\phi_2^1 & \dots & (x_2-y_{c_2})\phi_2^N & 0 & (x_2-y_{c_2})^2 \end{bmatrix} d\tau, \quad (3.44)$$

where  $\phi_j^i(\mathbf{x}, \tau) = \exp^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|}{2\sigma_i}} \cos(\omega_i \tau + \nu_i)$ ,  $i = 1 \dots N$ ,  $j = 1, 2$ , and  $T > 0$ .

Let  $\bar{K}_R(t) = [K_{R_1}(t), K_{R_2}(t)]^\top$  be two dimensional vectors with time-varying parameters.

Let  $\bar{K}_M = [K_{M_1}, K_{M_2}]^\top$  be two dimensional vectors with desired fixed parameters. Let

$\bar{\vartheta}$ ,  $\bar{\theta}$ , and  $\mathbf{e} \otimes \phi$  be in  $\mathbb{R}^{2N}$ ; that is,  $\bar{\vartheta}(t) = [\vartheta_1^1(t), \dots, \vartheta_1^N(t), \vartheta_2^1(t), \dots, \vartheta_2^N(t)]^\top$ .  $\mathbf{e} \otimes \phi$  and  $\bar{\theta}(t)$  are represented by equations (3.67) and (3.68), respectively, where  $\otimes$  is the Kronecker product. Let  $\tilde{\gamma}$  be any positive constant. We design adaptation laws for time-varying parameters  $\bar{\vartheta}$ ,  $\bar{K}_R$ , and  $\bar{\Gamma}_R$  by the following equations.

$$\dot{\bar{\vartheta}}(t) = \tilde{\gamma} \mathbf{e} \otimes \phi(\mathbf{x}, t) \quad (3.45)$$

$$\dot{\bar{K}}_R(t) = \tilde{\gamma}(\bar{\mathbf{x}} - \bar{\mathbf{y}}_c)^\top \mathbf{e}. \quad (3.46)$$

We prove error and parameter convergence to show that the vehicle controlled by the feedback controller moves along the predicted trajectory. Because zero CLPE means that the estimated trajectory follows the predicted trajectory, we prove CLPE convergence by using the adaptation laws represented by equations (3.45) and (3.46) in Theorem 3.6.

*Theorem 3.6* Under Assumptions 3.1 and 3.2, and using equations (3.45) and (3.46), CLPE converges to zero when time goes to infinity; that is,  $\mathbf{e}(t) \rightarrow \vec{0}$  as  $t \rightarrow \infty$ .

*Proof.* Consider a candidate Lyapunov function:

$$V(\mathbf{e}, \bar{\vartheta}, \bar{K}_R) = \frac{1}{2} \left\{ \mathbf{e}^\top \mathbf{e} + \frac{1}{\bar{\gamma}} (\bar{\theta} - \bar{\vartheta})^\top (\bar{\theta} - \bar{\vartheta}) + \frac{1}{\bar{\gamma}} (\bar{K}_M - \bar{K}_R)^\top (\bar{K}_M - \bar{K}_R) \right\}. \quad (3.47)$$

The derivative of  $V$  is

$$\dot{V} = -\mathbf{e}^\top K_M \mathbf{e} + (\bar{\theta} - \bar{\vartheta})^\top \left( \mathbf{e} \otimes \phi(\mathbf{x}, t)^\top - \frac{1}{\bar{\gamma}} \dot{\bar{\alpha}}_R \right) + (\bar{K}_M - \bar{K}_R)^\top \left( (\tilde{\mathbf{x}} - \tilde{\mathbf{y}}_c)^\top \mathbf{e} - \frac{1}{\bar{\gamma}} \dot{\bar{K}}_R \right). \quad (3.48)$$

By Assumption 3.2, Remark 3.2, and using equation (3.45) and (3.46),  $\dot{V} = -\mathbf{e}^\top K_M \mathbf{e} \leq 0$ .  $\dot{V}$  is negative semi-definite and this implies  $\mathbf{e}$ ,  $\bar{\vartheta}$ , and  $\bar{K}_R$  are bounded. In addition,  $\dot{V} = -2\mathbf{e}^\top K_M \dot{\mathbf{e}} = -2\mathbf{e}^\top K_M \{ (\theta - \vartheta) \phi(\mathbf{x}, t) + (K_M - K_R)(\mathbf{x} - \mathbf{y}_c) - K_M \mathbf{e} \}$ . By Assumption 3.1,  $\mathbf{y}_c$  is bounded, and  $\mathbf{y}$  is bounded because equation (3.40) represents linear systems.  $\mathbf{x}$  is bounded because  $\mathbf{x} = \mathbf{e} + \mathbf{y}$ . In addition,  $\vartheta$  and  $K_R$  are bounded. Thus,  $\dot{V}$  is bounded, and hence  $\dot{V}$  is uniformly continuous. By Lemma 2.4,  $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$ . Since  $K_M$  is the diagonal matrix,  $\mathbf{e}(t) \rightarrow \vec{0}$  as  $t \rightarrow \infty$   $\square$

Flow velocity may not be accurately estimated in that multiple parameters that represent flow are estimated from trajectory information. When the feedback controller cancels actual flow by using estimated flow, inaccurate estimated flow makes CLPE be non-zero. We need to show that all the estimated parameters converge to true parameters; Theorem 3.7 shows parameters convergence as follows.

*Theorem 3.7* Under the same setting of Theorem 3.6,  $\bar{\vartheta}$  and  $\bar{K}_R$  converges to  $\bar{\theta}$  and  $\bar{K}_M$ , respectively; that is,  $\bar{\vartheta}(t) \rightarrow \bar{\theta}$  and  $K_R(t) \rightarrow K_M$  as  $t \rightarrow \infty$ .

*Proof.* Let  $\tilde{\eta}_1$ ,  $\tilde{\eta}_2$ , and  $\tilde{\eta}_3$  be  $(\theta_1 - \vartheta_1)^\top$ ,  $(\theta_2 - \vartheta_2)^\top$ , and  $(\bar{K}_M - \bar{K}_R)$ , respectively. Let  $\tilde{\phi}_1 = \begin{bmatrix} \phi_1^1 & \cdots & \phi_1^N \\ 0 & \cdots & 0 \end{bmatrix}$  and  $\tilde{\phi}_2 = \begin{bmatrix} 0 & \cdots & 0 \\ \phi_2^1 & \cdots & \phi_2^N \end{bmatrix}$  be in  $\mathbb{R}^{2 \times N}$ . We rewrite equation (3.42) using  $\tilde{\eta}_1$ ,  $\tilde{\eta}_2$ , and  $\tilde{\eta}_3$  as follows:

$$\dot{\mathbf{e}} = \tilde{\phi}_1(\mathbf{x}, t) \tilde{\eta}_1 + \tilde{\phi}_2(\mathbf{x}, t) \tilde{\eta}_2 + (\tilde{\mathbf{x}} - \tilde{\mathbf{y}}_c) \tilde{\eta}_3 - K_M \mathbf{e}. \quad (3.49)$$

Because equation (3.49) has the same form of equation (2.28), and  $w$  is persistently exciting by Assumption 3.5, we use the same proof of Theorem 2.3. Hence,  $\bar{\vartheta}$  and  $\bar{K}_R$  converge to  $\bar{\theta}$  and  $\bar{K}_M$ , respectively.  $\square$

### 3.3.2 Input constraints for adaptive control algorithm

Since equation (3.43) has one more term  $\delta \mathbf{u} = \mathbf{u} - \mathbf{v}_R$ , which shows the discrepancy between input and output of the saturator than equation (3.42). we reject additional disturbance  $\delta \mathbf{u}$  from saturation in equation (3.43) by using the same scheme in Section 3.3.6. We generate additional signal  $\mathbf{e}_\delta$  governed by a differential equation as follows:

$$\dot{\mathbf{e}}_\delta = -K_M \mathbf{e}_\delta + \Lambda \delta \mathbf{u}, \quad (3.50)$$

where  $\Lambda = \text{diag}\{\Lambda_1, \Lambda_2\}$  with time varying parameters. Let  $\varepsilon = \mathbf{e} - \mathbf{e}_\delta$  be the difference between CLPE and the additional signal. When subtracting (3.50) from (3.43), we have

$$\begin{aligned} \dot{\varepsilon} &= \dot{\mathbf{e}} - \dot{\mathbf{e}}_\delta \\ &= (\theta - \alpha_R) \phi(\mathbf{x}, t) + (K_M - K_R)(\mathbf{x} - \mathbf{y}_c) - K_M \mathbf{e} + (I - \Lambda) \delta \mathbf{u}. \end{aligned} \quad (3.51)$$

$\Lambda$  is designed to be  $I$ . Let  $\delta \mathbf{u} = [\delta u_1, \delta u_2]^\top$  be two dimensional vectors. We design adaptation laws for time-varying parameters  $\bar{\vartheta}$  and  $\bar{K}_R$  by the following equations.

$$\dot{\bar{\vartheta}} = \gamma \varepsilon \otimes \phi(\mathbf{x}, t) \quad (3.52)$$

$$\dot{\bar{K}}_R = \gamma (\tilde{\mathbf{x}} - \tilde{\mathbf{y}}_c)^\top \varepsilon. \quad (3.53)$$

*Theorem 3.8* Under Assumptions 3.1, 3.2, 3.3, 3.4, and adaptation laws (3.52) and (3.53), CLPE is ultimately bounded.

$$\|\mathbf{e}\| \leq \frac{\lambda_{\max}(\Lambda) \|\delta \mathbf{u}\|}{\xi}, \quad (3.54)$$



where the positive constant  $\xi < 1$ .

*Proof.* Let  $V_{\mathbf{e}_\delta} = \frac{1}{2}\mathbf{e}_\delta^\top K_M^{-1}\mathbf{e}_\delta$ . The derivative of  $V_{\mathbf{e}_\delta}$  is  $\dot{V}_{\mathbf{e}_\delta} = -\mathbf{e}_\delta^\top \dot{\mathbf{e}}_\delta + \mathbf{e}_\delta^\top \Lambda \delta \mathbf{u}$ . Then,  $\dot{V}_{\mathbf{e}_\delta} \leq -(1-\xi)\|\mathbf{e}_\delta\|^2 - \xi\|\mathbf{e}_\delta\|^2 + \|\mathbf{e}_\delta\|\lambda_{\max}(\Lambda)\|\delta \mathbf{u}\|$ . When  $\|\mathbf{e}_\delta\| \geq \frac{\lambda_{\max}(\Lambda)\|\delta \mathbf{u}\|}{\xi}$  given positive constant  $\xi < 1$ ,  $\dot{V} \leq -(1-\xi)\|\mathbf{e}_\delta\|^2$ . This means  $\dot{V}$  is not positive. Thus,  $\|\mathbf{e}_\delta\| \leq \frac{\lambda_{\max}(\Lambda)\|\delta \mathbf{u}\|}{\xi}$ . We will show  $\varepsilon$  goes to zero when time goes infinity by the following candidate Lyapunov function.

$$V(\varepsilon, \bar{\vartheta}, \bar{K}_R) = \frac{1}{2} \left\{ \varepsilon^\top \varepsilon + \frac{1}{\gamma} (\bar{\theta} - \bar{\vartheta})^\top (\bar{\theta} - \bar{\vartheta}) + \frac{1}{\gamma} (\bar{K}_R - \bar{K}_M)^\top (\bar{K}_R - \bar{K}_M) \right\}, \quad (3.55)$$

By Assumption 3.2 and using equations (3.52) and (3.53),  $\dot{V} = -\varepsilon^\top K_M \varepsilon \leq 0$ .  $\dot{V}$  is negative semi-definite and this implies  $\varepsilon$ ,  $\bar{\alpha}_R$ , and  $\bar{K}_R$  are bounded. In addition,  $\dot{V} = -2\varepsilon^\top K_M \dot{\varepsilon} = -2\varepsilon^\top K_M \{(\bar{\theta} - \bar{\vartheta})\phi(\mathbf{x}, t) + (K_M - K_R)(\mathbf{x} - \mathbf{y}_c) - K_M \varepsilon\}$ . By Assumption 3.1,  $\mathbf{y}_c$  is bounded, and  $\mathbf{y}$  is bounded because of linear system of equation (3.65). Since  $\mathbf{e} = \mathbf{e}_\delta + \varepsilon$ ,  $\mathbf{e}$  is bounded. This implies that  $\mathbf{x}$  is bounded. In addition,  $\bar{\vartheta}$  and  $K_R$  are bounded. Thus,  $\dot{V}$  is bounded, and then  $\dot{V}$  is uniformly continuous. By Lemma 2.4,  $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$ . Since  $K_M$  is the diagonal matrix,  $\mathbf{e} \rightarrow \mathbf{e}_\delta$  when  $t \rightarrow \infty$ . Thus, CLPE is ultimately bounded.  $\square$

### 3.3.3 Inaccuracy in flow modeling for adaptive control algorithm

Although the basis functions well capture the spatial variability of true flows in a specific region, the functions still include deterministic errors induced by the variability out of the region. In this section, we address the robustness of the proposed adaptive control algorithm.

We show the boundedness of CLPE when the true flow model has deterministic disturbances such as unstructured uncertainties. We assume  $\mathbf{F}_R(\mathbf{x}, t) = \theta \phi(\mathbf{x}, t) + \Delta$ , where  $\|\Delta\|$

is bounded by  $\Delta_{\max} \in \mathbb{R}$ . Then,

$$\dot{\mathbf{e}} = (\boldsymbol{\theta} - \boldsymbol{\alpha}_R) \phi(\mathbf{x}, t) + (K_M - K_R)(\mathbf{x} - \mathbf{y}_c) - K_M \mathbf{e} + \Delta. \quad (3.56)$$

*Theorem 3.9* Under the same setting of Theorem 3.6, CLLE is ultimately bounded.

$$\|\mathbf{e}\| \leq \frac{1}{\tilde{\beta}} \|\Delta\|, \quad (3.57)$$

where the positive constant  $\tilde{\beta} < \lambda_{\min}(K_M)$ .

*Proof.* Let  $V$  be the Lyapunov function represented by equation (3.47). By using equation (3.56), the derivate of  $V$  is

$$\dot{V} = -\mathbf{e}^T K_M \mathbf{e} + (\bar{\boldsymbol{\theta}} - \bar{\boldsymbol{\vartheta}})^\top \left( \tilde{\boldsymbol{\phi}}(\mathbf{x}, t)^\top \mathbf{e} - \frac{1}{\tilde{\gamma}} \dot{\bar{\boldsymbol{\vartheta}}} \right) + (\bar{K}_M - \bar{K}_R)^\top \left( (\tilde{\mathbf{x}} - \tilde{\mathbf{y}}_c)^\top \mathbf{e} - \frac{1}{\tilde{\gamma}} \dot{\bar{K}}_R \right) + \mathbf{e}^\top \Delta. \quad (3.58)$$

Then, we plug the adaptive law represented by equations (3.45) and (3.46) into equation (3.58). Then,

$$\begin{aligned} \dot{V} &= -\mathbf{e}^T K_M \mathbf{e} + \mathbf{e}^\top \Delta \\ &\leq -\lambda_{\min}(K_M) \mathbf{e}^\top \mathbf{e} + \mathbf{e}^\top \Delta \\ &\leq -\lambda_{\min}(K_M) \|\mathbf{e}\|^2 + \|\mathbf{e}\| \|\Delta\| \\ &\leq -(\lambda_{\min}(K_M) - \beta) \|\mathbf{e}\|^2 + \|\mathbf{e}\| \|\Delta\| - \beta \|\mathbf{e}\|^2 \end{aligned} \quad (3.59)$$

When  $\|\mathbf{e}\| \geq \frac{1}{\tilde{\beta}} \|\Delta\|$  given positive constant  $\beta < \lambda_{\min}(K)$ ,  $\dot{V} \leq -(\lambda_{\min}(K_M) - \beta) \|\mathbf{e}\|^2$ , which means  $\dot{V}$  is negative definite. Thus, CLPE is ultimately bounded. The bound of CLPE is  $\|\mathbf{e}\| \leq \frac{1}{\tilde{\beta}} \|\Delta\|$ .  $\square$

### 3.3.4 Controlled Lagrangian Localization Error

We first derived controlled Lagrangian localization error (CLLE) dynamics that models how much the estimated trajectory is deviated from the identified trajectory. By subtracting

equation (3.65) from equation (3.64), CLLE dynamics is represented by

$$\begin{aligned}
\dot{\mathbf{e}} &= \dot{\mathbf{x}} - \dot{\mathbf{z}} \\
&= (\theta - \alpha) \phi(\mathbf{x}, t) - (\xi(t) - \alpha) \phi(\mathbf{z}, t) - K_D(\mathbf{x} - \mathbf{y}_c(t)) + K_L(t)(\mathbf{z} - \mathbf{y}_c(t)) - \zeta(t) \quad (3.60) \\
&= (\theta - \alpha) \phi(\mathbf{x}, t) - (\xi(t) - \alpha) \phi(\mathbf{z}, t) + (K_L(t) - K_D)(\mathbf{z} - \mathbf{y}_c(t)) - K_D \mathbf{e} - \zeta(t).
\end{aligned}$$

For example, if  $\xi(t)$  converges to  $\alpha$ ,  $K_L(t)$  converges to  $K_D$ ,  $\zeta(t) = (\theta - \alpha)\phi(\mathbf{x}, t)$ , and  $K_D$  is positive definite, CLLE goes to zero over time, which implies that the identified trajectory follows the estimated trajectory. Note that we cannot use  $\zeta(t) = (\theta - \alpha)\phi(\mathbf{x}, t)$  because of unknown  $\theta$ .

Then, we derive CLLE dynamics under control input constraints. Let  $\delta \mathbf{u} = \mathbf{u}(\mathbf{x}, t, \mathbf{y}_c(t)) - \mathbf{v}_R(\mathbf{x}, t, \mathbf{y}_c(t))$  be the saturation term of the vehicle controller. We combine equations (3.35) and (3.36), and subtract (3.65) from the combination of (3.35) and (3.36) to obtain CLLE dynamics. It becomes

$$\dot{\mathbf{e}} = (\theta - \alpha) \phi(\mathbf{x}, t) - (\xi(t) - \alpha) \phi(\mathbf{z}, t) + (K_L(t) - K_D)(\mathbf{z} - \mathbf{y}_c(t)) - K_D \mathbf{e} + \delta \mathbf{u} - \zeta(t), \quad (3.61)$$

We view  $\delta \mathbf{u}$  as an additional disturbance to CLLE dynamics. Equation (3.61) is used for the proposed controller that ensures CLLE to be ultimately bounded in spite of saturation of the vehicle controller.

### 3.3.5 Adaptive Learning Algorithm

Let gains  $K_L(t) = \text{diag}\{K_{L_1}(t), K_{L_2}(t)\}$  be diagonal matrices with time varying parameters. Let  $K_D = \text{diag}\{K_{D_1}, K_{D_2}\}$  be diagonal matrices with desired fixed parameters. Let  $\zeta(t) \in \mathbb{R}^2$

be a learning injection parameter. Then,

$$\mathbf{v}_R(\mathbf{x}, t, \mathbf{y}_c(t)) = -\mathbf{F}_M(\mathbf{x}, t) - K_D(\mathbf{x} - \mathbf{y}_c(t)) \quad (3.62)$$

$$\mathbf{v}_L(\mathbf{z}, t, \mathbf{y}_c(t)) = -\mathbf{F}_M(\mathbf{z}, t) - K_L(t)(\mathbf{z} - \mathbf{y}_c(t)) + \boldsymbol{\zeta}(t). \quad (3.63)$$

The vehicle controller represented by equation (3.62) is similar to feedforward-feedback controllers for the applications of wind turbines and hard disks [51, 52]. The feedforward term is a flow canceling term ( $-\mathbf{F}_M(\mathbf{x}, t)$ ), which is used to cancel estimated flow, the feedback terms have a positional negative feedback term including a positional command ( $-K_D(\mathbf{x} - \mathbf{y}_c(t))$ ). A controller that identifies vehicle motion, represented by equation (3.63), is designed for the learning algorithm. The controller for the learning algorithm has the same structure of the vehicle controller; but, all the parameters in the feedforward and feedback terms are time-varying, and an additional term called learning injection is included. By plugging equations (2.4), (2.5), and (3.62) into equation (3.1), the closed loop vehicle dynamics including the vehicle controller is

$$\dot{\mathbf{x}} = (\boldsymbol{\theta} - \boldsymbol{\alpha})\boldsymbol{\phi}(\mathbf{x}, t) - K_D(\mathbf{x} - \mathbf{y}_c(t)). \quad (3.64)$$

Meanwhile, the closed loop vehicle dynamics that includes a controller identifying vehicle motion represented by equation (3.63) is

$$\dot{\mathbf{z}} = (\boldsymbol{\xi}(t) - \boldsymbol{\alpha})\boldsymbol{\phi}(\mathbf{z}, t) - K_L(t)(\mathbf{z} - \mathbf{y}_c(t)) + \boldsymbol{\zeta}(t), \quad (3.65)$$

which is derived by plugging equations (2.5), (2.8), and (3.63) into equation (3.3).

Our goal is to design the learning algorithm for updating parameters  $\boldsymbol{\xi}(t)$  and  $K_L(t)$  by using CLLE dynamics so that CLLE converges to zero. To design the adaptive learning algorithm for controlled Lagrangian particles, we need the following definitions and assumptions.

*Assumption 3.6* The feedback gain matrix of equation (3.64),  $K_D$  is positive definite and diagonal.

*Remark 3.6* Positive definite matrix  $K_D$  is required for the vehicle controller with negative state feedback.

Let  $\bar{K}_L(t) = [K_{L_1}(t), K_{L_2}(t)]^\top$  be time-varying vectors of where  $\bar{K}_L(t)$  is feedback gain. Let  $\bar{K}_D = [K_{D_1}, K_{D_2}]^\top$  be two dimensional vectors with desired fixed elements. Let  $\tilde{\mathbf{z}} = \text{diag}\{z_1, z_2\}$ , and  $\tilde{\mathbf{y}}_c = \text{diag}\{y_{1c}, y_{2c}\}$  be diagonal matrices. Let  $\bar{\xi}$ ,  $\bar{\theta}$ , and  $\mathbf{e} \otimes \phi \in \mathbb{R}^{2N}$  be row vectors. That is,

$$\bar{\xi}(t) = [\xi_1^1(t), \dots, \xi_1^N(t), \xi_2^1(t), \dots, \xi_2^N(t)]^\top \quad (3.66)$$

$$\mathbf{e} \otimes \phi = [e_1\phi^1, \dots, e_1\phi^N, e_2\phi^1, \dots, e_2\phi^N]^\top \quad (3.67)$$

$$\bar{\theta}(t) = [\theta_1^1(t), \dots, \theta_1^N(t), \theta_2^1(t), \dots, \theta_2^N(t)]^\top, \quad (3.68)$$

where  $\otimes$  is the Kronecker product. Then, we design learning injection parameter  $\zeta(t)$  as follows:

$$\zeta(t) = (\xi(t) - \alpha)\phi(\mathbf{x}, t) - (\xi(t) - \alpha)\phi(\mathbf{z}, t). \quad (3.69)$$

Time-varying parameters  $\bar{\xi}$  and  $\bar{K}_L$  are updated according to the following rules:

$$\dot{\bar{\xi}}(t) = \gamma \mathbf{e} \otimes \phi(\mathbf{x}, t) \quad (3.70)$$

$$\dot{\bar{K}}_L(t) = -\gamma(\tilde{\mathbf{z}} - \tilde{\mathbf{y}}_c)^\top \mathbf{e}, \quad (3.71)$$

where  $\gamma$  be any positive constant. We will prove CLLE and parameters convergence to identify vehicle motion controlled by the feedback controller. Since CLLE is a measure that represents the deviation of the identified trajectory from the estimated trajectory, CLLE converging to zero indicates that the identified trajectory follows the estimated trajectory. Thus, we prove CLLE convergence as follows.

*Theorem 3.10* Under Assumptions 3.1 and 3.6, and using equations (3.69), (3.70), and

(3.71), CLLE converges to zero when time goes to infinity; that is,  $\mathbf{e}(t) \rightarrow \vec{0}$  as  $t \rightarrow \infty$ .

*Proof.* Consider a candidate Lyapunov function:

$$V(\mathbf{e}, \xi, \bar{K}_L) = \frac{1}{2} \left\{ \mathbf{e}^\top \mathbf{e} + \frac{1}{\gamma} (\theta_1 - \xi_1(t)) (\theta_1 - \xi_1(t))^\top + \frac{1}{\gamma} (\theta_2 - \xi_2(t)) (\theta_2 - \xi_2(t))^\top + \frac{1}{\gamma} (\bar{K}_L(t) - \bar{K}_D)^\top (\bar{K}_L(t) - \bar{K}_D) \right\}. \quad (3.72)$$

The time derivative of  $V$  is

$$\begin{aligned} \dot{V} = & -\mathbf{e}^\top K_D \mathbf{e} + \mathbf{e}^\top (\theta - \xi(t)) \phi(\mathbf{x}, t) + \mathbf{e}^\top (\xi(t) - \alpha) \phi(\mathbf{x}, t) - \mathbf{e}^\top (\xi(t) - \alpha) \phi(\mathbf{z}, t) \\ & - \mathbf{e}^\top \zeta(t) + (\bar{K}_L(t) - \bar{K}_D)^\top \left( \frac{1}{\gamma} \dot{\bar{K}}_L(t) + (\bar{\mathbf{z}} - \bar{\mathbf{y}}_c)^\top \mathbf{e} \right) - \frac{1}{\gamma} (\bar{\theta} - \bar{\xi}(t)) \dot{\bar{\xi}}(t). \end{aligned} \quad (3.73)$$

We know that  $\mathbf{e}^\top (\theta - \xi(t)) \phi(\mathbf{x}, t) = (\bar{\theta} - \bar{\xi}(t)) \mathbf{e} \otimes \phi(\mathbf{x}, t)$ . Then, under Assumption 3.6 and Remark 3.6, using equation (3.69), (3.70), and (3.71),

$$\dot{V} = -\mathbf{e}^\top K_D \mathbf{e} \leq 0. \quad (3.74)$$

$\dot{V}$  is negative semi-definite and this implies  $\mathbf{e}$ ,  $\xi(t)$ , and  $\bar{K}_L(t)$  are bounded. In addition, the second order time derivative of  $V$  satisfies

$$\begin{aligned} \ddot{V} = & -2\mathbf{e}^\top K_D \dot{\mathbf{e}} = -2\mathbf{e}^\top K_D \{ (\theta - \xi(t)) \phi(\mathbf{x}, t) - K_D \mathbf{e} + (K_L(t) - K_D)(\mathbf{z} - \mathbf{y}_c) \\ & - \zeta(t) + (\xi(t) - \alpha) \phi(\mathbf{x}, t) - (\xi(t) - \alpha) \phi(\mathbf{z}, t) \}. \end{aligned} \quad (3.75)$$

From equation (3.69),  $\zeta(t)$  is bounded. By Assumption 3.1 and Remark 3.1,  $\mathbf{y}_c$  is bounded, and  $\mathbf{z}$  is bounded because equation (3.65) represents linear systems with sinusoidal inputs. In addition,  $K_L(t)$  is bounded. Thus,  $\ddot{V}$  is bounded, and hence  $\dot{V}$  is uniformly continuous. By Lemma 2.4,  $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$ . Since  $K_D$  is the diagonal matrix,  $\mathbf{e}(t) \rightarrow \vec{0}$  as  $t \rightarrow \infty$ .  $\square$

Although CLLE converges to zero, estimation error of through-water velocity and flow velocity may be large due to the identification of multiple flow parameters from one type of

trajectory information (i.e. the estimated trajectory). Therefore, we need to prove that all the identified parameters converges to true parameters so that the vehicle motion controlled by the feedback controller is accurately identified.

*Theorem 3.11* Under the same setting of Theorem 3.10,  $\bar{\xi}(t)$  and  $\bar{K}_L(t)$  converge to  $\bar{\theta}$ ,  $\bar{K}_D$ , respectively; that is,  $\bar{\xi}(t) \rightarrow \bar{\theta}$ , and  $\bar{K}_L(t) \rightarrow \bar{K}_D$  as  $t \rightarrow \infty$

*Proof.* Let  $\eta_1$ ,  $\eta_2$ , and  $\eta_3$  be  $(\theta_1 - \xi_1(t))^\top$ ,  $(\theta_2 - \xi_2(t))^\top$ , and  $(\bar{K}_L(t) - \bar{K}_D)$ , respectively. Let  $\tilde{\phi}_1 = \begin{bmatrix} \phi_1^1 & \cdots & \phi_1^N \\ 0 & \cdots & 0 \end{bmatrix}$  and  $\tilde{\phi}_2 = \begin{bmatrix} 0 & \cdots & 0 \\ \phi_2^1 & \cdots & \phi_2^N \end{bmatrix}$  be in  $\mathbb{R}^{2 \times N}$ . We rewrite equation (3.60) using equation  $\eta_1$ ,  $\eta_2$ ,  $\eta_3$ , and  $\eta_4$  as follows:

$$\dot{\mathbf{e}} = \tilde{\phi}_1(\mathbf{x}, t)\eta_1 + \tilde{\phi}_2(\mathbf{x}, t)\eta_2 + (\tilde{\mathbf{z}} - \tilde{\mathbf{y}}_c)\eta_3 - K_D \mathbf{e}. \quad (3.76)$$

Because equation (3.76) has the same form of equation (2.28), and  $w$  is persistently exciting by Assumption 3.5, we use the proof of Theorem 2.3. Thus,  $\bar{\xi}(t)$  and  $\bar{K}_L(t)$  converge to  $\bar{\theta}$  and  $\bar{K}_D$ , respectively. □

### 3.3.6 Input constraints for adaptive learning algorithm

The AUVs have limited control energy for controlling their velocities. The maximum control power generated from motors and thrusts is restricted by their hardware configuration. This induces constraint to controlling AUVs in the ocean.

Since equation (3.61) includes one saturated term represented by  $\delta \mathbf{u} = \mathbf{u} - \mathbf{v}_R$ , which shows the discrepancy between input and output of the saturator, we reject additional disturbance  $\delta \mathbf{u}$  from saturation in equation (3.61) by using a scheme in [45]. We generate additional signal  $\mathbf{e}_\delta$  governed by a differential equation as follows:

$$\dot{\mathbf{e}}_\delta = -K_D \mathbf{e}_\delta + \Lambda \delta \mathbf{u}, \quad (3.77)$$

where  $\Lambda = \text{diag}\{\Lambda_1, \Lambda_2\}$  is the matrix with parameters that we design. Those parameters are determined to reject the disturbance. Let  $\varepsilon = \mathbf{e} - \mathbf{e}_\delta$  be the difference between CLLE and the additional signal. When subtracting (3.77) from (3.61), we have

$$\begin{aligned}\dot{\varepsilon} &= \dot{\mathbf{e}} - \dot{\mathbf{e}}_\delta \\ &= (\theta - \alpha)\phi(\mathbf{x}, t) - (\xi(t) - \alpha)\phi(\mathbf{z}, t) + (K_L(t) - K_D)(\mathbf{z} - \mathbf{y}_c(t)) \\ &\quad - K_D\varepsilon + (I - \Lambda(t))\delta\mathbf{u} - \zeta(t).\end{aligned}$$

From equation (3.69), we obtain

$$\dot{\varepsilon} = (\theta - \xi(t))\phi(\mathbf{x}, t) + (K_L(t) - K_D)(\mathbf{z} - \mathbf{y}_c(t)) - K_D\varepsilon + (I - \Lambda)\delta\mathbf{u}. \quad (3.78)$$

$\Lambda$  is designed to be  $I$ .  $\delta\mathbf{u} = [\delta u_1, \delta u_2]^\top$  be two dimensional vectors. With learning parameter injection represented by (3.69), we design the updating rules for time-varying parameters  $\bar{\xi}$  and  $\bar{K}_L$  by the following equations.

$$\dot{\bar{\xi}}(t) = \gamma\varepsilon \otimes \phi(\mathbf{x}, t) \quad (3.79)$$

$$\dot{\bar{K}}_L(t) = -\gamma(\tilde{\mathbf{z}} - \tilde{\mathbf{y}}_c)^\top \varepsilon \quad (3.80)$$

For the reliability of the proposed updating rules when saturation occurs, we show that CLLE is bounded as follows.

*Theorem 3.12* Under Assumptions 3.1-3.4, and the update rules (3.79) and (3.80), CLLE is ultimately bounded.

$$\|\mathbf{e}\| \leq \frac{\lambda_{\max}(\Lambda)\|\delta\mathbf{u}\|}{\rho}, \quad (3.81)$$

where the positive constant  $\rho < 1$ .

*Proof.* Let  $V_{\mathbf{e}_\delta} = \frac{1}{2}\mathbf{e}_\delta^\top K_D^{-1}\mathbf{e}_\delta$ . The derivative of  $V_{\mathbf{e}_\delta}$  is  $\dot{V}_{\mathbf{e}_\delta} = -\mathbf{e}_\delta^\top \mathbf{e}_\delta + \mathbf{e}_\delta^\top \Lambda \delta\mathbf{u}$ . Then,  $\dot{V}_{\mathbf{e}_\delta} \leq$



$-(1-\rho)\|\mathbf{e}_\delta\|^2 - \rho\|\mathbf{e}_\delta\|^2 + \|\mathbf{e}_\delta\|\lambda_{\max}(\Lambda)\|\delta\mathbf{u}\|$ . When  $\|\mathbf{e}_\delta\| \geq \frac{\lambda_{\max}(\Lambda)\|\delta\mathbf{u}\|}{\rho}$  given positive constant  $\rho < 1$ ,  $\dot{V} \leq -(1-\rho)\|\mathbf{e}_\delta\|^2$ . This means that  $\dot{V}$  is not positive. Thus,  $\|\mathbf{e}_\delta\| \leq \frac{\lambda_{\max}(\Lambda)\|\delta\mathbf{u}\|}{\rho}$ .

We will show  $\varepsilon$  goes to zero as time goes to infinity by the following candidate Lyapunov function.

$$\begin{aligned} V(\varepsilon, \bar{\xi}, \bar{K}_L) &= \frac{1}{2} \left\{ \varepsilon^\top \varepsilon + \frac{1}{\gamma} (\bar{\theta}_1 - \xi_1(t))^\top (\bar{\theta}_1 - \xi_1(t)) + \frac{1}{\gamma} (\bar{\theta}_2 - \xi_2(t))^\top (\bar{\theta}_2 - \xi_2(t)) \right. \\ &\quad \left. + \frac{1}{\gamma} (\bar{K}_L - \bar{K}_D)^\top (\bar{K}_L - \bar{K}_D) \right\}, \end{aligned} \quad (3.82)$$

By Assumption 3.6 and using equations (3.79) and (3.80),  $\dot{V} = -\varepsilon^\top K_D \varepsilon \leq 0$ .  $\dot{V}$  is negative semi-definite and this implies  $\varepsilon$ ,  $\bar{\alpha}$ , and  $\bar{K}_L$  are bounded. In addition,  $\dot{V} = -2\varepsilon^\top K_D \dot{\varepsilon} = -2\varepsilon^\top K_D \{(\theta - \xi(t))\phi(\mathbf{x}, t) + (K_L(t) - K_D)(\mathbf{z} - \mathbf{y}_c) - K_D \varepsilon\}$ . By Assumption 3.1,  $\mathbf{y}_c$  is bounded, and  $\mathbf{z}$  is bounded because of linear system of equation (3.65). Since  $\mathbf{e} = \mathbf{e}_\delta + \varepsilon$ ,  $\mathbf{e}$  is bounded. In addition,  $\xi(t)$  and  $K_L(t)$  is bounded. Thus,  $\dot{V}$  is bounded, and then  $\dot{V}$  is uniformly continuous. By Lemma 2.4,  $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$ . Since  $K_D$  is the diagonal matrix,  $\varepsilon \rightarrow \bar{0}$  when  $t \rightarrow \infty$ . This shows  $\mathbf{e} \rightarrow \mathbf{e}_\delta$  when  $t \rightarrow \infty$ . Thus, CLLE is ultimately bounded.  $\square$

### 3.3.7 Inaccuracy in flow modeling for adaptive learning algorithm

Although the basis functions represented by equation (2.8) well capture the spatial variability of actual flows in a specific region, the functions still include deterministic errors induced by the variability out of the region. In this section, we address the robustness of the proposed adaptive learning algorithm.

We show the boundedness of CLLE when the actual flow model has unknown disturbances such as unstructured uncertainties. We assume  $\mathbf{F}_R(\mathbf{x}, t) = \theta\phi(\mathbf{x}, t) + \Delta$ , where  $\|\Delta\|$

is bounded by  $\Delta_{\max} \in \mathbb{R}$ . Then,

$$\dot{\mathbf{e}} = (\theta - \xi(t)) \phi(\mathbf{x}, t) + (K_L(t) - K_D)(\mathbf{z} - \mathbf{y}_c) - K_D \mathbf{e} + \Delta. \quad (3.83)$$

*Theorem 3.13* Under the same setting of Theorem 3.10, CLLE is ultimately bounded.

$$\|\mathbf{e}\| \leq \frac{1}{\beta} \|\Delta\|, \quad (3.84)$$

where the positive constant  $\beta < \lambda_{\min}(K_D)$ .

*Proof.* Let  $V$  be the Lyapunov function represented by equation (3.72). By using equation (3.83), the derivate of  $V$  is

$$\begin{aligned} \dot{V} = & -\mathbf{e}^T K_D \mathbf{e} + \mathbf{e}^\top \Delta + (\bar{K}_L(t) - \bar{K}_D)^\top \left( \frac{1}{\gamma} \dot{\bar{K}}_L(t) + (\tilde{\mathbf{z}} - \tilde{\mathbf{y}}_c)^\top \mathbf{e} \right) \\ & + \frac{1}{\gamma} (\bar{\theta} - \bar{\xi}(t)) \left( \frac{1}{\gamma} \dot{\bar{\xi}}(t) - \mathbf{e} \otimes \phi(\mathbf{x}, t) \right) \end{aligned} \quad (3.85)$$

Then, we plug the updating rules represented by equations (3.70) and (3.71) into equation (3.85). Then,

$$\begin{aligned} \dot{V} = & -\mathbf{e}^T K_D \mathbf{e} + \mathbf{e}^\top \Delta \leq -\lambda_{\min}(K_D) \mathbf{e}^\top \mathbf{e} + \mathbf{e}^\top \Delta \leq -\lambda_{\min}(K_D) \|\mathbf{e}\|^2 + \|\mathbf{e}\| \|\Delta\| \\ & \leq -(\lambda_{\min}(K_D) - \beta) \|\mathbf{e}\|^2 + \|\mathbf{e}\| \|\Delta\| - \beta \|\mathbf{e}\|^2. \end{aligned} \quad (3.86)$$

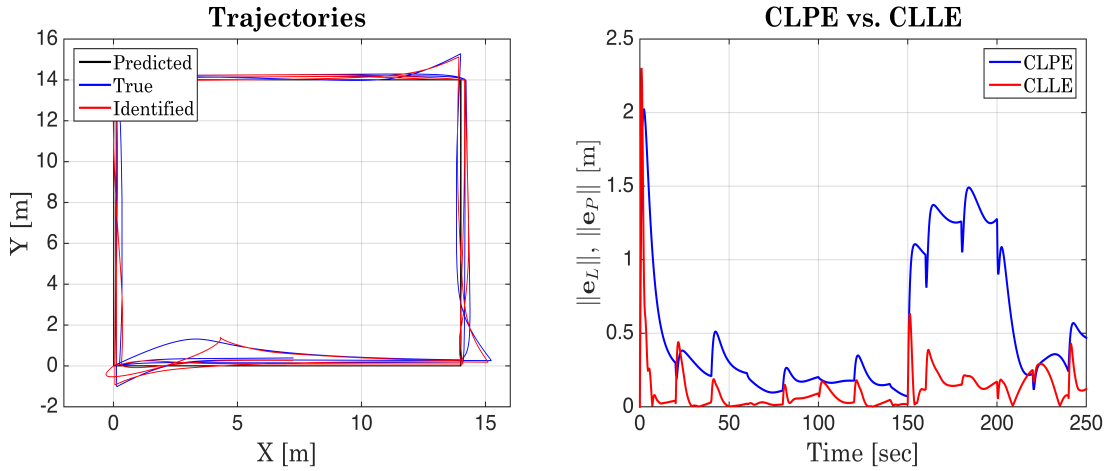
When  $\|\mathbf{e}\| \geq \frac{1}{\beta} \|\Delta\|$  given positive constant  $\beta < \lambda_{\min}(K_D)$ ,  $\dot{V} \leq -(\lambda_{\min}(K_D) - \beta) \|\mathbf{e}\|^2$ , which means  $\dot{V}$  is negative definite. Thus, CLLE is ultimately bounded. The bound of CLLE is  $\|\mathbf{e}\| \leq \frac{1}{\beta} \|\Delta\|$ .  $\square$

### 3.4 Mathematical Simulations of Incorporating Adaptive Learning and Control for Anomaly Detection

This section describes simulation results about incorporating adaptive control and learning for the anomaly detection algorithms in Section 4.2. For the presentation of 2D ocean flow,  $\theta_1 = [0.9 \ 0.5 \ 0.7]$  are selected as the true flow parameter along the horizontal direction;  $\theta_2 = [0.8 \ 0.5 \ 0.9]$  the true flow parameter along the vertical direction.  $\alpha_1 = [1.08 \ 0.6 \ 0.84]$  represents modeled flow along the horizontal direction;  $\alpha_2 = [0.96 \ 0.6 \ 1.08]$  represents modeled flow along the vertical direction. The three combined basis functions are composed of center  $c_i$ , width  $\sigma_i$ , harmonic frequency  $\omega_i$ , and harmonic phase  $v_i$ , where  $i = 1, 2, 3$ .  $c_1$ ,  $c_2$ , and  $c_3$  are  $[0, 0]^\top$ ,  $[10, 10]^\top$ , and  $[5, 5]^\top$ , respectively.  $\sigma_1$ ,  $\sigma_2$ , and  $\sigma_3$  are all equal to 5.  $\omega_1$ ,  $\omega_2$ , and  $\omega_3$  are represented by periods 600 secs, 300 secs, and 800 secs, respectively. Those harmonic periods are arbitrary chosen. Harmonic phases  $v_1$ ,  $v_2$ , and  $v_3$  are zeros. For the uncertainty of actual flow, flow bias 0.23m/s is added to simulated true flow along the X-axis, and flow bias 0.25m/s along the Y-axis.

For the anomaly detection algorithm, positive constant  $K$  in the learning parameter injection term is the identity matrix. Adaptation speed  $\tilde{\gamma}$  and  $\gamma$  is 1.0 and 0.1, respectively. In the prevention scheme of false alarm, false alarm factor  $\gamma_f$  is 0.07. Fig 3.3 represents trajectories of an AUV when the direction of the AUV in the horizontal plane is controlled by heading angle command  $\Psi_c = \frac{\pi}{2} \lfloor \frac{t}{20} \rfloor$  while the adaptive controller cancel out estimated flow. In Figure 3.3, the predicted trajectory represented by the black line has one square in that the vehicle controller completely cancel out modeled flow and make the vehicle follow heading angle command. The simulated true trajectory represented by blue line has similar shape of the square that the predicted trajectory has in that the adaptive controller makes the estimated trajectory follow the predicted trajectory. If the adaptive control scheme could not be applied to the AUV as Chapter 3, the simulated trajectory would have multiple squares different from one square that the predicted trajectory has. The identi-

fied trajectory represented by red line has a square similar to the square of the estimated trajectory. Thus, three trajectories show that the adaptive learning algorithm identifies vehicle motion from the estimated trajectory that represents vehicle motion controlled by the adaptive control algorithm. Figure 3.4 shows CLLE and CLPE. During most time of simulation, CLLE is less than CLPE, which means that identified through-water speed from the learning algorithm is reliable. However, CLLE is greater than equal to CLPE around 2 sec, 20 sec, and 150 sec; we do not trust identified through-water speed at that time.

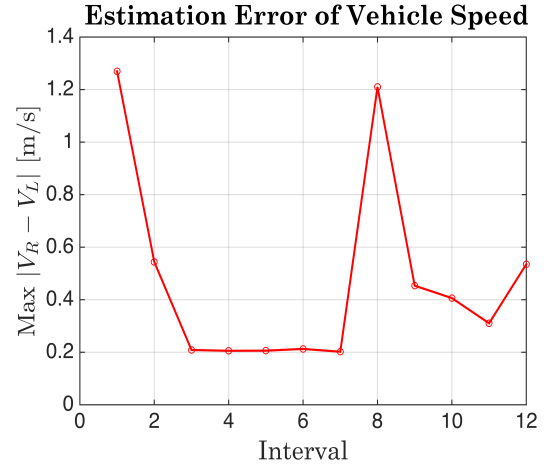
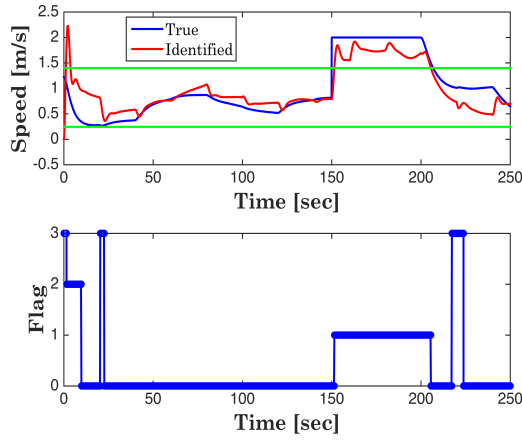


**Figure 3.3:** Predicted (black), estimated (blue), and identified (red) trajectories

**Figure 3.4:** Controlled Lagrangian localization and prediction Error (CLLE, CLPE)

Figures 3.5 and 3.6 shows identified through-water speed, anomaly detection results, and identification error of through-water speed. In the bottom panel of Figure 3.5, a false alarm denoted by flag 3 occurs from the condition that CLLE is greater than and equal to CLPE. Flag 2 that represents a false alarm associated with large identification error of through-water speed happens between 2 sec and 10 sec. Because identified flow is significantly different from modeled flow during that time, we do not trust identified through-water speed. This result is supported by Figure 3.6; identification error of through-water speed is maximum between 0 sec and 20 sec. In addition, we see that identification error of through-water speed is 0.2m/s between 60 sec and 140 sec. Because additional flow bias is added in the entire simulation time, identification error of through-water speed is not

zero, but bounded. At 150 sec, actual through-water speed is increasing abruptly in that abnormal motion happens. Then, identification error is increasing due to uncertainty in a transient period, but error is reduced after the period. These results support our theoretical analysis of Section 4.2.



**Figure 3.5:** Identified through-water speed (top) **Figure 3.6:** Identification error of through-water and flag (bottom) speed

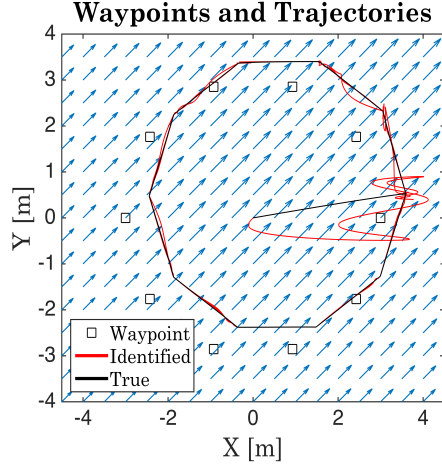
### 3.5 Mathematical Simulations of Adaptive Learning and Control Algorithms

#### 3.5.1 Adaptive learning algorithm

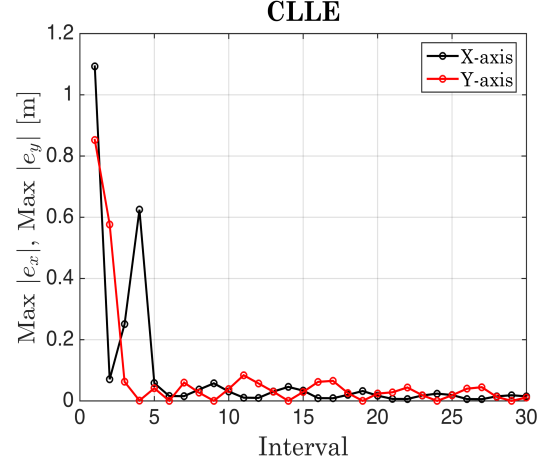
For the adaptive learning algorithm, one gain matrix in the feedback term of the vehicle controller,  $K_D$  is  $\text{diag}\{1, 1\}$ . Identification speed, or  $\gamma$  is designed as 0.8.  $K_L(t)$  and  $\xi(t)$ , which represent parameters in the updating rules of equations (3.70) and (3.71), have all the zero initial values. Figure 3.7 represents waypoints and trajectories of the AUV when we consider 10 waypoints generated from  $\mathbf{z}_c = [r \cos \Theta, r \sin \Theta]^\top$  with  $r = 3$ , and  $\Theta = \frac{\pi}{5} \lfloor \frac{t}{20} \rfloor$ . Arrows represents true ocean flow spatially distributed at the initial time. The direction of arrow changes over time by the tidal basis functions.

An AUV starts to go to the waypoint (3,0) from the origin, and then keeps moving the next waypoint counter-clockwise direction. Waypoints are changed every 20 sec. The AUV completes a cycle when it sequentially travels nine waypoints counter-clock direction from starting at waypoint (3,0), and arrives back at waypoint (3,0).

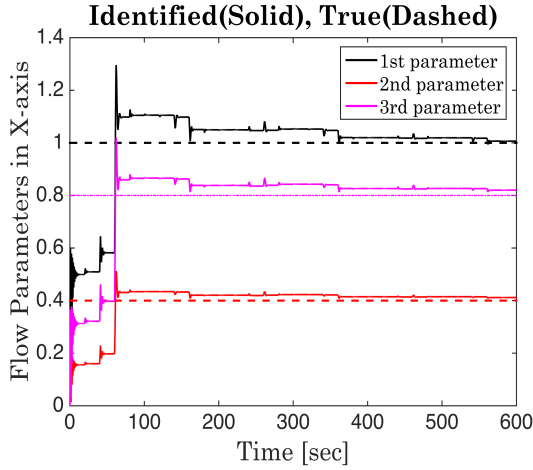
In Figure 3.7, the simulated true trajectory represented by the black line would pass waypoints if the vehicle controller could cancel true flow. However, because the vehicle controller represented by equation (3.62) cancel modeled flow, the difference between modeled and actual flows push the AUV toward north east direction off the waypoints. Figures 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, and , 3.14 show simulation results of CLLE, six flow parameters, and four controller gains. CLLE goes to zero over 10 intervals (200 seconds), which is about one cycle. Moreover, the six flow parameters converge to their true values. Feedback and feedforward gains converge to the gains of the desired controller while showing the similar trend of identified flow parameters. These results support our theoretical analysis in Section 3.3.5.



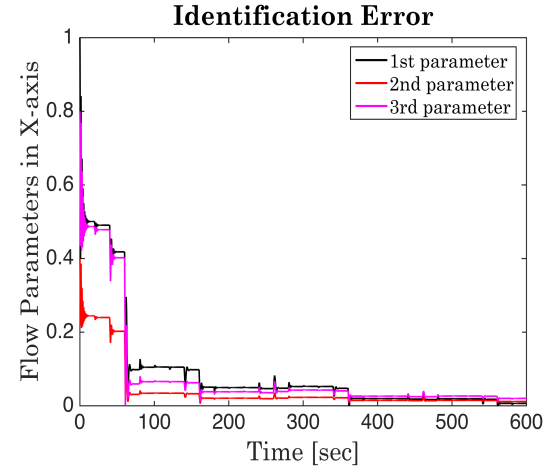
**Figure 3.7:** Identified trajectory and simulated true trajectory



**Figure 3.8:** Convergence of CLLE: CLLE converges to zero over one cycle (1 cycle=10 intervals=200 sec).



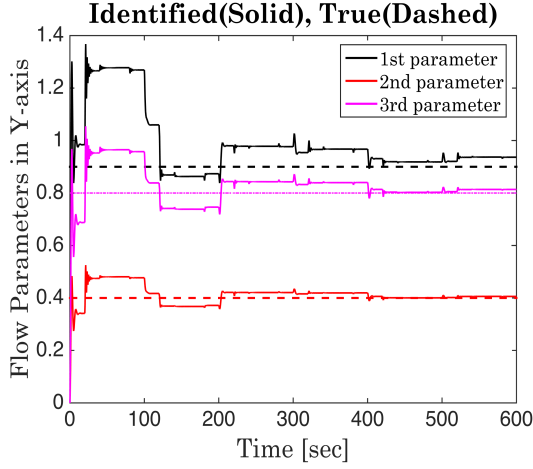
**Figure 3.9:** Convergence of horizontal flow parameters: Parameters for horizontal flow converge to the true values over 200 secs



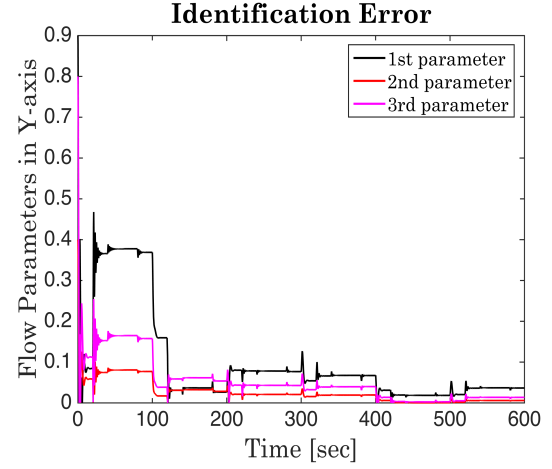
**Figure 3.10:** Identification error of horizontal flow parameters: Identification error converge to zero over 200 secs

### 3.5.2 Adaptive control algorithm

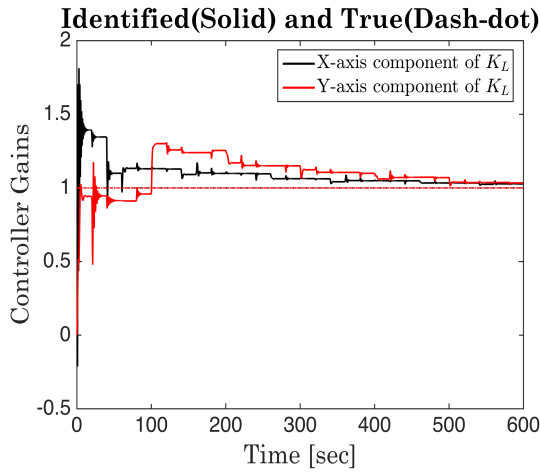
In this section, we perform simulation of the adaptive control algorithm that enables the simulated true trajectory pass the waypoints. The desired controller represented by equation (3.39) generating the predicted trajectory is set by choosing gains  $K_M$  as the two by two identity matrices. Adaptation speed  $\tilde{\gamma}$  is designed as 0.6. In the vehicle controller



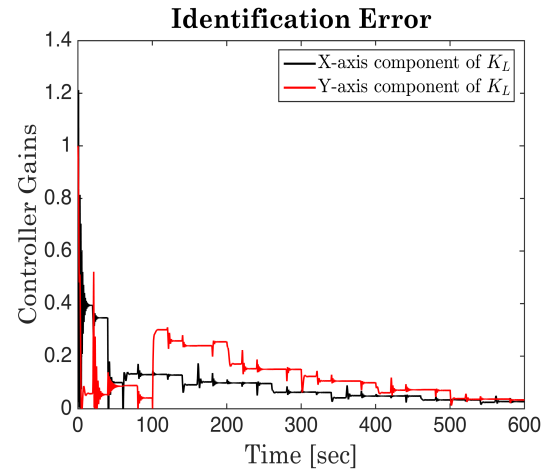
**Figure 3.11:** Convergence of vertical flow parameters: Parameters for vertical flow converge to the true values over 200 sec



**Figure 3.12:** Identification error of vertical flow parameters: Identification error converge to zero over 200 sec



**Figure 3.13:** Convergence of controller gains: Controller gains converge to the true values over 200 sec



**Figure 3.14:** Identification error of controller gains: Identification error converge to zero over 200 sec

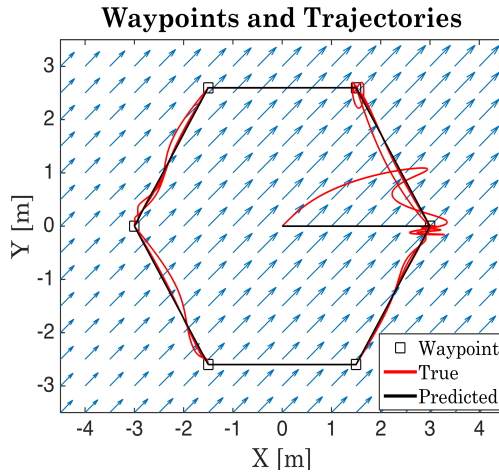
represented by equation (3.38),  $\vartheta_1$ , which represents esimated flow parameters along the horizontal axis, has  $[0.1 \ 0.05 \ 0.1]$  at the initial time.  $\vartheta_2$ , which represents estimated flow parameters along the vertical axis, has  $[0.05 \ 0.1 \ 0.2]$  at the initial time. Gains  $K_R$  in the vehicle controller are initially zeros.  $\vartheta$  and  $K_R$  are constantly updated by the adaptation laws represented by equations (3.45) and (3.46). Figure 3.15 represents waypoints and trajectories of the AUV when we consider 6 waypoints generated from  $\mathbf{z}_c = [r \cos \Theta, r \sin \Theta]^\top$



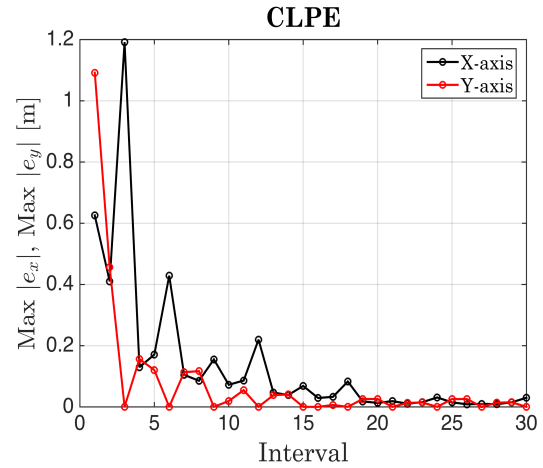
with  $r = 3$ , and  $\Theta = \frac{\pi}{3} \lfloor \frac{t}{20} \rfloor$ .

An AUV starts to go to the waypoint (3,0) from the origin, and then keeps moving the next waypoint counter-clockwise direction. Waypoints are changed every 20 sec. The AUV completes a cycle when it sequentially travels five waypoints counter-clock direction from starting at waypoint (3,0), and arrives back at waypoint (3,0). Arrows represents true ocean flow spatially distributed at the initial time. The direction of arrow changes over time by the tidal basis functions.

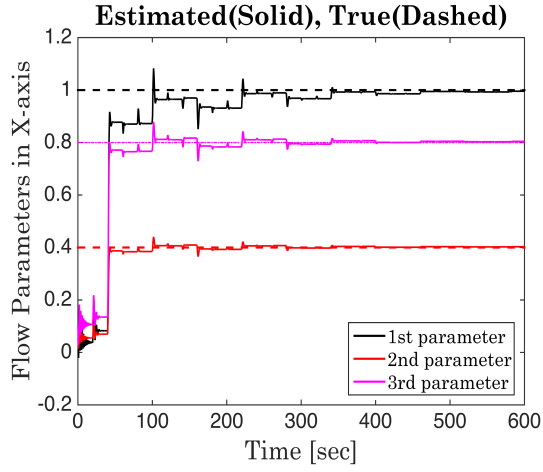
In Figure 3.15, the true trajectory represented by the black line pass waypoints and follow the predicted path as time goes by. Because the vehicle controller represented by equation (3.38) cancel identified flow, the AUV maintain the predicted path when estimated flow parameters are close to true flow parameters. Figures 3.16, 3.17, 3.18, 3.19, 3.20, 3.21, and 3.22 show simulation results of CLPE, six flow parameters, and four controller gains. CLPE goes to zero over 15 intervals (300 seconds), which is about one and half cycle. Moreover, the six flow parameters converge to their true values. Feedback and feedforward gains converge to the gains of the desired controller while showing the similar trend of estimated flow parameters. These results support our theoretical analysis in Section 3.3.1.



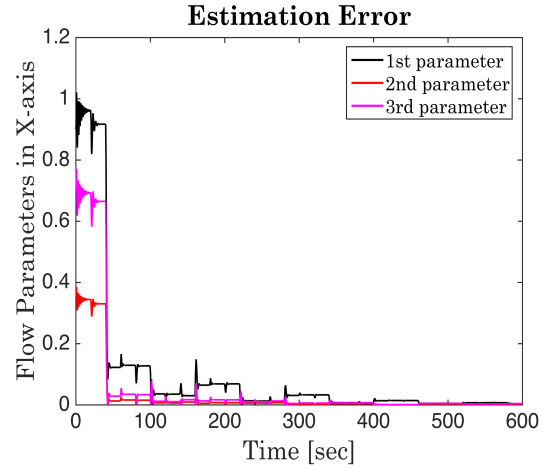
**Figure 3.15:** Simulated true trajectory and predicted trajectory



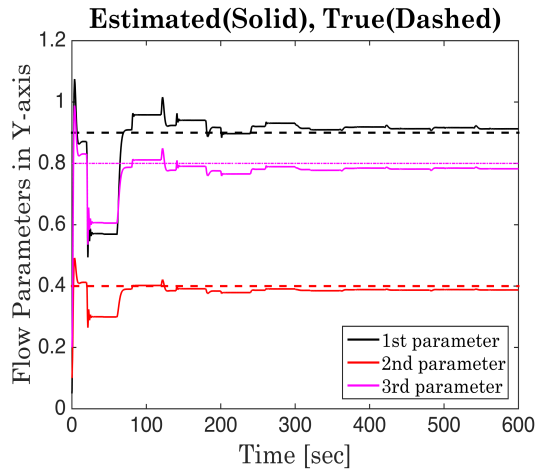
**Figure 3.16:** Convergence of CLPE: CLPE converges to zero over one and half cycle (1.5 cycle=15 intervals=300 sec).



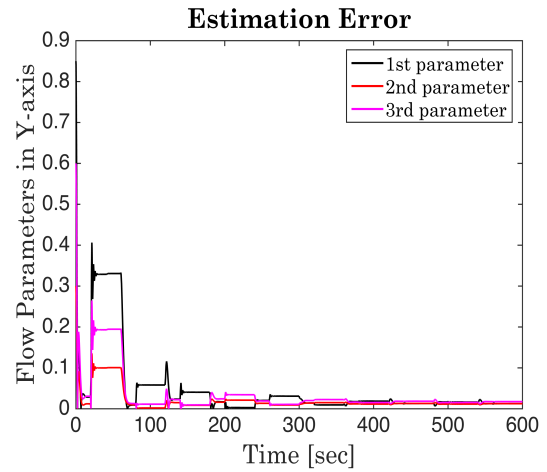
**Figure 3.17:** Convergence of horizontal flow parameters: Parameters for horizontal flow converge to the true values over 300 sec



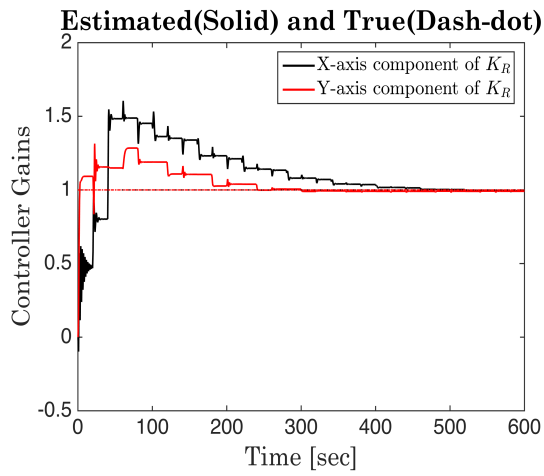
**Figure 3.18:** Estimation of horizontal flow parameters: Estimation error converge to zero over 300 sec



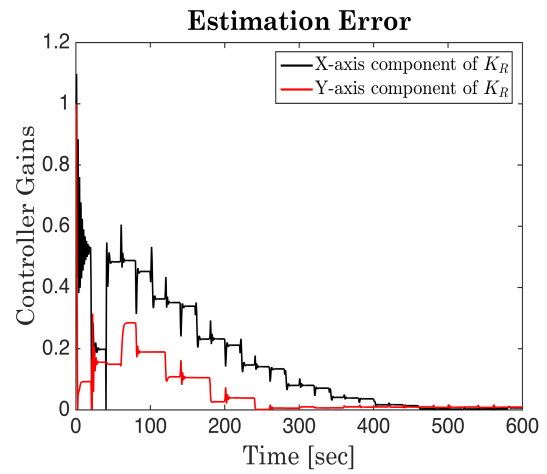
**Figure 3.19:** Convergence of vertical flow parameters: Parameters for vertical flow converge to the true values over 300 sec



**Figure 3.20:** Estimation of vertical flow parameters: Estimation error converge to zero over 300 sec



**Figure 3.21:** Convergence of controller gains:  
Parameters for controller gains converge to the  
true values over 300 sec



**Figure 3.22:** Estimation error of controller gains:  
Estimation error converge to zero over 200 secs

## CHAPTER 4

### ACOUSTIC LOCALIZATION USING A PASSIVE RECEIVER

Estimated trajectory information for the proposed algorithm of anomaly detection for controlled Lagrangian particles can be obtained from a localization algorithm using passive acoustic telemetry. Acoustic telemetry monitors the presence of fish tagged with transmitters (or tags) attached to inside or outside skin of fish (depending on size). Commercially available fish tags can be used as transmitters for localization with relatively low-cost comparing to an acoustic baseline system [53]. The acoustic receiver records the time stamp of detection and identification number of transmitter. Since acoustic telemetry provides detections and non-detections, or binary information of the receivers, this new passive acoustic method can be thought of as a binary acoustic method that consumes less power than the most common acoustic methods that use range measurements of the active acoustic receivers for AUV localization in the literature [29, 30]. For vehicle localization, we use a single Vemco VMT receiver as the passive acoustic receiver, and multiple V13 tags. The Vemco VMT receiver uses an omnidirectional hydrophone that detects multiple transmitters. When the V13 tags send individual ping randomly in a certain time interval at specific frequency, the Vemco VMT receiver records time stamps of pinging and identities of the V13 tags. When a receiver equipped in the vehicle detects tags installed at known positions, the localization algorithm is likely to estimate vehicle positions.

Developing the localization algorithm with the passive receivers is significantly challenging because detection range of the receiver is critical to the accuracy of estimated trajectory information. Although the manufacturer's stated detection range is 1km, actual detection range varies significantly due to wind, tides, buoyancy input, stratification, water depth, and other environmental factors [34]. In this dissertation, we seek to examine how tides, which dominate variability on the Georgia continental shelf [54], may affect acoustic

telemetry in this environment. We first show the localization algorithm using the passive acoustic receiver to find relationship between accuracy of estimated trajectory information and detection range. Then, we analyze variability of hourly detection data collected in an acoustic array.

#### **4.1 Localization Incorporating Flow Model and Acoustic Detection**

Many researchers have proposed various methods to solve the localization problem of AUVs (eg. [29, 30]). Most methods fundamentally focus on decreasing the dead-reckoning or odometry error of AUVs by using external sensors, and assume that the dead-reckoning and odometry models are basic kinematic models of AUVs. However, the basic kinematic models do not account for ocean currents [55], which can greatly affect odometry errors. Some researchers developed algorithms to cope with the effect of ocean currents [56]; their kinematic models assume that ocean flow is unknown constant, which cannot explain substantially more complex ocean flow.

We develop odometry using flow estimation to deal with complex ocean flow. Controlled Lagrangian particle tracking (CLPT) enables analysis of the AUV trajectory in complex flow, combining ocean flow models with the vehicle kinematics. We combine the flow models and their uncertainties in odometry by extending the framework of CLPT, thus allowing analytical study of the growth of odometry error.

To reduce the growth of odometry error based on flow estimates, we assume that while underwater, an AUV is able to detect the presence of acoustic transmitters installed at known locations. However, we do not assume the vehicle is able to measure its distance to the acoustic transmitters. When a signal is detected, the only information the vehicle may have is that its distance to the transmitter is less than a certain threshold. We will show that a reduction of localization error can be achieved under this setting. This method differs from previous work [29, 30] in that the acoustic measurements do not contain direct measurement of range.

We extract geometric information from binary measurements, which uniquely determine sub-regions that correspond to which subset of transmitters can be detected. All the sub-regions can then be represented by a graph. The graph leads to the assignment of likelihood, or the probability of obtaining binary measurements given sub-regions. We describe an acoustic measurement model for binary measurements in the next section.

#### 4.1.1 Acoustic measurement model

We use multiple transmitters installed at known locations and a receiver installed on the AUV. The receiver passively listens to transmitted signals generated from the stationary transmitters. In previous work [57], one AUV that has multiple passive acoustic receivers tracks a shark with a fish tag. Given AUV position, velocity, and attitude angle in a horizontal plane, the AUV estimates the location of the shark and follows the shark by using signal strength measurements and a measurement of relative bearing angle to the tag detected by the passive receivers. Our work differs from [57] because we solve the localization problem of an AUV equipped with one receiver that can detect multiple stationary transmitters installed at known locations. Furthermore, the passive receiver only provides binary information, detection or non-detection.

Given one AUV and  $N$  transmitters, let  $j$  be the index numbers of transmitters, where  $j = 1, \dots, N$ . Let  $\mathbf{x}$  be the true position of the AUV in the horizontal plane relative to an origin. Let  $\mathbf{x}_{\text{trans}}^j \in \mathbb{R}^2$  be the position of the  $j^{\text{th}}$  transmitter. Let  $R_{\text{trans}}^j$  be the range within which the AUV detects the  $j^{\text{th}}$  transmitter. We define  $t_{[k]}$  as follows.

*Definition 5.1:*  $t_{[k]}$  represents the moment when the receiver receives an acoustic signal from one of the transmitters, where  $k = 1, 2, 3, \dots$ . Initial value  $t_{[0]}$  is zero.

We employ the following measurement equation for acoustic detection.

$$z_{jk} = \begin{cases} 1 & \text{if } \|\mathbf{x} - \mathbf{x}_{\text{trans}}^j\| \leq R_{\text{trans}}^j, \\ 0 & \text{if } \|\mathbf{x} - \mathbf{x}_{\text{trans}}^j\| > R_{\text{trans}}^j \end{cases}, \quad (4.1)$$

where  $z_{jk}$  is a binary measurement from  $j^{th}$  transmitter at time  $t_{[k]}$ . Equation (4.1) shows that the AUV detects the signal of the  $j^{th}$  transmitter at time  $t_{[k]}$  within a detection range.

#### 4.1.2 Vehicle motion model

For the proposed odometry model, we add stochastic eddy term  $\mathbf{v}$  to the vehicle motion model described by equation (3.1). The result leads to the Langevin equation [2],

$$d\mathbf{x} = (\mathbf{F}_R(\mathbf{x}, t) + \mathbf{v}_R(\hat{\mathbf{x}}, t) + \mathbf{v})dt \quad (4.2)$$

$$d\mathbf{v} = -M\mathbf{v}dt + \Lambda d\boldsymbol{\omega}, \quad (4.3)$$

where  $\hat{\mathbf{x}}$  is the estimated position of the AUV in the horizontal plane relative to an origin, and  $\boldsymbol{\omega}$  is a stochastic input that is Gaussian white noise with a zero mean. Matrix  $\mathbf{A}$  is  $\begin{bmatrix} \frac{1}{\tau} & \Omega \\ -\Omega & \frac{1}{\tau} \end{bmatrix}$ , and matrix  $\Lambda$  is  $\sigma\sqrt{\frac{2}{\tau}}I_{2 \times 2}$ , where  $\sigma$  represents the variance of stochastic fluctuations,  $\tau$  is the Lagrangian correlation time, and  $\Omega$  is a spin parameter. Here, we assume that  $\sigma$  and  $\tau$  are known constants and  $\Omega = 0$ . Our goal is to localize the AUV moving under the control of the state feedback controller with flow canceling. Let  $\mathbf{x}_k$  be  $\mathbf{x}$  at time  $t_{[k]}$ . Let  $\mathbf{z}_k$  be a measurement vector, which consists of all binary measurements from all transmitters at time  $t_{[k]}$ . That is,  $\mathbf{z}_k = [z_{1k}, \dots, z_{Nk}]^T$ . Let  $\mathbf{Z}^k$  be all binary measurement vectors until time  $t_{[k]}$ . Then  $\mathbf{Z}^k = [\mathbf{z}_1, \dots, \mathbf{z}_k]$ . We estimate the position of the AUV with probabilistic approach as follows:

$$\hat{\mathbf{x}}_k = \arg \max_{\mathbf{x}_k} p(\mathbf{x}_k | \mathbf{Z}^k). \quad (4.4)$$

To compute probability of equation (4.4), we first derive odometry using flow estimation for state propagation in the next section.

### 4.1.3 Odometry using flow estimation

To deal with nonlinear time-varying stochastic differential equations (4.2) and (4.3), we incorporate a flow model and the vehicle motion model as follows:

$$d\mathbf{x} = (\mathbf{F}_M(\hat{\mathbf{x}}, t) + \mathbf{v}_R(\hat{\mathbf{x}}, t) + \mathbf{F}_R(\mathbf{x}, t) - \mathbf{F}_M(\hat{\mathbf{x}}, t) + \mathbf{v})dt \quad (4.5)$$

$$d\mathbf{v} = -A\mathbf{v}dt + \Lambda d\boldsymbol{\omega}. \quad (4.6)$$

When we take the expectation of both sides in (4.6) and assume  $\mathbf{v}(t_0) = 0$  where  $t_0$  is the initial time,  $E(\mathbf{v}) = 0$ . Because all terms of the right side in (4.5) are deterministic except for  $\mathbf{v}$ , it follows that

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{F}_M(\hat{\mathbf{x}}, t) + \mathbf{v}_R(\hat{\mathbf{x}}, t) + \mathbf{F}_R(\hat{\mathbf{x}}, t) - \mathbf{F}_M(\hat{\mathbf{x}}, t). \quad (4.7)$$

A waypoint controller is designed for controlled velocity  $\mathbf{v}_R(\hat{\mathbf{x}}, t)$  [9, 31]. The waypoint controller guides the vehicle to a waypoint while canceling estimated flow velocity from available flow models. The cancellation of the normal component of estimated flow velocity with respect to a line between the vehicle and the waypoint enables the vehicle to reach the waypoint. Figure 4.1 show a schematic diagram of the waypoint controller. Controlled velocity  $\mathbf{v}_R(\hat{\mathbf{x}}, t)$  is the following equation:

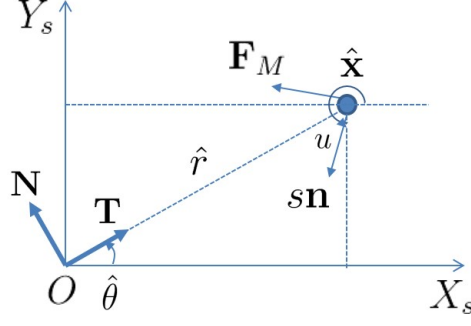
$$\mathbf{v}(\hat{\mathbf{x}}, t) = (\mathbf{sn}(u(\hat{\mathbf{x}}, t)) \cdot \mathbf{N})\mathbf{N} + (\mathbf{sn}(u(\hat{\mathbf{x}}, t)) \cdot \mathbf{T})\mathbf{T} \quad (4.8)$$

$$\mathbf{sn}(u(\hat{\mathbf{x}}, t)) \cdot \mathbf{N} = -\mathbf{F}_M(\hat{\mathbf{x}}, t) \cdot \mathbf{N} \quad (4.9)$$

$$\mathbf{sn}(u(\hat{\mathbf{x}}, t)) \cdot \mathbf{T} = \pm \sqrt{|s^2 - (\mathbf{F}_M(\hat{\mathbf{x}}, t) \cdot \mathbf{N})^2|}. \quad (4.10)$$

Let  $\hat{x}_1$  and  $\hat{x}_2$  be components of the positional estimate  $\hat{\mathbf{x}}$ . To conveniently handle the nonlinear system represented by equation (4.7), we transform a Cartesian coordinate system into a polar coordinate system by the following definition.





**Figure 4.1:** The schematic of the waypoint controller:  $s\mathbf{n}$  represents the velocity vector of the vehicle with constant speed  $s$ .  $u$  is a desired heading angle called a control input.  $\mathbf{T}$  is the unit vector of  $\overrightarrow{O\hat{\mathbf{x}}}$ ,  $\mathbf{N}$  is the rotation vector of  $\mathbf{T}$  by 90 degrees counterclockwise, and  $X_s$ - $Y_s$  is the  $X$ - $Y$  axis in an inertial frame.

*Definition 5.1:* Let  $\hat{r}$  and  $\hat{\theta}$  be an estimated range and an estimated angle, respectively in the polar coordinate system. The components  $x_1$  and  $x_2$  of  $\hat{\mathbf{x}}$  are  $\hat{r}\cos\hat{\theta}$  and  $\hat{r}\sin\hat{\theta}$ , respectively. Since the waypoint controller determines the controlled velocity of the AUV, the deterministic error growth of equation (4.7) depends on the discrepancy between real and modeled flows; we call  $\mathbf{F}_R(\hat{\mathbf{x}}, t) - \mathbf{F}_M(\hat{\mathbf{x}}, t)$  a perturbation term. When the perturbation term is zero,

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{F}_M(\hat{\mathbf{x}}, t) + \mathbf{v}_R(\hat{\mathbf{x}}, t). \quad (4.11)$$

*Definition 5.2:* Let  $\hat{r}_0$  and  $\hat{\theta}_0$  be solutions of equation (4.11). They are called nominal solutions.

Let  $f_{M_1}(\hat{\mathbf{x}}, t)$  be the projected component of  $\mathbf{F}_M$  on  $\mathbf{T}$ . Let  $f_{M_2}(\hat{\mathbf{x}}, t)$  be the projected component of  $\mathbf{F}_M$  on  $\mathbf{N}$ . Then,  $\mathbf{F}_M = f_{M_1}(\hat{\mathbf{x}}, t)\mathbf{T} + f_{M_2}(\hat{\mathbf{x}}, t)\mathbf{N}$ . Let  $f_{R_1}(\hat{\mathbf{x}}, t)$  be the projected component of  $\mathbf{F}_R$  on  $\mathbf{T}$  and  $f_{R_2}(\hat{\mathbf{x}}, t)$  be the projected component of  $\mathbf{F}_R$  on  $\mathbf{N}$ . Then,  $\mathbf{F}_R = f_{R_1}(\hat{\mathbf{x}}, t)\mathbf{T} + f_{R_2}(\hat{\mathbf{x}}, t)\mathbf{N}$ .

*Assumption 5.1:* Let  $\varepsilon_{r_0}$  and  $\varepsilon_{\theta_0}$  be in  $\mathbb{R}$ . Let  $g_1(\hat{\mathbf{x}}, t)$  and  $g_2(\hat{\mathbf{x}}, t)$  be  $f_{R_1}(\hat{\mathbf{x}}, t) - f_{M_1}(\hat{\mathbf{x}}, t)$  and  $f_{R_2}(\hat{\mathbf{x}}, t) - f_{M_2}(\hat{\mathbf{x}}, t)$ , respectively. Then we suppose that the components of the perturbation term are bounded,  $\|g_1(\hat{r}\cos\hat{\theta}, \hat{r}\sin\hat{\theta}, t)\| \leq \varepsilon_r$ ,  $\|g_2(\hat{r}\cos\hat{\theta}, \hat{r}\sin\hat{\theta}, t)\| \leq \varepsilon_\theta$ , where  $\varepsilon_r \in [0, \varepsilon_{r_0}]$  and  $\varepsilon_\theta \in [0, \varepsilon_{\theta_0}]$ .

*Assumption 5.2:* Let  $\varepsilon_r(t_{[k-1]})$  be the value of  $\varepsilon_r$  at time  $t_{[k-1]}$ . Let  $\varepsilon_\theta(t_{[k-1]})$  be the value of

$\varepsilon_\theta$  at time  $t_{[k-1]}$ . We assume that both  $\varepsilon_r(t_{[k-1]})$  and  $\varepsilon_\theta(t_{[k-1]})$  are time-invariant constants; but,  $\varepsilon_r(t_{[k]})$  and  $\varepsilon_\theta(t_{[k]})$  have different constant values from  $\varepsilon_r(t_{[k-1]})$  and  $\varepsilon_\theta(t_{[k-1]})$ .

*Remark 5.1:*  $\mathbf{F}_R \neq \mathbf{F}_M$  because real flows differ from modeled flows. This difference means that the solution of equation (4.7) is not the nominal solution due to the perturbation term. To analytically derive the solution of equation (4.7), we let  $\varepsilon_r$  and  $\varepsilon_\theta$  be upper bounds of perturbation terms, representing the worst case scenario about the difference between real and modeled flows. Given equations (4.5), (4.6), (4.7), and (4.8), the position estimate of an AUV is as follows:

$$\hat{\mathbf{x}}(t_{[k]}) = \begin{pmatrix} \hat{r}(t_{[k]}) \cos(\hat{\theta}(t_{[k]})) \\ \hat{r}(t_{[k]}) \sin(\hat{\theta}(t_{[k]})) \end{pmatrix} + \mathbf{w}(t_{[k]}), \quad (4.12)$$

where

$$\begin{aligned} \hat{r}(t_{[k]}) &= \hat{r}_0(t_{[k]}) + \varepsilon_r(t_{[k-1]}) \left( e^{\int_{t_{[k-1]}}^{t_{[k]}} A(t) dt} \int_{t_{[k-1]}}^{t_{[k]}} e^{-\int_{t_{[k-1]}}^{t_{[k]}} A(t) dt} dt \right) \\ &\quad + \varepsilon_\theta(t_{[k-1]}) \left( e^{\int_{t_{[k-1]}}^{t_{[k]}} A(t) dt} \int_{t_{[k-1]}}^{t_{[k]}} B'(t) e^{-\int_{t_{[k-1]}}^{t_{[k]}} A(t) dt} dt \right), \\ \hat{\theta}(t_{[k]}) &= \hat{\theta}_0(t_{[k]}) + \varepsilon_\theta(t_{[k-1]}) \int_{t_{[k-1]}}^{t_{[k]}} \frac{1}{\hat{r}_0(t_{[k]})} dt, \\ \mathbf{w}(t_{[k]}) &\sim \mathcal{N} \left( 0, \frac{L^2}{2a} (1 - e^{-2at_{[k]}}) I_{2 \times 2} \right). \end{aligned}$$

Here  $A(t) = \frac{\partial F}{\partial \hat{r}}(\hat{r}_0, \hat{\theta}_0, t)$ ,  $B(t) = \frac{\partial F}{\partial \hat{\theta}}(\hat{r}_0, \hat{\theta}_0, t)$ , and  $B'(t) = B(t) \int_{t_0}^t -\frac{1}{\hat{r}_0} dt$ . Using equations (4.5), (4.6), (4.7), and (4.8) and Assumptions 5.1, and 5.2, we have

$$\dot{\hat{r}} = f_{M1}(\hat{r} \cos \hat{\theta}, \hat{r} \sin \hat{\theta}, t) - \sqrt{s^2 - (f_{M2}(\hat{r} \cos \hat{\theta}, \hat{r} \sin \hat{\theta}, t))^2} + \varepsilon_r \quad (4.13)$$

$$\dot{\hat{\theta}} = -\frac{1}{\hat{r}} \varepsilon_\theta. \quad (4.14)$$

Range  $\hat{r}$  and angle  $\hat{\theta}$  consist of nominal solutions and perturbation-related terms as follows:

$$\hat{r} = \hat{r}_0 + \varepsilon_r \hat{r}_1 + \varepsilon_\theta \hat{r}'_1 \quad (4.15)$$

$$\hat{\theta} = \hat{\theta}_0 + \varepsilon_\theta \hat{\theta}_1. \quad (4.16)$$

For the first step, we assume that  $\varepsilon_r = 0$ . Then equations (4.15) and (4.16) are changed as follows:

$$\hat{r} = \hat{r}_0 + \varepsilon_\theta \hat{r}'_1 \quad (4.17)$$

$$\hat{\theta} = \hat{\theta}_0 + \varepsilon_\theta \hat{\theta}_1. \quad (4.18)$$

When we plug equations (4.17) and (4.18) in equations (4.13) and (4.14),

$$\dot{\hat{r}}_0 + \varepsilon_\theta \dot{\hat{r}}'_1 = F(\hat{r}_0, \hat{\theta}_0, t, 0) + \left. \frac{\partial F(\hat{r}, \hat{\theta}, t, 0)}{\partial \varepsilon_\theta} \right|_{\varepsilon_\theta=0} \varepsilon_\theta \quad (4.19)$$

$$\dot{\hat{\theta}}_0 + \varepsilon_\theta \dot{\hat{\theta}}_1 = G(\hat{r}_0, \hat{\theta}_0, t, 0) + \left. \frac{\partial G(\hat{r}, \hat{\theta}, t, \varepsilon_\theta)}{\partial \varepsilon_\theta} \right|_{\varepsilon_\theta=0} \varepsilon_\theta. \quad (4.20)$$

Let  $A(t)$ ,  $B(t)$ , and  $B'(t)$  be  $\frac{\partial F}{\partial \hat{r}}(\hat{r}_0, \hat{\theta}_0, t, 0)$ ,  $\frac{\partial F}{\partial \hat{\theta}}(\hat{r}_0, \hat{\theta}_0, t, 0)$ , and  $B(t) \int_{t_0}^t -\frac{1}{\hat{r}_0} dt$ , respectively.

From Definition 5.2, we obtain linear time varying first-order differential equations as follows:

$$\dot{\hat{r}}'_1 = A(t) \hat{r}_1 + B(t) \hat{\theta}_1 \quad (4.21)$$

$$\dot{\hat{\theta}}_1 = -\frac{1}{\hat{r}_0}. \quad (4.22)$$

Thus, the solutions of the equations are

$$\hat{r}'_1 = e^{\int_{t_0}^t A(t) dt} \left( \int_{t_0}^t B'(t) e^{-\int_{t_0}^t A(t) dt} dt \right) \quad (4.23)$$

$$\hat{\theta}_1 = \int_{t_0}^t -\frac{1}{\hat{r}_0} dt, \quad (4.24)$$

because  $\hat{r}(t_0) = \hat{r}_0(t_0) + \varepsilon_\theta \hat{r}'_1(t_0)$ . For the second step, we assume that  $\varepsilon_\theta = 0$ . Equations (4.15) and (4.16) lead to

$$\hat{r} = \hat{r}_0 + \varepsilon_r \hat{r}_1 \quad (4.25)$$

$$\hat{\theta} = \theta_0. \quad (4.26)$$

When we use similar way to the first step, we obtain

$$\hat{r}_1 = e^{\int_{t_0}^t A(t)dt} \left( \int_{t_0}^t e^{-\int_{t_0}^t A(t)dt} dt \right). \quad (4.27)$$

For the final step, when we plug equations (4.15) and (4.16) in equations (4.13) and (4.14), we obtain the following equation related to  $\varepsilon_r$ .

$$\hat{r}'_1 = \left. \frac{\partial F(\hat{r}, \hat{\theta}, t, \varepsilon_r)}{\partial \hat{r}} \right|_{\varepsilon_r=0} \hat{r}_1 + \left. \frac{\partial F(\hat{r}, \hat{\theta}, t, \varepsilon_r)}{\partial \varepsilon_r} \right|_{\varepsilon_r=0}. \quad (4.28)$$

In addition, the remaining terms related to  $\varepsilon_\theta$  are

$$\hat{r}'_1 = \left. \frac{\partial F(\hat{r}, \hat{\theta}, t, \varepsilon_r)}{\partial \hat{r}} \right|_{\varepsilon_r=0} \hat{r}'_1 + \left. \frac{\partial F(\hat{r}, \hat{\theta}, t, \varepsilon_r)}{\partial \hat{\theta}} \right|_{\varepsilon_r=0} \hat{\theta}_1 \quad (4.29)$$

$$\hat{\theta}_1 = \left. \frac{\partial G(\hat{r}, \hat{\theta}, t, \varepsilon_\theta)}{\partial \hat{r}} \right|_{\varepsilon_\theta=0} \hat{r}_1 + \left. \frac{\partial G(\hat{r}, \hat{\theta}, t, \varepsilon_\theta)}{\partial \hat{\theta}} \right|_{\varepsilon_\theta=0} \hat{\theta}_1 + \left. \frac{\partial G(\hat{r}, \hat{\theta}, t, \varepsilon_\theta)}{\partial \varepsilon_\theta} \right|_{\varepsilon_\theta=0}. \quad (4.30)$$

All solutions of equations (4.28), (4.29), and (4.30) correspond to the solutions of the first and the second step. The results are as follows:

$$\hat{r} - \hat{r}_0 = \varepsilon_r \left( e^{\int_{t_0}^t A(t)dt} \int_{t_0}^t e^{-\int_{t_0}^t A(t)dt} dt \right) + \varepsilon_\theta \left( e^{\int_{t_0}^t A(t)dt} \int_{t_0}^t B'(t) e^{-\int_{t_0}^t A(t)dt} dt \right) \quad (4.31)$$

$$\hat{\theta} - \hat{\theta}_0 = \varepsilon_\theta \int_{t_0}^t \frac{1}{\hat{r}_0} dt. \quad (4.32)$$

Because we need error growth of odometry between  $t_{[k]}$  and  $t_{[k-1]}$ , deterministic error growth of odometry is

$$\begin{aligned}\hat{\mathbf{r}}(t_{[k]}) &= \hat{\mathbf{r}}_0(t_{[k]}) + \boldsymbol{\varepsilon}_r(t_{[k-1]}) \left( e^{\int_{t_{[k-1]}}^{t_{[k]}} A(t) dt} \int_{t_{[k-1]}}^{t_{[k]}} e^{-\int_{t_{[k-1]}}^t A(t) dt} dt \right) \\ &+ \boldsymbol{\varepsilon}_\theta(t_{[k-1]}) \left( e^{\int_{t_{[k-1]}}^{t_{[k]}} A(t) dt} \int_{t_{[k-1]}}^{t_{[k]}} B'(t) e^{-\int_{t_{[k-1]}}^t A(t) dt} dt \right), \\ \hat{\boldsymbol{\theta}}(t_{[k]}) &= \hat{\boldsymbol{\theta}}_0(t_{[k]}) + \boldsymbol{\varepsilon}_\theta(t_{[k-1]}) \int_{t_{[k-1]}}^{t_{[k]}} \frac{1}{\hat{\mathbf{r}}_0(t_{[k]})} dt, \\ \mathbf{w}(t_{[k]}) &\sim \mathcal{N} \left( 0, \frac{L^2}{2a} (1 - e^{-2at_{[k]}}) I_{2 \times 2} \right).\end{aligned}$$

For stochastic error growth of odometry stochastic eddy diffusion term described by equation (4.6), let  $M = (1/\tau) I_{2 \times 2}$  and  $\Lambda = (\sigma \sqrt{2/\tau}) I_{2 \times 2}$ . Let  $\mathbf{v}'$  be a component in  $\mathbf{v}$ , where  $\mathbf{v} = \mathbf{v}_1 = \mathbf{v}_2$ . Let  $\boldsymbol{\omega}'$  be a component in  $\boldsymbol{\omega}$ , where  $\boldsymbol{\omega} = \boldsymbol{\omega}_1 = \boldsymbol{\omega}_2$ . To solve equation (4.6), let

$$d\mathbf{v}' = -a\mathbf{v}'dt + Ld\boldsymbol{\omega}', \quad (4.33)$$

where  $a = \frac{1}{\tau}$ , and  $L = \sigma \sqrt{\frac{2}{\tau}}$ . Let  $\mathbf{v}(t_0) = 0$ . Because  $\boldsymbol{\omega}'$  is Gaussian white noise with a zero mean and  $\mathbf{v}(t_0) = 0$ , it is clear that  $E(\mathbf{v}') = 0$  when we take the expectation of both sides in equation (4.33). Let  $\phi(\xi) = \mathbf{v}'^2$  where  $\xi = \mathbf{v}'$  and  $g = L$ . By using the Ito's differentiation rule, we obtain

$$\text{VAR}(\mathbf{v}') = \frac{L^2}{2a} (1 - e^{-2at}). \quad (4.34)$$

The stochastic error growth is calculated by

$$E(\mathbf{v}') = 0, \text{VAR}(\mathbf{v}') = \frac{L^2}{2a} (1 - e^{-2at}). \quad (4.35)$$

The stochastic error growth in equation (4.35) is not distributed by Gaussian. Thus, we model the error growth at time  $t_{[k]}$  as Gaussian with zero mean and variance of equation

(4.35). Note that the variance of equation (4.35) is changed every time  $t_{[k]}$ . Therefore,

$$\mathbf{w}(t_{[k]}) \sim \mathcal{N}\left(0, \frac{L^2}{2a} \left(1 - e^{-2at_{[k]}}\right) I_{2 \times 2}\right). \quad (4.36)$$

*Remark 5.2:*  $\hat{\mathbf{r}}(t_{[k]})$  depends on nominal solution  $\hat{\mathbf{r}}_0(t_{[k]})$  and bounded constant  $\varepsilon_r(t_{[k-1]})$ .  $\hat{\boldsymbol{\theta}}(t_{[k]})$  depends on nominal solution  $\hat{\boldsymbol{\theta}}_0(t_{[k]})$  and bounded constant  $\varepsilon_\theta(t_{[k-1]})$ . Nominal solutions at time  $t_{[k]}$  are computed from time  $t_{[k-1]}$ . Thus, equation (4.12) will be used for propagating particles in the proposed particle filter algorithm of Section 4.1.5.

#### 4.1.4 Division of sensing regions

When an object that has a transmitter moves around the sensor, the sensor detects a signal transmitted from the transmitter. The detection informs us that distance to the transmitter is less than or equal to a certain threshold. A transmitted signal is not detected when the distance is greater than the threshold. The region where the sensor can detect the transmitter is called a sensing region; here, we use sensing regions defined in [58, 59]. We divide sensing regions into sub-regions under the following assumptions:

*Assumption 5.2* Given  $N$  transmitters, the transmitters are fixed at known locations in the 2D space.

*Assumption 5.3* The receiver is installed in the AUV.

We draw circles with radii equal to the ranges of detection centered around each transmitter, and assume that the receiver can detect the transmitters inside these circles, each of which represents a disk-shaped sensing region. Geometric sub-regions are defined as the intersections of sensing regions. Let labels of sub-regions be  $C^0, \dots, C^S$ . We assume that  $S$  is  $N^2 - N + 1$ , which is the maximum number of possible sub-regions given  $N$  transmitters [60]. Let  $m \leq N$  be the number of transmitters that the AUV detects. Let  $i_l$  be an index number of the  $l^{th}$  transmitter, where  $l = 1, \dots, m$ . By Assumption 5.2, we know the location of the  $l^{th}$  transmitter when the AUV detects the  $l^{th}$  transmitter. The region where the

location is in is labeled as follows (see Figure 4.2):

If  $m = 0$ , then the vehicle is in region  $C^0$

If  $m = 1$ , then the vehicle is in one of regions  $C^1, \dots, C^N$

If  $m \neq 1$  and  $m \neq N$ , then the vehicle is in one of regions

$$C^{mN-N+1}, \dots, C^{mN}$$

If  $m = N$ , then the vehicle is in region  $C^{N^2-N+1}$ .

There are many ways to assign labels to regions. Fortunately such assignments can be determined once and for all after the transmitter locations are known. Figure 4.2 shows an example with three transmitters ( $N = 3$ ). When the AUV detects transmitters 1 and 2, the AUV is in region  $C^4$ . Similarly, the AUV is in region  $C^7$  when it detects transmitters 1, 2, and 3, and we match all labels of sub-regions with the combination of index numbers that show detected transmitters. This match is one-to-one correspondence, since the number of sub-regions equals the number of index numbers. In addition, all labels of sub-regions are distinct, and the combination of index numbers represented by detected transmitters are distinct.

We convert sensing regions into a graph. The benefit of the graph is that we can explicitly use adjacency information on the sub-regions at the graph. We define the graph as follows:

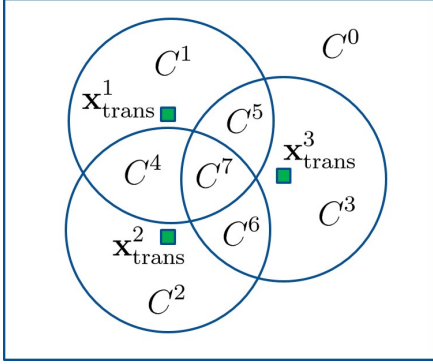
*Definition 5.3* A graph of sub-regions  $C^1, C^2, \dots$  is graph  $G$ , which is defined to be a set  $\{C^\rho\}$  of nodes and a set  $\{e_{\rho\gamma}\}$  of arcs that connect nodes, where  $\rho, \gamma = 0, 1, \dots \leq N^2 - N + 1$  and  $\rho \neq \gamma$ .

*Definition 5.4*  $C^\nu$  is a neighbor node of  $C^\mu$  if there exists an arc  $e_{\mu\nu}$  between  $C^\mu$  and  $C^\nu$ .

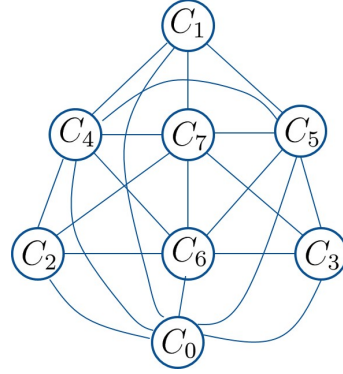
*Definition 5.5* Neighboring operator  $\Gamma$  is defined such that  $\Gamma : \{C^\rho\} \rightarrow \{C^\gamma\}$ .

As an example of the transformation of sub-regions into a graph, we consider three transmitters in Figure 4.2, which shows eight sub-regions in sensing regions.  $\mathbf{x}_{\text{trans}}^1$ ,  $\mathbf{x}_{\text{trans}}^2$ , and

$\mathbf{x}_{\text{trans}}^3$  represent the locations of transmitters. Figure 4.3 illustrates the graph derived from Figure 4.2. The graph has eight nodes and their undirected arcs. Using this graph structure,



**Figure 4.2:** Three transmitters and regions of detection



**Figure 4.3:** The graph transformed from Figure 4.2

we model a likelihood function as follows.

$$q(\mathbf{x}_{k(-)} \in C^p | z_k), \quad (4.37)$$

where  $\mathbf{x}_{k(-)} \in C^p$  is vehicle position in sub-region  $C^p$  before updating detection measurements. The likelihood represents the probability of obtaining detection measurements given sub-regions. We design the likelihood for developing a particle filter algorithm in the next section.

#### 4.1.5 Graph-based particle filter

The proposed odometry model in equation (4.12) has nonlinear dynamics with non-Gaussian noise. The acoustic detection model is a nonlinear mapping function in equation (4.1). The following section develops the graph-based particle filter algorithm, which incorporates odometry and acoustic measurements for the positional estimation of the AUV.

We derive posterior probability density function  $p(\mathbf{x}_k \in C^p | \mathbf{Z}^k)$  under the Bayesian



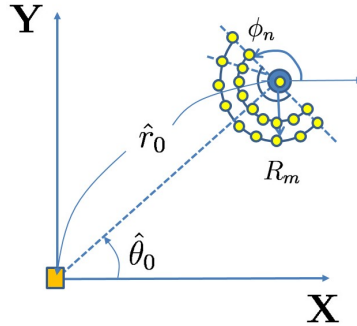
approach for the estimation of the position of the AUV in a sub-region as follows:

$$p(\mathbf{x}_k \in C^p | \mathbf{Z}^k) = \frac{q(\mathbf{x}_{k(-)} \in C^p | z_k) p(\mathbf{x}_k \in C^p | \mathbf{Z}^{k-1})}{\int q(\mathbf{x}_{k(-)} \in C^p | z_k) p(\mathbf{x}_k \in C^p | \mathbf{Z}^{k-1})}, \quad (4.38)$$

where  $p(\mathbf{x}_k \in C^p | \mathbf{Z}^{k-1})$  is a prior probability density. When approximating probability density function with the classical histogram approach, we compute a prior probability density by using odometry represented by equation (4.12), and model likelihood on the graph to determine the weights of particles. Then, we estimate the position of the AUV with the highest weight of the particles using the maximum a posteriori (MAP) estimator. We introduce the particle filter algorithm to estimate the position of the AUV in the following steps.

#### 1. Initialization of particles

The true positions of the AUV may be distributed with respect to the straight line, the nominal trajectory of the AUV under the waypoint controller between the starting and goal points. Initial particles are placed on the area of a half circle centered about the straight line. The radius of the half circle represents the uncertainty of the initial position of the AUV. We uniformly discretize the area with intersected points between the arcs of half circles and radial lines at the starting point as shown in Figure 5.3.



**Figure 4.4:** Generation of initial particles;  $\hat{r}_0$  and  $\hat{\theta}_0$  represent the initial position of the AUV.  $\phi_n$  and  $R_m$  are the orientation and the size of the half circle, respectively.

## 2. Propagation of particles

Let  $\mathbf{x}_k^{(l),(-)}$  be the  $l^{th}$  particle before updating measurements. To obtain a priori probability density function  $p(\mathbf{x}_k \in C^\rho | \mathbf{Z}^{k-1})$ , we generate propagated particle  $\mathbf{x}_k^{(l),(-)}$  using equation (4.12).

## 3. Computation of likelihood

We compute likelihood  $q(\mathbf{x}_k^{(l),(-)} | z_k)$  from new acoustic measurements. The proposed likelihood model uses three types of information on the graph: the previous estimated position, the predicted positions, and the measurements. The previous estimated position is an estimate of the AUV at time  $t_{[k-1]}$ . Because the estimated position belongs to a node on the graph, we obtain the index number of the node. The predicted positions correspond to the particles in Step 2. We calculate the index numbers of nodes that contain the particles. The likelihood function on the graph is modeled as follows:

$$q\left(\mathbf{x}_k^{(l),(-)} | z_k\right) = \begin{cases} q_1 & \text{if } \hat{\mathbf{x}}_{k-1}, \mathbf{x}_k^{(l),(-)} \in C^\rho \text{ and } z_k = \rho \\ q_2 & \text{if } \mathbf{x}_k^{(l),(-)} \in C^\nu \text{ and } z_k = \nu \\ & \text{and } \mathbf{x}_k^{(l),(-)} \in \Gamma(\hat{\mathbf{x}}_{k-1} \in C^\rho), \nu \neq \rho \\ q_3 & \text{if } \mathbf{x}_k^{(l),(-)} \in C^\nu \text{ and } z_k \neq \nu \\ & \text{and } \mathbf{x}_k^{(l),(-)} \in \Gamma(\hat{\mathbf{x}}_{k-1} \in C^\rho), \nu \neq \rho \\ q_4 & \text{if } \mathbf{x}_k^{(l),(-)} \in C^\nu \text{ and } z_k = \nu \\ & \text{and } \mathbf{x}_k^{(l),(-)} \notin \Gamma(\hat{\mathbf{x}}_{k-1} \in C^\rho), \nu \neq \rho \\ q_3 & \text{if } \mathbf{x}_k^{(l),(-)} \in C^\nu \text{ and } z_k \neq \nu \\ & \text{and } \mathbf{x}_k^{(l),(-)} \notin \Gamma(\hat{\mathbf{x}}_{k-1} \in C^\rho), \nu \neq \rho. \end{cases} \quad (4.39)$$

The first case is when the previous estimated position, the position of the propagated particle, and the measurement vector belong to the same node,  $C^\rho$  indicating that the

AUV is staying on the node. The second case means that the AUV moves to another node, which is a successor node of the previous node. Third and fifth cases deal with inconsistency between the nodes for a measurement vector and for the position of a propagated particle. For the fourth case, the node indicated by the measurement vector and the node for the position of a propagated particle are the same. However, the node is not a successor node of the node where the particle with the highest weight is previously in. Among all the likelihoods,  $q_1$  should be the highest and  $q_3$  should be the lowest.

#### 4. *Computation of weights*

Using the likelihoods generated in the previous step, we compute weights of particles using  $\frac{q(\mathbf{x}_k^{(l),(-)}|z_k)}{\sum_{l=1}^{N_s} q(\mathbf{x}_k^{(l),(-)}|z_k)}$ . Then, we estimate the positions of AUVs from maximum-a-posteriori (MAP) estimation. The estimate  $\hat{\mathbf{x}}_k$  is  $\arg \max_{\mathbf{x}_k} p(\mathbf{x}_k \in C^\rho | \mathbf{Z}_k^i)$ .

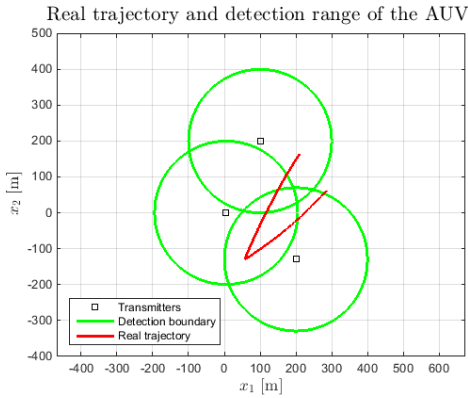
#### 5. *Resampling*

We use a standard resampling technique. After particles are resampled, the localization algorithm returns to Step 2 and repeat from Steps 2 to 5.

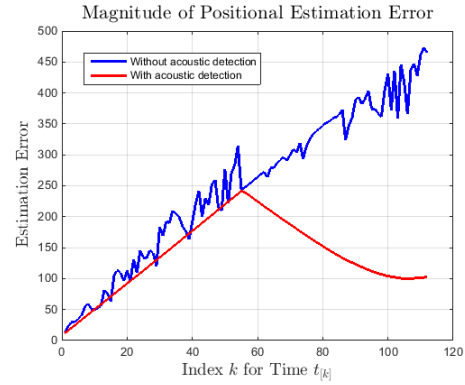
##### 4.1.6 Mathematical Simulation of Acoustic Localization Algorithm

In this section, we describe simulation results for the positional estimation of the AUV. For likelihood on the graph, we select that  $q_1 = 1$ ,  $q_2 = 0.9$ ,  $q_3 = 0.1$ , and  $q_4 = 0.5$ . We select the detection range of the receiver as 200m. Figure 4.5 shows the AUV and the locations of transmitters with circular sensing regions. We simulate one AUV and three transmitters with their circular sensing regions. The AUV moves through sensing regions of one to three transmitters. The unit of flow velocity is m/s. We arbitrary choose flow parameters that represent low and high frequency components of flow. We assume that one component of true flows is  $f_{R_1}(r, \theta, t)$ , which equals  $(10^{-6}r \cos \theta - 0.2) \sin \omega_f t + 0.01$ . The other component  $f_{R_2}(r, \theta, t)$  equals  $(-10^{-6}r \cos \theta + 0.2) \cos \omega_f t + 0.01$ . In terms of

components of modeled flows,  $f_{M_1}(r, \theta, t)$  equals  $(10^{-6}r \cos \theta - 0.1) \sin \omega_f t$ . In addition,  $f_{M_2}(r, \theta, t)$  equals  $(-10^{-6}r \cos \theta + 0.1) \cos \omega_f t$ . Here  $\omega_f = \frac{2\pi}{T}$ , and  $T = 12.42$  hours.  $f_{M_1}$  and  $f_{M_2}$  represent a tidal model of the M2 constituent. Then, the deterministic error growth of odometry evolves at every measurements by maximum bounds  $\varepsilon_{r_0} = 0.2$  and  $\varepsilon_{\theta_0} = 0.03$  for the worst case. For the stochastic eddy flows represented by equation (4.3), we choose Lagrangian correlation time  $\tau$  is five seconds and stochastic fluctuation  $\sigma$  is 0.01. We perform simulations by selecting two waypoints in 2D space: (-400, 200) and (800, 600). Every 1150 seconds, the AUV follows a new waypoint. Figure 4.5 shows the simulated true trajectory of the AUV starting at (285, 62). The line segments of the trajectory are curved because of the difference between true and modeled flows. We assume that each transmitter pings an acoustic signal every 20 seconds. Figure 4.6 shows the magnitude of positional estimation error of the AUV.



**Figure 4.5:** True trajectories of the AUV



**Figure 4.6:** Estimation error

Overall, Figure 4.6, which represents the estimation error of the AUV, shows a trend of reduction in the estimation error of the position of the AUV. The figure shows that the slope of estimation error is slightly changed at 480 sec represented by time index 22 when the AUV detects two transmitters. Estimation error is significantly reduced between 1140 sec and 2050 sec (time index 55 and 100) when the AUV detects two or three transmitters. The value of root-mean-square (RMS) error when the AUV detects transmitters is 148.46m while it is 277.55m without acoustic detection. Thus, the RMS error decreases by around

46.5% with respect to that of odometry using flow estimation. Although the AUV obtains coarse measurements such as on-off measurements from transmitters, the positional estimation error significantly decreases whenever the AUV detects more transmitters.

From the proposed particle filter, the accuracy of positional estimation depends on the likelihood function. The characteristic of the likelihood function shows that the AUV visits the overlapped sensing region with high probability. If the overlapped region is inaccurate due to changing detection range, it is expected that positional accuracy is low. We examine how environmental factors affect acoustic detections with a specially designed acoustic array.

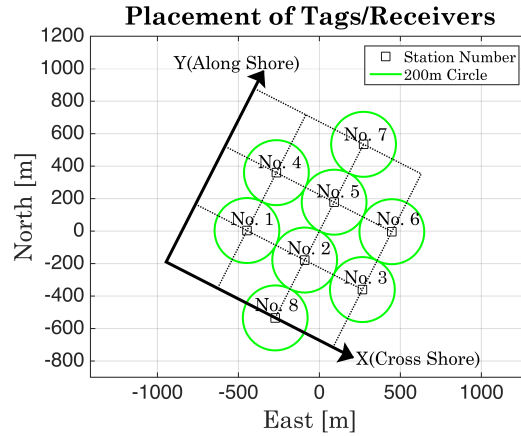
## **4.2 Acoustic Detection Rate**

This section presents detection rate to analyze acoustic detections collected in the specially designed acoustic array. First, we describe configuration of the acoustic array deployed off Savannah, GA in 2014. Then, acoustic detection rate is defined to be a measure of detection efficiency for the identification of influence from environmental factors.

### 4.2.1 Configuration of acoustic array

Eight Vemco VR2W receivers and 14 Vemco V13 tags were deployed in August 2014 in a static array at Gray's Reef National Marine Sanctuary (GRNMS), located 40 nm SE of Savannah, GA. The area is designated for controlled scientific study on more than 200 species of fish by the US National Oceanic and Atmospheric Administration (NOAA). The placement of receivers and tags in the horizontal plane is shown in Figure 4.7; water depth is approximately 21m throughout the receiver array. The array was designed with receivers 400 meters apart, based on results from [34], which found that detection rate at 200 meters distance falls to a minimum 8% during late summer in August compared to 97% in February. A 600 kHz upward-looking acoustic Doppler current profiler (ADCP) [61] was moored approximately 1.1km away to remove the potential for interference between

the ADCP and the acoustic array (Vemco, pers. comm.).



**Figure 4.7:** The two dimensional placement of receivers and tags: The origin of the figure is the center of the array, located 3.09-km ENE of the National Data Buoy Center (NDBC) buoy at Gray’s Reef (31.400N, 80.868W). The circle about each receiver reflects a nominal 200-m radius of detection during summer as observed by [34].

Figure 4.7 shows the array of receivers and tags installed at Gray’s Reef, with the station number indicated. We define two directions that represent the direction of combination of receivers and tags: alongshore direction and cross-shore direction. For instance, the pair of one tag in station 1 and one receiver in station 2 is aligned with the direction of cross shore, 27.3 degrees clockwise of east. The pair of one tag in station 1 and one receiver in station 4 is aligned with the alongshore direction. Receiver 3 is removed from this statistical analysis because of receiver malfunction.

The number of tags are distributed according to station number and depth in Figure 4.7. Each station has from one to three tags at 18.6 meters (near-surface), 12.5 meters (mid-depth), and 6 meters (near-bottom) depth from bottom. Using the acronyms S, M, and B that represent surface, mid-depth, and bottom depth, respectively, we index the tags by station number and vertical placement: 1S, 1M, 1B, 2M, 3S, 3M, 3B, 4M, 5S, 5M, 6M, 7S, 7M, 8M. When the receiver’s omnidirectional hydrophone detects the transmitter, the receiver records time stamps of detection and identification number of the transmitter. Each surgically-implantable transmitter [62] is programmed to send one ping randomly in each 45-second interval at 69kHz with power output 147dB. Minimum delay and maximum

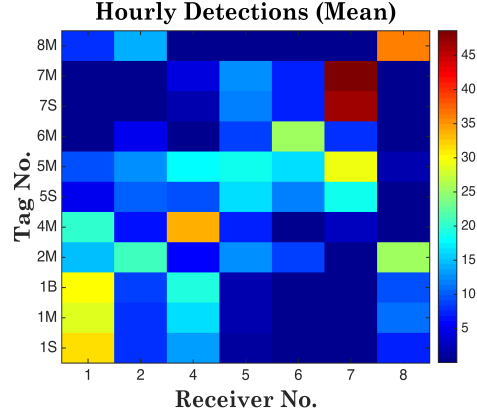
delay for randomly pinging are zero and 90 seconds, respectively.

#### 4.2.2 Detection rate

Many studies [63, 64] have focused on detection probability, defined as the ratio of the number of detections to the number of pings. Detection probability is a measure of the efficiency of detection with the relations of input and output for an acoustic channel that can be modeled by an acoustic sound propagation [64]. Estimating input by ping over each 45s interval with this configuration is not accurate and can lead to significant error of detection probability. Hourly detections computed by the number of detections over a longer interval (1 hr) is a more representative metric for detection efficiency. With one ping randomly each 45s, we expect a mean value of 80 pings per hour, and detection “rate” in any one hour interval is thus compared to the mean value of 80 hourly detections. Figure 4.8 shows the mean hourly detections at the receivers by tag. However, the maximum observed detections is less than 50 per hour, with only two receivers with hourly detections greater than 40. The proportion of measured hourly detections to the expected value of 80 serves as a proxy for detection probability. However, since the ratio is seldom larger than 50%, we look to signal collision as a potential cause for some of this loss.

We find the receivers with the highest signal collision among eight receivers by comparing the mean hourly detections. We first focus on the diagonal, which shows the number of detections at tags that are co-located with the receivers. The distance between the receiver and the tag at the same station is minimum 6 meters and maximum 19 meters, shorter than any distance between the receiver at one station and the tag at the other station. Because detection probability is assumed to be highest over the shortest distance [32], we expect co-located tags and receivers to have the largest number of hourly detections.

However, Receiver 2 recorded fewer than 20 hourly detections of Tag 2M. In contrast, Receiver 8 has more hourly detections of Tag 2M despite its distance of over 400m from the Tag. In addition, the comparison of mean hourly detections of Tag 5M shows that Receiver



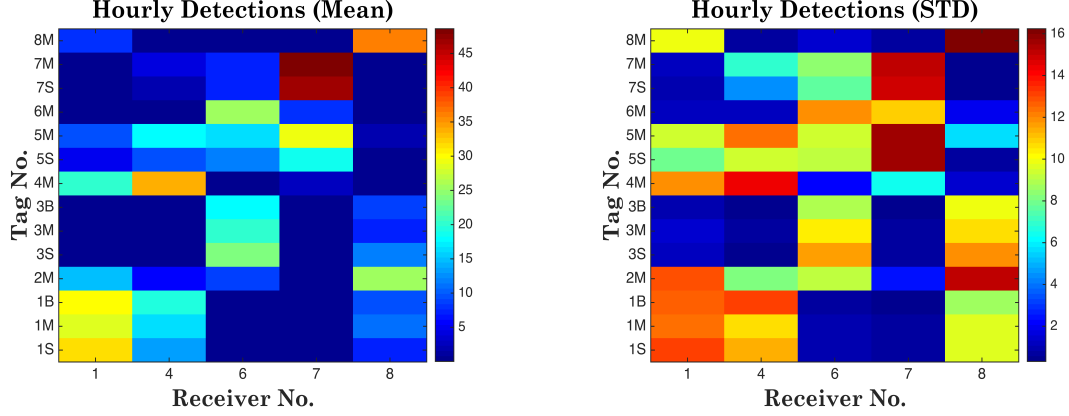
**Figure 4.8:** Mean hourly detections: The horizontal and vertical axes in the figure shows the index number of receivers and tags, respectively. The color bar represents mean values of hourly detections.

7, which is 400 meters away from Tag 5M has more detections than Receiver 5, located just 12 meters away from Tag 5M. It is worth noting that the vertical position of the receiver and tags are different, and stratification may prevent clear transmission in the vertical. However, we assume that conditions are spatially invariant, and this loss is constant over the array. Because these receivers are centrally located within the array, we suspect that signal collision may contribute to the lower than expected hourly detections compared to other co-located receiver/tag pairs. Receivers 2 and 5 are located in the densest part of the array, with ten and seven tags within 400 meters distance, respectively. Signal collision probability of 70% is predicted by the metrics in [65]. We therefore remove these receivers from the subsequent analysis.

We find Receivers 2 and 5 with the highest signal collision among eight receivers by comparing the mean hourly detections [66], and then examine variability of hourly detections over time. Figure 4.9 shows mean and standard deviation of hourly detections about the pairs of five receivers and 14 tags. When we ignore co-located receiver/tag pairs, the largest number of mean hourly detections is found between Receiver 1 and Tag 4M, Receiver 4 and Tag 1B, Receiver 6 and Tag 3S, Receiver 6 and Tag 3M, Receiver 7 and Tag 5M, Receiver 8 and Tag 2M. The direction of each of these pairs is aligned with alongshore direction in Figure 4.7, suggesting that the pairs parallel to alongshore direction have more



detections than cross-shore direction. Standard deviation of hourly detections follows the same pattern. Most pairs aligned with alongshore direction have the second highest standard deviation of hourly detections. Further, the standard deviation of the pairs is fairly large, and is comparable to the mean value. This result motivates closer examination of the source of detection variability.



**Figure 4.9:** Mean hourly detections (on the left) and standard deviation of hourly detections (on the right)

### 4.3 Tidal Analysis

We formulate our analysis of detections with respect to tidal currents, and will present comparison of detection range with respect to cross- and alongshore tidal variability. In particular,  $M_2$ ,  $N_2$ ,  $S_2$ ,  $K_2$ ,  $O_1$ ,  $K_1$ ,  $P_1$ , and  $Q_1$  are eight tidal constituents that explain 90% of the variance in measured depth-averaged flow. If the signal-to-noise ratio (SNR) of a tidal constituent is greater than and equals 1, the tidal constituent is significant in hourly detections.

We choose eight combinations of receivers and tags in the configuration. Because acoustic sound propagation varies according to the range of receivers that detect tags, we pick tags 1M, 2M, 3M, 4M, and 5M installed at two-dimensional distance 400 meters and depth 8.5 meters from Receivers 1, 4, 6, 7 and 8 to maintain the same range of each re-

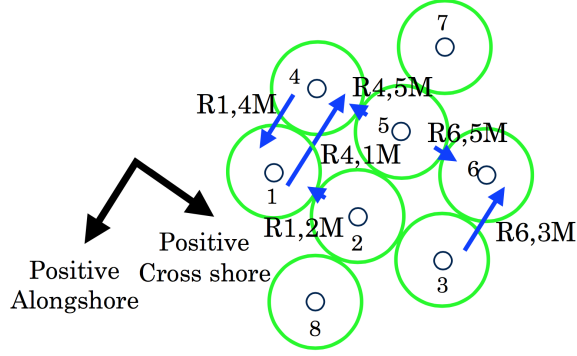
ceiver/tag pair in the 3D space. The `t.tide` Matlab toolbox [67] extracts harmonic fits to astronomical tidal frequencies given a scalar or vector time series. 35 tidal components are identified from the 53-day record of hourly detections. We choose these eight largest constituents of tidal current variability, summarized in Table 4.1, for joint analysis of signal detection. In addition to amplitude and phase, the `t.tide` toolbox calculates signal-to-noise ratio (SNR) by the square of the ratio of amplitude to amplitude error at the 95% confidence level.

**Table 4.1:** Signal to noise ratio of tidal constituents <sup>†</sup>

Tidal	R6,5M	R6,3M	R4,5M	R4,1M	R1,2M	R1,4M	R7,5M	R8,2M
$M_2$	<b>5.2</b>	<b>23</b>	<b>1.5</b>	<b>34</b>	<b>1.2</b>	<b>16</b>	0.15	0.71
$N_2$	<b>4.5</b>	<b>4.1</b>	0.24	<b>3.7</b>	<b>0.91</b>	<b>1.8</b>	<b>1.7</b>	0.26
$S_2$	0.74	<b>1.4</b>	<b>1.7</b>	<b>4.9</b>	<b>10</b>	<b>1.6</b>	<b>2.5</b>	<b>34</b>
$K_2$	<b>6.2</b>	<b>5.7</b>	0.87	<b>14</b>	<b>17</b>	<b>12</b>	<b>2</b>	<b>59</b>
$O_1$	<b>14</b>	0.62	0.35	<b>1.74</b>	<b>1.4</b>	<b>2</b>	<b>3.2</b>	<b>1.4</b>
$K_1$	<b>10</b>	<b>23</b>	<b>7.7</b>	1	0.78	<b>1.4</b>	<b>24</b>	<b>17</b>
$P_1$	<b>8.8</b>	<b>9.9</b>	<b>6.6</b>	<b>9.2</b>	<b>5.3</b>	<b>2.4</b>	<b>24</b>	<b>8.5</b>
$Q_1$	0.81	<b>3.4</b>	<b>4.2</b>	<b>2</b>	0.049	<b>3.1</b>	0.4	<b>1.6</b>

<sup>†</sup> Values of SNR > 1 are given in bold

SNR values for the eight major tidal constituents are given in Table 4.1. Receiver/tag pairs are given in the notation defined in Section 4.2.1. All eight major tidal constituents are represented with SNR > 1 for the following pair: Receiver 4 and Tag 1M. However, Receiver 6 and Tag 5M, Receiver 4 and Tag 5M, Receiver 1 and Tag 2M, Receiver 7 and Tag 5M, and Receiver 8 and Tag 2M have two or three constituents with SNR below 1. Receiver 6 and Tag 3M contains seven of the eight but is not significant for  $O_1$ . Receiver 1 and Tag 4M contains seven of the eight but is not significant for  $K_1$ . We focus on the  $M_2$  tidal constituent, which explains approximately 80% of the measured current variance at GRNMS. All combinations but R7,5M and R8,2M are significant for the  $M_2$  tidal constituent. Thus, we remove the last two columns of the table to choose six pairs to analyze the relationship between flow direction and hourly detections.



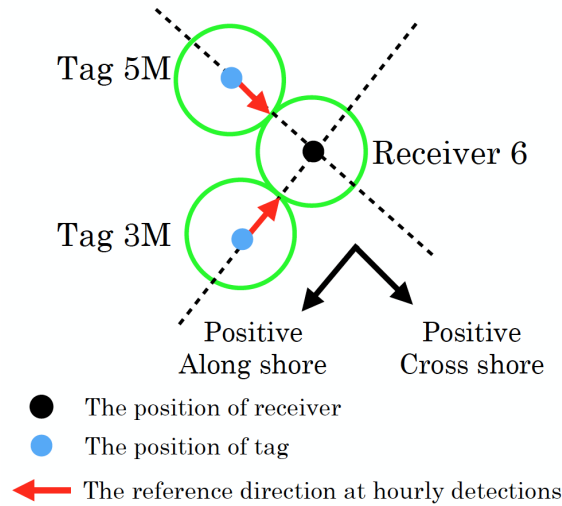
**Figure 4.10:** SNR on the configuration: The direction of each arrow represents the direction of the tag toward the receiver. The length of each arrow represent the magnitude of SNR. Alongshore pairs have much stronger SNR than cross-shore pairs

Figure 4.10 shows that the highest SNR at the  $M_2$  frequency is found between transmitter/receiver pairs that are oriented alongshore. SNR is greater than 15 between R4,1M, R6,3M, and R1,4M but much lower between the cross-shore pairs. In addition, the variability in Section 4.2.2 shows receiver-tag pairs aligned with alongshore direction tend to have higher mean and standard deviation of hourly detections than cross-shore direction, overall, not just at specific tidal frequency. The dominant tidal frequency  $M_2$  is strong in high hourly detections, suggesting that the fluctuation of hourly detections is significantly related to tidal flows. These spatial patterns of tidal variability suggests that the shape of detection range is not uniform, but rather changes significantly in space and time.

#### 4.4 Flow Direction

We identify the relationship between hourly detections and flow direction. We show that time series of hourly detections of pairs oriented in the alongshore direction contain significant tidal variability at the  $M_2$  frequency [66]. To investigate the relationship between direction of tidal currents and detection probability further, we consider the time series of hourly detections and flow components aligned with the receiver/tag pair. The acoustic signal path length, or the distance sound must travel between the tag and receiver, may be reduced when the current is in the direction from the tag to the receiver; when the current

opposes the direction from the tag to the receiver, the acoustic signal path length may be increased. Figure 4.11 shows shore direction and the reference direction of acoustic signal path at hourly detections. The acoustic signal path length is main factor that affects detections of receivers. If the acoustic signal path is long, then signal power is greatly attenuated, and this leads to non-detections of receivers; the receivers has much more likely to detect signal when the acoustic signal path is short. The red arrows in Figure 4.11 indicate directions that make shortest acoustic signal path lengths. When Tag 3M transmits a signal to Receiver 6 along the direction of the red arrow starting at Tag 3M, the acoustic signal path is the shortest in that the straight line between Tag 3M and Receiver 6 is the shortest distance. Thus, we let the direction of the red arrow be the reference direction at hourly detections.

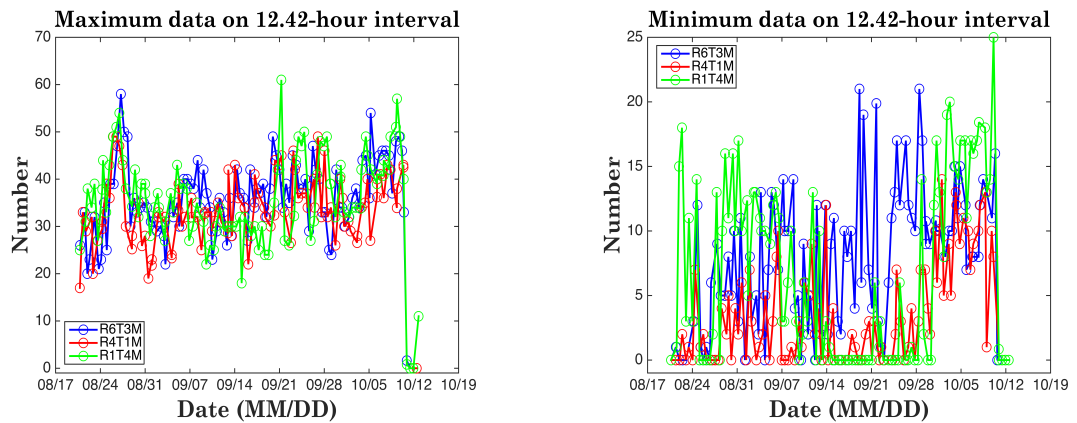


**Figure 4.11:** The reference direction at hourly detections

The ADCP measures horizontal and vertical flow as a function of time and depth. To obtain along- and cross-shore components of flow measurements, we eliminate bins close to the surface that have been contaminated with surface effects and side lobe interference, and form a depth-average from the remaining bins. Depth-averaged flow is decomposed into along- and cross-shore components. we assume that 1) flow does not vary significantly over the array, and 2) the 1.1km distance to the ADCP from the array is smaller than

the length scale of variability; in other words, ADCP data are representative of the flow measurements within the array. With the two assumptions, we compare the time series of hourly detections between the receiver/tag pair to the along- or cross-shore component of flow. First, we window detection data every tidal period, and find when windowed data are maximum and minimum. Then we compile a time series of maxima and minima on detections. Second, because we know when along- and cross-shore components are maximum and minimum, we also compile a time series of maxima and minima on flow components. By comparing two time series of maxima and minima, we can examine flow direction at maxima and minima on detections to see if there is a consistent relationship between detection probability and tidal phase.

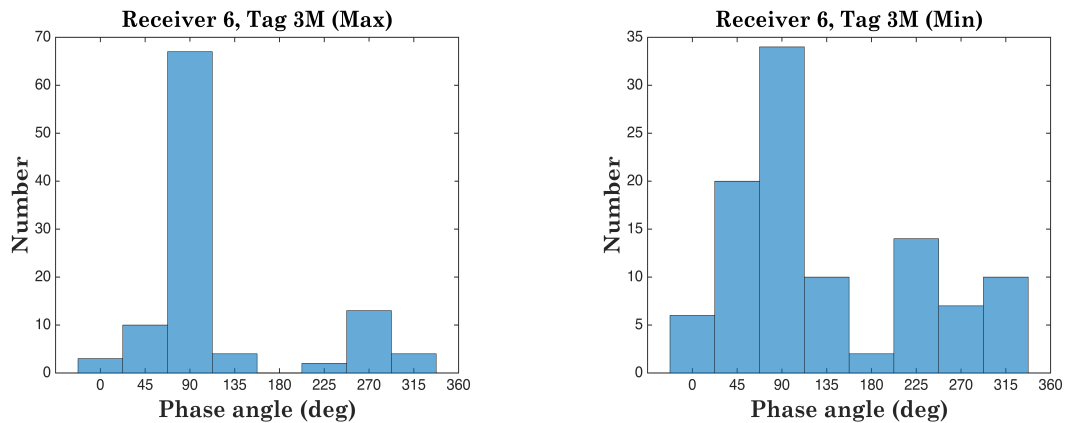
We compute time differences between the rate of detections and flow components in that we have a time series of maxima on detections and on flow components; in addition, we have a time series of minima on detections and on flow components. Then we have a time series of computed phase angles between the rate of detections and flow components. Because we know the reference direction of each receiver/tag pair shown in Figure 4.11, we can identify flow directions when the rate of detections is maximum and minimum by using a histogram approach.



**Figure 4.12:** Maximum (left) and minimum (right) hourly detections, taken over each successive  $M_2$  tidal cycle, for three receiver-tag pairs oriented alongshore.

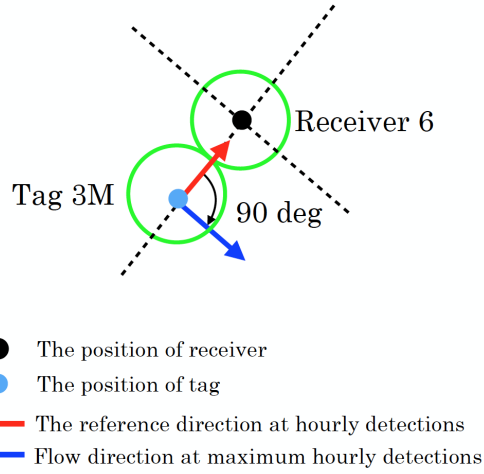
Figure 4.12 shows maximum and minimum hourly detections for each successive 12.42-

hour interval. In the left panel of Figure 4.12, maximum hourly detections at all three receiver/tag pairs are consistent, with a relatively sharp increase around August 24th, and a slower increase over the deployment period. This change may be caused by a frontal passage that led to cooling and loss of stratification in the transition from summer into early fall. There is no obvious spring/neap cycle in the rate of detection. In contrast, the right panel shows that minimum hourly detections have no consistent pattern among the receiver/tag pairs and overall. For example, there are no detections of Tag 4M at Receiver 1 for much of mid-to late September, while minimum detection rates of Tag 3M at Receiver 6 are much higher than any other pairs. At that time, minimum detection rates of Tag 3M at Receiver 6 (20 detections per hour) is only 50% lower than maximum detection rates of Tag 3M at Receiver 6 (43 detections per hour). This trend is not seen in the time series of detection rates of other two pairs.

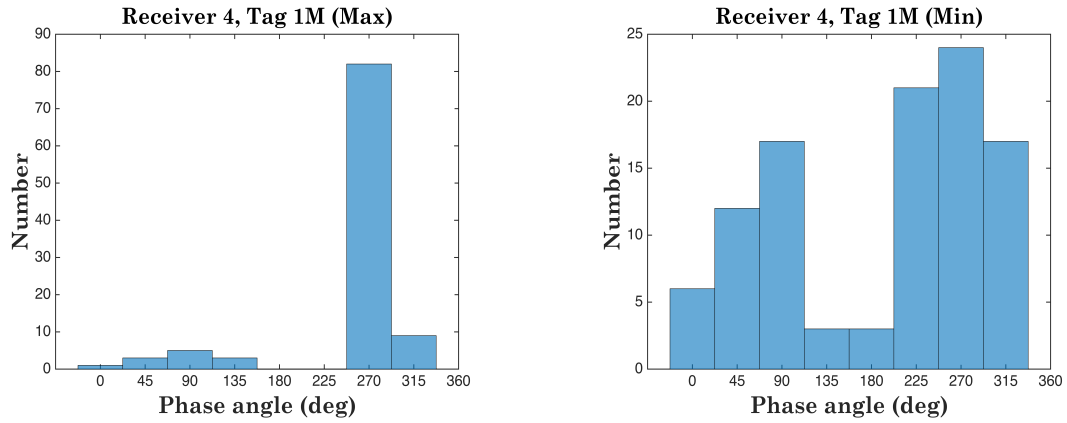


**Figure 4.13:** Phase of flow with respect to the direction from Tag 3M to Receiver 6 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (negative alongshore, NNE); phases of 90 and 270 degrees correspond to offshore/positive and onshore/negative cross-shore flow, respectively.

Figures 4.13, 4.15, and 4.16 show the distribution of phase angle of flow at the time of maximum and minimum hourly detection for one alongshore pair. Phase of flow is given with respect to the reference angle, in the direction of acoustic path from the tag to the receiver, and is positive clockwise. Figure 4.14 shows flow direction when detection rates of Tag 3M at Receiver 6 is maximum shown in the left panel of Figure 4.13. The



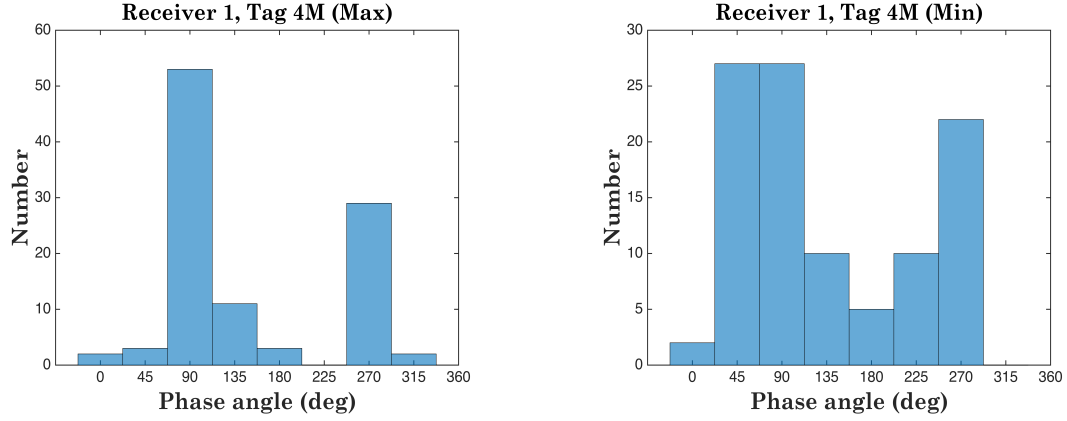
**Figure 4.14:** Flow direction at minimum detection rates of Tag 3M at Receiver 6



**Figure 4.15:** Phase of flow with respect to the direction from Tag 1M to Receiver 4 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (negative alongshore, NNE); phases of 90 and 270 degrees correspond to offshore/positive and onshore/negative cross-shore flow, respectively.

distributions in the left panels are bimodal, with strong maxima at 90 or 270 degrees; these angles represent flow component in the cross-shore direction. In contrast, the right panels showing phase angle with time of minimum detections are more evenly distributed over the full range of angles. This result suggests that the relationship between flow direction and increased detection probability is not related to the acoustic path but rather the phase of the tide relative to the reference direction.

From analyzing the relationship between alongshore pairs detections and flow direction

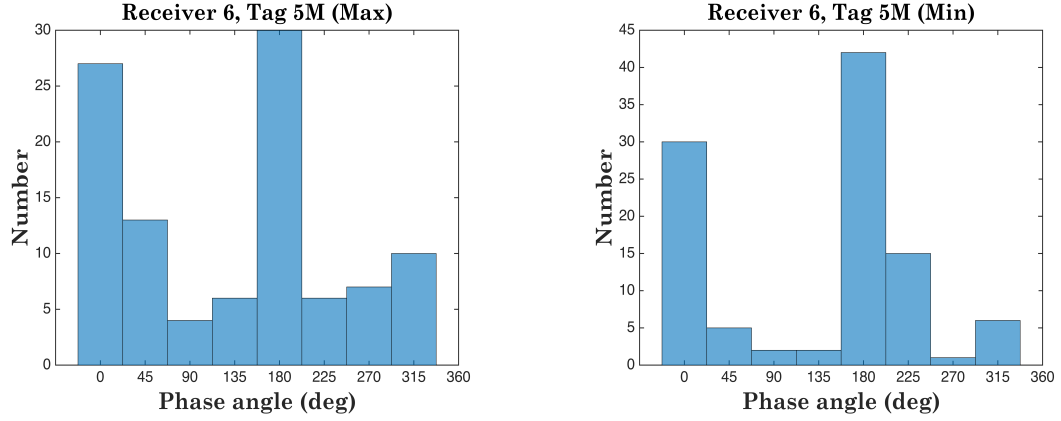


**Figure 4.16:** Phase of flow with respect to the direction from Tag 4M to Receiver 1 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (positive alongshore, SSW); phases of 90 and 270 correspond to onshore/negative and offshore/positive cross-shore flow, respectively.

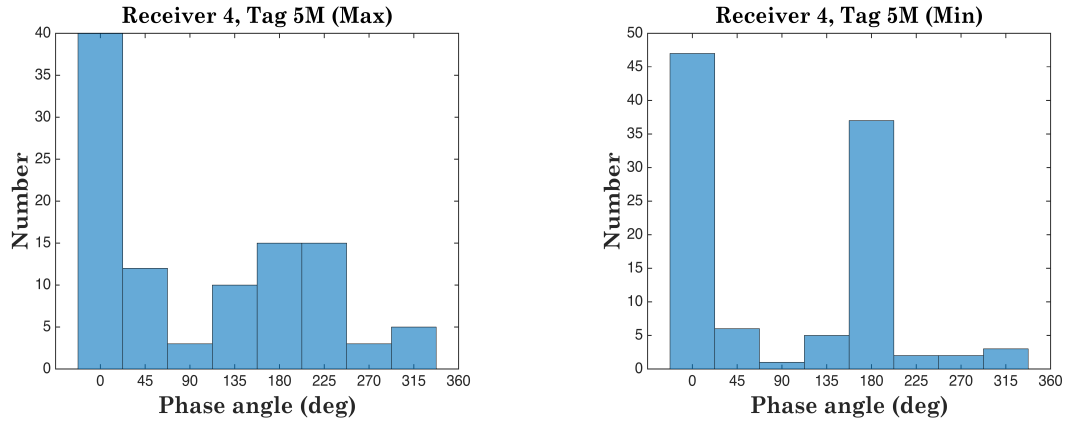
through a histogram approach, we hypothesize that Doppler frequency shift may affect detection among alongshore pairs. Doppler frequency shift is determined by relative motion between sound and the receiver. If flow is strong in the direction from the tag to the receiver, sound signal may be shifted to a slightly higher frequency. For example, Doppler frequency shift is 4.6Hz when flow speed is 50cm/s, and the frequency of transmitting signal is 69kHz. If the receiver is sufficiently sensitive to the frequency of the transmitted signal, it may not detect the transmission. Because cross-shore flow is much stronger than alongshore flow in Gray's Reef, Doppler frequency shift in cross-shore detection could be strong and it may lead to minimum detections of cross-shore tag-receiver pairs when flow direction is aligned with cross-shore direction.

Figures 4.17, 4.18, 4.19 show the distribution of flow at the time of maximum and minimum detection for cross-shore pairs selected from SNR analysis in Section 4.3. The distributions in the right panels are bimodal, with strong minima at 0 or 180 degrees; these angles represent the cross-shore direction. In contrast, the left panels showing phase angle with time of maximum detections are more evenly distributed over the full range of angles. Doppler frequency shift could potentially explain the lower rate of detection among cross-shore pairs than among alongshore pairs.





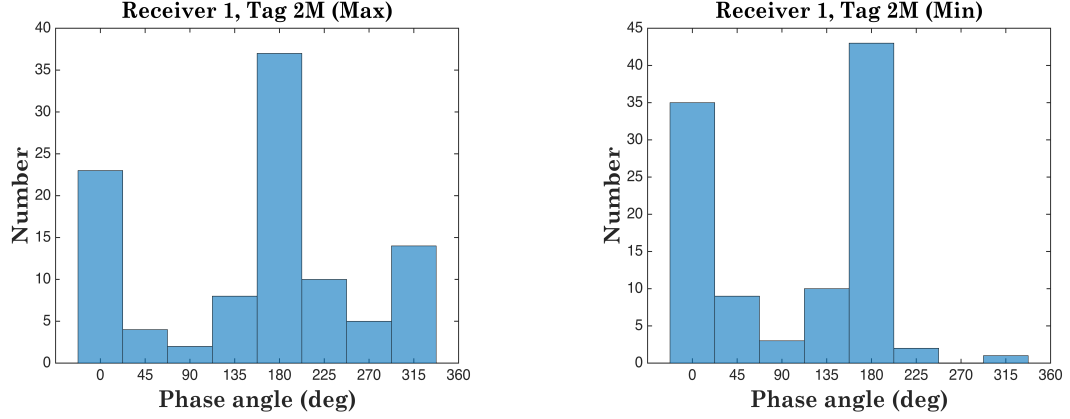
**Figure 4.17:** Phase of flow with respect to the direction from Tag 5M to Receiver 6 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (positive cross-shore, ESE); phases of 0 and 180 degrees correspond to offshore/positive and onshore/negative cross-shore flow, respectively.



**Figure 4.18:** Phase of flow with respect to the direction from Tag 5M to Receiver 4 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (negative cross-shore, WNW); phases of 0 and 180 degrees correspond to onshore/negative and offshore/positive cross-shore flow, respectively.

## 4.5 Stratification

Tidal currents are largely oriented cross-shore on the Georgia shelf.  $M_2$  tidal ellipse orientation is cross-shore at Gray's Reef, so offshore or onshore flow is associated with the largest tidal current magnitude. When tidal flow is stronger, the height of the bottom boundary layer can extend higher into the water column [68, 69], which can lead to a tidal asymmetry in mixing and stratification. Since acoustic propagation is enhanced under well-mixed



**Figure 4.19:** Phase of flow with respect to the direction from Tag 2M to Receiver 1 at maximum hourly detections (left) and at minimum hourly detections (right). Angle is given in degrees clockwise of the reference angle (negative cross-shore, WNW); phases of 0 and 180 correspond to onshore/negative and offshore/positive cross-shore flow, respectively.

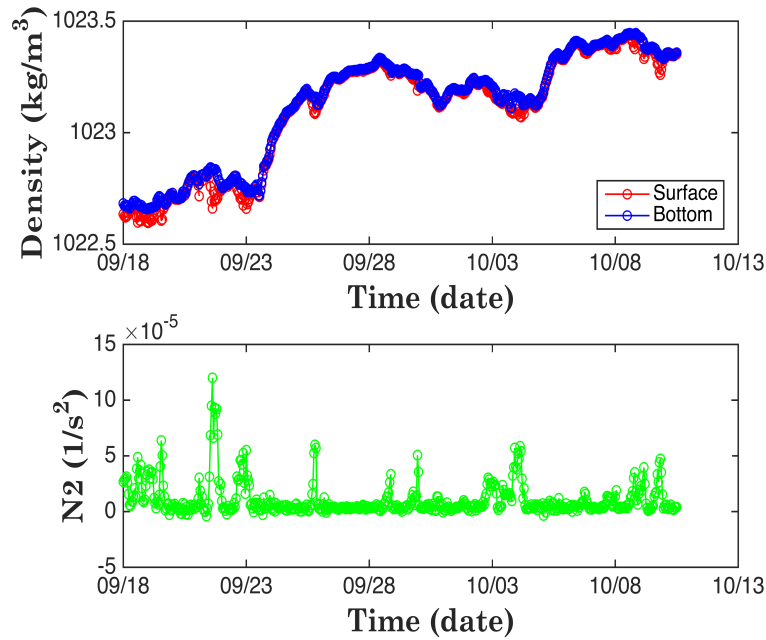
conditions, this mechanism may explain why maximum hourly detections are associated with cross-shore flow along the  $M_2$  semi-major axis. The relative shallow depth at Gray’s Reef (21-m) makes it likely that the frictional boundary layer extends through most or all of the water column, and that tidal variation in mixing can affect stratification in the mid- or upper layers.

For the investigation of water stratification in GRNMS, an underwater glider was deployed for about 23 days during the acoustic telemetry experiment. The underwater glider, equipped with conductivity, temperature, and density (CTD) sensors, traveled around GRNMS to consistently collect CTD data as shown in Figure 4.20.

The top panel in Figure 4.20 is collected density data from the CTD sensors. Blue represents density data at the surface; red represents density at the bottom. A bulk estimate of the Brunt-Vaisala frequency  $N^2$  index is calculated [70] from near-surface and near-bottom values of glider-measured density from the following equation:

$$N^2 = -g \frac{1}{\rho} \frac{d\rho}{dz}, \quad (4.40)$$

where  $g$  is gravitational acceleration ( $9.81 \text{ m/s}^2$ ),  $z$  depth ( $\text{m}$ ), and  $\rho$  water density ( $\text{kg/m}^3$ ).



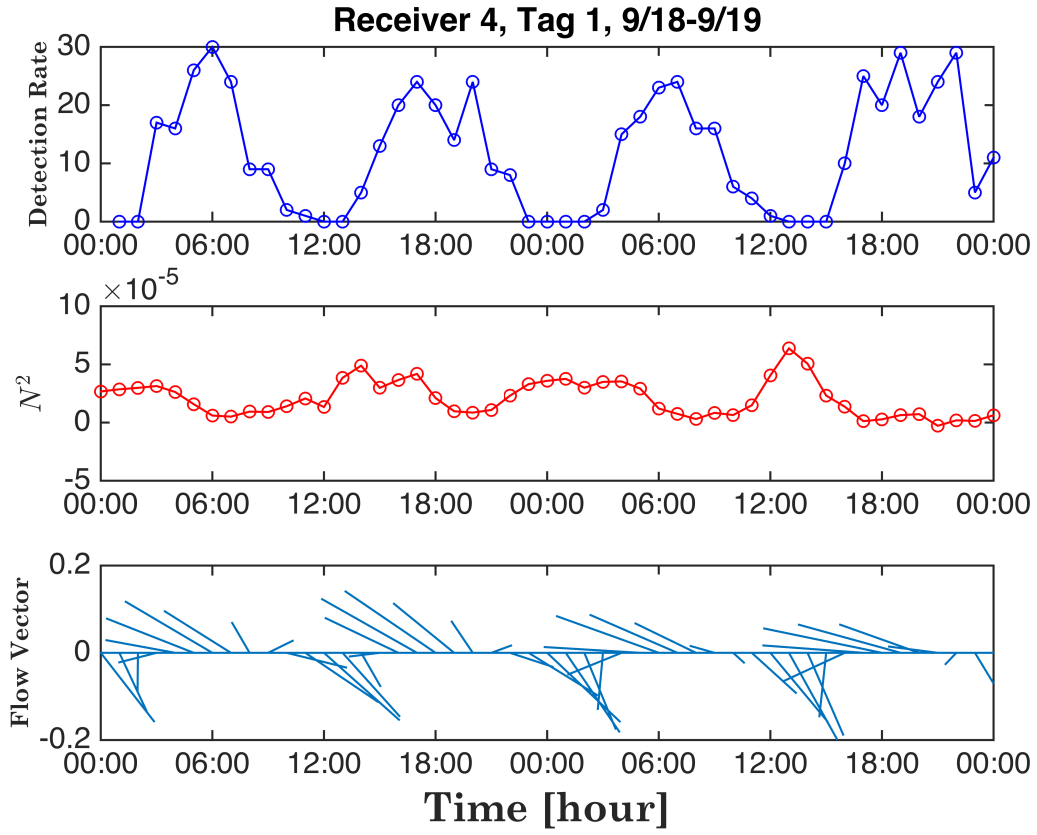
**Figure 4.20:** Collected density data (top), and post processed  $N^2$  stratification index (bottom)

$N^2$  is high when water column is stratified;  $N^2$  is low when water column is less stratified. The time series of stratification is shown in the bottom panel of Figure 4.20.

Figure 4.21 shows time series of hourly detection of the alongshore pair (Receiver 4 and Tag 1),  $N^2$ , and flow vectors during 48 hours beginning September 18. This period is representative of variability during the initially stratified portion of the experiment, and the glider is closest to the acoustic array. As suggested by tidal analysis of detection rate, detection rate has a visible semidiurnal trend, with one maximum and minimum approximately every 12.42 hours.

The time series of bulk stratification also shows a semidiurnal pattern, with stronger stratification when the currents are directed offshore, and weaker stratification with onshore flow. A weak diurnal pattern in stratification becomes more apparent in longer time series, but diminishes with the first fall storm.

The maximum detection rate roughly coincides with the onshore cross-shore flow and minimum stratification. For example at 06:00 and 18:00 September 18th, and 07:00 and



**Figure 4.21:** Hourly detections (top),  $N^2$  (middle), and depth-averaged ADCP currents (bottom)

19:00 September 19th, detection rate is greatest while stratification is low and currents are directed offshore. However, while the previous analysis suggested that both onshore and offshore flow may be associated with an increase in detection rate, the time series of detections and currents do not show consistent patterns with stratification at quarterdiurnal frequencies (i.e., every 6.21 hours), as would be expected if the detections vary more strongly with current magnitude. The increase of stratification with offshore flow may be similar to Strain Induced Periodic Stratification (SIPS, Simpson et al., 1990 [71]) described in estuaries.

Future work will investigate the links between stratification and detection rate using a combination of glider, acoustic, and oceanographic data. Wavelets or multiple short term harmonic analysis can be used to reveal when tidal variability is important to detection

rate, and link changes in frequency and magnitude to oceanographic conditions. Future work will also investigate the effects that a mobile platform may have on detection rate. Preliminary work with an inertial model developed for this experiment (not included in this thesis) suggests that vehicle attitude and angle between the tag and receiver may cause significant differences between predicted and actual detection rates. Future work could detectability with respect to both vehicle motion and environmental variability.

## CHAPTER 5

### DEVELOPMENT OF AN INDOOR TEST BED AND EXPERIMENTAL RESULTS

The verification of the anomaly detection algorithms from the use of actual autonomous underwater vehicles (AUVs) requires spending significant amount of time and money, and may suffer the risk of vehicle loss. Generally, deploying AUVs is restricted by environmental conditions; for instance, the severe weather such as heavy rain and high wave prohibits vehicle deployment. Moreover, underwater gliders, one class of AUVs that have been widely used by oceanographers, are too expensive to be used to test the anomaly detection algorithms repeatedly. We need intentional faults to be made on the underwater gliders which may lead to vehicle loss.

This dissertation verifies the proposed algorithms derived in Chapters 3 and 4 with indoor experimental results. Indoor experiments prove the value of the algorithms. For this purpose, we develop one flying robot and one ground robot: the Georgia Tech Miniature Autonomous Blimp (GT-MAB) and the Georgia Tech Wind Measuring Robot (GT-WMR).

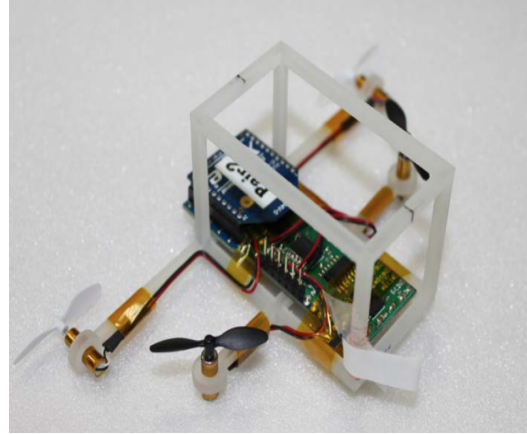
#### 5.1 Georgia Tech Miniature Autonomous Blimp (GT-MAB)

There is a need for small flying vehicles to support autonomy research. Unmanned aerial vehicles such as quad-rotors and multi-copters have become popular for this purpose. However, the indoor usage of these unmanned aerial vehicles (UAVs) is limited by a number of factors. UAVs usually have short flight durations per battery charge, typically less than 20 minutes, which restricts the duration of experiments. Some UAVs have large spinning propellers and some fly at relatively high speeds. This can cause safety concerns for humans sharing the same space. Safety nets or cages are usually installed for human protection, which limits the potential for human-robot interaction experiments.

We develop miniature autonomous blimps (MAB) as flying vehicles for indoor exper-



**Figure 5.1:** Children play with the GT-MAB



**Figure 5.2:** The GT-MAB's gondola includes four motor thrusters. Two motor thruster are vertically installed to control up-down motion. The other two motor thrusters are horizontally installed to control forward-backward and spinning motions, which are presented in [72].

iments that can support research on mobile sensor networks, human-robot interaction, 3D motion control, networked robotics, and other aspects of autonomy. The GT-MAB (Georgia Tech Miniature Autonomous Blimp) has relatively long flight durations of up to two hours per battery charge. Furthermore, the blimps are naturally cushioned and do not cause any harm if they collide with a human. It offers a fun experience that encourages physical contact, as illustrated in Figure 5.1. With its small size, low cost, and safe operation, the GT-MAB also serves to educate and excite young students about robot design and control.

A significant body of literature exists for both outdoor and indoor robotic blimps. Earlier developments are very similar to airships [73, 74]. Indoor blimps have been previously developed for entertainment [75], artistic performance in museum [76], and telepresence [77] and also for the emulation of underwater vehicles [78]. A class of small blimps was previously designed for robotics research on indoor localization and mapping [79]. The advantages of blimps over other aerial vehicles, including reduced energy consumption, lower cost, and better safety, have been noted in the work reviewed.

The GT-MAB is a unique design in several perspectives. It is perhaps the smallest indoor robotic blimp up to the time this thesis is written. The blimp envelop has an el-

lipsoidal shape with the lengths of the semi-major and semi-minor axes as 0.36 and 0.23 meters, respectively. The smallest previous blimp design known in the literature has the length of the semi-major axis greater than 0.5 meter [80]. Being small allows multiple blimps to be flown at the same time to support indoor experiments on swarming and sensor networks. But being small also causes limited load capacity, which has motivated us to design small and lightweight driving and sensing hardware systems. Recent advancements in electronics, computing, and MEMS allow us to achieve this goal. The GT-MAB is also highly maneuverable. Most previous designs [81, 78, 82, 79] inherited the envelop design of airships and employ longitudinal tail fins for maneuvering. This design favors stable forward cruise motion for long-range flights, but it is not the optimal choice for maneuverability in an indoor lab setting. The GT-MAB uses a “saucer-shaped” envelope that makes turning motions easier. Multiple propellers can provide vector thrusts that achieve better maneuverability in indoor environments, see Fig. 5.1 and 5.2.

#### 5.1.1 Physical design

The physical design of the GT-MAB reflects a balance among design challenges such as stability of the structure, limited payload, and maneuverability. The GT-MAB has two major modules, a “saucer-shaped” envelope and a gondola (see Fig. 5.1). The center of gravity is below the center of buoyancy, which facilitates dampening of pitching and rolling motions and hence increases the stability of operation. The symmetric shape of the envelop allows easy spinning in place, which increases the maneuverability.

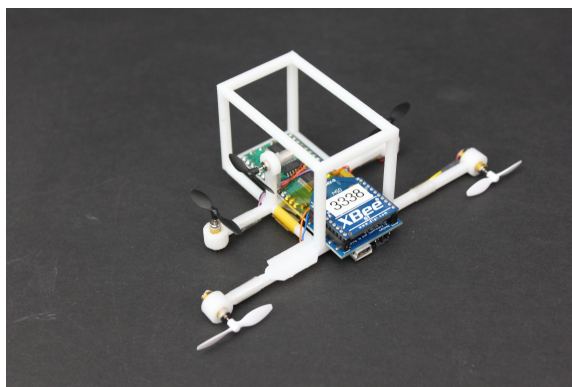
Fig. 5.2 shows a gondola that is attached to the bottom of the envelope near the center. The gondola is a 3D printed housing for the control and sensing hardware, which includes motors, propellers, a microprocessor, an inertial measurement unit (IMU), and other sensors. The limited lift provided by the envelope imposes a constraint on the total weight of the gondola. This constraint was satisfied by a careful selection of electronic components and a light-weight design of the mechanical structure. The total weight is 85.9 grams, and



offers 12.1 grams for payload.

The motion of the GT-MAB is generated by four motors connected to small propellers that are mounted to the gondola. This particular configuration of the four motors gives the blimp high maneuverability. The two motors, mounted facing upwards near the center-line, provide counter torque to each other and stabilize the upward motion of the blimp. The other two motors facing the head of the GT-MAB provide counter torque to stabilize the horizontal motion of the blimp and can also be used to provide differential thrusts for quick turning motion.

When we consider GT-MAB motion under strong flow generated by an artificial wind source, the two motors facing the head of the GT-MAB may not be enough to provide thrusts for horizontal motion. If error along the side-way direction would occur because of strong flow, the two motors could not compensate error in that the motors do not generate thrust for side-way motion of the GT-MAB. Therefore, we add one more motor to the current gondola so that the GT-MAB has additional motion control. The Figure 5.3 shows the gondola that contains five motors.



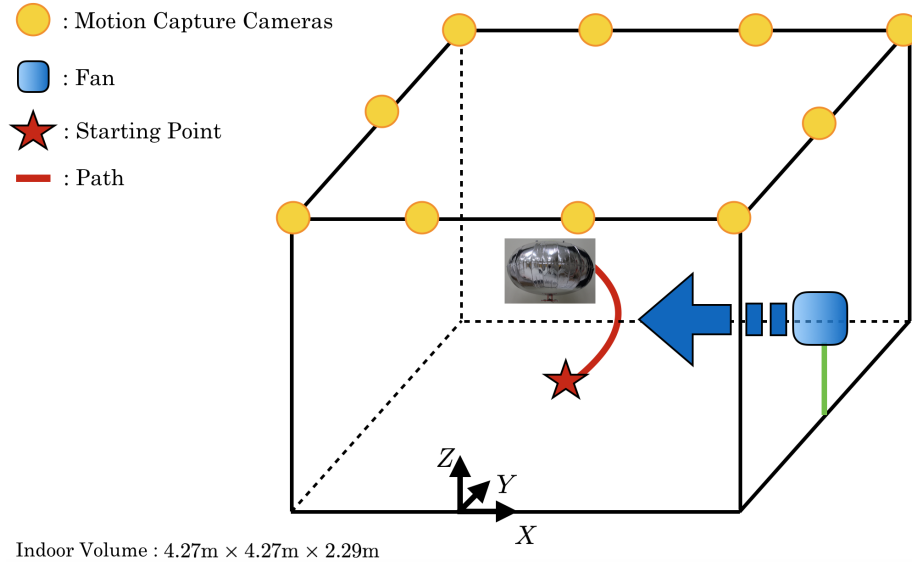
**Figure 5.3:** This gondola has one more motor than that of Figure 5.2. The motor installed along the side-way direction provides thrust for side-way motion of the GT-MAB.

## 5.2 GT-MAB Field Experiments with GT-WMR

We choose the GT-MAB as a controlled Lagrangian particle for the verification of the

proposed algorithms in this dissertation. The GT-MAB has dynamics that are similar to the dynamics of underwater vehicles [83],[84], [85], [78]. The lighter gas of the GT-MAB induces buoyancy, which plays the same role in restoring force and moment of the GT-MAB as underwater vehicles. On top of that, the GT-MAB is subjected to significant fluid dynamic influences, which are common to underwater vehicles.

In order to generate flow that affects the motion of the GT-MAB, a Dyson fan is used as an artificial wind source. The Dyson fan creates more consistent flows along the direction of blowing wind than rotating fans that produce inconsistent flows. The use of the Dyson fan is beneficial due to providing consistent flows whenever the GT-MAB is flying. Utilizing the GT-MAB and the Dyson fan, we establish an indoor test bed in the confined space of indoor environments, which is shown in Fig 5.4.

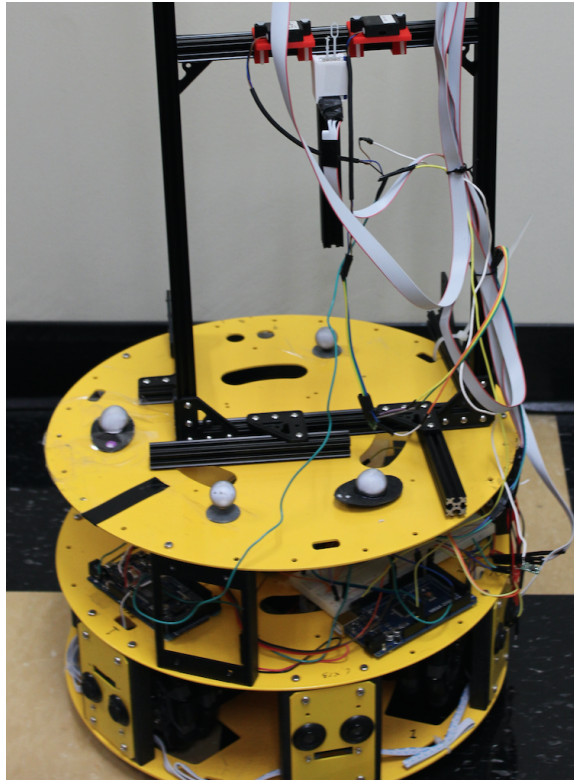


**Figure 5.4:** Indoor test bed: The yellow bulbs represent infrared motion capture cameras. The blue square represents the Dyson fan. The star represents the starting point of the GT-MAB. The red line represents the trajectory of the GT-MAB. When the GT-MAB is flying at the starting point, the GT-MAB motion is disturbed by flow generated from the Dyson fan. Then, the motion capture cameras collect the attitudes and trajectory of the GT-MAB.

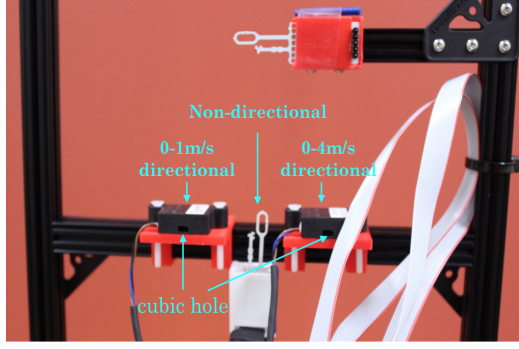
### 5.2.1 Georgia Tech wind measuring robot (GT-WMR)

On the indoor test bed, measuring flow generated from the Dyson fan is necessary for algorithm verification. Because the GT-MAB is able to fly in any position in the volume of the indoor test bed, we obtain flow estimates produced by the proposed algorithms from anywhere. To verify the flow estimates, we use wind field measurements of the test bed as ground truth. Although the Dyson manufacturer tells us that the maximum wind speed of Dyson fans is around 3m/s, the value may be different because of product tolerance. The Dyson fan generates multiple wind streams in the horizontal plane. The maximum wind speed is valid inside the boundary of the wind streams; however, we have no knowledge on wind speed outside the boundary of the wind streams, where the GT-MAB can fly.

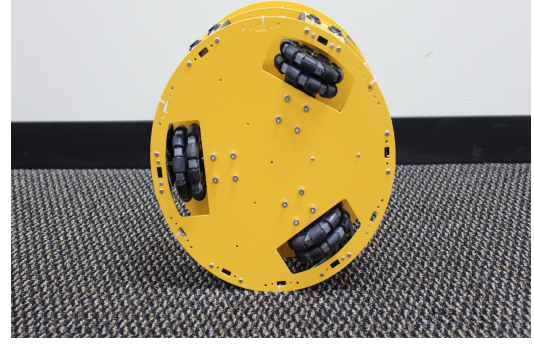
We develop a wind measuring robot (WMR) to measure the wind fields of the indoor environment. The GT-WMR (Georgia Tech Wind Measuring Robot) collects low wind speed measurements ranged from 0 to 4 m/s in all directions at predetermined locations, autonomously. The GT- WMR is shown in Figure 5.5. The GT-WMR integrates wind sensors on an omnidirectional robot. Figure 5.6 shows directional and non-directional wind sensors, and Figure 5.7 an omni-directional robot. In Figure 5.6 , the directional wind sensors measure wind speed which is perpendicular to the front side of the sensors. The directional sensors have one cubic hole that enables measuring wind speed when flow goes inside the hole. Each of the directional sensors has different range of measurements. One OMRON sensor provides wind speed between 0.2m/s and 1m/s with 0.01m/s accuracy (left sensor). The other OMRON sensor provides wind speed between 1m/s and 4m/s with 0.01m/s accuracy (right sensor). The non-directional sensor manufactured by Modern Device measures wind speed between 0 and 40m/s. Because the Modern Device sensor measures the wind speed with the change of temperature of the wire in the sensor, wind speed is measured regardless of direction. In Figure 5.7, the omnidirectional robot includes three wheels whose each rotational axis with respect to the omnidirectional robot's center has 120 degrees apart. It allows the omnidirectional robot to move in any direction.



**Figure 5.5:** GT-WMR: The GT-WMR contains two main components: an omnidirectional robot called omnibot and three wind sensors. The three wind sensors on an horizontal black frame are connected to the omnibot. The black frame can be moved vertically to measure wind speed at different heights. The omnibot has two Arduino board; one is an embedded Arduino board that receives wheel command from an ground station, and drive the wheels of the omnibot. The other is an additional Arduino board that collects wind measurements and send them to the ground station via the embedded Arduino board. The four gray spheres represent makers that the motion capture cameras recognize. The motion capture cameras collect attitudes and trajectories of the omnibot.



**Figure 5.6:** Two types of wind sensors: directional and non-directional



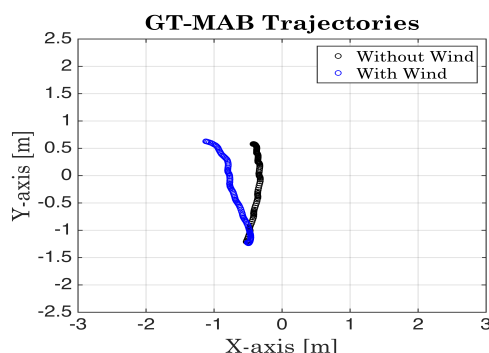
**Figure 5.7:** Three wheels of the omnidirectional robot

### 5.2.2 Adaptive learning algorithm

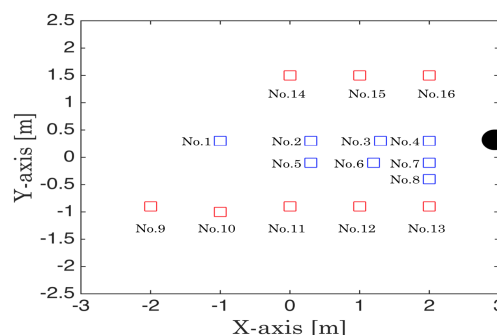
This section describes the experimental results applying the adaptive learning algorithm of Section 2.3. We have performed two steps for the verification of the adaptive learning algorithm. The first step is to identify air flow generated from the Dyson fan installed at certain position in the indoor test bed. At this stage, we measure the through-air speed from the derivative of the GT-MAB trajectory when there is no wind. We use the adaptive learning algorithm to identify air flow from the GT-MAB trajectory after we deploy the GT-MAB under air flow. The next step is to verify identified flow from ground truth. The GT-WMR autonomously moves along predetermined waypoints, and then the GT-WMR collects wind measurements at an altitude where wind sensors are fixed on the omnidirectional robot.

For the adaptive learning algorithm, we design four spatial basis functions composed of center  $c_i$ , width  $\sigma_i$ , where  $i = 1, 2, 3, 4$ .  $c_1, c_2, c_3$ , and  $c_4$  are  $[1.5594, 0.3]^\top$ ,  $[2.0594, 0.3]^\top$ ,  $[2.5594, 0.3]^\top$ , and  $[1.5594, -1.5]^\top$  respectively.  $\sigma_1, \sigma_2, \sigma_3$ , and  $\sigma_4$  are all equal to 1. Time-varying basis functions with harmonic frequencies and phases are removed because of consistent flow from the Dyson fan. Figure 5.8 shows the GT-MAB trajectories with and without the wind source. The black represents the GT-MAB trajectory without the wind source, and the blue with the wind source. Because the GT-MAB motion is disturbed by the wind source, the blue trajectory is bended over the black trajectory. With the black trajectory, we estimate through-air speed of the GT-MAB, which is 0.0185m/s. Multiple

starting points together with the location of wind source are shown in Figure 5.9. The black circle, which represents the Dyson fan, located in (2.94m, 0.3m). The eight blue squares represent starting points for forward or backward motion of the GT-MAB; the eight red squares represent starting points for diagonal motion toward the wind source or away the wind source.



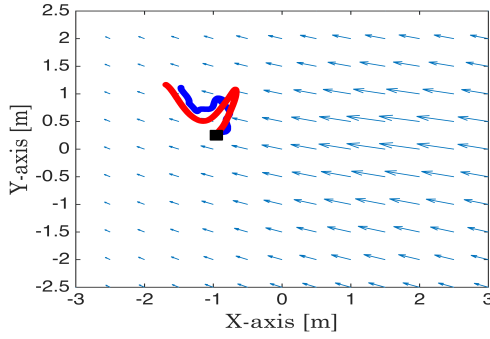
**Figure 5.8:** The GT-MAB trajectory without the wind source (the black) and with the wind source (the blue)



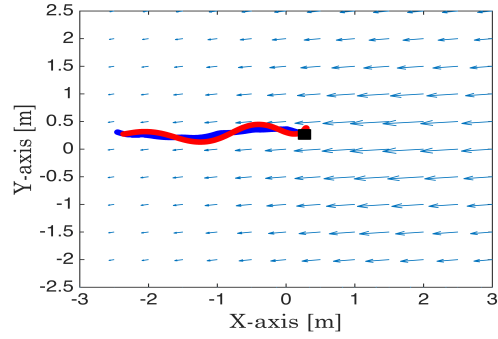
**Figure 5.9:** Location of the wind source (black circle), and multiple starting points for the GT-MAB deployment

Figures 5.10-5.17 show estimated and identified trajectories of the GT-MAB according to the blue starting points of Figure 5.9. At the individual starting point represented by one black square, we select one waypoint along the Y-axis for the forward or backward motion of the GT-MAB. The blue arrows, which represent wind identified by the adaptive learning algorithm, moves away from the wind source largely. The direction of identified flow corresponds to the wind direction generated from the wind source. Since the blue starting points are located inside the main stream of the Dyson's wind in Figure 5.9, most estimated trajectories are aligned with the main stream direction.

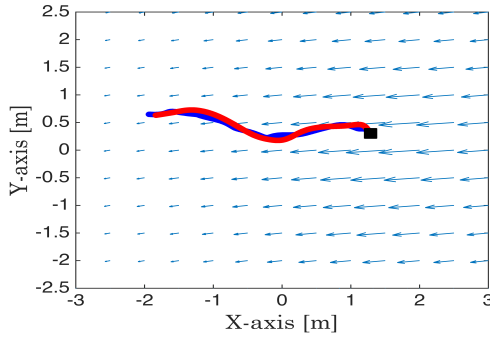
Figures 5.18-5.27 show estimated and identified trajectories of the GT-MAB according to the red starting points of Figure 5.9. Because wind strength at the red starting points is much smaller than at the blue starting points, one waypoint at individual red starting point is determined to make the GT-MAB moves diagonally away or toward the wind source; it enables the GT-MAB to identify spatially varying flow, covering the horizontal plane of the



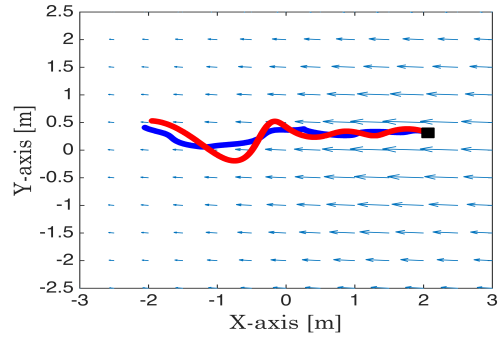
**Figure 5.10:** Estimated (blue) and identified (red) trajectories starting at No. 1 (black)



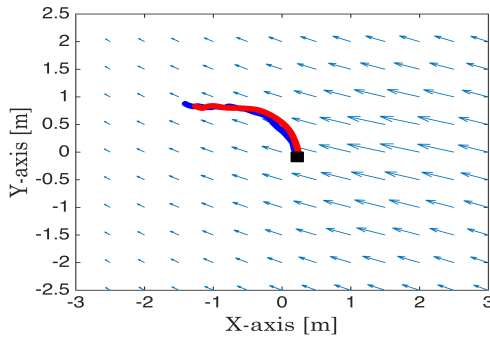
**Figure 5.11:** Estimated (blue) and identified (red) trajectories starting at No. 2 (black)



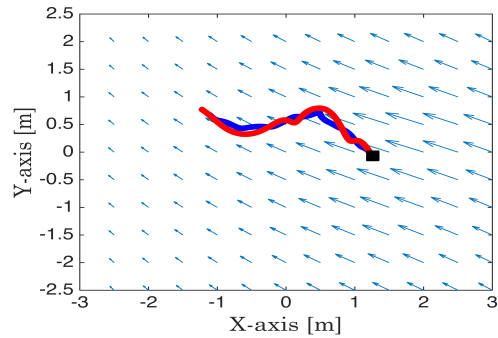
**Figure 5.12:** Estimated (blue) and identified (red) trajectories starting at No. 3 (black)



**Figure 5.13:** Estimated (blue) and identified (red) trajectories starting at No. 4 (black)

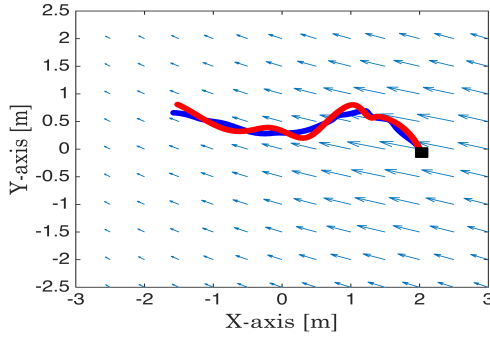


**Figure 5.14:** Estimated (blue) and identified (red) trajectories starting at No. 5 (black)

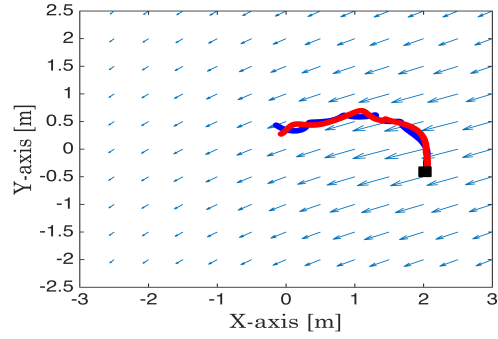


**Figure 5.15:** Estimated (blue) and identified (red) trajectories starting at No. 6 (black)

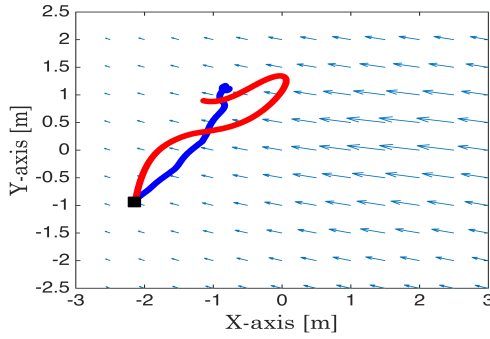
test bed. The magnitude of the blue arrows at the place closing to the wind source is larger than at other places; this tendency corresponds to the fact that strong wind is generated at positions close to the wind source.



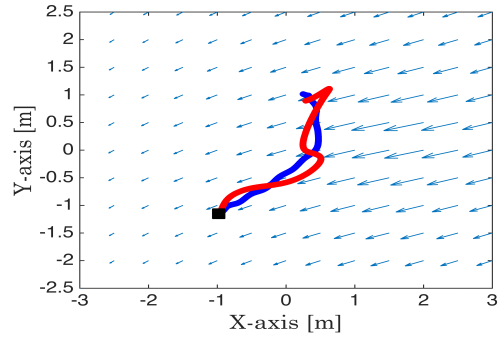
**Figure 5.16:** Estimated (blue) and identified (red) trajectories starting at No. 7 (black)



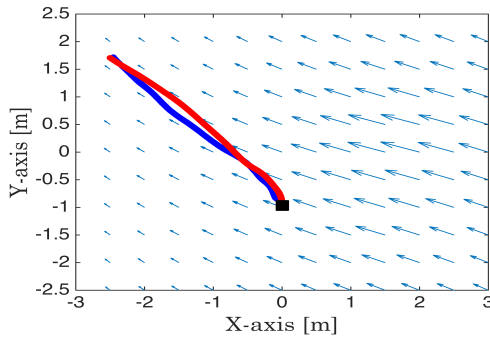
**Figure 5.17:** Estimated (blue) and identified (red) trajectories starting at No. 8 (black)



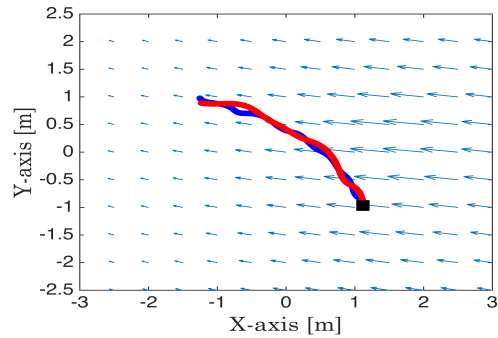
**Figure 5.18:** Estimated (blue) and identified (red) trajectories starting at No. 9 (black)



**Figure 5.19:** Estimated (blue) and identified (red) trajectories starting at No. 10 (black)



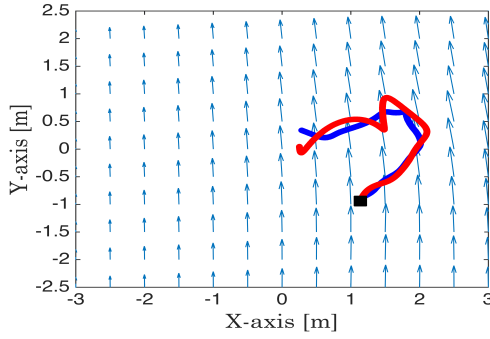
**Figure 5.20:** Estimated (blue) and identified (red) trajectories starting at No. 11 (black)



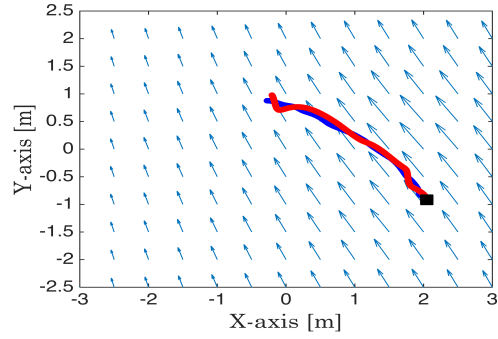
**Figure 5.21:** Estimated (blue) and identified (red) trajectories starting at No. 12 (black)

For the next step, we deploy the GT-WMR in the indoor test bed instead of the GT-MAB, maintaining the same wind environment of the GT-MAB experiments. To densely collect wind measurements in the indoor test bed, we determine multiple waypoints that

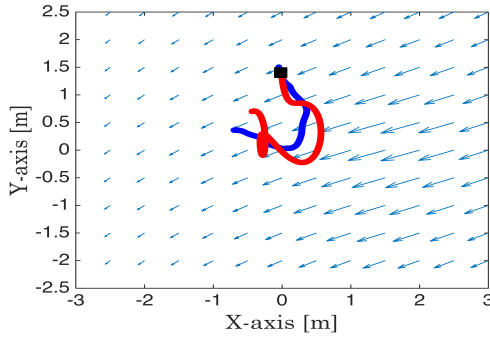




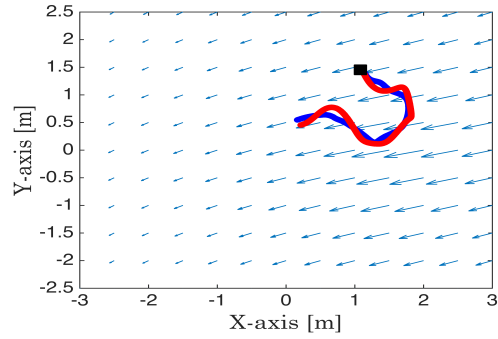
**Figure 5.22:** Estimated (blue) and identified (red) trajectories starting at No. 12 (black)



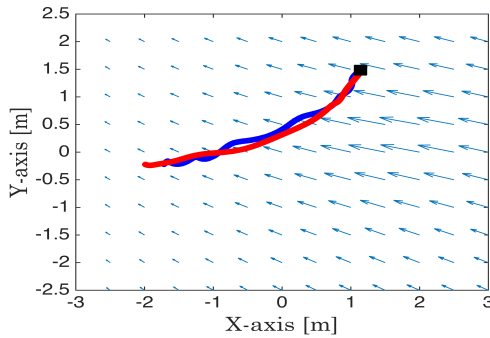
**Figure 5.23:** Estimated (blue) and identified (red) trajectories starting at No. 13 (black)



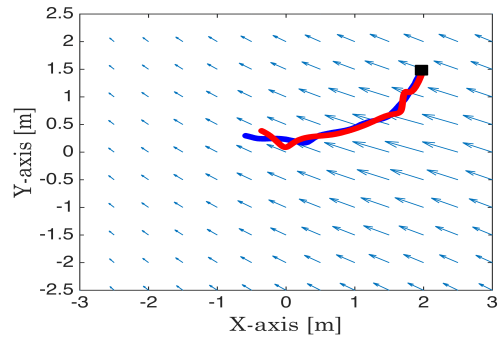
**Figure 5.24:** Estimated (blue) and identified (red) trajectories starting at No. 14 (black)



**Figure 5.25:** Estimated (blue) and identified (red) trajectories starting at No. 15 (black)

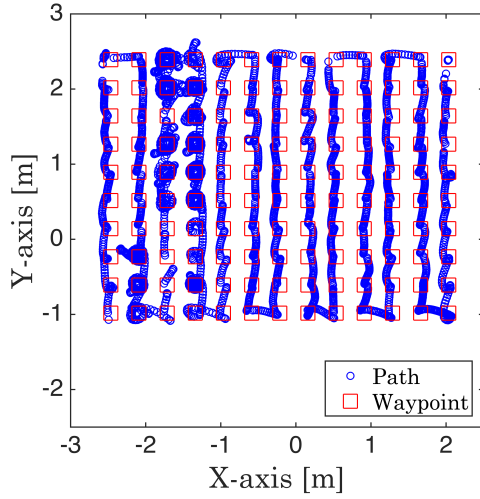


**Figure 5.26:** Estimated (blue) and identified (red) trajectories starting at No. 15 (black)

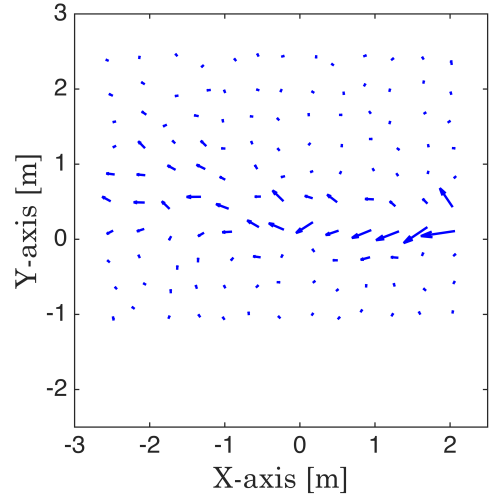


**Figure 5.27:** Estimated (blue) and identified (red) trajectories starting at No. 16 (black)

can generate the GT-WMR path having a type of lawn mower patterns. Figure 5.28 shows the waypoints and the GT-WMR's path, and Figure 5.29 shows measured wind velocity at each waypoints.



**Figure 5.28:** The path (blue) and waypoints (red) of the GT-WMR



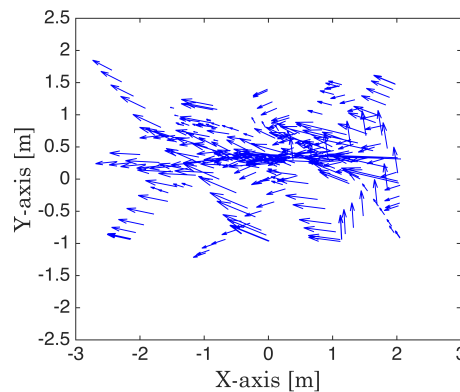
**Figure 5.29:** Measured flow velocity at each waypoint

When the GT-WMR controlled by a waypoint controller accurately reaches one waypoint shown in Figure 5.28, the GT-WMR stays at the waypoint until the wind speed sensors of Figure 5.6 measure wind speed along the orientation with 30 degrees increments starting from 0 to 360 degrees, and then the GT-WMR moves to the next waypoint and measure wind velocity at that waypoint; this collecting process is performed repeatedly for all the waypoints. At each waypoint, we assume that the wind speed sensors containing cubic holes of Figure 5.6 measure maximum wind speed when wind velocity vectors are perpendicular to the surface on the their cubic holes. Finding maximum value of wind speed measurements at each waypoint enable measuring wind velocity from the wind speed sensors. Measured wind of Figure 5.29 is valid in that the largest measurement of wind speed is acquired in the closest to the wind source. In addition, the width of strong wind streams is around 0.2m. This value is similar to the diameter of the Dyson fan, which is 0.254m. Outside of the width of strong wind streams, wind speed measurements is very small; it shows flow consistency of the Dyson fan. Measured wind direction is from the left to the right, which corresponds to wind blowing direction of the Dyson fan.

Verifying the adaptive learning algorithm is significantly difficult by simply comparing

Figures 5.10-5.27 with Figure 5.29. Although wind velocities are identified in the entire space of the test bed as shown Figures 5.10-5.27, the accuracy of the wind velocities is not consistent. Wind velocities identified along the GT-MAB trajectory is accurate; but, wind velocities identified at other positions except the positions that the trajectory occupy is not accurate. Because using constant heading angle command and constant spatial basis functions does not satisfy persistent excitation (PE) condition in equation (2.22), flow parameters for identified flow may be inaccurately identified. Therefore, identified flow may not be valid for the entire space of the indoor test bed, but for locations along the GT-MAB trajectory. Furthermore, measured flow velocities in Figure 5.29 rely on sensor accuracy. Instead of measurement comparison, we create two wind field maps based on identified and measured wind velocities, individually, The Kriging method, known as the optimal interpolation technique of spatial data, is used to make the wind field maps. Then, root-mean -square errors of wind speed and direction are computed to find the accuracy of the wind field map based on wind velocities identified from the adaptive learning algorithm with respect to the wind field map based on measured wind velocities.

Figure 5.30 and 5.31 shows identified wind velocities along the GT-MAB trajectories and the identified wind field map from the wind velocities.



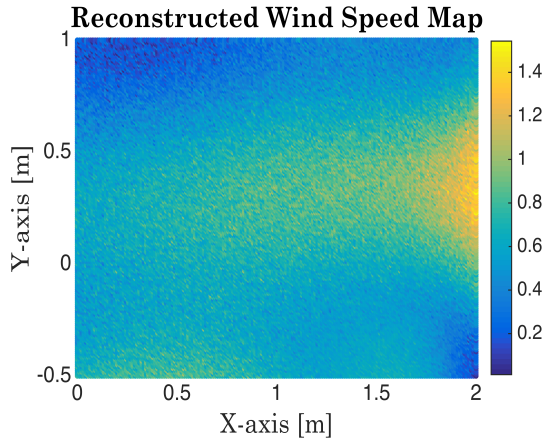
**Figure 5.30:** Identified wind velocities along the GT-MAB trajectories

Because dense data is necessary for wind field mapping, identified wind velocities within a subspace of Figure 5.30, ranging from 0 to 2 along X-axis and from -0.5 to 1

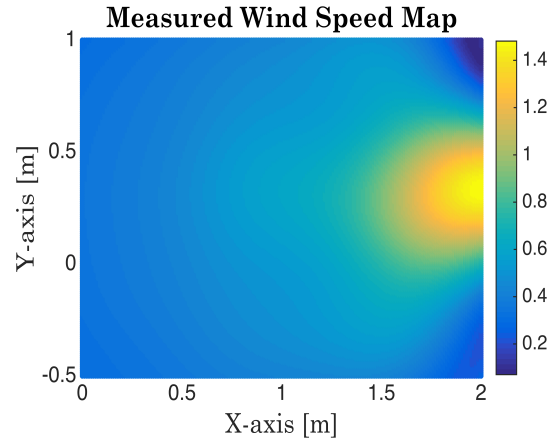
along Y-axis, are used to reconstruct a wind field map. The reconstructed wind speed map in the subspace is shown in Figure 5.31. Wind speed is the strongest at (2m, 0.3m), which is close to wind source located at (2.94m, 0.3m); wind is relatively stronger near 0.3m of Y-axis than other locations, which shows the main stream of wind blown by the Dyson fan. Figure 5.33, which represents the reconstructed wind direction map, consistently shows that wind directions are mostly 180 degrees. The direction is aligned with the X-axis negative direction. To verify both maps shown in Figures 5.31 and 5.33, we consider the measured wind velocity from the GT-WMR as ground truth. Although measured wind velocities are collected at 63 points evenly distributed in the square space where it is from -1.1m to 2.0m horizontally, and from -1.4m to 1.7m vertically, the number of data points are insufficient to make the measured wind map. For more dense data points, we utilize radial basis functions (RBF) with same centers and width that the adaptive learning algorithm use for flow identification; however, we do not use the same flow parameters that the adaptive learning algorithm identify. Given 63 data points, we identify flow parameters for the measured wind field map by using the least square algorithm. Identifying parameters enables making the measured wind field map. With dense data points generated from the RBF including the estimated flow parameters, the Kriging method makes the measured wind map; Figures 5.32 and 5.34 represent a wind speed map and a wind direction map constructed from measured wind velocity, respectively.

When we compare the reconstructed wind map of Figure 5.31 with the measured wind speed map of Figure 5.32, the range of wind speed is the same in both figures. The strongest wind of the figures is commonly identified at (2m, 0.3m) near the wind source. Furthermore, the main stream of blowing wind generated from the wind source is identified between 1.5m and 2m along the X-axis.

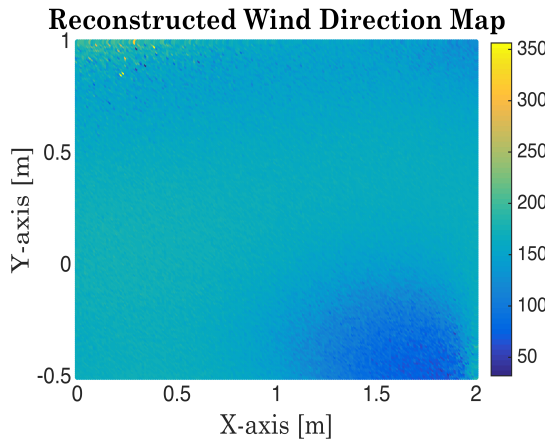
When we compare the reconstructed wind direction map of Figure 5.33 with the measured wind direction map of Figure 5.34, the range of wind direction is the same in both figures. Most wind directions in the square space are  $180^\circ$ , which corresponds to the



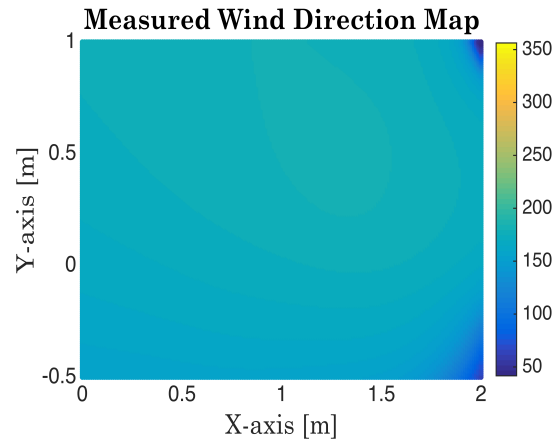
**Figure 5.31:** Reconstructed wind speed map using identified velocities of Figure 5.30



**Figure 5.32:** Measured wind speed map using measured velocities of Figure 5.29



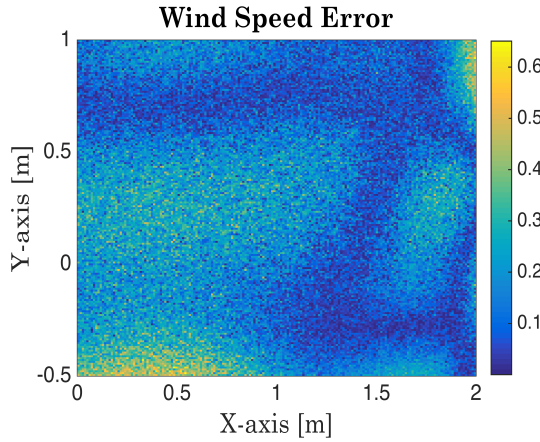
**Figure 5.33:** Reconstructed wind direction map using identified velocities of Figure 5.30



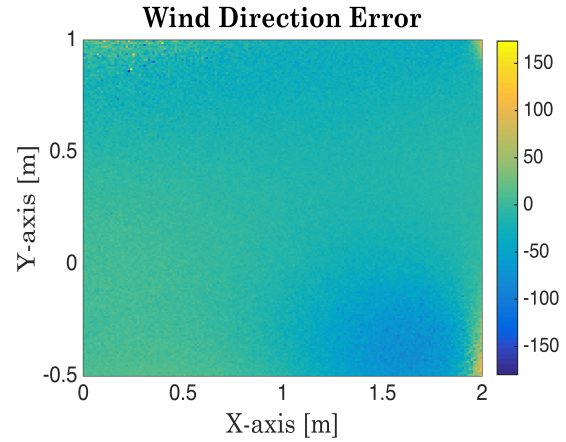
**Figure 5.34:** Measured wind direction map using measured velocities of Figure 5.29

direction away from the wind source.

In order to evaluate the accuracy of the reconstructed wind field map, we compute the wind speed error map and the wind direction error map, shown in Figures 5.35 and 5.36. The maximum wind speed error occurs in areas around (2,0.8) and (0.3,-0.5). Because the number of trajectories of the GT-MAB passing on the areas is much smaller than in the center of the space, the estimated flow is likely to be inaccurate. The root-mean-square



**Figure 5.35:** Error map of wind speed



**Figure 5.36:** Error map of wind direction

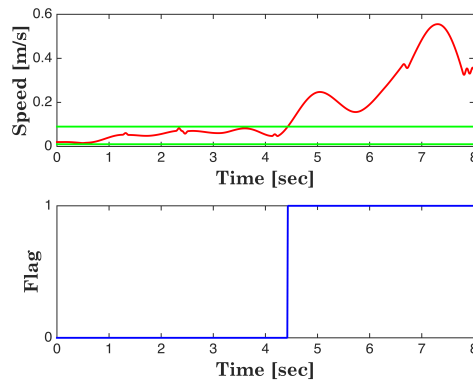
(RMS) error is a measure of the difference between estimated and measured data. Since we assume that the measured wind field map is ground truth, we compute RMS error of the reconstructed wind field map to see how the reconstructed wind field map is close to the measured wind field map. The RMS error of the reconstructed wind speed map is 0.16m/s; the value is around 11% of measured maximum wind speed. The RMS error of the reconstructed wind direction map is  $25.86^\circ$ ; the value is around 14% of measured maximum wind speed. Therefore, the reconstructed wind field map is reliable due to small RMS error, which is less than 15% of measured maximum values. Since the reconstructed wind field map uses air flow parameters identified by the adaptive learning algorithm of Section 3.3, we say that the adaptive learning algorithm of Section 3.3 is verified experimentally.

### 5.2.3 Anomaly detection algorithm

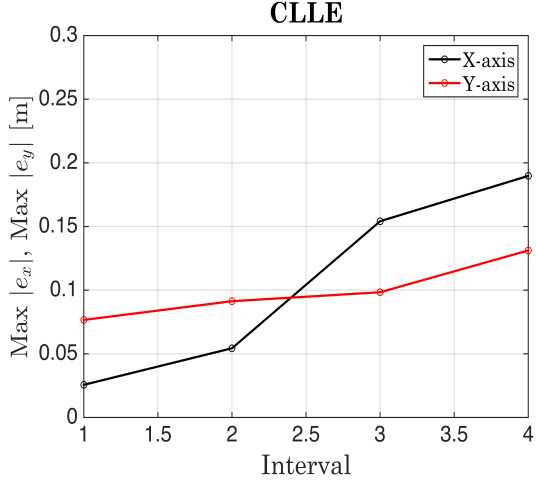
This section presents the experimental results associated with the anomaly detection algorithm. Intentionally making a fault of the GT-MAB is needed to verify whether the anomaly detection algorithm detects the fault. Although a variety of fault scenarios including software and hardware faults can be considered to verify the anomaly detection algorithm, we focus on a thruster fault, which is one of the most common faults to AUVs. Among the

five motors generating forces and moments for GT-MAB motions, we stop the fifth motor for a simulated fault during certain amounts of time. Because the fifth motor provides a force that controls the side-way motion of the GT-MAB, significant changes of a trajectory along side-way direction happens after the motor stops. The trajectory enables finding abnormal through-air speed from the anomaly detection algorithm. Our goal is to estimate through-air speed from the GT-MAB trajectory, and detect anomaly with estimated through-air speed.

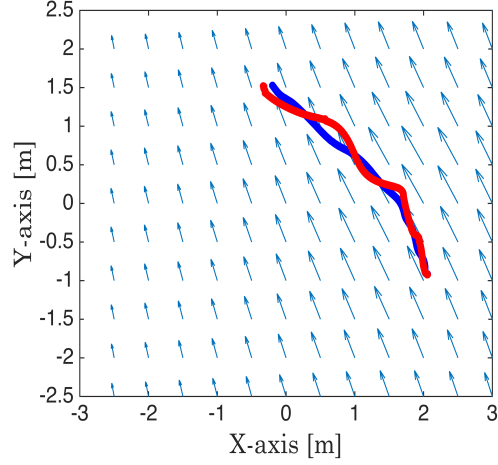
For the anomaly detection algorithm, we exploit the same four spatial basis functions composed of center  $c_i$ , width  $\sigma_i$ , where  $i = 1, 2, 3, 4$  as the four spatial basis functions in the adaptive learning algorithm of Section 5.2.2; but, we do not use the same flow parameters of the adaptive learning algorithm. The flow parameters are updated by the updating rules for the given trajectory. When we force the fifth motor to stop at 4 sec, Figure 5.37 shows through air speed (top) and flags (bottom). The normal speed range represented by two green lines has through-air speed estimate until 4.43 sec; after 4.43 sec, the through-air speed estimate escapes from the range. In this situation, flag change shows whether or not an anomaly is detected. Flag 0 maintains until 4.43 sec showing no anomaly detection. Flag 1 means that anomaly occurs because the through-air speed estimate is out of the range. Figures 5.38 and 5.39 show CLLE and the GT-MAB trajectory, respectively.



**Figure 5.37:** Identified wind velocities along the GT-MAB trajectories



**Figure 5.38:** CLLE along the X-axis (black), and CLLE along the Y-axis (red)



**Figure 5.39:** Estimated (blue) and identified (red) trajectories with identified flow (black)

#### 5.2.4 Adaptive control algorithm

This section describes the experimental results associated with the adaptive control algorithm of Section 3.3.1. We know that the GT-MAB trajectory is not straight due to flow generated from the wind source, as shown in Figure 5.8. If the adaptive control algorithm cancels flow that disturbs the trajectory, the GT-MAB trajectory would be straight, which is the goal of the adaptive control algorithm. Because the adaptive control algorithm requires a predicted trajectory that the GT-MAB trajectory wants to follow, we measure the positions of the GT-MAB controlled by a waypoint controller for the predicted trajectory when there is no wind source. When we assign waypoint (0,2) to the GT-MAB, the waypoint controller in the GT-MAB makes the vehicle move forward till reaching a waypoint at (0,2). Then, we use the set of measured positions of the GT-MAB according to time as the predicted trajectory.

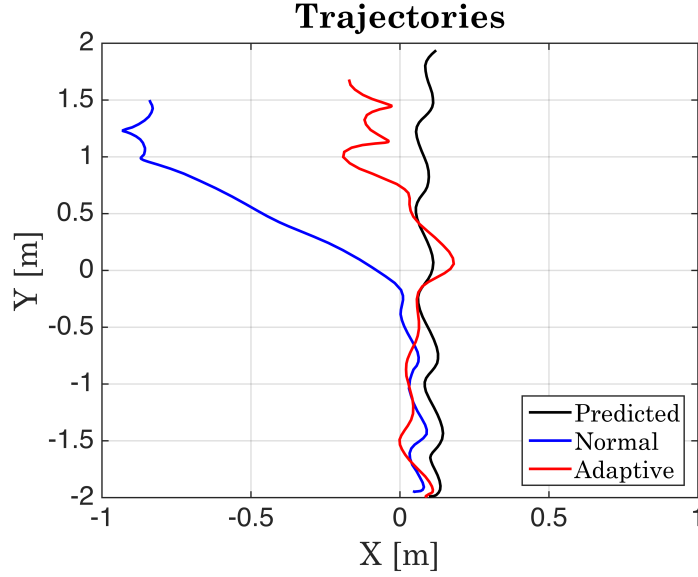
Applying the adaptive control algorithm to the GT-MAB is difficult because of the GT-MAB controller structure. Given one waypoint to the GT-MAB, the waypoint mission is performed by using two distance controllers. One distance controller that generates thrust command to forward thrusters reduces Y-axis error distance between current position and



waypoint along Y-axis. The other distance controller, which generates thrust command to the side-way thruster, decrease X-axis error distance between current position and waypoint along X-axis. Both controllers allow the GT-MAB to achieve the waypoint mission. However, because the adaptive control algorithm generates a velocity command to decrease error distances, we cannot connect the output of the adaptive control algorithm to the input of the distance controllers. To solve this problem, we convert the velocity command to a waypoint command. When we fix the Y-axis waypoint, we only compute the X-axis waypoint according to the direction of the velocity command. The reason why we fix the Y-axis waypoint is that we make the GT-MAB motion insensitive to the direction of the weak flow along the Y-axis. We know that flow along the Y-axis is very weak in Figure 5.29. If the Y-axis waypoint varies according to positive and negative direction of weak flow along the Y-axis, the GT-MAB is likely to move back and forth, and may not reach at the waypoint (0,2). Thus, we fix the Y-axis waypoint to only cancel negative direction of weak flow along the Y-axis; this method is valid when the direction of dominant flow is along the X-axis.

For the adaptive control algorithm, we utilize the same four spatial basis functions of the adaptive learning algorithm described in Section 5.2.2; but, we do not use the same flow parameters of the adaptive learning algorithm. The flow parameters are updated by the on-line adaptation laws represented by equations (3.45) and (3.46). Then, the adaptive controller described in equation (3.38) cancel the identified flow. Figures 5.40 shows the GT-MAB trajectory.

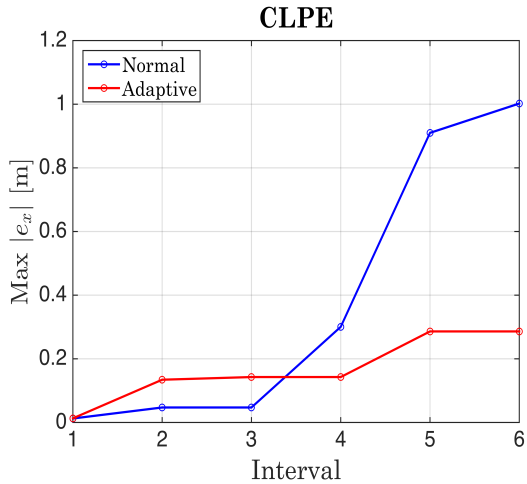
The black represents the predicted trajectory when the GT-MAB follows waypoint (0,2) without the wind source. The blue represents the GT-MAB trajectory when the GT-MAB follows the waypoint without the adaptive control algorithm under the wind source. The predicted trajectory is the straight line, but the GT-MAB trajectory curves in that flow disturbs the GT-MAB motion. The red represents the GT-MAB trajectory when the GT-MAB follows the waypoint with the adaptive control algorithm under the wind source.



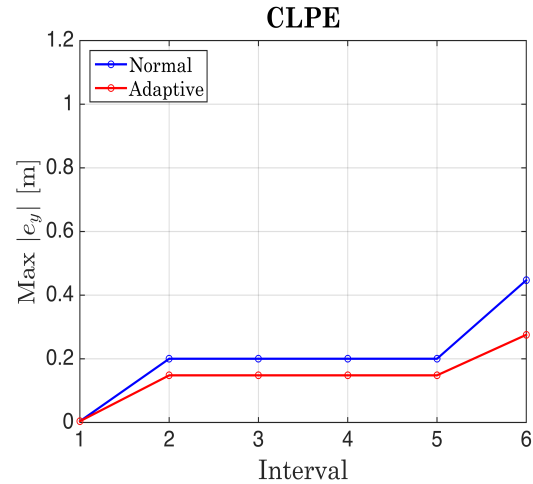
**Figure 5.40:** The adaptive control algorithm makes the GT-MAB trajectory (red) follow the predicted trajectory (black); without the adaptive control algorithm, the GT-MAB trajectory (blue) is away from the predicted trajectory.

It shows that the GT-MAB trajectory follows the predicted trajectory by canceling flow.

Figures 5.41 and 5.42 show CLPE along the X-axis and the Y-axis, respectively.



**Figure 5.41:** CLPE along the X-axis keeps increasing until 1m without the adaptive control algorithm (blue); but, CLPE along the X-axis is bounded by 0.3m with the adaptive control algorithm (red).



**Figure 5.42:** The similar trend of the increasing rate is shown in both CLPE along the Y-axis without the adaptive control algorithm (blue) and CLPE along the Y-axis with the adaptive control algorithm (red).

In Figures 5.41 and 5.42, the blue represents CLPE without the adaptive control al-

gorithm, and the red with the adaptive control algorithm. The adaptive control algorithm significantly makes CLPE reduction along the X-axis in Figure 5.41; comparing the blue and the red, the adaptive control algorithm decreases CLPE by 70 % with respect to non-adaptive control algorithm. For CLPE along the Y-axis, both cases have similar trend of increasing CLPE.

## CHAPTER 6

### CONCLUSION AND FUTURE WORK

The main contribution of this dissertation is a set of algorithms that detect anomaly of autonomous underwater vehicles (AUVs) without sensors monitoring vehicle components. Only using trajectory information, the proposed strategy detects abnormal vehicle motion under unknown ocean flow. It has the potential for mitigating abnormal vehicle motion with path-planning and controller design of AUVs. The experimental results of the Georgia Tech Miniature Autonomous Blimp (GT-MAB) and Georgia Tech Wind Measuring Robot (GT WMR) in an indoor test bed verify the proposed strategy. The summary of the contributions are described as follows.

- *Estimating through-water speed and ambient flow from trajectory information:* Given an estimated vehicle trajectory obtained after AUV deployment, The on-line adaptive learning algorithm simultaneously estimates through-water speed and flow velocity, guaranteeing convergence and robustness.
- *Detecting anomaly using estimated through-water speed and flow velocity:* Suppose we know the range of through-water speed when AUVs work normally. The anomaly detection algorithm determines that vehicle motion is abnormal when the through-water speed estimate is out of the normal speed range; if the through-water speed estimates is within the normal speed range, the algorithm determines that vehicle motion is normal. False alarms can happen due to estimation error of through-water speed during a transient period. Because we have knowledge on flow from flow models, comparing modeled flow velocity and estimated flow velocity enables avoiding the false alarms. Moreover, incorporating an adaptive flow canceling controller into AUVs mitigates false alarms induced by error of the estimated vehicle trajectory.

- *Controlling through-water speed with feedback and feedforward adaptive controller:*

When through-water speed can be controlled by feedback and feedforward controllers, the adaptive learning and control algorithms are developed to estimate through-water speed and flow velocity at the same time, guaranteeing convergence and robustness.

- *Estimating vehicle trajectory from underwater acoustic localization:* An acoustic localization algorithm of AUVs that use acoustic receivers primarily used for monitoring tagged fish is developed to estimate vehicle trajectory. The accuracy of estimated trajectory relies on the detection ranges of the receivers. Because the detection ranges vary due to environmental factors, we identify certain environmental factor to improve the accuracy of the estimated trajectory.

Future work will incorporate the estimated trajectory from acoustic localization into the adaptive learning algorithm and the anomaly detection algorithm. After modeling detection range that reflects tides and water stratification, the acoustic localization algorithm estimate the vehicle trajectory. The estimated trajectory is used for the adaptive learning algorithm that estimates through-water speed. Then the anomaly detection algorithm determine the abnormal vehicle motion with through-water estimates.

# **Appendices**

## APPENDIX A

### PUBLICATIONS

[1] Sungjin Cho and Fumin Zhang, “Localization of Autonomous Underwater Vehicles Incorporating Flow Models and Acoustic Detection,” in *Proc. of the 10th ACM International Conference on Underwater Networks and Systems*, Article No. 34, Washington DC, USA, October 22-24, 2015.

[2] Sungjin Cho, Fumin Zhang, and Catherine Edwards, “Adaptive Learning for Controlled Lagrangian Particle Tracking,” *Oceans 2016 MTS/IEEE*, pp. 1-6, Montrey, USA, September 19-22, 2016.

[3] Sungjin Cho, Fumin Zhang, and Catherine Edwards, “Tidal Variability of Acoustic Detection,” in *Proc. of the 2016 IEEE International Conference on Sustainable Computing and Communications*, pp. 431-436, Atlanta, USA, October 8-10, 2016 (**Invited**).

[4] Sungjin Cho and Fumin Zhang, “An Adaptive Control Law for Controlled Lagrangian Particle Tracking,” in *Proc. of the Eleventh ACM International Conference on Underwater Networks and Systems*, Article No. 11, Shanghai, China, October 24-26, 2016 (**Runner-up Best Student Paper Award**).

[5] Sungjin Cho, Vivek Mishra, Qiuyang Tao, Paul Varnell, Matt King-Smith, Aneri Muni, Weston Smallwood, and Fumin Zhang, “Autopilot Design for A Class of Miniature Autonomous Blimps,” *2017 IEEE International Conference on Control Technology and Applications (CCTA)*, pp. 841-846, Hawaii, USA, August 27-30, 2017 (**Invited**).

[6] Sungjin Cho, Fumin Zhang, and Catherine Edwards, “Anomaly Detection for Controlled Lagrangian Particles,” *Oceans 2017 MTS/IEEE*, Anchorage, USA, September 18-21, 2017.

[7] Sungjin Cho, Fumin Zhang, and Catherine Edwards, “Detecting Abnormal Speed of Marine Robots using Controlled Lagrangian Particle Tracking Methods,” in *Proc. of the 12th ACM International Conference on Underwater Networks and Systems*, Halifax, Canada, November 6-8, 2017.

[8] Ningshi Yao, Emily Anaya, Qiuyang Tao, Sungjin Cho, Hongrui Zheng, and Fumin Zhang, “Monocular vision-based human following on miniature robotic blimp,” *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3244-3249, Singapore, May 29-June 3, 2017.



## APPENDIX B

### AUTOPILOT DESIGN FOR GT-MAB

#### B.1 Dynamics and Control

Blimps have dynamics that are significantly different from aerial vehicles such as quadrotors and small airplanes. The lighter than air gas (Helium for GT-MAB) creates a fixed amount of lift. Blimps are subjected to significant fluid dynamic influences that are similar to the dynamics of underwater vehicles [83],[84], [85], [78]. In order to design controllers for the GT-MAB, the six degree-of-freedom (6-DOF) dynamics model for the blimp is first derived. We then simplify the model based on the goal to achieve two types of stable motion: the longitudinal translation motion and the spinning motion. We apply system identification techniques to compute unknown parameters in these models. The simplified models also allow us to design simple PID controllers for stable flight.

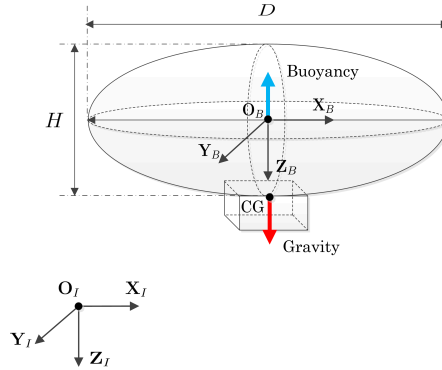
#### B.2 Dynamics with 6-DOF

The position of the blimp is determined in a world-fixed inertial coordinate frame. But it is more convenient to represent the linear and angular velocities in a body-fixed coordinate frame; see Fig. B.1. The control commands are supplied to the on-board thrusters which apply thrust in the body-fixed frame, but the position and orientation of the blimp are described with respect to the inertial coordinate frame. We follow the established procedures in the literature [85, 78] to derive the dynamic modeling equations.

Let Euler angles  $\phi$ ,  $\theta$ ,  $\psi$  be the roll, pitch, and yaw angles respectively and  $\eta^B = [u, v, w]^\top$ ,  $\eta^I = [\dot{x}, \dot{y}, \dot{z}]^\top$  be the linear velocity vectors in the body-fixed frame and the inertial frame respectively. The linear velocity vector can be transformed to the body-fixed frame from the inertial frame using  $\eta^B = C_I^B \eta^I$ , where  $C_I^B = C_\phi C_\theta C_\psi$ , and  $C_\phi, C_\theta, C_\psi$

are the rotation matrices corresponding to roll, pitch and yaw respectively where  $C_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}$ ,  $C_\theta = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}$ , and  $C_\psi = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$ . Let  $\boldsymbol{\omega} = [p, q, r]^\top$  be the angular velocity of the blimp in the body-fixed frame such that  $\boldsymbol{\omega} = J[\dot{\phi}, \dot{\theta}, \dot{\psi}]^\top$  where

$$J = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix}.$$



**Figure B.1:** Inertial and body coordinate frames.  $O_B - X_B Y_B Z_B$  represents the body-fixed axis of the blimp.  $O_I - X_I Y_I Z_I$  represents the inertial axis of the blimp. The origin  $O_B$  is the center of the buoyancy (CB) of the blimp. The center of buoyancy (CB) is the same as the center of volume of the balloon. The height  $H$  of the blimp is 0.46 meters and the diameter  $D$  is 0.72 meters.

Let  $\mathbf{f} = [f_x, 0, f_z]^\top$  represents the translational forces generated by the propellers of the blimp. There is no sideway force in the body-fixed frame. Let  $\boldsymbol{\tau} = [0, 0, \tau_z]$  be the turning torque generated by the propellers that control the yaw moment only. Let  $\mathbf{F} = [F_x, F_y, F_z]^\top$  be a vector that represents the gravity forces, the buoyancy forces, and other aerodynamic forces acting on the blimp in the body-fixed frame. Let  $\mathbf{M} = [M_x, M_y, M_z]^\top$  be a vector that represents all the external moments exerted on the blimp except for those generated by the

propellers. Let  $m$  be the mass of the blimp and

$$\mathbf{I} = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix}$$

be the moment of inertia about the origin of the body-fixed frame. Then the dynamic equations of motion for the blimp can be derived as

$$m(\dot{\boldsymbol{\eta}}^B + \boldsymbol{\omega} \times \boldsymbol{\eta}^B) = \mathbf{F} + \mathbf{f} \quad (\text{B.1})$$

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) = \mathbf{M} + \boldsymbol{\tau}. \quad (\text{B.2})$$

### B.3 Motion Primitives

The 6-DOF model described by equations (B.1) and (B.2) is nonlinear and coupled. A general controller design for such system is difficult, especially when some of the parameters of the model are unknown. Therefore, we need to find simplified models to achieve controllable flight. The GT-MAB is designed mainly to achieve three *motion primitives*:

1. Maintaining speed. The blimp should be able to maintain a desired constant speed along its  $X_B$  direction while having zero vertical speed, and zero yaw angular speed e.g.,  $u = u_0$ ,  $w = 0$ , and  $r = 0$ .
2. Changing altitude. The blimp should be able to ascend or descend to a desired height while maintaining zero forward speed and zero yaw angular speed e.g.  $z = z_0$ ,  $u = 0$  and  $r = 0$ .
3. Changing orientation. The blimp should be able to spin in place so that its yaw angle can be stabilized at any desired value while maintaining zero forward speed, and zero vertical speed e.g.  $\psi = \psi_0$ ,  $u = 0$  and  $w = 0$ .

These motion primitives can be combined to achieve stable flights that allow the blimp to move in 3D space. The design goals for the autopilot are now reduced to stabilizing these three motion primitives.

In order to simplify the dynamics, we make the following practical assumption that holds for all three motion primitives.

**Assumption 1.** *The roll angle and the roll angular velocity ( $\phi$ ,  $p$ ), the pitch angle and the pitch angular velocity ( $\theta$ ,  $q$ ), and the side-slipping velocity ( $v$ ) are negligibly small during the transient phase of the flight and zero during the steady state flight.*

**Remark 1.** *The assumption is justified because the GT-MAB is under the influence from the restoring forces due to its bottom-heavy design. The restoring forces induced by gravity and buoyancy effectively damp out roll and pitch motion. Since the blimp is very light with a large envelop, the side-slipping velocity of the blimp will vanish quickly due to air drag when the blimp flies forward. We understand that a sideways force will be generated by the term  $\boldsymbol{\omega} \times \boldsymbol{\eta}^B$  while the blimp is spinning and flying forward at the same time. However, this term is viewed as a vanishing disturbance force that is damped out by air drag. The force can also be ignored under the assumption that the spinning speed is almost zero during forward flight, and the forward speed is almost zero during the spinning motion.*

We are then able to separate the longitudinal and the spinning motion from the full nonlinear motion model and obtain three dynamic equations

$$m\dot{u} = F_x + f_x \quad (\text{B.3})$$

$$m\dot{w} = F_z + f_z \quad (\text{B.4})$$

$$I_z \dot{r} = M_z + \tau_z. \quad (\text{B.5})$$

Equations (B.3) and (B.4) describe the motion of the blimp in the  $X_B - Z_B$  plane, assuming no sideways motion and ignoring the sideways forces. Equation (B.5) represents the spinning motion of the blimp around the  $Z_B$  axis.

**Remark 2.** *Due to the symmetry of the blimp envelop, the inertia matrix  $\mathbf{I}$  becomes diagonal. One advantage of the “saucer-shaped” envelop is that  $I_x = I_y$ , which further simplifies the dynamics. The term  $\boldsymbol{\omega} \times \mathbf{I} \boldsymbol{\omega}$  does not generate rotation moments in equation (B.5) if the roll and pitch angular velocities are zero. In the case when roll and pitch moments are not zero, the contribution from the term  $\boldsymbol{\omega} \times \mathbf{I} \boldsymbol{\omega}$  to the roll and pitch moments are also small allowing them to be easily damped out by the restoring force from the gravity.*

**Remark 3.** *Note that the models (B.3-B.5) are still nonlinear because the external forces  $F_x, F_z$  and the external moment  $M_z$  include the forces and moments generated by the gravity and the ambient air. These forces are nonlinear functions of accelerations and velocities that are quite difficult to model. This is quite different from quad-rotors and multi-copters with powerful thrusters that generate forces  $f_x, f_z$  and moment  $M_z$  at least a magnitude larger than the influences from ambient air, which allow  $F_x, F_z$  and  $M_z$  to be ignored. For the blimp, because of the relatively large envelop and the relatively weak thrusters, the aerodynamics need to be considered, which is very similar to the modeling of underwater vehicles.*

## B.4 System Identification

The simplified models can be further linearized for each motion primitive. For flight at constant heading, we linearize the model around a desired forward speed, zero vertical speed, and zero yaw angular speed. For a change of altitude, we linearize the model around the desired height, zero forward speed and zero yaw angular speed. For a change of orientation, we linearize the model around a desired yaw angle, zero forward speed, and zero vertical speed. These models can be viewed as open-loop plants. The model for maintaining forward speed has  $f_x$  as its input and the forward speed  $u$  as its output. The model for changing altitude has  $f_z$  as its input and the  $z$  as its output. Since  $z$  is positive downward, the height of the blimp is negative  $z$ . And the model for changing orientation has the torque  $\tau_z$  as its input and the yaw angle  $\psi$  as its output. A set of experiments have been performed

where both input and output of the models are measured. Then the models can be identified using the MATLAB system identification toolbox.

The identified transfer functions are:

$$P_1(s) = \frac{0.9624s^2 + 0.5787s + 12.26}{s^3 + 0.7634s^2 + 14.62s + 4.691} e^{-0.17s} \quad (\text{B.6})$$

$$P_2(s) = \begin{cases} \frac{1.9535}{s^2 + 0.1267s} e^{-0.17s} & \text{if } f_z \leq 0 \\ \frac{0.9346}{s^2 + 0.0172s} e^{-0.17s} & \text{if } f_z > 0 \end{cases} \quad (\text{B.7})$$

$$P_3(s) = \frac{10.9365}{s^2 + 0.1855s} e^{-0.17s}. \quad (\text{B.8})$$

We notice that these models are greater than second order, most likely because they incorporate the parasitic effects caused by the non-ideal values of the state variables that we previously ignored.

We use the first order Pade approximation to approximate the time delays in the transfer functions. The Pade approximation  $e^{-\tau s} = \frac{1}{\tau s + 1}$  [86] leads to approximated open loop transfer functions. The model for maintaining forward speed is approximated by

$$P_1(s) = \frac{5.6612s^2 + 3.4041s + 72.1176}{s^4 + 6.6471s^3 + 19.1118s^2 + 90.7059s + 27.5941}. \quad (\text{B.9})$$

The model for changing altitude is approximated by

$$P_2(s) = \begin{cases} \frac{11.4912}{s^3 + 6.0088s^2 + 0.7453s} & \text{if } f_z \leq 0 \\ \frac{5.4976}{s^3 + 5.9s^2 + 0.1012s} & \text{if } f_z > 0. \end{cases} \quad (\text{B.10})$$

And the model for changing orientation is approximated by

$$P_3(s) = \frac{64.3324}{s^3 + 6.0676s^2 + 1.0912s}. \quad (\text{B.11})$$

The locations of open loop poles and zeros of the transfer function  $P_1(s)$  are plotted in

blue in Fig. B.2. For transfer functions  $P_1(s)$  and  $P_3(s)$ , there are no poles on the right half of the complex plane, but there are poles on the imaginary axis or the origin. This implies that forward speed and orientation systems are marginally stable. The altitude transfer function  $P_2(s)$  has different pole characteristics depending on input. When input  $f_z$  is non-positive, one pole is located at the origin; however, double poles are located at the origin when input  $f_z$  is positive. It shows that the altitude system is marginally stable under non-positive input, and unstable under positive input. Therefore, the desired speed, height, and yaw angle can not be achieved by open loop control.

In particular, the forward speed model  $P_1(s)$  has one pair of complex conjugate poles very close to the imaginary axis, which are generated by the coupling between the forward motion and pitch oscillation that are ignored when deriving the theoretical model. These poles will lead to very slowly vanishing oscillatory modes in the pitch angle when the blimp flies forward.

The altitude model  $P_2(s)$  is changed according to input sign. It implies that altitude dynamics is changed due to propeller efficiency and the asymmetric blimp shape. When we compare numerators of transfer function  $P_2(s)$ , the slope of upward thrust ( $f_z \leq 0$ ) is two times larger than that of downward thrust ( $f_z > 0$ ). Since propeller blades are mainly designed for generating upward thrust, reversal rotation of propellers produces much lower downward thrust than upward thrust. Comparing pole positions of transfer function  $P_2(s)$  shows asymmetric motion vertically. When the GT-MAB goes up, the upward motion is marginally stable in that the gondola located at the bottom of the envelope plays a role in a stabilizer aerodynamically. However, when the GT-MAB goes down, the downward motion is unstable because of nothing on the top of the envelope. Thus, the upward and the downward motions are not symmetric.

## B.5 Speed and Heading Controller Designs

Based on the identified linear input-output speed and heading models, we design controllers to achieve two motion primitives: maintaining speed and changing orientation. The goal is to make the two closed-loop systems asymptotically stable, and to compensate for the oscillations in the speed system. The forward speed controller uses  $u$  as feedback and  $f_x$  as control input. The heading controller uses  $\psi$  as feedback and  $\tau_z$  as the control.  $\psi$  is measured by the 3D localization system.

The two controllers are designed as the PID controllers. The PID gains for the two controllers are tuned in MATLAB based on the open-loop transfer functions identified. Table B.1 shows the gains.

**Table B.1:** Speed and Yaw PID Controller Gains

Controllers	P	I	D
Speed	1.095	1.095	0
Yaw	0.1955	0	0.192

The two closed-loop transfer functions under the PID controllers are:

$$G_1(s) = \frac{6.1988s^3 + 9.9265s^2 + 82.6865s + 78.9594}{s^5 + 6.6459s^4 + 25.3141s^3 + 100.6424s^2 + 110.2788s + 78.9594} \quad (\text{B.12})$$

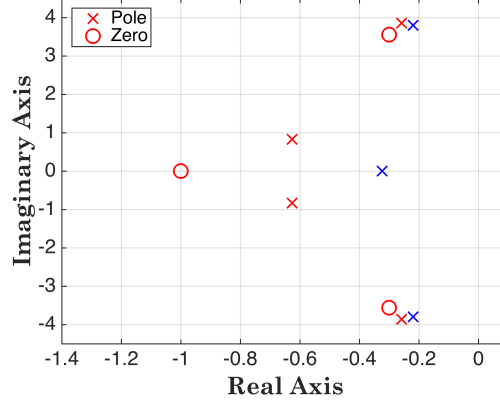
$$G_2(s) = \frac{12.5771}{s^3 + 6.0676s^2 + 13.4429s + 12.5771}, \quad (\text{B.13})$$

where  $G_1(s)$  is closed-loop transfer function of speed, and  $G_2(s)$  is closed-loop transfer function of yaw angle.

For two closed-loop transfer functions, all poles are on the left half plane, hence all the two closed-loop transfer functions are asymptotically stable. Furthermore, there is no steady-state errors under step function input. The poles and zeros of the closed-loop transfer function  $G_1(s)$  are plotted in red in Fig. B.2. The oscillatory complex poles in  $P_1(s)$  are compensated by a pair of complex conjugate zeros, which significantly reduce



the oscillation.



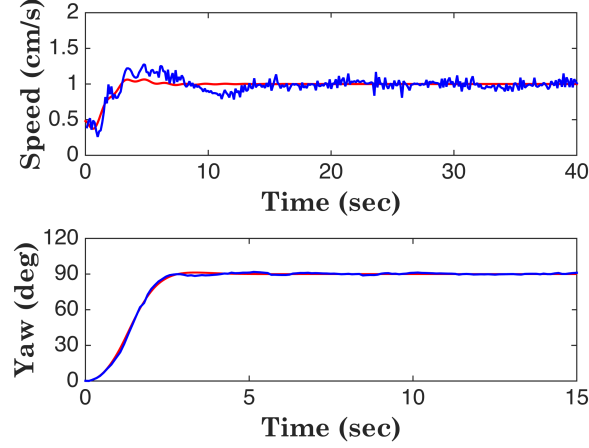
**Figure B.2:** Poles and zeros of the open loop transfer function  $P_1(s)$  and the closed-loop transfer function  $G_1(s)$  for forward speed control. Blue markers represent poles and zeros of the open-loop transfer function. Red markers represent poles and zeros of the closed-loop transfer function. One open-loop pole located at  $(-5.88, 0)$  and one closed-loop pole located at  $(-4.88, 0)$  are omitted in the figure.

Fig. B.3 shows the comparison of the simulated step response and the measured step response of the GT-MAB under each controller. It can be seen that the heading controller (bottom-graph) performs similarly to the simulated response and has a good rise and settling time. However, we still observe small oscillations in the speed controller (top graph). Due to lack of direct control input for the pitch motion, the linearized system is stabilizable but not controllable with respect to the pitch angle.

#### B.5.1 Altitude controller with a scheduling algorithm

We design an altitude controller to accomplish motion primitive for changing altitude. The altitude controller uses  $z$  as feedback and uses  $f_z$  as the control. Using one PID controller is not able to satisfy the stability and performance of altitude systems which changes according to the sign of  $f_z$ . We design a scheduling algorithm to switch between two set of PID gains.

The first PID gain is designed with the transfer function associated with upward thrust ( $f_z \leq 0$ ). The second PID gain is designed with the transfer function associated with down-



**Figure B.3:** Simulated step response (red) and measured step response (blue) for two motion primitives of the GT-MAB

ward thrust ( $f_z > 0$ ). The PID gains are tuned in MATLAB and shown in Table B.2.

**Table B.2:** Altitude PID Controller Gains

Controllers	P	I	D
# 1	0.6560	0.0087	0.7352
# 2	0.146	0.0027	1.955

We need to decide when to switch from one PID gain to the other PID gain, or vice versa. We design a scheduling algorithm for this purpose. Let  $e$  be altitude error with respect to desired altitude. Let flag represent switching modes. Let  $f_{z1}$  and  $f_{z2}$  represent output obtained by using #1 PID gain and #2 PID gain, respectively. Then, the scheduling algorithm is shown in Algorithm 3.

Note that positive  $f_z$  represents downward thrust; non-positive  $f_z$  represents upward thrust. For the scheduling algorithm, we select non-positive  $f_{z1}$  in that #1 PID gain is designed for upward thrust. In addition, we select positive  $f_{z2}$  because #2 PID gain is designed for downward thrust. We choose  $f_{z1}$  when both  $f_{z1}$  and  $f_{z2}$  are non-positive; i.e., both PID controllers generate upward thrust. We select  $f_{z2}$  when both  $f_{z1}$  and  $f_{z2}$  are positive. For two additional cases: when  $f_{z1}$  is non-positive and  $f_{z2}$  is positive, and when  $f_{z1}$  is positive and  $f_{z2}$  is non-positive, we are not able to choose either  $f_{z1}$  or  $f_{z2}$ . Then, we employ extra information to resolve this issue. The sign of altitude error allows us to

---

**Algorithm 3:** Scheduling Algorithm for Altitude Controller

---

**Input:** #1 PID controller output  $f_{z1}$ , #2 PID controller output  $f_{z2}$ , altitude error  $e$

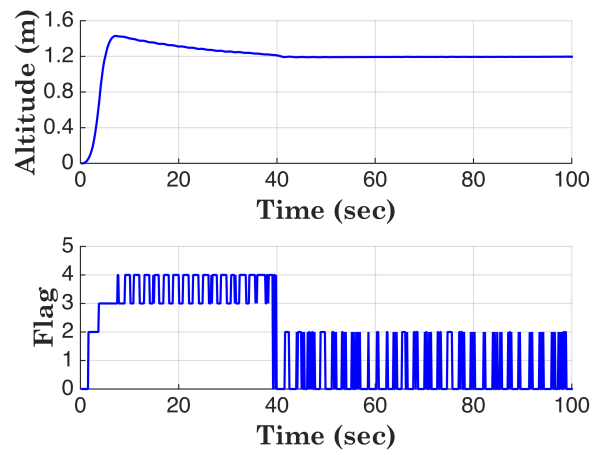
**Output:** Control input  $f_z$  and switching mode flag

```
1 if  $f_{z1} \leq 0$  &  $f_{z2} \leq 0$  then
2   |  $f_z = f_{z1}$ , flag = 0
3 else if  $f_{z1} \leq 0$  &  $f_{z2} > 0$  then
4   | if  $e \leq 0$ 
5     |  $f_z = f_{z2}$ , flag = 1
6     | else
7       |  $f_z = f_{z1}$ , flag = 2
8     | end
9 else if  $f_{z1} > 0$  &  $f_{z2} > 0$  then
10  |  $f_z = f_{z2}$ , flag = 3
11 else if  $f_{z1} > 0$  &  $f_{z2} \leq 0$  then
12  | if  $e \leq 0$ 
13    |  $f_z = f_{z2}$ , flag = 4
14    | else
15      |  $f_z = f_{z1}$ , flag = 5
16    | end
17 else
18  | break;
19 end
```

---

decide between  $f_{z1}$  and  $f_{z2}$ . Positive altitude error means current height is less than desired height. Non-positive altitude error means current height is greater than or equal to desired height. Therefore, #1 PID gain is used for positive altitude error, and #2 PID gain is used for non-positive altitude error at each case.

Fig. B.4 show measured step response of altitude (top) and corresponding time-series switching modes (bottom) while Algorithm 1 is running. Transient altitude response settles down after 40 secs by the altitude controller incorporating the scheduling algorithm. Furthermore, the proposed altitude controller has overshoot less than 20% and zero steady state error. In the bottom panel of Fig. B.4, flags 0 and 2 represent upward thrust that reduces positive altitude error before 4 secs and steady state error after 40 secs. Flags 3 and 4 represent downward thrust that reduces overshoot and transient altitude error.



**Figure B.4:** Measured step response for changing altitude (top) and switching modes (bottom)

## REFERENCES

- [1] M. J. Stanway, B. Kieft, and et. al., “White shark strike on a long-range AUV in Monterey bay,” in *Oceans 2015 IEEE/MTS*, 2015.
- [2] K. Szwaykowska and F. Zhang, “Trend and bounds for error growth in controlled Lagrangian particle tracking,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 10–25, 2014.
- [3] F. Zhang, G. Marani, R. N. Smith, and H. T. Choi, “Future trends in marine robotics,” *IEEE Robotics and Automation Magazine*, vol. 22, no. 1, pp. 14–21, pp. 122, 2015.
- [4] J. G. Bellingham and K. Rajan, “Robotics in remote and hostile environments,” *Science*, vol. 318, no. 1098-1022, 2007.
- [5] N. E. Leonard, D. A. Paley, and et. al., “Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey bay,” *Journal of Field Robotics*, vol. 27, no. 6, pp. 718–740, 2010.
- [6] S. Glenn, O. Schofield, J. Kohut, and et. al., “The trans-atlantic slocum glider expeditions: A catalyst for undergraduate participation in ocean science and technology,” *Marine Technology Society Journal*, vol. 45, no. 1, pp. 52–67, 2011.
- [7] L. Paull, S. Saeedi, M. Seto, and H. Li, “AUV navigation and localization: A review,” *IEEE Journal of Ocean Engineering*, vol. 39, no. 1, pp. 131–149, 2014.
- [8] F. Zhang, D. M. Fratantoni, and et. al., “Control of coordinated patterns for ocean sampling,” *International Journal of Control*, vol. 80, no. 7, pp. 1186–1199, 2007.
- [9] K. Szwaykowska, “Controlled lagrangian particle tracking: Analyzing the predicatability of trajectories of autonomous agents in ocean flows,” Ph. D. thesis, Georgia Institute of Technology, 2014.
- [10] S. S. SASTRY, “Model-reference adaptive control—stability, parameter convergence, and robustness,” vol. 1, no. 1, pp. 27–66, 1984.
- [11] K. J. Astrom, “Theory and applications of self-tuning regulators,” in *Control Theory, Numerical Methods and Computer Systems Modelling*, 1975, pp. 669–680.
- [12] R. Ortega and M. Spong, “Adaptive motion control of rigid robots: A tutorial,” in *Proceedings of the 27th IEEE Conference on Decision and Control*, 1988.

- [13] M. Spong and R. Ortega, "On adaptive inverse dynamics control of rigid robots," *IEEE Transactions on Automatic Control*, vol. 35, no. 1, pp. 92–95, 1990.
- [14] J. J. Craig, P. Hsu, and S. S. Sastry, "Adaptive control of mechanical manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 16–28, 1987.
- [15] D. Park, Z. Erickson, T. Bhattacharjee, and C. C. Kemp, "Multimodal execution monitoring for anomaly detection during robot manipulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 407–414.
- [16] B.-Y. Raanan, J. G. Bellingham, and et. al., "Automatic fault diagnosis for autonomous underwater vehicles using online topic models," in *Oceans 2016 MTS/IEEE*, 2016.
- [17] R. Isermann, "Model-based fault-detection and diagnosis – status and applications," *Annual Reviews in Control*, vol. 29, pp. 71–85, 2005.
- [18] J. J. Gertler, *Fault detection and diagnosis in engineering systems*. CRC Press, 1999.
- [19] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection : A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 15, 2009.
- [20] J. Chen and R. J. Patton, *Robust model-based fault diagnosis for dynamic systems*. Springer, 1998.
- [21] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 636–653, 2010.
- [22] G. Fagogenis, V. D. Carolis, and D. M. Lane, "Online fault detection and model adaptation for underwater vehicles in the case of thruster failures," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2625–2630.
- [23] Y. shan Sun, X. rui Ran, Y. ming Li, and et. al., "Thruster fault diagnosis method based on gaussian particle filter for autonomous underwater vehicles," *International Journal of Naval Architecture and Ocean Engineering*, vol. 8, pp. 243–251, 2016.
- [24] A. Caiti, F. D. Corato, and et. al., "Enhancing autonomy: Fault detection, identification and optimal reaction for over - actuated AUVs," in *Oceans 2015 IEEE/MTS*, 2015.
- [25] M. Caccia, R. Bono, and et. al., "Experiences on actuator fault detection, diagnosis and accomodation for ROVs," in *International symposium of unmanned untethered sub-mersible technology*, 2001.

- [26] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1114–1127, 2008.
- [27] P.-R. Lei, "A framework for anomaly detection in maritime trajectory behavior," *Knowledge and Information Systems*, vol. 47, pp. 189–214, 2016.
- [28] O. Rosén and A. Medvedev, "An on-line algorithm for anomaly detection in trajectory data," in *American Control Conference*, 2012, pp. 1117–1122.
- [29] P. Newman and J. J. Leonard, "Pure range-only sub-sea slam," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 1921–1926.
- [30] E. Olson, J. J. Leonard, and S. Teller, "Robust range-only beacon localization," *IEEE Journal of Ocean Engineering*, vol. 31, pp. 949–958, 2006.
- [31] S. Cho and F. Zhang, "Localization of autonomous underwater vehicles incorporating flow models and acoustic detection," in *Proceedings of the Tenth ACM International Conference on Underwater Networks and Systems (WUWNet '15)*, 2015.
- [32] S. T. Kessel, S. J. Cooke, M. R. Heupel, and et. al., "A review of detection range testing in aquatic passive acoustic telemetry studies," *Reviews in Fish Biology and Fisheries*, vol. 24, no. 1, pp. 199–218, 2014.
- [33] S. E. M. Munroe, C. A. Simpfendorfer, and M. R. Heupel, "Variation in blacktip shark movement patterns in a tropical coastal bay," *Environmental Biology of Fishes*, vol. 99, no. 4, pp. 377–389, 2016.
- [34] N. H. Mathies, M. B. Ogburn, and et. al., "Environmental interference factors affecting detection range in acoustic telemetry studies using fixed receiver arrays," *Marine Ecology Progress Series*, vol. 495, pp. 27–38, 2014.
- [35] R. Sakabe and J. M. Lyle, "The influence of tidal cycles and freshwater inflow on the distribution and movement of an estuarine resident fish *Acanthopagrus butcheri*," *Journal of Fish Biology*, vol. 77, pp. 643–660, 2010.
- [36] J. Yuh and M. West, "Underwater robotics," *Advanced Robotics*, vol. 15, no. 5, pp. 609–639, 2001.
- [37] J. Snyder, "Doppler velocity log (DVL) navigation for observation-class ROVs," in *Oceans 2010 MTS/IEEE*, 2010.
- [38] X. Liang, W. Wu, D. Chang, and F. Zhang, "Real-time modelling of tidal current for navigating underwater glider sensing networks," in *Procedia Computer Science*, vol. 10, 2012, pp. 1121–1126.

- [39] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, 1996.
- [40] H. K. Khalil, *Nonlinear Systems*, 2nd. Prentice Hall, 1996.
- [41] S. S. SASTRY and M. Bodson, *Adaptive Control*. Prentice Hall, 1989, pp. 27–66.
- [42] K. J. Astrom and B. Wittenmark, *Adaptive Control*, Second. Addison-Wesley, 1995.
- [43] H. K. Khalil, *Nonlinear Systems*, Third. Printice Hall, 2002.
- [44] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Prentice Hall, 1989.
- [45] S. P. Karason and A. M. Annaswamy, “Adaptive control in the presence of input constraints,” *IEEE Transactions on Automatic Control*, vol. 39, no. 11, pp. 2325–2330, 1994.
- [46] J. R. A. Luetlich, J. J. Westerink, and N. W. Scheffner, “ADCIRC: An advanced three-dimensional circulation model for shelves, coasts, and estuaries. Report 1. Theory and methodology of ADCIRC-2DDI and ADCIRC-3DLthree-dimensional circulation model for shelves, coasts, and estuaries,” Naval Research Lab, Tech. Rep. NRL/FR/7322–00-9962, 1992.
- [47] P. J. Martin, “Description of the navy coastal ocean model version 1.0,” Naval Research Lab, Tech. Rep. NRL/FR/7322–00-9962, 2000.
- [48] C. Wang, F. Zhang, and D. Schaefer, “Dynamic modeling of an autonomous underwater vehicle,” *Journal of Marine Science and Technology*, vol. 20, no. 2, pp. 199–212, 2015.
- [49] A. Spears, M. West, M. Meister, C. Walker, J. Buffo, T. Collins, A. M. Howard, and B. E. Schmidt, “Under ice in antarctica,” *IEEE Robotics and Automation Magazine*, vol. 23, no. 4, pp. 30–41, 2016.
- [50] D. Chang, W. Wu, C. R. Edwards, and F. Zhang, “Motion tomography: Mapping flow fields using autonomous underwater vehicles,” *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 320–336, 2017.
- [51] L. Yi and M. Tomizuka, “Two-degree-of-freedom control with robust feedback control for hard disk servo systems,” *IEEE/ASME TRANSACTIONS ON MECHATRONICS*, vol. 4, no. 1, pp. 17–24, 1999.
- [52] K. Selvam, S. Kanev, J. W. van Wingerden, T. van Engelen, and M. Verhaegen, “Feedback–feedforward individual pitch control for wind turbine load reduction,” *International Journal of Robust and Nonlinear Control*, vol. 19, pp. 72–91, 2009.



- [53] V. Corporation, *Receivers and transmitters for fish tags*, <http://vemco.com/products/>.
- [54] B. Blanton, F. Werner, H. Seim, R. Luettich, D. Lynch, K. Smith, G. Vougaris, F. Bingham, and F. Way, "Barotropic tides in the South Atlantic Bight," *JGR*, vol. 109, C12024, doi:10.1029/2004JC002455, 2004.
- [55] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, pp. 131–149, 2014.
- [56] P. Batista, C. Silvestre, and P. Oliveira, "Optimal position and velocity navigation filters for autonomous vehicles," *Automatica*, vol. 46, pp. 767–774, 2010.
- [57] C. M. Clark, C. Forney, E. Manii, D. Shinzaki, C. Gage, M. Farris, C. G. Lowe, and M. Moline, "Tracking and following a tagged leopard shark with an autonomous underwater vehicle," *Journal of Field Robotics*, vol. 30, pp. 309–322, 2013.
- [58] S. Oh and S. Sastry, "Tracking on a graph," in *Information processing in sensor networks*, 2005, pp. 1–8.
- [59] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri, "Target tracking with binary proximity sensors," *ACM Transactions on Sensor Networks*, vol. 5, 30:1–30:33, 2009.
- [60] P. Asadzadeh, L. Kulik, E. Tanin, and A. Wirth, "On optimal arrangements of binary sensors," *Spatial Information Theory 2011*, vol. 6899, pp. 168–187, 2011.
- [61] W. R. Geyer and R. Signell, "Measurements of tidal flow around a headland with a shipboard acoustic Doppler current profiler," *Journal of Geophysical Research*, vol. 95, no. 3, pp. 3189–3197, 1990.
- [62] J. Finn, J. Brownscombe, C. Haak, and et. al., "Applying network methods to acoustic telemetry data: Modeling the movements of tropical marine fishes," *Ecological Modelling*, vol. 293, pp. 139–149, 2014.
- [63] M. C. Melnychuk and C. J. Walters, "Estimating detection probabilities of tagged fish migrating past fixed receiver stations using only local information," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 67, no. 4, pp. 641–658, 2010.
- [64] K. Ø. Gjelland and R. D. Hedger, "Environmental influence on transmitter detection probability in biotelemetry: Developing a general model of acoustic transmission," *Methods in Ecology and Evolution*, vol. 4, pp. 665–674, 2013.
- [65] T. R. Binder, C. M. Holbrook, T. A. Hayden, and C. C. Krueger, "Spatial and temporal variation in positioning probability of acoustic telemetry arrays: Fine-scale

- variability and complex interactions,” *Animal Biotelemetry*, vol. 4, no. 4, pp. 1–15, 2016.
- [66] S. Cho, F. Zhang, and C. R. Edwards, “Tidal variability of acoustic detection,” in *Proceedings of the 2016 IEEE International Conference on Sustainable Computing and Communications*, 2016.
  - [67] R. Pawlowicz, B. Beardsley, and S. Lentz, “Classical tidal harmonic analysis including error estimates in matlab using T-TIDE,” *Computers and Geosciences*, vol. 28, no. 8, pp. 929–937, 2002.
  - [68] R. Soulsby, “Tidal-current boundary layers,” in *The Sea: Ocean engineering science*, B. Mehaute and D. Hanes, Eds., vol. 9, Wiley-Interscience, 1990, pp. 523–566.
  - [69] G. Weatherly, S. Blumsack, and A. Bird, “On the effect of diurnal tidal currents in determining the thickness of the turbulent Ekman bottom boundary layer,” *JPO*, vol. 10, pp. 297–300, 1980.
  - [70] R. H. Stewart, *Introduction To Physical Oceanography*. University Press of Florida, 2009.
  - [71] J. H. Simpson, J. Brown, J. Matthews, and G. Allen, “Tidal straining, density currents, and stirring in the control of estuarine stratification,” *Estuaries*, vol. 13, no. 2, pp. 125–132, 1990.
  - [72] S. Cho, V. Mishra, Q. Tao, P. Varnell, M. King-Smith, A. Muni, W. Smallwood, and F. Zhang, “Autopilot design for a class of miniature autonomous blimps,” in *2017 IEEE International Conference on Control Technology and Applications (CCTA)*, Hawaii, USA, 2017, pp. 841–846.
  - [73] A. Elfes, S. S. Bueno, M. Bergerman, and J. J. G. Ramos, “A semi-autonomous robotic airship for environmental monitoring missions,” in *Proceedings of the 1998 IEEE International Conference on Robotics and Automations*, 1998, pp. 614–619.
  - [74] G. Kantor, D. Wettergreen, J. P. Ostrowski, and S. Singh, “Collection of environmental data from an airship platform,” *Proceedings of SPIE*, vol. 4571, pp. 76–83, 2001.
  - [75] M. Burri, L. Gasser, M. Kach, S. Laube, and A. Ledergerber, “Design and control of a spherical omnidirectional blimp,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1873–1879.
  - [76] D. St-Onge, C. ment Gosselin, and N. Reeves, “Dynamic modelling and control of a cubic flying blimp using external motion capture,” *Journal of Systems and Control Engineering*, vol. 229, no. 10, pp. 970–982, 2015.

- [77] E. Paulos and J. Canny, “Prop: Personal roving presence,” in *CHI '98 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1998, pp. 296–303.
- [78] S. Zwann, A. Bernardino, and J. Santos-Victor, “Vision based station keeping and docking for an aerial blimp,” in *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000, pp. 614–619.
- [79] J. Müller and W. Burgard, “Efficient probabilistic localization for autonomous indoor airships using sonar, air flow, and imu sensors,” *Advanced Robotics*, vol. 27, no. January, pp. 711–724, 2013.
- [80] J.-C. Zufferey, A. Guanella, A. Beyeler, and D. Floreano, “Flying over the reality gap: From simulated to real indoor airships,” *Autonomous Robot*, vol. 21, pp. 243–254, 2006.
- [81] H. Zhang and J. P. Ostrowski, “Visual servoing with dynamics: Control of an unmanned blimp,” in *Proceedings of the 1999 IEEE/RSJ International Conference on Robotics and Automation*, 1999, pp. 618–623.
- [82] P. Gonzalez, W. Burgard, R. Sanz, and J. L. Fernandez, “Developing a low-cost autonomous indoor blimp,” *Journal of Physical Agents*, vol. 3, pp. 43–52, 2009.
- [83] T. I. Fossen, *Guidance and Control of Ocean Vehicles*. Wiley, 1993.
- [84] C. Wang, F. Zhang, and D. Schaefer, “Dynamic modeling of an autonomous underwater vehicle,” *Springer Journal of Marine Science and Technology*, vol. 20, pp. 199–212, 2014.
- [85] S. B. V. Gomes and J. J. G. Ramos, “Airship dynamic modeling for autonomous operation,” in *Proceedings of 1998 IEEE International Conference on Robotics and Automation (ICRA)*, 1998, pp. 3462–3467.
- [86] N. S. Nise, *Control Systems Engineering*. Addison-Wesley, 1995.