Evolutionary Path Planner with Shell Space Decomposition for Autonomous Underwater Vehicles in Ocean Environments

by

Zheng Zeng, *B.Eng. (Electrical Engineering),*
School of Computer Science, Engineering and Mathematics,
Faculty of Science and Engineering

Dec 2014

A thesis presented to the
Flinders University of South Australia
in total fulfilment of the requirements for the degree of
Doctor of Philosophy (Engineering)

Contents

List of Figures

LIST OF FIGURES

List of Tables

Abstract

Autonomous underwater vehicles (AUVs) have been proposed for a large spectrum of applications, ranging from environmental monitoring, ocean floor mapping, search and rescue operations, tracking of multiple targets, surveillance and reconnaissance, etc. Many of these applications consume much time lasting from days to weeks, and cover large areas of hundreds to thousands of square kilometres. Oceanographic processes, such as currents show great variability over such large expanses and will evolve over such durations. In most existing applications, AUVs are typically deployed from surface vessels with support personnel for deployment, piloting and recovery. During the course of the mission, the support vessel will shadow the AUV and provide much of the higher level decision making processes needed to deal with changes of oceanographic processes. The cost of keeping the support vessel on standby is generally by far the most significant component of the mission cost. Recently, there has been growing interest in developing long range AUVs with increased autonomy to conduct science missions over longer periods without supervision, thereby reducing mission costs and extending their applicability.

Many recent efforts towards improving AUV persistence focus on improvements in battery technologies, on-board power management, reduced hotel load, drag, efficient propulsion, navigation and guidance systems, and overall system reliability. In particular, an effective and versatile path planning system is of crucial importance to the safe, successful and efficient completion of long range missions. Rather than depend on the support of a manned surface vehicle, an AUV could be launched from shore where upon a path planning system could be used to generate a trajectory that

exploits the ocean energy taking use of the favourable currents to propel the vehicle and lead the vehicle to a remote work site, perform a survey, and then return to shore completely on its own. This will greatly diminish the costs for AUV operations since there is no need for a manned support vessel and vehicle operations can be monitored on-shore, thereby enhancing the affordability of AUVs to science and industry. In order to facilitate extended range, the AUV may be used in combination with an autonomous surface vehicle (ASV). The ASV will provides the AUV with localization support, and the AUV will periodically rendezvous and dock with the ASV to recharge and upload data and download instructions.

This thesis proposes several path planning and re-planning techniques for applications involving either a single vehicle or teams of vehicles operating in a dynamic, cluttered, and uncertain ocean environment.

This thesis presents a methodology for formulating the AUV path planning problem in the context of the environmental constraints. The turbulent, cluttered and uncertain environments modelled here incorporate strong currents field, irregularly shaped terrains and obstacles, the position of which may be dynamic and uncertain. The B-Spline based quantum-behaved particle swarm optimization (QPSO) path planning technique, introduced in this thesis, combines the main advantages of previously published approaches. These include smooth curvature paths represented by the Spline to accommodate constraints imposed by the manoeuvrability of the vehicle; and a QPSO algorithm enables the path planner to obtain a more optimized trajectory than the A*, rapidly exploring random tree (RRT), the conventional genetic algorithm (GA) and particle swarm optimization (PSO) derived trajectories.

A new shell space decomposition (SSD) scheme is then proposed to increase the searching efficiency of the B-Spline based QPSO path planner. This scheme decomposes the search space into shell regions radiating out from the starting point to the destination, and one or more control points generating the Splines are placed within each of these regions. This arrangement gives more freedom to the placement of the control points, but still restricts the search space for each control point to its respective regions to save computation time. The SSD scheme has been integrated with a QPSO based path planner and tested to find an optimal trajectory for an AUV navigating through a variable ocean environment in the presence of obstacles.

ABSTRACT

Subsequently, the generic SSD scheme is extended to account for the case of AUV operating in a spatiotemporal ocean environment. A dynamic SSD strategy is developed and incorporated with the on-line planning system that adapts and regenerates the trajectory during the course of the mission using continuously updated current profiles from on-board sensors. The next part of this thesis introduces path planning problems involving multiple autonomous marine vehicles (AMVs). The focus of the work is the problem of organising simultaneous arrival for multiple AMVs in the presence of variable ocean currents, irregularly shaped terrains and dynamic obstacles. A distributed shell space decomposition (DSSD) scheme that directly derives from the SSD concept is developed. The proposed QPSO-DSSD path planner is integrated with a novel optimized mass-centre rendezvous point selection scheme to identify the optimal rendezvous position, along with an optimal operational speed scheme to improve the performance of simultaneousness arrival at the rendezvous point, and reduce power consumption to complete the mission.

This research accommodates the current and future needs of persistent presence of AUVs. The new autonomous planning techniques developed in this thesis contribute to improve the capability of the AUVs to have longer mission durations as well as higher levels of autonomy.

Nomenclature

V_c	currents velocity
u_c, v_c, w_c	currents velocity components about x, y, z-axis, respectively
$\mathbb{R}$	2-D spatial domain
$\mathbb{R}_o$	coordinates of the centre of the vortex
ℓ, ζ	parameters control the radius and strength of the vortex, respectively
K	centre of the vortex
C	collision value
T	time consumption
V_r	water-referenced speed of the vehicle in the geographical frame
V_a	linear velocity of origin of body axes
V_{ay}, V_{ax}, V_{az}	O-XYZ velocity components of the vehicle respectively
ψ_r, θ_r	vehicle yaw and pitch, respectively
F	evaluation function
P	initial group of particles
$p_i(t)$	i^{th} particle at iteration t
$\mathcal{P}$	particle's best state
G	particle swarm's best state
ρ	population size
m	number of control points
g	maximum number of iterations
s, θ, ϕ	log-polar coordinates
E	radii of the concentric circles
$\mathcal{B}$	bounded searching region
$\mathcal{B}s, \mathcal{B}\theta$	bounds for s and θ, respectively
Ls, Lθ	lower bounds for s and θ, respectively
Us, Uθ	upper bounds s and θ, respectively
u, v, and w	surge, sway, and heave linear motions, respectively.

NOMENCLATURE

p, q, and r,	roll, pitch, and yaw rotational rates, respectively.
η	vehicle position vector
$\mathcal{A}_0, \mathcal{A}_\mathcal{G}$	vehicle's initial and final condition, respectively.
$\mathcal{r}_0, \mathcal{r}_\mathcal{G}$	vehicle's initial and final position, respectively.
$\mathcal{w}_0, \mathcal{w}_\mathcal{G}$	vehicle's initial and final Euler angles, respectively.
ϕ, θ, ψ	roll, pitch and yaw, respectively.
$\mathcal{V}$	water referenced speed
$\mathbb{p}$	path
$\wp_i$	point on the path
O	obstacle
$\mathbb{T}$	terrain
O_P, O_V, O_U	obstacle's position, velocity and uncertainty, respectively
$\mathbb{G}$	Rendezvous target
$\mathbb{G}_P, \mathbb{G}_V, \mathbb{G}_\mathbb{a}$	target's position, velocity and uncertainty, respectively
$\mathcal{G}$	rendezvous location
$\mathbb{M}$	Sum of travelling time used by all participating vehicles
$\mathbb{S}$	Sum of waiting time used by all participating vehicles
$\mathbb{E}$	overall power consumption among all participating vehicles
$\mathbb{e}$	sum of the power consumed by all vehicles wait to get rendezvous
ε_i	power consumption for the ith vehicle to arrive at the rendezvous

List of Abbreviations

AUVs	autonomous underwater vehicles
ASD	annular space decomposition
APF	artificial potential field
AMVs	autonomous marine vehicles
ASVs	autonomous surface vessels
CC,CS	concentric circles, concentric spherical
DVL	doppler velocity logger
DSSD	distributed shell space decomposition
FM	fast marching
FS	full space
GA	genetic algorithm
GNC	guidance, navigation and control
H-ADCP	horizontal acoustic doppler current profiler
HF	High Frequency
LOS	line-of-sight
LSM	level set method
LBL	long baseline
NP	non-deterministic polynomial-time
PSO	particle swarm optimization
QPSO	quantum-behaved particle swarm optimization
RRT	rapidly-exploring random tree
ROMS	regional ocean model system
SSD	shells space decomposition
SMC	sliding mode controller

Certification

I certify that this thesis does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university; and that, to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where due reference is made in the text.

As specified under Clause 15 of Appendix E of the Flinders University Research Higher Degrees Policies and Procedures, I hereby request that access to this thesis be restricted for a period of 18 months from the acceptance of the award of the degree.

Adelaide,   14 Dec 2014

Zheng Zeng

Acknowledgements

First of all, I would like to thank my advisor A/Prof. Karl Sammut for his guidance and support over the years of PhD study. His pursuit of excellence in research and his constant desire to push the limits were very inspiring. Many thanks to my co-supervisors A/Prof. Fangpo He and Dr. Youhong Tang for their advices, motivation and inspiration. I would like to thank Dr. Andrew Lammas, he provided me with a wealth of insights and ideas that led to significant advances in my research.

Thanks also to all my former and current lab-mates in the Centre for Maritime Engineering, Control, and Imaging (CMECI). In particular, I would like to thank Amaury de Feligonde, Minh Tran, Aodren Revaud, Fabien Voltek, Vincent Lockner, Taehwan Joung, Matthew Kokegei, Amirmehdi Yazdani, Somaiyeh Mahmoudzadeh, Ivor Wanders, Frank Weijers, and Laurie Overbeek.

Special thanks to Amaury, I always believed to meet you and share six months experience with you in my life is a gift from God.

Minh Tran, a heartfelt thank you for your accompany and support, especially over the last two years. You were truly a great source of motivation; I am blessed to have you in my life. You made the PhD years a joyful experience.

I am very grateful to the great friends I met in Australia. Endah Martini, Zadia Novita, Yu Lian, Lei Mai, Ting Liu, He Shan, Yuting Yang, Wenjun Liao, Grace Malizani, Josiah Stanley, Michael Yen, Kyra Kingma, Suryantara Made, Hoa Lun, Hoahuong Lundet, Lingli Peng, Qijing Ji, Haifeng Shen, Shaowen Qing, Xiaojun Zhang, Yong Ma, Weijian Chen, Yang Yu, Wei Han, Rengui Peng: you made my PhD years very memorable, and I will always cherish our friendship.

Finally, I would like to acknowledge my dearest family: Mom, Dad, and my two sisters. I cannot thank you enough for all the love and support you have showed me. I would not have done it without you. Mom, your brave face through all the pain from fighting cancer gives me the strength to face any difficulties.

To those stories that happened

Chapter 1

Introduction

Autonomous Underwater Vehicles (AUVs) are a class of submerged marine robots using various enabling technologies to navigate and perform various tasks. AUVs have a variety of military, scientific research and commercial applications [1]. Military applications include surveillance and reconnaissance, anti-submarine warfare, payload delivery, time-critical strike, harbor protection, and etc.; AUVs are useful for oceanography, littoral ocean floor mapping, water profile sampling and other types of scientific research; AUVs are also increasingly used in the oil and gas market, due to their capability to operate very close to the seabed or in close proximity to industrial structures like those used by offshore industries.

However, the actual autonomy of contemporary AUVs is limited in many ways restricting their potential uses. Further advances in AUV autonomy will enable new operations, such as executing very long endurance missions with a minimum of supervision in unknown, dynamic and hostile environments.

Path planning is one of the main factors determining the level of autonomy achieved by AUVs. A path planner should be capable of finding a trajectory that safely leads the AUV from its initial or current position to its destination, and optimize a certain objective function, such as time or energy consumption. Reactive path planning received an increasing amount of attention among underwater operation for AUVs [2]. The environment usually contains time-varying currents and obstacles which

might not be fully characterized at the start of a mission. Rendezvous location may change over time and new obstacles may be detected as the AUV moves through the environment. A special case of non-stationary rendezvous target, such as a surface vehicle, may have moved during the operation. The adaptive path planner should thus have the capability to adapt to changing ocean environment, mission goals and system status [2].

This thesis explores the path planning methodologies for a single AUV or multiple AUVs operating in dynamic, cluttered, and uncertain ocean environments, with the main goal of achieving optimal trajectories. These trajectories should safely lead the AUVs from their initial or current position to their destination using either a chosen minimal energy or time-related cost criterion [3, 4]. After studying the main path planning approaches applied in robotics, especially relating to AUVs, this work focuses on the following three parts:

- B-Spline based quantum-behaved particle swarm optimization (QPSO) technique for path optimization.

- With the consideration of increasing the searching efficiency of evolutionary based technique, a novel Shell Space Decomposition (SSD) method has been developed and integrated into the B-Spline based QPSO path planner.

- As an extension of the general SSD scheme, a dynamic shell space decomposition scheme is developed for an on-line AUV planning system and a distributed shell space decomposition scheme is developed for path planning of multiple AUVs rendezvous.

This introduction continues on to discuss the main considerations which have formed this thesis. First, some background information on the motivations for this work is provided. Then, the objectives of this thesis are given. Finally, a statement of the contributions of this thesis and an outline of this document are presented.

1.1 Background

Marine vehicles are now a rapidly growing market due to new capability developments and improved affordability. According to Douglas-Westwood's world AUV market forecast, the market for AUVs, will increase rapidly from $2.6 billion

in 2013 to \$7.3 billion in 2024 [1]. Currently, the main consumers for AUVs are the military/security, scientific research, and the oil and gas industries.

Improving persistent presence is at the forefront in AUV research. AUVs are increasingly expected to operate for longer mission durations as well as with higher levels of autonomy and sensor capability to make efficient use of the vehicle and to reduce support vessel costs.

1.1.1 AUVs

A review of the existing AUVs suggests that they can be categorized into three broad classes based on their operation endurance (see Table 1.1, Table 1.2 and Table 1.3).

- Category I (Table 1.1): Long-range AUVs can travel for several thousand of kilometres to conduct science missions over periods of weeks or even month. To achieve longer range operation, vehicles of this category usually go rather slowly as the required propulsive power increases rapidly with operating speed. Autosub Long Range is designed by the National Oceanography Centre (NOC) with a target range of 6000 kilometres with over six months endurance[5]. Another good example of this category is the Reliant AUV developed by the US Naval Research Laboratory (NRL), which executed a record setting endurance of 109 hours traveling nearly 315 miles, at a depth of 10 meters and an average speed of 2.5 knots through busy waterways and strong currents [6].

- Category II (Table 1.2): Medium-range AUVs are autonomous submersible platforms which have an endurance of more than 72 hours with nominal load. The main advantage of these platforms is their mission endurance which allows them to carry various heavy and high-quality navigation and mission sensors. These platforms are generally supported with a research vessel, so that their batteries can be regularly replaced and the vehicle can be maintained. Some examples include the Bluefin-12 (US), Autosub6000 (UK), Hugin 3000 (Norwegian) and the Remus 600 (US).

- Category III (Table 1.3): Short-range AUVs are autonomous submersible platforms which have an endurance of less than 24 hours. Platforms of this class have less endurance than those of Category II, but they feature low cost, greater mobility and more commercial availability than other categories. AUVs such as

the Bluefin-9, Remus 6000, Teledyne Gavia, and MUNIN AUV and others, fall into this category.

Table 1.1 Category I - Long Range AUV (Endurance≥72 hours)

Name	Manufacture	Size(LxWxH)	Weight	Depth	Speed	Energy	Endurance	Figure
Autosub Long Range [5]	National Oceanography Centre	3.60mx0.90mx0.90m	660 kg	6000 m	0.40 m/s	18.9 KWh	4400 hours	
Submaran UUSV [7]	Ocean Aero	2.43mx0.48mx2.13m	56kg	23m	1.00 m/s	5 KWh	4000 hours	
Tethys [8]	MBARI	2.30mx0.31mx0.31m	110kg	200m	0.50 m/s	Information not available	740 hours	
Reliant [6]	U.S. Naval Research Laboratory's (NRL) + Bluefin Robotics	6.1mx0.53mx0.53m	612kg	4500m	1.28 m/s	40 KWh	109 hours	
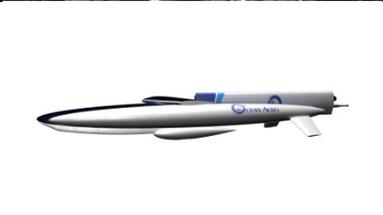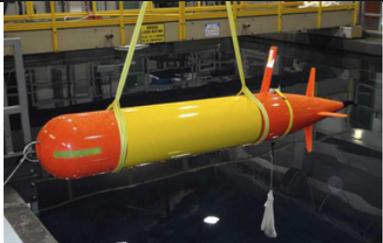
Table 1.2 Category II - Medium Range AUV (24 hours<Endurance<72 hours)

Name	Manufacture	Size(LxWxH)	Weight	Depth	Speed	Energy	Endurance	Figure
Autosub6000 [9]	National Oceanography Centre	5.50mx0.90mx0.90m	2000kg	6000m	1.00 m/s	4.5 KWh	70 hours	
Remus 600 [10]	Hydroid	4.27mx0.32mx0.32m	326kg	600m	1.50 m/s	5.2 KWh	50 hours	
Bluefin-12 [11]	Bluefin Robotics	3.77mx0.32mx0.32m	204kg	200m	1.50 m/s	4.5 KWh	26 hours	
Bluefin-21 [12]	Bluefin Robotics	4.10mx0.53mx0.53m	525kg	4500m	1.54 m/s	13.5 KWh	25 hours	
Hugin 1000 [13]	Kongsberg Maritime	4.70mx0.75mx0.75m	850kg	3000m	2.05 m/s	15 KWh	24 hours	

Table 1.3 Category III - Short Range AUV (Endurance≤24 hours)

Name	Manufacture	Size (LxWxH)	Weight	Depth	Speed	Energy	Endurance	Figure
MUNIN [14]	Kongsberg Maritime	3.0mx0.3mx0.3m	300kg	1,500m	2.05 m/s	5 KWh	22 hours	
Bluefin-9 [15]	Bluefin Robotics	1.65mx0.24mx0.24m	50kg	200m	1.52 m/s	1.5 KWh	12 hours	
Iver2 [16]	OceanServer Technology	0.13mx0.01mx0.01m	19 kg	100m	1.29 m/s	0.6 KWh	12 hours	
Remus 100 [17]	Hydroid	1.84mx0.19mx0.19m	45kg	100 m	2.3 m/s	1 KWh	10 hours	
Teledyne Gavia [18]	Gavia Defence	1.80mx0.20mx0.30m	49kg	1000m	1.00 m/s	1.2 KWh	7 hours	

The main feature of Categories I – long range AUVs is their significant endurance and range. It is a sign of the increasing maturity of AUVs that users are now demanding longer mission durations as well as higher levels of autonomy and sensor capability. With longer operation duration, AUV users will be able to make most effective use of the technology. AUVs of Categories I and II are generally suitable for open sea applications and require a suitable facility for launch and recovery. Most short-range AUVs of categories III can be applied open sea as well as constrained environment like coastal, river and harbor, or from small and narrow spaces and can be launched by hand.

1.1.2 Motivations

Many of current AUV applications consume much time lasting from days to weeks, and cover large areas of hundreds to thousands of square kilometres. Oceanographic processes, such as currents show great variability over such large expanses and will evolve over such durations. In most existing applications, AUVs are typically deployed from surface vessels with support personnel for deployment, piloting and recovery. During the course of the mission, the support vessel will shadow the AUV and provide much of the higher level decision making processes needed to deal with changes of oceanographic processes. The cost of keeping the support vessel on standby is generally by far the most significant component of the mission cost [5]. Recently, there has been growing interest in developing long range AUVs with increased autonomy to conduct science missions over longer periods without supervision [8], thereby human operator is removed from the control loop, reducing mission costs and extending their applicability

Two approaches have been explored to increase the range and endurance of the long-range AUV. The first approach is to increase the vehicle storage density-which indicates the capability to store more energy and sensors per unit volume. The second approach is to increase the operation efficiency of the vehicle. These approaches are illustrated in Figure 1.1.

Compact battery technology capable of storing more electricity is one way to increase the vehicle storage density and capacity. Most AUVs in use today are powered by rechargeable batteries, such as lithium ion, lithium polymer, nickel metal

hydride etc. Some vehicles use primary batteries which provide perhaps twice the endurance, but at a substantial extra cost per mission. A few of the larger vehicles are powered by aluminium based semi-fuel cells, however, these require substantial maintenance, require expensive refills and produce waste product that must be handled safely. An emerging trend is to combine different battery and power systems with super-capacitors [19, 20]. Another way to increase the vehicle's storage density capacity is to decrease the payload size [11], application of nanotechnology for developing sensor equipment holds potential to develop highly sophisticated underwater vehicles with reduced payload sizes and power consumption.

Research to increase the efficiency of the AUVs can be categorised along three directions:

- One direction is looking at reducing the AUV power consumption, including both propulsion power and hotel load [8]. Power for most AUVs is generated from power stored in on-board batteries. Propulsion power can be reduced through more efficient motion. Hotel load is defined as the power for on-board instrumentation, guidance, computers, and communication devices, and has steadily reduced through advancements in electronic systems.

- A second direction is looking at achieving advances in mechanical design, particularly the vehicle profile and surface design to reduce vehicle drag [21, 22]. A good example of this category is Tethys, its hull, motor and propeller were computer designed and tested to minimize drag and maximize efficiency of propulsion. The third major direction is increase the autonomy of vehicle [8].

- Mission scheduling and path planning hold tremendous potential to enable long range operations. A vehicle could be launched from shore where upon a path planning system could be used to generate a trajectory that exploits the ocean energy taking use of the favourable currents to propel the vehicle, and lead the vehicle to a remote work site, perform a survey, and then return to shore completely on its own. This will greatly diminish the costs for AUV operations since there is no need for a support vessel and vehicle operations can be monitored on-shore, thereby enhancing the affordability of AUVs to science and industry.
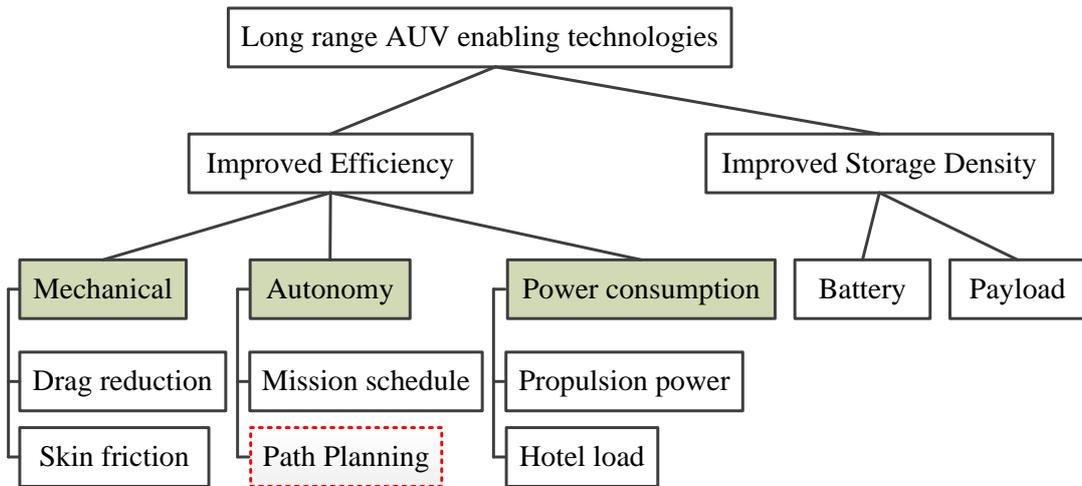
Figure 1.1 Methodologies to enable long range AUVs

1.1.3 Path Planning Challenges

Path planning for AUVs in the ocean has become crucial for many applications, ranging from security and acoustic surveillance, to collection of ocean data at specific locations, for ocean prediction and monitoring. This thesis addresses several challenges associated with path planning and re-planning of AUV missions through cluttered ocean environments and contributes to the desired increase in AUV autonomy.

The following is a general definition of the **AUV Path Planning Problem**:

Given the present state of a single AUV or team of AUVs and information of the ocean environment, compute a trajectory towards a desired goal state that optimizes a certain objective function while respecting the properties of the vehicle(s) and avoiding collisions with obstacles [23].

This definition encodes that there are two main concerns are desired as the outcome of a planning, which are feasibility and optimality.

(1) Path optimization

In general, the path planning problem aims to optimize at least one of the following aspects of performance:

- Safety

Traditionally, path planning for AUVs has been related to safety conditions. The path should be devoid of known obstacles or hazardous areas. Knowledge of an obstacle's position is usually not perfect. Obstacles may be detected as the vehicle moves

through the environment or their location may change over time. These moving obstacles in a special case could be other intelligent agents with which collisions should be avoided. The vehicle should thus have the capability to compute or update its path in real-time. Different computational methods were employed to plan safe paths for AUVs. Warren [5] used artificial potential field (APF) algorithms to solve the path planning problem. The algorithm generates safe paths by introducing an artificial potential field around the obstacles preventing vehicles from collision. While the use of the APF cost-terms allows fast convergence and is easy to apply to irregularly shaped obstacles, it is susceptible to local minima. Graph Search Methods include Dijkstra's algorithm [24], A* algorithm [25, 26] and the D* algorithm [27] have been employed for path planning. In these methods, a chart or graph is produced showing free space where no collision will occur and forbidden spaces where a collision will occur. Based on this graph, a path is selected by piecing together the free spaces or by tracing around the forbidden spaces. Graph Search methods are fast to solve few-dimensional problems, however, they are commonly criticized for their discrete state transitions which unnaturally constrain the motion of a vehicle to limited directions. Risk-aware path planners are recently reported in [28] described path-planning methods that make safe operating of AUVs in high ship and boat traffic areas. This work is extended in [29] by looking at minimizing risk through search in a time expanded planning environment while taking advantage of ocean current predictions. While it is good to see that this work use ocean current predictions instead of quasi-static currents data for planning AUV paths, however, it still applies A* as the optimization algorithm which is not suitable for many-dimensional problems.

- Energy consumption

The problem of AUV mission planning to optimize the energy cost of travelling in ocean environments with quasi-static current fields was considered by Alvarez et al. [30]. The developed planning algorithm integrated the currents map with an evolutionary path planner, providing the path with minimum energy requirements. In these studies the AUV speed with respect to the seabed was assumed to be constant. Thus, the AUV should be able to adapt its speed depending on the current field, keeping the total speed constant through the planned trajectory. One of main

drawbacks of this work is that it makes the assumption that all the paths are strictly monotonic with respect to the x-coordinate, while the y-coordinates are set up as the chromosome genes for the evolutionary algorithm, this assumption simplifies the problem but the trajectories are not allowed to go backward. Koay and Chitre using A* to find a path for an AUV to navigate in the presence of currents and obstacles while consuming minimum amount of propulsion energy [4]. The authors concluded that the expected energy saving is the largest when the vehicle is allowed to operate at speeds comparable to the water current while the savings diminish when the vehicle is required to operate at higher speeds. An alternate approach to minimum energy continuous path planning was proposed by [31]. This work considered vehicle actuation limits and use a multi-dimensional cost function for generating energy optimum paths in estuarine environments. Regarding the optimization algorithm, however, this work only applied a simple, gradient based approach for proving the concept. Thus, the successful optimization relies heavily on a careful choice of path parameterization. The Particle swarm optimization (PSO) algorithm has also been studied [32] for energy conservation by taking advantage of the time-varying ocean currents. Like other evolutionary algorithms, its drawback is that they may converge to a suboptimal solution within a finite time.

- Travel time

Garau et al [8] considered a more common situation on actual AUVs, where the thrust power is usually kept constant during the mission. Thus, in the usual situation the optimization of energy consumption agrees with finding the minimum-time path. This work also used A* search to find the optimal paths, so it will face the same problem of being computationally expensive to employ in many-dimensional search spaces. Fast marching method was applied by Petres et al. [33, 34] to obtain the minimum time path in a given current field. Similar research that applies wavefront methods are the spatiotemporal wavefront algorithm described in [35] and the sliding wavefront expansion algorithm described in [36, 37]. Level set methods [38] are more recently introduced for path planning in a currents fields whereby the time-optimal path of the AUV is generated by solving a particle tracking equation backward in time. These deterministic methods have proven to be quite effective in few-dimensional spaces, but significantly less effective in many-dimensional settings.

(2) Path re-planning

Despite the advances made through current AUV research endeavours, the existing or planned AUV systems are far from capable of fully autonomously taking over on-line dynamic and complex operations. An effective fully-autonomous AUV system, should be capable of reconfiguring itself in response to environmental changes in order to improve mission performance and optimize the strategy for detection and measurement of vector/scalar fields and features of particular interest (adaptive ocean sampling).

Dynamic path planning can be solved by correcting the previous path solution based solely on new information (such as newly detected obstacles). One approach by Wzorek et. Al. [39, 40] suggests that for those situations that need to be handled with urgency, only the colliding segments need be corrected. Given the range of a typical AUV sonar sensor and the maximum AUV travel speed, there may be sufficient time, however, to generate a better solution that can take into consideration both new and previous information. Previous research, such as [41] and [42], also looked at solving the dynamic path planning problem by using a hierarchical approach. This approach generates a solution with greater accuracy locally where the vehicle's immediate reaction to an obstacle or a threat is needed. However, it may not be suitable in dynamic environments where the current field is continuously changing during the mission. Another approach based on Rapidly-exploring Random Trees (RRT) [43] pro-poses a re-planning strategy called Dynamic RRT [44, 45], where the RRT is repaired by pruning newly-invalid branches and then growing the remaining tree until a new solution is achieved. RRT strategy is suitable for handling dynamic environments with moving targets and obstacles, however, it has not been designed to handle optimal path planning with dynamic currents.

(3) Cooperative path planning

Recently, there also has been widespread interest in the use of fleets of autonomous robots cooperating to achieve a desired goal. Multiple cooperative AUV systems hold great promise for use in large scale oceanographic surveys, mine counter-measurement missions and other similar underwater tasks due to better resource and task allocation. Simultaneous use of multiple vehicles can improve performance, reduce mission time, provide broader or more robust data, and increase the likelihood

of mission success. The ability to rendezvous at a specified location is one of the mission requirements for multiple cooperative vehicle systems [46]. The purpose of rendezvous could be an initial step in formation creation, data exchange, vehicle recharging, maintenance, or collection. The literature on cooperative use of multiple vehicles is vast. Specific attention is given to a number of problems including cooperative motion control of fleets of autonomous vehicles [47], and cooperative vehicle localization and navigation [48, 49].

Versatile path planning system is of crucial importance to success and efficient rendezvous. However, previous work is dominated by computing optimal trajectories for single-vehicle [50, 51]. There is still considerable work to be done to develop advanced methods for cooperative multiple marine vehicle path planning that explicitly addresses simultaneous rendezvous. One challenge that arises is that it may be very difficult and energy consuming for the early arrived vehicle to wait for the other vehicles to arrive, especially if there is a bad weather situation or a temporary adverse ocean current condition in the region of rendezvous.

As analysed above, although several path planning and re-planning methods have been previously proposed for autonomous vehicles, several difficulties still remain for AUV-oriented applications. Path planning and re-planning for AUVs that operate across a large geographical area is a large-scale optimization problem. The computational requirements grow exponentially for solving high dimensional problems. In order to speed up the planning process and reduce the memory requirement, most conventional path planning approaches project the 3D environment to 2D space [50, 52, 53]. However, this 2D space cannot completely embody all the 3D information of the ocean environment. The specific characteristics of ocean current environments and various manoeuvrability of different vehicles pose more challenges to cooperative path planning of multiple AUVs. Evolutionary algorithms have been proven to be an efficient and effective way of dealing with non-deterministic polynomial-time (NP) hard problems [54]. Also, evolutionary algorithms are population based optimization techniques and amenable to be implemented on a parallel machine to achieve super linear speed-up with the number of processors [55]. However, they have the drawbacks of inconsistency and incompleteness of searching for the optimal path. With acknowledge of this

limitation of the evolutionary path planner, therefore methodologies should be researched to improve evolutionary planners with increased path searching efficiency, and to have better robustness and quality.

1.2 Objectives

The objective of this thesis is to develop optimal path planning techniques towards safe and efficient operation of AUVs in dynamic, cluttered, and uncertain ocean environments. The thesis commences by researching the problem of path planning for a single AUV operating in complex ocean environment [56-58]. Subsequently, the work is extended to address the problem of developing efficient path re-planning for an AUV operating in spatiotemporal currents field [59]. Furthermore, the work is extended to the case of multiple autonomous vehicles, in the context of path planning for simultaneous rendezvous [60].

- **Path Planning in Complex Ocean Environments**

The first objective of this thesis is to develop and evaluate efficient but rigorous path planners that generate optimal paths for AUVs in realistic ocean environments. The challenge here is to develop computationally efficient and rigorous frameworks that accommodate both the environmental constraints and vehicle manuverbility while at the same time providing an optimal path for the vehicle [9,10]. The ocean environments are composed of strong currents field, irregularly shaped terrains and obstacles, the position of which may be dynamic and uncertain.

- **On-line Path Re-planning in Spatiotemporal Environments**

The second objective is to develop advanced methods that enable an AUV to adapt and regenerate its trajectory during the course of the mission using continuously updated current profiles from on-board sensors, such as a Horizontal Acoustic Doppler Velocity Logger. Again, the ocean environments may include spatiotemporal currents, irregularly shaped terrains as well as obstacles whose position coordinates are uncertain. An AUV may be instructed to rendezvous with a moving target, such as a mother ship or autonomous surface vehicle. Motivated by the observations illustrated in above section, this thesis looking at address the problem of developing the path re-planning technique that is computationally

efficient enough for real-time, online implementations.

- **Path Planning for Multi-AUVs Rendezvous**

As an extension of the work on path planning of a single vehicle, the third objective of this thesis is to develop an effective path planning guidance system for multiple marine vehicles rendezvous in ocean environments. This system should be able to generate trajectories for multiple vehicles with minimal time usage over all participating vehicles and simultaneous arrival of the vehicles at their selected rendezvous destination. It also needs to be lightweight in terms of running time and capable of incorporating different factors, such as vehicles' dynamic constraints and environmental conditions, influencing a given mission.

1.3 Assumptions and Scope

This section introduces the assumptions and scope of the research undertaken in the development of path planning techniques contribute to long range AUVs with increased autonomy.

- In this thesis, if not specially acknowledged for certain case study, it is assumed that the currents field is exactly known. This assumption somewhat simplifies the problem because the ocean environment information is always associated with some levels of uncertainty. Since the work is directed towards oceanic application, the path planners are tested in some case studies with currents field containing various level of uncertainty and its effect on the performance of these path planners is then analysed.

- Except for the on-line path re-planning system presented in Chapter 4, the research is aimed at generating globally optimal solutions for AUVs operating long-term missions of durations from one to several weeks and trajectories length of hundreds of kilometres.

- It is assumed that the interaction between the AUV and the currents is purely kinematic. In other words, it is assumed that the AUV behaves as a point mass and exhibits no inertia.

- It is assumed that the vehicle's water-referenced speed is constant. Since this speed is proportional to the cubic root of the constant thrust, equivalently, the

vehicle has constant thrust power consumption. This assumption comes from the observation that on actual AUVs, the thruster power is usually set to be constant during the mission.

- This work primarily focuses on a high-level planning architecture with simplified dynamics enabling optimum trajectory adjustments to take advantage of the favourable currents. Previous work has been done studying the full dynamics of the vehicle and the control strategies that drive the vehicle to the desired planned trajectories [61]. In this research, it is assumed that AUV controller is able to overcome currents disturbances and to accurately drive the AUV to a planned waypoint.

1.4 Statement of Contributions

This thesis addresses the problem of developing computationally efficient optimal path planning and re-planning technique for applications involving either a single or teams of autonomous marine vehicles operating in a dynamic, cluttered, and uncertain ocean environment. In this thesis, a B-Spline based QPSO path planner is proposed for both single and multiple AUVs which combines the main advantages of previously published approaches. Most importantly, several new concepts are proposed to improve the performance of this planner. The majority of the ideas and algorithms that are introduced can be considered independent of the underlying optimization method and as such can be generally used with path planning techniques other than QPSO. Furthermore, some of the concepts fit within the broader field of path planning for field vehicles, specially, unmanned air vehicles (UAV), however, it is restricted here to the problem of AUV guidance and chooses to illustrate the theory using B-Spline based QPSO path planner as the implementation framework. The original contributions of this thesis can then be stated as follows:

- **A Spline based QPSO algorithm is developed for solving the optimal path planning problem for an AUV operating in turbulent, cluttered and uncertain environments.**

   The proposed path planner combines the main advantages of previously

published approaches. These include smooth curvature paths represented by the Spline to accommodate constraints imposed by the manoeuvrability of the vehicle; and a QPSO algorithm enables the path planner to obtain a more optimized trajectory than the A*, RRT, GA and PSO based methods.

- **A new shell space decomposition (SSD) scheme is proposed to increase the searching efficiency of the B-Spline based QPSO path planner.**

  The SSD scheme decomposes the search space into multiple concentric shells radiating out from the start point to the destination, with one or more control points allocated to the region of each shell. The trajectory is then generated from this set of control points using Splines. This arrangement gives freedom to the placement of the control points, but still restricts the search space to reduce computation time. The SSD scheme has been integrated with the B-Spline based QPSO path planner and tested to find an optimal trajectory for an AUV navigating through a variable ocean environment in the presence of obstacles whose position coordinates are uncertain. Simulation results show that the proposed SSD approach is able to obtain a more optimized trajectory than the concentric circle (2D)/spherical surface (3D) constrained method. The SSD method also achieves faster convergence speed and uses less computation time than the unconstrained full space searching method.

- **The general SSD scheme is extended. A dynamic SSD scheme is proposed for an AUV on-line planning system.**

  This system combines path re-planning with the B-spline based QPSO technique to adapt and regenerate the trajectory during the course of the mission using continuously updated current profiles from on-board sensors, such as a Horizontal Acoustic Doppler Velocity Logger. The proposed path planner is tested to generate an optimal trajectory for an AUV navigating through a spatiotemporal ocean environment in the presence of irregularly shaped terrains as well as obstacles whose position are uncertain. Simulation results show that with same amount of computational load, a path planner based on the proposed dynamic SSD scheme utilizing the path re-planning methodology is able to obtain a more optimized trajectory than one relying on reactive path planning.

- **A distributed shell space decomposition (DSSD) scheme is developed and integrated into a novel path planner for rendezvous of multiple autonomous marine vehicles (AMVs) in variable ocean environments.**

  The proposed QPSO-DSSD path planner is integrated with an optimized mass-centre rendezvous point selection scheme to detect the optimal rendezvous position, along with an optimal operational speed scheme to improve the performance of simultaneousness arrival at the rendezvous point, and power consumption to complete the mission. Simulation results demonstrate that the proposed QPSO-DSSD path planner is capable of finding more optimized trajectories that not only take maximum advantage of favourable currents, but also lead the vehicles to arrive more close to each other at the selected rendezvous point in time.

1.5 Publications

This thesis is the result of work done whilst studying at the Flinders University, South Australia. The work developed in this thesis have been presented at several international conferences and published by peer review journals:

- B-spline based QPSO path planner and shell space decomposition for AUV application in various scenarios are presented in [J1, C1, C2, C3].

- The on-line dynamic path re-planning system for an AUV operating in a spatiotemporal ocean environment is presented in [J2].

- The cooperative path planning system for multiple AUVs simultaneous rendezvous will appear in [J4, C4].

- The analysis of applying the proposed shell space decomposition scheme with another evolutionary technique, Imperialist Competitive Algorithm, rather than the QPSO algorithm is discussed in [J3].

- **Journal Publications**

J1. Zeng, Z., Lammas, A., Sammut, K., He, F., Tang, Y.: Shell space decomposition based path planning for AUVs operating in a variable environment. Ocean Engineering 91, 181-195 (2014).

J2. Zeng, Z., Sammut, K., Lammas, A., He, F., Tang, Y.: Efficient Path Re-planning

for AUVs Operating in Spatiotemporal Currents. Journal of Intelligent & Robotic Systems, 1-19 (2014). doi:10.1007/s10846-014-0104-z

J3. Zeng, Z., Lammas, A., Sammut, K., He, F., Tang, Y.: Imperialist Competitive Algorithm for AUV Path Planning in a Variable Ocean. Applied Artificial Intelligence, 29(3), (2015).

J4. Zeng, Z., Lammas, A., Sammut, K., He, F., Tang, Y.: Evolutionary Path Planning for Rendezvous of Multiple AUVs Operating in a Variable Ocean. To be submitted to Ocean Engineering.

- **Conference Publications**

C1. Zeng, Z., Sammut, K., Lammas, A., He, F.: Efficient path evaluation for AUVs using adaptive B-spline approximation. Oceans, 2012. IEEE/MTS, 2012.

C2. Zeng, Z., Lammas, A., Sammut, K., He, F.: Optimal path planning based on annular space decomposition for AUVs operating in a variable environment. Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES. IEEE, 2012.

C3. Zeng, Z., Lammas, A., Sammut, K., He, F., Tang, Y.: Long-Range Path Planning for AUVs Exploiting Ocean Energy with Forecasted Currents. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2014) Workshop on Persistent Autonomy for Marine Robotics, 2014.

C4. Zeng, Z., Lammas, A., Sammut, K., He, F., Tang, Y., Ji, Q.: Path planning for rendezvous of multiple AUVs operating in a variable ocean. In: 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2014 2014, pp. 451-456.

1.6 Thesis Outline

The thesis is organized as follows.

Chapter 1 provides a review of the existing AUVs and categorizes them into three broad classes based on their operation endurance. Detailed definitions and background relating to the challenge of path planning for AUVs are also presented.

Chapter 2 describes the path planning missions and formulated the optimization problem. All aspects of the implemented planner including the model of ocean currents, obstacles and dynamic rendezvous target, cost functions, B-Spline path

formation and data structures are described in Section 2.3. This Chapter continues by providing extensive review of existing techniques to the problem of path planning for an AUV. In Section 2.4, path planners based on A*, RRT, GA and PSO and QPSO algorithms are developed, and their performance are compared in Section 2.5 through several scenarios for travelling through a static currents field.

Chapter 3 then presents a novel SSD scheme for the B-Spline based QPSO path planner. A series of simulation experiments are run to compare the proposed SSD approach the existing circle/sphere constrained methods and the full space searching method in Section 3.4. Monte Carlo trials are utilised to assess the robustness of the SSD method.

Chapter 4 starts by introducing the concept of on-line path planning system that adapt and regenerate the trajectory during the course of the mission using continuously updated current profiles from on-board sensors. This chapter then extends the general SSD scheme in Chapter 3, and developed a dynamic SSD scheme for an AUV on-line planning system. Various simulation scenarios in which the AUV regenerate the trajectory during the course of the mission using continuously updated current profiles are presented. Subsets of representative Monte Carlo simulations are also utilised to evaluate robustness and superiority of the proposed planner.

Chapter 5 then uses the basic SSD concept to construct a DSSD scheme for rendezvous path planning of multiple autonomous marine vehicles (AMVs) in a variable ocean environment. The proposed QPSO-DSSD path planner is integrated with an optimized mass-centre rendezvous point selection scheme to detect the optimal rendezvous position, along with an optimal operational speed scheme. A detailed description of a series of multiple AMVs scenarios as well as the simulation results will be given.

Chapter 6 includes a summary of the overall conclusions obtained from the research and recommendations for further directions of the research are given.

Chapter 2

Problem Formulation and Path Planning

This chapter presents a methodology for formulating the AUV path planning problem in the context of the environmental constraints. The turbulent, cluttered and uncertain environments modelled here incorporate strong currents field, irregularly shaped terrains and obstacles, the position of which may be dynamic and uncertain. A cost function with the objective of achieving minimum time consumption is introduced. Path planners based on A*, rapidly exploring random tree (RRT) and evolutionary algorithms including genetic algorithm (GA) and particle swarm optimization (PSO) and quantum-behaved particle swarm optimization (QPSO) are developed, and their respective performances are compared and analysed through several representative test scenarios.

2.1 Introduction

AUVs usually operate in dynamic, cluttered, and uncertain ocean environments. Path planning is of primary importance to ensure safe and efficient operation of a vehicle in such environments. A path planner should be capable of rapidly reacting to fast changing environments and finding a trajectory that safely leads the AUV from its initial or current position to its destination with minimum energy/time cost [3]. By choosing an appropriate trajectory, the path planner may enable an AUV to both

bypass adverse currents as well as exploit favourable currents to achieve greater speeds, while substantially saving energy.

A variety of approaches have been developed and applied to the AUV path planning problem. These include Dijkstra's algorithm, A* algorithm, Field D* algorithm, Fast Marching (FM) algorithm, and Artificial Potential Field. Details of these algorithms are presented in Subsection ☐ of this Chapter. Although several path planning methods have been proposed for autonomous vehicles, several difficulties still remain for AUV-oriented applications. Path planning for AUVs that operate across a large geographical area is a large-scale optimization problem. The computational requirements grow exponentially for high dimensional search space. In order to speed up the planning process and reduce the memory requirement, most conventional path planning approaches project the 3D environment to 2D space. However, this 2D space cannot completely embody all the 3D information, including currents, bathymetry and obstacles of the ocean environment. Evolutionary algorithms have been proven to be an efficient and effective way of dealing with non-deterministic polynomial-time (NP) hard problems [54]. Also, evolutionary algorithms are population based optimization techniques and amenable to be implemented on a parallel machine, such as an embedded computer, to achieve super linear speed-up with the number of processors [55].

The genetic algorithm (GA) [62-64] and the particle swarm optimization (PSO) [32, 65] are two well-known forms of evolutionary algorithms that are generally recognized to be effective optimization techniques for solving path planning problems. More details of these algorithms are presented in Subsection ☐ of this Chapter. Quantum-behaved particle swarm optimization (QPSO) is a new evolutionary algorithm first proposed by Sun et al. [66]. The inspiration of QPSO came from quantum mechanics and the trajectory analysis of PSO [67]. In QPSO, the particle is assumed to have quantum behaviour and to be in a bound state, and is further assumed to be attracted by a quantum potential well centred on its local attractor, thus having a new stochastic update equation for its position [66]. Later, a global point known as the mean best position was introduced into the algorithm in order to enhance the global search ability of the QPSO algorithm [66]. Recently, the QPSO algorithm has been successfully utilized to solve optimization problems in

many engineering applications such as electromagnetic design [68-70], composite structures design [71], engineering design [72], image processing [73, 74], economic power dispatch [75-77], to name only a few.

In this chapter, path planners based on classic A*, RRT and its improved version RRT*, as well as evolutionary algorithms such as GA, PSO and QPSO are developed, and their respective performance are compared in relation to the problem of finding the optimal trajectory for an AUV. Various scenarios are used to access the performance. Moreover, a thorough robustness assessment is presented for each algorithm to compare the effectiveness of these proposed path planners.

The rest of this chapter is organized as follows. In Section 2.2, a literature review of relevant research topics relevant to the scope of this thesis is given. Section 2.3 describes the path planning missions and formulates the optimization problem. Mathematical models describing the variable currents field, subsea terrain and behaviour of dynamic obstacles and moving target are also provided together with a measure of estimated travel time consumed along a path. Section 2.4 introduces path planners based on A*, RRT and improved version RRT*, GA, PSO and QPSO methods. The simulation tests and robustness assessment using Monte Carlo trials are presented in Section 2.5. Concluding remarks are then presented in Section 2.6.

2.2 Literature Overview

This section contains a literature review of the research topics relevant to the scopes of this thesis. In particular, it starts with a review of some trajectory generation techniques. Following this, it provides a review of the optimization techniques for path planning. Subsequently, algorithms and methodologies that are suitable for this research are identified.

2.2.1 Path Shape and Properties

The shape and properties of the path have a direct influence on the path planning system. Generally, the path-planning algorithm will first define a number of ordered waypoints on the operation space which will have to be connected sequentially so as to form the path. Connecting the waypoints can be achieved in many different ways, with each one having its own advantages and drawbacks (see Table 2.1). However

three main categories are distinguished:

Table 2.1 Literature review of path shape and properties

Path Shapes		Reference	Comments
Category 1: Straight Lines		Kwak, S.H., 1990	Not possible for smooth transition.
Category 2: Dubins	straight lines + circular arcs	Dubins, 1957	Difficult transition maneuvers between straight and circular segments.
	straight lines + clothoid arcs	Fraichard and Scheuer, 2004	Linearly varying curvature over the path length, increased computational cost.
	straight lines + pythagorean hodographs	Bruyninckx and Reynaerts, 1997	Guaranted to satisfy the curvature constraints with increased computational cost.
	straight lines + Fermat's spiral arc	A. R. Dahl, 2013	Curvature-continuous paths with a very low computational cost compared to clothoids.
Category 3: Piecewise Polynomial and Splines	piecewise-constant polynomials	Z. Qu et al., 2004	Twice differentiable, and the corresponding steering controls are piecewise continuous. but not be able to provide complex curvature profile for long-range global path planning.
	natural splines	T. I. Fossen,2011	Curvature continuous paths pass through the waypoints, but relocating one waypoint induces changes along the path.
	cubic hermite spline	Christoph Sprunk, 2008	Similar to natural spline with the possibility of assigning the derivative values at the control points. It does not have a continuous second derivative.
	B-spline	K. G. Jolly et al.,2009	Continuous curvature without passing through all the control points, relocating one waypoint only locally changes the path.

- Category 1: Straight line paths

In many early applications [78] the path is defined as the sum of the successive straight lines that connect these waypoints. Due to physical constraints though, it is not possible for a vehicle to achieve a smooth transition between two straight lines because such a path generally has a discontinuous first derivative at the locations of the waypoints.

- Category 2: Dubins and similar paths.

Continuing the first category, a simple and intuitive way of producing continuous paths without sharp corners can be achieved by inscribing a circle between two lines to form a curved path. In 1957, Dubins assumed that for a vehicle moving at a constant speed, the shortest possible path that meets a maximum curvature bound between a start and end position with their predefined orientation, respectively will consists of at most three pieces, each of which is either a straight line or an arc of a circle of appropriate radius [79]. Dubins Paths consisting of straight lines and circular arcs have been studied extensively in [80-82], Moreover, Techy and Woolsey [83], Bakolas and Tsiotras [84, 85], deal with the problem of finding a Dubins path for a vehicle that moves in a constant drifted (can be wind or current for aerial or marine vehicles, respectively). In this case, it is not always possible to find a Dubins path. The main disadvantage of the Dubins path is the curvature discontinuity which occurs at the meeting points of two consecutive path segments, for more details see [86].

The continuous circular arcs can also be produced by using other methods. Three of the most popular approaches are the clothoid arcs [87], the Pythagorean Hodographs [88] and the Fermat's spiral arc [89]. The clothoid is useful in path-planning applications due to its property of having its curvature change linearly with arc-length. This notion can be extended to three dimensions, and is consequently also true for the torsion [90]. Pythagorean hodographs employ a polynomial of the fifth degree in order to produce a closed-form solution that gives a flexible path with velocity continuity. Further work has been done in [88] so as to guarantee that the curvature constraints are satisfied as well. The main drawback of these approaches is the increased computational cost which can be a heavy burden when implementing

them in, for instance, on-line AUV path planning. A more detailed comparison of Dubins paths, clothoid arcs and Pythagorean hodographs can be found in [86]. Compared to clothoids, Fermat's spiral can be used in order to generate curvature-continuous paths with a very low computational cost. The approach was further extended in [91] where an alternative parameterization of Fermat's spiral was proposed, which are suitable for path planning.

- Category 3: Piecewise polynomial and Spline paths.

The third category of methodologies pertains to generate trajectories using piecewise polynomial and Splines. piecewise polynomial and Spline curves need only a few variables (coordinates of the control points) in order to define complicated curved paths, consequently, both for path optimization and for on-line implementation, it is easier to deal with small number of parameters when generating paths with minimum computational cost.

A family of sixth-order piecewise-constant polynomials is applied to generate feasible trajectories in [92]. The resulting trajectory is twice differentiable, and the corresponding steering controls are piecewise continuous. However, this technique is only tested for local analytical motion planning of car-like mobile robot, it may not be able to provide complex curvature profile that is desired for long-range global path planning. Natural splines pass through the waypoints and also produce curvature continuous paths, however, they do not possess local control which refers to the case where relocating one waypoint induces changes throughout a larger part of the path. In addition, the resulting paths are not very practical [93]. An alternative is the cubic Hermite Spline which passes through all the waypoints and allows the derivative values at the control points, and also permits local control over the path [94]. The disadvantage of this Cubic Hermite Spline is that it does not have a continuous second derivative.

B-splines can give paths a desired second derivative continuity without passing through all the control points used to define it [95]. The authors in [96] presented an implicit time-parameterization of the trajectory using a B-spline representation. Designing an obstacle-avoiding B-spline path was dealt with by [97], whereas the real-time modifications of a spline path is proposed in [98].

Another valuable characteristic of the adopted B-Spline curves is that the curve is

tangential to the control polygon at the starting and ending points. This characteristic can be used in order to define the starting direction of the curve, by inserting an extra fixed point after the starting one. These two points can define the direction of the curve at the corresponding region. This is essential for the path planning of autonomous vehicles, as their travelling angles are continuously defined. Consequently the direction of the designed path line in the starting position must coincide with the current direction of travel in this position, in order to ensure curvature continuity of the whole path line.

2.2.2 Optimization Techniques for Path Planning

Path planning in the robotics literature covers the topic of finding a feasible and usually optimal path to allow a robot to move autonomously from one location to another in the environment [99]. This subsection presents a detailed literature review of the state-of-the-art AUV path planning techniques with discussion of their assumptions and drawbacks.

A brief comparison of path planning techniques for AUV is available in Table 2.2. Two important properties of path planning algorithms are the completeness and the optimality of the algorithm. Two forms of completeness are probabilistic completeness and resolution completeness. An algorithm is called Resolution completeness if it is guaranteed to find an existing solution in finite time as long as the resolution of an underlying grid is fine enough. Most resolution complete planners are graph search methods such as Dijkstra, A* and Field D*. In contrast, an algorithm is considered probabilistically complete if the probability of finding a path approaches 100%. Several sample-based methods, such as RRT and evolutionary algorithms are probabilistically complete. The performance of a probabilistically complete planner is shown by the rate of convergence. Optimality is the property that the planner computes the optimal path with respect to some criterion, e.g., minimal time, energy consumption or distance. Probabilistic optimality and resolution optimality are similarly to the definition of probabilistic completeness and resolution completeness.

Table 2.2 Literature review of optimization techniques

Optimization Techniques		Reference	Completeness	Comments
Graph Search Methods	**Dijkstra**	Dijkstra, 1959	Resolution completeness	Grid-search based scheme, discrete state transitions
	A*	Carroll et al., 1992		
	Field D*	Ferguson & Stentz, 2006		Computationally expensive in many-dimensional problems
FM & LSM		Pêtrès et al., 2007; Lolla et al., 2012	Resolution completeness	Only allow linear cost function to preserve computational efficiency
APF		Warren, 1990; Kruger et al., 2007	Probabilistic resolution completeness	Fast but susceptible to local minima
RRT		Tan et al., 2004; Rao and Williams, 2009	Probabilistic completeness	Fast and effective with many-dimensional configuration spaces, but solutions are sub-optimal and often require further refinement.
Evolutionary Approaches	**GA**	Alvarez et al., 2004;	Probabilistic completeness	Practical in in many-dimensional problems, but may converge to a suboptimal solution within a finite time
	PSO	Saska et al., 2006; Witt and Dunbabin, 2008		
	QPSO	Fu et al., 2012		

- **Graph Search Schemes**

Graph-based methods are a classical path planning approach that lies in the category of Discrete Optimal Planning [100]. A grid-shape graph represents the search space with the edges labelled indicating the cost of traveling from a vertex to one of its neighbours. Dijkstra's algorithm is probably the first graph method adapted to search for a minimum cost paths, it computes every possible path from a starting point to a specified destination point [24]. With its heuristic searching ability, the A* algorithm [101] has proven to be more efficient. The heuristic function provides an estimate of the cost of the best route that passes through a particular node. The algorithm keeps track of the cost of the route leading up to a particular node along with the heuristic cost function to determine which node it must visit next.; Carroll et al. [25] applied A* on a quad-tree search space, which was adapted to the ocean currents field, i.e. it has higher resolution where the ocean currents vary more spatially; or more formally, where the gradient of the ocean currents is greater. Overall, these grid-based graph search method are commonly criticized for their discrete state transitions which unnaturally constrain the motion of a vehicle to limited directions. There exists a number of variants of A* that are worth mention. Any-angle methods, like Theta* [102, 103], try to obtain shorter paths alleviating the angle discretization problem caused by the search grid. The Field D* algorithm uses a linear interpolation-based method to allow continuous heading directions, but these variants of A* still not fix the problem of computationally expensive to employ in many-dimensional problems [27]. A review of multi-resolution decomposition for graph search schemes is detailed in Appendix A.

- **Fast Marching and Level Set Methods (FM & LSM)**

The FM algorithm can be regarded as a continuous version of Dijkstra's algorithm. It uses a first order numerical approximation of the nonlinear Eikonal equation. FM algorithm have been recently applied for AUV path planning by [33, 34]. A heuristically guided version of FM, known as FM*, maintains the accuracy of the FM algorithm along with the efficiency of the A* algorithm; however it is limited in that it uses a linear anisotropic cost function to improve the algorithm computational efficiency. The FM* scheme is improved in [36, 37, 104] by using wavefront

expansion to calculate shortest time paths and also determines the departure time of the vehicle from the starting point. The LSM is a more general technique than the Fast Marching algorithm for wavefront expansion [105]. This method had been applied for path planning in flow fields. The time-optimal path is generated by solving a particle tracking equation backward in time after it evolves a front from the vehicle's start location until it reaches the goal [38]. The level set method provides the ability to solve more complex problems, but it takes longer computation time than Fast Marching.

- **Artificial Potential Field (APF)**

An artificial potential field for global path planning based on a linear energy cost-function was originally proposed by Warren [106]. Since then, it has been widely used by the robotics community and many problem specific developments have been made to this algorithm [107]. The key idea of this approach is to introduce an artificial potential field on the obstacles that prevents vehicles from getting very close to them, thus, generating safe paths. Kreuger [31] then replaced the single term cost-function with one that incorporates a mixture of various linear terms, including energy, obstacle regions, distance, time and excess speed. Potential fields have also been used for underwater path planning in [32] with a cost function measuring the total drag experienced by the vehicle, total travel time and any obstacles in the field. After generating a feasible set of tracks, an optimization is performed on these tracks. This algorithm has the advantage of being inexpensive, thus allowing for easy real-time computations to adapt the vehicle path. However, it has the drawback of producing locally optimal solutions. Another problem with potential field methods is their adaptation to dynamic ocean currents. It is very inefficient to re-compute the potential field for the whole map for each time instant.

- **Rapidly-exploring Random Trees (RRT)**

Rapidly-exploring Random Trees (RRT) have also been used to solve the path planning problem. RRT incrementally grow a tree to explore the space until the tree branch reaches the goal location [43]. A number of extensions like the Dynamic RRT [108]and Multi-particle RRT [44], have been made to facilitate robot navigation in

dynamic and uncertain environments. RRT has been applied to both AUVs [109] and gliders [51] path planning in dynamic flow-fields. It is interesting to note that the RRT growth is inherently biased in the direction of ocean currents. A comparison of RRT and grid-based method is also presented in [51]. The RRT algorithm is very fast and effective to produce collision free paths to problems with many-dimensional configuration spaces, but it does not assure optimality and the paths found often requires further refinement.

- **Evolutionary algorithms**

Evolutionary algorithms are another technique used for path planning. The genetic algorithm (GA) [50, 110] and the particle swarm optimization (PSO) [111, 112] are two well-known forms of evolutionary algorithms that are generally recognized to be effective optimization techniques for solving path planning problems. GA and PSO algorithms are similar in the sense that they are population-based search scheme and that they all depend on information sharing among their population members to enhance their search processes using a combination of deterministic and probabilistic rules. However, PSO has its salient characteristics different from GA. PSO is a stochastic evolutionary algorithm that does not incorporate survival of the fittest, and there is no conventional evolutionary operators such as crossover and mutation. For PSO, all particles are retained through the course of the run and each particle adjusts its searching in the space in terms of its own searching experience and its companions' searching experience. A detailed comparison between PSO and GA can be found in [113]. Quantum-behaved particle swarm optimization (QPSO) is recognized as an improved version of the original PSO. It differs in that QPSO assumes that every particle in the swarm has quantum behaviour instead of using the conventional position and velocity update rules employed in PSO. Fu [114] applied the QPSO for path planning and showed that it has superior performance compared to the standard PSO and GA algorithms. However, Fu's work is focus on UAV-oriented applications, it do not consider ocean currents information, nor the effects of currents on the AUV completing the mission.

Planning for AUVs operating in a large geographical area is a typical large-scale optimization problem. Evolutionary computation techniques have been proven to be

an effective way of dealing with NP-hard problem. Their computational cost grows linearly with the number of vehicles and geometrically with spatial dimensions. Also, they can escape from local minima. Evolutionary algorithms are population based optimization techniques, they search for a solution in parallel, which allows them to be implemented on parallel machines to achieve super linear speed-up with the number of processors [112]. Their drawback is that they may converge to a suboptimal solution within a finite time.

2.3 Problem Formulation

The objective of the AUV path planning system is to find the optimal paths $\mathbb{p}^{\dagger}$ among the set of all feasible paths $\mathcal{p}$ for the AUV to travel through the variable ocean environment and arrive at the target point with minimization of time usage T. The ocean environment V_c is modelled as a time varying current field occupied with obstacles O and terrain $\mathbb{T}$, the position of which may be dynamic and uncertain (see Subsection 2.3.1).

Consider an AUV with initial condition $\mathcal{A}_0 = (r_0, w_0)$, where $r_0 = [x_0, y_0, z_0]$ defines vehicle's position in a spherical coordinate frame, and $w_0 = [\phi_0, \theta_0, \psi_0]$ defines vehicle's Euler angles roll, pitch and yaw. The AUV arrive at the target point with the final condition $\mathcal{A}_G = (r_G, w_G)$. The AUV has constant water referenced speed $\mathcal{V}$. The resultant ground velocity of the vehicle V_a is resolved using the water-referenced speed of the vehicle $\mathcal{V}$ in the geographical frame and the velocity of the water current V_c (see Subsection 2.3.2).

In this study, the potential AUV trajectories are represented by a sequence of points along the path $\mathbb{p} = \{\wp_1, \wp_2, \dots, \wp_\lambda\}$, where λ is the number of these points. The travel time T along a given path $\mathbb{p}$ is the sum of time $t(\wp_i)$ required to cover each of these segments constituting the path.

Therefore the path planning problem is formulated as the following optimization problem:

$$\mathbb{p}^{\dagger} = \frac{\text{argmin}}{\mathbb{p} \in p} \, F(\mathcal{A}, V_c, O, \mathbb{T}, \mathcal{V}),$$

$$\text{with} \quad F = T = \sum_{1}^{\lambda} t(\wp_i)$$

$$\text{s.t.} \quad M(\eta, \dot{\eta})=0,$$

$$\wp_1 = \mathcal{A}_0, \wp_\lambda = \mathcal{A}_{\mathcal{G}}$$

$$\forall \, i \in \{0, \dots, \lambda\} \quad \wp_i(t_i) \notin O(t_i) \cup \mathbb{T}$$

(2-1)

where $M(\eta, \dot{\eta})$ is the vehicle's kinematic model.

2.3.1 Ocean Environment

The marine environment poses a rich field of challenges for AUV path planning systems, such as ocean currents, irregularly shaped terrains and uncertain obstacles, and potentially dynamic rendezvous targets. The currents model is described in Subsection 2.3.1-a. The models of obstacles with uncertainty in position and of dynamic targets are described in 2.3.1-b and c, respectively.

a. Current Model

This subsection gives an overview of the ocean model products used for path planning. Although most test scenarios in this thesis use analytic ocean model equations to generate the currents field in our experiments, real currents data from Bluelink products is also integrated. They constitute a complementary contribution of this work, being useful to evaluate the algorithms in realistic maritime environments.

- **Predictive Ocean Model**

Oceanographic environmental information can be obtained from remote observations, such as High Frequency (HF) radar surface current measurements and satellite observations, or from in-situ moorings, or from numerical forecast models. An example of a predictive tool is the open-source Regional Ocean Model System (ROMS) which is widely accepted and supported throughout the oceanographic and modelling communities. ROMS was developed to study ocean processes along the western U.S. coast with increasing resolution ranging from 15 km down to 1 km resolutions [52], an example of the currents profile generated by ROMS as shown in Figure 2.1 . Bluelink is another example of another predictive ocean model that is capable of generate high-resolution, forecast out to seven days for the coastal and

continental shelf regions around Australia [115, 116]. An example of the currents profile generated by Bluelink is shown in Figure 2.2.
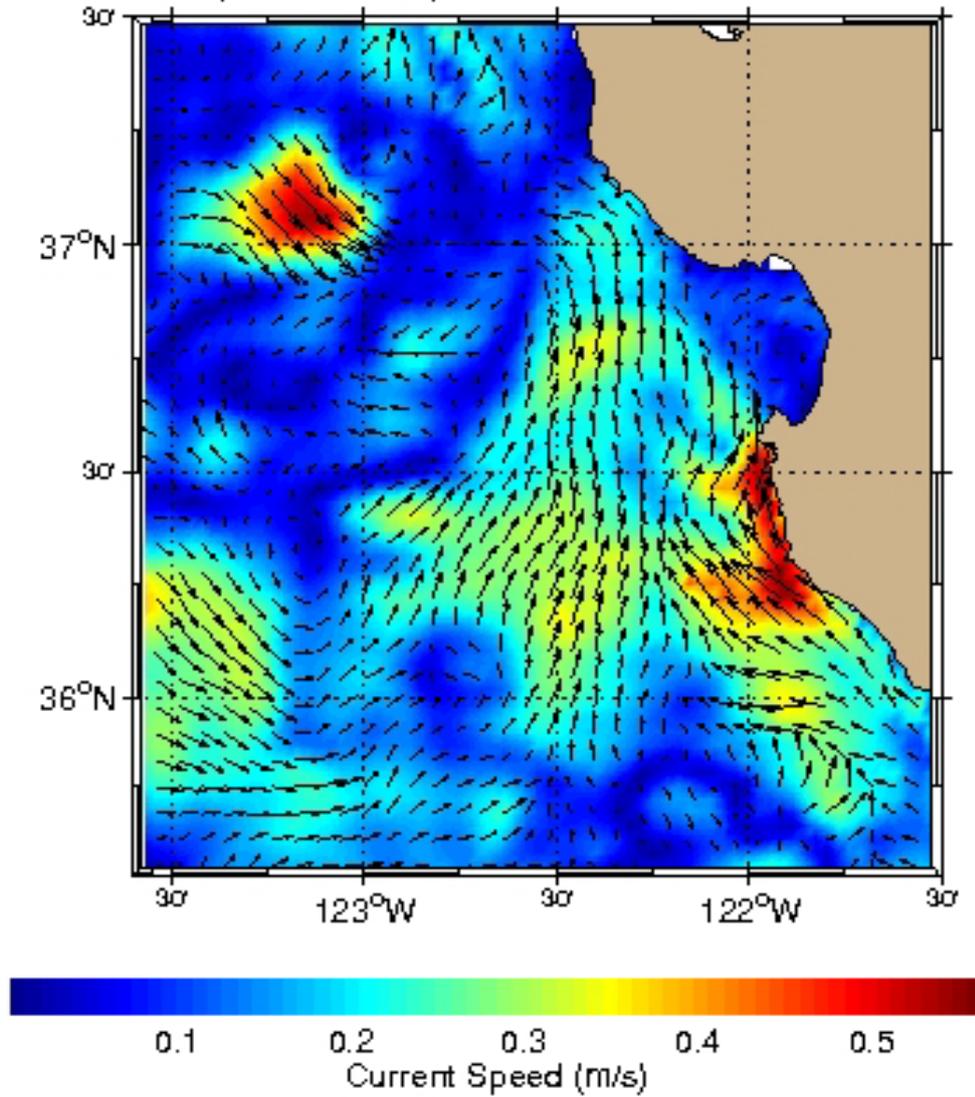
Figure 2.1 Example of currents profile generated by ROMS
(http://www.cencoos.org/sections/models/roms/ca/nowcast/)
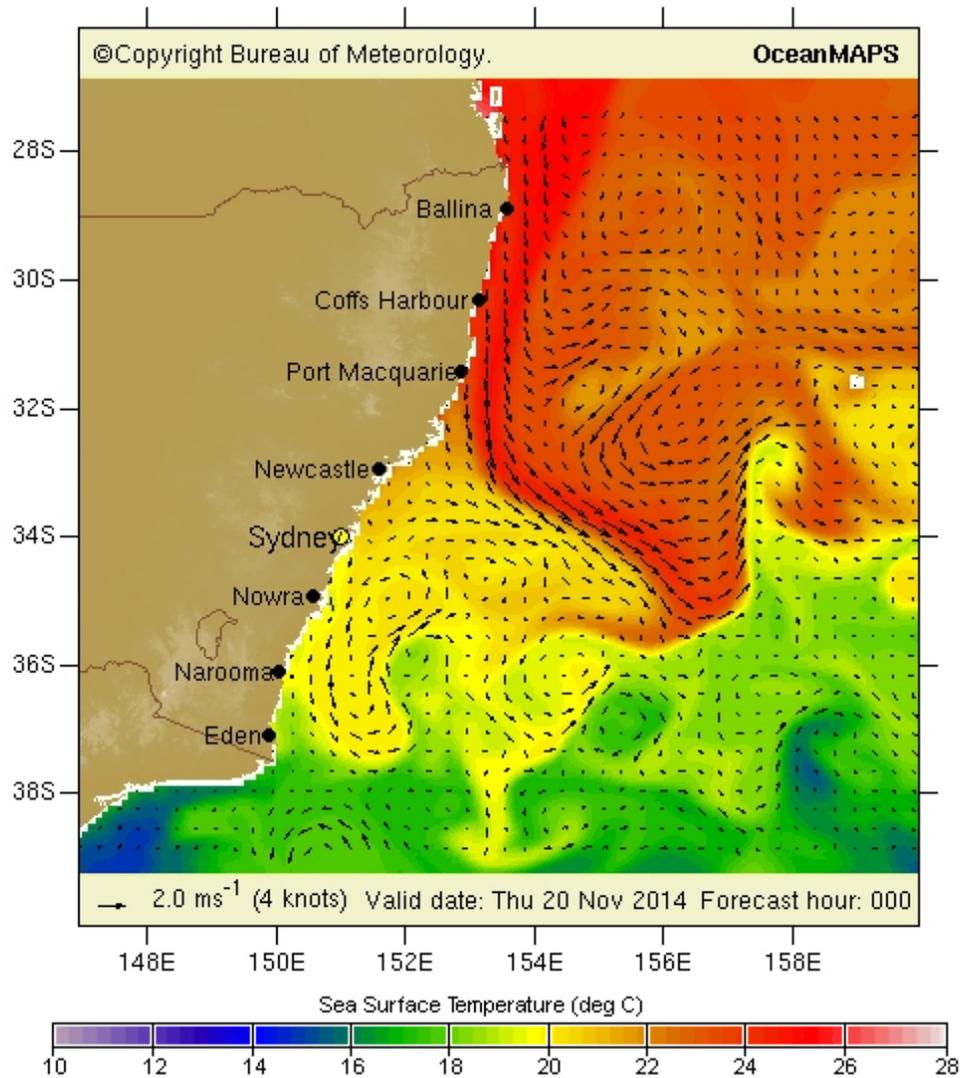
Figure 2.2 Example of currents profile generated by Bluelink
(http://www.bom.gov.au/oceanography/forecasts/idyoc14.shtml?region=14&forecast)

- **Analytical Ocean Model**

Apart from the ocean currents provided by ROMS for a particular region of the oceans, it is also possible to generate these currents field using analytic equations. In this subsection, a procedure is described to generate realistic ocean currents synthetically. The basis of the method is to represent ocean currents as eddies. This method allows to model areas with constant ocean currents in a given direction, sinks or sources, circular or elliptical eddies, and spiral vortex structures. All these forms of structures can be modelled to some extend with analytic equation as shown in the following. Furthermore, the space can be tile with multiple instances of the Equation

(2-2).

In previous work, a synthetic 2D turbulent ocean currents field $V_c = (u_c, v_c)$ was estimated by superposition of multiple viscous Lamb vortices [117].

$$V_c = f(\mathbb{R}^o, \eta, \zeta)$$

$$u_c(\mathbb{R}) = -\eta \frac{y - y_0}{2\pi(\mathbb{R} - \mathbb{R}^o)^2}\left[1 - e^{-\left(\frac{(\mathbb{R} - \mathbb{R}^o)^2}{\zeta^2}\right)}\right],$$

$$v_c(\mathbb{R}) = \eta \frac{x - x_0}{2\pi(\mathbb{R} - \mathbb{R}^o)^2}\left[1 - e^{-\left(\frac{(\mathbb{R} - \mathbb{R}^o)^2}{\zeta^2}\right)}\right],$$

(2-2)

where $\mathbb{R} = \begin{bmatrix} x \\ y \end{bmatrix}$ represents the 2-D spatial domain, $\mathbb{R}^o = \begin{bmatrix} x_o \\ y_o \end{bmatrix}$ is the centre of the vortex and η, ζ are parameters that control the radius and strength of the vortex. Since ocean environments typically have relatively weak vertical motions, a parameter γ is applied to scale the vertical profile from the horizontal profile of the currents field.

The 3D ocean environments can be approximated by a layered structure in which each layer is defined similarly to the 2D case but with a well-defined vertical profile [50]. The vertical profile of the 3D turbulent ocean environments $V_c(u_c, v_c, w_c)$ is estimated by a probability density function of the multivariate normal distribution $N(\mathbb{R}_o, \Sigma_w)$, which is given by

$$w_c(\mathbb{R}) = \gamma\zeta \frac{1}{\sqrt{\det(2\pi\Sigma_w)}} \exp\left[-\frac{1}{2}(\mathbb{R} - \mathbb{R}^o)^T \Sigma_w^{-1}(\mathbb{R} - \mathbb{R}^o)\right],$$

$$\text{with } \Sigma_w = \begin{bmatrix} \eta & 0 \\ 0 & \eta \end{bmatrix}$$

(2-3)

where Σ_w is a covariance matrix related to the radius of the convex.

The continuous time varying ocean environment can be estimated by recursive application of Gaussian noise to the three parameters $\mathbb{R}^o, \eta$ and ζ. The random walk dynamic equations of these parameters are therefore:

$$\mathbb{R}_i^o = A\mathbb{R}_{i-1}^o + BX_{i-1}^{\mathbb{R}x} + CX_{i-1}^{\mathbb{R}y} \qquad (2\text{-}4)$$

$$\eta_i = A\eta_{i-1} + BX_{i-1}^{\eta} \qquad (2\text{-}5)$$

$$\zeta_i = A\zeta_{i-1} + BX_{i-1}^{\zeta}, \qquad (2\text{-}6)$$

where $X_{i-1}^{\mathbb{R}x} \sim N(0, \sigma_{\mathbb{R}x})$, $X_{i-1}^{\mathbb{R}y} \sim N(0, \sigma_{\mathbb{R}y})$, $X_{i-1}^{\eta} \sim N(0, \sigma_{\eta})$ and $X_{i-1}^{\zeta} \sim N(0, \sigma_{\zeta})$ are Gaussian. The parameter metrics are given by;

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad , \quad B = \begin{bmatrix} \Delta_t \\ 0 \end{bmatrix} \quad , \quad C = \begin{bmatrix} 0 \\ \Delta_t \end{bmatrix} \qquad (2\text{-}7)$$

The current field is updated at every Δ_t value of time.

b. Obstacle Model

In practice, knowledge of an obstacle's position is usually not perfect. In this study, the position uncertainty is modelled with independent Gaussian distribution $X_o \sim N(\mu_o, \sigma_o)$. The modelling also takes into account that in the case of no pertinent observations, the position uncertainty grows with time travelled [118], so that σ_o grows linearly with time. Additionally, two forms of obstacle behaviour are considered here: quasi-static obstacles suspended in the water, and obstacles moving under their own power through the water in a constant direction. For quasi-static obstacles, their centres of position uncertainty, μ_o, will remain at their initial locations, whereas for obstacles moving at a constant speed and direction, their centres will change accordingly.

Throughout this thesis, the obstacle model is assumed to be a linear, discrete-time system defined as:

$$O_i = \mathbb{F}_{i-1}^o(O_{i-1}, X_{i-1}, d_U), \qquad (2\text{-}8)$$

where O_{i-1} is the state of the system, $X_{i-1} \sim N(0, \sigma_O)$ is the Gaussian disturbance resulting in a random walk dynamic model, and this disturbance is independent from the disturbances of previous state: $X_{0 \sim i-2}$, and d_U is the uncertainty growing rate of time travelled. The state O_i is defined as:

$$O_i = \begin{bmatrix} O_P \\ O_V \\ O_U \end{bmatrix} \quad \begin{matrix} -\text{position} \\ -\text{velocity} \\ -\text{uncertainty} \end{matrix} \qquad (2\text{-}9)$$

The equation of this dynamic object with uncertainty is therefore:

$$O_i = H_O O_{i-1} + Z_O X_{i-1} + L_O d_U \qquad (2\text{-}10)$$

The parameter matrices are written as:

$$H_O = \begin{bmatrix} 1 & \Delta_t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad Z_O = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad L_O = \begin{bmatrix} 0 \\ 0 \\ \Delta_t \end{bmatrix} \qquad (2\text{-}11)$$

c. Moving Target

Similar to the model of obstacle, the target model is also assumed to be a linear,

discrete-time system defined as:

$$\mathbb{G}_i = \mathbb{F}^{\mathbb{G}}_{i-1}(\mathbb{G}_{i-1}, X_{i-1}, d_{\text{a}}) \tag{2-12}$$

where $\mathbb{G}_{i-1}$ is the state of the system, $X_{i-1} \sim N(0, \sigma_{\mathbb{G}})$ is the Gaussian disturbance resulting in a random walk dynamic model, and this disturbance is independent from the disturbances of previous state: $\mathbb{G}_{0 \sim i-2}$, and d_{a} is the rate of angle growth with time travelled. The state $\mathbb{G}_i$ is defined as:

$$\mathbb{G}_i = \begin{bmatrix} \mathbb{G}_P \\ \mathbb{G}_V \\ \mathbb{G}_{\text{a}} \end{bmatrix} \quad \begin{matrix} -\text{position} \\ -\text{velocity} \\ -\text{angle} \end{matrix} \tag{2-13}$$

The state representation of this dynamic object with uncertainty is therefore:

$$\mathbb{G}_i = H_{\mathbb{G}}\mathbb{G}_{i-1} + Z_{\mathbb{G}}X_{i-1} + L_{\text{a}}d_{\text{a}} \tag{2-14}$$

The parameter matrices are written as:

$$H_{\mathbb{G}} = \begin{bmatrix} 1 & \Delta_t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Z_{\mathbb{G}} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad L_{\text{a}} = \begin{bmatrix} 0 \\ 0 \\ \Delta_t \end{bmatrix} \tag{2-15}$$

2.3.2 Travel Time Evaluation

As previously mentioned, the travel time T along a given path $\mathbb{p}$ is the sum of time $t(\wp_i)$ required to cover each of these segments constituting the path.

$$T = \sum_1^\lambda t(\wp_i) = \sum_1^\lambda |(\wp_{i+1}) - (\wp_i)| / |V_a| . \tag{2-16}$$

where $\wp_{i+1}$, $\wp_i$ are two adjacent points along the path. In previous work [119], the vehicle's thrust power is assumed to be constant, equivalently, the vehicle has constant water-referenced speed. The resultant ground velocity of the vehicle V_a is resolved in a 2D horizon using the water-referenced speed of the vehicle in the geographical frame $\mathcal{V}$ and the velocity of the water current V_c. With reference to Figure 2.3, V_a is the resultant ground velocity of the vehicle which is resolved using the water-referenced speed of the vehicle in the geographical frame $\mathcal{V}$ and the velocity of the water current V_c - it can be written as:

$$V_{ay} = |\mathcal{V}| \sin \psi_r + |V_c| \sin \psi_c. \tag{2-17}$$

$$V_{ax} = |\mathcal{V}| \cos \psi_r + |V_c| \cos \psi_c. \tag{2-18}$$

where V_{ay}, V_{ax} are the north and east velocity components of the vehicle, respectively. The resultant ground velocity should orient along the segment $\wp_{i-1}\wp_i$ in the direction of the desired motion of the vehicle - it is defined as:

$$\frac{V_{ay}}{V_{ax}} = tan\psi_a = \frac{|y_i - y_{i-1}|}{|x_i - x_{i-1}|} ..$$ (2-19)

V_a, ψ_a are obtained by solving Eqs. (2-17)~(2-19) as simultaneous equations.
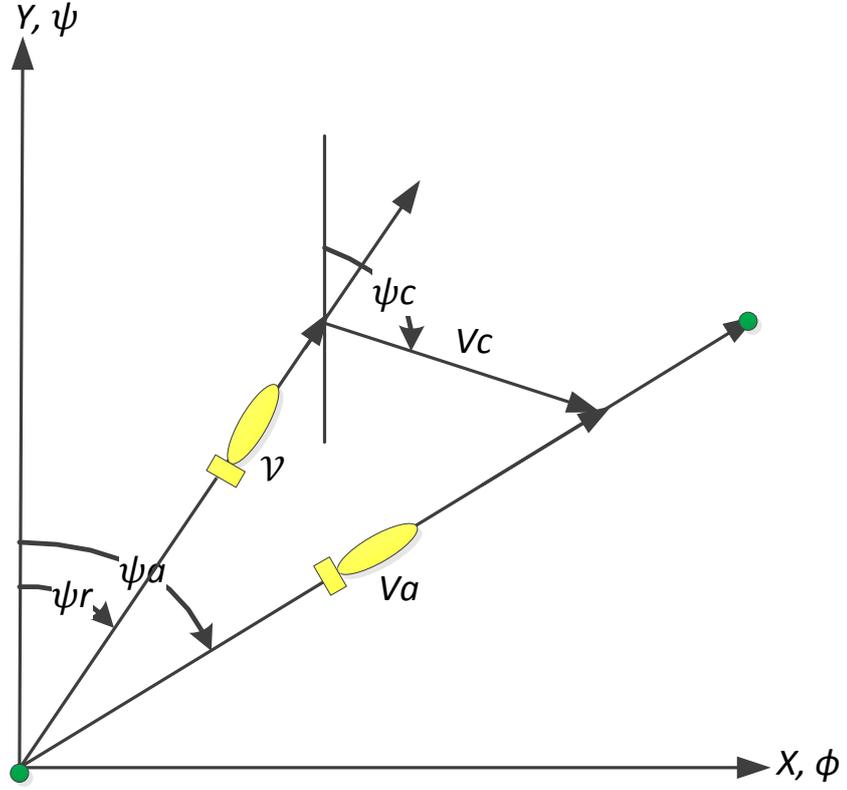
Figure 2.3 Resultant ground velocity of the vehicle under influence of the ocean current in 2D space.

V_a could also be resolved in 3D space using a similar approach, as shown in Figure 2.4.

$$V_{ax} = |\mathcal{V}|cos\theta_r \, cos\,\psi_r + |V_c| \, cos\theta_c cos\,\psi_c \,,$$ (2-20)

$$V_{ay} = |\mathcal{V}| \, cos\theta_r sin\,\psi_r + |V_c| \, cos\theta_c sin\,\psi_c \,,$$ (2-21)

$$V_{az} = |\mathcal{V}|sin\theta_r + |V_c|sin\theta_c \,.$$ (2-22)

where V_{ay}, V_{ax} V_{az} are the O-XYZ velocity components of the vehicle respectively.
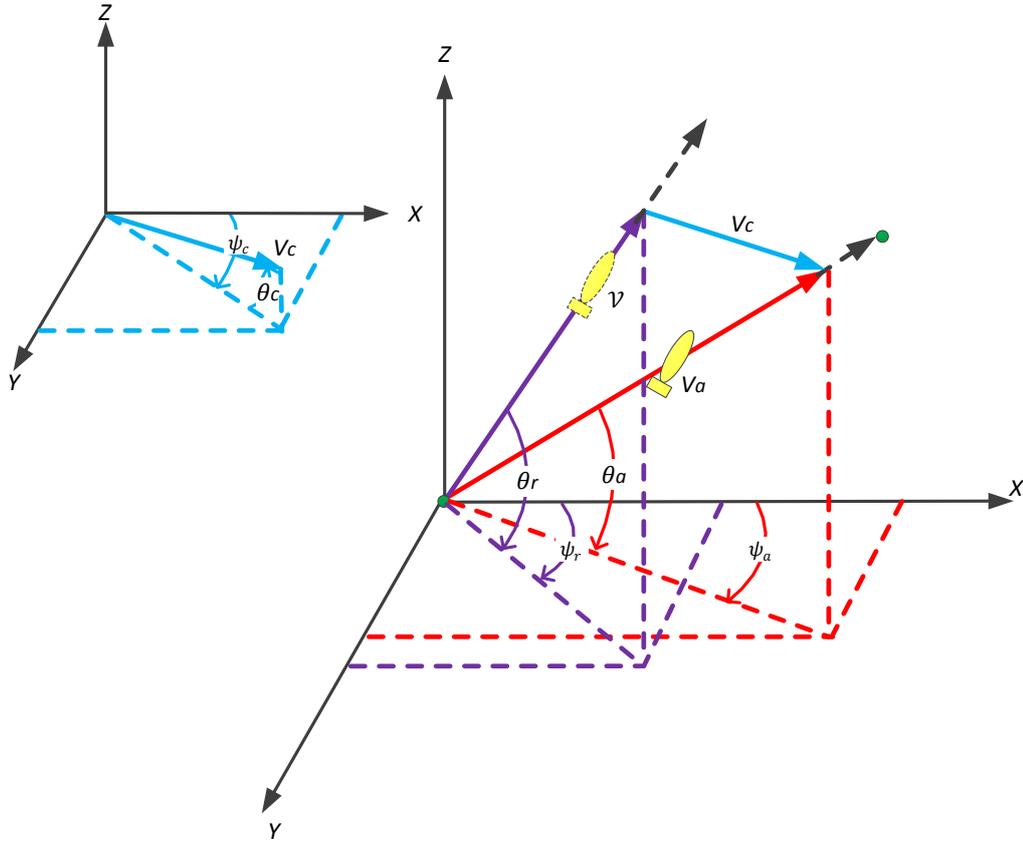
Figure 2.4 Resultant ground relative velocity of the vehicle under the influence of ocean currents in 3D space.

The resultant ground velocity should orient along the segment $\wp_{i-1}\wp_i$ in accordance with the vehicle's desired motion direction:

$$\frac{V_{ay}}{V_{ax}} = \frac{|y_i - y_{i-1}|}{|x_i - x_{i-1}|}, \tag{2-23}$$

$$\frac{V_{az}}{V_{ax}} = \frac{|z_i - z_{i-1}|}{|x_i - x_{i-1}|}, \tag{2-24}$$

where V_{ax}, V_{ay}, V_{az}, ψ_r, θ_r can be obtained by solving the simultaneous set of Equations (2-20)~(2-24).

2.4 Path Planners

In this section, path planners based on A*, RRT and evolutionary algorithms (GA, PSO and QPSO) are developed.

2.4.1 A* Based Path Planner

A* graph-based path planning algorithms [101] is shown in Algorithm 2.1.

Algorithm 2.1 A* Based Path Planner ($\wp_1, \wp_\lambda$)

---

1. openset = set containing the initial node

2. came_from = the empty map

3. $g_{cost}(\wp_1) = 0$

4. $h_{cost}(\wp_1) = \mathcal{H}_{cost}(\wp_1, \wp_\lambda)$

5. $f_{cost}(\wp_1) = h_{cost}(\wp_1)$

6. **while** openset is not empty

7. $\quad$ $\wp_i$ = the node in openset having the lowest f_{cost} value

8. $\quad$ **if** $\wp_i = \wp_\lambda$

9. $\quad\quad$ **return** reconstruct_path(came_from, $\wp_\lambda$)

10. $\quad\quad$ **remove** $\wp_i$ from openset

11. $\quad\quad$ **for** each $\wp_{i+1}$ in neighbor_nodes of $\wp_i$

12. $\quad\quad\quad$ $\mathcal{T}g_{cost} = g_{cost}(\wp_i) + t_{cost}(\wp_i, \wp_{i+1})$

13. $\quad\quad\quad$ **if** $g_{cost}(\wp_{i+1})$ is not set or $\mathcal{T}g_{cost} < g_{cost}(\wp_\lambda)$

14. $\quad\quad\quad\quad$ add $\wp_{i+1}$ to openset

15. $\quad\quad\quad\quad$ came_from($\wp_{i+1}$) = $\wp_i$

16. $\quad\quad\quad\quad$ $g_{cost}(\wp_{i+1}) = \mathcal{T}g_{cost}$

17. $\quad\quad\quad\quad$ $h_{cost}(\wp_{i+1}) = \mathcal{H}_{cost}(\wp_{i+1}, \wp_\lambda)$

18. $\quad\quad\quad\quad$ $f_{cost}(\wp_{i+1}) = g_{cost}(\wp_{i+1}) + h_{cost}(\wp_{i+1})$

19. $\quad\quad\quad$ **end if**

20. $\quad\quad$ **end for**

21. **end while**

22. **return** failure

---

function reconstruct_path (came_from, $\wp_i$)

---

1. **if** $\wp_i$ is in came_from

2. $\quad$ p = reconstruct_path(came_from, came_from[$\wp_i$])

3. $\quad$ **return** (p + $\wp_i$)

4. **else**

5. $\quad$ **return** $\wp_i$

6. **end if**

Originally, it was applied to solve the shortest path problem. More precisely, A* solves the problem of finding the shortest path from a single start node $\wp_1$ to a single target node $\wp_\lambda$. To apply this method to solve path planning problems with minimum time/energy consumption for AUV oriented application, it requires some adaptation of the cost function to deal with a time/energy cost instead of a distance cost. In the following of this subsection, the model of the search space, the cost function and the heuristic, as well as some implementation details are included.

The algorithm minimizes the cost to the target, which in this case consists of a temporal cost based on the maximum speed an AUV can achieve within the ocean current field and the distance between two locations, represented by the nodes of the gridded search graph.

A* updates the g-value and the reconstruct path of an unexpanded neighbour nodes $\wp_{i+1}$ of node $\wp_i$ by considering the time consumption $g_{cost}(\wp_i)$ of the path from $\wp_1$ and the time consumption $t_{cost}(\wp_i, \wp_{i+1})$ from $\wp_i$ to $\wp_{i+1}$, resulting in $g_{cost}(\wp_i) + t_{cost}(\wp_i, \wp_{i+1})$. It updates the g-value and reconstruct path of $\wp_{i+1}$ if the new path distance $g_{cost}(\wp_{i+1})$ is shorter than the shortest path from start to $\wp_{i+1}$ found so far (see line 13 of Algorithm 2.1).

$$t_{cost}(\wp_i, \wp_{i+1}) = \frac{|\wp_{i+1} - \wp_i|}{|V_a^i|}. \tag{2-25}$$

Where V_a^i is the resultant ground velocity of the vehicle which can be resolved in using the water-referenced speed of the vehicle in the geographical frame V_r and the velocity of the current V_c^i at the node $\wp_i$.

To compute the heuristic we estimate the maximum velocity V_c^{max} of the ocean currents through the path P the AUV will follow. Since P is unknown in advance, V_c^{max} is estimated by a domain spatial and temporally centred at the current location. From the given currents field V_c and the AUV nominal speed V_r, an underestimated temporal cost to reach the target from the current location of node $\wp_i$ can be computed, using the straight line path directly goes to the target.

$$h_{cost}(\wp_i) = \frac{\|\wp_\lambda - \wp_i\|}{V_r + V_c^{max}}. \tag{2-26}$$

which is a conservative but admissible heuristic function. λ is the number of nodes along the straight line connection to the target.

2.4.2 RRT Based Path Planner

The RRT technique [99] is based on incremental construction of search trees that attempt to rapidly and uniformly explore the state space. Contrary to other algorithms which focus on finding an optimal path, RRT is more likely to find feasible trajectories that are not necessarily optimal. Its main advantage is that it produces a solution trajectory with relatively less sampling than other methods, and consequently much faster to run for a solution. Indeed, this performance gain is enabled in part at the cost of optimality.

In 2.4.3 the basic RRT algorithm is presented. Consider a vehicle starting at point $\wp_1$ in the 2D plane. The algorithm started with entering this first state $\wp_1$ as a vertex in the tree. The tree is then expanded as follows. A random state $\wp_{rand}$ is then selected based on a uniform distribution within the vehicle's configuration space.

Algorithm 2.2 RRT Based Path Planner ($\wp_1, \wp_\lambda$, g)

1. Input: Start $\wp_1$ and target $\wp_\lambda$ location, and number of iterations g.

2. add start location $\mathcal{G}.\textbf{init}(\wp_1)$

3. **for all** 1 to g **do**

4.   new random location: $\wp_{rand} = \textbf{rand}$

5.   nearest neighbour location in the *swath* $\boldsymbol{S}(\mathcal{G})$: $\wp_{near} = \boldsymbol{NN}(\boldsymbol{S}(\mathcal{G}), \wp_{near})$

6.   new location: $\wp_{new} = \textbf{new}(\wp_{near}, \Delta_t)$

7. $\mathcal{G}.\textbf{add vertex}(\wp_{new})$

8. $\mathcal{G}.\textbf{add edge}(\wp_{near}, \wp_{new})$

9. **end for**

10. **return** $\mathcal{G}$

The nearest neighbour to $\wp_{rand}$ within the tree is found to be $\wp_i$, and the vehicle moves from $\wp_i$ towards $\wp_{rand}$ over a fixed time-step Δ_t. This process is then repeated to grow the tree.

The next state of the vehicle can then be found by

$$\wp_{i+1} = \wp_i + V_a^i \Delta_t, \tag{2-27}$$

where V_a^i is the resultant ground velocity of the vehicle in the direction pointing towards $\wp_{rand}$, this velocity can be resolved from the water-referenced speed of the

vehicle in the geographical frame V_r and the velocity of the water current V_c^i.

Heuristically biasing the growth of the RRT can improve the result, in fact, the implementation here is based on the work of Rao and Williams [51]. Sertac and Emilio Frazzoli [120] also presented a modified version of the RRT algorithm - RRT*, which reduce the memory required for storing the tree. RRT* is designed such that it runs until the tree has grown to a predefined number of nodes, following which a weak node is removed whenever a high performance node is added. RRT* has been implemented here to compare its performance with other existing and proposed techniques.

2.4.3 Spline based Evolutionary Path Planners

Existing path planners represent the mission path, either as a sequence of straight line segments connecting the monotonic path nodes [50], or as a sequence of B-Spline curve segments, generated from a set of control points, connecting the initial and destination points [110]. An evolutionary-algorithm based framework is then used to optimize the path based on the object function. For an AUV operating in a turbulent ocean environment, the effect of currents and the vehicle dynamics impose specific constraints for the smoothness and the curvature of the trajectory, thus in this work, B-Spline curves are used to define the AUV path (see Subsection 2.4.3-a). Several evolutionary path planners are developed in Subsection 2.4.3-b.

a. B-Spline Path Formation

B-Splines are generated from a set of control points $P = [p_1, ..., p_i, ..., p_m]$ that are located in the environment. The following defines the B-Spline by piecewise polynomials, and describes the method to obtain discretised points from B-Spline by uniform subdivision.

- B-Spline

The B-Spline curves are defined by the piecewise polynomials [121]:

$$p(t) = \sum_{i=0}^{n} p_i N_{i,\varepsilon}(t), \tag{2-28}$$

where $N_{i,\varepsilon}(t)$ is the basic function of the curve and ε is the order of the curve, which

is associated with the smoothness of the curve.

$$N_{i,\varepsilon}(t) \equiv N_{i,\varepsilon}(f,t) = \frac{1}{f} \int_{t-f/2}^{t+f/2} N_{i,\varepsilon-1}(f,t)\, dx, \tag{2-29}$$

$$= \frac{1}{h} N_{0,1}(f,t) * N_{i,\varepsilon-1}(f,t), \tag{2-30}$$

where $*$ represents convolution, f is the step size.

$$N_{i,1}(f,t) = \begin{cases} 1, & -f/2 \leq t - fi \leq f/2 \\ 0, & otherwise \end{cases}. \tag{2-31}$$

- Discretised Points from B-Spline Curve

The B-Spline curve is discretized by uniform subdivision [56, 121]. Let $\mathbb{p}$ be the uniform B-Spline subdivision of order $\varepsilon \geq 2$ and step size 1 to the control points $P = [p_1, \dots, p_i, \dots, p_m], n \geq \varepsilon$. That is,

$$\mathbb{p} = \sum_{i=0}^{n} p_i N_{i,\varepsilon}(1,x), \tag{2-32}$$

where $N_{i,\varepsilon}(1,x)$ is given by Eq. 2-17 and $x \in [(\varepsilon - 2)/2, n - (\varepsilon - 2)/2]$. Then

$$\mathbb{p} = \sum_{i=0}^{2n-\varepsilon+2} p_i^\varepsilon N_{i,\varepsilon}(0.5,x), \tag{2-33}$$

where p_i^ε is recursively obtained from

$$p_i^\varepsilon = (p_i^{\varepsilon-1} + p_{i+1}^{\varepsilon-1})/2$$
$$i = 0,1,\dots,2n+\varepsilon+2, \varepsilon > 2, \tag{2-34}$$

and

$$p_i^2 = \begin{cases} p_{\frac{i}{2}} & i\ is\ even \\ \left(p_{\frac{(i-1)}{2}} + p_{\frac{(i+1)}{2}}\right)/2 & i\ is\ odd \end{cases} \quad i = 0,1,..,2n. \tag{2-35}$$

b. Evolutionary planners

GA, PSO and QPSO algorithms all belong to the class of evolutionary computation that use iterative progress, such as growth or development in a population. This population is then selected through a heuristic search to achieve the desired solution. A numerical comparison of the GA, PSO and QPSO for continuous function optimization can be found in Appendix B.

In this Section, the state of the ith individual, swarm or population at iteration t is represented as:

$$P_i(t) = [p_{i1}(t), p_{i2}(t), \dots, p_{im}(t)]. \tag{2-36}$$

where m is the dimension of the problem space (equivalent to the number of path nodes).

- GA Based Path Planner

Error! Reference source not found. provides an overview of the GA based path planner. In this implementation, the initial population is randomly generated where the coordinates of the spline control points form the chromosome genes.

Emulating the evolution process found in nature, children solutions are created from the following operators: A fraction of the individuals in the current generation with the best fitness values are called elite children. These individuals automatically survive to the next generation. A roulette-wheel selection mechanism is applied to stochastically select from one generation to create the basis of the next generation. This ensures that fitter individuals will tend to have a better probability of survival and will go forward to form the pool for crossover. The crossover operator serves to create the children by combining the vectors of their parents and mutating them to introduce random variations to the remaining population. Finally, the parent generation is replaced by the children generation. This evolution cycle continues for a maximum number of generations or until the stop criterion has been met. A detailed implementation of the GA algorithm for path planning can be found in [58].
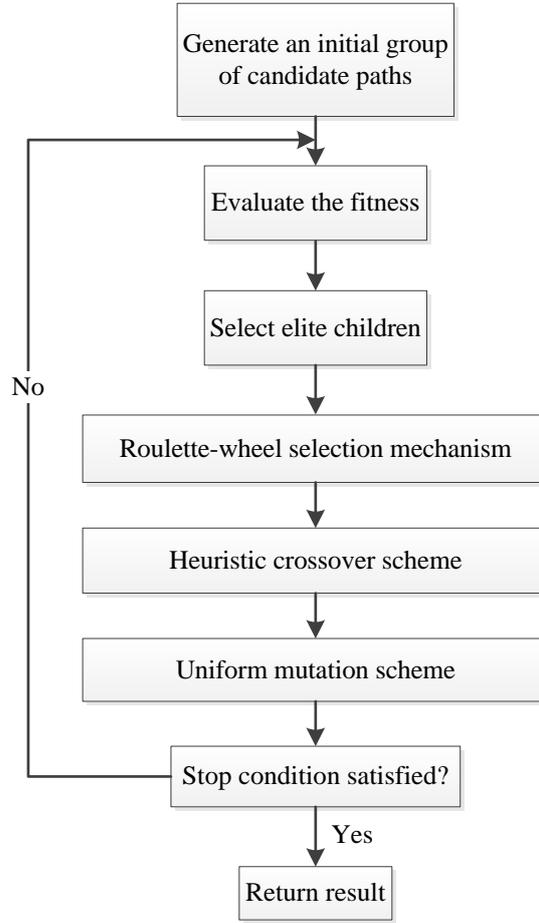
Figure 2.5 Flowchart of the GA based path planner

Selection: A roulette-wheel selection mechanism is applied to stochastically select from one generation to create the mating pool for the next generation. The fitness value $F(P_i(t))$ of each individual is associated with its probability of being selected:

$$Q_i = \frac{F(P(t))}{\sum_{i=1}^{\rho} F(P_i(t))}. \tag{2-37}$$

Crossover: Children are created by combining the vectors of their parents. A heuristic crossover scheme is applied to the selected chromosomes from the previous step. Assuming that $F(p_i(t)) < F(p_j(t))$, the new individual is generated by linear combination of their parents, defined as:

$$p_i'(t) = \alpha p_i(t) + (1 - \alpha)p_j(t), \tag{2-38}$$

$$p_j'(t) = p_i(t), \tag{2-39}$$

where α is a pre-set ratio.

Mutation: Mutation creates a random variation of a single parent. The uniform

mutation scheme is applied to the remaining individuals, for each mutated individual, a proportion ∂_m of genes are randomly selected and replaced by new genes from the constrained bounds. Let U_j and L_j be the upper and lower bounds, $Rand_{ij}^M$ be a random number within the interval $(0,1)$, and $p_{ij}(t)$ be one of the genes selected to be replaced by:

$$p'_{ij}(t) = L_j + Rand_{ij}^M(U_j - L_j). \qquad (2\text{-}40)$$

- PSO Based Path Planner

In PSO [67], each particle adjusts its position in search space according to its own searching experience and companions' searching experience. **Error! Reference source not found.** provides an overview of the PSO based path planner.
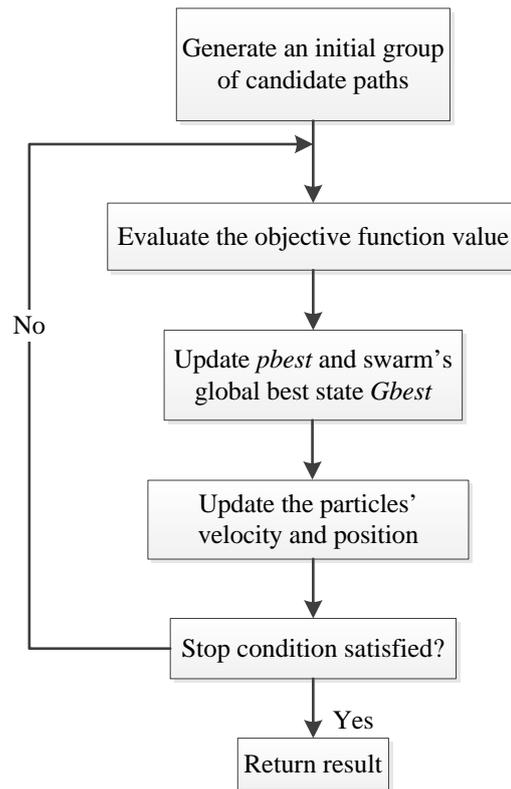
Figure 2.6 Flowchart of the PSO based path planner

In this implementation, the initial population is randomly generated where the coordinates of the spline control points form the particles. The ith particle $P_i = [p_{i1}, p_{i2}, \dots, p_{im}]$ is associated with a velocity vector $V_i = [v_{i1}, v_{i2}, \dots, v_{im}]$. During the search process, every particle keeps track of its own personal best (*Pbest*) position

$\mathcal{P}_{\text{pbest}} = [\mathcal{P}_{i1}, \mathcal{P}_{i2}, \ldots, \mathcal{P}_{im}]$ and the global best (*Gbest*) position $\mathcal{P}_{\text{gbest}} = [\mathcal{P}_{g1}, \mathcal{P}_{g2}, \ldots, \mathcal{P}_{gm}]$ achieved by any particle in the swarm. During the iteration procedure, the velocity and position of the particle are updated according to Eq.(2-41) and (2-42) [122, 123].

Velocity update:

$$v_{i1}(t+1) = \omega v_{i1}(t) + c_1 \text{Rand}_{i1}(t)\left(\mathcal{P}_{i1}(t) - v_{i1}(t)\right) + c_2 \text{Rand}_{i2}(t)(\mathcal{P}_{i2}(t) - v_{i2}(t)), \quad (2\text{-}41)$$

Position update:

$$p_{i1}(t+1) = p_{i1}(t) + v_{i1}(t+1) \qquad (2\text{-}42)$$

where v_{ij}, which is similar to p_{ij}, $\mathcal{P}_{ij}$, and $\mathcal{P}_{ij}$, is the jth dimension of the ith particle's velocity, it is constrained to the interval $[-v_{\min}, v_{\max}]$ avoiding the explosion of the particles. Coefficients Rand_1 and Rand_2 are two pseudorandom scalar values drawn uniformly from the unit interval. The superscript t in Eq. (2-41) denotes the tth iteration. The acceleration coefficients c_1 and c_2 are 2.0 for almost all applications [123]. Factor ω is the inertial weight, this inertial weight plays the role of balancing the global search (large inertial weight) with the local search (small inertial weight). Parsopoulos and Vrahatis [124] pointed out that parameter adaptation during the optimization process could significantly improve the performance of the algorithm. The variation of inertial weight in PSO is analogous to that of temperature in simulated annealing.

- QPSO Path Planner

In QPSO, the particle is assumed to have quantum behaviour instead of using the conventional position and velocity update rules employed in PSO. The particles in the swarm is assumed to be attracted by a quantum potential well centred on its local attractor defined as Equation (2-47), the new stochastic update equation for its position can be found in Equation (2-46). To enhance the global search ability, the QPSO algorithm also introduces the mean best position as defined in Equation (2-43).

The QPSO algorithm samples the new position with a double exponential distribution as defined in Equation (2-48). The QPSO algorithm uses an adaptive strategy to update the particles and has fewer parameters to be adjusted, leading to a

better performance of the algorithm as an overall result compare with the PSO algorithm.

Algorithm 2.3 provides an overview of the iterative QPSO algorithm for path planning. Every particle in the swarm represents a potential path, the parameters of each particle corresponds to the coordinates of control points generating the path. As the QPSO algorithm iterates, every particle is attracted towards its respective local attractor based on the outcome of the particle's individual search as well as the particle swarm's search results.

The detailed implementation of the QPSO algorithm for path planning can be described as follows:

Step 1.   Choose appropriate parameters for the population size, ρ (equivalent to the number of candidate paths), the dimension of the problem space, m (equivalent to the number of control points used to generate the B-Spline path), and the maximum number of iterations, g. Input the ocean field information.

Step 2.   Generate an initial group of particles with random states representing the candidate paths, the state of the ith particle at iteration t is represented as $P_i(t) = [p_{i1}(t), p_{i2}(t), \dots, p_{im}(t)]$. Initialize each particle's current best state, set $\mathcal{P}_i(1) = P_i(1)$.

Step 3.   Set $\mathcal{P}_i(t)$ to be the state with the best fitness found so far at iteration t for the ith particle. Evaluate the mbest, which is defined as the mean of the best states of all particles

Step 4.   Evaluate the cost function, $F(P_i(t))$, of each candidate path (particle) as defined in Equation (2-1).

Step 5.   Compute the ith particle's personal best position ($\mathcal{P}_i$) and the swarm's global best position (G) using (2-44) and (2-45), respectively.

Step 6.   Update the state of the particle in the swarm according to (2-46)~(2-48).

Step 7.   Repeat steps 3 to 6 for g iterations or until the algorithm meets the stop criterion.

Step 8.   Output G as the optimal fitness value and its correlated path as the optimal solution when the loop ends.

Algorithm 2.3 QPSO for Path Planning

Begin Initialize the current state P and the pbest state $\mathcal{P}$ of all particles representing

the candidate paths;

  while the terminate condition is not met **do**

    Compute the mean best state

$$mbest(t) = \sum_{i=1}^{\rho} \frac{\mathcal{P}_i(t)}{\rho}. \tag{2-43}$$

   for t = 1 to g do

     Evaluate the cost function $F(P_i(t))$;

     Update $\mathcal{P}$ and G;

$$\mathcal{P}_i(t) = \begin{cases} \mathcal{P}_i(t-1), & if\ F(P_i(t)) \geq F(\mathcal{P}_i(t-1)) \\ P_i(t), & if\ F(P_i(t)) < F(\mathcal{P}_i(t-1)) \end{cases}, \tag{2-44}$$

$$G(t) = \begin{matrix} arg\ min \\ 1 \leq i \leq \rho \end{matrix} F(\mathcal{P}_i(t)). \tag{2-45}$$

     for i = 1 to ρ **do**

$$\delta_{ij}(t) = Rand_{ij}^{\delta}\mathcal{P}_{ij}(t) + \left(1 - Rand_{ij}^{\delta}\right)G_j(t), \tag{2-46}$$

$$L_{ij}(t) = 2\beta\left|mbest_j(t) - p_{ij}(t)\right|, \tag{2-47}$$

$$p_{ij}(t+1) = \begin{cases} \delta_{ij}(t) + \frac{1}{2}\left(L_{ij}(t)\ln\left(\frac{1}{Rand_{ij}^p}\right)\right), & if\ Rand \geq 0.5 \\ \delta_{ij}(t) - \frac{1}{2}\left(L_{ij}(t)ln(\frac{1}{Rand_{ij}^p})\right), & if\ FRand < 0.5 \end{cases}, \tag{2-48}$$

    end

   end

   Set t = t+1;

  end

end

2.5 Simulations

To evaluate the performance of the A*, RRT and RRT*, GA, PSO and QPSO based path planners, a number of case studies with various mission scenarios are tested in this section.

2.5.1 Simulation Setup

Suppose the currents field to be a square region with a size of 25×25 grid squares and with a resolution of 1km $\times$ 1km for each grid. Two test cases, in the presence of currents field with or without obstacles, respectively, are designed to compare the performance of these algorithms. The obstacles are assumed static, and the location and size of the obstacles present in the search field are known and remain constant over the course of operation. The water-referenced speed of the vehicle in the geographical frame is set at 2 m/s. The starting point is (6, 5) and the goal point is (18, 18).

The common parameters of the GA, PSO and QPSO: population size, particle dimension (equivalent to the number of path nodes), maximum number of iterations, and number of runs are set to 100, 8, 100, and 100, respectively. Meanwhile, each algorithm has its own particular parameters, including the crossover and mutation probability of GA, inertial weight coefficient of PSO, and contraction-expansion coefficient of QPSO. The settings of these private algorithm parameters are given in Table 2.3, which are selected based on the suggestions in other literatures [71, 125-127] where these values have been found, empirically, to provide good performance.

Table 2.3 Parameter Values

Parameters		Value
GA	Selection rate	0.1
	Crossover rate	0.8
	Mutation rate	0.1
PSO	Coefficient ω	$0.9 \rightarrow 0.4$
	c_1 and c_2	2.0
QPSO	Coefficient β	$1 \rightarrow 0.5$

2.5.2 Simulation Results of Path Planners

- A* Based Path Planner

The path generated using A* based path planner for each of the problem instances are shown below in Figure 2.7 (without obstacles) and Figure 2.8 (with obstacles), indicated by the red line connecting the start and goal nodes. The extent of obstacles is marked in black through the corresponding vertices. In both cases the path appears to be bumpy slightly, this effect is results from the fact that only a finite number of angles are representable on the discrete grid. In this case, because of the 8-connected structure of the graph, the vehicle heading deviations are limited to multiples of 45°.
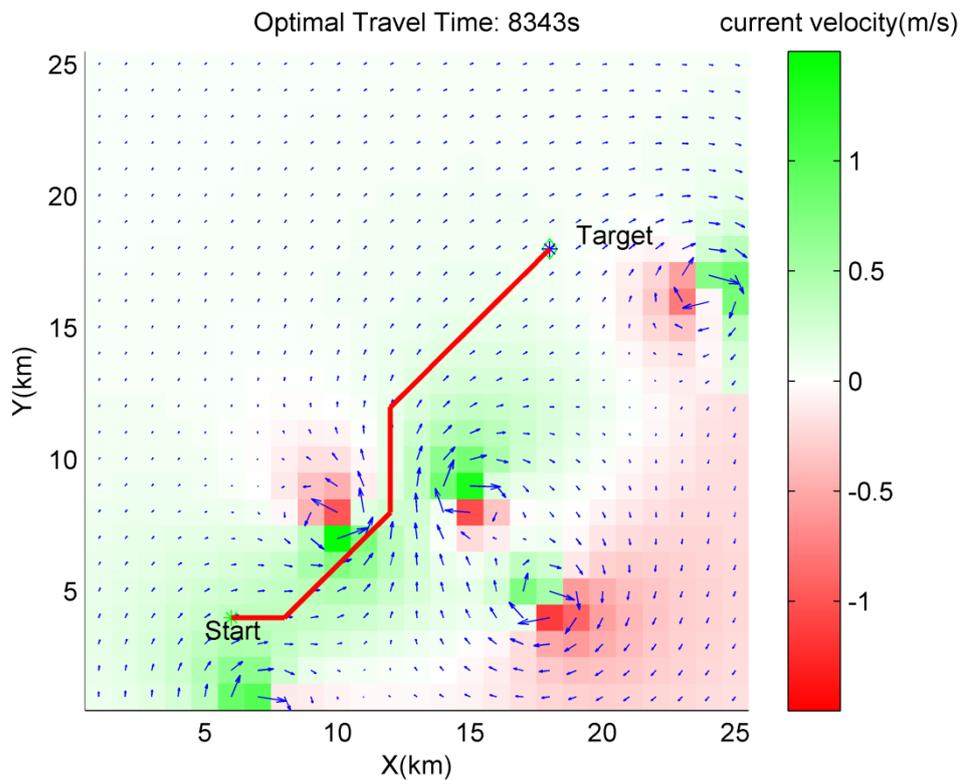
Figure 2.7 Path projections in the 2D current field without obstacles produced by A* path planner.
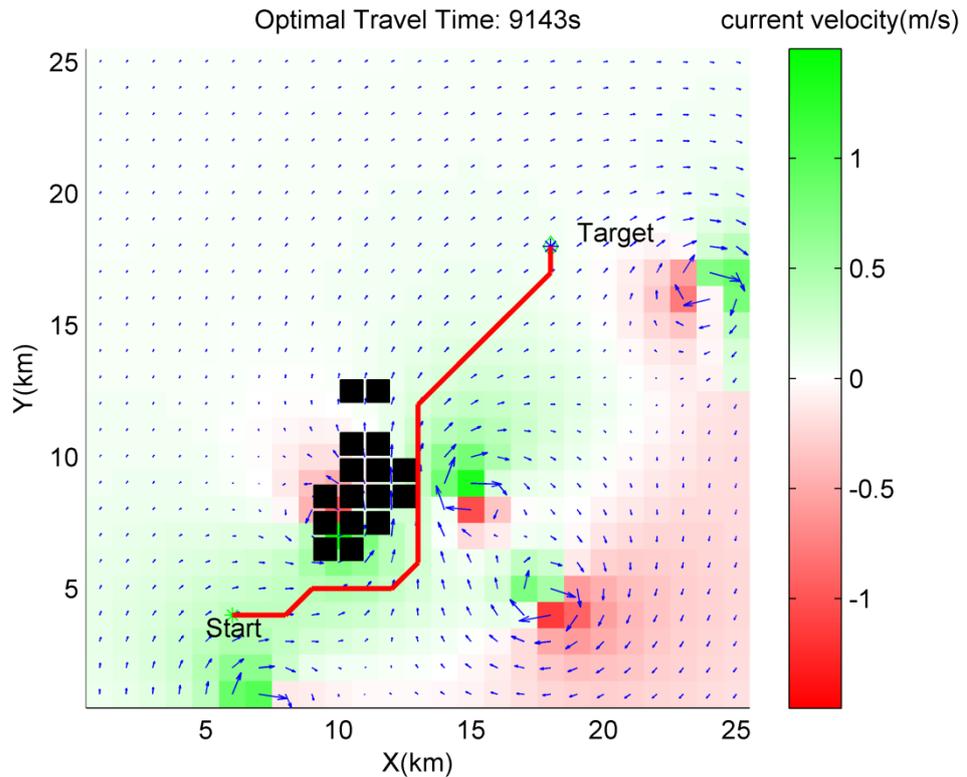
Figure 2.8 Path projections in the 2D current field with obstacles produced by A* path planner.

- RRT/RRT* Path Planner

The RRT and RRT* based path planners are next applied to the same two problem scenarios. The RRT algorithms have been set to run for a fixed 200 iterations. RRT is a probabilistic optimization algorithm, it is susceptible to variation over multiple runs. The inconsistency of this algorithm had been studied in [128], wherein it is suggested to alleviate the inconsistency by finding the best RRT path over multiple runs of the algorithm. Given the long timescale of AUV missions, computation time is not a limiting factor, and the algorithm can easily be run multiple times to yield improved results.

Figure 2.9 shows an example run of the RRT path planner for the problem scenario without obstacles. The black lines denote all RRT trees and the red lines denote the generated path. RRT tries to explore the whole space as shown in Figure 2.9 for the most part. This comes from the fact that the basic RRT algorithm always selects the candidate point randomly which makes the tree growth uniform. It is not an efficient

way to search the whole space as the primary goal of navigation is to find a feasible path between start and goal configurations. Figure 2.10 shows the tree expansion result of the RRT path planner in an environment case with obstacles. Compared with the free space case, the trees grow avoiding the space blocked by obstacles. After 200 iterations, the RRT algorithm returns a totally different trajectory for the case with obstacles compared to the case without obstacle.

Improved results can be obtained by applying the enhanced version of RRT – RRT* [99]. The results for similar scenarios with and without obstacles are shown in Figure 2.11 and Figure 2.12, respectively. These figure illustrate that the RRT* enables a feasible path to be found with less cost and less intrusion into unnecessary areas compared to the basic RRT case as can be seen in Figure 2.9 and Figure 2.10.

It is also easy to observe in Figure 2.9 and Figure 2.10 that despite of the good exploration capabilities of this approach, the solution seem rough and will require further refinement or some sort of smoothing procedure, even for RRT*, the obtained trajectory shown in Figure 2.11 and Figure 2.12 seem not to be very optimal.
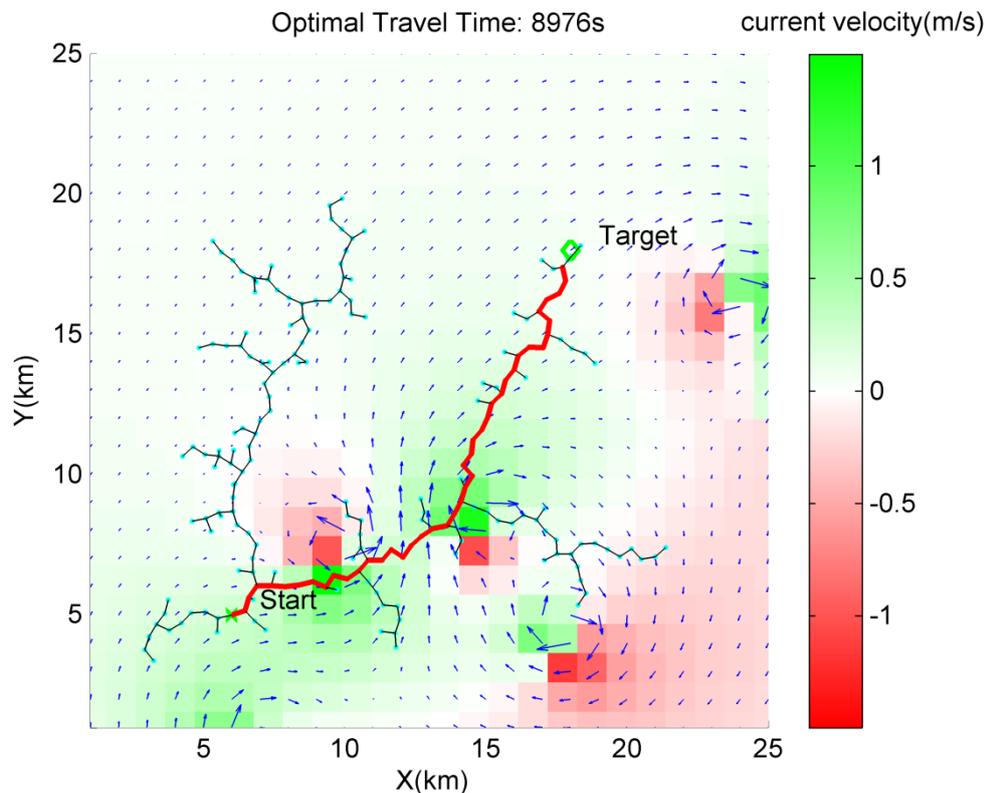
Figure 2.9 Path projections in the 2D current field without obstacles produced by RRT path planner.
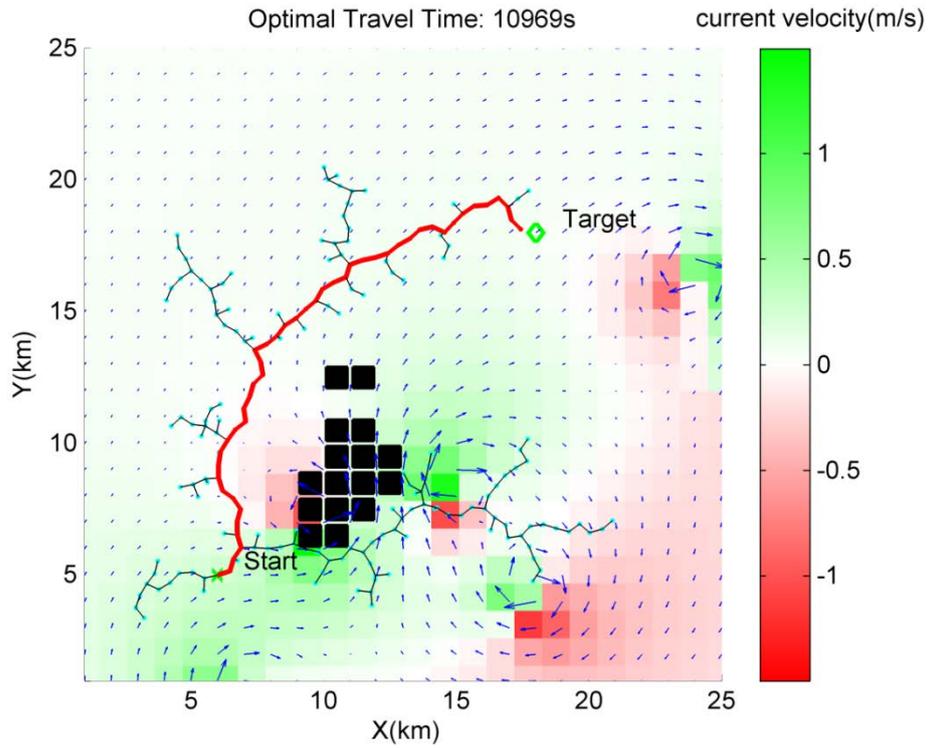
Figure 2.10 Path projections in the 2D current field with obstacles produced by RRT path planner.
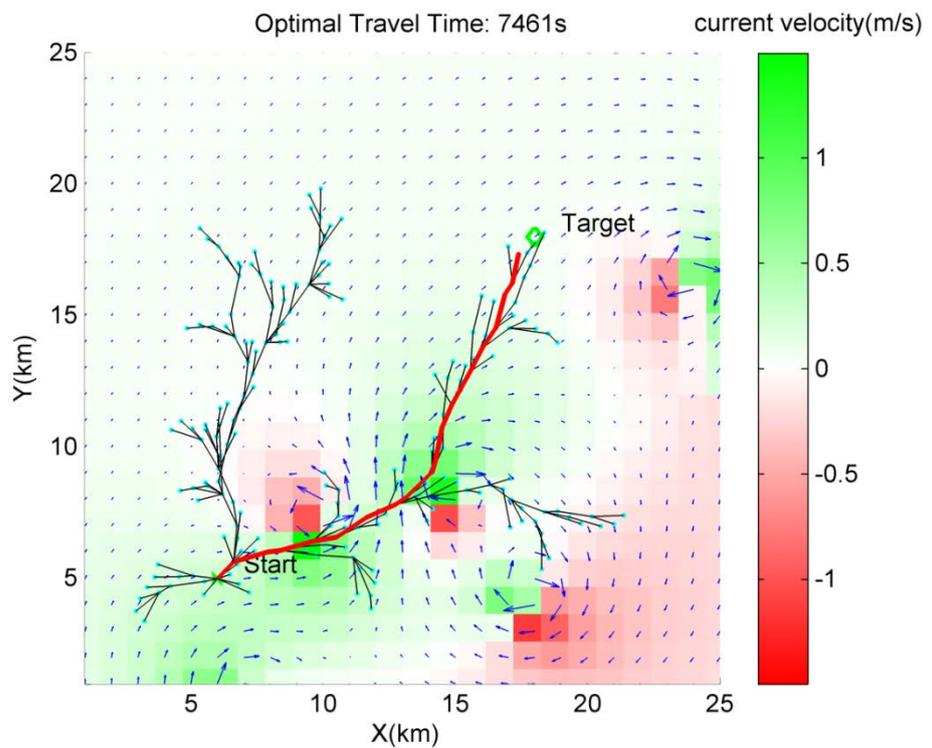
Figure 2.11 Path projections in the 2D current field without obstacles produced by RRT* path planner.
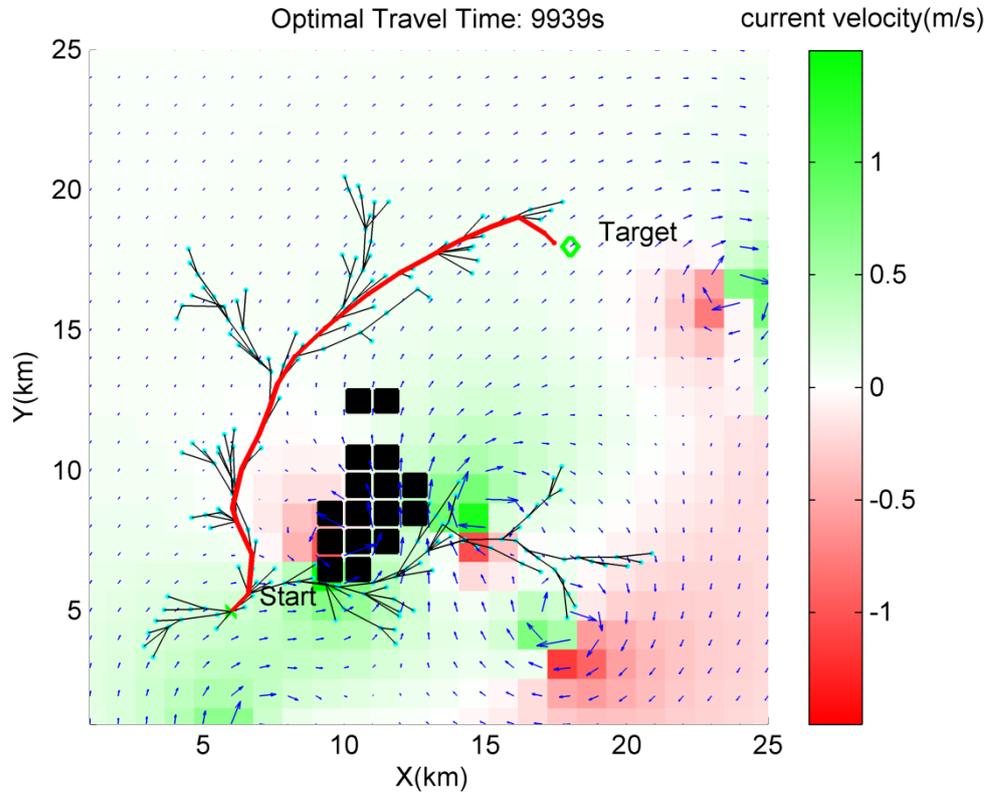
Figure 2.12 Path projections in the 2D current field with obstacles produced by
RRT* path planner.

- Evolutionary Path Planners

The optimal path projections generated by the proposed schemes: GA, PSO, QPSO, are shown in Figure 2.13(a), and Figure 2.14(a). The fitness value is the value of the fitness function for the individual. The fitness function is defined to evaluate the travel time for the AUV. It is worth noting that the best trajectory generated which is by QPSO is able to follow the current much better than those produced by GA and PSO. Similar experiments are also performed in a turbulent current field containing obstacles. With the current environment and other conditions being equal, the problem becomes more challenging since a large population of the candidate paths will become invalid if they fail to avoid the obstacles. Therefore, it is more difficult for all three algorithms to find the global optimal solution. By comparing the convergence curves of the fitness value shown in Figure 2.13(b), and Figure 2.14(b), it is evident that as the scenario becomes increasingly difficult, the convergence speeds of the all three algorithms slow down. However, the QPSO based path planner

still outperforms the other two planners and eventually finds a better trajectory with less time consumption.

(a)

(b)
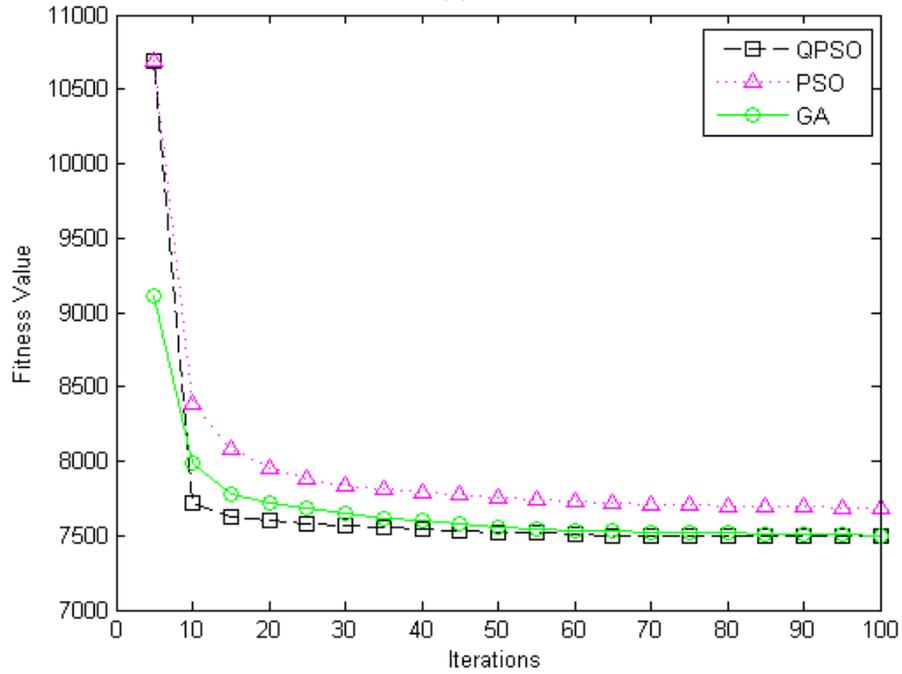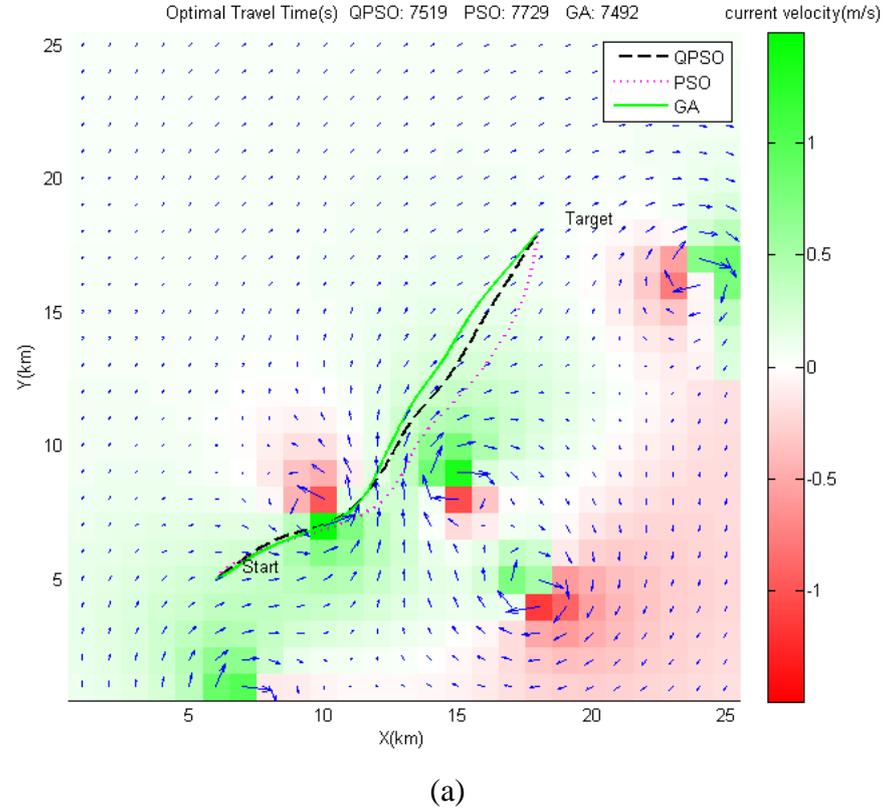
Figure 2.13 Result produced by GA, PSO and QPSO path planner: (a) path projections in the 2D current field without obstacles, (b) convergence curve of best fitness values.
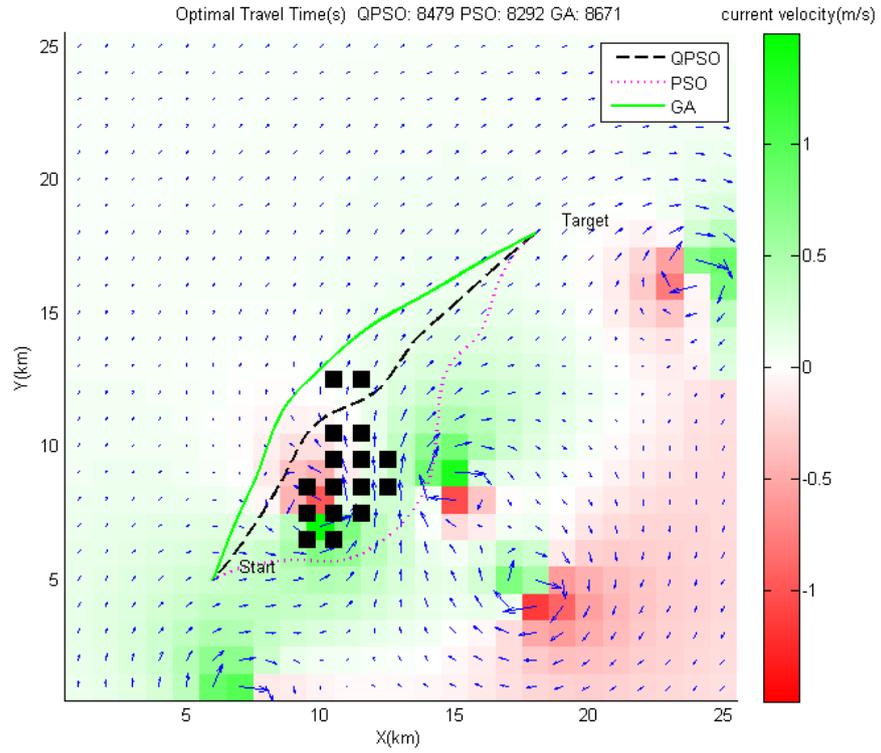
(a)
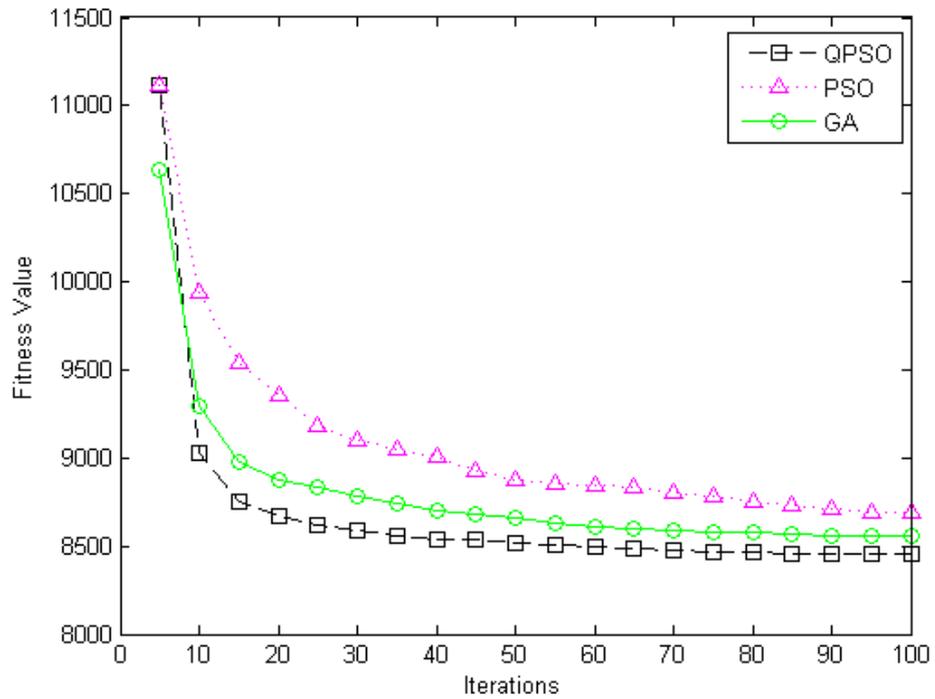
(b)

Figure 2.14 Result produced by GA, PSO and QPSO path planner: (a) path projections in the 2D current field with obstacles, (b) convergence curve of best fitness values.

2.5.3 Monte-Carlo Simulations

In this subsection, the simulations are performed on a 100-runs basis for the two test cases as described in Subsection 2.5.2. The corresponding numerical results are recorded in Table 2.4, for the stochastic algorithms (RRT and RRT*, GA, PSO and QPSO algorithms), the mean and the standard deviation of cost values are presented, since A* is a deterministic algorithm, only the cost value is reported. From Table 2.4 it can be observed that the mean cost value and the corresponding standard deviation of the QPSO are both less than those of the GA and PSO algorithms. Since the mean fitness value and the standard deviation value reflect the searching ability and stability, it can be concluded that the QPSO achieves better searching ability and robustness than the other two algorithms.

Table 2.4 Performance comparison the mean and standard deviation of A*, RRT/RRT*, GA, PSO and QPSO based path planner

Algorithm	Without obstacles		With obstacles	
	Mean	*Std*	*Mean*	*Std*
QPSO	7.50E+03	1.19E+01	8.45E+03	2.30E+02
PSO	7.68E+03	7.91E+01	8.69E+03	2.77E+02
GA	7.50E+03	9.29E+01	8.55E+03	2.13E+02
RRT	1.10E+04	9.69E+02	1.13E+04	1.30E+03
RRT*	7.80E+03	7.10E+01	8.10E+03	2.50E+02
A*	8.79E+03		9.32E+03	

The basic RRT and RRT* find paths with much larger mean and the standard deviation for the cost values than the evaluation algorithms. This is due to the RRT's tree structure rather than the comparative evaluation algorithms is with B-Spline. In addition, RRT* reduces the cost value greatly in comparison with the basic RRT.

It should be noted that the performance of the deterministic scheme A* for both test cases are relatively poor. As illustrated in Table 2.4, the cost value is bigger than that of all other algorithms except RRT.

2.5.4 Relative Performance Comparison

As can be seen in Figure 2.7 and Figure 2.8, the paths generated by A* path planner appear to oscillate slightly. This is result from the fact that the whole searching space is partitioned into discrete grid, and because of the 8-connected structure of the grids,

only a finite number of angles is representable on the discrete grid, thus the vehicle heading deviations are limited to multiples of 45°. This effect, termed digitization bias, can be found with more detail in [128]. To reduce this effect, regular grids are often finely grained, however, the computation complexity and number of function evaluations for A* is largely determined by the size of the grid and the relative proximity of the goal location to vehicle's initial position. Generally, these factors increase with the square of the number of grid points in each direction.

In the case of RRT, this algorithm samples through the configuration space searching for feasible solutions. As in the example test case shown in Figure 2.11, after 200 iterations, the RRT algorithm exhibits its innate biased towards exploration, filling the configuration space. However, despite of the good exploration capabilities of this approach, the solution seem rough and will require further refinement or some sort of smoothing procedure.

It is evident from the statistical results presented in Table 2.4 that the GA, PSO and QPSO path planners achieve better results than the that based on A* and RRT/RRT* schemes. The reasons can be found from Figure 2.13(a) and Figure 2.14(a). The proposed Spline base evolutionary path planners generate more smooth shaped paths which also closely match with the direction of currents to reduce travelling time. Compare the performance of the three evolutionary path planners, it is also evident that the vehicle consumes less travel time by following the optimized trajectories produced by QPSO. This can be explained by QPSO uses double exponential distribution and an adaptive strategy to sample particle's positions which is very different from that of PSO, and leads QPSO to be global convergent. Besides, unlike GA, QPSO uses no conventional evolutionary operators such as crossover and mutation, and also has fewer parameters to adjust, making it produced better and more robust trajectories, as indicated by the significantly smaller standard deviation for the best fitness value.

Therefore, it is concluded that the B-Spline based QPSO path planner producing more optimal and robustness solutions relative to the A*, RRT and RRT*, GA and PSO algorithms based path planners.

2.6 Chapter Summary

This chapter presents a novel B-Spline based QPSO path planner for AUV. Simulation tests have been performed to generate an optimal trajectory with minimum time consumption for an AUV travelling through turbulent ocean fields in the presence of various irregularly obstacles. Based on the results of these tests, the B-Spline based QPSO path planner is shown to be capable of finding a more optimized trajectory than the comparing stochastic RRT/RRT*, GA and PSO methods, as well as the deterministic A* method. In addition, Monte Carlo simulations demonstrate the robustness of the proposed B-Spline based QPSO scheme compared with the previous methods.

Note that the environment with static currents considered in this chapter represent the bare minimum in terms of capabilities required by an AUV path planner. It neither included any dynamic obstacles for the vehicle to avoid, nor modelled spatiotemporal currents. In fact, it may also possible to implement A* algorithm for such situations, however, a dramatic increase in computational effort can be expected. The situation will get worse as one begins considering cooperative path planning for multiple AUVs. In this case, it may not even be feasible to apply a graph search scheme. In such situations, stochastic algorithms have a distinct advantage, as long as the problem can be modelled via population and expressed in terms of cost function, solutions can be evolved. Thus, stochastic algorithms hold significant potential for solving an array of problems like path re-planning in a spatiotemporal ocean environment or cooperative path planning for multiple AUVs, which are out of reach of other path planning techniques. QPSO would be the favourable choice as it outperforms other evolutionary algorithms like GA and PSO, as well as sampling-based algorithms like RRT and RRT*.

Chapter 3

Shell Space Decomposition for AUV Path Planning

This chapter presents a shell space decomposition (SSD) scheme for B-Spline based QPSO path planner presented in Chapter 2. In this scheme, the search space is decomposed into shells radiating out from start to destination with a control point placed within each region. The trajectory is then generated from the control points using Splines. This arrangement gives freedom to the placement of the control points, while still restricting the search space to reduce computation time. The SSD scheme has been integrated with a QPSO based path planner and tested to find an optimal trajectory for an AUV navigating through a variable ocean environment in the presence of obstacles. Simulation results show that the proposed SSD approach is able to obtain a more optimized trajectory than the circle/sphere constrained methods and achieve faster convergence speed than the full space searching method. Monte Carlo trials were run to assess the robustness of the SSD method, the results demonstrate the inherent superiority of the proposed SSD method.

3.1 Introduction

In previous Chapter 2, a B-Spline based QPSO path planner is proposed, and several simulation results are presented to show that the planner yields good overall performance. One of main drawbacks, however, of evolutionary-based path planners

is their inconsistency and incompleteness of searching for the optimal path. Space decomposition and path representation techniques are viable means of increasing the searching efficiency of evolutionary based path planners. Alvarez [62] made the assumption that all the paths are strictly monotonic with respect to the x-coordinate, while the y-coordinates are set up as the chromosome genes for the evolutionary algorithm. Unlike Alvarez's work, Zheng [64] and Nikolos [63] represented the trajectory using B-Splines where the coordinates of the spline control points form the chromosome genes. The control points can be freely located anywhere in the search space. This scheme is flawed, however, because there is no clear division of search work among the control points, thus the evolutionary algorithm attempts to search the whole space with every individual control point which is highly inefficient.

In order to save computation time, the control points can be constrained to lie, for instance, on deterministically spaced concentric circles, one control point per circle, radiating out from the start point to the destination [129, 130]. However, this decomposition method distorts the search space since the area between neighbouring circles become unreachable; consequently the obtained path solution is usually suboptimal. Moreover, this approach limits the flexibility of the planner to plot optimal paths around obstacles.

An alternative control point placement method, herein referred to as shell space decomposition (SSD) scheme, is proposed in this study to facilitate more efficient searching and greater flexible placement of the control points. The SSD scheme decomposes the search space into multiple concentric shells radiating out from the start point to the destination, with one or more control points allocated to the region of each shell. This arrangement gives more freedom to the placement of the control points, but still restricts the search space for each control point to its respective regions to save computation time.

In this chapter, both 2D annulus and 3D shell decomposition schemes are developed and applied in the context of a QPSO-based path planner. Preliminary work on this line of research relating to the 2D annulus scenario scheme has already been presented in [56]. As an extension, a complete and detailed description of the scheme and its performance in a 3D scenario is proposed in this Chapter. An AUV simulator that closely matches an actual vehicle operating in a real maritime environment is

applied to predict how the vehicle will perform in tracking trajectories that have been optimised using the QPSO scheme. Moreover, a thorough robustness assessment, including performance analyses for a pre-determined scenario as well as statistical assessment of performance with various scenarios, is presented to compare the proposed SSD method with the other two control point placement methods.

The rest of this Chapter is organized as follows: Section 3.2 introduces previous methods and Section 3.3 discusses the proposed shell space decomposition method as well as its integration into the path planner. The simulation tests and results generated with the AUV simulator are then presented in Section 3.4, followed by a robustness assessment using Monte Carlo trials, the results of which are shown in Section 0. Concluding remarks are then presented in Section 3.6.

3.2 Exist Space Decomposition Methods for Spline-based Planner

This section initially provides a review of the two existing control point placement mechanisms: the unconstrained full space (FS) [63] searching mechanism and the concentric circles (CC) [130] space mechanism. Then the ASD approach is presented for the 2D scenario.

3.2.1. Unconstrained Full Space (FS) Searching

The FS searching mechanism is the original method which allows the control points to be freely located anywhere in the environment. Let the p_{ij} represent the jth control point on the ith path, p_{ij} is defined as $p_{ij}(x_{ij}, y_{ij})$ or expressed in log-polar coordinates as $p_{ij} \sim (s_{ij}, \theta_{ij})$, where

$$x_{ij} = s_{ij} \cos \theta_{ij} , \qquad (3\text{-}1)$$

$$y_{ij} = s_{ij} \sin \theta_{ij}, \qquad (3\text{-}2)$$

$$L = \sqrt{(y_\lambda - y_1)^2 + (x_\lambda - x_1)^2}, \qquad (3\text{-}3)$$

$$s_{ij} = \text{Rand}_{ij}^{Fs} l , \qquad (3\text{-}4)$$

$$\theta_{ij} = \text{Rand}_{ij}^{F\theta} \left(\frac{\pi}{2} \right), \qquad (3\text{-}5)$$

$i = 1, \dots, \rho$, ρ is equivalent to the number of particles, $j = 1, \dots, m$, m is equivalent to the number of control points used to generate the path. s is the distance of the control point from the initial position and θ is the angle with respect to the north axis, l is the length of the straight line connecting the initial and destination points, $\wp_1 (x_1, y_1)$ and $\wp_\lambda (x_\lambda, y_\lambda)$, respectively, and Rand_{ij}^{Fs} and $\text{Rand}_{ij}^{F\theta}$ are random numbers between $(0,1)$.

3.2.2. Concentric Circles/Spherical (CC/CS) Space Decomposition

3.3.1.1. Concentric Circles Space Decomposition

The CC space decomposition mechanism in contrast reduces computation time by constraining the control points to lie on deterministically spaced concentric circles, one point per circle, radiating out from the start point to the destination.

The radii of the concentric circles $E = [r_1, \dots, r_i, \dots, r_m]$ are generated as follow:

$$r_i = \left(\frac{i}{m+1} \right) l \qquad i = 1, \dots, m. \qquad (3\text{-}6)$$

In this case, $p_{ij} \sim (s_{ij}, \theta_{ij})$ is defined as

$$s_{ij} = r_i, \qquad (3\text{-}7)$$

$$\theta_{ij} = \text{Rand}_{ij}^{\theta} \left(\frac{\pi}{2} \right). \qquad (3\text{-}8)$$

3.3.1.2. Concentric Spherical Space Decomposition

As introduced above, in a 2D environment, the layers are in the form of an annulus centred at the initial position; similarly, in a 3D space, these shells appear as spherical layers.

Concentric spherical space decomposition, which is an extension of the CC space decomposition but in 3D space, constrains the control points to lie on deterministically spaced concentric spherical surfaces. The control point in 3D space is defined as $p_{ij}(x_{ij}, y_{ij}, z_{ij})$, which can be specified in spherical coordinates

$p_{ij} \sim (s_{ij}, \theta_{ij}, \phi_{ij})$ by

$$x_{ij} = s_{ij} \sin \theta_{ij} \sin \phi_{ij} \,, \qquad (3\text{-}9)$$

$$y_{ij} = s_{ij} \sin \theta_{ij} \cos \phi_{ij} \,, \qquad (3\text{-}10)$$

$$z_{ij} = s_{ij} \cos \theta_{ij}, \qquad (3\text{-}11)$$

where s and θ are defined by (3-7) and (3-8), respectively, and ϕ_{ij} is defined by

$$\phi_{ij} = Rand_{ij}^{\phi}(\tfrac{\pi}{2}) \,. \qquad (3\text{-}12)$$

3.3 Shell Space Decomposition for Spline-based Path Planner

Shell space decomposition is a general term for methods that decompose the search space into shells radiating outwards from a point in space. It is implemented as an annular space decomposition (ASD) approach for 2D space and a shell space decomposition (SSD) approach for 3D space.

3.3.1. 2D Annular Space Decomposition
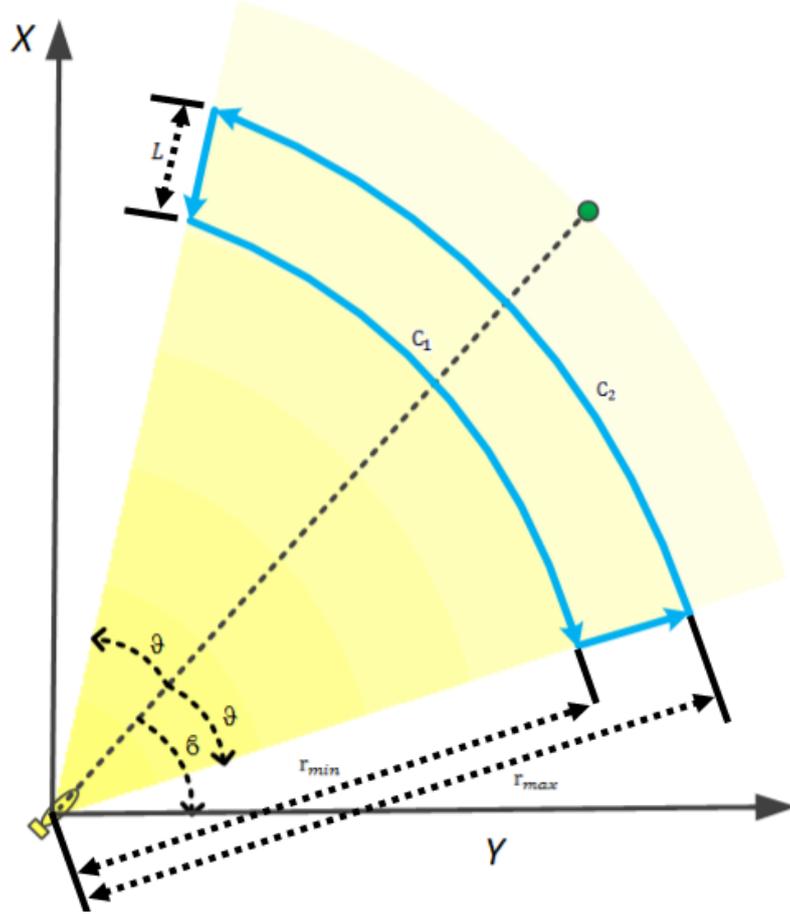
Figure 3.1 Example of the annular space decomposition.

Let us assume that R is a sector domain in the (s, θ)plane specified by the radii r_{min} and r_{max} and the angles ϱ_{min} and ϱ_{max}. The closed curve defined by the union of the circles $C_1 = \{(x, y) \in R^2, x^2 + y^2 = r^2_{min}\}$, $C_2 = \{(x, y) \in R^2, x^2 + y^2 = r^2_{max}\}$, and the line segment $L = \{(x, 0) \in R^2, r_{min} \le x \le r_{max}\}$

$$\beta = \text{atan2}(y_\lambda - y_1, x_\lambda - x_1), \tag{3-13}$$

$$\varrho_{min} = \beta - \vartheta, \tag{3-14}$$

$$\varrho_{max} = \beta + \vartheta, \tag{3-15}$$

where $\wp_1(x_1, y_1)$ is the vehicle's initial position and (l, β) is the distance and angle position with respect to the target $\wp_\lambda(x_\lambda, y_\lambda)$.

The ASD mechanism decomposes the search space into a sequence of evenly spaced

concentric annuli with one or more control points located per annulus. The annuli are formed by the space between every pair of adjacent concentric circles of radii r_{i-1} and r_i, where

$$r_i = \left(\frac{i}{\omega}\right) l \qquad i = 1, \dots, \omega \qquad (3\text{-}16)$$

The parameter ω is obtained from:

$$\omega = \frac{m}{q}, \qquad (3\text{-}17)$$

where m is the number of control points used to generate the spline path and q is the number of control points allowed within each annulus. The ASD placement mechanism allows tighter path forming around obstacles and adverse currents than the CC method.

3.3.2. 3D Shell Space Decomposition
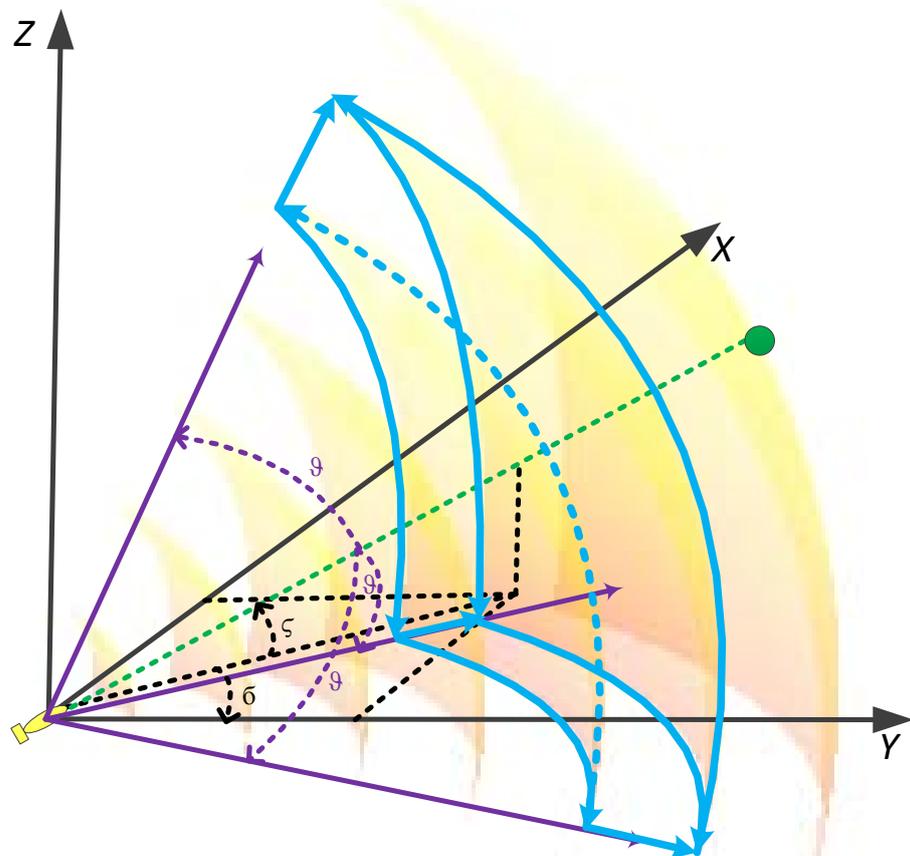
Figure 3.2 Example of the 3D shell space decomposition.

Similar to the relationship between CC and ASD in 2D space, the SSD layers are formed by the space between every pair of adjacent spheres of radii r_{i-1} and r_i, where r_i is defined in (3-16).

Let us assume that R is a sector domain in the (s, θ, ϕ) plane specified by the radii r_{min} and r_{max} and the angles ϱ_{min}, ϱ_{max} and the angles ϑ_{min}, ϑ_{max}. The closed curve defined by the union of the spheres $S_1 = \{(x, y, z) \in R^3, x^2 + y^2 + z^2 = r_{min}^2\}$, $S_2 = \{(x, y, z) \in R^3, x^2 + y^2 + z^2 = r_{max}^2\}$, and the line segment $L = \{(x, 0) \in R^3, r_{min} \leq x \leq r_{max}\}$

$$l = \sqrt{(x_\lambda - x_1)^2 + (y_\lambda - y_1)^2 + (z_\lambda - z_1)^2}, \tag{3-18}$$

$$\eth = \text{atan2}(y_\lambda - y_1, x_\lambda - x_1), \tag{3-19}$$

$$\varsigma = \text{acos2}(z_\lambda - z_1, l), \tag{3-20}$$

$$\varrho_{min} = \eth - \vartheta, \tag{3-21}$$

$$\varrho_{max} = \eth + \vartheta, \tag{3-22}$$

$$\vartheta_{min} = \varsigma - \vartheta, \tag{3-23}$$

$$\vartheta_{max} = \varsigma + \vartheta, \tag{3-24}$$

3.3.3. Integration of Shell Space Decomposition for Path Optimization

The Spline based QPSO path planner developed in Chapter 2-Section 2.4 is used as the platform to evaluate the proposed SSD mechanism. To integrate the proposed SSD scheme into optimization algorithms is quite similar to the process of integrate the FS scheme. The main modifications concern the lower and upper bounds constrain, the population initialization and the position check if the updated particle position is also within bounds.

3.3.3.1. Bound Constraints

For ASD, every point has its bounded searching region. This region is represented as

$\mathcal{B} = [\mathcal{B}s, \mathcal{B}\theta]$, where $\mathcal{B}s$, $\mathcal{B}\theta$ are the bounds for s and θ, respectively. Let $\mathcal{B}s = [Ls\ Us]$, and $\mathcal{B}\theta = [L\theta\ U\theta]$, where Ls, L$\theta$ are the lower bounds and Us, Uθ are the upper bounds, which are defined as follows:

$$Ls = [0, r_1, \dots r_{i-1}, \dots r_{m-1}], \tag{3-25}$$

$$Us = [r_1, r_2, \dots r_i, \dots r_m]. \tag{3-26}$$

$$L\theta = [\varrho_{min}, \varrho_{min}, \dots, \varrho_{min}, \dots, \varrho_{min}], \tag{3-27}$$

$$U\theta = [\varrho_{max}, \varrho_{max}, \dots, \varrho_{max}, \dots, \varrho_{max}]. \tag{3-28}$$

The constraint on every control point is therefore:

$$\left(s_{ij} \in \mathcal{B}s_j\right) \cap \left(\theta_{ij} \in \mathcal{B}\theta_j\right). \tag{3-29}$$

For SSD, since the control points are defined as $p_{ij} \sim (s_{ij}, \theta_{ij}, \phi_{ij})$, their corresponding bounds are modified to be $\mathcal{B} = [\mathcal{B}s, \mathcal{B}\theta, \mathcal{B}\phi]$, where $\mathcal{B}\phi = [L\phi\ U\phi]$, and L$\phi$, U$\phi$ are written as:

$$L\phi = [\vartheta_{min}, \vartheta_{min}, \dots, \vartheta_{min}, \dots, \vartheta_{min}], \tag{3-30}$$

$$U\phi = [\vartheta_{max}, \vartheta_{max}, \dots, \vartheta_{max}, \dots, \vartheta_{max}]. \tag{3-31}$$

The constraint on control points in SSD is defined as:

$$\left(s_{ij} \in \mathcal{B}s_j\right) \cap \left(\theta_{ij} \in \mathcal{B}\theta_j\right) \cap \left(\phi_{ij} \in \mathcal{B}\phi_j\right), \tag{3-32}$$

3.3.3.2. Constrained Optimization

- **Population Initialization**

QPSO is initialized to have a population size of ρ candidate paths, and each candidate path is generated from m control points. For 2D ASD each point $p_{ij} \sim (s_{ij}, \theta_{ij})$ is generated from:

$$s_{ij} = Ls_j + Rand^s_{ij}(Us_j - Ls_j), \tag{3-33}$$

$$\theta_{ij} = \text{Rand}_{ij}^{\theta}(U\theta_j - L\theta_j)\,, \tag{3-34}$$

with $i = 1, \ldots, m$ and $j = 1, \ldots, \rho$, Rand_{ij}^{s} and $\text{Rand}_{ij}^{\theta}$ are randomly generated numbers within the interval $(0,1)$.

For 3D SSD, the extra parameter ϕ_{ij} is generated from:

$$\phi_{ij} = \text{Rand}_{ij}^{\phi}(U\phi_j - L\phi_j)\,. \tag{3-35}$$

- **Position Check**

One more step is added to follow the position update to check if the new particle position is within bounds. The following expression ensures that every dimension of the updated particle meets its constraints:

$$p_{ij}(t+1) = \begin{cases} p_{ij}(t+1), & \text{if } s_{ij} \in (LS_j, US_j) \cap \theta_{ij} \in (\theta S_j, \theta S_j) \\ p'_{ij}(t+1), & \text{otherwise} \end{cases}, \tag{3-36}$$

where $p'_{ij}(t+1)$ is a regenerated particle from Chapter 2. Equations (2-46)~(2-48).

3.4 Simulations

To evaluate the performance of the proposed SSD scheme relative to the existing FS, CC control point placement schemes, case studies with various mission scenarios are performed in Subsection 3.4.1. As further testimony that the trajectories generated by the proposed path planners are indeed capable of being faithfully followed by an AUV in a realistic maritime environment, an AUV simulator that incorporates known hydrodynamic and hydrostatic effects is employed to execute a set of trajectories in Subsection 3.4.2.

3.4.1 Simulation Experiments with Proposed Control Point Placement Mechanisms

The FS, CC and SSD mechanisms were integrated in the QPSO-based path planner and implemented in Matlab. Three case studies are tested to compare the performance of the three path planning mechanisms within currents field in the presence of different obstacles. The first case deals with a time-minimized trajectory

through a dynamic currents field from start to destination. The next two cases deal with an environment where obstacles of propagated uncertainty in position (Section 2.3.1) occur.

3.4.1.1 Simulation Setup

As mentioned in Chapter 2, Section 2.3, the current field for the scenario reported in this study, within a 2D spatial domain, is generated from a random distribution of 50 Lamb vortexes represented by a 100×100 grid. The 3D ocean environment is approximated by a five-layered structure in which each layer is defined similarly to the 2D case but with a well-defined vertical profile [62, 131]. The distance between nearest neighbour grid points corresponds to 1000 m for 2D space and 100m for 3D space. The strength ζ and radius η of each vortex are set as 15 m/s and 2 m, respectively. The water-referenced speed of the vehicle in the geographical frame is set at 2 m/s. The initial and final destination points are located at grid locations (1, 1) and (60, 80) for the 2D environment test, and (1, 1, 1) and (55, 55, 55) for the 3D environment test, respectively. Each B-spline curved path is uniformly subdivided by 1,152 points, the quantity of the points is conservatively chosen such that it is large enough to guarantee that the spline formed paths closely conform to the intended paths.

The experimentally optimized settings of the QPSO algorithm are as follows: the population size is 1000 for the 2D scenario and 5000 for the 3D scenario, and the maximum number of iterations is 100. The contraction-expansion coefficient β is set to linearly decrease from 1.0 to 0.5 over the 100 iterations. Each individual B-spline path is formed by 6 control points. For simplicity, 6 annuli/spheres are generated for the ASD and SSD based path planner such that only one control node is located within each shell.

3.4.1.2 Current Field with Static Obstacles of Fixed Uncertainty in Position

Figure 3.3 displays the result of the trajectory optimization in a scenario containing seven static obstacles with varying levels of position uncertainty within a current field. The position uncertainty of each obstacle is represented as a black circle around the obstacle with radius $2\sigma_o$ indicating a confidence of 95.4% that the

obstacle is located within this area. The optimal path projections generated by the proposed schemes: QPSO-FS, QPSO-CC, QPSO-ASD, are shown in Figure 3.3(a). The magenta concentric circles represents the circles on which control points are constrained to lie by the CC decomposition method, while the green concentric circles represents the boundaries of the annuli within which control points are placed by the ASD decomposition method. The convergence curves showing the best fitness values for each method with the corresponding numbers of iterations is shown in Figure 3.3(b). Similar experiments are also performed in a 3D space containing eight obstacles with different levels of position uncertainty. The results are shown as Figure 3.4.

By comparing the convergence curves shown in Figure 3.3(b) and Figure 3.4(b), it can be seen that FS scheme has the slowest rate of convergence relative to the CC/CS and SSD schemes. The reason can be found from Figure 3.3(a) which reveals that the FS scheme permits control points to cluster closely together leading to inefficient searching, whereas in the case of the CC and SSD schemes (i.e., the ASD scheme in Figure 3.3(a) and the SSD scheme in Figure 3.4(a)), the control points are spread out over the entire search space, thus avoiding misplaced effort.

It is also evident from Figure 3.3(b) and Figure 3.4(b) that the proposed SSD scheme achieves better results than the CC/CS and FS schemes in both 2D and 3D environment. The reasons can be found from Figure 3.3(a) and Figure 3.4(a). The proposed CC/CS scheme constrains the control points to lie on the circumference of its respective concentric circle/sphere, while in contrast, the SSD scheme allows more flexibility to search around obstacles and generate more complex shaped paths which reduces travelling time. In principle, although the FS scheme should also offer the same flexibility as the SSD scheme and have the potential advantage of constructing complex shaped paths, it is more likely to become stuck at a local minimum because of its potential for overlapped searching.
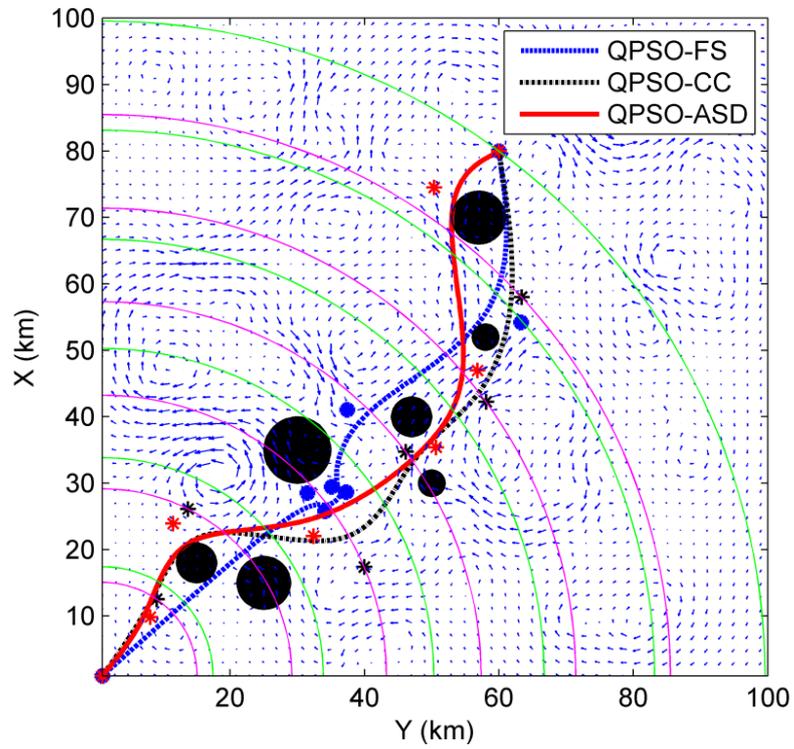
(a)
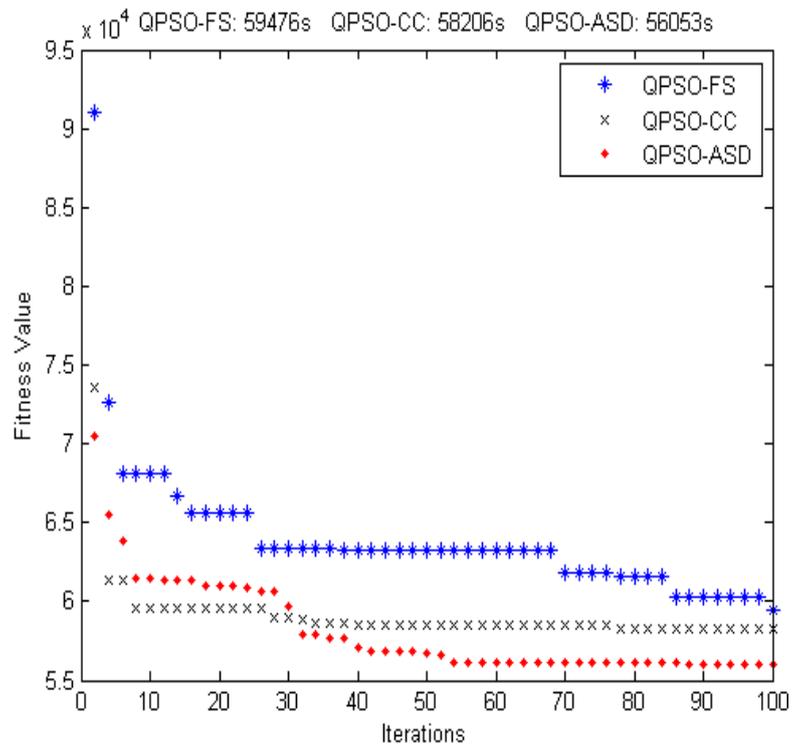
(b)

Figure 3.3 Comparison of results produced by the FS, CC and ASD schemes with the QPSO path planner for static obstacles with fixed uncertainty: (a) path projections in the 2D current field, (b) convergence curve of best fitness values.
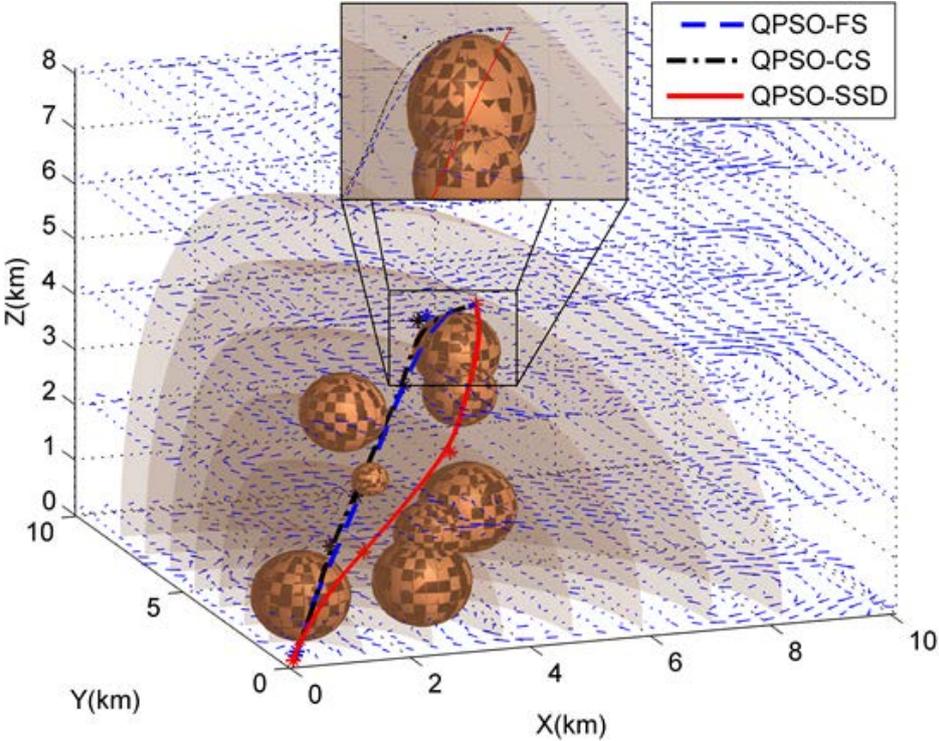
(a)
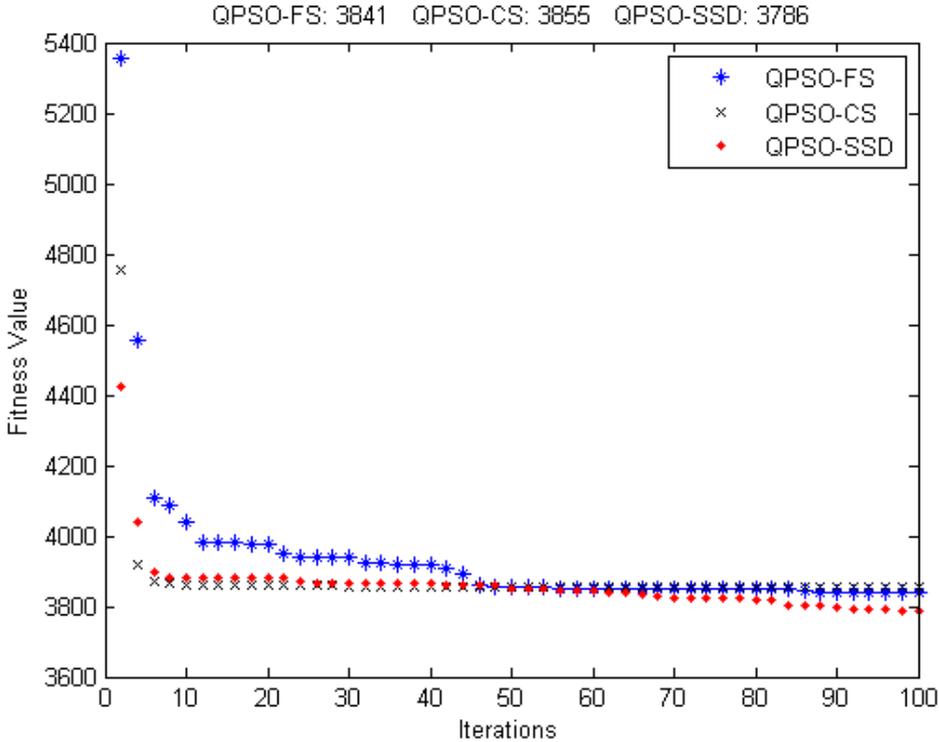
(b)

Figure 3.4 Comparison of results produced by the FS, CS and SSD schemes with the QPSO path planner for static obstacles with fixed uncertainty: (a) path projections in the 3D current field with zoomed in section shown alongside, (b) convergence curve of best fitness values.

3.4.1.3 Current Field with Obstacles of Propagated Uncertainty in Position

Similar simulations are conducted here with obstacles of propagated uncertainty in position. Two cases, based on quasi-static and constantly moving obstacles with uncertainty in position, respectively, are considered here.

- **Quasi-Static Obstacles**

The positions of these obstacles are modelled as Gaussian distributions and their position uncertainty is expressed as the growth of the variance. Since the propagation of the uncertainty is assumed linear with time, it is expressed in the plots shown in Figure 3.5(a) and Figure 3.6(a) as a proportional increment in the collision boundary encircling the object. The position uncertainty of obstacles at different times is represented in these plots by different colours. Note, the optimal curve obtained from QPSO-SSD is also represented by changing colours. These colours reflect the position of the vehicle at the respective point in time when following the path. At each discrete time point, the obstacle position uncertainty and the corresponding path node (vehicle location) are represented by the same colour; thus if the node is not inside the correspondingly coloured obstacle boundary, no collision will occur.
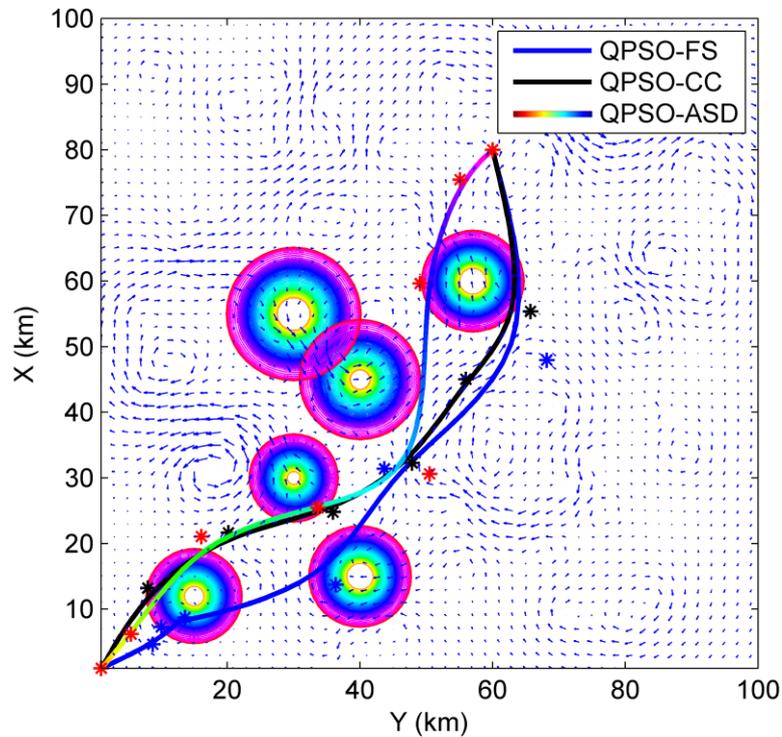
(a)
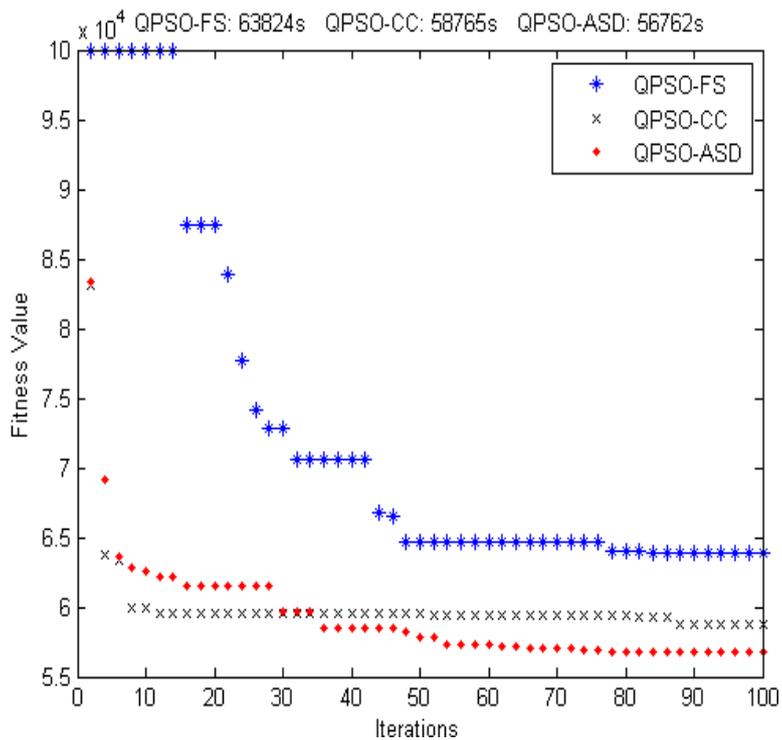
(b)

Figure 3.5 Comparison of results produced by the FS, CC and ASD schemes with the QPSO path planner for quasi-static obstacles with propagated uncertainty: (a) Path projections in the 2D current field. (b) Convergence curve of best fitness values.
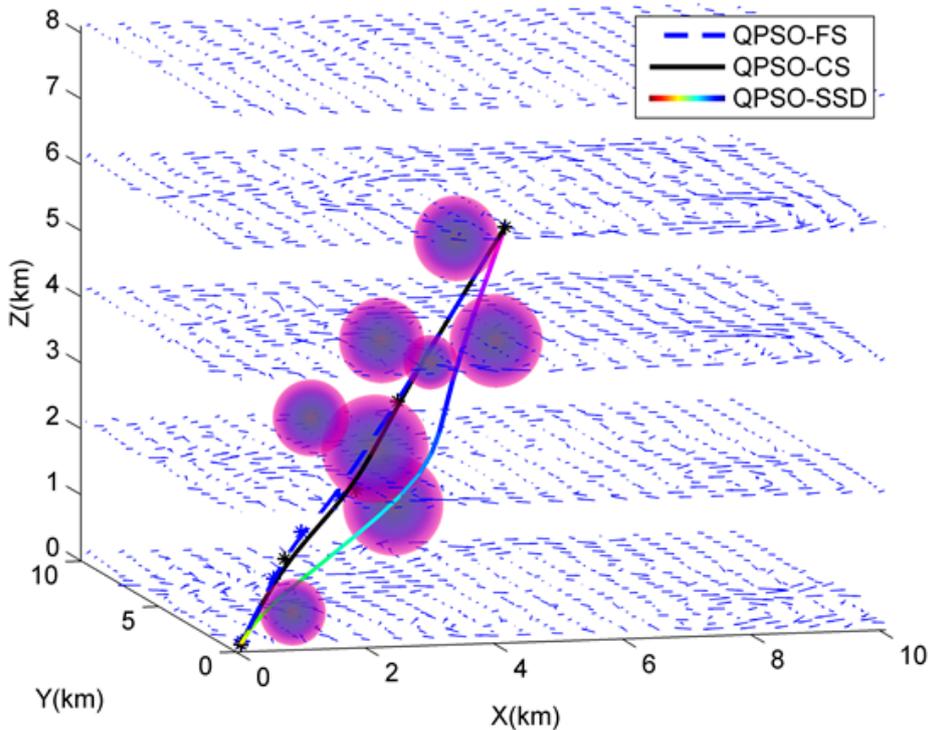
(a)
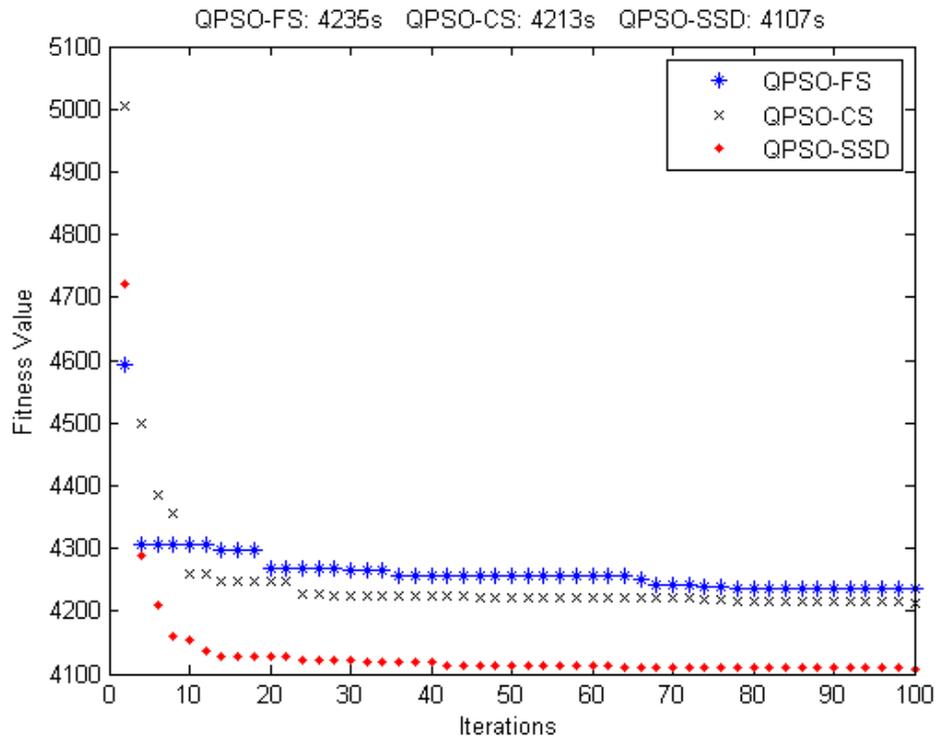
(b)

Figure 3.6 Comparison of results produced by the FS, CS and SSD schemes with the QPSO path planner for quasi-static obstacles with propagated uncertainty: (a) Path projections in the 3D current field. (b) Convergence curve of best fitness values.

- **Constantly Moving Obstacles**

Based on the previous case of quasi-static obstacles with propagated position uncertainty, little modification is needed to model the obstacles' positions as constantly moving with growing uncertainty. The initial positions of the obstacles are randomly located and modelled as Gaussian distributions. Their centres (relating to the mean of the distribution) are configured to move independently at various rates in different directions.

Figure 3.5 and Figure 3.6 display the simulation results of the scenario with quasi-static obstacles. These results demonstrate that the SSD scheme achieves faster convergence speed and improved global searching ability relative to the FS and CC/CS schemes. The same is true for the results of the scenario with constantly moving obstacles, as observed from Figure 3.7 and Figure 3.8.
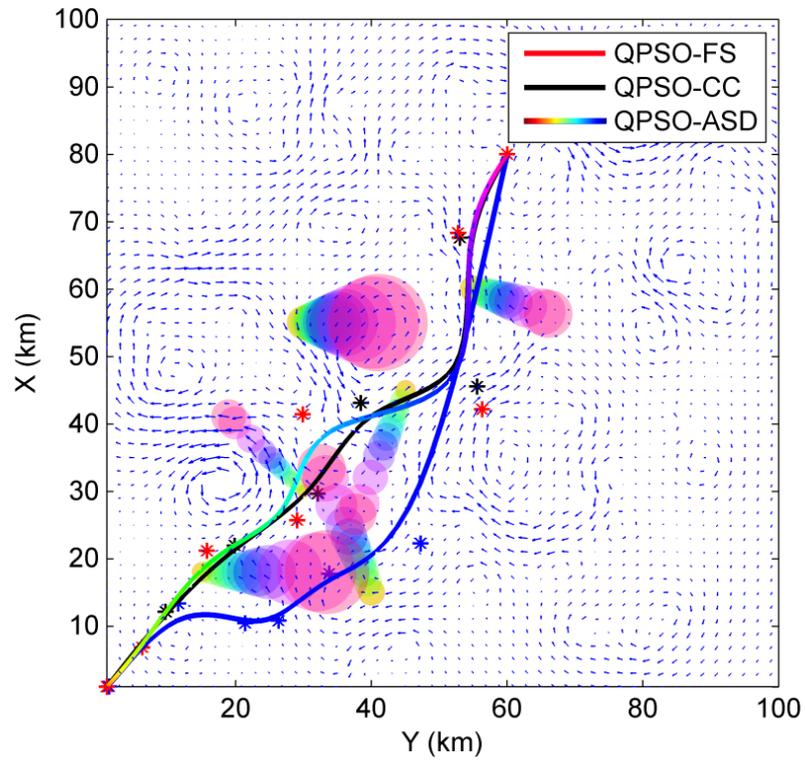
(a)
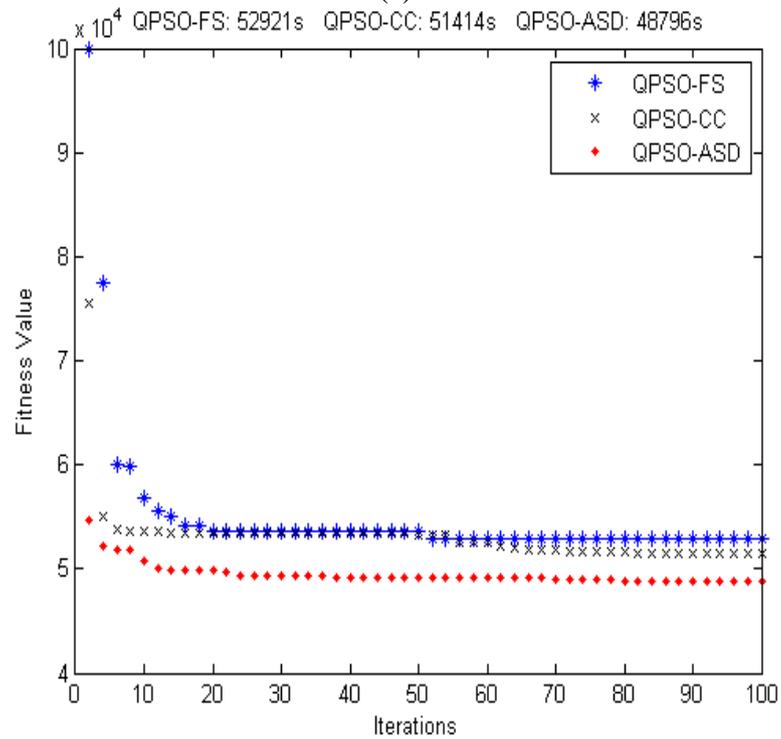
(b)

Figure 3.7 Comparison of results produced by the FS, CC and ASD schemes with the QPSO path planner for constantly moving obstacles with propagated uncertainty: (a) Path projections in the 2D current field. (b) Convergence curve of best fitness values.
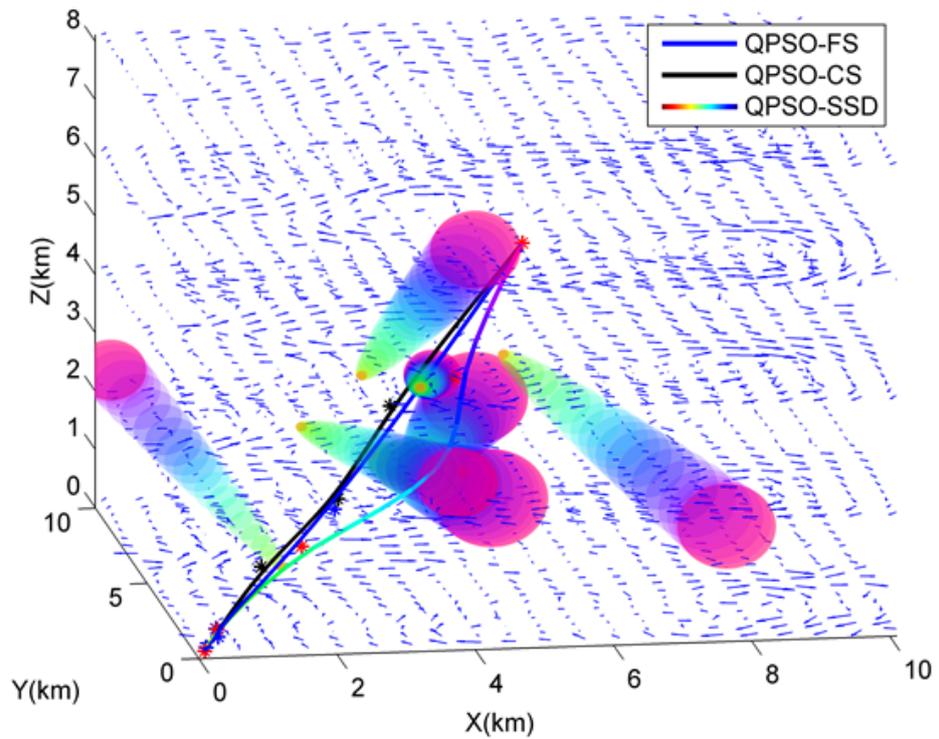
(a)
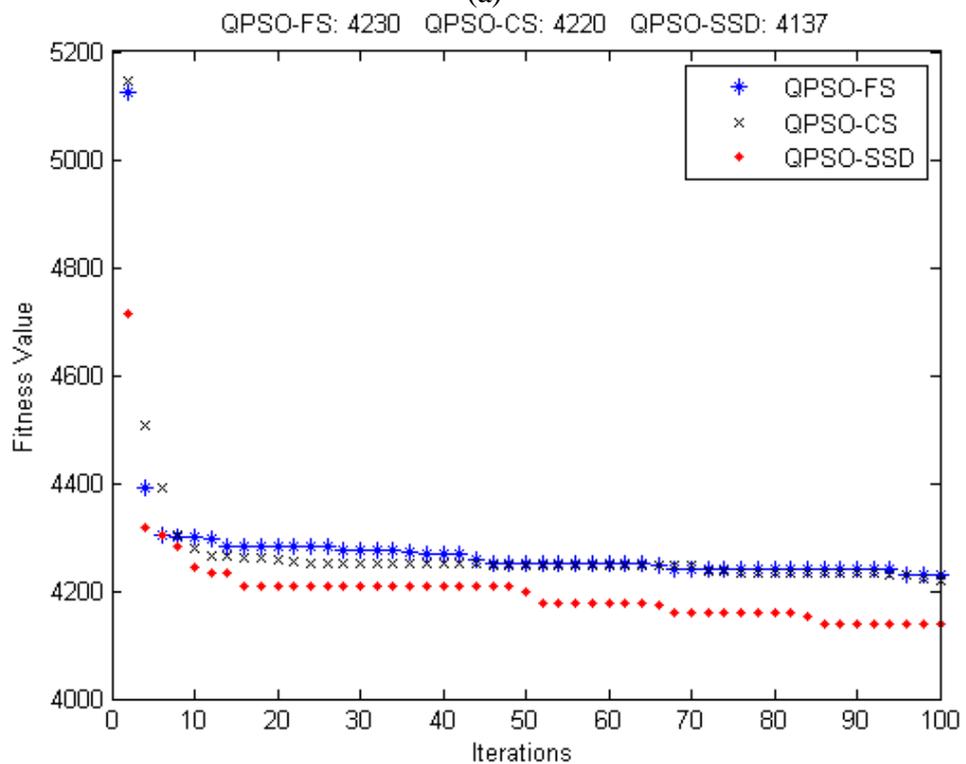
(b)

Figure 3.8 Comparison of results produced by the FS, CC and ASD schemes with the QPSO path planner for constantly moving obstacles with propagated uncertainty: (a) Path projections in the 3D current field. (b) Convergence curve of best fitness values.

Based on these simulation results, it is seen that in both the 2D and 3D environments, the SSD scheme in comparison with the FS and CC/CS schemes gives the best performance with the QPSO-based path planner regardless of whether the obstacles are quasi-static or constantly moving.

3.4.2 Experimental testing with AUV simulator

In this section, an AUV simulator (developed using Simulink, see Appendix C) is used to test the trajectories generated by the proposed path planners that have been developed in Matlab. Figure 3.9 provides an overview of this implementation. The path planner passes the generated trajectory (in the form of discretized curve points) to the line-of-sight (LOS) guidance system of the simulator, then the sliding mode controller (SMC) controls the vehicle to follow this desired trajectory. The output is the dynamic state of the vehicle executing this trajectory.

The AUV simulator employed models the behaviour of a Remus 100 AUV [132] when operating in a maritime environment. All components within the simulator are modelled as realistically as possible. The AUV simulator incorporates the same environmental behaviour, such as ocean currents and waves, and the same measurement references, such as water-referenced speed, as the path planner.
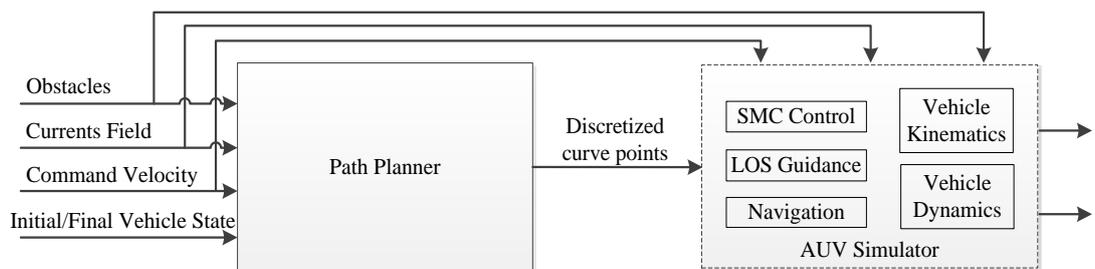
Figure 3.9 Path planner and AUV simulator.

3.4.2.1 The AUV Simulator

This simulator contains two principal parts. The Kinematics and Dynamics modules are described in 3.4.2.1 -a. The Guidance, Navigation and Control (GNC) modules are described in 3.4.2.1 -b.

- **Kinematics of the Vehicle**

The AUV is modelled as a free-moving body within 6 degrees of freedom (6DOF)

space. The states η, represented in the North-East-Down (NED) frame (3-37) are defined as North, x, East, y, Down, z, and the Euler angles roll, ϕ, pitch, θ, and yaw, ψ. The velocities ν, which are defined in the body frame (3-38), comprise the components u, v, and w which represent the surge, sway, and heave linear motions, respectively while p, q, and r, represent the roll, pitch, and yaw rotational rates, respectively.

$$\eta = [x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T, \tag{3-37}$$

$$\nu = [u \quad v \quad w \quad p \quad q \quad r]^T, \tag{3-38}$$

- **Dynamics of the Vehicle**

The mathematical framework describing the motion of a torpedo shaped AUV in a kinetic sense is as following:

$$M\dot{\boldsymbol{v}} + \boldsymbol{C}_{RB}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{C}_A(\boldsymbol{v}_r)\boldsymbol{v}_r + \boldsymbol{D}(\boldsymbol{v}_r)\boldsymbol{v}_r + \boldsymbol{L}(\boldsymbol{v}_r)\boldsymbol{v}_r + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{w}, \tag{3-39}$$

$$\boldsymbol{v}_r = \boldsymbol{v} - \boldsymbol{v}_c, \tag{3-40}$$

where

$\dot{\boldsymbol{v}}$: vehicle acceleration vector

$\boldsymbol{v}$: vehicle velocity vector

$\boldsymbol{\eta}$: vehicle position vector

$\boldsymbol{v}_c$: water velocity vector

$\boldsymbol{u}$: control vector

$\boldsymbol{M}$: Mass Matrix

$\boldsymbol{C}_{RB}(\boldsymbol{v})$: Rigid Body Coriolis Force Matrix

$\boldsymbol{C}_A(\boldsymbol{v}_r)$: Added Mass Coriolis Force Matrix

$\boldsymbol{D}(\boldsymbol{v}_r)$: Drag Matrix

$\boldsymbol{L}(\boldsymbol{v}_r)$: Lift Matrix

$\boldsymbol{g}(\boldsymbol{\eta})$: Buoyancy and Gravitational Force Vector

$\boldsymbol{B}$: Actuation Dynamic Matrix

$\boldsymbol{w}$: Vector of Un-modelled Disturbances

A more detailed description of the general structure of this equation and the phenomena that contribute to these terms is given in [133].

- **GNC system**

The GNC system includes a LOS guidance module, a fully-coupled sliding mode control (SMC) and a navigation module. It is worthwhile noting that the SMC controller takes into account as much coupling within the vehicle model as possible. This means that the equations of motion, including all modelled cross-coupling hydrodynamic coefficients, are retained during the design process. Hence, the resulting fully-coupled controller can optimally compensate all DoFs at once; consequently, the proposed compensation technique reduces overuse of the actuators, and hence increases the power efficiency of the control strategy. Detail of this control system can be found in [134].

3.4.2.2 Results from the Simulator

As previously mentioned, the vehicle is commanded to track the trajectory (i.e., the discretised curve points, not the B-Spline control points) provided by the path planner. For comparison, two reference tests are also added to each test scenario; the first commands the vehicle to go in a straight line directly towards the destination point [135] while the second directs the vehicle along the optimized trajectory but in an environment devoid of currents. Figure 3.10 and Figure 3.11 show the results generated by the AUV simulator for the 2D and 3D scenarios, respectively. Figure 3.10(a) and Figure 3.11(a) show the corresponding trajectories tracked by the vehicle as generated by the four guidance strategies in the simulator. The remaining set of plots in Figure 3.10 and Figure 3.11 show, where pertinent, the velocities and rotational angles of the vehicle corresponding to each of the trajectory generating schemes. It can be seen from these figures, for both 2D and 3D scenarios, that the vehicle achieves higher velocity by taking the QPSO optimized trajectory rather than the straight-line trajectory.
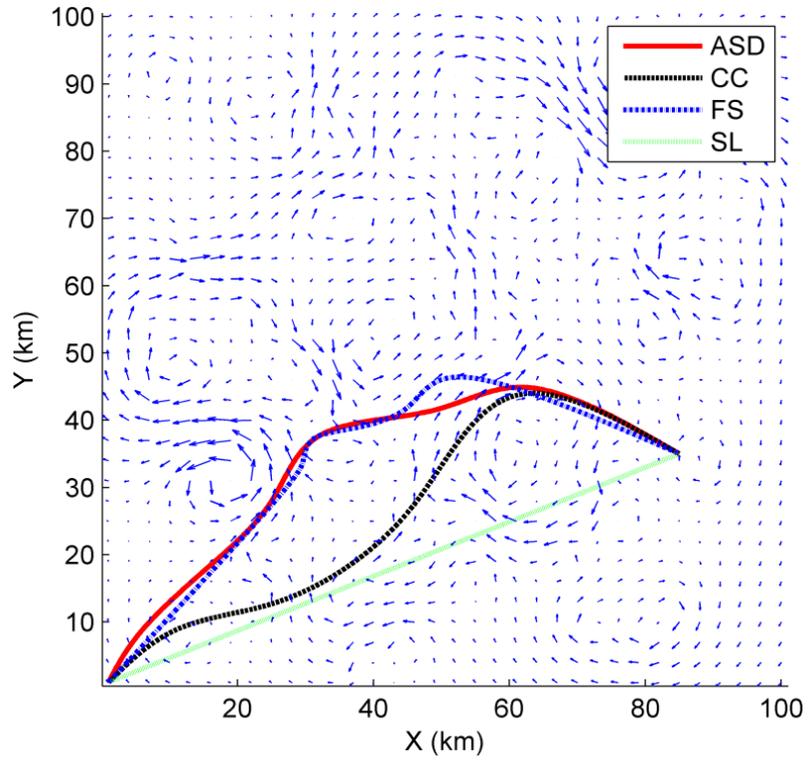
(a)
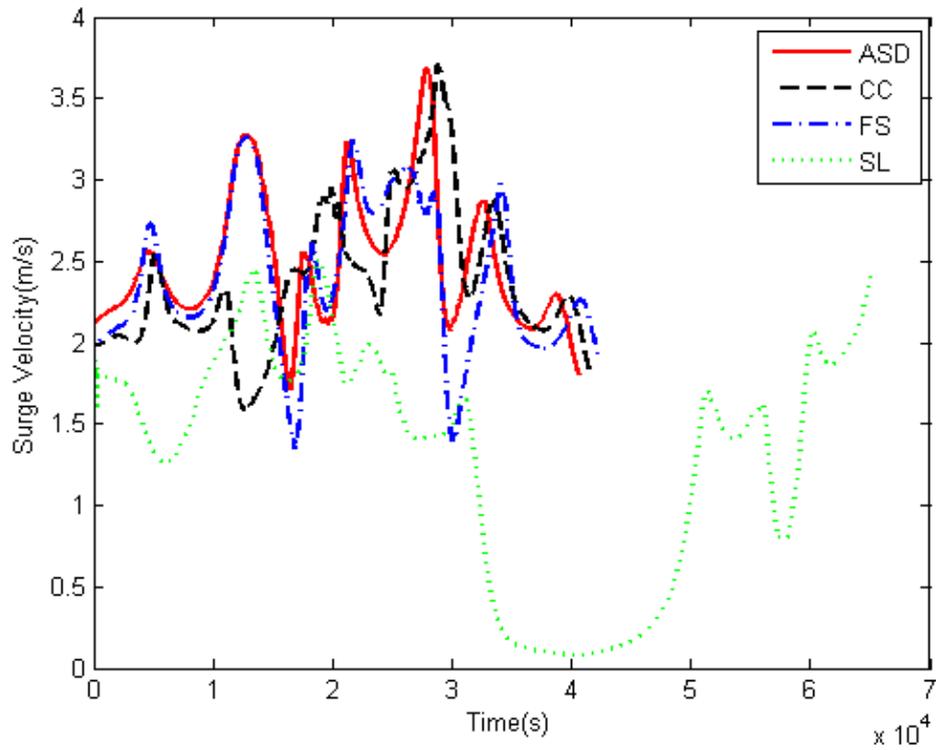
(b)
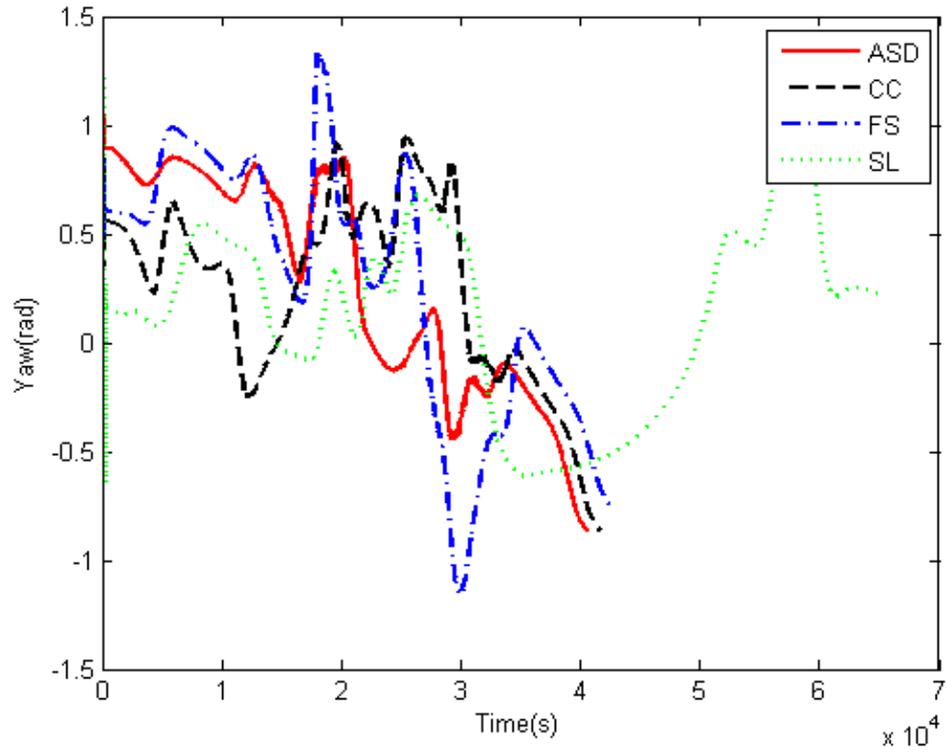
(c)

Figure 3.10 Results generated by the AUV simulator with the QPSO path planner: (a) 2D plot of generated trajectories, (b) surge velocity vs. time, (c) yaw vs. time.

(a)
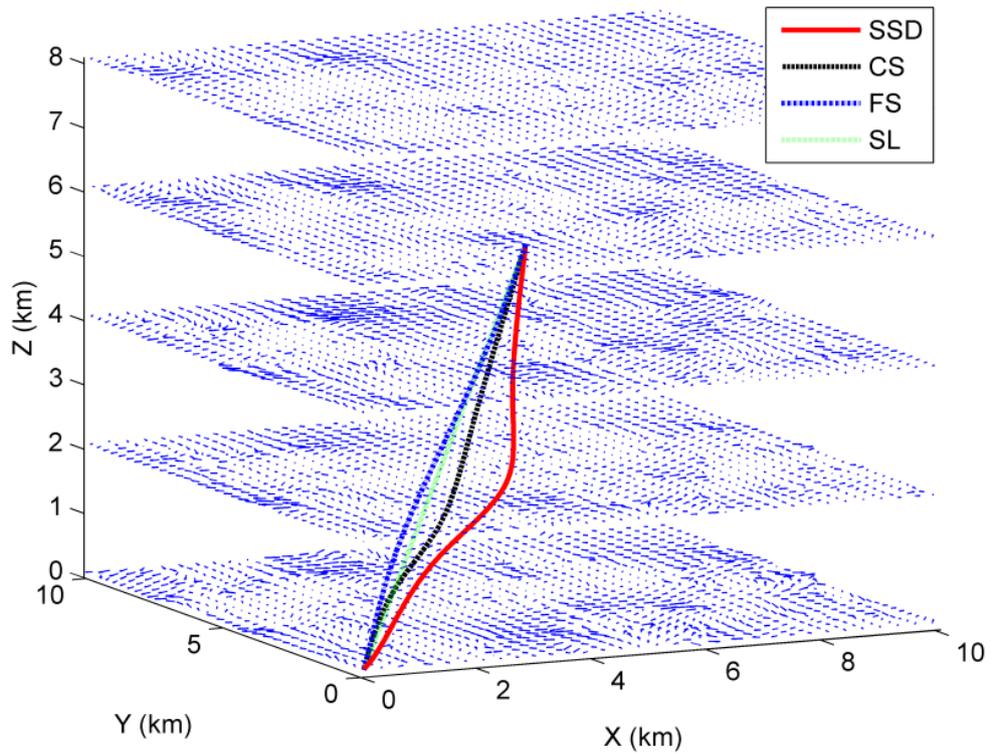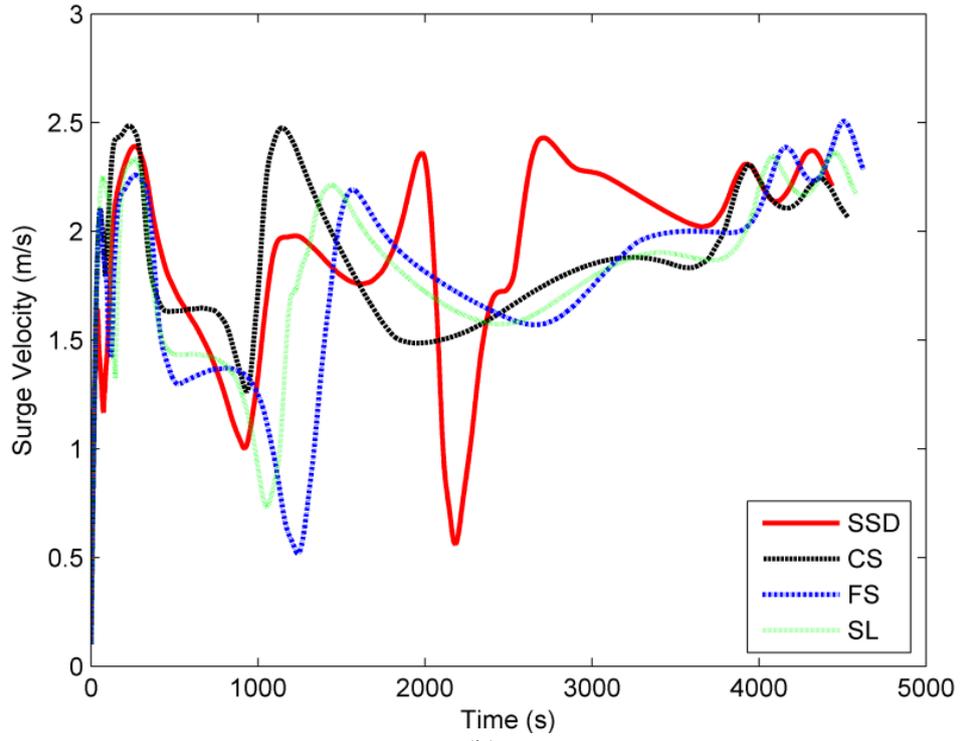
(b)
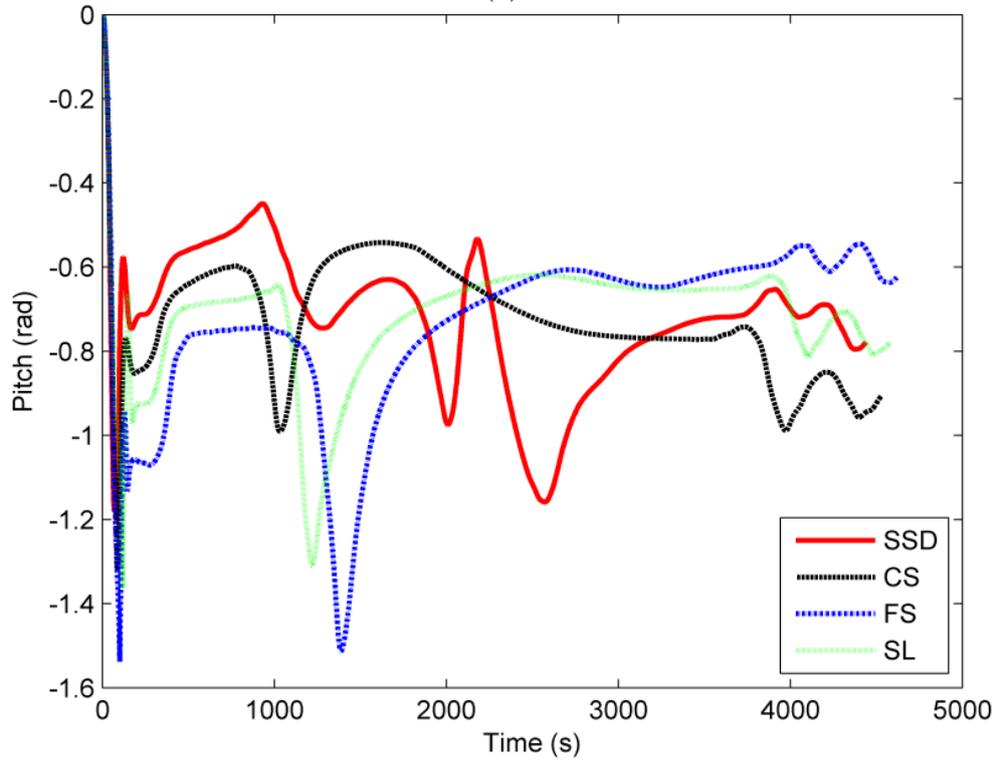
(c)
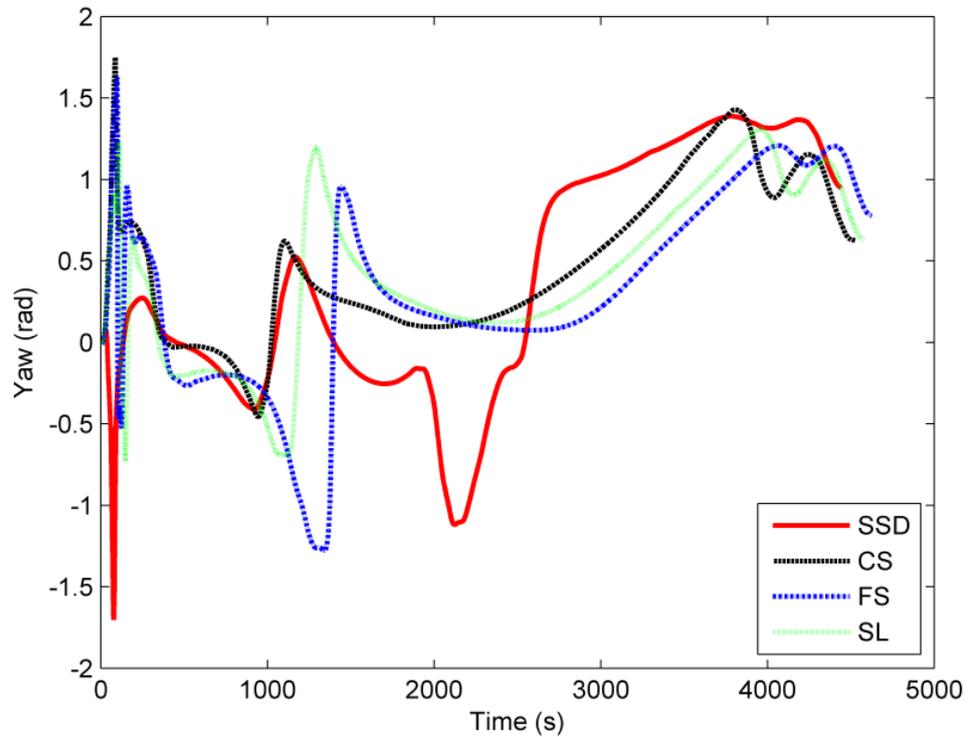
(d)

Figure 3.11 Results generated by the AUV simulator with the QPSO path planner: (a) 3D plot of generated trajectories, (b) surge velocity vs. time, (c) pitch angle vs. time, (d) yaw angle vs. time.

The travel times of these tests are listed in **Error! Reference source not found.**. It is evident that the vehicle consumes less travel time by following the optimized trajectories produced by QPSO. This can be explained by observing for the straight-line trajectory that the vehicle may proceed through a region with the current running contrary to the vehicle resulting in a markedly reduced ground-relative velocity. In contrast, all optimized trajectories are formed by following the direction of the currents, especially the trajectory returned by the SSD-based path planner. This procedure allows the vehicle to ride the currents towards its destination hence attaining increased velocity and reducing overall travel time, even though the length of the path may be extended relative to the straight line path. By comparing the time consumed for the corresponding tests along the optimised trajectories, with and without currents, it can be seen that the vehicle takes significantly less travel time in the presence of the currents, which further testifies that the optimised trajectories take advantage of favourable currents, and avoid adverse currents to minimize time expenditure.

Table 3.1 Comparison of mission times achieved with the different control point placement schemes

Tests with the AUV simulator		Time (sec)	
		With currents	Without currents
2D	Straight line (SL) path	65,244	45,140
	ASD optimized path	40,575	52,349
	FS optimized path	41,591	50,979
	CC optimized path	42,366	52,945
3D	Straight line (SL) path	4,580	4,901
	SSD optimized path	4,439	5,109
	FS optimized path	4,531	4,940
	CS optimized path	4,624	5,105

3.5 Robustness Assessment

Monte Carlo simulation runs were conducted to assess the performance for a single scenario and also to perform robustness assessment with various scenarios. The settings for each run are mostly the same as those described in Section 3.4, except that a stopping criterion is added which permits the simulation to terminate early if this criterion is met. This stopping criterion is satisfied when the weighted average change, *E,* (3-41) in the fitness function value over a set number of iterations (20 iterations) is less than the function tolerance (1×10^{-6}).

$$E = \sum_{i=1}^{20} \left(((\frac{F_{I-i}(t) - F_{I-i-1}(t))}{F_{I-i}(t)}) \right) (0.5)^{i-1}, \tag{3-41}$$

where *i* is the number of the iteration index and F is the relevant fitness value.

3.5.1 Performance for a Single Scenario

In this subsection, the simulations are performed on a 1000-runs basis for a single scenario as described in 3.4.1.2. The performances of the FS, CC/CS and SSD schemes are compared based on the following three factors: searching ability, convergence speed and computation time. The searching ability is reflected in the quality of the solution as given by the mean best fitness value and the stability. Since the proposed optimization is a minimization problem, the smaller the mean value, the stronger the searching ability. The stability is evaluated from the standard deviation. The iteration count reveals how many iterations the algorithm takes to converge to the optimal solution. Computation time is the mean CPU time consumed over 1000

runs for each strategy. Note that the implementation uses the Matlab R2012a, and these tests were run on a 3.33GHz Duo-core computer.

From the statistical results presented in **Error! Reference source not found.**, within both 2D and 3D domains, it can be seen that the SSD approach generated the solution with the lowest mean best fitness value and standard deviation compared with those of the other two methods. This result suggests that SSD is able to achieve a more optimized trajectory than the CC/CS method, and with faster convergence and less computation time than the FS method.

Table 3.2 Performance Comparison of FS, CC/CS, SSD Schemes with QPSO Based Path Planner

Algorithm		Best Fitness		Iteration Count		Computation Time (sec)	
		Mean	*Std*	*Mean*	*Std*	*Mean*	*Std*
2D	**QPSO-FS**	4.3081e+04	176.6893	74.62	6.4154	1.3751e+03	300.5370
	QPSO-CC	4.2020e+04	89.4763	97.84	8.5538	2.1632e+03	457.1066
	QPSO-ASD	4.1979e+04	174.2667	51.90	3.5641	1.2231e+03	243.8654
3D	**QPSO-FS**	3.3864e+03	18.5203	99.942	1.8341	2.8899e+03	513.7782
	QPSO-CS	3.3744e+03	0.1250	99.698	1.0119	3.1113e+03	561.8822
	QPSO-SSD	3.3708e+03	0.0117	75.962	1.2017	2.3762e+03	824.5257

It can also be noted by comparing the results of the 2D and 3D tests that the 2D scenario converged significantly faster with lower computation consumption than the 3D scenario. The application of the path planner for the 3D tests took more iterations to converge than for the 2D case but eventually produced better and more robust trajectories, as indicated by the lower standard deviation of the iteration count value and the significantly smaller standard deviation for the best fitness value. Therefore, as the problem becomes more complex in higher dimensional searching space, it results in more computation consumption and slower convergence speed. Nevertheless, the SSD based path planner continues to produce more optimal solutions relative to the CS and FS based path planners.

3.5.2 Robustness Assessment for Multiple Test Scenarios

Ten different scenarios, each representing a different arrangement of start and destination positions, current fields and obstacles, were created and used to compare the relative robustness of the three control point placement schemes with the QPSO

planner. 1000 Monte Carlo simulation runs were performed with each placement scheme for each of the ten scenarios. The scheme that returned the minimum fitness value for each run was counted as having a 'win' while that with the maximum fitness value was counted as having a 'loss'. Information about the ten test scenarios is presented in

Table **3.3**. The first three scenarios, 1-3, were designed to have no obstacles, with the start/destination positions and current fields either randomly generated or kept consistent for all the 1000 Monte Carlo runs. Scenarios 4-10 all involve obstacles, the position of which, were modelled with uncertainty by defining a mean position and a confidence region for each obstacle. For scenarios 7-10, the obstacles' positions vary from one run to the next, with both the mean position and the confidence being randomly generated. It is worth noting that in such cases the sum of wins and losses may be less than 1000. This is because as either the obstacles' positions, or the start/destination positions, or both are randomly generated, it is possible that the start or destination point is located within an obstacle zone and hence no feasible path can be returned by the path planner. In such situations, no result will be recorded.

Table 3.3 Comparison of Performance of FS, CC/CS and SSD Schemes with QPSO Based Path Planner using Multiple Test Scenarios

Scenario	Start/ destination positions	Current field	Number of Obstacles	Control point placement mechanisms	2D		3D	
					Number of wins	Number of losses	Number of wins	Number of losses
1	Varying	Fixed	None	FS	206	471	165	730
				CC/CS	218	427	404	136
				SSD	576	102	431	134
2	Fixed	Varying	None	FS	325	471	5	966
				CC/CS	215	397	59	26
				SSD	460	132	936	8
3	Varying	Varying	None	FS	408	271	53	752
				CC/CS	155	564	347	190
				SSD	437	165	600	58
4	Varying	Fixed	Fixed	FS	134	617	173	740
				CC/CS	363	150	381	144
				SSD	393	123	446	146
5	Fixed	Varying	Fixed	FS	187	623	0	1000
				CC/CS	393	244	0	0
				SSD	403	116	1000	0
6	Varying	Varying	Fixed	FS	184	449	97	772
				CC/CS	240	197	334	166
				SSD	348	126	569	62
7	Fixed	Fixed	Varying	FS	264	398	31	955
				CC/CS	188	424	186	26
				SSD	458	87	783	19
8	Varying	Fixed	Varying	FS	176	389	150	741
				CC/CS	220	246	422	142
				SSD	340	101	428	117
9	Fixed	Varying	Varying	FS	185	450	61	677
				CC/CS	244	313	90	298
				SSD	450	116	849	25
10	Varying	Varying	Varying	FS	156	459	91	726
				CC/CS	244	181	356	199
				SSD	323	83	553	75

From the results presented in

Table **3.3**, for both the 2D and 3D environments, it can be seen that for all scenarios, the SSD scheme has a significantly higher chance (over 50% in most scenarios) of obtaining a better trajectory and is much less likely (around 10% or less) of returning a worse trajectory than the CC/CS and FS method. In fact, as shown in scenarios 3, 5 and 9, there is near or above 90% confidence that the SSD approach will achieve the best result. It should also be noted that in most of these cases, the performance of the FS method is poor and far from perfect.

In summary, these Monte Carlo tests of the path planning system demonstrate the improved performance and robustness of the proposed SSD method.

3.6 Chapter Summary

This chapter presents a novel SSD scheme and its subsequent integration with the QPSO based path planner. This scheme decomposes the search space into shell regions which gives freedom to the placement of the control points, while still restricting the search space to avoid overlapped searching. Simulation tests have been performed to generate an optimal trajectory with minimum time consumption for an AUV travelling through turbulent ocean fields in the presence of various obstacles with fixed/propagated uncertainty in position. Based on the results of these tests, the SSD scheme is shown to be capable of finding a more optimized trajectory than the CC method in 2D space and the CS method in 3D space. At the same time, the proposed SSD scheme shows faster convergence speed and uses less computation time than the FS method. These superiorities are verified with an AUV simulator which closely matches the vehicle operating in a realistic marine environment. In addition, Monte Carlo simulations demonstrate the robustness of the proposed SSD scheme compared with the previous methods.

One main limitation of the work completed thus far is that the path planner is only suitable for an AUV operating in an ocean environment where the currents are assumed to remain consistent during the mission period. However, real currents vary over time in both direction and strength, thus, the path generated for the earlier current map could well be disadvantageous as the mission progresses. In the next

chapter, as a natural extension of the above work, an on-line based SSD scheme will be developed and incorporated into a vehicle's guidance system to allow it to regenerate the trajectory during the course of the mission using continuously updated current profiles from on-board sensors.

Chapter 4

Dynamic Shell Space Decomposition for On-line Path Re-planning

This chapter extends the basic shell space decomposition (SSD) scheme for the problem of AUV path planning in quasi-static currents presented in Chapter 2 to account for the case of an AUV operates in a spatiotemporal ocean environment. A dynamic shell space decomposition strategy is developed and incorporated with an on-line planning system that adapt and regenerate the trajectory during the course of the mission using continuously updated current profiles from on-board sensors, such as a Horizontal Acoustic Doppler Velocity Logger. Example scenarios are presented that illustrate that with same amount of computational load, path planner based on the proposed dynamic shell space decomposition scheme utilizes path re-planning methodology is able to obtain a more optimized trajectory than one relying reactive path planning. Subsets of representative Monte Carlo simulations demonstrate the inherent robustness and superiority of the proposed planner.

4.1 Introduction

AUVs are increasingly expected to operate over long ocean transects in environments where they could possibly encounter strong, time-varying currents [136]. However, unless such disturbances are accounted for when planning a mission, the currents could have a profound, possibly adverse, impact on the mission duration

and battery usage, reduce the capability to conduct or complete the mission, and even endanger the safety of the vehicle [137]. Planning an AUV mission trajectory so that it is robust to current variability is a key element to guaranteeing the success of the mission and ensuring that it can be accomplished both safely and optimally.

Previous research of path planning for variable environment conditions have assumed that planning occurs with perfect knowledge of future environment changes [30, 35, 138]. The reality, however, is that it is difficult to forecast the environment accurately: ocean currents prediction is not perfect. Although existing predictive ocean models work reasonably well over short time periods and over large spatial scales, they lack sufficient accuracy to predict current behaviour over long time periods particularly in areas with low resolution information [139]. As a consequence of the current changes over the time period, the intended rendezvous, such as a surface vehicle, may have moved during the operation and unexpected obstacles may appear along the planned trajectory. For these reasons, any pre-planned trajectory based on forecast maps may turn out to be invalid or suboptimal.

Recent research makes use of continuous measurements of local ocean conditions from on-board current profiling sensors mounted in AUVs, e.g., a Horizontal Acoustic Doppler Current Profiler (H-ADCP) [117] and a Doppler Velocity Logger (DVL) for on-line path planning. A Long Range ADCP (75 kHz) is powerful enough to measure currents profiles up to 1000 meters in front of the AUV. During operation, the ADCP sends out and receives several acoustic pulses every second. An on-board computer processes the returned signal and real-time data of the magnitude and direction of the current throughout the water column is produced. This way, the changing ocean current structure can be observed nearly continuously while the vehicle is in motion. As the AUV proceeds along its trajectory, the path planner can re-generate the path using the latest updates for currents in the vicinity of the vehicle while the vehicle is en-route [140]. An example of this work is the absolute current velocity profiles captured from glider-based ADCPs by the Oregon State University (OSU) Glider Research Group. Results of missions performed in 2011 by OSU off the Oregon Shelf and in the Sargasso Sea are available in [141]. It is important to note that errors or missing data may occur in current measurement from on-board ADCP/DVL [142]. The ADCP manufacturer claims to have an accuracy of $\pm 1\%$ of

the measured current magnitude $\pm$ 5mm/s. The measurement accuracy is, however, also affected by sensitivity and alignment errors as described in [143]. This work primarily assumes that the measured current data is reliable and accurate. A future research activity can be dedicated to applying actual measured data to the proposed path planner and study how it affects the performance of the path planner.

It would be a waste of computation effort if at every sampling event the previously estimated path is abandoned and the path planner restarted afresh to find the new optimized path based on the updated information. This is especially significant as it can be computationally expensive to find the optimal path for an AUV in complex dynamic undersea environments with a large search space. Hence, it is more appropriate to plan the new trajectory starting from the previously estimated optimal trajectory. This previous path contains useful information that can be taken advantage of in subsequent trajectory re-planning. Since the ocean currents normally change gradually over time, the newly updated current map would most likely contain similar features as the previous recent map, thus the new path nodes have high possibility of being located in close vicinity to their respective previous locations.

Previous research, such as [41] and [42], looked at solving the dynamic path planning problem by using a hierarchical approach. In such an approach, the path planning system decomposes the searching space into several levels of resolution (higher resolution around the immediate current location of the vehicle, with decreasing resolution further away) and constructs an optimal path from the current location of the vehicle to the target location. The re-planning process uses previous information to refine the original cell channel in the immediate area of the path. This approach is inspired by the fact that it is reasonable, from a computational point of view, to generate a solution with greater accuracy locally where the vehicle's immediate reaction to an obstacle or a threat is needed. However, it may not be suitable in dynamic environments where the current field is continuously changing during the mission. The current changes in the environment force the planning system to frequently refine the original cell channel and update the local paths. The process of updating the local paths is computationally expensive because it often needs to update most if not all of the original cell channel. Dynamic path planning

may also be solved by correcting the previous path solution based solely on new information (such as newly detected obstacles). One approach by Wzorek et. Al. [39, 40] suggests that for those situations that need to be handled with urgency, only the colliding segments need be corrected. Given the range of typical AUV sonar sensor and the maximum AUV travel speed, there may be sufficient time, however, to generate a better solution that can take into consideration both new and previous information. Another approach based on Rapidly-exploring Random Tree (RRT) [43] proposes a re-planning strategy called Dynamic RRT [44, 45], where the RRT is repaired by pruning newly-invalid branches and then growing the remaining tree until a new solution is achieved. RRT strategy is suitable for handling dynamic environments with moving targets and obstacles, however, it has not been designed to handle optimal path planning with dynamic currents.

This chapter extends the SSD approach presented in Chapter 3 to address efficient online path re-planning issues. In particular, the new dynamic SSD algorithm, presented in this chapter does not explicitly require that the system states be able to reach the target set over the planning horizon. Instead, the planner can update the shell space decomposition as the system evolves. The performance of the proposed scheme is studied and compared with a path planner that relies on a naïve reactive path planning scheme. Moreover, a thorough robustness assessment is done with subsets of representative Monte Carlo simulations to compare the proposed path re-planning scheme based on dynamic SSD with the reactive path planning scheme.

The rest of this chapter is organized as follows. Section 4.2 provides an overview of the dynamic on-line AUV guidance system and the optimization problem. Section 4.3 introduces the preliminary method based on reactive path planning and discusses the proposed path re-planning scheme based on dynamic SSD. The simulation tests under various scenarios are presented in Section 4.4, followed in Section 0 by a set of robustness assessments using Monte Carlo trials. Concluding remarks are then presented in Section 4.6.

4.2 Problem Formulation

The marine environment poses a rich field of challenges for AUV path planning

systems, such as ocean currents, irregularly shaped terrains and dynamic and uncertain obstacles [144, 145]. Dynamic path planning is a continuous process which generates a new path by adapting previously computed paths for onwards travel. This concept is illustrated in Figure 4.1. The planner continually computes and refines its path during the course of the mission. The time $t_{|P|}$ at the end point of this trajectory is the planning time horizon. In the figure, at time t_i, the vehicle starts to execute the trajectory planned in the preceding time step while it computes a new trajectory that will start at $t_i + \Delta_t$ whereupon the updated trajectory is sent to the vehicle's controller. The committed section is a portion of the path for the time period $t_i < t < t_i + \Delta_t$. The modifiable section of the trajectory is a portion of the path from the time period $t_i + \Delta_t$ to $t_{|P|}$. Δ_t is the time interval between two consecutive path updates, an update trajectory is sent to the vehicle's controller every Δ_t until the end of the mission. Δ_t specifies the maximum time available for the planner to compute its path update. During this time horizon, the planner can incorporate any new information about the environment that becomes available. The trade-off between time available to plan and adaptability is important. One would want the committed sections of the planned path to be short in a highly dynamic environment. However, that in-turn shortens the time available to plan. A short planning window requires faster planning algorithms. Hence, dynamic planning algorithms must be fast enough to compute new plans within the time constraints.
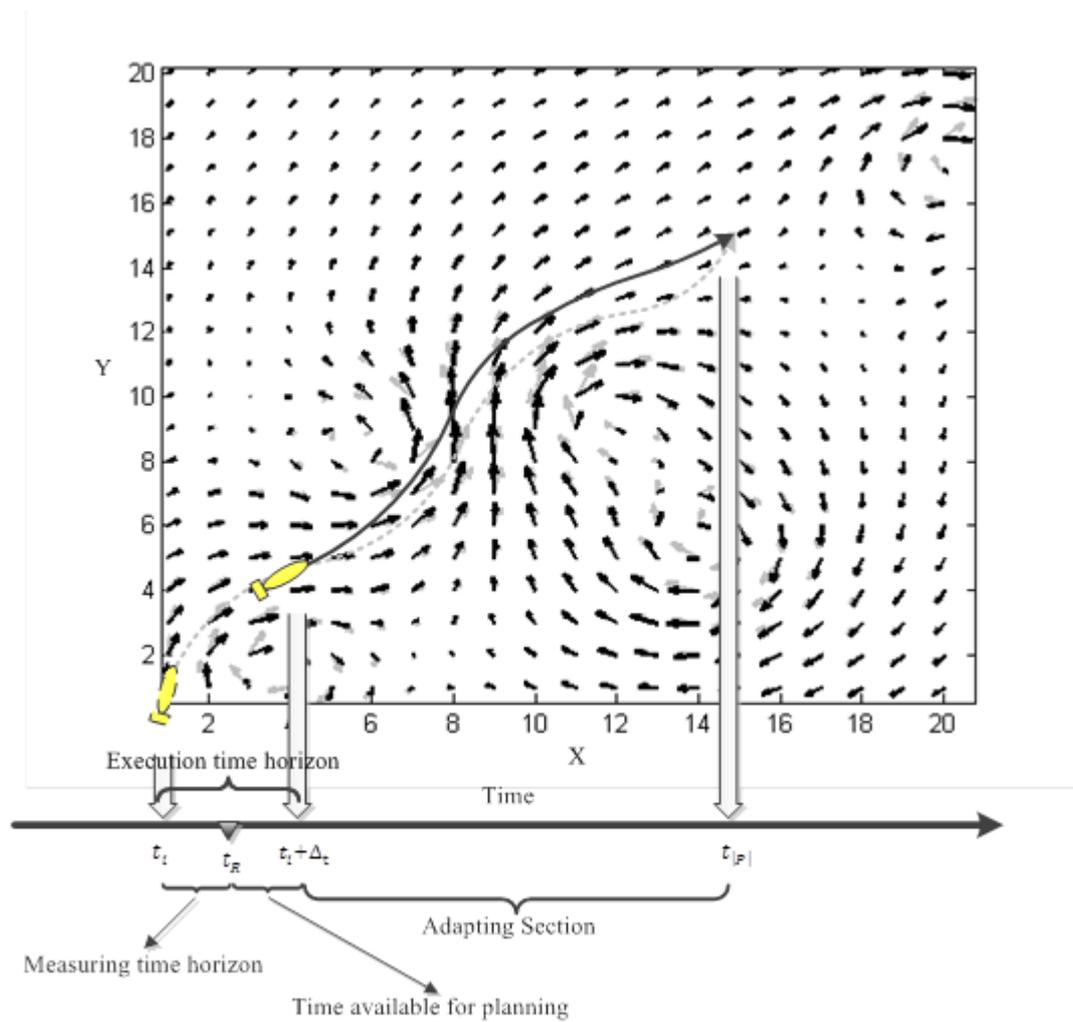
Figure 4.1 Mission execution using the dynamic path planning system. The AUV is executing a path previously planned at t_i which is shown as a grey dashed line, while it computes a new path (black line) starting at $t_i + \Delta_t$ based on newly acquired measurements and current predictions.

Figure 4.2 contains a schematic representation of the AUV dynamic guidance system. Given a candidate path and the current environment information, the state predictor provides the planner with predicted future states of the system using a dynamic model. By using this prediction, the AUV path planner computes a path that optimizes an objective function. The generated path profile is then sent to the guidance controller to generate the guidance commands for the vehicle. This sequence is repeated in the next time period.
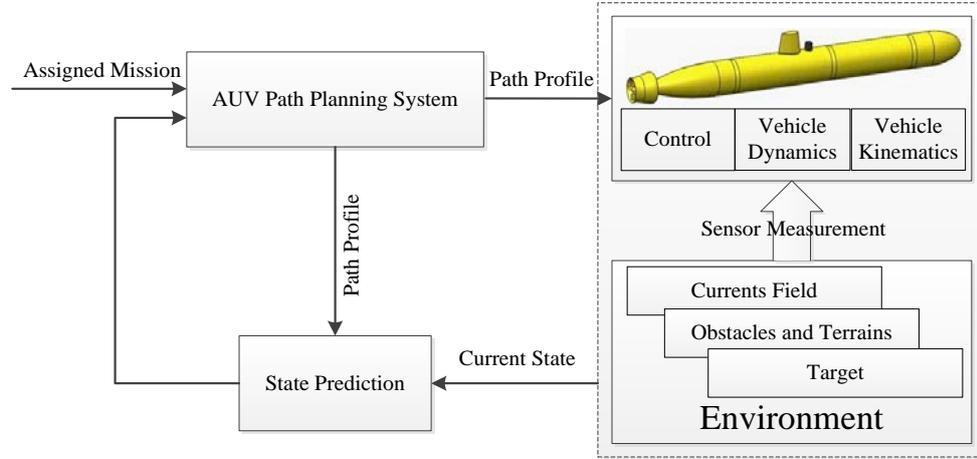
Figure 4.2 AUV dynamic guidance system.

Consider an AUV traveling with constant water-referenced velocity $\mathcal{V}$ with initial condition $\mathcal{A}_0 = (r_0, w_0, t_o)$, where $r_0 = [x_0, y_0, z_0]$ defines the vehicle's position in a spherical coordinate frame, and $w_0 = [\phi_0, \theta_0, \psi_0]$ defines the vehicle's Euler angles: roll, pitch and yaw. The objective of the AUV path planning system is to find the optimal set of paths $\mathbb{p}^\dagger$ among the set of all feasible paths $\mathcal{P}$ for the AUV to travel through the spatiotemporal ocean environment and arrive at the destination $\mathcal{A}_\mathcal{G} = (r_\mathcal{G}, w_\mathcal{G}, t_\mathcal{G})$ with minimal time usage. The ocean environment V_c is modelled as a time varying current field occupied with terrain $\mathbb{T}$ and obstacles O, the position of which may be dynamic and uncertain.

In this study, the potential AUV trajectories are defined in space by B-Spline curves obtained from a set of control points $P = \{p_1, p_2, \dots\}$. The fitness value of each path is measured by using uniform subdivision whereby the continuous spline curves are discretized by a sequence of waypoints along the path $\mathbb{p} = \{\wp_1, \wp_2, \dots\}$. The travel time T along a given path $\mathbb{p}$ is the sum of time $t(\wp_i)$ required to cover each of these segments constituting the path.

Assumption 4-1: The vehicle's water-referenced velocity is assumed to be constant. Since this velocity is proportional to the cube root of the thrust, equivalently, the vehicle has constant thrust power and thus the energy consumed for a path is a constant multiple of the distance travelled.

Assumption 4-1 is derived from the observation that it is common on AUV missions, to set the thrust power to be constant. The resultant ground velocity of the vehicle V_a

is resolved in a 2D horizon using the water-referenced velocity of the vehicle $\mathcal{V}$ in the geographical frame and the velocity of the water current V_c.

Assumption 4-2: The corresponding angular velocity required for the vehicle to execute the B-spline curve is within minimum and maximum bounds at all times.

The dynamic constraints of the problem are determined by the capabilities of the autonomous vehicle [146, 147] and the curvature of the paths represented by the spline curve [148]. This problem has been resolved in previous work. The work presented in this chapter primarily focuses on a high-level planning architecture with simplified dynamics enabling it to find the optimum trajectory adjustments to take advantage of the favourable currents field. Previous work has been done on studying the full dynamics of the system and the control strategies that drive the vehicle to the desired planned trajectories [61, 149-151]. The work presented in this chapter makes use of these previous works to implement the trajectory generation method. The curvature of the b-spline paths is achievable with the angular velocity and radial acceleration constraints of the vehicle. This simplification, along with that given by Assumption 4-1, is critical to ensure the controller is able to overcome current disturbances and to accurately drive the AUV to a planned waypoint.

Therefore the path planning problem is formulated as the following optimization problem:

$$\mathbb{p}^{\dagger} = \frac{argmin}{\mathbb{p} \in \mathbb{p}} \, F(\mathcal{A}, V_c, O, \mathbb{T}, \mathcal{V}),$$

with $\quad F = \sum_{1}^{|\mathbb{p}|} T\left(\wp_i^{t_i}\right) = \sum_{1}^{\lambda} \left| \left(\wp_{i+1}^{t_{i+1}}\right) - \left(\wp_i^{t_i}\right) \right| / |V_a|$

$$\text{s.t.} \quad M(\eta, \dot{\eta}) = 0,$$

(4-1)

$$\wp_0^{t_o} = \mathcal{A}_0, \wp_\lambda^{t_\lambda} = \mathcal{A}_\mathcal{G}$$

$$\forall \, i \in \{0, \dots, \lambda\} \quad \wp_i^{t_i} \notin O(t_i) \cup \mathbb{T}$$

where $\wp_i = (x_i, y_i, z_i, \phi_i, \theta_i, \psi_i, t_i)$ is a waypoint along the path $\mathbb{p}$, and $M(\eta, \dot{\eta})$ is the vehicle's kinematic model.

4.3 Dynamic Shells Space Decomposition for Re-planning

This section expands upon the SSD scheme for Spline based evolutionary path planning of AUVs operating in turbulent, cluttered and uncertain environments introduced in [58, 152]. In that section, the authors presented a SSD that decomposes the search space into multiple concentric shell regions radiating out from the start point to the destination, with one or more control points allocated to each shell region. The trajectory is then generated from this set of control points using Splines. This arrangement facilitate more efficient searching and greater flexible placement of the control points, but still restricts the search space for each control point to its respective regions to save computation time.

The dynamic SSD strategy proposed in this section is designed to cope with the continual path updates while the vehicle is conducting its mission. Dynamic path planning is a continuous process. A diagram describing the concept of on-line path planning based on the dynamic SSD is shown in Figure 4.3. The planning process starts with the off-line path planning process to generate an initial population $\mathcal{P}_0$ and find the first best candidate path $\mathbb{p}_0^\dagger \in \mathcal{P}_0$ depicted as the black path in Figure 4.3. The vehicle continually executes the updated path during the mission. The planner continually updates its path while the vehicle is approaching the target position. The planning problem in each cycle is a similar problem to that in the previous cycle.
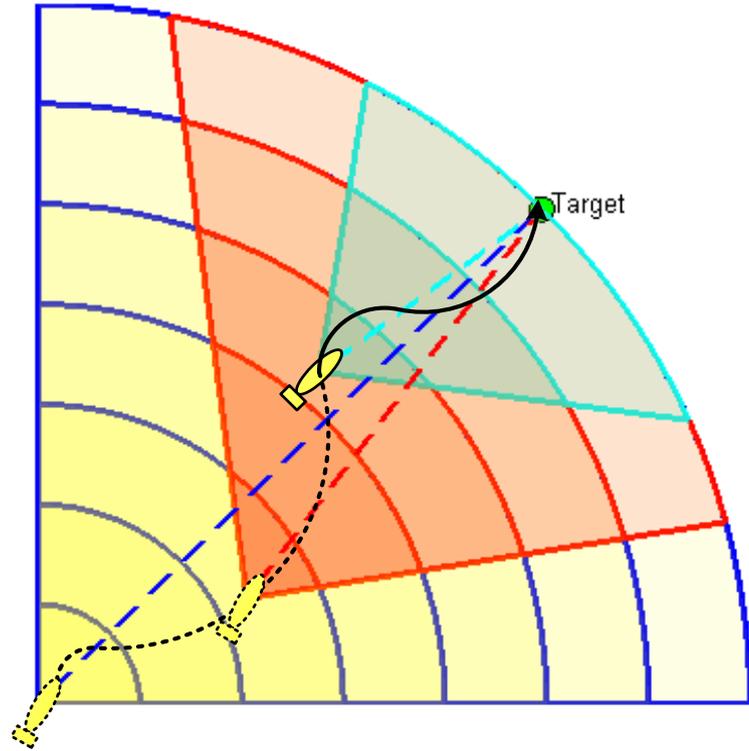
Figure 4.3 On-line path planning based on the dynamic SSD scheme.

This work expands upon the prior work by providing a framework that reconstructs the shell space explicitly using the original decomposed shell spaces for subsequent planning. To initialize a new population for the next planning cycle, the lower and higher bounds of the shell regions first need to be identified. Those annuli which are behind the vehicle's predicted current position are then removed, and those annuli located ahead of this position trimmed back, as shown in Figure 4.3, to form the new searching region relative to this position.

In this section, two online path planners are proposed based on reactive path planning and path re-planning schemes, respectively, that utilize the DSSD scheme.

4.3.1 Primary: Reactive Path Planning

Let p represents one of the control point of the ith planning iteration, p^i is expressed in log-polar coordinates as $p^i(r^i, \theta^i)$, every point sh ould be located within its respective search region. Let $\mathcal{B}^i(\mathcal{B}_r^i, \mathcal{B}_\theta^i)$ be the search sector of the previous planning iteration, where $\mathcal{B}_r^i, \mathcal{B}_\theta^i$ are the bounds for r^i and θ^i, respectively. Let $\mathcal{B}_r^i = [S_r^i \ P_r^i]$, and $\mathcal{B}_\theta^i = [S_\theta^i \ P_\theta^i]$, where S_r^i, S_θ^i are the starboard bounds and P_r^i, P_θ^i are the port

bounds. Each point $p(r, \theta)$ is generated from:

$$r^i = S_r^i + Rand_i^r (P_r^i - S_r^i) \,, \tag{4-2}$$

$$\theta^i = S_\theta^i + Rand_i^\theta (P_\theta^i - S_\theta^i) \,, \tag{4-3}$$

At each planning cycle, the segments of the annuli located ahead of the vehicle's predicted position (x^{i+1}, y^{i+1}), i.e., the new start point, are defined and the control point population reconstructed. Let $p^{i+1}(r^{i+1}, \theta^{i+1})$ represent one of the control points of the $(i + 1)^{th}$ planning iteration. $\mathcal{B}^{i+1} = [\mathcal{B}_r^{i+1}, \mathcal{B}_\theta^{i+1}]$ can be defined as:

$$\mathcal{B}_r^{i+1} \in \mathcal{B}_r^i$$

$$\forall r^{i+1} \colon \theta^{i+1} \in \mathcal{B}_\theta^{i+1}[S_\theta^{i+1} \; P_\theta^{i+1}],$$

$$\text{with} \quad S_\theta^{i+1} = arcsin \frac{y^{i+1}}{r^{i+1}}, \tag{4-4}$$

$$P_\theta^{i+1} = arccos \frac{x^{i+1}}{r^{i+1}}.$$
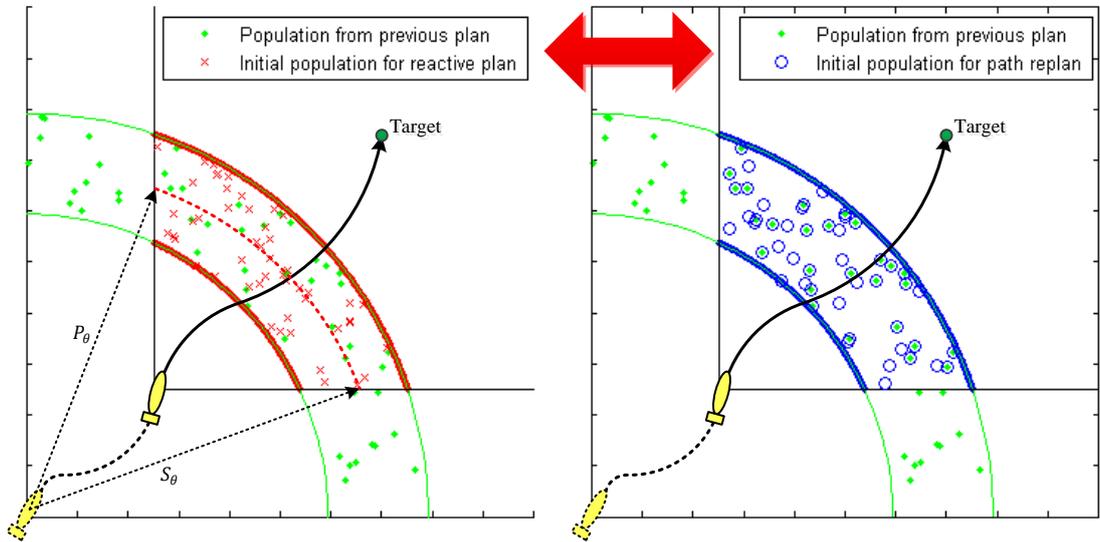
Figure 4.4 Comparison of the initial populations (paths) generated for the reactive path planner (left) which discards all previous solutions and the proposed path re-planner (right) which reuses the subset of the population from the previous plan.

4.3.2 Path Re-planning with Reuse of Previous Solution

The re-planning approach attempts to preserve some information of past solutions and use it as the basis to compute new solutions (as shown in Algorithm 4.1). This

approach is based on the analysis that the new path is slightly different from the previous problem and it takes advantage of evolutionary algorithms that several candidate solutions are available at any time during the optimization process.

Algorithm 4.1 Path Re-planning with Reuse of Previous Solution

Initialization

1. Input: Final population P_0 and the best candidate path $\mathbb{p}_0^{\dagger} \in P_0$ from the off-line path planner.

Iteration Loop

2. Identify the new searching region $\mathcal{B}^{i+1}$ according to vehicle's predicted starting position for the next plan.

3. Generate a new population P^{i+1} from the current population P^i. The control points in the current population are passed through a filter, those still within the new searching region $\mathcal{B}^{i+1}$ will be reserved while those out of the new search region will be relocated (regenerated) inside the new search region. All the paths in the current population will begin at the new predicted start position.

4. Run the static planning algorithm continuously to update the population and to find the best candidate path.

5. Send the updated candidate path to the vehicle guidance system once the vehicle reaches the time interval Δ_t.

6. Update the estimates of the locations of sites in the environment.

7. Return to step 2.

As shown in Figure 4.4, the reactive path planner generates a completely new population represented by red crosses (previous solution represented by blue points will be discarded), and reruns the optimization algorithm to find a new solution. In contrast, the re-planning approach attempts to preserve some information of the past solutions (blue points with red circles) and uses it as the basis to compute new solutions.

4.3.2.1 **Dynamic Shell Space Decomposition with Dynamic Obstacles**

The status of the obstacles: coordinates, moving speeds and directions, are measured by the on-board sensors with a certain amount of uncertainty modelled as Gaussian

distributions. This current state information of the obstacles is continuously measured from on-board sensors and periodically fed back to the state predictor, the state predictor provides the dynamic path planning system with predicted future states of the obstacles. The state predictor assumes the obstacles will move independently during the next execution time horizon, Δ_t, at their measured rates and directions.

4.3.2.2 Dynamic Shell Space Decomposition with Moving Target

Since the ASD scheme decompose the search space into shell regions expending out from the start point to the target, moving of target may need modification to the decomposed shell accordingly. One case is that the target moved closer to the start point, if this movement is within the range of one shell, no additional process is required. If the target moves more than one shell into the region of adjacent shell, then the most outside shell becomes redundant and will be removed (see Figure 4.5(a)). Another case is the target moves further and get outside of the shell regions, in this case, an extra shell should be added to expend the origin space (see Figure 4.5(b)), and meanwhile, another control point is added to construct the path, which will be placed to search within this new shell.

(a)
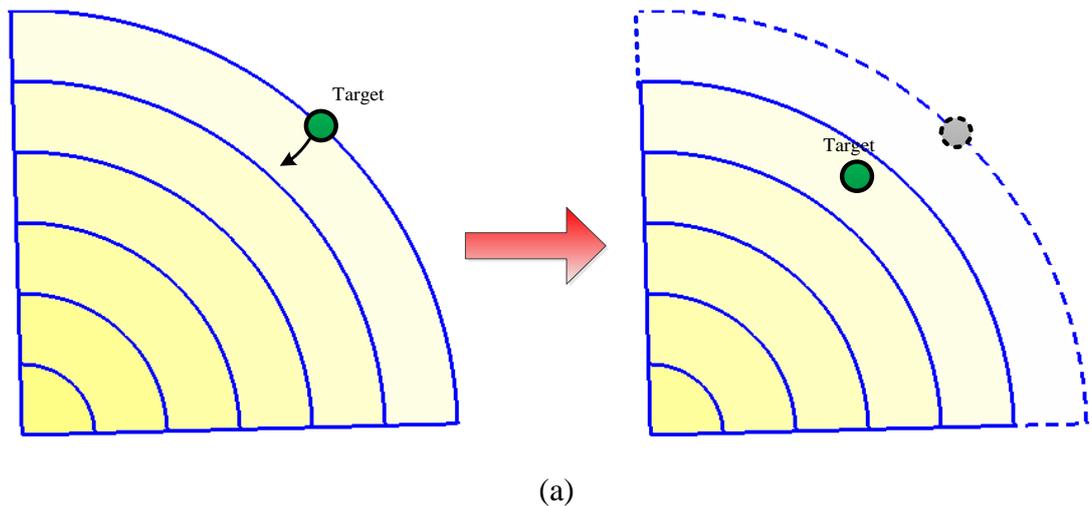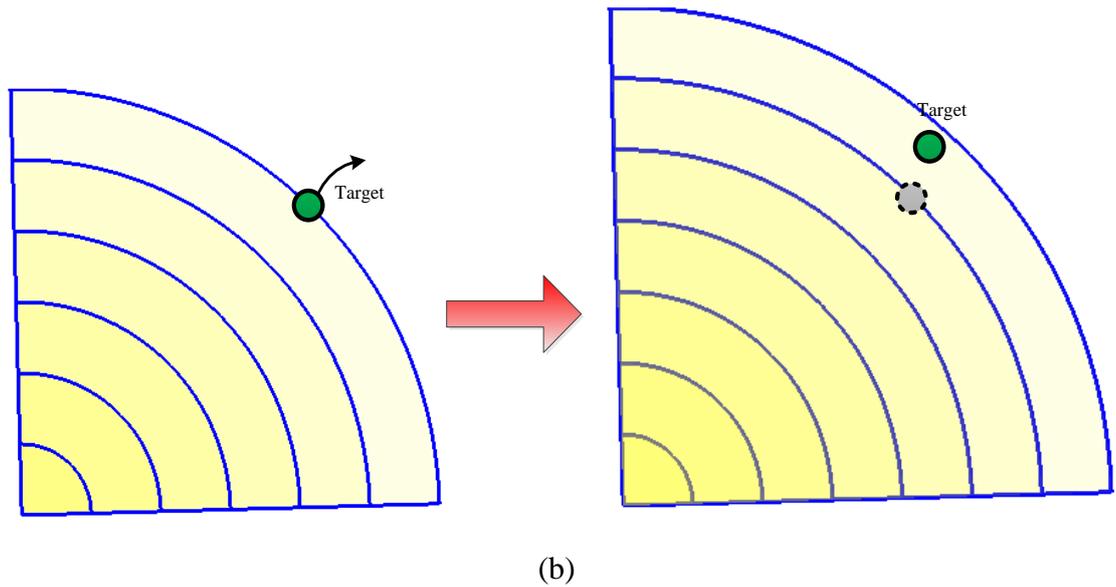
(b)

Figure 4.5 Reconstruct the shell space according to the moving target: (a) target moves closer into the adjacent shell, (b) target moves further and get outside of the shell regions.

4.4 Simulations

In this section, four case studies are conducted to compare the performance of the reactive path planner and the path re-planner methods. The mission objective is to find the minimal time trajectory through the spatiotemporal currents field to the target position. The first case study is a simple ocean environment with a variable spatiotemporal current field. The next two case studies deal with an environment where either dynamic obstacles represented with uncertainty in their positions, or moving rendezvous points are present. The fourth case study demonstrates a complex scenario, with irregularly shaped terrains, dynamic uncertain obstacles and moving rendezvous points.

4.4.1 Simulation Setup

In these case studies, the path planner runs the QPSO based optimization algorithm with a population size of 100. Prior to the start of the mission, the off-line planning system generates an initial solution based on available information in 50 iterations. During the simulation, the dynamic on-line planning system recursively generates new solutions based on updated information in 10 iterations, this number of iterations is intentionally set to be small to accommodate the limited on-board computing

resources on an AUV. The time interval Δ_t between two adjacent path updates is 64 seconds. In the following plots given in Figure 4.6~Figure 4.10 showing the simulation results for each case study, the original off-line path planning is shown in red, while the subsequent first, second and third planning generations are shown in green, blue and purple, respectively.

The contract-expansion coefficient β of QPSO is set to linearly decrease from 1.0 to 0.5 over the iterations. Each individual B-spline path is formed by 15 control points for Case 4-1 and Case 4-2, and formed by 12 control points for the other cases, this quantity of control point can be tuned according to the complexity of the environment and computation capacity available. The B-spline path is uniformly subdivided by 1,152 internal knots, the quantity of the points is conservatively chosen such that it is large enough to guarantee that the spline formed paths closely conform to the intended paths. The current field for the scenario reported in this study is generated from a random distribution of 50 Lamb vortices represented by a 100×100 grid within a 2D spatial domain (see the Subsection 2.3.1 for the definitions of the variables and equations). Equations (2-4)-(2-7) were then used to compute the spatiotemporal currents field.

4.4.2 Case 4-1: Spatiotemporal Current Field

The scenario in this case study, is that an AUV is travelling through a variable spatiotemporal currents field from the start position (1,1) to the destination position (500,500). The vehicle's water-referenced velocity is set to be 3m/s (3m/s is reasonable for long-range operation, and this is keep the same in the following case studies). The current field is updated every 4s, and keeps changing during the AUV's travelling period. As the AUV moves along the path provided by the off-line generated plan, it periodically (every 64s for this case study) uses the H-ADCP to infer a new current profile ahead of the vehicle and passes the new map onto the dynamic planning system to generate a new path. The same procedure is repeated until the AUV arrive at its destination point.

Figure 4.6 presents the first case study revealing the on-line path planner performance based on the reactive path planning method and the path re-planning method, respectively. Figure 4.6(a) displays the result of trajectories generated by the

path re-planning method during the mission. As demonstrated, the re-planned trajectory is slightly different from the previous planned trajectory, and it would seem able to make better use of favourable currents within the updated currents profile compared to the previous planned trajectory. It can be seen that in process of the third re-plan cycle, the new trajectory (in purple) no longer makes a detour at the end of the trajectory as per previous generated trajectories. This is because if the AUV had followed the green trajectory generated from the previous re-plan cycle it would have faced severe drift from local adverse current flow that eventuated after the second current map was generated. The reactive path planner shows a similar performance in Figure 4.6(b).

(a)

(b)

(c)
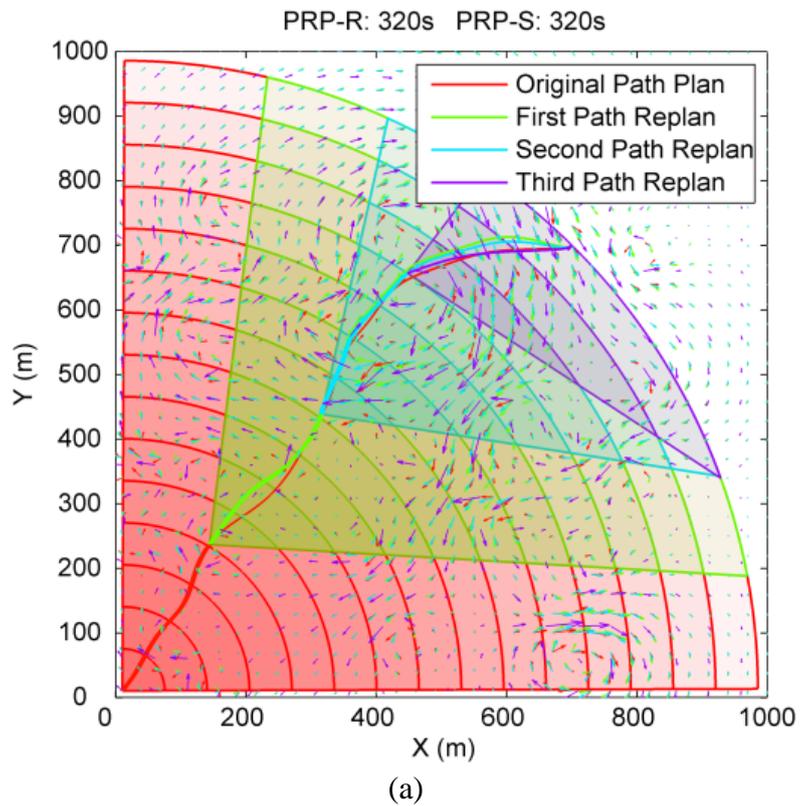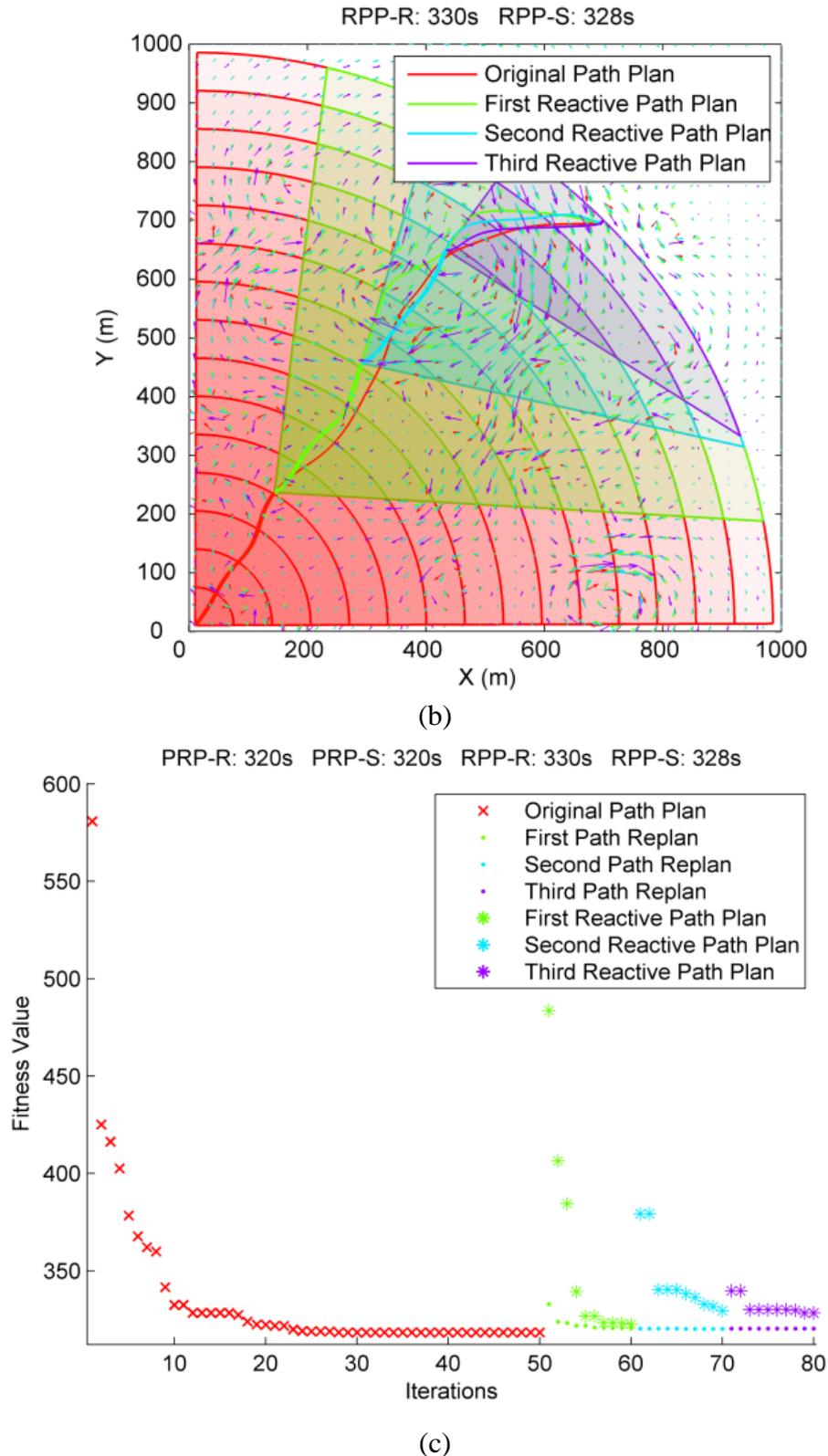
Figure 4.6 Comparison of results produced by the path planner based on the dynamic shell space decomposition scheme: (a) paths planned on-line with re-planning scheme, (b) paths planned on-line with reactive planning scheme, (c) convergence curve of best fitness values.

The convergence curves showing the best fitness values with the corresponding numbers of iterations is displayed in Figure 4.6(c). Since the proposed optimization is a minimization problem, the smaller the best-fitness value, the better the performance. The red cross with the first 50 iterations displays the convergence curve of the off-line planning process, followed by convergence curves for the recursive dynamic on-line planning process every 10 iterations. By comparing the convergence curves of the dynamic planning schemes based on reactive planning and re-planning, it is clear that the reactive planning scheme has significantly slower rate of convergence relative to the re-planning scheme. The reason for the reactive planner's slower rate of convergence is evident, in each dynamic planning cycle, the reactive path planner generates a completely new population and reruns the optimization algorithm to find a new solution leading to inefficient searching, whereas in the case of the re-planning approach, a subset of the population from the previous plan is used as the basis to compute the new solution, thus speeding up the search process. As a consequence, at the end of each cycle, the proposed re-planning scheme achieves better results than the reactive planning scheme. The execution time for the trajectory planned with the reactive planning scheme in simulation (RPP-S) is 328s whereas that for the trajectory planned with the path re-planning scheme (PRP-S) is reduced to 320s. When the execution time of these trajectories are tested with a currents map acquired with a high update rate to more closely emulate the slow but continuous changes of the real ocean environment, the vehicle, as shown Figure 4.6(c), takes less travel time using the proposed path re-planning scheme: 320s (PRP-R) vs 330s (RPP-R).

4.4.3   Case 4-2: Spatiotemporal Current Field and Dynamic Obstacles

This scenario contains seven moving obstacles with varying levels of position uncertainty within a variable spatiotemporal current field identical to that used in Section 4.4.2. Case 4-1. Note that the obstacles are intentionally placed at locations to potentially block the optimum trajectory found in Case 4-1. The obstacles are configured individually to move independently at various rates (-0.05m/s to 0.05m/s along both X and Y directions). Their uncertainty in position is assumed linearly propagated with time at rates randomly chosen between 0.005m/s to 0.035m/s. The

positions of the obstacles are represented by the mean and variance of their uncertainty distribution. As mentioned in 4.3.2.1, obstacles are regularly measured from the on-board sonar sensors at the same time as the currents field is measured, and these obstacles are assumed to move at their measured rates and directions during the next execution period Δt.

Figure 4.7 displays the results for Case 4-2. Since the propagation of the uncertainty is assumed linear with time, it is expressed in the plots shown in Figure 4.7(a) and (b) as a proportional increment in the collision boundary encircling the object. Corresponding colours are used to represent the successive growth in the collision boundaries at the start of each planning cycle and the trajectory path update for that cycle. At each discrete time point, if the vehicle location does not cross inside the corresponding obstacle boundary, no collision will occur. It is seen in this simulation that both trajectories generated with reactive path planning and path re-planning schemes are able to navigate the AUV around dynamic obstacles.

(a)
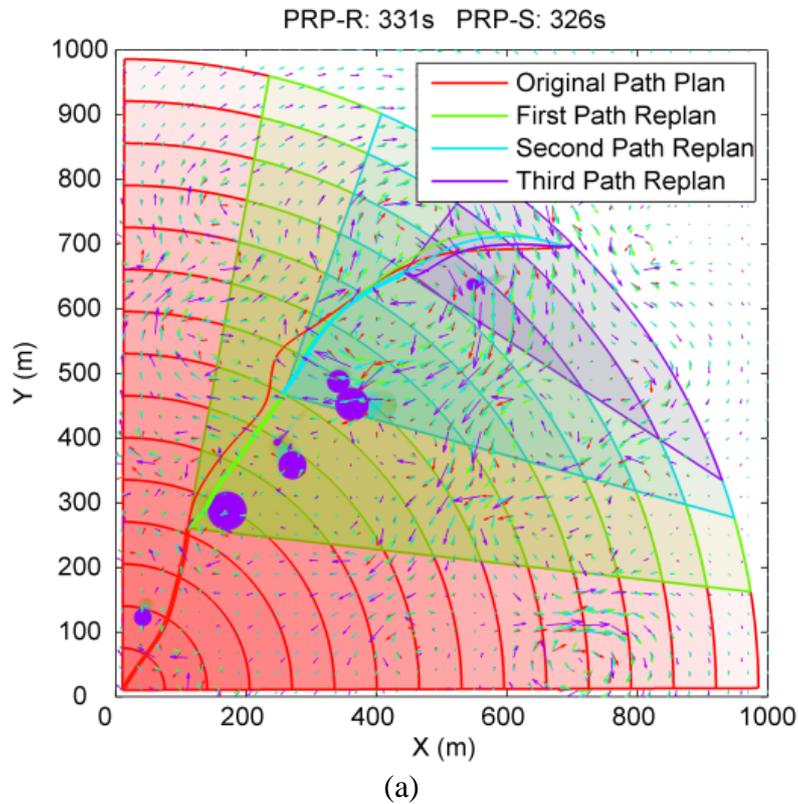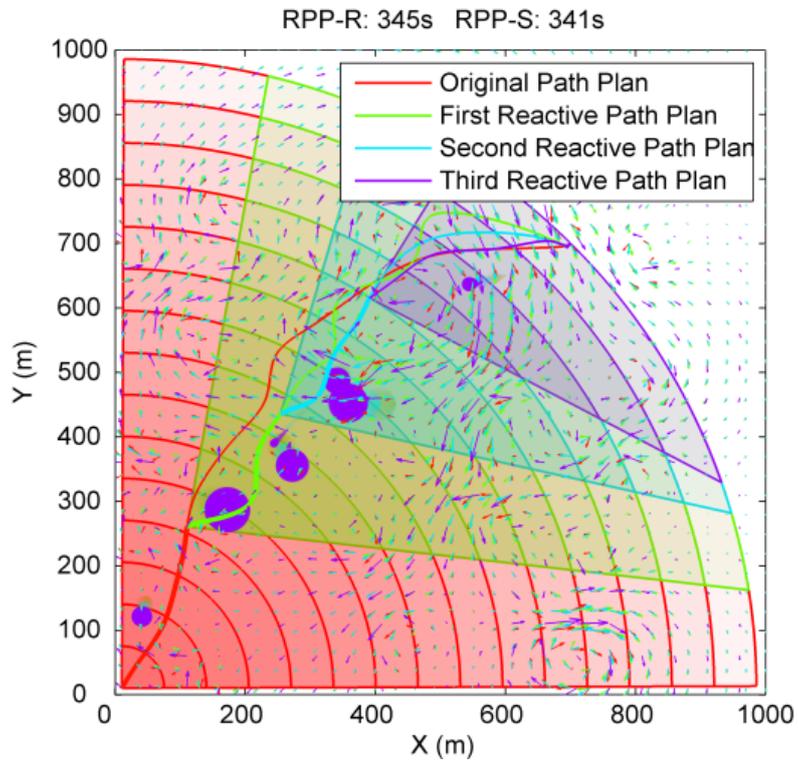
(b)
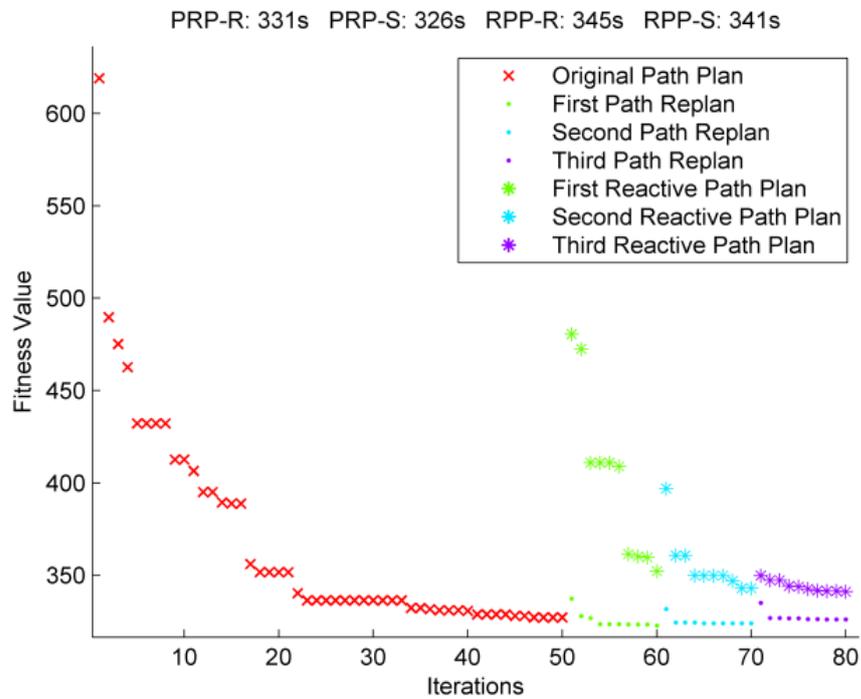
(c)

Figure 4.7 Comparison of results produced by the path planner based on reactive path plannnig and path re-planning schemes with dynamic obstacles: (a) paths planned on-line with path replanning scheme, (b) paths planned on-line with reactive path planning scheme, (c) convergence curve of best fitness values.

By comparing the execution time of these trajectories shown in Figure 4.7(a) and (b), it can be seen that the vehicle taking the trajectory provided by the path re-planning scheme consumes less travel time than for the trajectory provided by reactive path planning scheme. The reason can be found from Figure 4.7(c) which reveals the convergence rates of the best fitness values with the numbers of iterations. It is clear that the convergence curve of the planner based on path re-planning with reuse of previous solution is always lower than that of the planner based on reactive path planning, so there is indeed faster convergence rate using the proposed path re-planning scheme than using the reactive path planning scheme.

4.4.4   Case 4-3: Spatiotemporal Current Field and Moving Rendezvous

Case 4-3 considers the scenario where the AUV is instructed to rendezvous with a moving target, such as a mother ship or ASV. Once again the current flow is the same as for Case 4-1. As defined in Chapter 2. Subsection 2.3.1-c, the model of the rendezvous target includes three elements: position, velocity vector, and growth rate of relative heading angle. The initial position and orientation of the target is at (500,500) and -1.25 π, respectively. The standard deviation of the Gaussian noise added to the x and y positions of the target representing the changing of these positions is selected to be 0.01m/s, and that of the Gaussian noise added to orientation of the ASV representing the changing of this orientation is selected to be 0.005π rad/s for this case.

Figure 4.8(a) and (b) show the corresponding trajectories, as generated by these two strategies, and tracked by the vehicle during the mission. As can be seen, both path planning schemes succeed in generating trajectories that allow the AUV to intercept the ASV. By comparing the time consumed for the corresponding optimised trajectories produced by the reactive path planning and path re-planning schemes, it can be seen that the trajectory planned with the path re-planning scheme continually adapts to the predicted position of the target while preserving the path shape that is synchronized with the direction of the flow. In this way, the AUV is always located in a position where the current field propels it to move forward to the goal. This allows the vehicle to take significantly less travel time than that consumed by taking the trajectory provided by the reactive path planning scheme. Convergence curves

display in Figure 4.8(c) showing that path re-planning scheme has better convergent features than the reactive path planning scheme, which further verifies that the former is more likely to generate a more optimized solution.
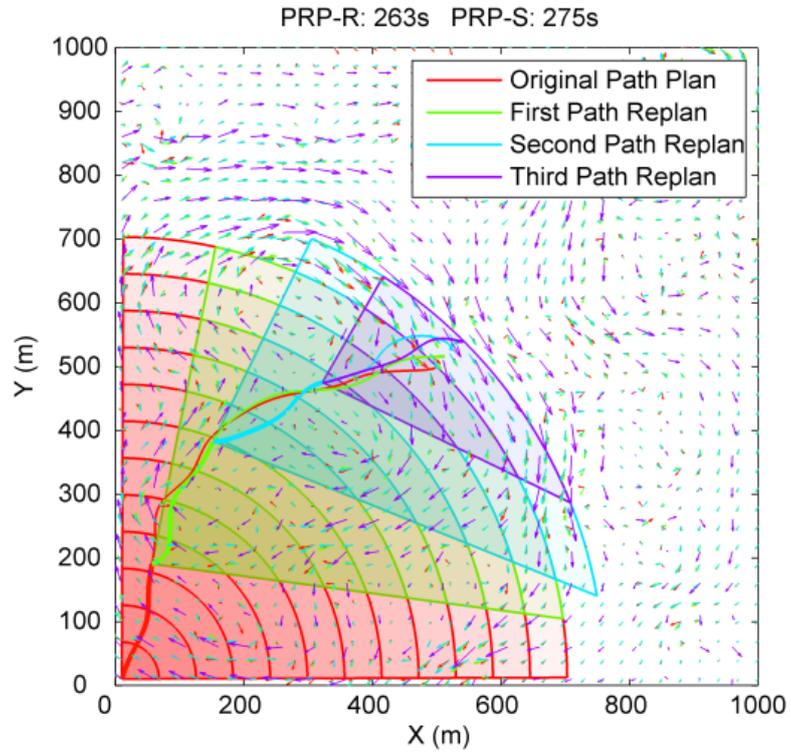
(a)
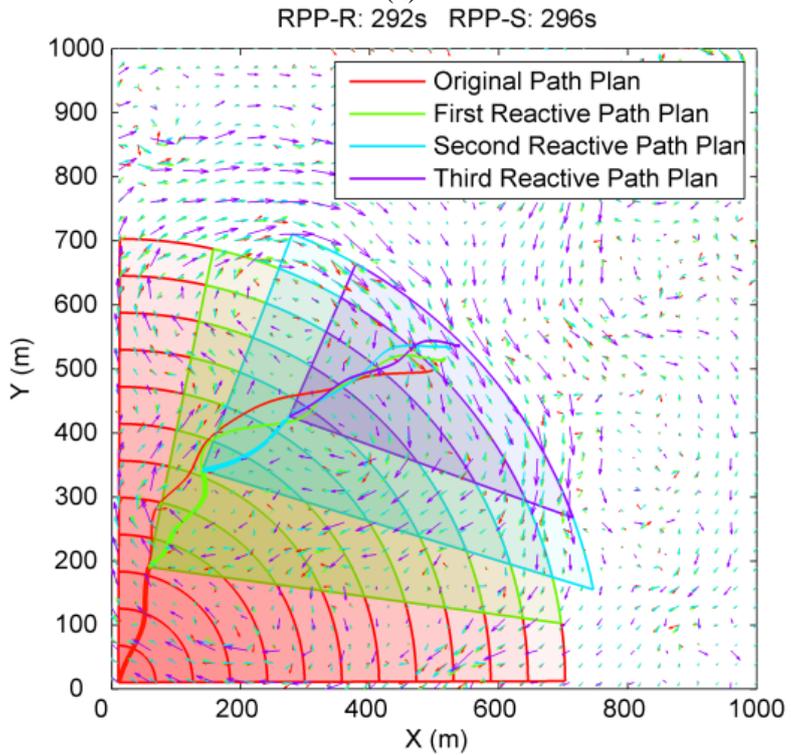
(b)
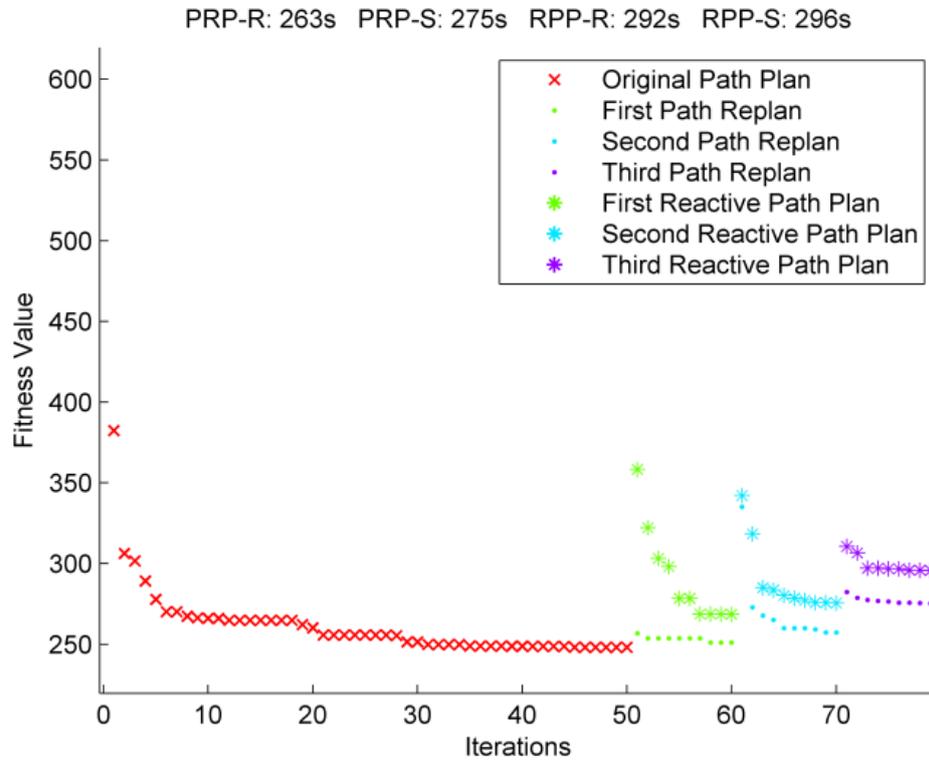
(c)

Figure 4.8 Comparison of results produced by the path planner based on reactive path planning and path re-planning schemes with moving target: (a) paths planned on-line with path re-planning scheme, (b) paths planned on-line with reactive path planning scheme, (c) convergence curve of best fitness values.

4.4.5 Case 4-4: With Unexpected Dynamic Obstacles

Case 4-4 studies the scenario where the AUV has limited ranges of on-board sensors for detection of obstacles. With the current environment and obstacle model being the same as for Case 4-3, the problem becomes more challenging since at each path refresh cycle, the newly planned path should be able to take care of the appearance of unexpected obstacles. In this scenario, the range of on-board sensors is described by a circle centred at the location of the AUV and of radius 25m, only the obstacles within the range of sensors are considered.

The paths produced by the planner based on the dynamic SSD scheme utilizing the reactive path plan and path re-plan methods are plotted in Figure 4.9. The time-correlated locations of the vehicle and sensor ranges at each path refresh cycle are indicated by the dotted lines with same colour. The obstacles within the AUV's sensor range at each path refresh cycle are plotted using the same colouring scheme as above in Figure 4.9, only these obstacles will be taken into account for path

planning in each cycle.

(a)

(b)
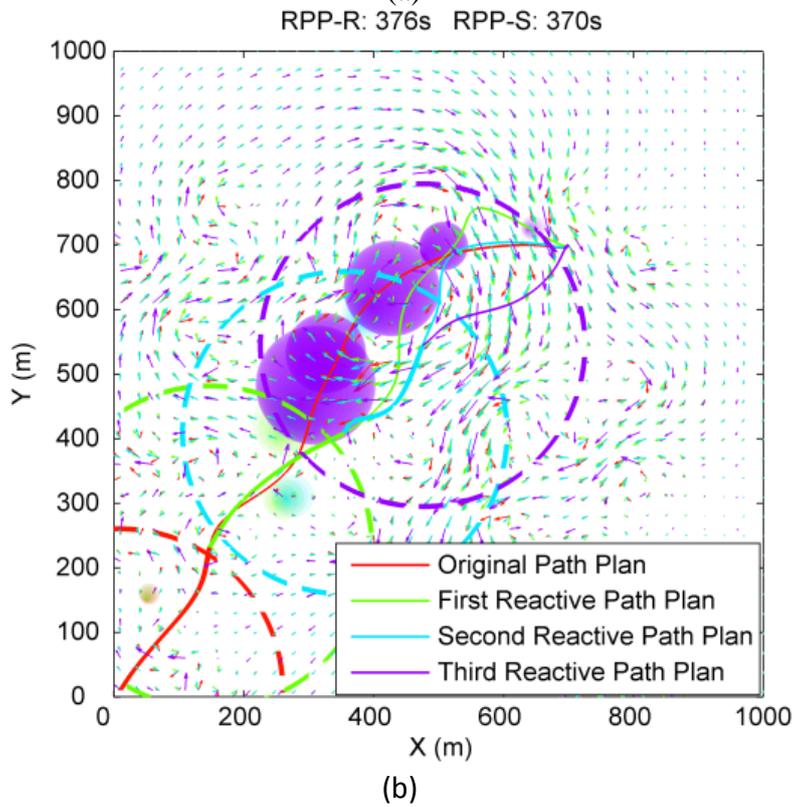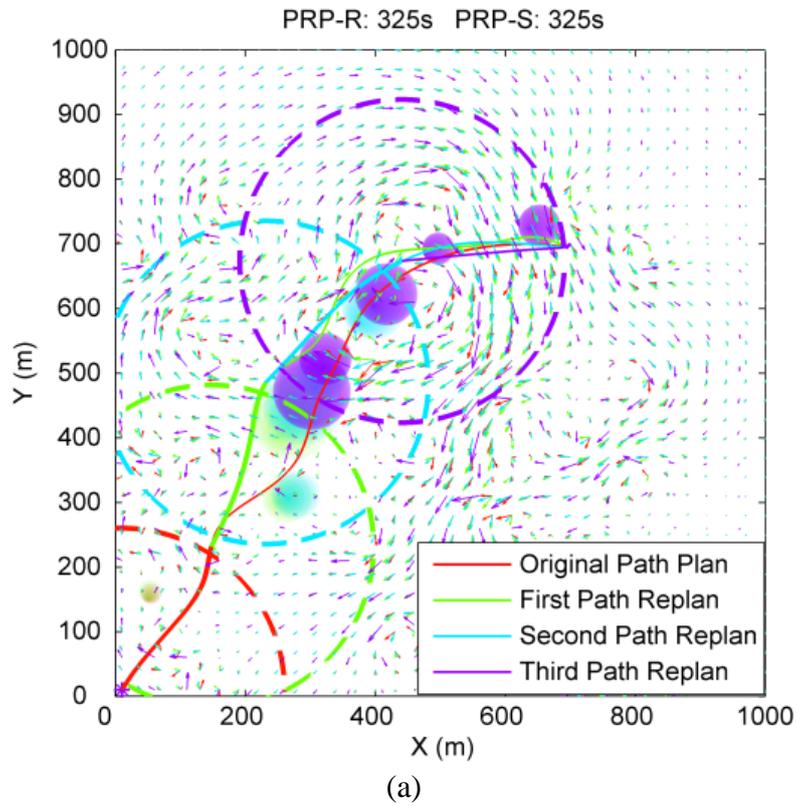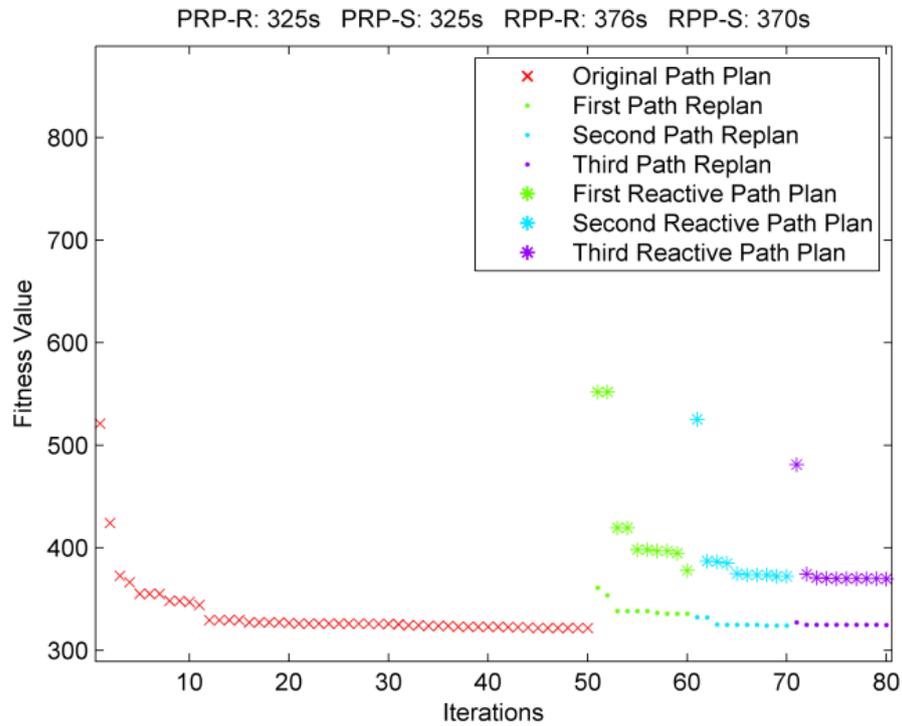
(c)

Figure 4.9 Comparison of results produced by the path planner based on the dynamic shell space decomposition scheme with dynamic obstacles: (a) paths planned on-line with path re-planning scheme, (b) paths planned on-line with reactive path planning scheme, (c) convergence curve of best fitness values.

As can be seen in Figure 4.9(b), when the vehicle detects the presence of obstacles intermittently, the initially planned and subsequently planed paths for the reactive path plan approach differ greatly. This discrepancy implies that the reactive path plan approach based on shell space decomposition scheme randomly initialize the population in each path refresh cycle and optimize the solution from there. The path re-plan solution is much closer to the original planned trajectory, and again, the path re-plan approach deviates less from the previous path refresh during subsequent path refresh (see in Figure 4.9(a)). One might say that the path re-plan is more aggressive, as it does not shy away from keeping close contact with the obstacles, as the planner knows it will be taking future measurements that help the AUV avoid collision. Like the previous example, the shell space decomposition scheme allows the path re-plan approach keeps track of favourable features from previous plans, and reply on the updated measurements (currents and obstacles) to obtain a reasonable refresh path. In fact, the execution time of the obtained trajectory with the path re-plan approach

(PRP-R) is 325s, compared to 376s for the reactive path plan approach.

4.4.6 Case 4-5: Path Planning in Spatiotemporal, Cluttered, and Uncertain Oceans

In Case 4-5, the problem complexity is increased, by incorporating a cluttered ocean environment with irregularly shaped terrains, a dynamic current field, similar to Case 4-1, dynamic obstacles with growing uncertainty in positions, similar to Case 4-2, and a moving rendezvous, similar to Case 4-3, all at the same time. The results for the path planner trials are presented in Figure 4.10. The realistic-looking terrain is generated using fractional Brownian motion and fractal generation methods [152]. Trajectories generated by both path planning schemes succeed in avoiding the static irregular terrains as well as the clustered dynamic obstacles and maintain the AUV heading towards the moving rendezvous.

(a)

(b)

(c)
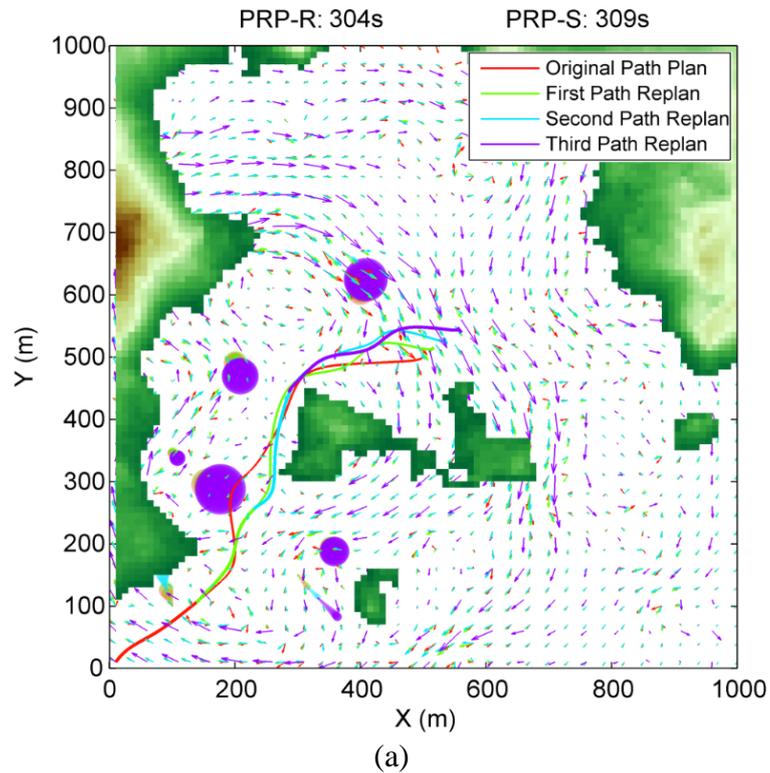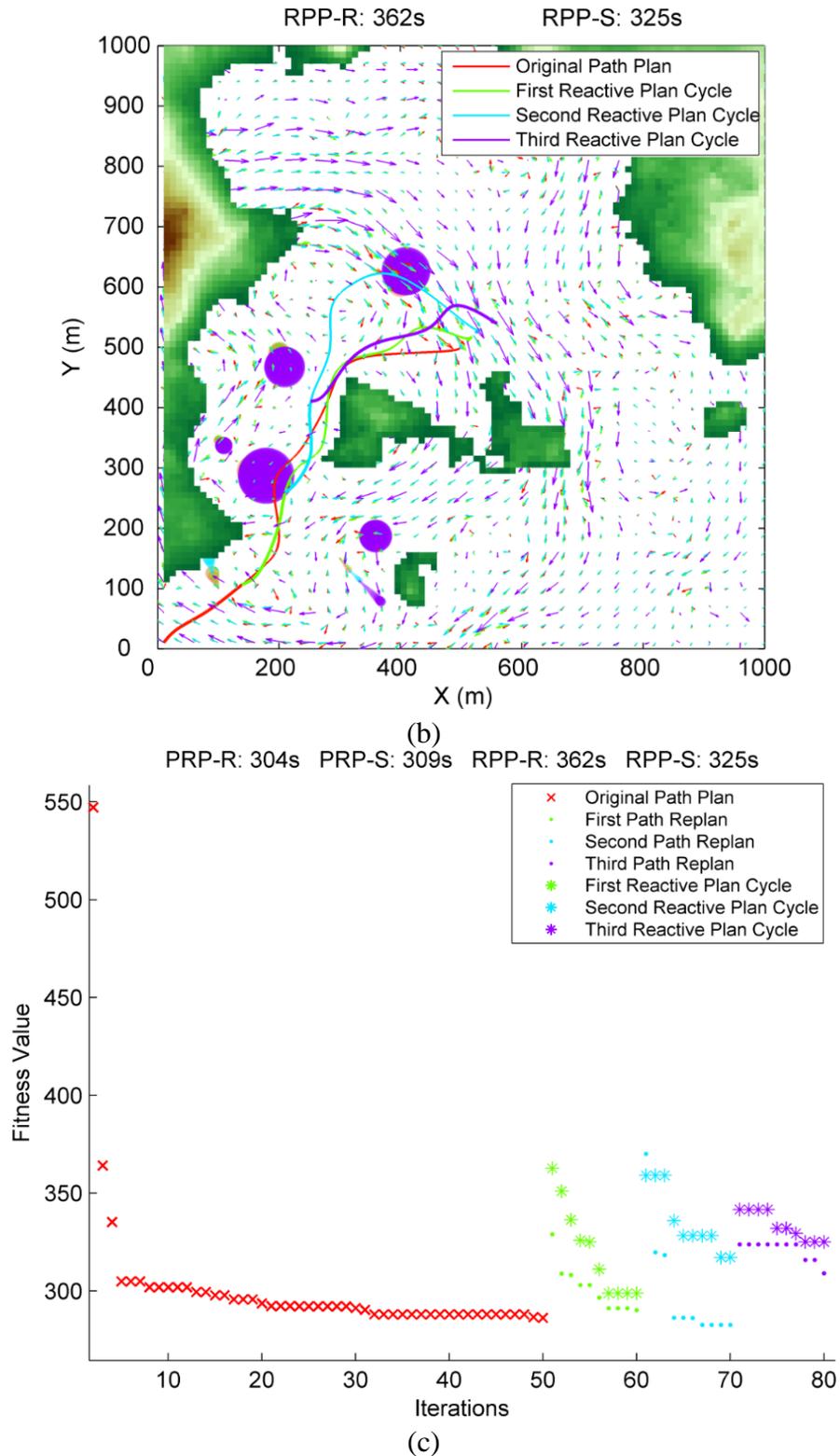
Figure 4.10 Comparison of results produced by the path planner based on reactive path planning and path re-planning schemes in a dynamic, cluttered and uncertain environment: (a) paths planned on-line with path re-planning scheme, (b) paths planned on-line with reactive path planning scheme, (c) convergence curve of best fitness values.

By comparing the convergence curves, shown in Figure 4.10(c), of the two path planners, it can be seen that the trajectory provided by the path re-planning scheme is superior in terms of fitness compared with that of the reactive path planning scheme. The trajectories and corresponding execution times for the two planners are shown in Figure 4.10(a) and (b), and reveal that the trajectory provided by the path re-planning scheme consumes significantly less travel time (58s), around 17% less, than that consumed for the trajectory provided by reactive path planning scheme.

4.5 Robustness Assessment

In this section, a set of representative Monte Carlo simulations are performed on the planners to determine under what circumstances can the reuse of previous planning information in the path re-planner help to improve the future trajectory with respect to the reactive path planner's generated trajectory. This set of simulations is designed to reveal the performance of the planners in a realistic scenario with variable current condition. The first subset of simulations performed involves varying the number of obstacles. The second subset considers the velocity of the moving rendezvous. In the third set of simulations, the rate of change for the current field is varied across the simulation set. This is realized by applying a variable quantity of Gaussian noise to the related parameters in Equations (2-4)~(2-7) as described in Subsection 2.3.1. In the fourth set of simulations the update rate of the trajectory regeneration is varied across the simulation set. Higher on-line trajectory update rates should provide more responsive paths based on more recent and reliable information, however, it also allows less time for the planner to compute its new path, implying that more powerful computing resources may be needed to prevent overruns. This analysis will help to determine what improvement the update rate has on the planner's performance.

In the following plots as shown in Figure 4.11–Figure 4.14, two sets of fitness values are provided for each planner; Simulated and Executed. Similar to the previous four case studies, the executed trajectory is more realistic because the rate of change of the environment is more continuously modelled, i.e., it has a higher update rate that more closely emulates the slow but continuous changes of the real ocean

environment, rather than assuming discrete time steps as in the case of the simulated.

4.5.1 Increasing Numbers of Obstacles

In the first study, simulations are performed on a 100-runs basis, with the same current environment, the same starting point and the same destination point, and the same number of obstacles which are randomly located within the map for each run. The number of obstacles varies from 1 to 10 and is changed in each set of 100-runs. As shown in Figure 4.11, the best fitness values for both simulated and executed trajectories generated with the reactive path planning scheme, are worse than those generated with the path re-planning scheme. It is further observed from Figure 4.11 that as more obstacles are added, the boxplots of the best fitness values for paths generated by the reactive planner display a relatively constant standard deviation. This consistency is a result of the reactive path planning randomly initializing its population at every update which allows more diversity of paths to search around obstacles. In contrast the path re-planning scheme reveals an expanding standard deviation for the best fitness values as more obstacles are added. That is to say, as the number of obstacles increases, the problem becomes more complex and more difficult for the path re-planning scheme to find the global optimal solution for the problem. However, it should be noted that even though the performance of the path re-planning scheme become less stable with increasing numbers of obstacles, the best fitness values achieved with this scheme are still superior to those obtained with the reactive path planning scheme.
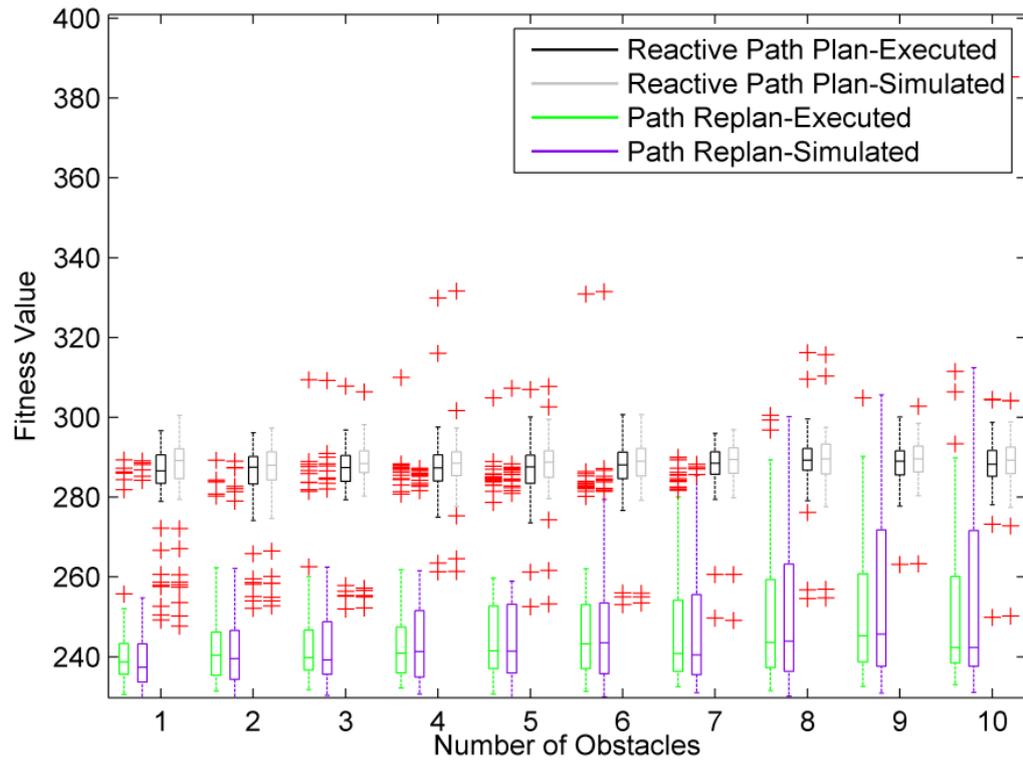
Figure 4.11 Comparison of results produced by the reactive path planner and the path re-planning schemes for 100-run Monte Carlo simulations each with 1 - 10 obstacles.

4.5.2 Moving Rendezvous Target

In this second study the simulations are performed on a 100-runs basis, where the ocean environment and initial position of the vehicle are maintained constant, but with a different rendezvous target velocity for each set of 100 runs. The standard deviation of the Gaussian noise added to the x and y positions of the rendezvous target are selected to be 0.005 to 0.05, respectively, with a step size of 0.005.

It should be noted that in Figure 4.12 the boxplots of the best fitness values of paths generated by both schemes show expanding standard deviation. This is reasonable since, as the rendezvous velocity increases, the target will get further away from its original location and consequently the fitness values become more and more varied. It should also be noted that, for all ten scenarios, including both simulated and executed trajectories, the best fitness values of trajectories generated with the reactive path planning scheme are all inferior to those generated with the path re-planning scheme, which further demonstrates the improved performance of the proposed path re-planning scheme.
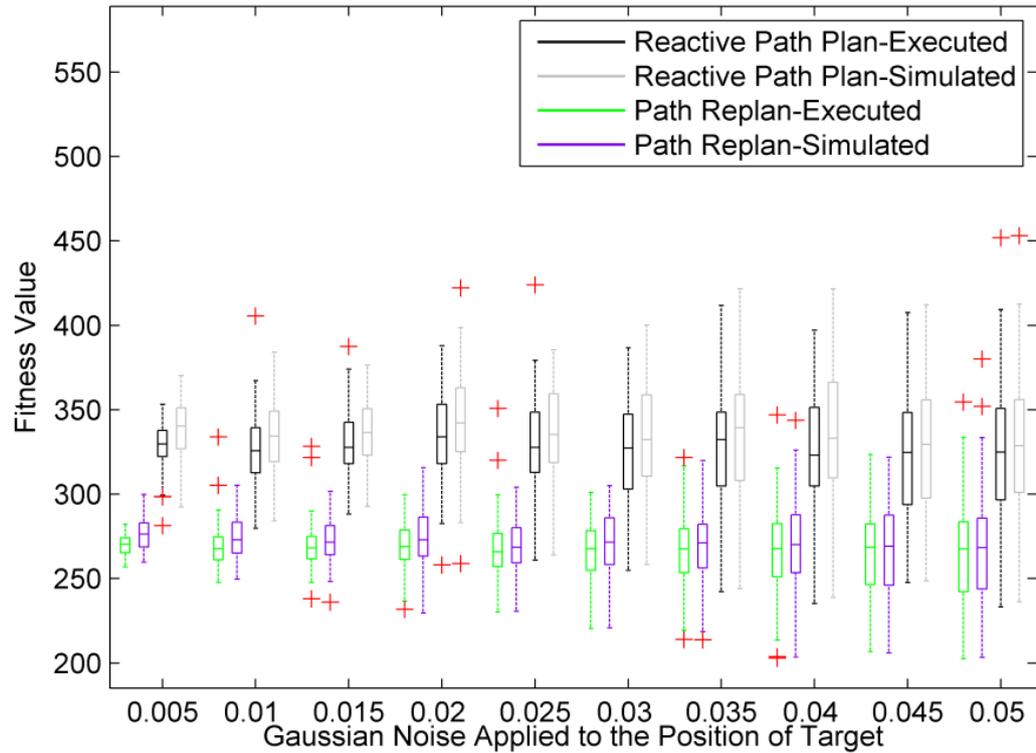
Figure 4.12 Comparison of results produced by the reactive path planner and the path re-planner for 100-run Monte Carlo simulations of a moving rendezvous target.

4.5.3 Effects of Changing Rate of Currents

In this third study, ten scenarios are evaluated, where the current field is set to change at different rates. To simulate the effect of the current field changing at different rates with time, a variable quantity of Gaussian noise, (0.1 to 1 with step size of 0.1), is applied to the three parameters $\mathbb{R}^o, \eta$ and ζ in Equations (2-4)~(2-6) given in Subsection 2.3.1. For each of the ten scenarios, the simulations are performed on a 100-runs basis. As can be seen from Figure 4.13, when the rate of change is relatively low (Gaussian noise parameter value between 0.1 and 0.5), the best fitness values of the simulated and the executed paths produced for both planners are similar. When the currents change at a higher rate (Gaussian noise parameter value greater than 0.6), the best fitness values of the executed paths becomes significantly higher than those of the simulated paths. Theoretically, other conditions remaining constant, the lower the rate of change of the currents, the closer the executed path is to the simulated path. Hence, for a dynamic path planning system, if the path update rate is significantly less than the rate of change of the current field then the simulated trajectory may fail to approximate the executed

trajectory.

By comparing the best fitness values of the executed paths produced by both planners, it can be seen that the difference between them decreases as the rate of change of the current field increases. When the Gaussian noise parameter value is 1, the fitness values of the executed paths produced by both planners is nearly the same which indicates that when the current field is changing so rapidly, the benefits of the proposed path re-planning scheme over the reactive path planning scheme are no longer significant.
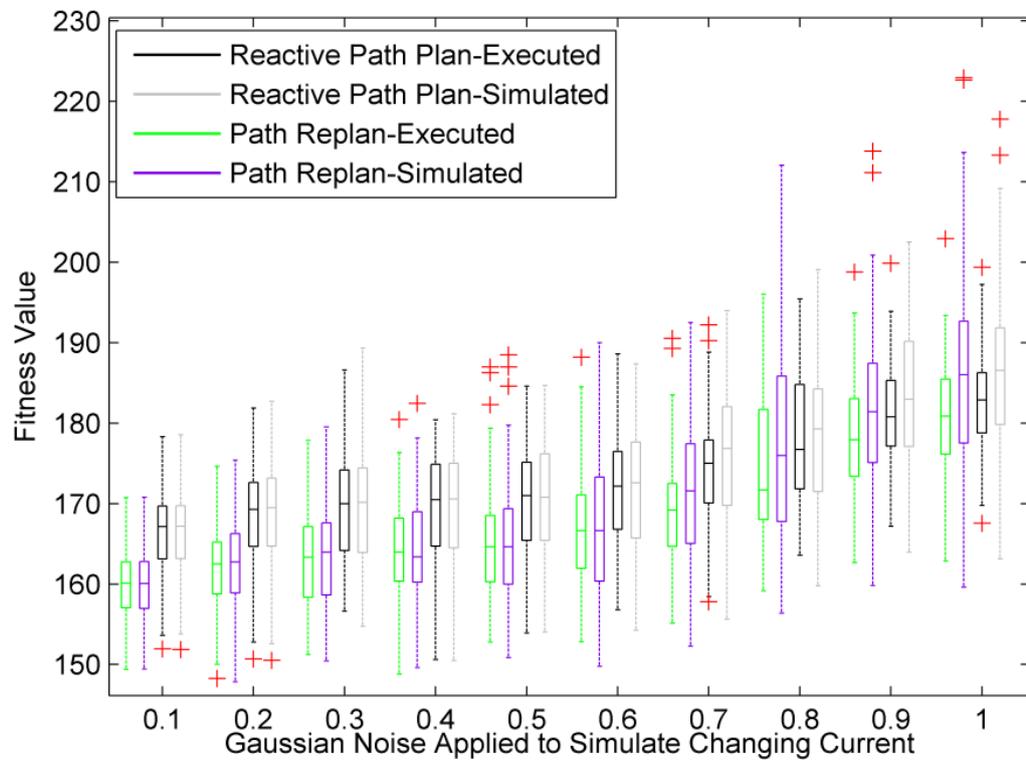
Figure 4.13 Comparison of results produced by the reactive path planner and the path re-planner for 100-run Monte Carlo simulations with currents changing at various rates.

4.5.4   Effects of Time Interval for Dynamic Path Planning

The study performed here is to examine the effect of changing the trajectory refresh rate on the quality of the generated trajectory as measured by its best fitness value. The simulation is conducted on a 1000-run basis, initially starting with an refresh period of 128 seconds and then repeating it a further three times, halving the refresh period at each increment. From Figure 4.14 it can be seen that as the refresh rate increases, the mean best fitness values of the executed paths produced with the

proposed path re-planning scheme decreases steadily while those for the reactive path planner fluctuate irregularly. Therefore, there is a potential to achieve better solutions for the dynamic planning system based on the path re-planning scheme by re-planning frequently. It should be noted, however, that a higher refresh rate implies less time available for the on-board AUV computer to optimize the path in each cycle, which may lead to very suboptimal result. The solution to this is to use higher performance central processing unit (CPU) based computing resources within the AUV, but this will increase battery consumption which is not desirable for persistent AUV missions.
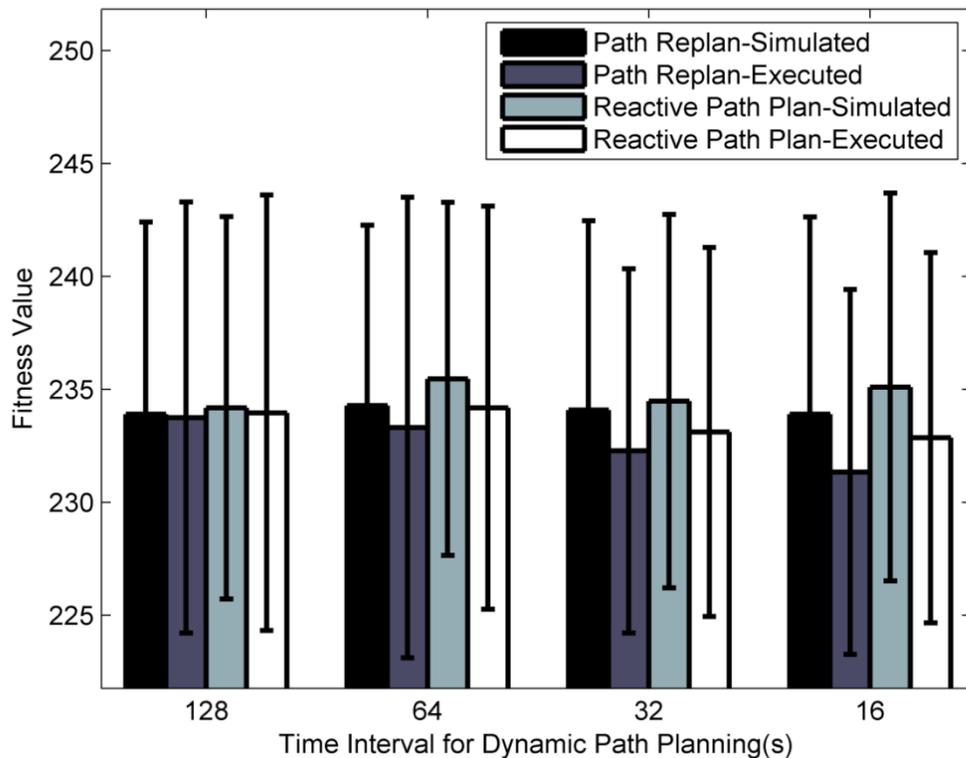
Figure 4.14 Comparison of fitness values produced by the reactive path planner and the path re-planner for 1000-run Monte Carlo simulations with four refresh rates for the dynamic planning system.

4.6 Chapter Summary

This chapter extended the previous SSD scheme to include continuous reshaping and regeneration of the optimal trajectories during the course of an AUV mission through a spatiotemporal current field. We developed an on-line planning system that integrated the dynamic SSD scheme, together with an efficient path re-planning scheme, into the B-spline based QPSO technique. Simulation tests have been

performed to assess the performance of the planner for finding an optimal trajectory with minimum time consumption for an AUV travelling through turbulent ocean fields to rendezvous with a moving target, in the presence of terrain obstacles and dynamic suspended obstacles with uncertainty in position. The on-line planning system with re-planner scheme is evaluated against that with reactive planning scheme which generates a completely new trajectory at every refresh iteration. From the results of these tests, the proposed on-line planning system with re-planning scheme is shown to be capable of finding a more optimized trajectory with same amount of computation load. In addition, the representative Monte Carlo simulations varying closely related parameters, involving the number of obstacles, the velocity of the moving rendezvous, the rate of change for the current field and the rate of the trajectory regeneration (as analysed in chapter 4.5), demonstrate inherent superiority of the proposed scheme compared with the reactive planning method under various scenarios.

In the next chapter, as an extension of the work on path planning of a single vehicle, a distributed SSD scheme will be developed and incorporated into an effective path planning guidance system for multiple marine vehicles operating in ocean environments. Several techniques will then be investigated for cooperative multiple marine vehicle path planning that explicitly address simultaneous rendezvous.

Chapter 5

Distributed Shell Space Decomposition (DSSD) for Rendezvous Path Planning of Multiple Autonomous Marine Vehicles (AMVs)

In this Chapter, a distributed shell space decomposition (DSSD) scheme is proposed for rendezvous trajectory planning of multiple AMVs, this category of vehicle includes both AUVs and autonomous surface vessels (ASVs). The DSSD builds on the concept of SSD developed in Chapter 3. This scheme is combined with an optimized mass-centre rendezvous point selection scheme, together with the B-spline based QPSO technique to find rendezvous trajectories for multiple AMVs with minimal time usage over all participating vehicles and simultaneous time of arrival for all the participating vehicles. The path planner identifies the optimal rendezvous location and generates the corresponding rendezvous trajectories based on the capabilities of each vehicle and the dynamics of the ocean environment. Several examples of rendezvous involving multiple AMVs resolved with the proposed method are presented.

5.1 Introduction

Multiple cooperative AUV systems hold great promise for undertaking such as large scale oceanographic surveys [47], mine counter-measures, adaptive sampling of large marine habitats [153], bathymetry surveys, monitoring the environment for dynamic events and other similar underwater tasks [154, 155]. Simultaneous use of multiple vehicles can improve performance, reduce mission time, provide broader or more robust data, and increase the likelihood of the mission's success [156, 157]. Multiple AUVs were used for an adaptive ocean sampling project [158]. In this project, a group of AUVs were spread across the Monterey Bay, California. The data collected was used to analyse and predict ocean processes. As the data needed to be collected from various locations simultaneously, a single AUV would obviously not have been suitable. It is not necessary for all the vehicles in an operation to be the same, and in fact, heterogeneity where vehicles carry different sensors could become a powerful driver of multiple AUVs operations. Different missions could be accomplished by different combinations of the vehicles [48]. Each submersible type has different strengths and weaknesses, and using complementary vehicles with different capabilities can lower costs while providing great operational flexibility [159]. Cooperative use of ASVs and AUVs for persistent seafloor exploration and monitoring is proposed in [160]. It envisages a next generation ASVs that could conduct deploy, recharge and recover AUVs automatically. Such ASVs would continue to assist with the navigation of the AUV fleet (ASVs act as a mobile two-element long baseline (LBL) array to navigate fully submerged AUVs for extended periods of time), as well as exchanging information with the central shore station.

The literature on the cooperative operation of AMVs is vast and a great deal of attention has focussed specifically on problems such as cooperative motion control of fleets of autonomous vehicles [47], and cooperative vehicle localization and navigation [48, 49]. Path planning is critical to ensure the vehicles' safe and efficient operation [157]. However, earlier work on this topic has been dominated by single-AUV planning and determining feasible strategies off-line that minimize a chosen energy or time-related cost criterion [30]. Considerable work has still to be done to develop advanced methods for cooperative multiple AMVs path planning that

explicitly address the problem of simultaneous rendezvous.

The purpose of the rendezvous may be the initial step in formation creation, for the exchange of data, or to recharge, maintain or collect vehicles [86, 161]. One example: Figure 5.1 depicts a conceptually simple scenario where, at the completion of the mission, the AUV surfaces and rendezvous with the ASV for recovery. AUV nowadays are commonly equipped with a GPS and Iridium Satellite communication, after surfacing, the AUV transmits data regarding its profile, position and status via satellite to the ASV. The path planner on the ASV then determines the trajectories on-line based on this information received from the AUV and the latest data on ocean currents. These trajectories should allow the AUV and ASV to arrive at their rendezvous point at approximately the same time and at optimum cost. It is only after the AUV receives the trajectory profile from the ASV that the AUV can dive down and accomplishes the rendezvous. Another application example is in accordance with the server vehicle concept proposed by Marco and Healey [162] for retrieving data from multiple AUVs involved in surveys, surveillance, or similar mission. Given the serve vehicle is able to rendezvous with the work vehicles and operate in close proximity to them, higher acoustic bandwidth data transfer is possible. A current FP2-TRIDENT project [163] also looking at reactive cooperation between AUVs for autonomous underwater intervention missions.
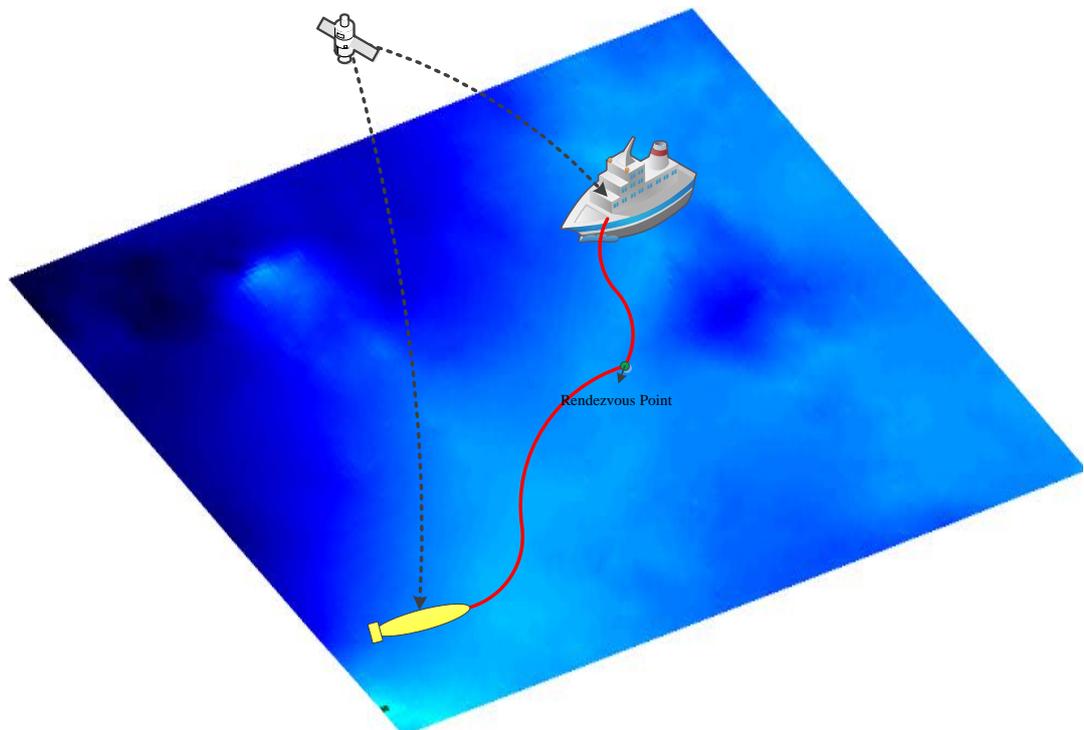
Figure 5.1 Example of rendezvous path planning of an AUV with an ASV

A number of challenges exist however, for these applications to work well in practice. The more time an AUV spends on the surface, the more likely it is to be involved in collisions with ship and boat traffic; as well, strong winds and heavy waves increase the risk of the AUV being damaged. Therefore, it is important to keep the AUV's time on the surface to a minimum. Consequently, it is crucial that the path planning scheme is inexpensive and efficient, and thus satisfy the demand of online applications. It is also required that the path planner is capable of incorporating the many different factors influencing a given mission, e.g. environmental conditions and the vehicle's dynamic constraints [164]. In addition, the path planner needs to take into account requirements imposed by multiple vehicle scenarios (of which collision avoidance is an important issue), and inter-vehicle communication constraints [165, 166]. Mission-related measures have to be incorporated additionally, such as minimization of time/energy usage over all participating vehicles [161, 167].

Central to the path planning task for simultaneous rendezvous of multiple autonomous vehicles is rendezvous point selection. Rendezvous point selection is an important factor for efficient optimal path searching, and a good rendezvous point selection scheme must take explicitly into account both the capabilities of each

vehicle and the dynamics of the ocean environment. A review of the existing rendezvous point selection scheme are presented in Section 5.2. This chapter looks at a novel hybrid scheme that combines deterministic and heuristic technique. The deterministic method (centroid or mass-centre) is applied first so as to obtain a reasonable first-attempt approximate (guess) solution. The heuristic scheme is then applied to obtain a refined optimal solution; generally, the result is close to the guess. Searching efforts using this hybrid scheme are more effectively concentrated at meaningful region, and is often capable of converging to the desired result even in the presence of a poor initial guess.

In this study, a DSSD scheme is developed and integrated into the QPSO-based path planner presented in Chapter 2. This path planner is combined with a novel optimized mass-centre rendezvous point selection scheme to find trajectories for multiple AUVs that ensures they reach their destination in the shortest possible time and arrive simultaneously. Work on this line of research relating to the general shell space decomposition method for single-AUV path optimization has already been presented in Chapter 3. As an extension, this chapter proposes a detailed description of the DSSD scheme for multiple AMVs path planning with optimized rendezvous point and the scheme's performance in both 2D and 3D.

In addition, for path planning with minimum time consumption, the state of the art methods set the vehicle velocity to maximum and determine the optimal path by taking advantage of favourable currents. This scheme is reasonable for single AUV path planning, however, for simultaneous multiple vehicle rendezvous, the time taken is decided by the vehicle that is last to arrive. The scheme could avoid the problem of some vehicles operating at maximum velocity arriving early and then having to spend more energy while they wait for the other vehicles to arrive. In this thesis, in addition to the novel hybrid rendezvous point selection scheme, the multi-vehicle rendezvous path planner will also involve one extra optimization module. This module is designed so that the operational speed for every vehicle can be flexible in a certain range, and the choice of an appropriate operational speed will be optimized based on the objective function.

The rest of this chapter is organized as follows. Section 5.2 describes the path planning missions for multiple AMVs rendezvous and provides an overview of the

multiple AMVs guidance system. It also formulates the optimization problem. Section 5.4 introduces the particle structure for the QPSO as well as the modules for detection of the optimal speeds and rendezvous point. Section 5.5 presents the distributed shell space decomposition method in both 2D and 3D. The simulation tests and results of extensive simulations completing realistic maritime missions in regional ocean surrounds Australia/New Zealand are presented in Section 5.6, followed by a set of robustness assessments using Monte Carlo trials presented in Section 0, concluding remarks are then presented in Section 0.

5.2 Related Work

In general, optimal rendezvous path planning is arrived at by one of two methods: deterministic or heuristic. With the deterministic methods, the best attempt rendezvous point is identified and the path is optimized from there based on the objective function. The Fermat point method is widely researched as one of the deterministic method for rendezvous point selection as it minimizes the sum of distances between itself and the positions of all the robots [168]. Unfortunately, it has been proven that the Fermat point method is unable to cope with large number of robots and thus cannot be used to solve the rendezvous problem [169]. Another method for determining a suitable rendezvous point is to find the centroid of all the robots [170]. The centroid, defined similarly to the geometric median, can be determined by using a simple formula — the centroid's coordinates can be determined by calculating the averages of the coordinates of the points. Despite its simplicity, the centroid method is not useful for determining the rendezvous point as the centroid is only the geometric centre of a space and unable to calculate multiple vehicles moving towards it at various speeds. To overcome this, the centre of mass could be used as the rendezvous point. The centre of mass is the weighted vector sum of the vehicles' positions with the velocity of each vehicle being used as the mass of the corresponding point [161]. Compared with the centroid method, it is a better choice to use as the rendezvous point, as the distances from this point to each vehicle position is proportional to the velocity of that vehicle. This scheme is still flawed, however, because it does not take environmental conditions into account. There are many

challenges when dealing with the ocean such as variable ocean currents and dynamic obstacles which may affect the velocity of vehicles and the shape of path, and further affect the time for the vehicle to arrive at the rendezvous point.

These impediments have propelled the development of effective heuristic methods to solve the problem of optimal path planning for rendezvous operations. Heuristic methods normally adopt evolutionary or artificial intelligence algorithms, to try to improve a candidate solution (here referred as the rendezvous point and the corresponding paths) iteratively based on the stochastic optimization technique [54]. The unconstrained full-scale searching mechanism is a raw version of this method that allows the rendezvous point to be located anywhere. This scheme is preliminary however, because the evolutionary algorithms' attempt to search the entire space is inefficient.

5.3 Problem Formulation

The marine environment poses a rich field of challenges to multiple vehicle path planning systems to be dealt with, such as ocean currents, irregularly shaped terrains and dynamic and uncertain obstacles. Central to the implementation of a multiple vehicle path planning systems is the availability of an algorithm, which can take into account the constraints of each vehicle as well as important environmental conditions [166].

Figure 5.2 illustrates this problem and shows how the initial and final conditions of the vehicles, as well as various internal and external constraints, produce, if possible, a trajectory that meets the constraints at minimal cost. The spatial and temporal coordinates of this trajectory yield both a spatial path and a corresponding velocity profile.
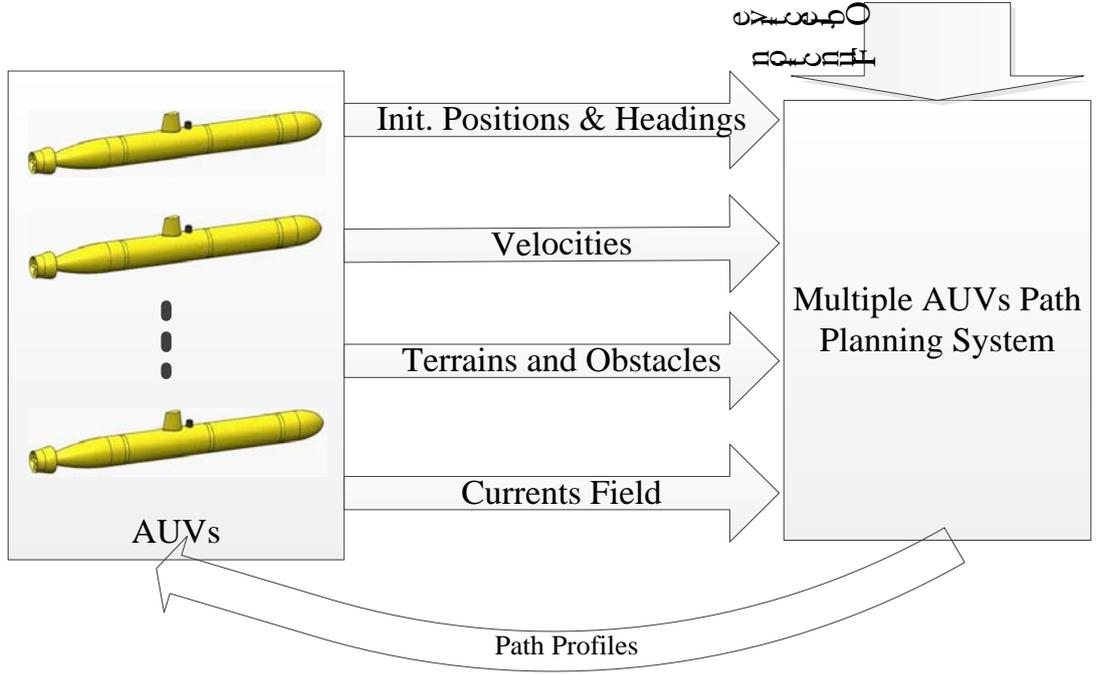
Figure 5.2 Multiple AUVs Path Planning System

Consider n AUVs $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n\}$, and assume these vehicles have constant but various water-referenced speed $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_n\}$ with their initial condition $\mathcal{A}_i = (r_i, w_i, t_o)$, where $r_i = [x_i, y_i, z_i]$ defines each vehicle's position in a spherical coordinate frame, and $w_i = [\phi_i, \theta_i, \psi_i]$ defines each vehicle's Euler angles roll, pitch and yaw. The objective of a Multiple AUVs path planning system is to find the optimal paths $\mathbb{p}^\dagger = \{\mathbb{p}_1^\dagger, \mathbb{p}_2^\dagger, \ldots, \mathbb{p}_n^\dagger\}$ among the set of all feasible paths $\mathcal{p}$ for all AUVs to travel through the ocean and arrive at the rendezvous point $\mathcal{G} = (x_\mathcal{G}, y_\mathcal{G}, z_\mathcal{G})$ simultaneously with minimization of time usage over all vehicles. The ocean environment V_c is modelled as a strong current field occupied with obstacles O and terrain $\mathbb{T}$, the position of which may be dynamic and uncertain. Therefore the path planning problem is formulated as the following optimization problem:

$$[\mathcal{G}, \mathbb{p}^\dagger] = \frac{argmin}{\mathbb{p} \in \mathcal{p}} F(\mathcal{A}, V_c, O, \mathbb{T}, \mathcal{V}),$$

$$\text{s.t.} \quad M(\eta, \dot{\eta}) = 0, \tag{5-1}$$

$$\forall m, n \in \{1, \ldots, n\}, j \in \{1, \ldots, |\mathbb{p}|\} \quad \wp_j^m(t_j) \cap \wp_j^n(t_j) = \emptyset \ m \neq n,$$

$$\forall i \in \{1, \ldots, n\}, j \in \{0, \ldots, |\mathbb{p}|\} \quad \wp_j^i(t_j) \notin O(t_j) \cup \mathbb{T},$$

where $\wp_j^i = (x_j^i, y_j^i, z_j^i)$ is a waypoint along the path $\mathbb{p}_i^\dagger = \{\wp_1^i, \wp_2^i, \wp_3^i, \ldots\}$ for the ith AUV, and $M(\eta, \dot{\eta})$ is the vehicle's kinematic model.

5.3.1 Optimization Criterion

Two factors are used to determine the cost criterion. One is the time used by all participating AUVs represented by $\mathbb{M}$, and the other factor is $\mathbb{S}$ which represents the simultaneous arrival of the AUVs at their selected rendezvous.

$$F = \mathbb{M} + \wp \cdot \mathbb{S} \tag{5-2}$$

where $\wp$ is a pre-set ratio can stipulate the expected time zoom within which all participating AUVs should arrived at the rendezvous and can trade the performance of simultaneous arrival against time usage over all participating AUVs.

Factor 1: This factor $\mathbb{M}$ determines, for a given team of n vehicles initially at location r, constantly traveling at water-referenced speed $\mathcal{V}$ within a ocean currents field V_c with model of obstacles O and terrain $\mathbb{T}$, a set of path profiles for every participating vehicle to reach a rendezvous location $\mathcal{G}$ that achieves minimum time consumption.

$$\mathbb{M} = \sum_1^n T_i, \tag{5-3}$$

where T_i is the time consumption for the ith vehicle to arrive at the rendezvous location, this is covered in more detail in Subsection 2.3.2.

Factor 2: Given a team of n vehicles with various water-referenced speed $\mathcal{V}$ initially at location r, then $\mathbb{S}$ is the sum of the waiting time of every vehicle $\Delta T = (\Delta T_1, \Delta T_2, \ldots, \Delta T_n)$ to the moment when the last vehicle arrives at the rendezvous location $\mathcal{G}$.

$$\mathbb{S} = \sum_1^n \Delta T_i, \tag{5-4}$$

$$\text{with} \qquad \Delta T_i = \max(T_1, T_2, \ldots, T_n) - T_i$$

5.4 QPSO Particle Structure and Optimization

The Spline based QPSO path planner developed in Section 2.4 is modified and used

as the platform to evaluate the proposed DSSD mechanism. Every particle in the swarm represents a potential path, the parameters of each particle corresponds to the coordinates of control points generating the paths for all AUVs. The whole population is denoted by a matrix $\mathbb{Q} = [\mathbb{Q}_1, \mathbb{Q}_2, ..., \mathbb{Q}_\mathbb{p}]$, where $\mathbb{p}$ is the population size. Different from traditional path planning problem, the end point of all paths are engaged in the coding of particles which is the same and shared by all paths. Suppose n paths are needed for n participating vehicles and each path is composed of q control points, together with one extra end point and one extra module, then in order to record the spatial position (x, y, z) of each path, the number of dimensions of each particle should be $3((q + 1) \times n + 1)$.
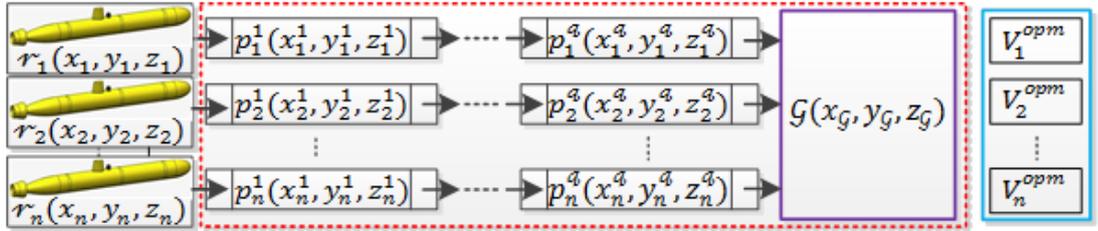
Figure 5.3 Particle structure.

5.4.1 Rendezvous Point Selection

a. Deterministic: Centroid or Weighted Mass-centre Scheme

5.4.1.1   Centroid Scheme

The centroid is the point defining the geometric centre of system. The centroid of a finite set of n vehicles located at $r_1, r_2, ..., r_\mathrm{n}$ is

$$\mathcal{G}^c = \frac{1}{n} \sum_1^n r_i \tag{5-5}$$

This point minimizes the sum of squared Euclidean distances between itself and each vehicle's position in the set.

5.4.1.2   Mass-centre Scheme

The mass centre of a set of vehicles is the centroid of the set of points at which the vehicles are located with the strength of each point being equivalent to the velocity of the corresponding vehicle. For a system of n vehicles, the weighted mass centre $\mathcal{G}^\mathrm{m} = (x_\mathcal{G}^\mathrm{m}, y_\mathcal{G}^\mathrm{m}, z_\mathcal{G}^\mathrm{m})$ can be written as

$$x_{\mathcal{G}}^m = \frac{\sum_1^n x_i \mathcal{V}_i}{\sum_1^n \mathcal{V}_i}, y_{\mathcal{G}}^m = \frac{\sum_1^n y_i \mathcal{V}_i}{\sum_1^n \mathcal{V}_i}, z_{\mathcal{G}}^m = \frac{\sum_1^n z_i \mathcal{V}_i}{\sum_1^n \mathcal{V}_i}. \tag{5-6}$$

b. Heuristic: Optimized Full-scale Scheme

The full-scale searching mechanism is the original heuristic method that allows the rendezvous point to be freely located anywhere in the environment. This full-scale searching region is represented as $[\mathcal{R}_L, \mathcal{R}_U]$, where $\mathcal{R}_L$ is the lower bounds and $\mathcal{R}_U$ is the upper bounds. The rendezvous point in this scheme can be computed as:

$$\mathcal{G}^f = \mathcal{R}_L + rand \cdot (\mathcal{R}_U - \mathcal{R}_L),$$

$$\text{with} \quad \mathcal{R}_L = \min(r_1, r_2, \ldots, r_n), \tag{5-7}$$

$$\mathcal{R}_U = \max(r_1, r_2, \ldots, r_n),$$

c. Hybrid Deterministic & Heuristic: Optimized Mass-centre Scheme

The hybrid deterministic and heuristic technique applies the deterministic mass-centre scheme initially to get a reasonable first-attempt approximate solution $\mathcal{G}^m$. A heuristic scheme is then applied to generate a refined optimal solution, which is usually located in the proximity of the guess. This search region is an enclosed circle centred at $\mathcal{G}^m$ with radius $\mathfrak{R}$ such that all rendezvous points are inside or on the circle. By specifying the radius $\mathfrak{R}$, it can stipulate the expected zone where the optimal rendezvous point will be located, and can trade off the scale of this searching zone against convergence speed.

$$\mathcal{G}^{\mathcal{M}} = \mathcal{G}^m + \mathbb{R} \cdot \mathbb{A},$$

$$\text{with} \quad \mathbb{R} = rand \cdot \mathfrak{R}, \theta^{\mathcal{M}} = rand \cdot 2\pi, \varphi^{\mathcal{M}} = rand \cdot \pi,$$

$$\mathbb{A} = \begin{bmatrix} \cos(\theta^{\mathcal{M}})\sin(\varphi^{\mathcal{M}}) & 0 & 0 \\ 0 & \sin(\theta^{\mathcal{M}})\sin(\varphi^{\mathcal{M}}) & 0 \\ 0 & 0 & \cos(\varphi^{\mathcal{M}}) \end{bmatrix}, \tag{5-8}$$

5.4.2 Detection of the Optimal Operational Speeds

- Rendezvous with deterministic full speed

The characteristics of this scheme mean that vehicles have to operate at (their) full speed. It is then up to the path planner to identify the optimal rendezvous point and simultaneous time of arrival. The upper plot of Figure 5.4 shows the simultaneous rendezvous of AUVs with deterministic speeds. Since the operational speed is determined for the optimization problems considered here, the time for optimal rendezvous only occurs at a particular point.
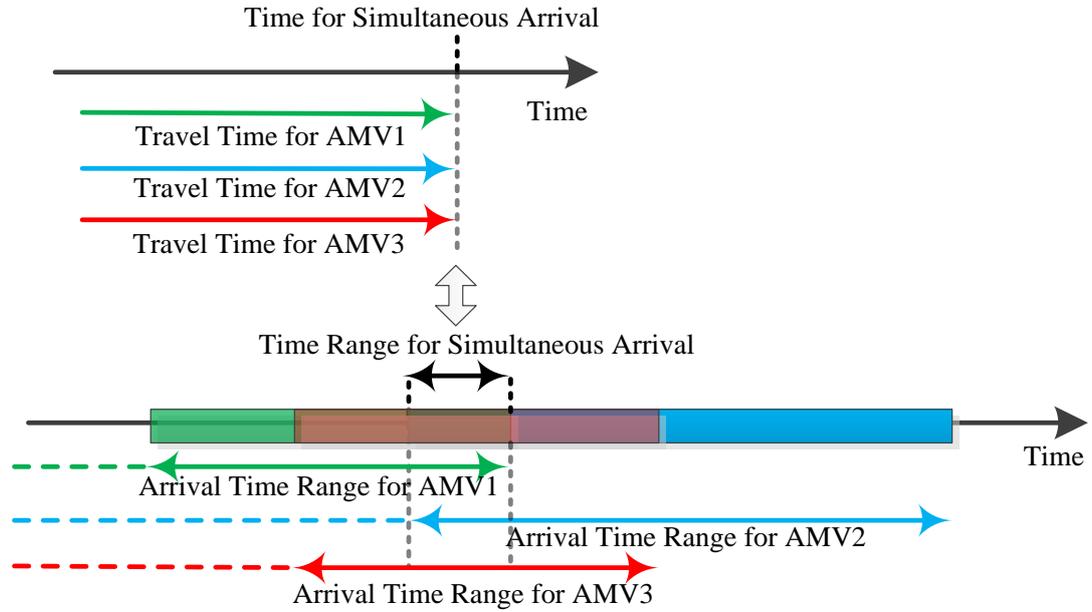
Figure 5.4 Comparison of simultaneous rendezvous of AMVs with deterministic speeds and flexible speeds.

- Rendezvous with optimized operational speed

This scheme is characterized by allowing vehicles' operational speed flexible to be chosen over certain range. It is the path planner's responsibility to identify the optimal rendezvous point as well as the exact operational speed for each vehicle. Assume that the participating AUVs has their operational speeds $V_a(v_{a1}, v_{a2}, \dots, v_{an})$ vary over the feasible range $[V_a^{min}(v_{a1}^{min}, v_{a2}^{min}, \dots, v_{an}^{min}), V_a^{max}(v_{a1}^{max}, v_{a2}^{max}, \dots, v_{an}^{max})]$, which determines a range of estimated arrival times. Figure 5.4 depicts the arrival time window for each vehicle with the horizontal bars. As can be seen, the overlapping interval of arrival times among those three vehicles indicates the time window where the optimal rendezvous will occur.

Path planners utilizing optimized operational speed can expect to benefit in three ways. First, and perhaps most importantly, constraints such as vehicles having to operate at pre-determined speeds limit the solutions for this problem, and consequently the path solution is usually suboptimal. Alternatively, the proposed approach facilitates greater flexibility for solving the task of path planning for

optimal rendezvous by implementing one extra optimization module that identifies the optimal operational speed for every vehicle. Rather than limiting the rendezvous to a particular time, this arrangement provides a wider timeframe for the optimal rendezvous to occur; as a result, the ability of path planners to arrive at an optimal solution is much enhanced. Second, vehicles travelling at full speed will arrive at the rendezvous point before the others, but QPSO-DSSD based path planners are able to determine the most favourable speed so that every vehicle arrives at the rendezvous point simultaneously or as close as possible. Third, flexible operational speeds, mean that path planners have the flexibility to generate trajectories and make detours that allow vehicles to make the most drift and propulsion from the currents.

As described in Subsection 5.3.1 above, the main criterion for the proposed path planner is the amount of time vehicles spend travelling and waiting. The optimized operational speed trajectories generated by path planners are expected to increase travelling time, compared to the deterministic full speed model. To better evaluate the generated solutions, power consumption is also measured.

The energy needed to propel the vehicle is related to the work to overcome the drag induced by relative current as the vehicle moves through water.

$$\varepsilon = \sum_{1}^{|\mathbb{p}|} \mathcal{V}^3 \times K_e \times t\left(\wp_i^{t_i}\right), \tag{5-9}$$

where K_e is the drag constant based on the vehicle design, for simplicity, here K_e is set to be 1 and t is the travel time as defined in Chapter 2, Equation (2-16).

The overall power consumption among all participating vehicles is therefore:

$$\mathbb{E} = \sum_{1}^{n} \varepsilon_i, \tag{5-10}$$

where ε_i is the power consumption for the ith vehicle to arrive at the rendezvous location as defined in Equation (5-9).

The sum of the power consumed by all vehicles wait to the moment that the last vehicle arrives at the rendezvous location is defined as.

$$\mathbb{e} = \sum_1^n \mathcal{V}^3 \times K \times \Delta T_i,$$
$$\text{with} \quad \Delta T_i = \max(T_1, T_2, \dots, T_n) - T_i \tag{5-11}$$

5.5 Distributed Shells Space Decomposition

The DSSD approach extends that of the SSD approach by generating multiple sets of shells radiating out from the starting position of each vehicle to the rendezvous destination. Each vehicle generates its trajectory within each subset of SSD, but these trajectories will be optimized as an entire fleet based on the global objective. Consider n AUVs $\mathcal{A}=\{\mathcal{A}_1, \mathcal{A}_2,\ldots, \mathcal{A}_n\}$, assume these vehicles have their initial positions at $\Re = \{r_1, r_2,\ldots, r_n\}$, $r_i = [x_i, y_i, z_i]$ defines the vehicle $\mathcal{A}_i$'s position in a spherical coordinate frame, $\mathcal{G} = (x_{\mathcal{G}}, y_{\mathcal{G}}, z_{\mathcal{G}})$ is the rendezvous point for all AUVs. For path planning of these multiple AUVs, the search space will be decomposed into multiple sets of shells, where each set of shells is formed according to each AUV's distance and angle position with respect to the rendezvous point $\mathcal{G}$ as described in Chapter 3.

5.5.1 2D Distributed Annular Shells Space Decomposition

Figure 5.5 illustrates the use of the DSSD for path planning of multiple AUVs, and for simplicity, proposes processing three sets of shells for three AUVs.
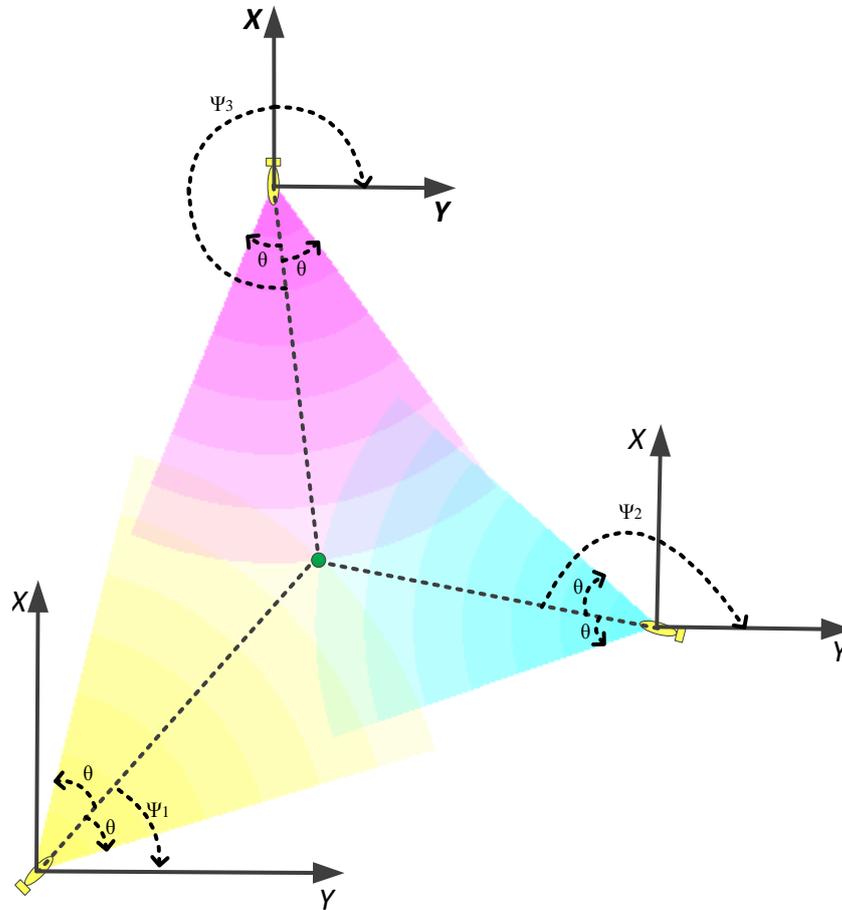
Figure 5.5 2D distributed annular shells space decomposition.

Angle θ can stipulate the expected space zone within which the optimal trajectory should be located, and can trade off the performance of path optimization against the available computational resources.

5.5.2 3D Distributed Spherical Shells Space Decomposition

The extension of DSSD scheme to 3D is similar to the mechanism developed in Chapter 3. Figure 5.6 illustrates the use of the DSSD for path planning of three AUVs. As can be seen, three sets of shells are constructed between the AUVs' initial location and target position.
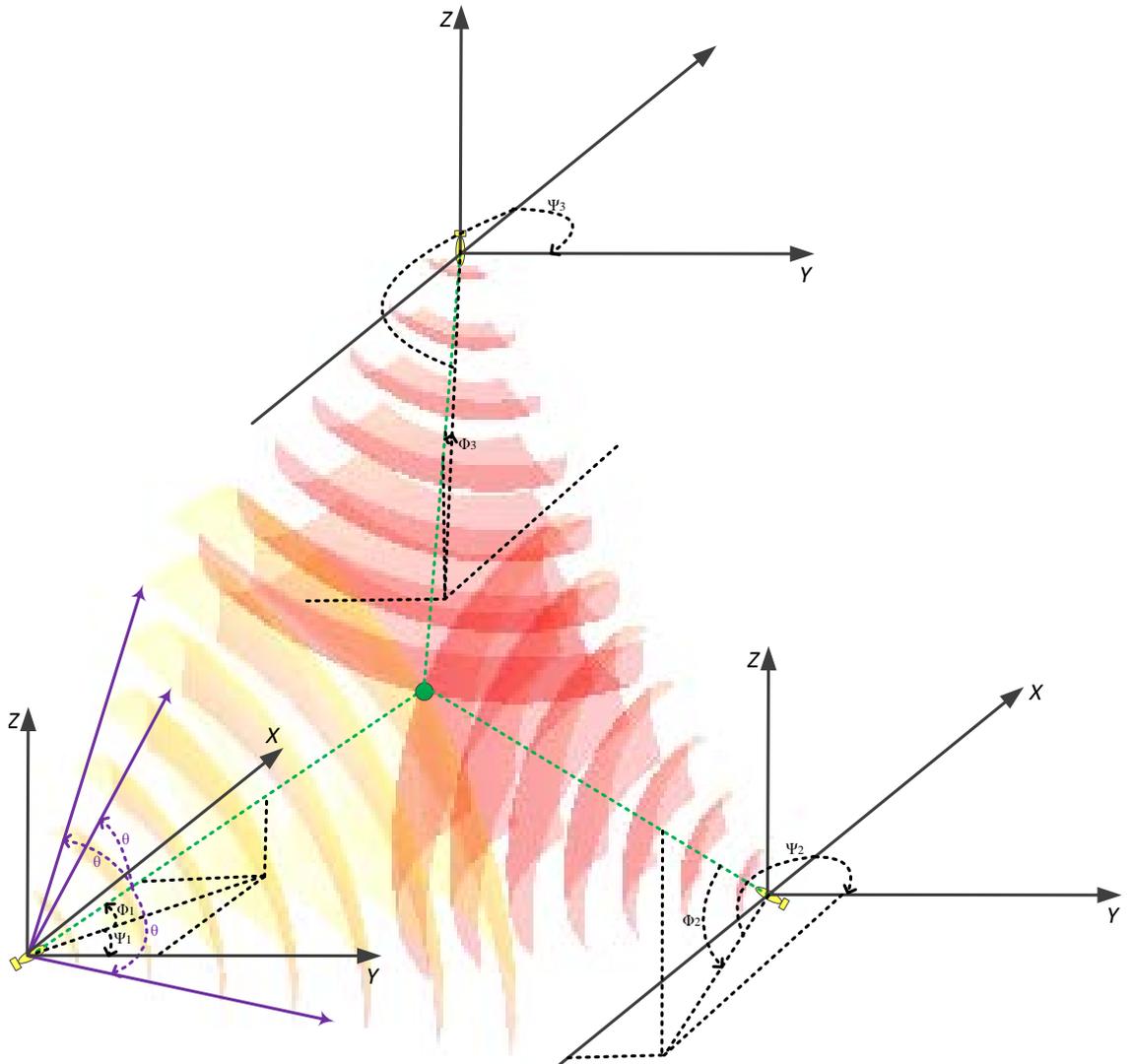
Figure 5.6 3D Distributed spherical shells space decomposition.

5.6 Simulations

The DSSD scheme was integrated into the Spline based QPSO path planner and implemented in Matlab. Three sets of simulation tests were designed to study the performance of the path planner for rendezvous of multiple AUVs in both 2D and 3D scenarios.

The first set, comprised of three case studies show the performance of the path planner based on the DSSD:

- Case 5.6.2-1: Path planning for an AUV to rendezvous with an ASV
- Case 5.6.2-2: Path planning for a server vehicle rendezvous with two worker

vehicles

- Case 5.6.2-3: Path planning for a rendezvous involving multiple AMVs in a complex ocean context.

In the second set of simulations, another three case studies were designed to compare the proposed four rendezvous point selection scheme in different scenarios. In the third set of simulations, path planner performance with both deterministic operational speed and optimized operational speed, respectively, is studied, with one additional case using extensive simulation with actual current data for multiple AUVs to complete realistic maritime missions in the oceans surrounding Australia and New Zealand.

5.6.1 Simulation Setup

As mentioned in Chapter 2.3, the current field for the scenario reported in this study, is generated from a random distribution of 50 Lamb vortexes represented by a 100×100 grid for the 2D scenario. The 3D ocean environment is approximated by a five-layered structure in which each layer is defined similarly to the 2D case but with a well-defined vertical profile [57]. The distance between nearest neighbour grid points corresponds to 1000 m for 2D space and 100m for 3D space. The strength ζ and radius η of each vortex is set to 5 m/s and 1 m, respectively. The water-referenced speed of the vehicle in the geographical frame is set at 1.5m/s. The contract-expansion coefficient β of QPSO is set to linearly decrease from 1.0 to 0.5 over the total iterations. Each individual B-spline path is formed by 6 control points, and is uniformly subdivided by 1,152 internal knots.

5.6.2 Case Studies with Distributed Shell Space Decomposition Scheme

Case 5.6.2-1: Path Planning for an AUV to Rendezvous with an ASV

This case study involves an AUV that surfaces at the completion of the mission and would like to rendezvous with the ASV for recovery. After surfacing, the AUV transmits its location via satellite to the ASV. The path planner system on the ASV then identifies the optimal rendezvous point and calculates the trajectories that allow the AUV and the ASV to arrive at the rendezvous point at about the same time and as quickly as possible.

In this case study, the AUV surfaces at (10, 5) and the ASV's position is at (75, 70) at that time, the AUV and ASV's water-referenced velocity is set to be 1.2m/s and 0.6m/s, respectively. The distributed annular spaces for path searching of these two vehicles are shown in yellow and pink in Figure 5.7. The red sphere represents the rendezvous point searching region for the optimized mass-centre scheme, this sphere is centred at the rendezvous point of the mass-centre scheme.
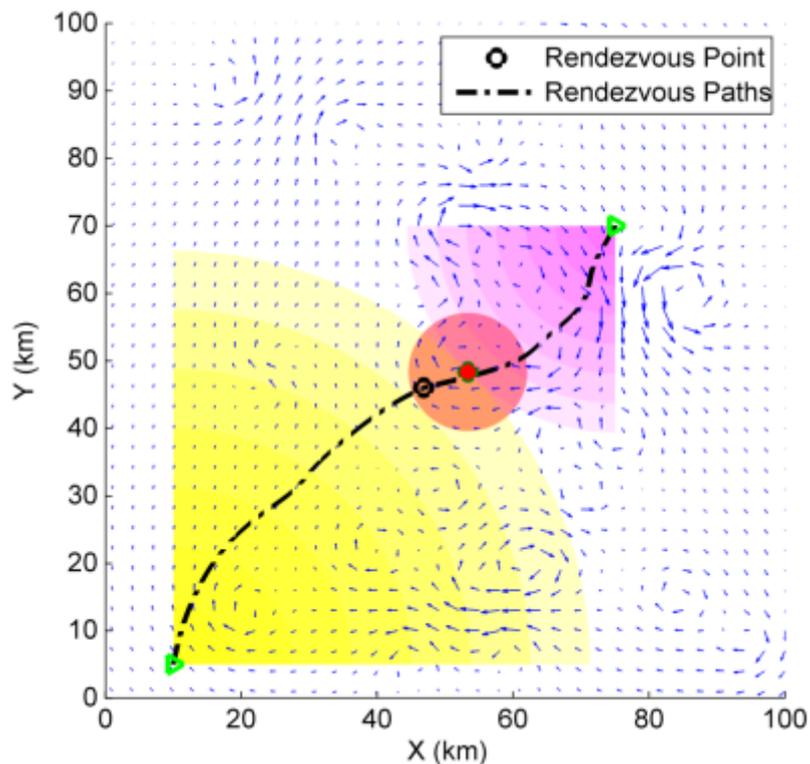
Figure 5.7 Trajectories produced by the path planner based on the DSSD scheme for an AUV to rendezvous with an ASV in 2D scenario.

For comparison, two reference tests are also added to each scenario. The first commands the vehicles to go in a straight line towards the rendezvous point [135] while the second directs the vehicles along their optimized trajectory but in an environment devoid of currents. The travel times of this test are listed in Table 5.1. It is evident that by following the optimized trajectories produced by QPSO-DSSD scheme, the AUV and ASV not only take less time to get to the rendezvous point, but also spend less time on waiting for the other vehicle to arrive at the rendezvous point. This can be explained by noting that by taking the straight-line trajectory, the vehicle may proceed through a region with the currents running contrary to the vehicle resulting in a reduced ground-relative velocity. In contrast, both optimized trajectories

produced by the QPSO-DSSD scheme followed the direction of the currents. This procedure allows the vehicles to ride the currents towards their destination and at increased velocity, which reduced their overall travelling time, even though the travel path was longer than the straight-line course. By comparing the time consumed for the corresponding tests along the optimized trajectories, with and without currents, it can be seen that the vehicles take significantly less travel time $\mathbb{M}$ in the presence of the currents, which further testifies that the optimized trajectories take advantage of favourable currents, and avoid adverse currents to minimize time expenditure. It should also be noted that the vehicles arrive at their rendezvous point approximately at the same time in the presence of the currents, which further demonstrates that the QPSO-DSSD scheme takes into account the effect of the currents to the performance of simultaneousness of rendezvous.

Table 5.1 Path planning for an AUV to rendezvous with an ASV

Scenarios	Time (sec)			
	AUV	ASV	Factor $\mathbb{M}$	Factor $\mathbb{S}$
Trajectories optimized with QPSO-DSSD with currents	34180	34094	68274	86
Straight line path with currents	36862	43691	80554	6829
Optimized trajectories without currents	37981	53024	91005	15042

Case 5.6.2-2: Path Planning for a Server Vehicle Rendezvous with Two Worker Vehicles

The second case study involves a server AUV which will rendezvous with two AUVs to retrieve data from them by means of high acoustic bandwidth data transfer. The serve AUV travels from the starting position (20, 80) at a water-reference velocity of 2m/s. Two data-gathering AUVs travel from their starting positions (80, 20) and (80, 80); their water-referenced velocity is set to be 1m/s and 1.5m/s, respectively.

The results of the trajectory optimization are shown in Figure 5.8. Similar results to those in the previous case study were obtained as shown in Table 5.2, and the vehicles performed well for rendezvous by following the optimized trajectories produced by the QPSO-DSSD scheme.
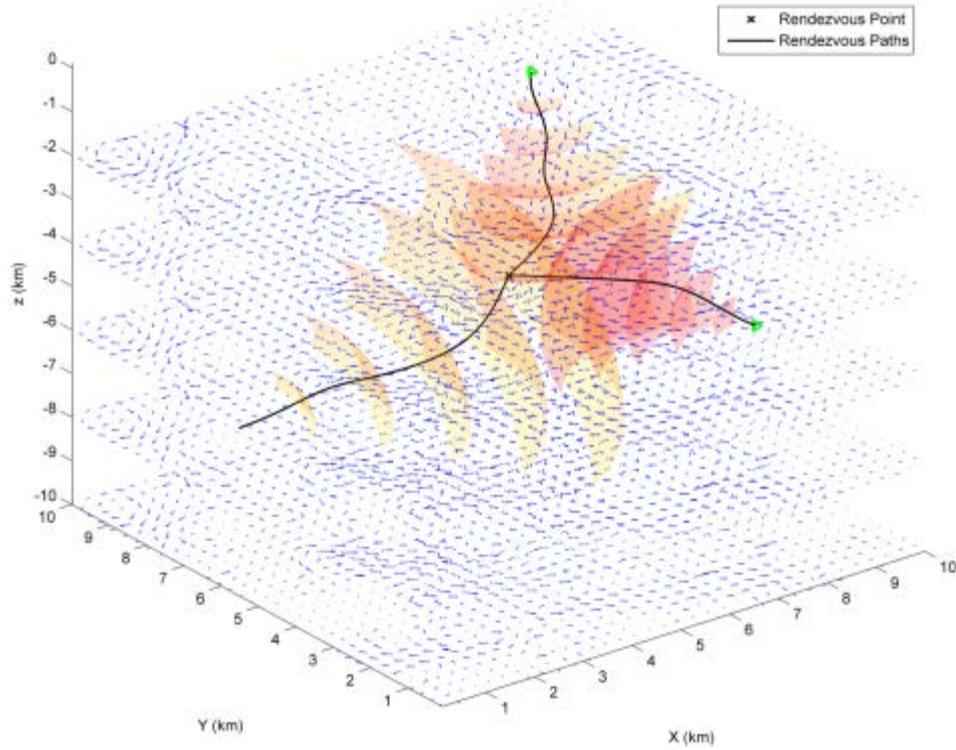
Figure 5.8 Trajectories produced by the path planner based on the DSSD scheme for a serve vehicle rendezvous with two worker vehicles in 3D scenario.

Table 5.2 Path planning for a serve vehicle rendezvous with two worker vehicles

Scenarios	Time (sec)				
	AUV-I	AUV-II	AUV-III	Factor $\mathbb{M}$	Factor $\mathbb{S}$
Optimized with QPSO-DSSD with currents	3264	3255	3209	9727	63
Straight line path with currents	2860	6733	2318	11911	8289
Optimized trajectory without currents	2886	4577	3371	10834	2897

It should also be noted that the recorded waiting time $\mathbb{S}$ for the vehicles following the optimized trajectories was 86 second for the 2D scenario, and 63 second for the 3D scenario. These times are negligible compared with the sum of the travel times $\mathbb{M}$ for each scenario - around 0.12% for the 2D scenario and around 0.64% for the 3D. It can be concluded that the proposed QPSO path planner based on the DSSD scheme performs well in solving path planning problem for AUV simultaneous rendezvous in variable currents.

Case 5.6.2-3: Path Planning for Multiple AMVs Rendezvous in a Complex Ocean

This case study involves three AMVs with starting positions of (10,10), (40,90) and (90,40). Their goal was to make their way through a region with variable currents and rendezvous at a certain point. The water-referenced speed of the vehicles was 3m/s, 1.5m/s and 2m/s. Figure 5.9(a) and (b) shows the results of the scenario in which the operations environment came with or without irregularly shaped terrain and dynamic suspended obstacles. The obstacles are configured individually to move independently at various rates (-0.05m/s to 0.05m/s along both X and Y directions). Their uncertain positions are assumed linearly propagated with time at rates randomly chosen between 0.005m/s to 0.035m/s. The positions of the obstacles are represented by the mean and variance of their uncertainty distribution.

While the current environments in both scenarios are equivalent, the difficulties posed by the static terrain and dynamic obstacles become more challenging. By comparing the convergence curves shown in Figure 5.9(b), it is evident that as the scenario becomes increasingly difficult, the convergence speeds of the QPSO-DSSD based path planner slows down. As can be seen in Table 5.3, it is also evident that the vehicle takes less time $\mathbb{M}$ for the scenario without obstacles and terrains: 66395s, compare with that of the scenario with obstacles and terrains: 69791s.
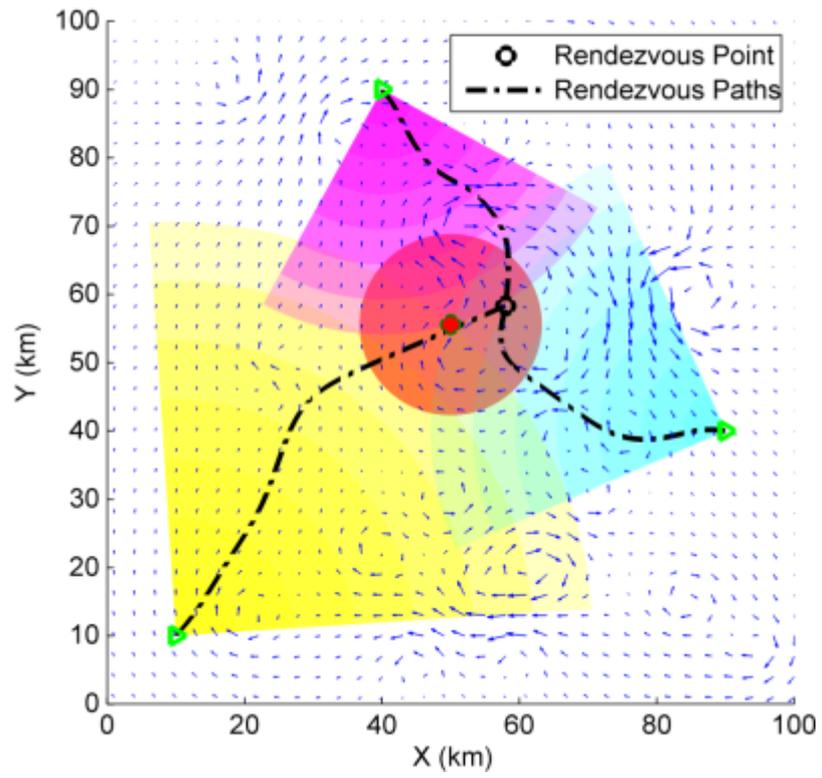
(a)
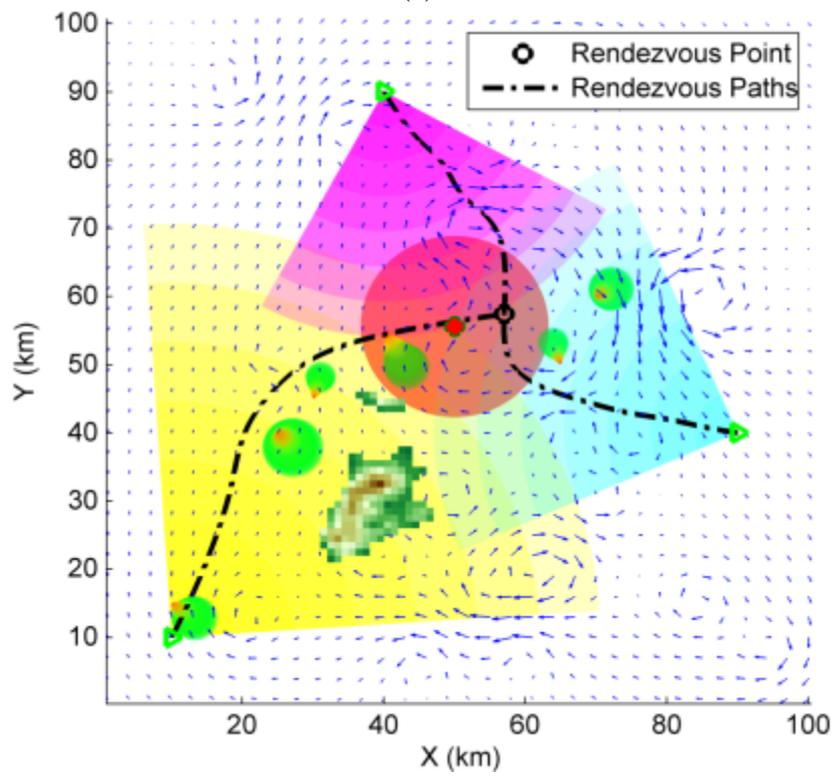
(b)
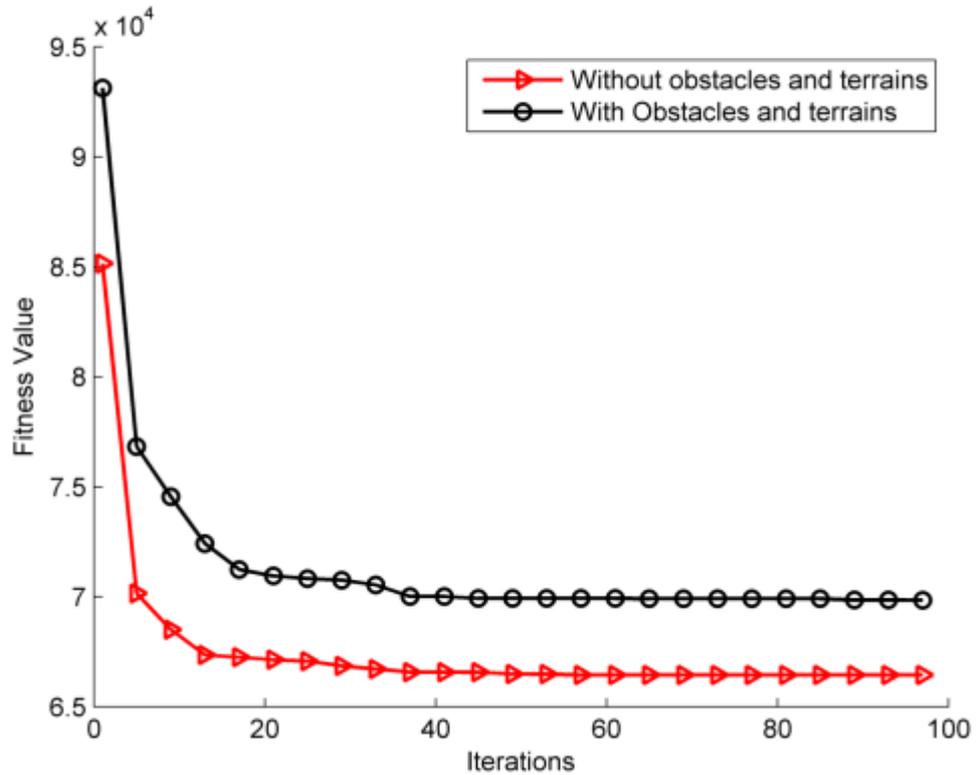
(c)

Figure 5.9 AUVs rendezvous based on the QPSO-DSSD scheme in different scenarios: (a) paths planned in a currents field without obstacles and terrains, (b) paths planned in a currents field with obstacles and terrains, (c) convergence curve of best fitness values.

Table 5.3 A comparison of AUVs' mission times achieved by the QPSO-DSSD path planner in different environment.

Scenarios	Time (sec)				
	AUV-I	AUV-II	AUV-III	Factor $\mathbb{M}$	Factor $\mathbb{S}$
Without obstacles & terrains	22114	22147	22133	66395	46
With obstacles & terrains	23281	23273	23237	69791	51

However, the complexity of the scenario had only a slight effect on the simultaneous arrivals calculated by the proposed QPSO-DSSD path planner. The sum of the waiting times $\mathbb{S}$ were 46s (without obstacles and terrains) and 51s (with obstacles and terrains), both are under 0.01% of their corresponding travel time.

5.6.3 Case Studies with Rendezvous Point Selection Schemes

Case 5.6.3-1: Path Planning for Multi-AMV Rendezvous in a Variable Ocean

Figure 5.10 shows the result of the trajectory optimization for three AMVs rendezvous in a region with variable currents. The optimal path projections generated by the proposed schemes: centroid, mass-centre, optimized full-scale and optimized mass-centre, are shown in Figure 5.10(a). The magenta sphere represents the rendezvous point searching region for the optimized mass-centre scheme, the sphere is situated at the centre of the rendezvous point of the mass-centre scheme. The convergence curves showing the best fitness values for each method with the corresponding numbers of iterations is shown in Figure 5.10(b).
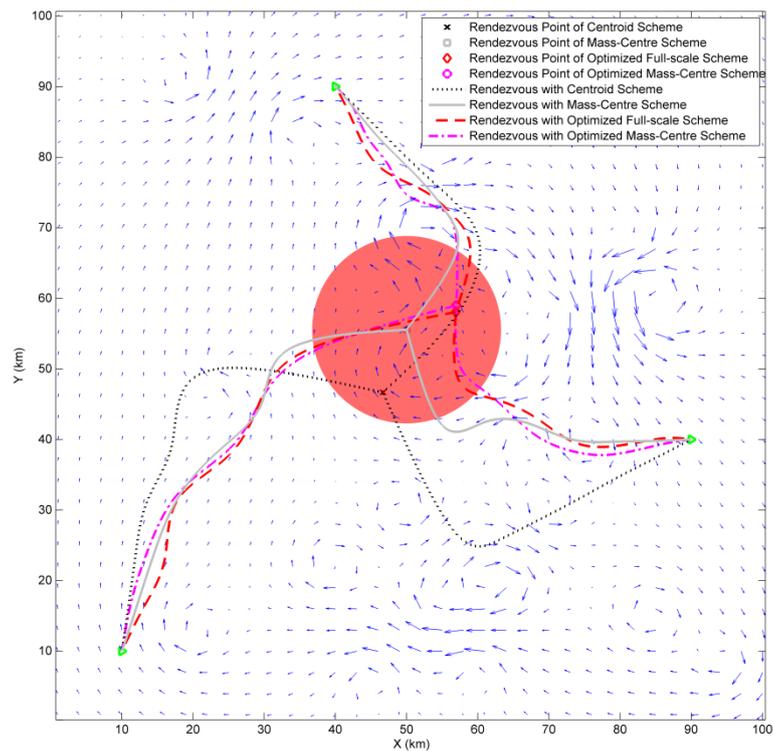
(a)
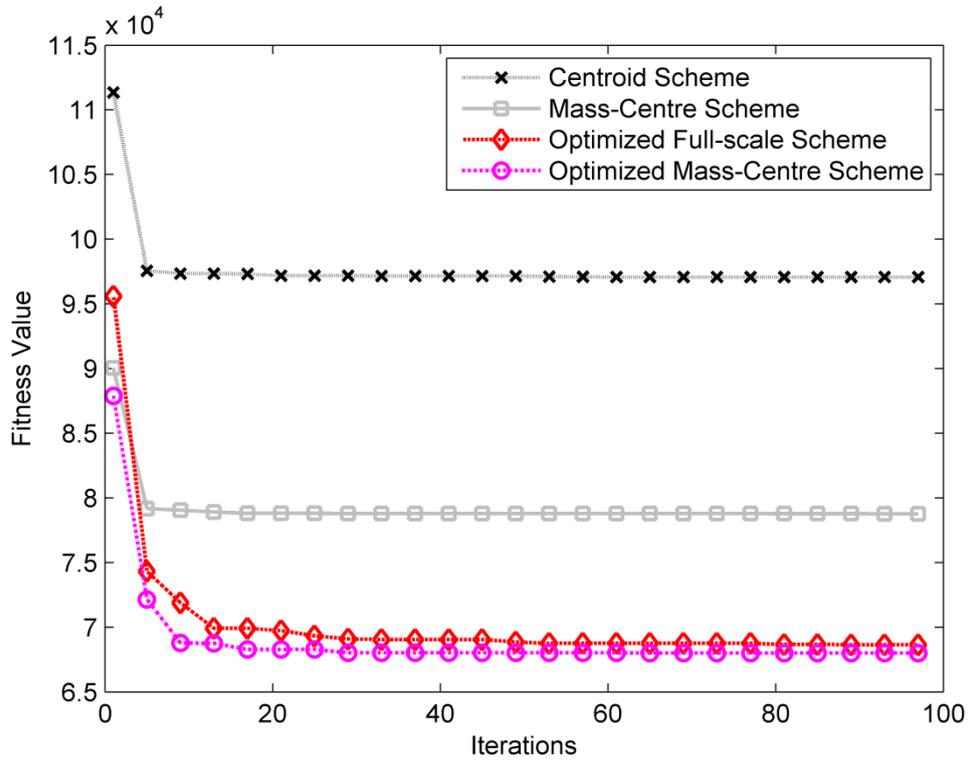
(b)

Figure 5.10 Comparison of results produced by the path planner based on Centroid, Mass-centre or Optimized full-scale and Optimized mass-centre rendezvous point selection schemes: (a) path projections in a spatial variable currents field, (b) convergence curve of best fitness values.

Table 5.4 A comparison of AUVs' mission times achieved with different rendezvous point selection schemes

Rendezvous Schemes	Time (sec)				
	AUV-I	AUV-II	AUV-III	Factor $\mathbb{M}$	Factor $\mathbb{S}$
Centroid	21349	32357	31437	85144	11928
Mass-centre	20478	26258	25556	72291	6482
Optimized full-scale	22510	22867	22865	68242	358
Optimized mass-centre	22459	22611	22613	67683	156

By comparing the convergence curves shown in Figure 5.10(b), it can be seen that both centroid and mass-centre schemes convergence relatively faster than those of the optimized full-scale and mass-centre schemes. The reason can be found from Section 5.4 that the rendezvous points of the centroid and mass-centre schemes are deterministically selected, whereas in the case of the optimized full-scale/mass-centre schemes, the rendezvous points are engaged in the optimization process, consequently, these two schemes will have higher dimensional search-spaces than the deterministic

schemes and thus slower convergence.

It is also evident from Figure 5.10(b) and Table 5.4 that the proposed hybrid optimized mass-centre scheme achieves better results than the centroid/mass-centre and optimized full-scale scheme. The reasons can be found from Figure 5.10(a). The proposed centroid/mass-centre scheme fixed the rendezvous points, while in contrast, the optimized mass-centre scheme allows more flexibility to search around the first-attempt approximate solution (rendezvous location of the mass-centre scheme) and generate more optimized paths for AUVs which reduces the overall travelling time $\mathbb{M}$. In principle, although the optimized full-scale should also offer the same flexibility as the optimized mass-centre scheme and have the potential advantage of searching all possible locations of the rendezvous point over the whole space, it is more likely to become stuck at a local minimum because of its undirected searching.

The corresponding travelling time of each AUV's planned trajectory is recorded in Table 5.4, where it can be observed that corresponding sum of the waiting times $\mathbb{S}$ for all AUVs with the centroid scheme was significantly higher than the other three algorithms, This indicates that the centroid scheme is preliminary since vehicles if travelling at various speeds will obviously arrive at their geometric centre at different times. On the other hand, the AUVs in the optimized mass-centre scheme had the lowest execution time of the four methods.

Case 5.6.3-2: AUVs Rendezvous in a Dynamic, Cluttered, and Uncertain Ocean

Figure 5.11 and Figure 5.12 show the results for multiple AUVs navigating through a spatiotemporal currents field with irregularly shaped terrains and dynamic suspended obstacles for the purpose of making a rendezvous in 2D and 3D scenarios. With current environments being equal, dealing with static terrains and cluttered dynamic obstacles with growing uncertainty in positions, posed real challenges for the vehicles, especially when simultaneous arrival was a criterion. Note that the terrains and obstacles were placed deliberately to block the straight-line trajectories. The centres of the obstacles (relating to the mean of the distribution) are configured to move independently at various rates (-0.25m/s to 0.25m/s in both X and Y component) in different directions, while their uncertainty in terms of position is assumed linearly propagated with time at the rates randomly chosen between 0.066m/s to 0.166m/s. it was assumed that the obstacles would move at their pre-set rates and directions during the mission period of the AUVs. Since the propagation of the uncertainty is assumed linear with time, it is expressed in the plots shown in Figure 5.11(c)-(f) and Figure 5.12(c)-(f) as a proportional increment in the collision boundary encircling the object. The position uncertainty of the obstacles at

different times is represented in these plots by different colours. At each discrete time point, if the vehicle location is not inside the correspondingly coloured obstacle boundary, no collision would occur.
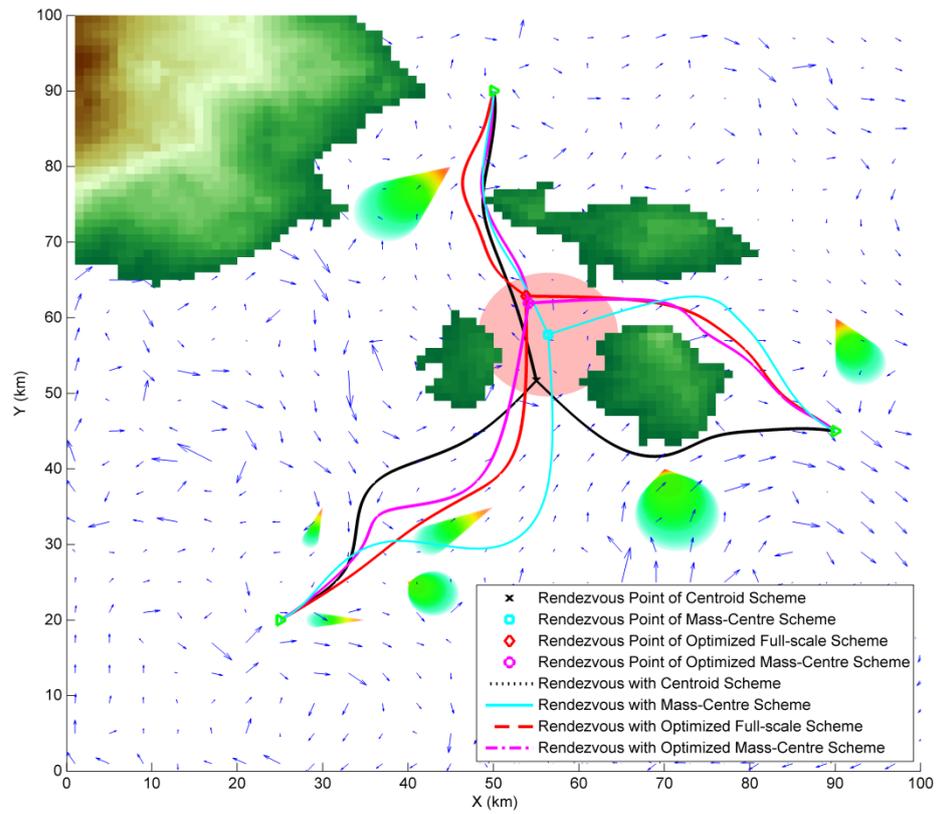
(a)
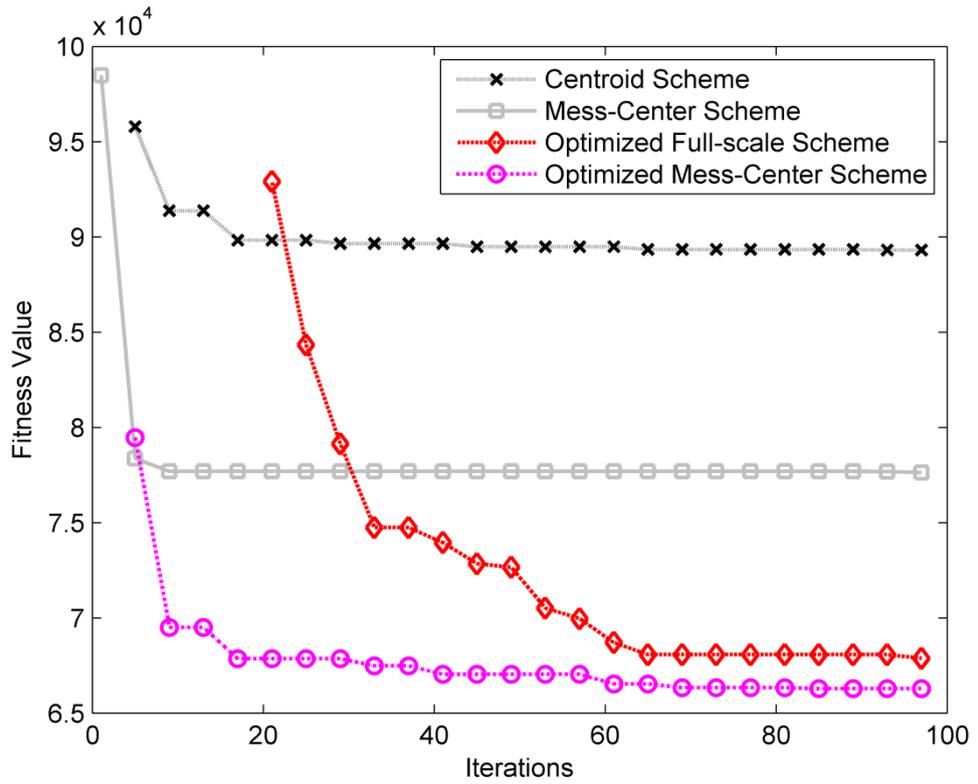
(b)

(c)
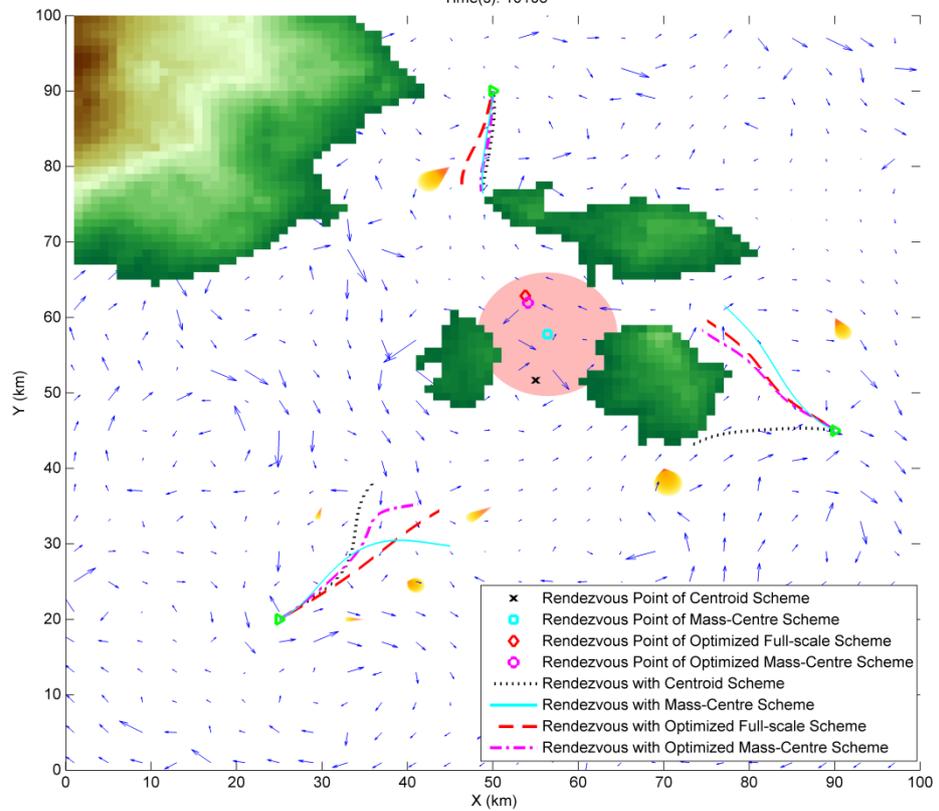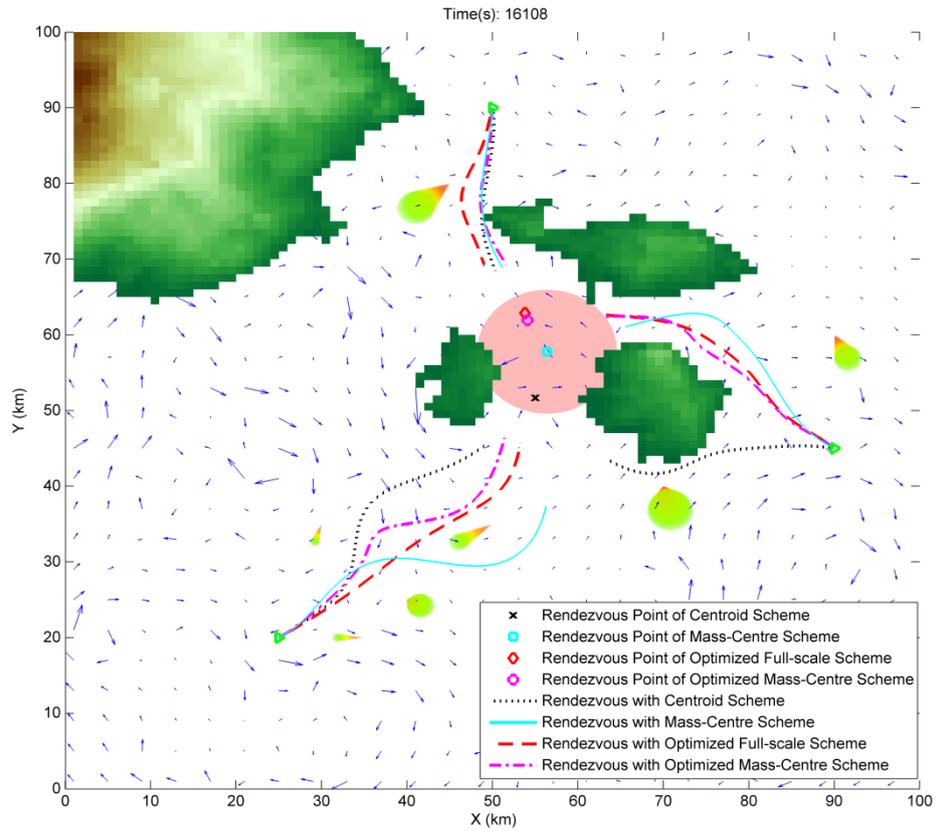
(d)
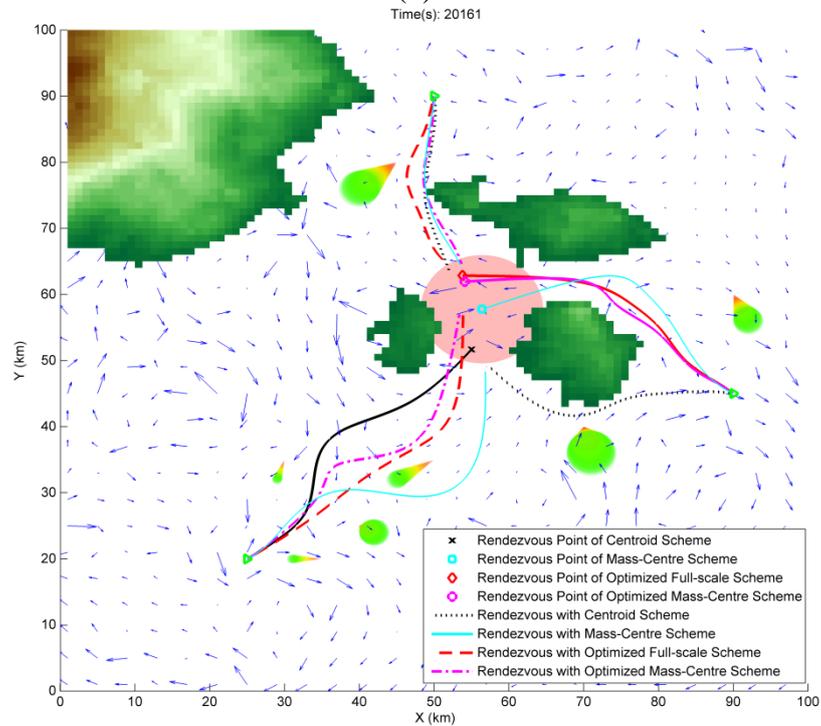
(e)
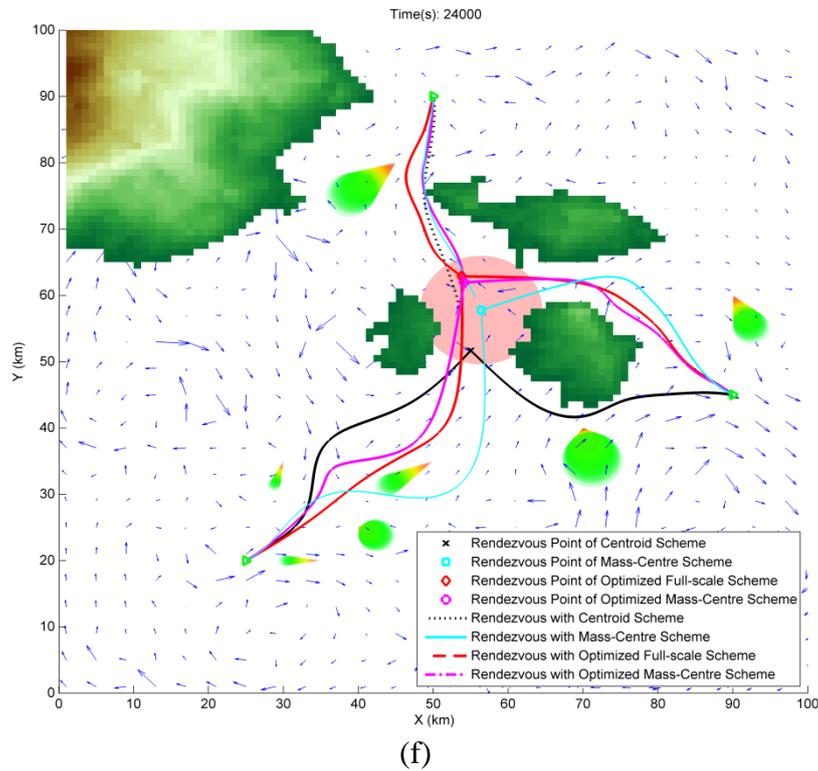
(f)

Figure 5.11 Comparison of results for the 2D scenarios produced by the path planner based on centroid, mass-centre or optimized full-scale and optimized mass-centre rendezvous point selection schemes: (a) path projections in a current field with irregularly shaped terrains and dynamic obstacles, (b) convergence curve of best fitness values, (c)–(f) representative screenshots of the executed trajectories at various times.

Table 5.5 A comparison of AUVs' mission times achieved with different rendezvous point selection schemes in 2D scenario

Rendezvous Schemes	Time (sec)				
	AUV-I	AUV-II	AUV-III	Factor $\mathbb{M}$	Factor $\mathbb{S}$
Centroid	19452	29770	22788	72010	17299
Mass-centre	24157	25879	20287	70323	7314
Optimized full-scale	22487	22436	19972	64895	2566
Optimized mass-centre	22087	22094	19975	64156	2125

As can be seen, the trajectories generated by all schemes succeeded in avoiding static irregular terrains as well as the clustered dynamic obstacles.

Figure 5.11(c)-(f) show the corresponding trajectories, as generated by these four strategies, and tracked by their corresponding vehicles in a time sequence for the 2D scenarios. By comparing the convergence curves shown in Figure 5.11(b), it is evident that the fitness values for trajectories generated with deterministic centroid or

mass-centre schemes are at the very top of those generated with the heuristic optimized full-scale or mass-centre schemes. Since the proposed optimization is a minimization problem, the smaller the fitness value, the better the performance, which means the path planner based on heuristic methods is able to achieve a more optimized trajectory than the deterministic methods. By comparing the convergence curves of the path planner based on the optimized full-scale and mass-centre schemes, it is clear that the optimized full-scale scheme has significantly slower rate of convergence relative to the optimized mass-centre scheme. According to Figure 5.11(b), the optimized mass-centre scheme converges around 20 iterations, in contrast, the optimized full-scale only comes up with a first validate solution around 20 iterations and not able to converge until 60 iterations. The reason can be found from Figure 5.11(a) which reveals that the path planner with the optimized mass-centre scheme is directed to search within the magenta sphere for the optimal location of rendezvous point leading to efficient searching, whereas in the case of the optimized full-scale scheme, path planner searching over the entire space for the optimal rendezvous location, most of the locations are very unlikely to be the potential solution, and is thus a waste of computational effort to those solutions.

Similar experiments were performed in a 3D scenario and the results are shown as in Figure 5.12. As can be seen, the AUVs were challenged by having to navigate through a narrow valley or need to climb over territory to reach the rendezvous point. In Table 5.6, it is easy to see that the vehicles taking the trajectories provided by the planner that apply the optimized mass-centre scheme take significantly less time to arrive at the rendezvous point than the vehicles taking the trajectories based on other schemes - over 11% less compare to centroid scheme and over 5% less compare with mass-centre scheme. It is obvious that there is indeed a lower Factor $\mathbb{S}$ value for both heuristic optimized full-scale/mass-centre schemes than deterministic centroid/mass-centre schemes. It is worth to note that the value of Factor $\mathbb{S}$ for the optimized mass-centre scheme is less than one second, which is negligible compare with all the other schemes. Since the Factor $\mathbb{S}$ value reflects the performance of simultaneous arrivals, it can be concluded that the heuristic optimized schemes achieves better results than the other two deterministic algorithms.

(a)

(b)
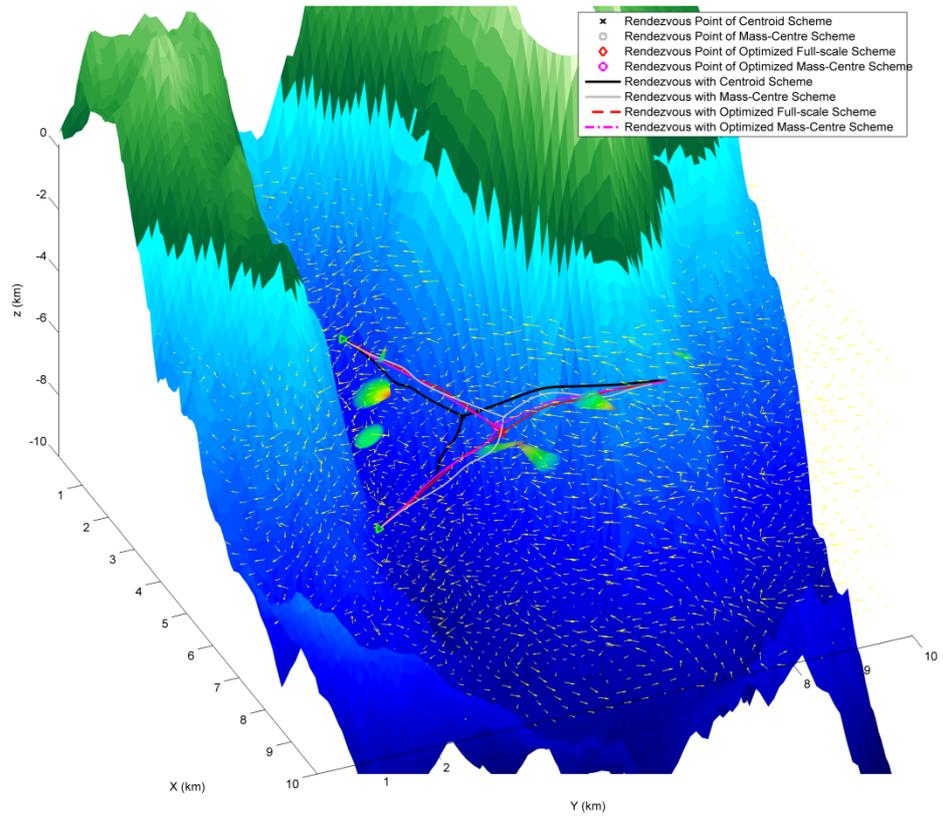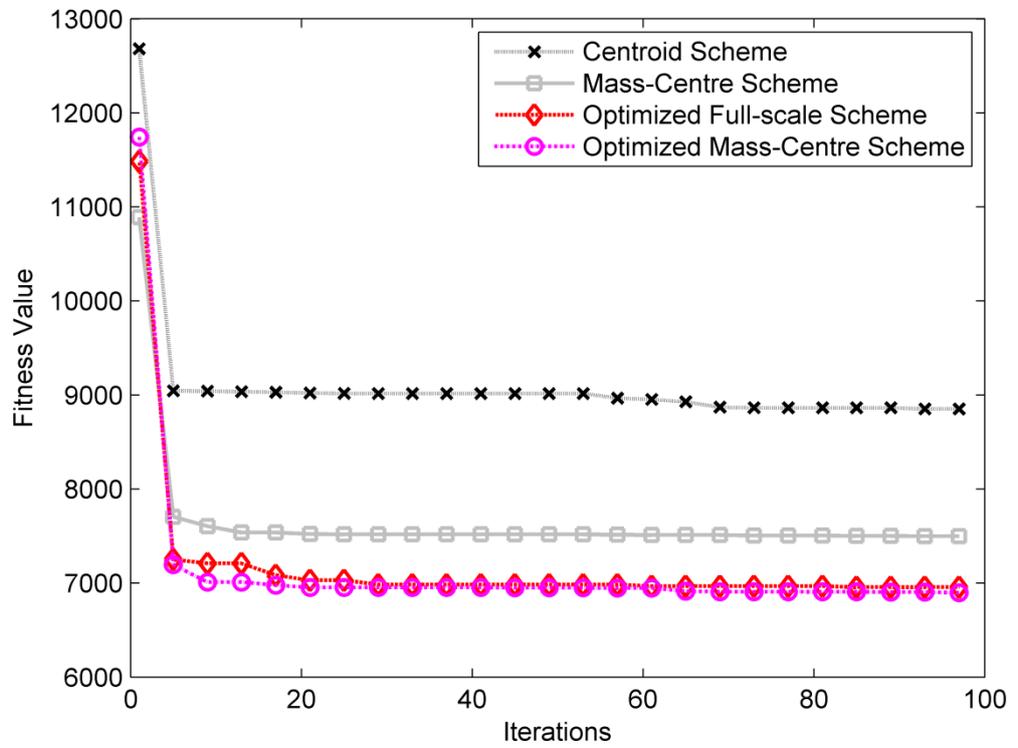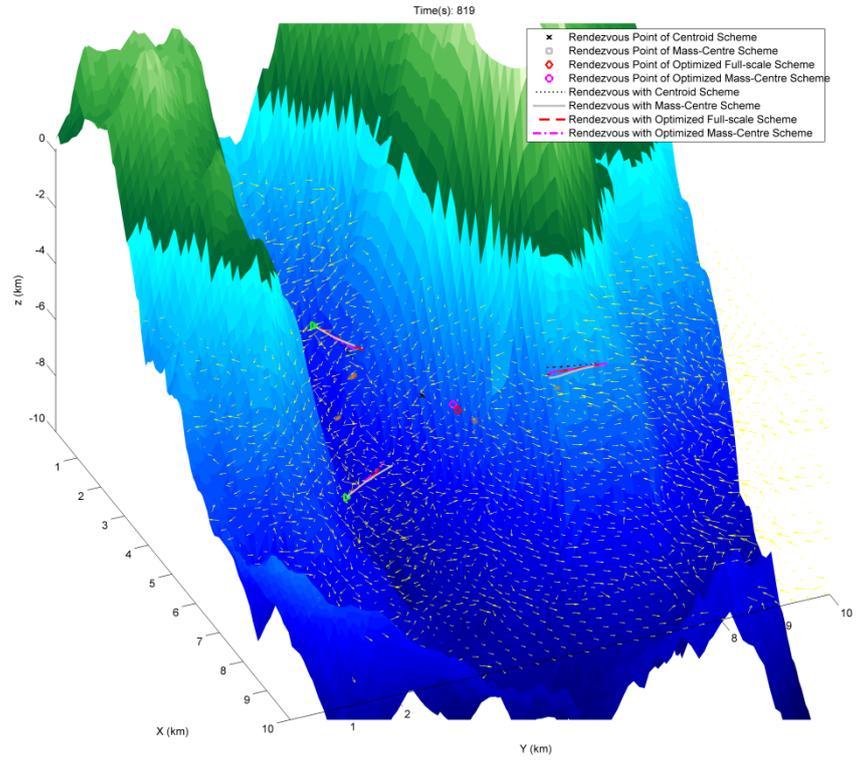
(c)
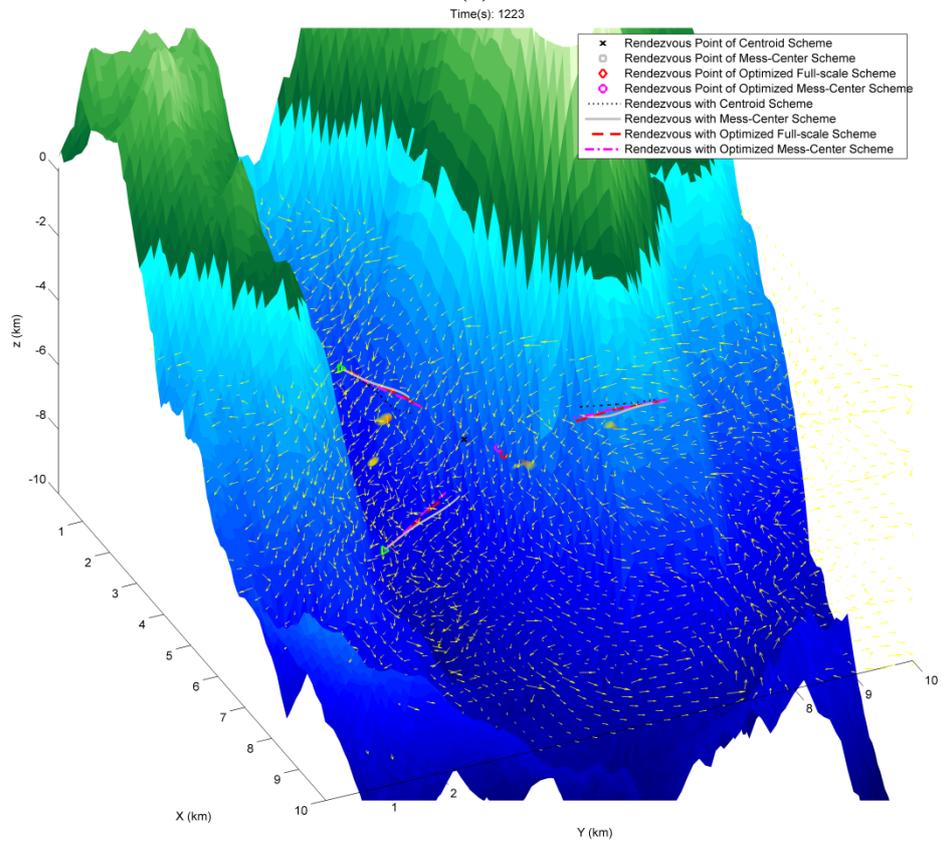
(d)
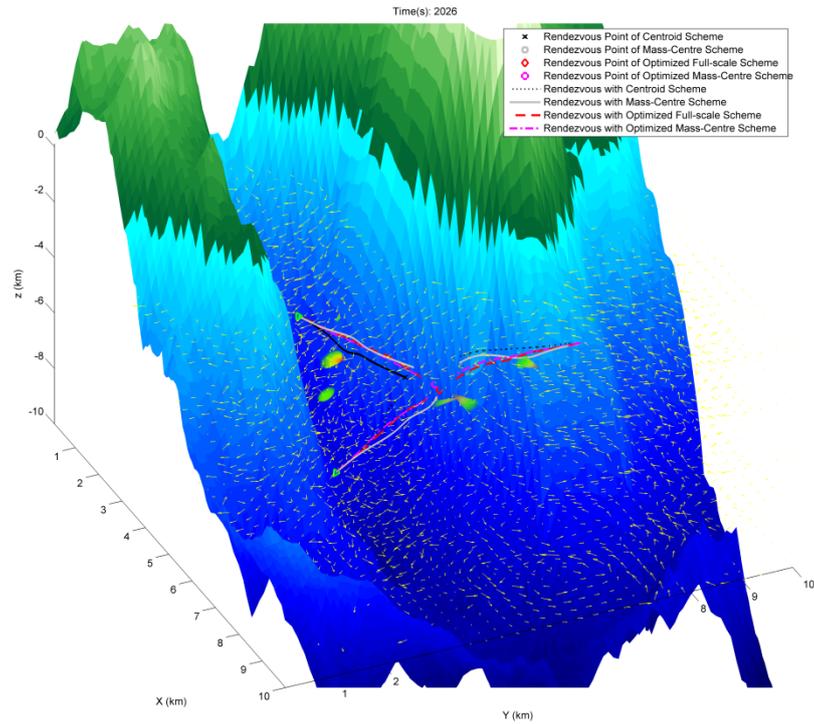
(e)
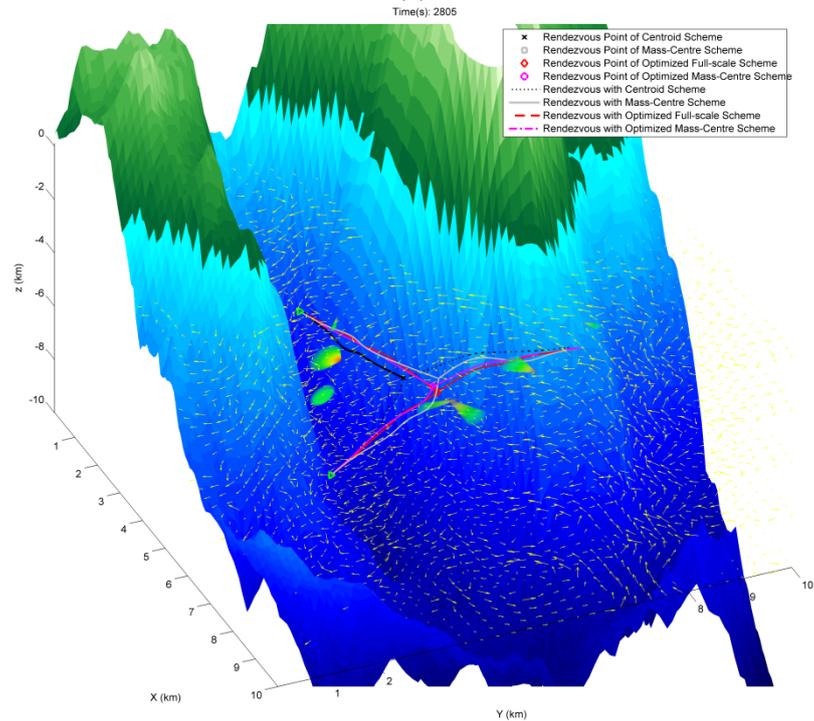
(f)

Figure 5.12 Comparison of results for the 3D scenarios produced by the path planner based on centroid, mass-centre or optimized full-scale and optimized mass-centre rendezvous point selection schemes: (a) path projections in a current field with irregularly shaped terrains and dynamic obstacles, (b) convergence curve of best fitness values, (c)–(f) representative screenshots of the executed trajectories at various times.

The performance of the centroid scheme in both 2D and 3D scenario are all relatively poor. As illustrated in Table 5.5 and Table 5.6, the sum of the time consumed (Factor $\mathbb{M}$) and the sum of the waiting times (Factor $\mathbb{S}$) using centroid scheme are the biggest of the four schemes.

Table 5.6 A comparison of AUVs' mission times achieved with different rendezvous point selection schemes in 3D scenario

Rendezvous Schemes	Time (sec)				
	AUV-I	AUV-II	AUV-III	Factor $\mathbb{M}$	Factor $\mathbb{S}$
Centroid	1892.15	2949.37	2907.27	7748.79	1099.32
Mass-centre	2279.80	2496.05	2500.03	7275.89	224.19
Optimized full-scale	2315.91	2318.86	2317.01	6951.79	4.78
Optimized mass-centre	2298.46	2298.70	2298.00	6895.18	0.94

Both these simulations further testify the optimized mass-centre scheme can help path planners to make better use of the searching efforts at meaningful region. Use of this scheme is more likely to result in the AUVs reaching the rendezvous point together and in less time.

5.6.4 Case Studies with Optimal Operational Speed Schemes

Case 5.6.4 -1: QPSO-DSSD Path Planner with Optimal Operational Speed

Figure 5.13 shows the performance of the path planner with both deterministic operational speed and optimized operational speed. With the deterministic scheme, the vehicles' water-referenced velocity is fixed at 2.5m/s, 1.5m/s and 2m/s: these are assumed to be the vehicles' full speed. In contrast, the scheme applying optimized operational speed gives path planners the flexibility to find the optimal operational speed so as to reduce costs. To improve the searching efficiency of this scheme, the operational speed is constrained with the lower bound at 90% of the full speed, and the upper bound is the full speed.

Figure 5.13(a) shows the trajectories generated by these two strategies. As can be seen in this figure, it appears that the paths generated on the basis of deterministic operational speed tend to go straight to the rendezvous point while the paths generated on the basis of optimized operational speed are more curvature, these curvature reveal that the paths will be able to make better advantage of the favourable currents. As mentioned Section 5.4, given that the dimension of a particle

increases at optimised operational speed, higher dimension of the particle will further results in slower convergence speed. As shown in Figure 5.13(b), the convergence curve of the scheme with optimized operational speed is slower than that of the scheme with deterministic operational speed.
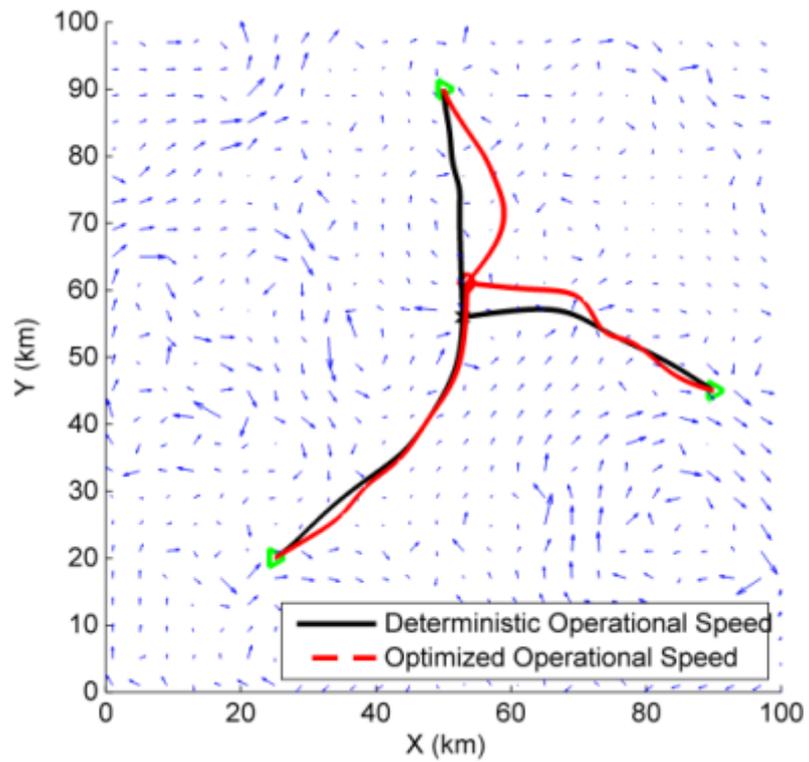
(a)

(b)

(c)

(d)

(e)

(f)
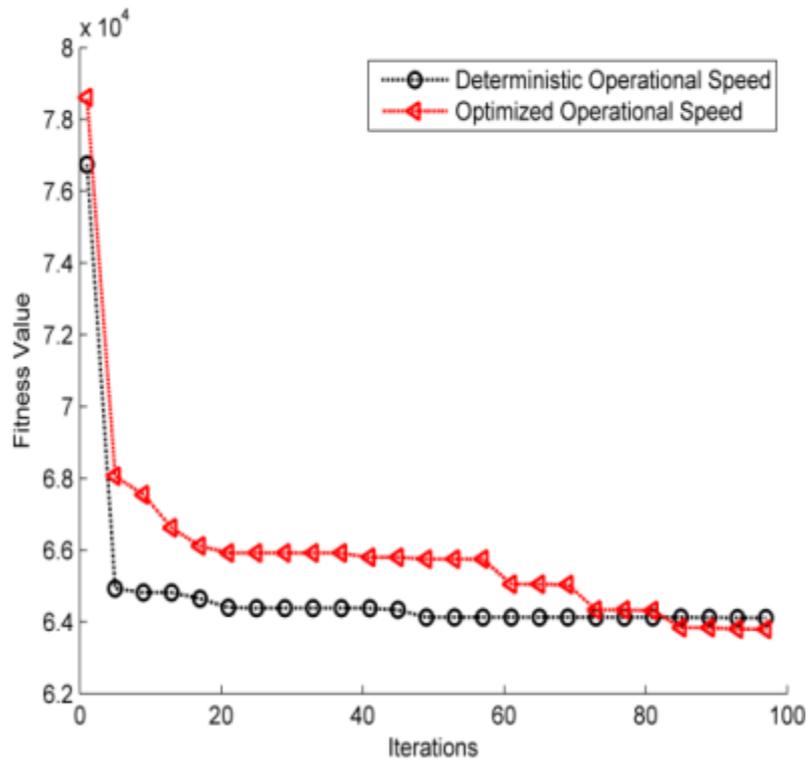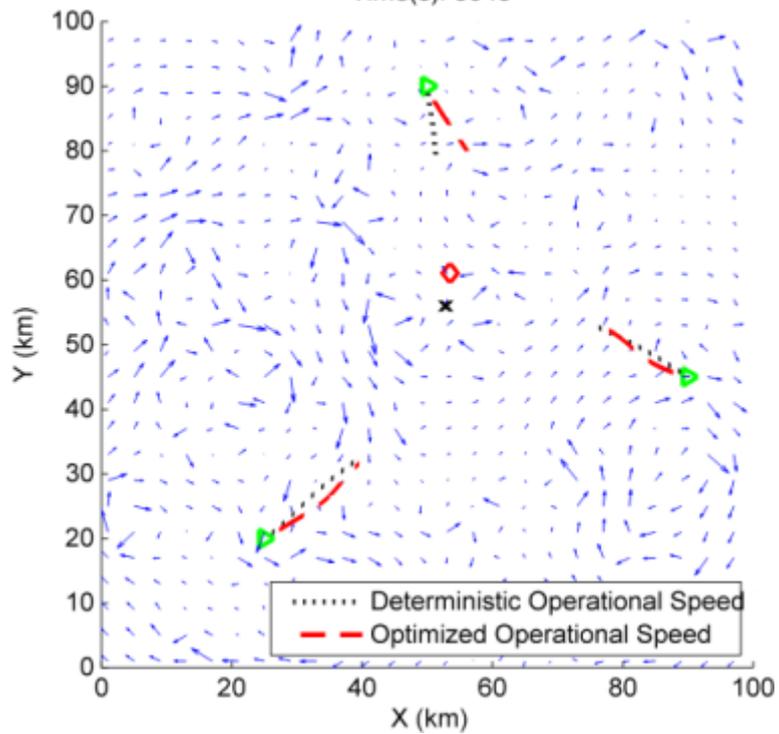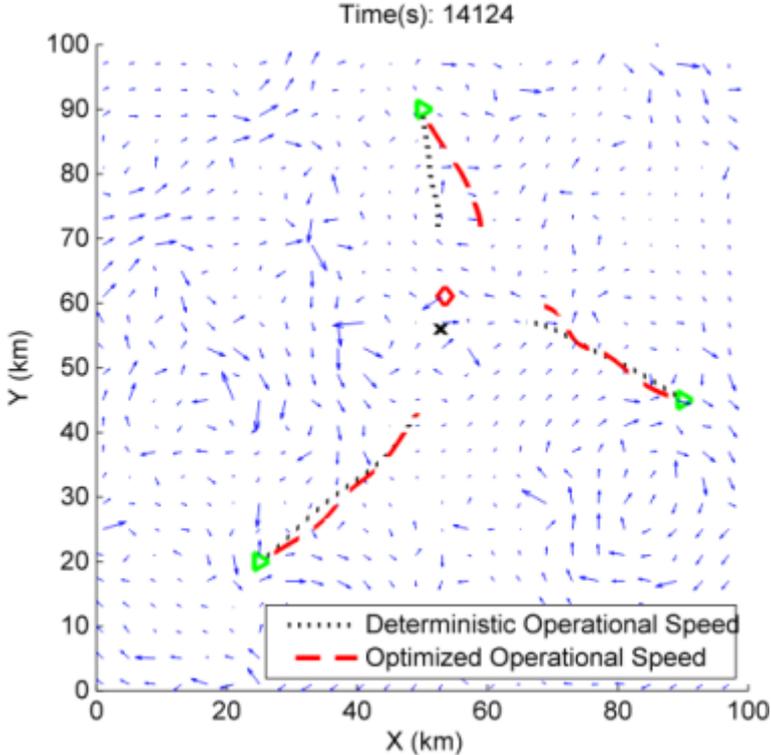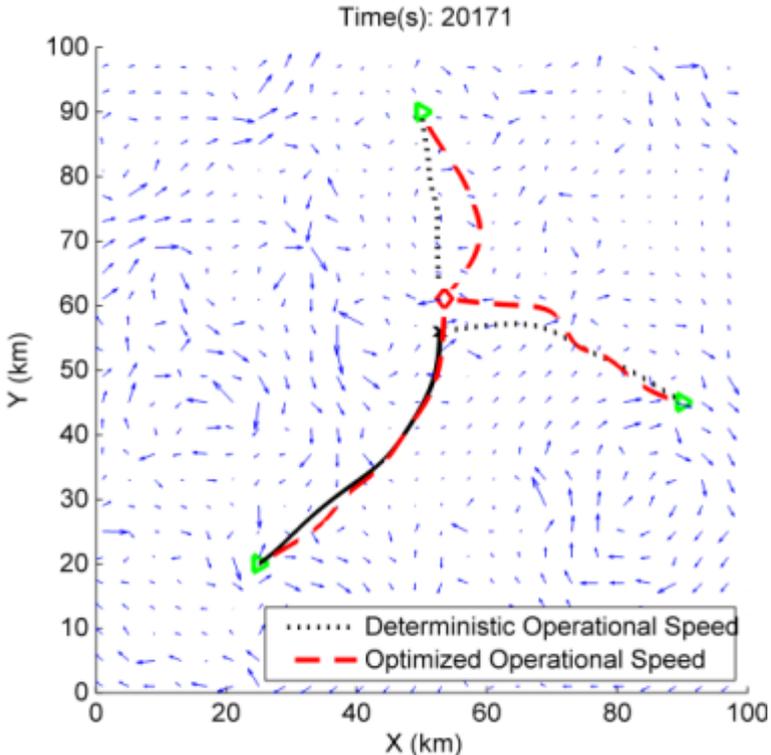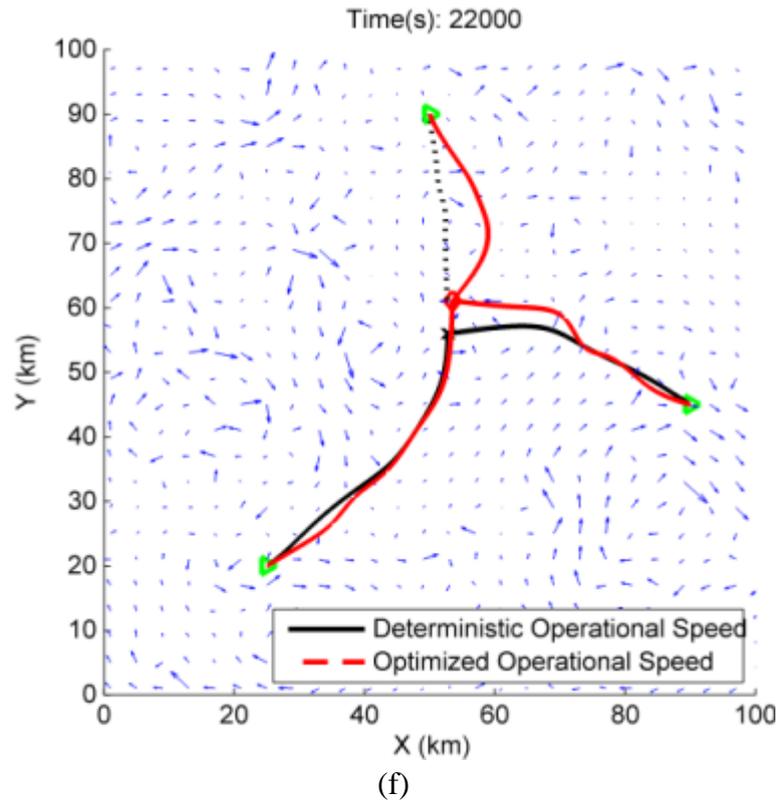
Figure 5.13 Comparison of results produced by the QPSO-DSSD path planner with deterministic and optimized operational speeds for participating vehicles: (a) path projections in a current field, (b) convergence curve of best fitness values, (c)–(f) representative screenshots of the executed trajectories at various times.

Both the travel time and power consumption are measured for each scheme, as recorded in Table 5.7. By comparing the sum of time consumption of these trajectories shown in this table, it is easy to see that the vehicles taking the trajectories provided by the planner based on the optimized operational speed consumes more travel time than the vehicles taking the trajectories produced based on deterministic operational speed. The reason should be evident from Table 5.7 that the proposed scheme with optimized operational speed chooses not to run at vehicles' full speeds. This contributes to longer mission duration for the vehicles get to the rendezvous point. This small "investment" has been greatly rewarded, as recorded in Table 5.7, the sum of waiting time of the trajectories planned with deterministic operational speed is 1,784s whereas that of the trajectories planned with optimized operational speed is reduced significantly to 352s. Therefore, it is testify that, compared to path planner with deterministic operational speed, path planner with optimized operational speed produce trajectories that allow the vehicles to rendezvous

closer to each other.

Table 5.7 A comparison of solutions achieved by the QPSO-DSSD path planner with deterministic and optimized operational speed

Rendezvous Schemes (Optimal Speeds)		AUV-I	AUV-II	AUV-III	Factor $\mathbb{M}$	Factor $\mathbb{S}$
Deterministic	Time (s)	21372	21362	19598	62332	1784
	Energy	333939	72097	156787	1189017	14224
Optimized	Time (s)	21241	21168	20963	63372	352
	Energy	327309	69607	140612	537528	2109
	Speeds(m/s)	2.48	1.48	1.88		

It should also be noted that the power consumption of QPSO-DSSD path planners with optimized operational speed are far less than that with deterministic operational speed, as can be seen from Tables 5.7, the power saving for travel is about 55% with additional power saving for waiting to get rendezvous at about 85%. Such a significant change in the power consumption indicates that path planner with optimized operational speed has an advantage over than that with deterministic operational speed.

Case 5.6.4 -2: AUVs Rendezvous with Optimal Operational Speeds in Australia/New Zealand Ocean Region

In order to test the validity of the proposed planners in the realistic maritime missions, a region of the ocean between Australia and New Zealand was selected. With strong currents between the Antarctic-driven southeast portions of the Indian Ocean and the Tasman Sea's Pacific Ocean waters, this region is renowned for its turbulent seas. The currents field used here is an actual measurement of currents information provided by the Australian Bureau of Meteorology, available at ftp://ftp.bom.gov.au/anon/sample/access/oceanmaps/. This currents data provides a gridded prediction of the current fields up to 7 days at a 10 km gird resolution from the OceanMAPS. The ocean terrain information used here is obtained from ETOPO, provided by NOAA's National Geophysical Data Center (NGDC), ETOPO is a global relief model of Earth's surface that integrates land topography and ocean bathymetry built from numerous global and regional data sets. For detailed information, see the NGDC web site

http://www.ngdc.noaa.gov/mgg/global/global.html.

It is considered in this study the mission of minimum-time rendezvous of four vehicles launched from Sydney, Brisbane, Hobart of Australia and Auckland of New Zealand. The searching region for this mission, shown in Figure 5.14(a), from 146°E to 176°E zonal, -47.8°S to -17.8°S.

(a)
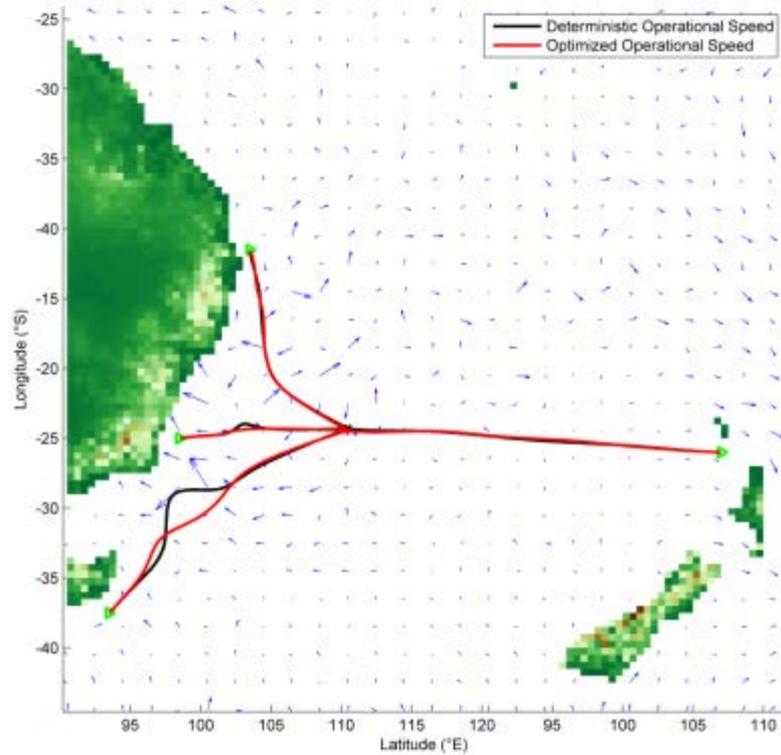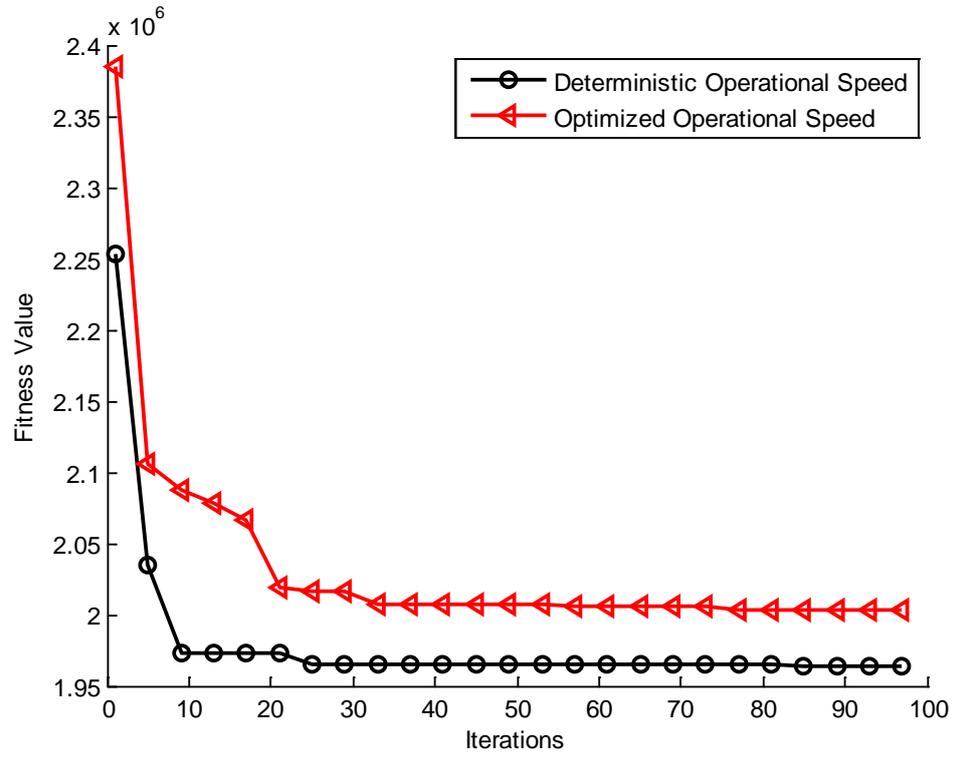
(b)
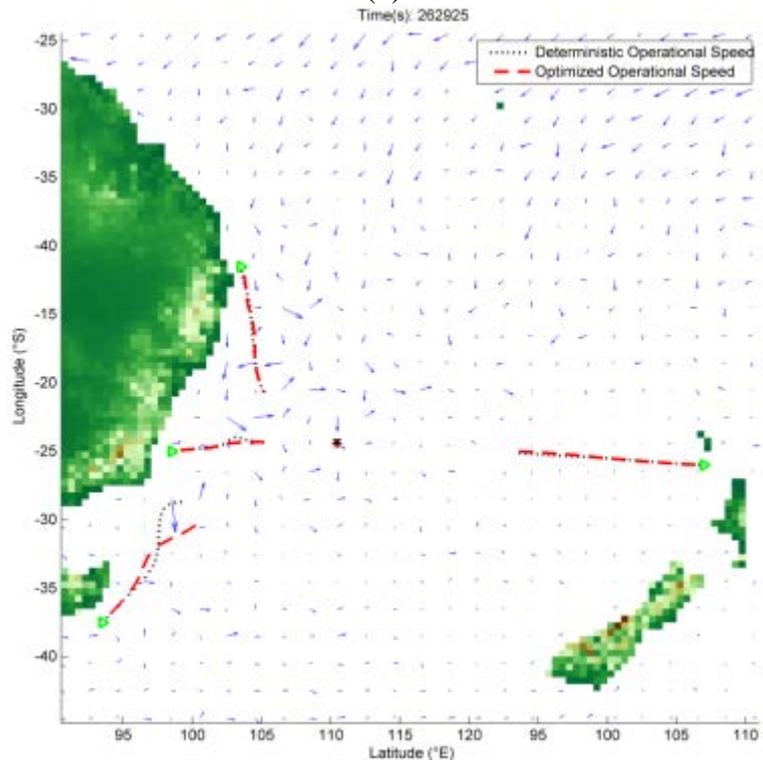
(c)
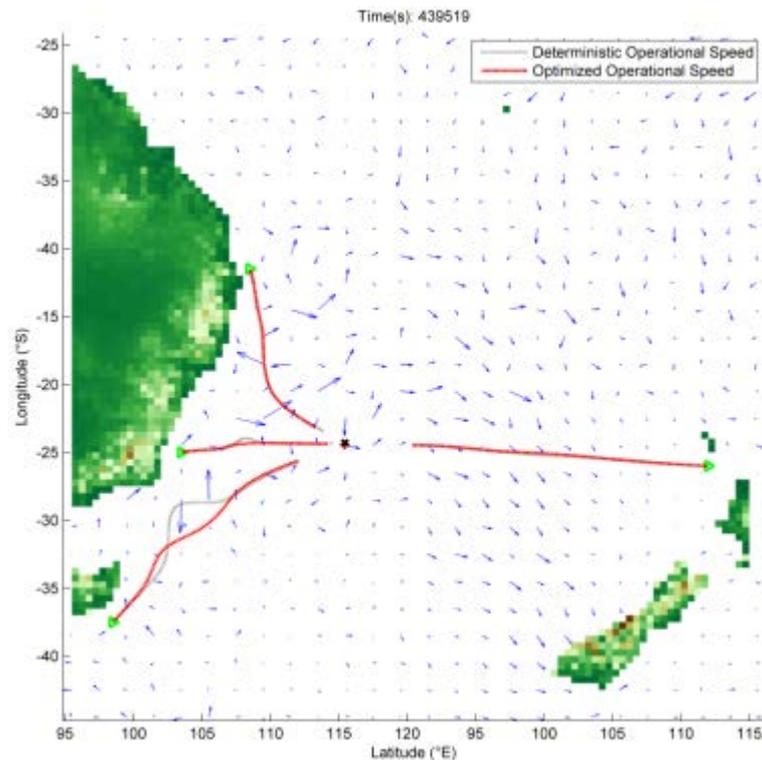
(d)

Figure 5.14 Comparison of results produced by the path planner with determined and optimized operational speeds in Australia/ New Zealand Ocean Region: (a) path projections for four AUVs rendezvous, (b) convergence curve of best fitness values, (c)–(d) representative screenshots of the executed trajectories at various times.

Path planners with both deterministic operational speed and optimized operational speed are applied to generate optimal trajectories for AUVs to get rendezvous. Figure 5.14 display the simulation results of this study, and corresponding data is recorded in Table 5.8. From the detailed statistical results presented in Table 5.8, it is easy to see that the QPSO-DSSD path planners with both deterministic and optimized operational speed demonstrate high simultaneousness for rendezvous. They generate the solutions with small sum of waiting time relative to the sum of travel time, 2.2% and 0.8% (Factor $\mathbb{S} / \mathbb{M}$), among the four AUVs. By comparing the results of the two path planners, it can be seen that QPSO-DSSD path planners with optimized operational speed provides large power saving (especially the proportion for waiting to get rendezvous is reduced over 78%), but it also increases travel time. Meanwhile, compare the corresponding sum of waiting time for the two planners, it is not hard to find the trajectory provided by the scheme with optimized operational speed consumes significantly less waiting time (25,721s), around 60% less, than that

173

consumed for the trajectory provided by deterministic scheme.

Table 5.8 Comparison of solutions achieved by the QPSO-DSSD path planner with deterministic and optimized operational speeds in Australia/New Zealand Ocean

Rendezvous Schemes (Optimal Speeds)		AUV-I	AUV-II	AUV-III	AUV-IV	Factor $\mathbb{M}$	Factor $\mathbb{S}$
Deterministic	Time (s)	490041	484850	490668	454236	1919796	42878
	Energy	1653889	7575778	13248049	3633890	26111606	384492
Optimized	Time (s)	488842	499187	499281	492658	1979968	17157
	Energy	1460222	6762178	12853741	3892248	24968389	84789
	Speeds (m/s)	1.44	2.38	2.95	1.99		

Similar experiment result obtained on Case 5.6.4-1 and Case 5.6.4-2 testify that the QPSO-DSSD path planners with optimized operational speed can produce trajectories with less power consumption and better performance of simultaneousness to get rendezvous.

5.7 Robustness Assessment

In this section, a set of representative Monte Carlo simulations are performed on the planners to determine under what circumstances improvements can be expected through the use of planner based on the QPSO-DSSD scheme. The first subset of simulations performed involves vary the vehicle's water-referenced speed and vehicles' initial positions, respectively, to investigate how it affects the performance of the four rendezvous schemes. The second subset of simulations varies the population size that is used in the optimization algorithm. Higher population size provides more computation load to search for a better solution. Here how this will affect the planner's performance will be examined. In the third set of simulations, it is assumed that various quantity of vehicles are participating the rendezvous mission. This subset of simulations focuses on the performance of the planners in a challenge scenario with AMV swarm. The performances of the rendezvous schemes are compared based on the solutions' best fitness values and the two related factors: sum of time consumption (Factor $\mathbb{M}$) and sum of waiting time (Factor $\mathbb{S}$) of all participating vehicles. The fitness value reflects the quality of the solution. The sum of AUVs' waiting time reveals how close to each other the vehicles arrive at the

rendezvous point. A final subset of Monte-Carlo simulation is used to justify the use of optimized operational speed instead of deterministic operational speed for the QPSO-DSSD based path planners. Additional power consumption of the generated trajectories is measured to compare the performance of the path planners.

5.7.1 Effects of Vehicles' Speed and Initial Position

The first study comprised of two sub-studies show the performance of the proposed QPSO-DSSD path planner:

- Monte-Carlo Simulation of vehicle with various water-referenced speeds
- Monte-Carlo Simulation of vehicle with various initial positions

Both simulations here are performed on a 1000-runs basis. In the first set of simulations, the initial positions of the three participating vehicles are set to be at (15.35, 20), (50, 80), (84.64, 20), respectively, and vehicles' water-referenced speed are randomly selected between 1m/s to 3m/s. In the second set of simulations, the water-referenced speed of the three participating vehicles are all set to be at 2m/s and their initial positions are sampled from a uniform distribution with standard deviation at 10 and centred at (15.35, 20), (50, 80), (84.64, 20). Figure 5.15 and Figure 5.16 show the plot of the results obtained by separately applying the four rendezvous schemes of these two simulations.

As shown in Figure 5.15(a), the boxplot of tests with optimized full-scale and mass-centre schemes is centred around 1.8e+006 which is obviously smaller than those of tests with centroid and mass-centre scheme, 2.1e+006 and 2.4e+006, respectively. The boxplot showing the sum of AUVs' waiting time of each run is shown in Figure 5.15(c), it can be seen that optimized mass-centre scheme has the smallest sum of vehicles' waiting time relative to the centroid, mass-centre and optimized full-scale schemes.
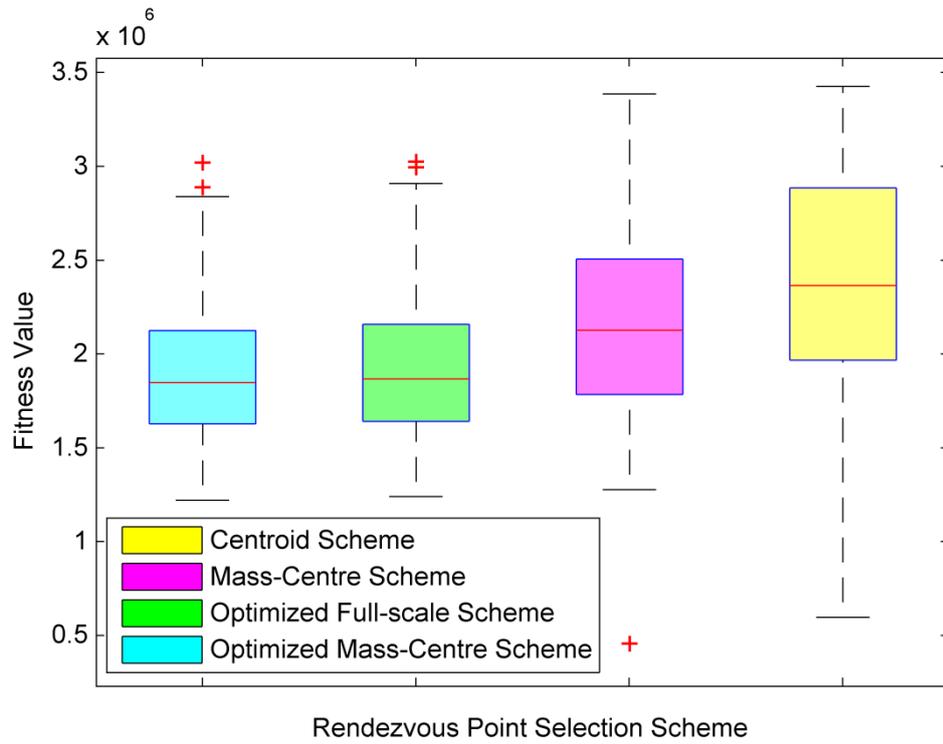
(a)
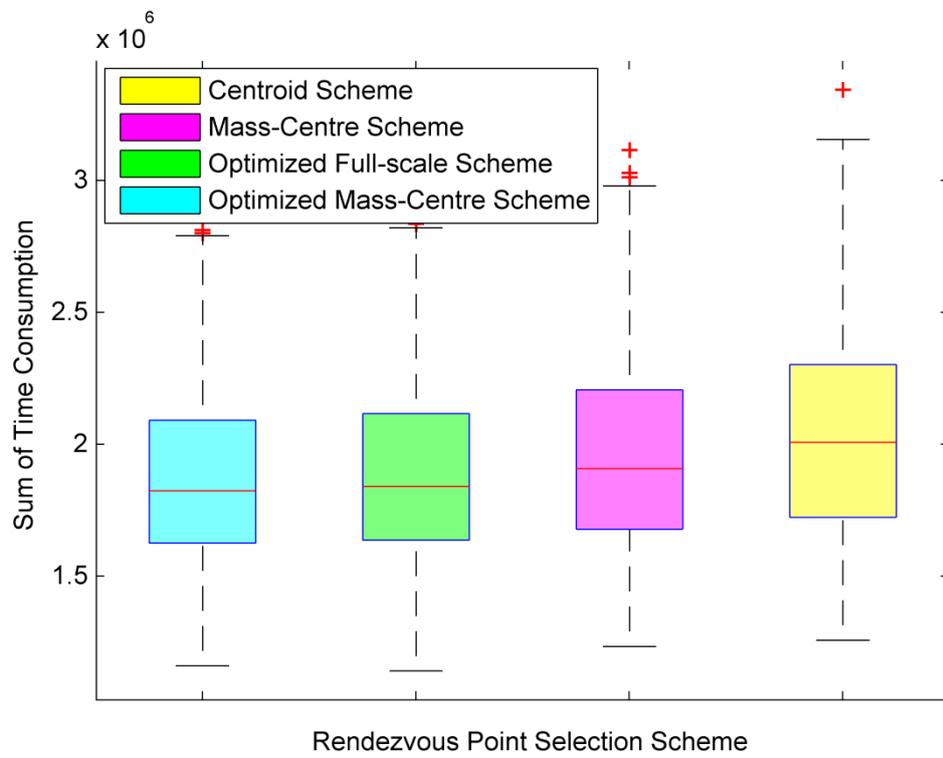
(b)
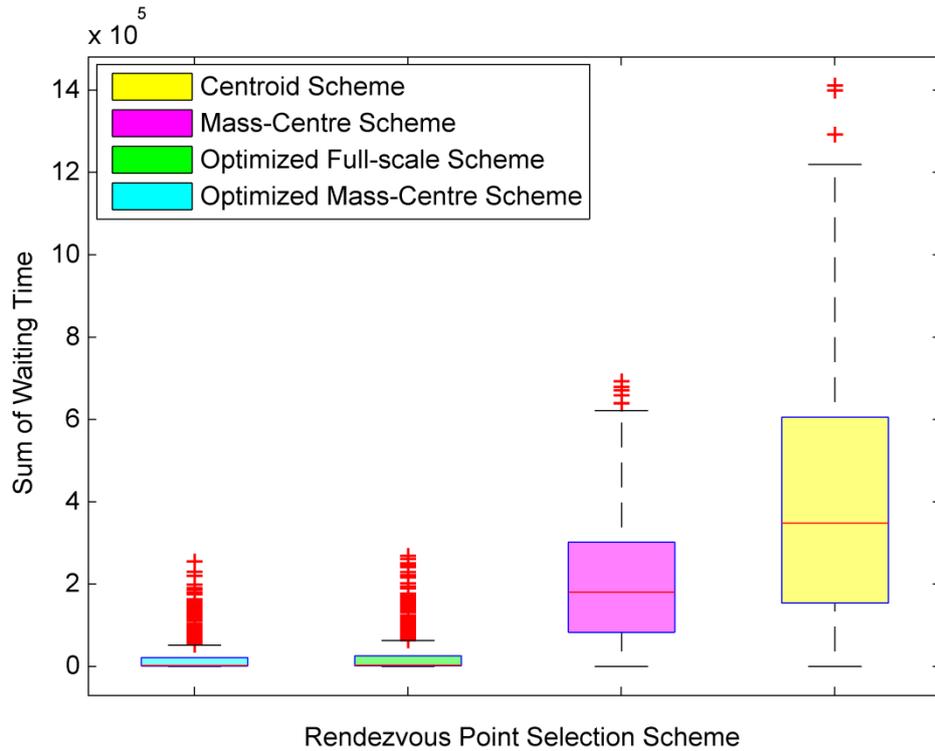
(c)

Figure 5.15 Comparison of results produced by path planners based on Centroid, Mass-centre or Optimized full-scale and Optimized mass-centre rendezvous point selection schemes on Monte Carlo simulation with various vehicles' water-referenced speed.

The same is true for the results of the second set of simulations with various vehicles' initial positions as shown in Figure 5.16. The sum of AUVs' travelling times obtained with all four schemes are similar, but both the two heuristic schemes achieved significantly lower value for the sum of AUV' waiting times compared with the deterministic schemes. Consequently, the fitness values of the paths produced by the heuristic schemes are significantly less than those produced by the deterministic schemes.

(a)

(b)

(c)
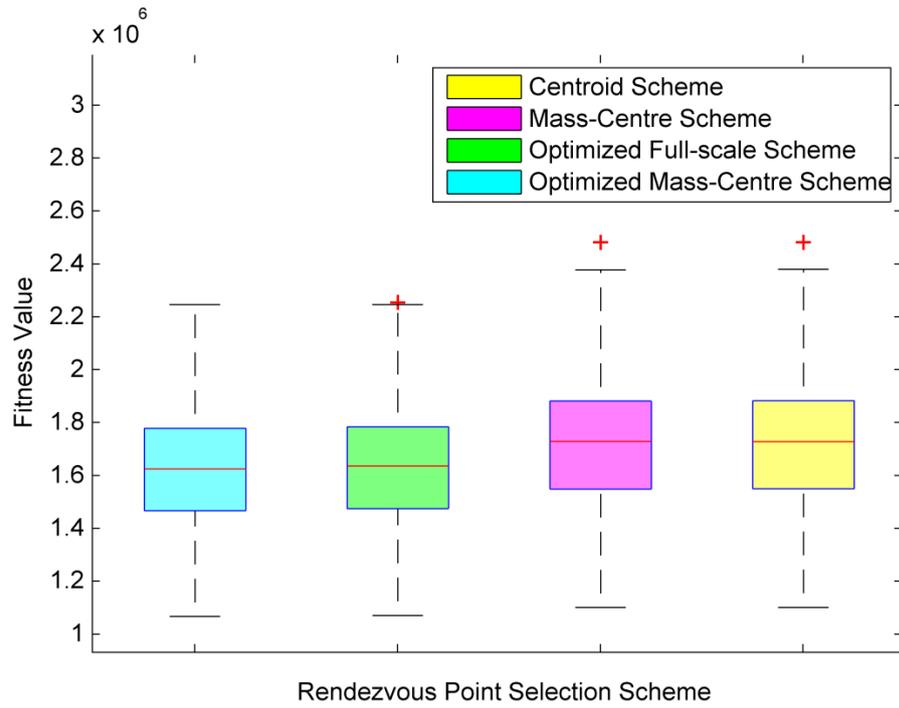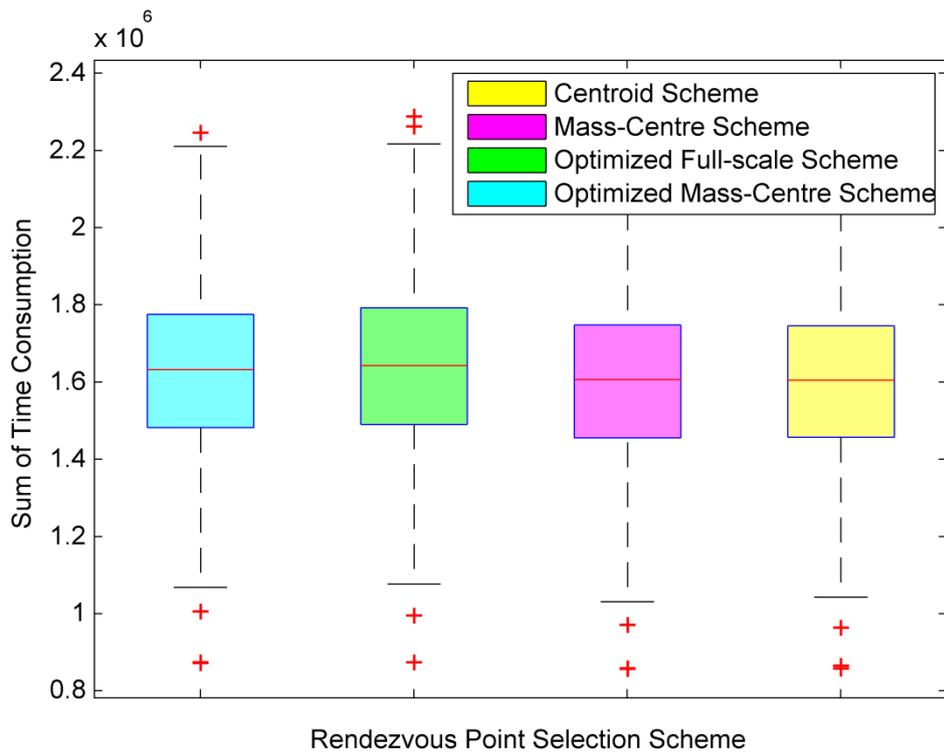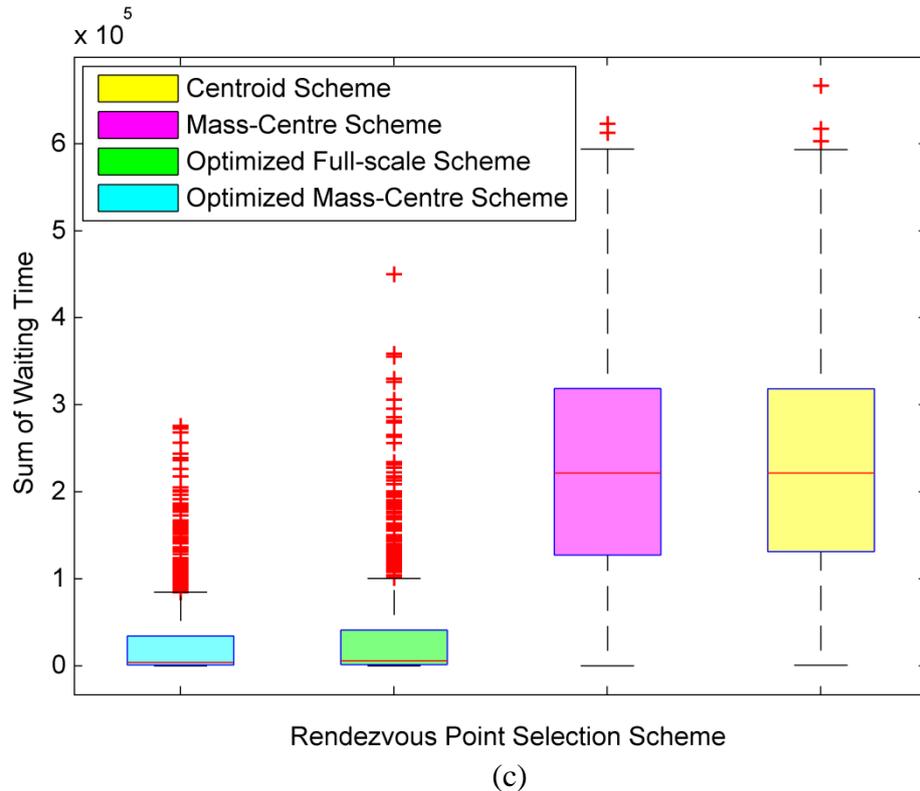
Figure 5.16 Comparison of results produced by path planners based on Centroid, Mass-centre or Optimized full-scale and Optimized mass-centre rendezvous point selection schemes on Monte Carlo simulation with various vehicles' initial positions.

These boxplots illustrate that no matter the vehicles travel at various water-referenced speeds or departure at different locations, the planner with optimized mass-centre scheme inherent superiority of rendezvous performance to those with the other three schemes.

5.7.2 Effects of Population Size

In this second study, ten scenarios, where the population size of particles for optimization is set to be from 200 to 2000 with the step size of 200, were created. For each of the ten scenarios, the simulations are performed on a 100-runs basis. As shown in Figure 5.17(a), the best fitness value of trajectories generated with deterministic schemes: centroid and weighted mass-centre, are at the very top of those generated with heuristic optimized full-scale and mass-centre schemes. Since the proposed optimization is a minimization problem, the smaller the mean value, the better the performance, which means the approach based on heuristic schemes are able to achieve a more optimized trajectory than the other methods. Meanwhile, it is

not hard to find in Figure 5.17(a) that the mean best fitness value of paths produced with deterministic schemes decrease slightly with the increase of population size, and the corresponding sum of AUVs' waiting time (in Figure 5.17(c)) are all inferior to those results obtained with heuristic schemes. In contrast, it should be noted that with more particle population, the planner based on heuristic optimized full-scale and mass-centre schemes achieve significant improvement in the mean best fitness value with stable decrease in sum of AUVs' waiting time.

(a)
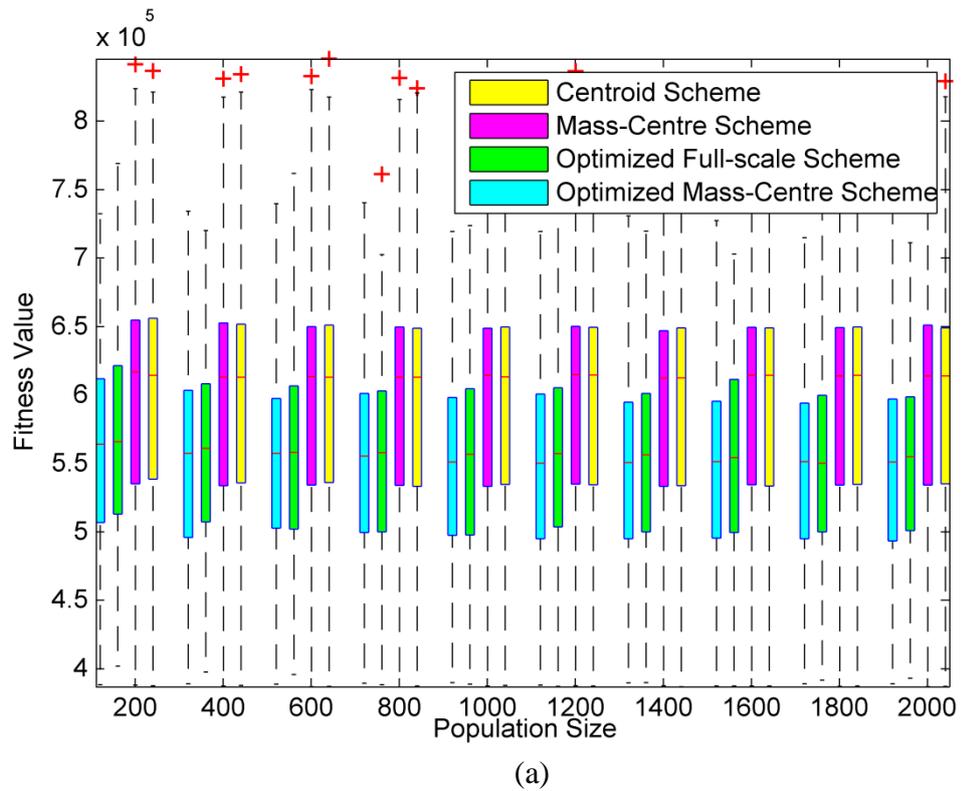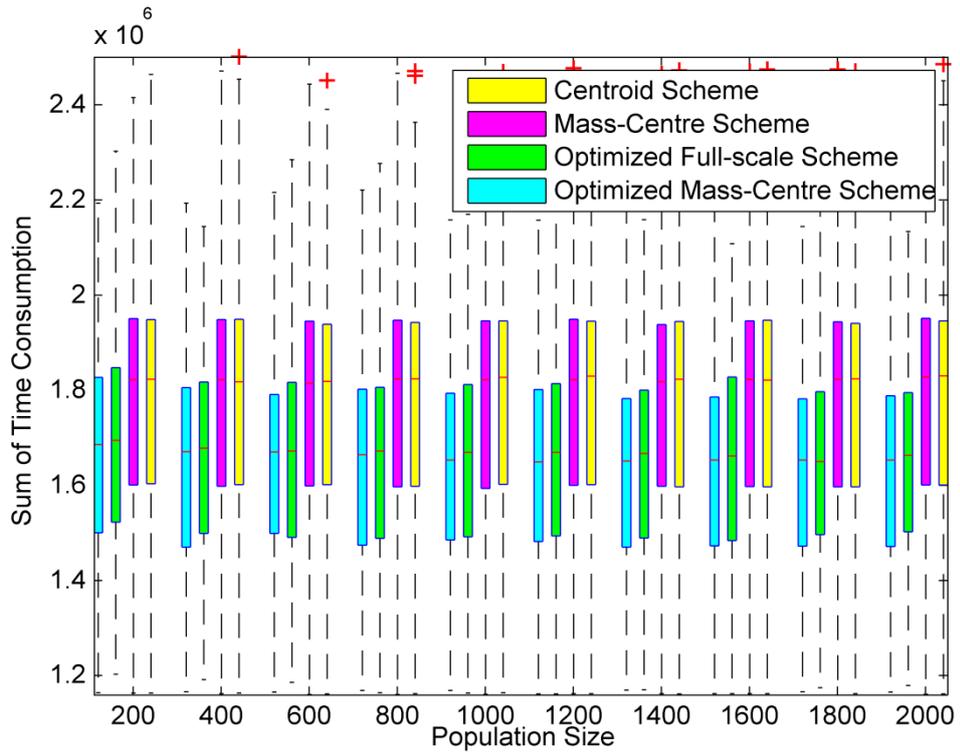
(b)
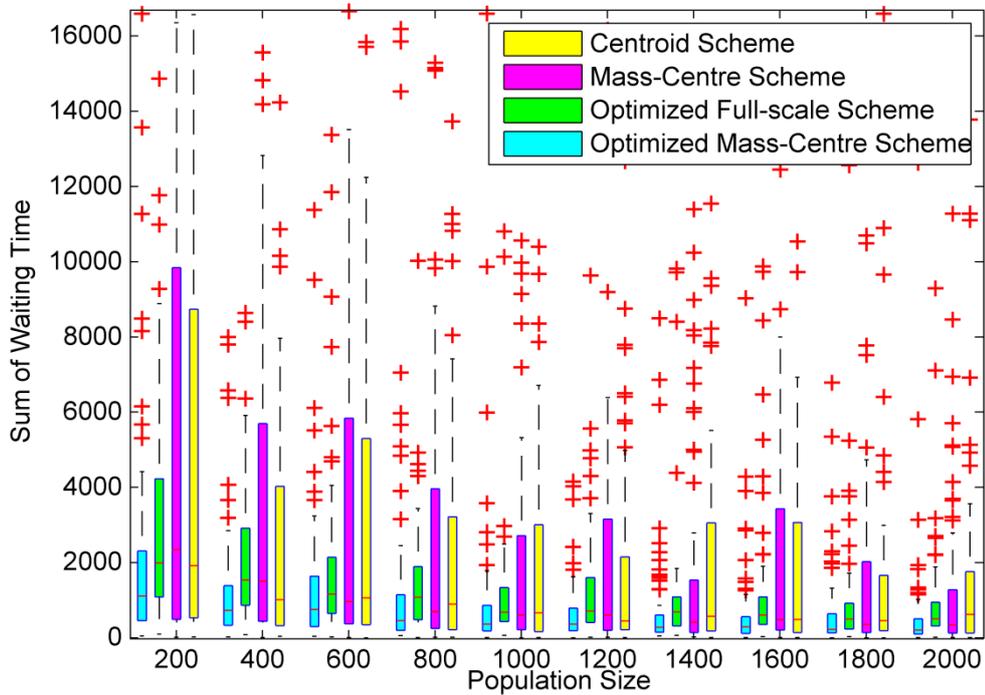
(c)

Figure 5.17 Comparison of results produced by path planners based on Centroid, Mass-centre or Optimized full-scale and Optimized mass-centre rendezvous point selection schemes on Monte Carlo simulation with various particle population size.

Therefore, it is concluded that path planner based on heuristic optimized full-scale and mass-centre schemes have better performance and robustness than the one based on deterministic schemes, and there is a potential to achieve better solution with larger population size. It should be noted, however, that a larger population size implies more computation load. As indicated before, in on-line path planning for multiple AMVs, it is important to keep AUV's time on the surface to a minimum, and thus planning should be complete within limited time available for computation.

5.7.3 Effects of Quantity of vehicles

In this third study, five scenarios, where the quantity of vehicles participating the rendezvous mission is set to increase from 3 to 7, were created. For each of the five scenarios, the simulations are performed on a 200-runs basis. As can be seen in Figure 5.18(a), the boxplots of the results obtained by these four schemes show expanding standard deviation. This is reasonable since, with more vehicles participate the rendezvous mission, more paths will be generated and the fitness values become more and more varied. It should also be noted when there is small number of participating vehicles (three AMVs), the mean best fitness value of paths produced with heuristic optimized full-scale and mass-centre schemes are slightly smaller than that generated with deterministic schemes. When there is more participating vehicles (more than four AMVs), the best fitness value of paths produced with four schemes grows with the increase of number of participating AMVs, and the corresponding standard deviation expands significantly as well. It is worthwhile to note that, as can be found from Figure 5.18(b) and (c), with the increased number of participating vehicles, the sum of AMVs' travelling time consumption for paths produced with heuristic or deterministic schemes have the similar values, however, this sum of AMVs' waiting time for paths produced with deterministic schemes increases significantly faster than that of produced by planner with heuristic optimized full-scale and mass-centre schemes, which indicates that there is more beneficial of using the heuristic optimized full-scale scheme than just relying on the deterministic schemes.

(a)

(b)

(c)
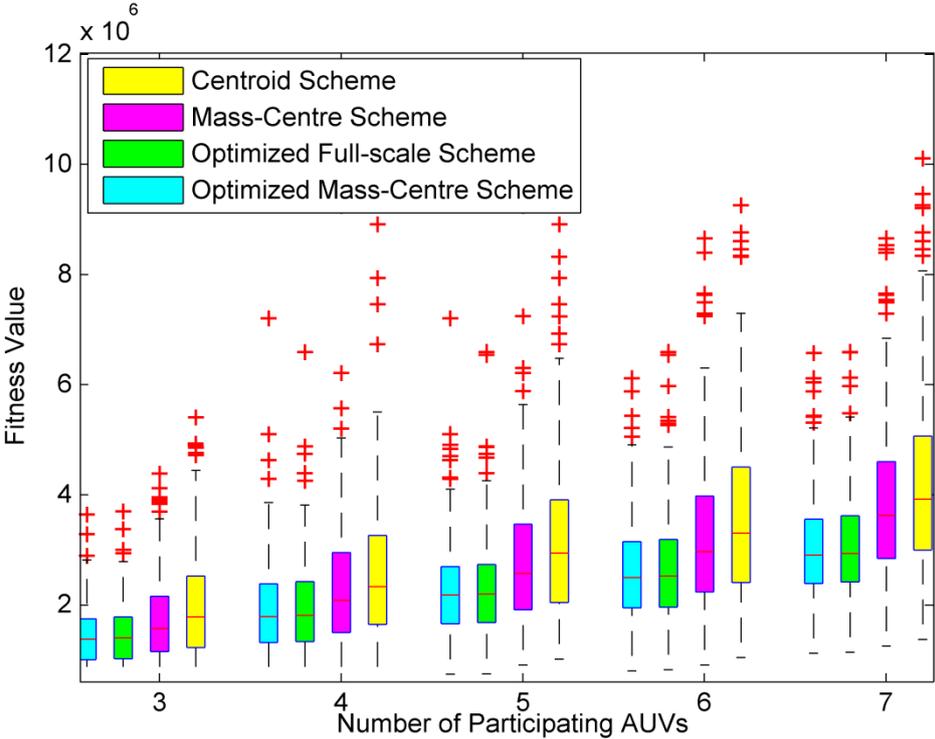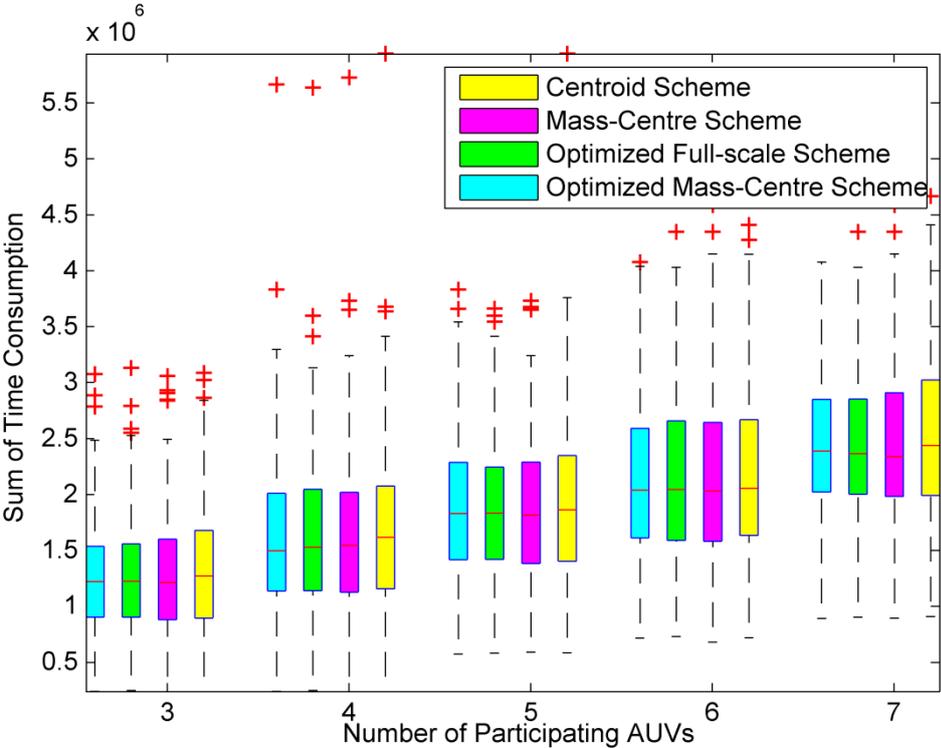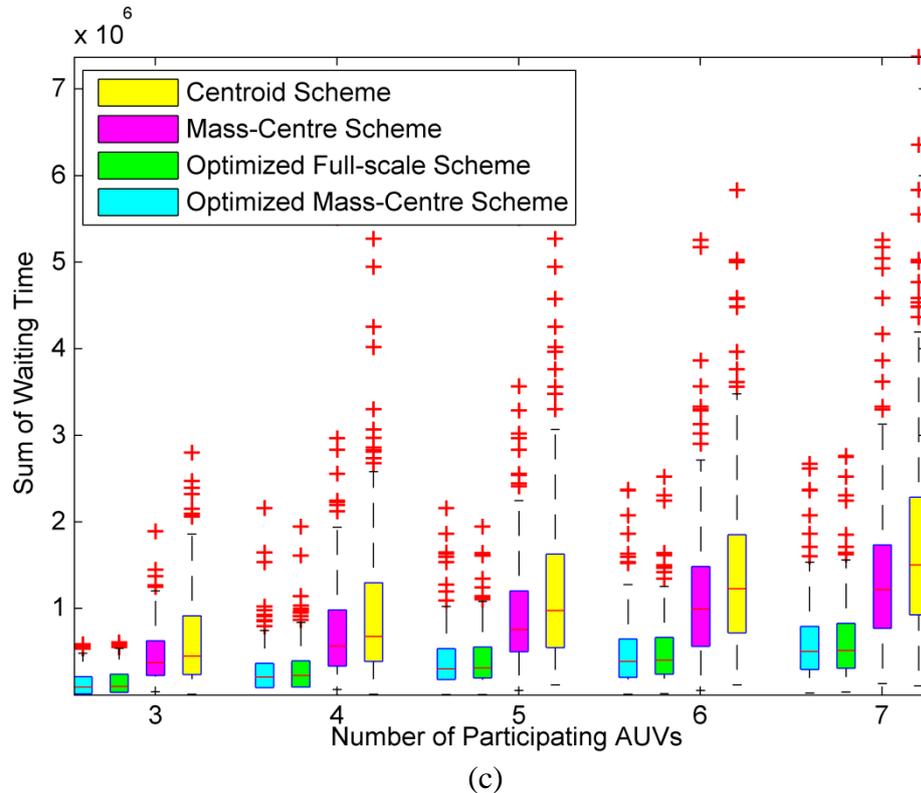
Figure 5.18 Comparison of results produced by path planners based on Centroid, Mass-centre or Optimized full-scale and Optimized mass-centre rendezvous point selection schemes on Monte Carlo simulation with various quantities of participating vehicles.

Therefore, it is concluded that path planner based on heuristic mass-centre scheme has better performance and robustness than the one based on the deterministic schemes, it should also be note that, when the number of participating vehicles gets higher, that path planner based on heuristic mass-centre scheme is more likely to achieve remarkable better solution.

5.7.4 Deterministic and optimized operational speed

The study performed here is to examine the performance of the path planner with both deterministic operational speed and optimized operational speed. The simulation is repeated 1000 times with randomized initial conditions: the x and y locations of the vehicles are sampled (from a uniform distribution with standard deviation at 10) from (15.35, 20), (50, 80), (84.64, 20), respectively, and vehicles' water-referenced speeds are randomly selected between 1m/s to 3m/. With the deterministic scheme, the vehicles' water-referenced velocity is fixed at their

selected speeds. In contrast, the scheme applying optimized operational speed gives path planners the flexibility to find the optimal operational speed. Similar to those case studies in Subsection 0, the operational speed is constrained with the lower bound at 90% of the full speed, and the upper bound is the full speed.

The results are tabulated into two histograms. The first histogram (Figure 5.19) shows the results of AUVs' mission time by following trajectories produced by QPSO-DSSD based path planners with deterministic and optimized operational speed, and the second (Figure 5.20) shows the corresponding power consumption for the AUVs to complete the rendezvous missions.
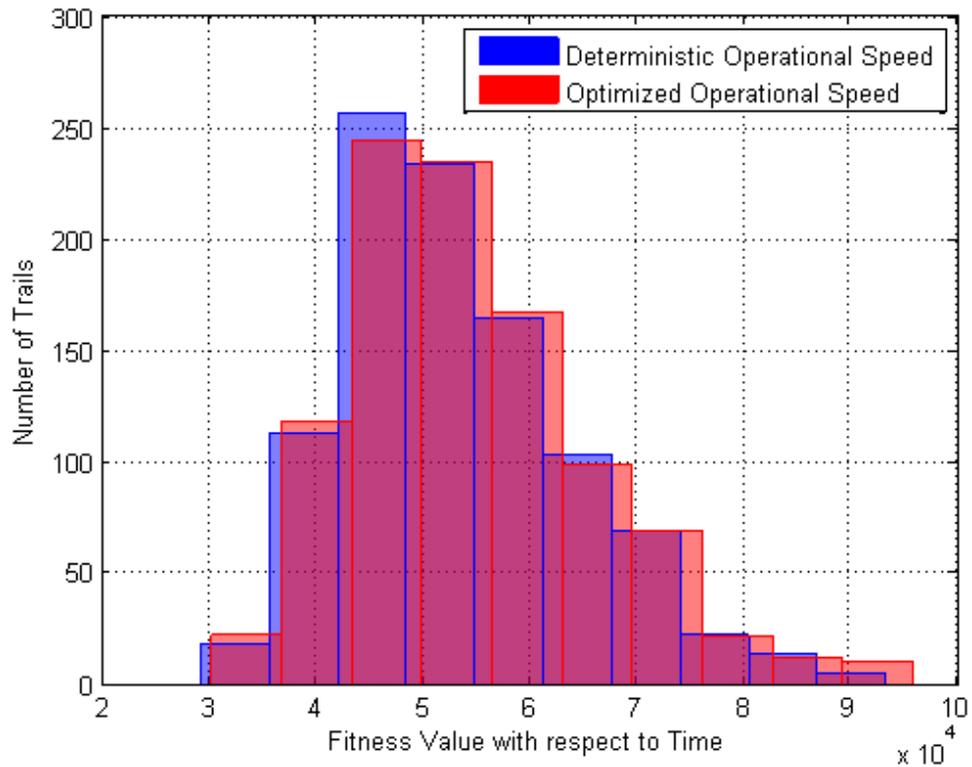
(a)
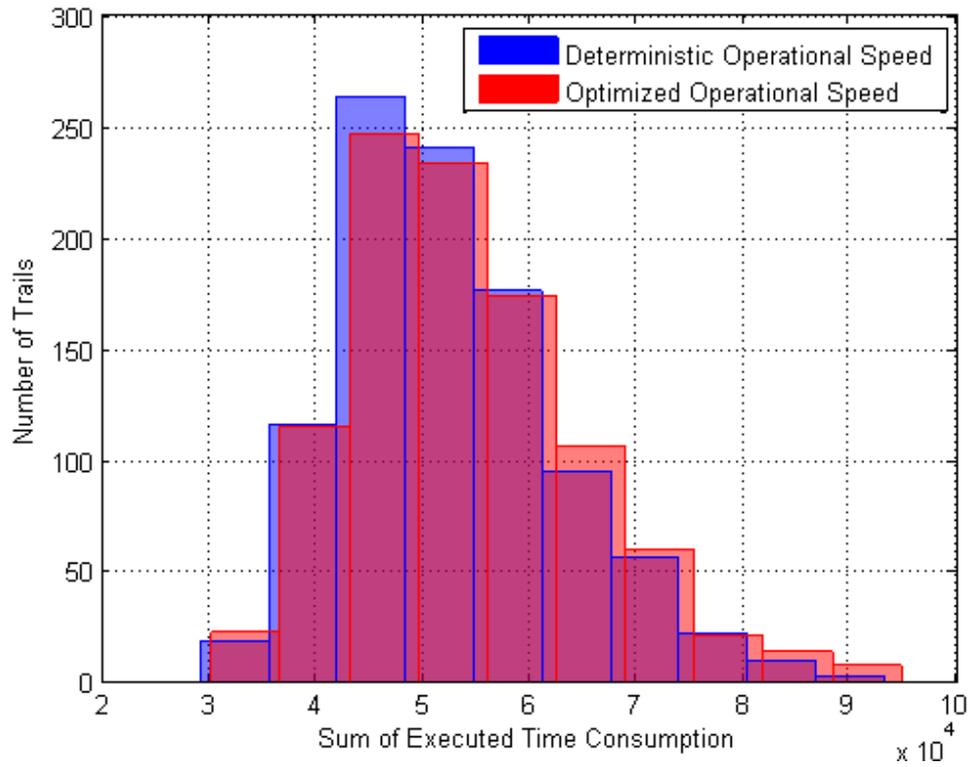
(b)
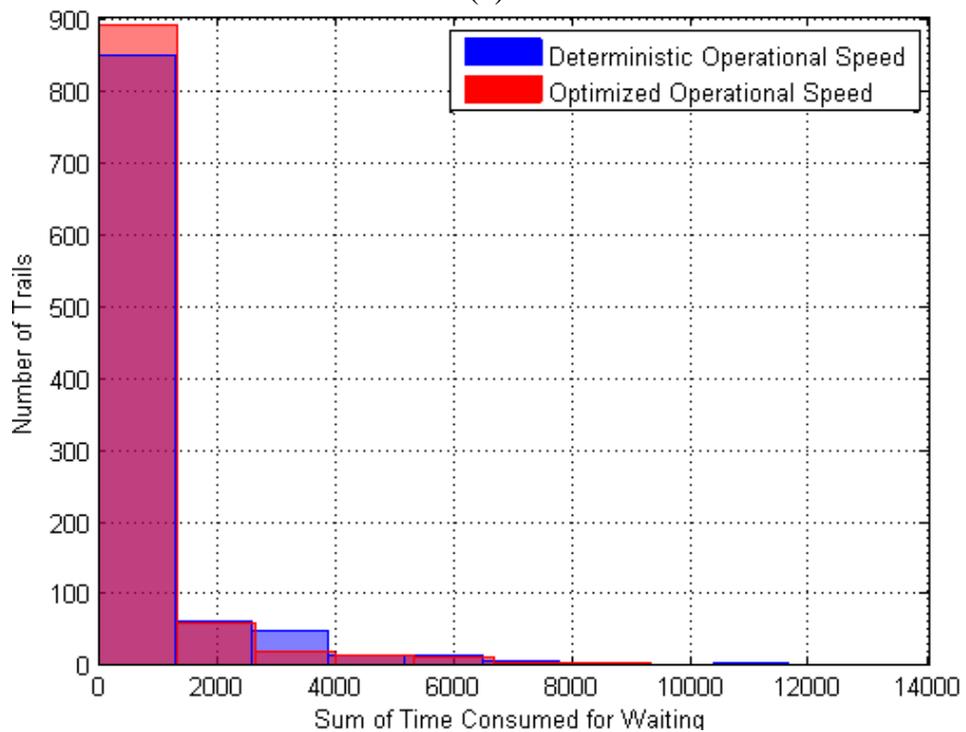
(c)

Figure 5.19 Comparison of AUVs' mission time by following trajectories produced by QPSO-DSSD based path planners with deterministic and optimized operational speed: (a) histogram of the fitness value, (b) histogram of the sum of AUVs' travel time, (c) histogram of the sum of AUVs' waiting time.
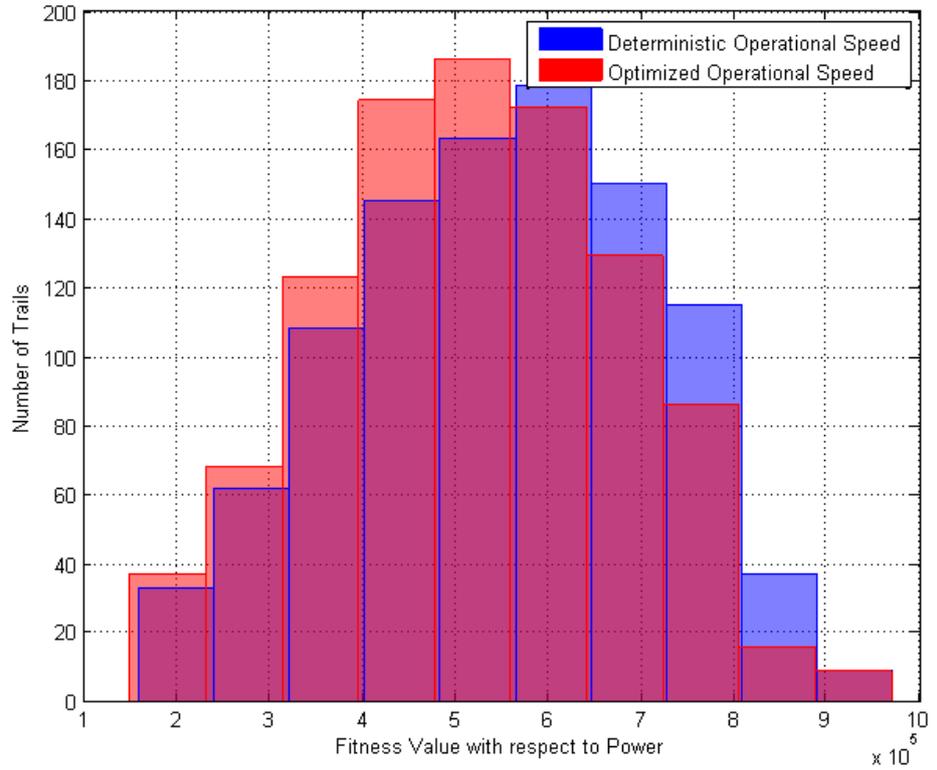
(a)
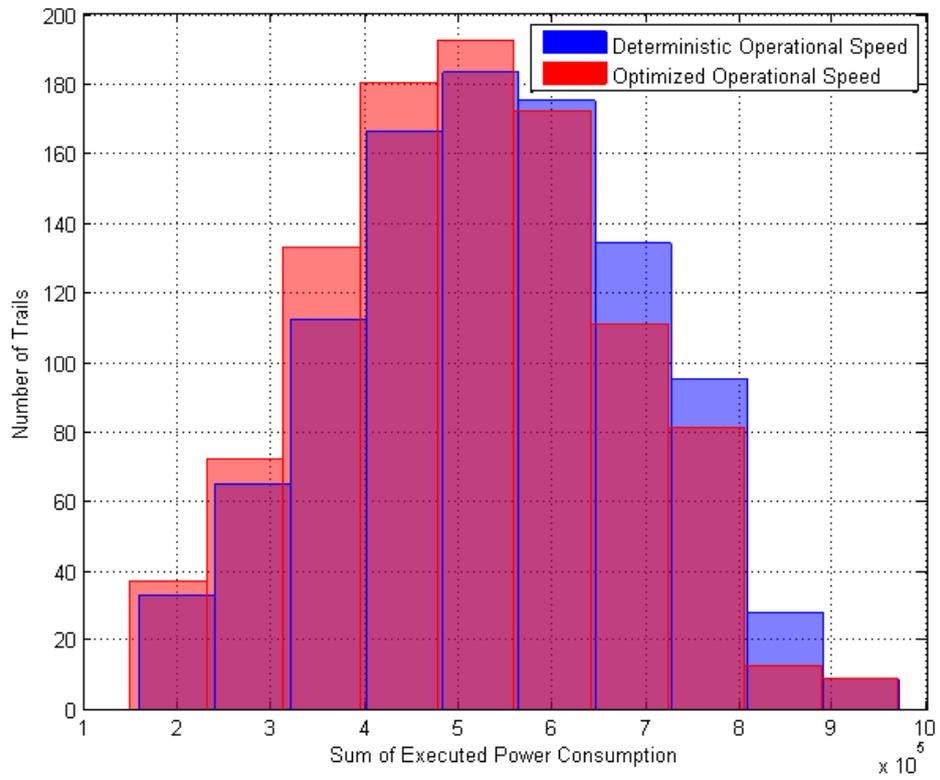
(b)
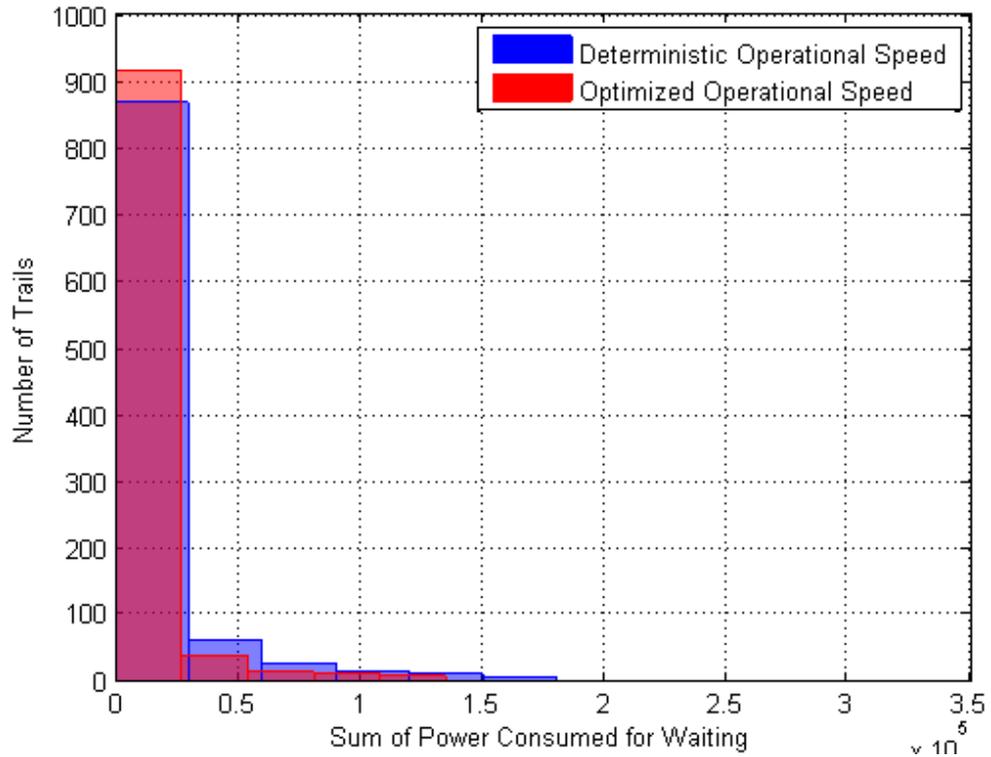
(c)

Figure 5.20 Comparison of AUVs' power consumption by following trajectories produced by QPSO-DSSD based path planners with deterministic and optimized operational speed: (a) histogram of the fitness value with respect to power consumption, (b) histogram of the sum of AUVs' power consumption for travel, (c) histogram of the sum of AUVs' power consumption for waiting.

From the histograms of the fitness value (Figure 5.19(a)) and the sum of AUVs' travel time (Figure 5.19(b)), the QPSO-DSSD based path planners with deterministic operational speed seems more often performs better to find paths for the vehicles to the rendezvous point (from the height of the peak centred around 4.5x10^4) than that with optimized operational speed. Differently, QPSO-DSSD based path planners with optimized operational speed achieves a higher peak for the sum of waiting time, which indicates that this approach shown better performance of simultaneousness for rendezvous.

By comparing AUVs' power consumption by following trajectories produced by QPSO-DSSD based path planners with deterministic and optimized operational speed (Figure 5.20), in fact, the mean of the fitness value with respect to power consumption for QPSO-DSSD based path planners with optimized operational speed is 5.2x10^6, compared to 6.1x10^6 for that with deterministic operational speed. On

average, the QPSO-DSSD based path planner with optimized operational speed obtains paths for vehicle that consumes less power for both travel and waiting.

In conclusion, there is an overall benefit for the path planner to use the optimized operational speed scheme over the deterministic operational speed scheme (as established from the Monte-Carlo simulation), in particular, the benefit is very substantial in save of power consumption.

5.8 Chapter Summary

This chapter presented a new DSSD scheme combine with the Spline based QPSO path planner for multi-vehicle trajectory optimization with the purpose of simultaneous rendezvous. This path planner is also integrated with an optimized mass-centre rendezvous point selection scheme to detect the optimal rendezvous position. Experimental results of rendezvous involving multiple AMVs resolved with the proposed method demonstrates many superiority of this algorithm compare with the centroid, mass-centre or optimized full-scale rendezvous point selection schemes. It is capable of finding more optimized trajectories that not only take maximum advantage of favourable currents, but also lead the vehicles arrive much closer to each other at the selected rendezvous point in time.

In addition, this chapter incorporates the QPSO-DSSD path planner with optimal operational speed. The effectiveness and performance of this novel path planner was compared to the one with deterministic operational speed. Experimental results demonstrated that path planner with optimal operational speed obtain better solution that improves the performance of simultaneousness for vehicles to arrive at the rendezvous point, and most importantly, reduces the power consumption to complete the mission.

Chapter 6

Conclusion and Future Work

This Chapter summarizes the main components of the work accomplished and highlights key findings of this thesis. Following which, directions for possible future work is proposed.

6.1 Summary

This thesis covered a wide spectrum of problems ranging from path planning and replanting problems involving a single AUV to rendezvous problems involving teams of AMVs operating in a dynamic, cluttered, and uncertain ocean environment. Chapter 1 provided a review of the existing AUVs and categorized them into three broad classes based on their operation endurance. Detailed definitions and background relating to the challenge of path planning problem for AUVs were presented in Subsection 1.1.3.

Chapter 2 described the path planning missions and formulated the optimization problem. All aspects of the implemented planner including the model of ocean currents, obstacles and dynamic rendezvous target, cost functions and B-Spline path formation were described in detail in Section 2.3. This Chapter also reviewed and evaluated existing approaches to the problem of path planning for an AUV. In particular, as presented in Section 2.4, path planners based on A*, RRT, GA, PSO and QPSO algorithms are developed and their performance are compared through several scenarios. Simulation results show that the proposed QPSO approach is able to obtain a more optimized trajectory than the A*, RRT, GA or PSO based method.

Chapter 3 presented and examined a novel SSD algorithm. The SSD scheme has been integrated with the B-Spline based QPSO path planner in Chapter 2 and tested to find an optimal trajectory for an AUV navigating through a variable ocean environment in the presence of obstacles whose position coordinates are uncertain. Simulation results show that the proposed SSD approach is able to obtain a more optimized trajectory than the concentric circle (2D)/spherical surface (3D) constrained method. The SSD method also achieves faster convergence speed and uses less computation time than the unconstrained full space searching method. Monte Carlo trials were run to assess the robustness of the SSD method, the results demonstrate the inherent superiority of the proposed SSD method.

Chapter 4 first illustrated the concept of on-line path planning system that adapt and regenerate the trajectory during the course of the mission using continuously updated current profiles from on-board sensors. This chapter then extended the general SSD scheme in Chapter 3, and developed a dynamic SSD scheme for the AUV on-line planning system. This system combines path re-planning with B-spline based QPSO technique, and tested to generate an optimal trajectory for an AUV navigating through a spatiotemporal ocean environment in the presence of irregularly shaped terrains as well as obstacles whose position are uncertain. Simulation results show that with same amount of computational load, path planner based on the proposed dynamic SSD scheme utilizes path re-planning methodology is able to obtain a more optimized trajectory than one relying reactive path planning.

After having discussed path planning and re-planning problems involving a single AUV, subsequently, Chapter 5 addressed problems involving multiple spatially distributed AMVs. In particular, a rendezvous problem was formulated for the path planning system to find trajectories for multiple AMVs with minimal time usage over all participating vehicles and simultaneous arrival of the vehicles at their selected rendezvous destination. To deal with multi-AMV case, this Chapter first extended the SSD scheme in Chapter 3 to DSSD scheme. Then, an Optimized Mass-centre rendezvous point selection scheme was developed to detect the optimal rendezvous position, as well as a module to detect the optimal operational speed for each vehicle. Several examples of multi-AMV rendezvous resolved by the proposed method are presented. This novel QPSO-DSSD path planner is proven to be capable

of identifying the optimal rendezvous location and generating the corresponding rendezvous trajectories based on the capabilities of each vehicle and existing dynamic ocean environment.

6.2 Conclusions

The conclusions can be drawn based on the proposed SSD scheme for B-Spline based evolutionary path planner and associated planning and re-planning implementation on the AUVs. The main conclusions are summarised below.

6.2.1. B-Spline based QPSO path planner (Chapter 2)

The B-Spline based QPSO path planner is developed for solving the optimal path planning problem for AUV operating in turbulent, cluttered and uncertain environments. The benefit of this proposed path planner over state-of-the-art techniques was demonstrated through several simple example tests in a static environment: obtained better smooth curvature paths represented by the Spline to accommodate constraints imposed by the manoeuvrability of the vehicle; and the QPSO algorithm allows to find a more optimized trajectory than the A*, RRT, GA and PSO based method.

The proposed B-spline based evolutionary path planner framework is suitable for solving constrained nonlinear optimization problem. Currently, graph search schemes, such as A*, are more often used as a solver for path planning problems. They generate reasonable good solutions efficiently and consistently in the simple scenarios. The B-spline based evolutionary path planner was found to have the best optimality and robustness with the consequence of increased computational load. However, in modern AUVs this increased load is manageable and the extra, easier of implementation in many-dimensional problems and high adaptability to the special characteristics of the considered problem more than compensates for this limitation.

6.2.2. Shell space decomposition for efficient path optimization (Chapter 3)

A novel SSD scheme was developed to increase the searching efficiency of the B-Spline based QPSO path planner. The proposed SSD approach was testified to be

able to obtain a more optimized trajectory than the circle/sphere constrained methods and achieve faster convergence speed than the full space searching method. The results of Monte Carlo trials demonstrate the inherent superiority of the proposed SSD method.

6.2.3. Dynamic shell space decomposition for on-line path re-planning (Chapter 4)

As an extension of the basic SSD scheme, a dynamic SSD strategy is developed and combined with an efficient re-planning scheme which preserves some information from previous solutions and uses it as the basis to compute new solutions. The proposed on-line path planning system is able to continuously adjust to environmental changes, and able to generate an optimal trajectory using a short planning horizon. The robust stability of the path re-planning scheme proven by showing that the candidate solution gives a cost that is always better than that generated with reactive path planning scheme under various scenarios.

6.2.4. Distributed Shell Space Decomposition for Rendezvous Path Planning of Multiple AMVs (Chapter 5)

A DSSD scheme is developed and integrated into the B-Spline based QPSO path planner for rendezvous of AMVs in variable ocean environments. Simulation results demonstrate the benefit of the QPSO-DSSD approach over standard approaches in the cooperative path planning scenarios. This proposed approach is proven to be capable of finding more optimized trajectories that not only take maximum advantage of favourable currents, but also lead the vehicles arrive closer to each other at the selected rendezvous point in time.

It should be noted that the solutions found by the QPSO-DSSD approaches cannot achieve the vehicles to arrive at the rendezvous point exactly at the same time. However, the time vehicles spend on waiting for the other vehicles to arrive are negligible compared with the time vehicles spend on travelling (normally less than 1%). It is possible, when needed, to generate the trajectories accommodating precisely to the time of the last arrived vehicle, however these often require significant computation capabilities, making on-line path planning more difficult. This is especially unacceptable in on-line path planning for multiple AMVs. It is

important to keep AUV's time on the surface to a minimum to minimize their risk of collisions with ship and boat traffic and being damaged by strong winds and heavy waves, and thus planning should be complete within limited time available for computation. The approaches proposed in this thesis provide a suitable trade-off between tractability and optimality.

6.3 Future Research Directions

Building on the contributions listed above, suggestions for the future research directions are outlined below.

6.3.1. SSD combine with other optimization algorithms

The SSD schemes that are introduced can be considered independent of the underlying optimization method and as such can be generally used with path optimization techniques other than QPSO as proposed in Chapter 2. The Imperialist Competitive Algorithm (ICA) is a new evolutionary algorithm first proposed by Atashpaz-Gargari and Lucas [171] for solving continuous optimization problems. ICA has been successfully utilized to solve optimization problems in many engineering applications such as control [172], power dispatch [173], production management [174] and has shown high performance in both global searching ability and convergence rate. It would be valuable to put together different recent evolutionary algorithms into one framework and compare the approaches in terms of performance, computation complexity, and the performance improvement with integration of the proposed SSD scheme.

6.3.2. Dynamic SSD with adapting opening angle

The opening angle is closely relating to the accuracy of the path and computational efficiency, greater value of opening angle results in more computational time and slower convergence speed. On the other hand, the larger the value of opening angle, the larger searching space will be covered, and the more accurate the path will be.

An inertia weight can be employed to decrease the opening angle for re-planning when the vehicle getting close to the target, thereby influencing the trade-off between global (long rang) and local (nearby) exploration abilities of planner. A larger inertia

weight facilitates global exploration (searching large areas) while a smaller inertia weight tends to facilitate local exploration to fine-tune the current search area as the vehicle gets close to the target. Suitable selection of the inertia weight can provide a balance between global and local exploration abilities and thus require less iteration on average to find the optimal solution.

Figure 6.1 sheds some light on how this scheme can be implemented. As can be seen, during the dynamic planning process, the opening angle is set at three different values. The annular regions allow for searching is large with the opening angle at 120° for the first planning, but when the AUV gets closer to the rendezvous target, these searching regions decrease their sizes with the opening angle set at 90° and then 60° in the following planning.
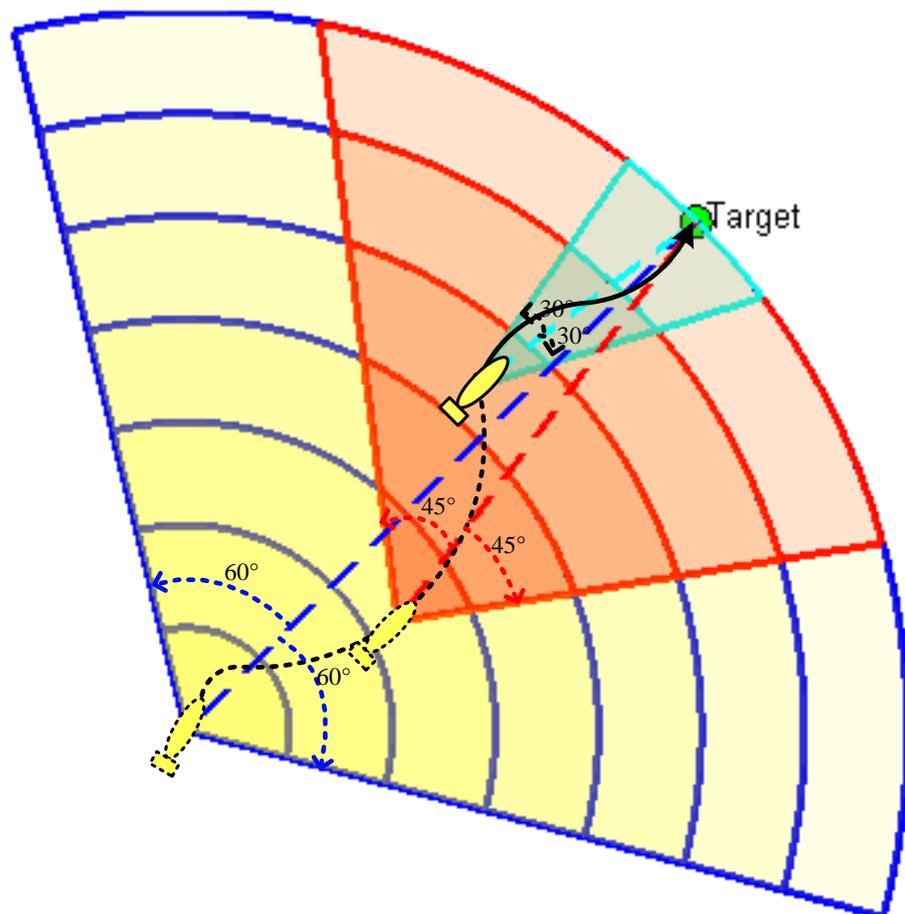
Figure 6.1 Varying the opening angle for subsequent planning

6.3.3. Optimal departure time for multi-vehicle rendezvous

One limitation of the state of the art approaches for multi-vehicles rendezvous

mission is that they assume the departure time is fixed in advance [175]. However, in many applications, the departure time can be flexible. Indeed the departure time is required to know the state of the currents, which is necessary to evaluate the travel cost of the vehicle. In such situations, the choice of an appropriate departure time is a critical issue. It should avoid starting a maritime mission near the coast in the presence of strong adverse tides. On top of the detection of the optimal rendezvous point and vehicles' operational speed proposed in Chapter 5, it would be valuable also involve one extra optimization module to optimize the departure time for each vehicle. The possible design would be allowing the departure time for every vehicle some flexible in a given time window, and the choice of an appropriate departure time will be optimized based on the global objective function.

6.3.4. Test SSD based path planner through hardware experiments

Most of the results presented in this work have been run in simulation. The next step should consist on running the proposed techniques on a real AUV operating in the ocean, and command it to follow the trajectory found by the path planner. The effective of the proposed techniques can be evaluate this way, as well as the accuracy of the AUV motion model run by the simulator and the ocean currents provided by the Bluelink. Moreover, it is interesting to do more tests for more practical problems, like dynamic path planning and multi-vehicle missions. In the particular case of dynamic path planning, realize recovering an AUV into a moving dock with aware of the variable currents would be an interesting topic of future work. In fact, the path re-planning system shown in Chapter 4 will efficient regenerate the trajectory and continuously adjust trajectory response to the moving dock and changing currents. For multi-vehicle missions, rendezvous an AUV with an unmanned surface vehicle (USV) would be a good start. The path planner shown in Chapter 5 will be tested to do the path planning required for rendezvous, and it is interesting to see how simultaneous will the vehicles arrive at the rendezvous point.

6.3.5. Application of the proposed techniques for other field vehicles

The SSD based evolutionary path planners would fit within the broader field of path planning for all field vehicles. Among the various types of field vehicles, unmanned

ground vehicles (UGVs) usually facing cluttered and uncertain urban environment that the planning methods need to be both robust to uncertainties as well as computationally feasible; Unmanned air vehicles (UAV) form another special class and are particularly interesting for operations in dynamic environments. Like the ocean, the atmospheric environment is characterized by its continuous dynamics. The magnitude and direction of the wind field significantly affects the on-board power of UAVs. Thus, optimal path planning techniques considering variable, uncertain and cluttered environmental conditions would also be suitable for UGVs and UAVs. To examine the proposed path planning techniques on these field vehicles for enabling persistent operation would be another interesting topic of future work.

6.4 Concluding Remarks

This thesis presents several path planning and re-planning techniques, mainly the Spline-based QPSO Path Planner with SSD, dynamic SSD and DSSD schemes, for applications involving either a single vehicle or teams of vehicles operating in a dynamic, cluttered, and uncertain ocean environment. Results of simulation and Monte Carlo trials have allowed critical evaluation of the proposed techniques to AUV path planning over a variety of scenarios. Future work should be interest to verify the proposed techniques on marine vehicles operating in the real ocean scenarios. From this thesis, it is evident that the proposed techniques have advantages over the existing techniques in both, stability and capability. There are many opportunities and research to be done in the field before robust, persistent, and full-autonomy AUV operation becomes a reality.

Appendix A

Multi-resolution Decomposition Methods for Graph Search Schemes

Multi-resolution space decomposition initially comes with the idea that it is not necessary to represent the searching space in a uniform way. Some regions may be of more interest than others from a trajectory planning point of view. In this Appendix A, a review of the multi-resolution decomposition methods for graph search schemes is introduced.

Multi-resolution Decomposition involves the representation of the graph space with different levels of accuracy. Quadtree method [128, 176] (see Figure A.1) and triangular space decompositions [33] (see Figure A.2), are the most extensively used multi-resolution space decomposition technique. These methods generate a planar cell decomposition consisting of small cell sizes that accurately capture obstacle boundaries and larger cell sizes that efficiently represent large areas in the free space.

(a)   (b)
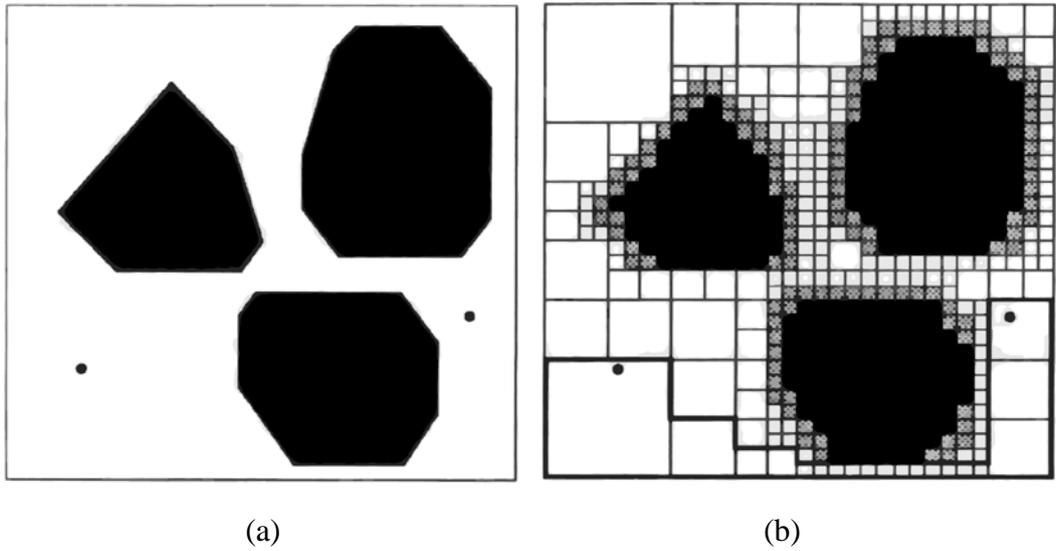
Figure A.1 Quadtree: an approximate cell decomposition method (figure adapted from [176]).

(a)   (b)
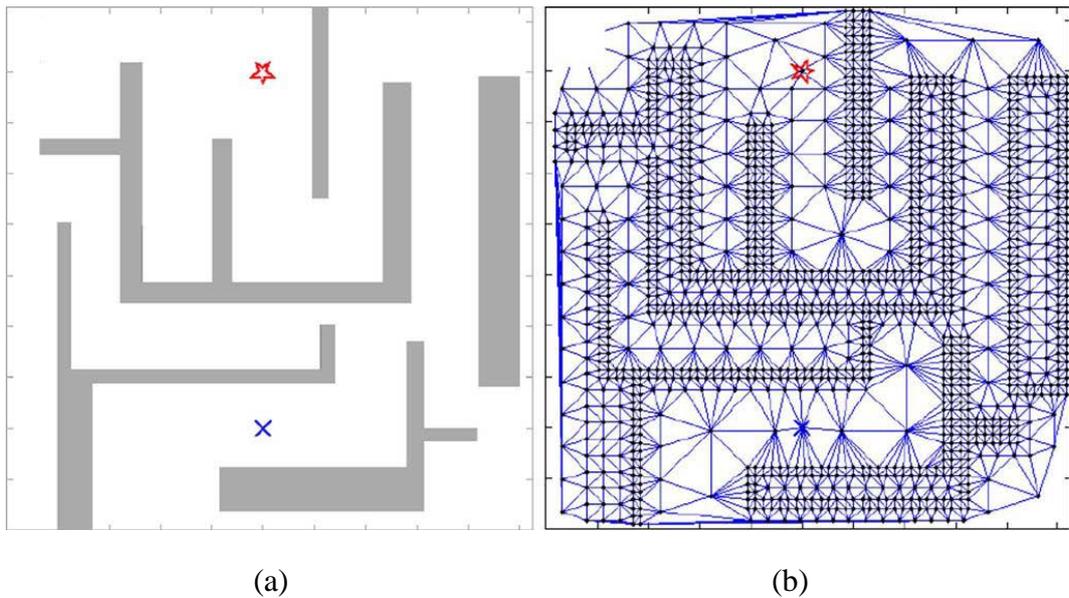
Figure A.2 Triangular cell Quadtree decomposition (figure from [33]).

[177] proposed another multi-resolution decomposition method that the environment is represented with higher accuracy in the vehicle's immediate vicinity, and with lower accuracy in regions farther away (See Figure A.3). The path planner based on this method, taking better care of neighbouring grids, potentially is able to find a high accuracy initial path and re-plan continuously as the vehicle moves at reduced computation costs. Multi-resolution space decompositions are able to compact large environment maps, and thus enable efficient online path planning. Furthermore, this

method decompositions the environment naturally in such way that the uncertainty or incomplete knowledge about the environment is higher in regions farther away from the vehicle's current location.

| (a) | (b) |
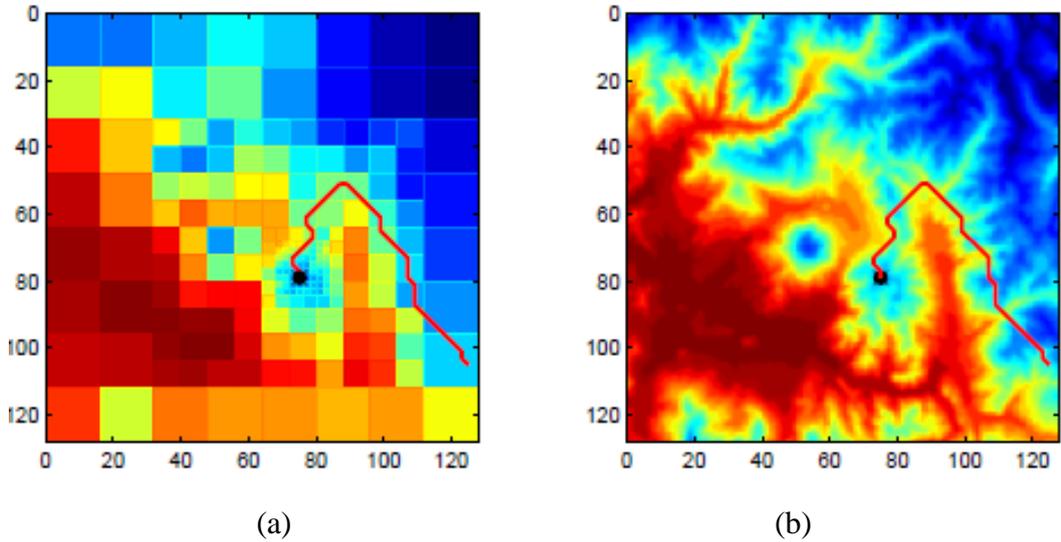
Figure A.3 Multi-resolution cell decompositions (figure adapted from [177]).

Apart from multi-resolution method applied on Cartesian occupancy grid maps, multi-resolution decomposition of log-polar grids can be found for vehicle path planning in the literature [178, 179]. Like the local multi-resolution grid, this approach emphasizes a more precise path planning in the robot's vicinity as shown in Figure A.4. Furthermore, polar grids have the advantage of an easy integration of obstacles perceived by ultrasonic sensors and cameras.

| (a) | (b) |
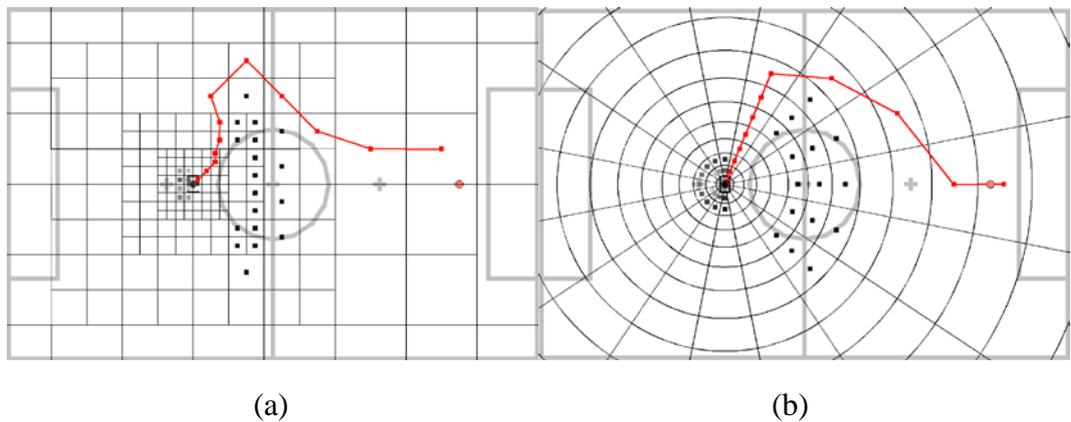
Figure A.4 Guidance system and the AUV simulator (figure adapted from [179])

Appendix B

A Numerical Comparison of Evolutionary Algorithms

To compare the performance of GA, PSO and QPSO algorithms for continuous function optimization, six well-known benchmark functions are selected for testing, and a total of 50 runs are performed on each test function. The function name, analytic expression, and variable domain for each function are listed in Table B.1. The global minimum for each function is 0. The Sphere, Quadric, and Rosenbrock are three unimodal functions, and the rest functions are multimodal.

The population size m is set to 20. A total of 100 runs are performed on each test function. The initial positions of particles are randomly generated from the uniform distribution on the intervals given in Table B.1. Following the suggestion of Sun *et al.* [49], [50], the contraction-expansion coefficient is decreased linearly from 1.0 to 0.5. Roulette wheel selection, heuristic crossover, and uniform mutation are used as genetic operators. The selection, crossover and mutation probability for the GA [13], [14] here are 0.1, 0.8 and 0.1, respectively. Following the suggestion of Shi and Eberhart [27], the inertial weight coefficient w was varied linearly from 0.9 to 0.4, the acceleration coefficients c_1 and c_2 are set to be 2, and the v_{max} used in PSO was set to x_{max}. The above parameters values are selected based on the suggestions in other literatures [13], [14], [16], [27], [49], [50] where these values have been found, empirically, to provide good performance.

The performance of GA, PSO and QPSO algorithms are based on the following three

factors: solution quality, stability, and convergence speed of each algorithm. The solution quality can be expressed by the mean value of best fitness, also accounting for the search ability of an algorithm. Therefore, for minimization problem, the smaller the mean value of best fitness, the higher the quality of solution, and the stronger the search ability. Similarly, the stability of an algorithm can be reflected by the standard deviation. The larger the standard deviation value, the worse the stability of an algorithm. The convergence speed of an algorithm is defined as how fast the algorithm can converge to the optimal or sub-optimal solution. Hence, under the given maximum number of iterations, the smaller the number of iterations needed to ensure that the algorithm converges to the optimal or sub-optimal solution, the faster the convergence speed.

Table B.1 Benchmark Functions

Functions	Formulations	Interval
Ackley	$f_1(X_i) = 20 + e - 20\,exp\left(-0.2\sqrt{\dfrac{\sum_{i=1}^{n} x_i^2}{n}}\right)$ $- exp\left(-0.2\sqrt{\dfrac{\sum_{i=1}^{n} \cos 2\pi x_i}{n}}\right)$	[-30,30]
Quadric	$f_2(X_i) = \sum_{i=1}^{n}\left(\sum_{j=1}^{d} x_{ij}^2\right)$	[-100,100]
Spherical	$f_3(X_i) = \sum_{i=1}^{n} x_{ij}^2$	[-5.12,5.12]
Griewank	$f_4(X_i) = 1 + \dfrac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\dfrac{x_i}{\sqrt{i}}\right)$	[-1,1]
Rosenbrock	$f_5(X_i) = 10n + \sum_{i=1}^{n}[x_i^2 + 10\cos(2\pi x_i)]$	[-5.12,5.12]
WeightedSphere	$f_3(X_i) = \sum_{i=1}^{n} ix_{ij}^2$	[-5.12,5.12]

Appendix B. A Numerical Comparison of Evolutionary Algorithms

The average best fitness values (*Mean*) of the 100 trials along with its standard deviation (*Std*) for each combination of function and algorithm are shown in Table B.2. From the results, it is observed that QPSO can find the global minimum of these benchmark functions with significantly lower standard deviation. This means that the global minimum of these functions can be reached by QPSO with good stability. The average best fitness value produced by QPSO on Spherical and WeightedSphere drops down by 15 and 13 orders of magnitude compared with that resulted by the PSO and GA, respectively.

Table B.2 Performance comparison of GA, PSO and QPSO Algorithms on the benchmark functions

Algorithm	QPSO		PSO		GA	
	Mean	*Std*	*Mean*	*Std*	*Mean*	*Std*
Ackley	1.67E-10	3.90E-10	2.47E-04	2.71E-04	1.43E-01	7.04E-01
Quadric	4.67E-18	4.57E-17	6.38E-09	2.10E-08	1.76E-07	1.73E-06
Spherical	3.88E-22	2.37E-21	1.30E-07	1.91E-07	1.11E-09	1.02E-09
Griewangk	2.14E-09	6.59E-09	1.56E-09	3.06E-09	2.27E-02	5.34E-02
Rastrigin	2.76E-10	1.86E-09	1.64E-05	4.32E-05	9.95E-03	9.95E-02
Weighted Sphere	1.14E-22	3.45E-22	2.29E-07	8.68E-07	1.29E-09	1.08E-09

Appendix C

AUV Simulator

In the absence of a physical platform to evaluate the performance of path planning system, instead an AUV simulator is used to test the trajectories generated by the proposed path planners. This simulation model was built in MATLAB Simulink as this has the capability to allow modelling, simulation, and analysis of dynamic systems. An overall Simulink block diagram of the AUV simulator is shown in Figure C.1.

This simulator contains two principal parts. The Dynamics modules are described in Section C.1. The Guidance, Navigation and Control (GNC) modules are described in C.2.

- A dynamic model of the vehicle incorporating;
    - Hydrodynamics of the vehicle, and
    - Kinematics of the vehicle.
- GNC loop incorporating the following sub-systems;
    - Guidance,
    - Control,
    - Control Allocation.

C.1 Vehicle Dynamics

The vehicle dynamical model was developed by implementing the generic dynamic equations [133] which was parameterized to model the dynamic behaviour of a

torpedo shaped AUV. The parametric vehicle model of the REMUS 100 [180], was chosen to define the vehicle simulation model.

C.2 GNC System

The GNC system includes a line-of-sight (LOS) guidance module, a fully-coupled sliding mode control (SMC) and a navigation module. The simulink block diagram of the GNC system is shown in Figure C.2. The path planner passes the generated trajectory (in the form of discretized curve points) to the LOS guidance system of the simulator, and then the SMC controls the vehicle to follow this desired trajectory. The output is the dynamic state of the vehicle executing this trajectory.

C.2.1 Guidance System

The desired raster scan trajectory used for the mission was defined a priori, in an open loop sense, by the operator ensuring any manoeuvres, due to the requested trajectory, are within the manoeuvring capabilities of the AUV regardless of operating conditions.

The trajectory was built from a set of paths linking a set of waypoints. The guidance system implemented for the purposes of these simulations was a LOS guidance system.

C.2.2 Control System

For the purpose of control, a fully coupled 6DOF model based SMC [181] was implemented here. This controller calculates the actuation force by modelling the forces generated due to the motion of the vehicle and calculating an additional force that would be required to minimize the trajectory error utilizing an error surface. Detail of this control system can be found in [134].
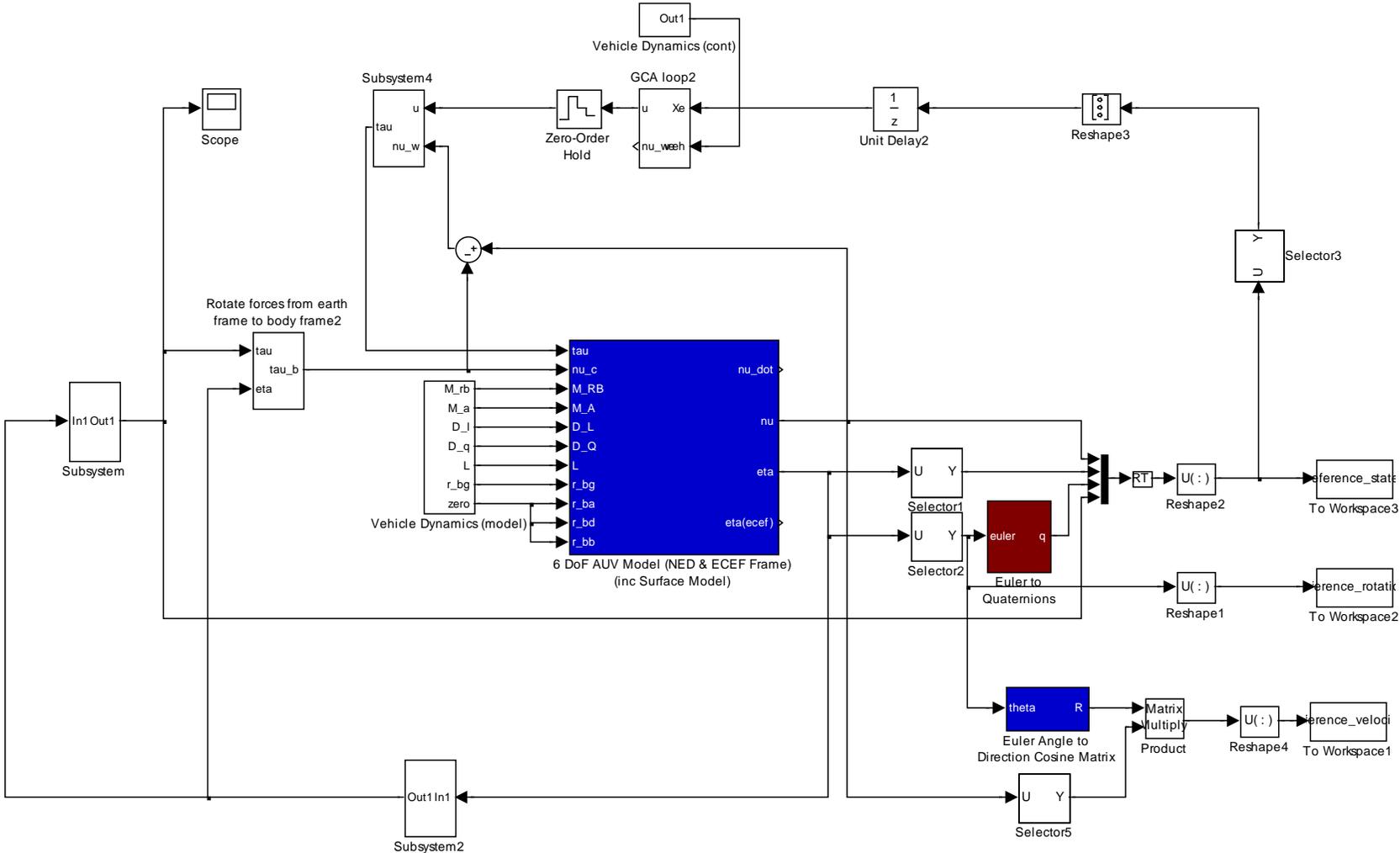
Figure C.1 Simulink block diagram of the AUV simulator
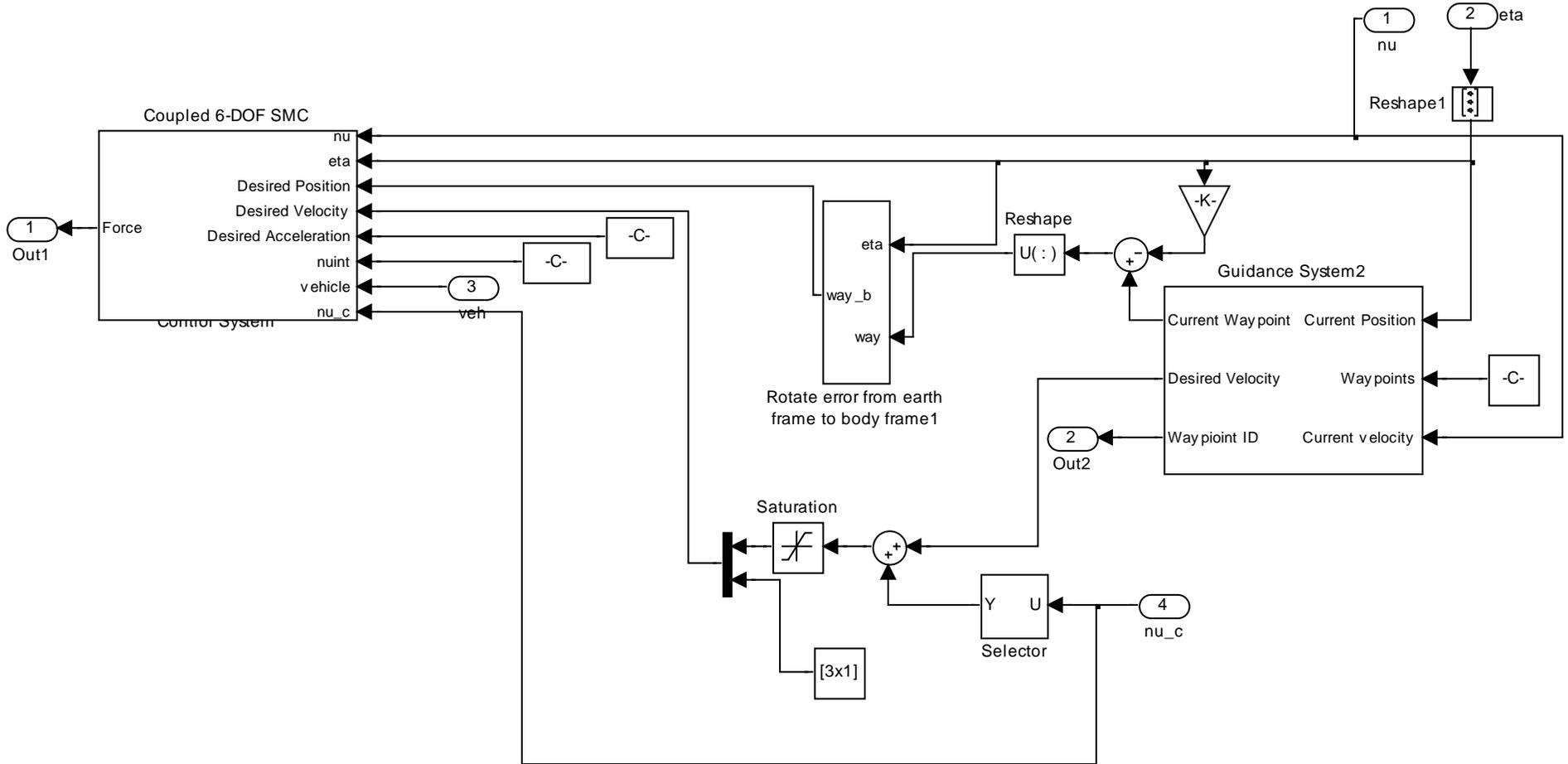
Figure C.2 Simulink block diagram of the guidance and control system

Appendix C. AUV Simulator
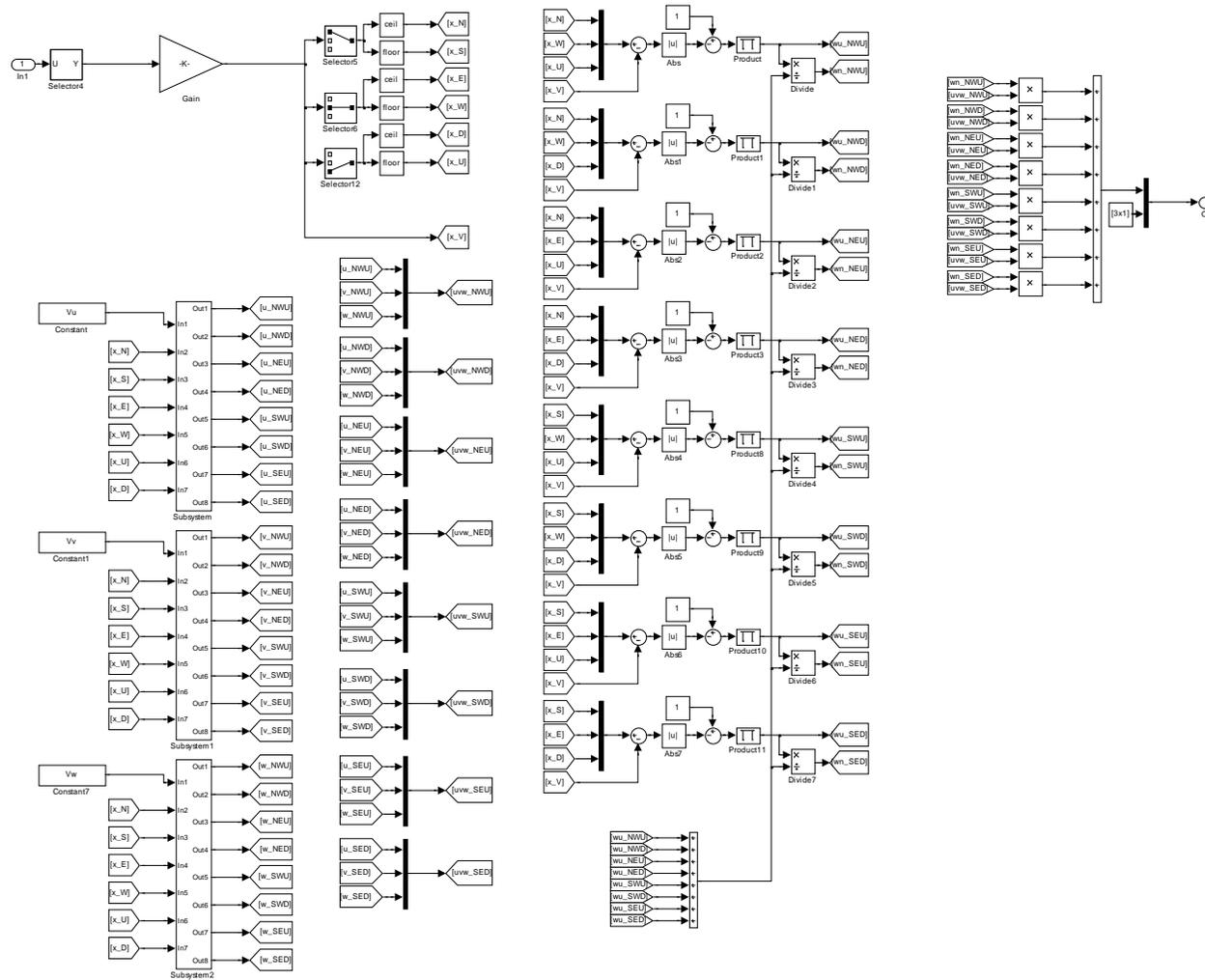
Figure C.3 Simulink block diagram of incorporating currents

Bibliography

1. Westwood, J., *The AUV market place.* Presentation at Oceanology International, London, 10th March, 2010.

2. Wang, D., et al., *Acoustically focused adaptive sampling and on-board routing for marine rapid environmental assessment.* Journal of Marine Systems, 2009. **78**(SUPPL. 1): p. S393-S407.

3. Chyba, M., et al., *Design and implementation of time efficient trajectories for autonomous underwater vehicles.* Ocean Engineering, 2008. **35**(1): p. 63-76.

4. Koay, T.B. and M. Chitre. *Energy-efficient path planning for fully propelled AUVs in congested coastal waters.* in *OCEANS 2013 MTS/IEEE Bergen: The Challenges of the Northern Dimension.* 2013.

5. Furlong, M.E., et al. *Autosub Long Range: A long range deep diving AUV for ocean monitoring.* in *2012 IEEE/OES Autonomous Underwater Vehicles, AUV 2012.* 2012.

6. *Navy 'Mine-Hunter' AUV Sets Mission Endurance Record* http://www.nrl.navy.mil/media/news-releases/2013/navy-mine-hunter-auv-sets-mission-endurance-record, 2013.

7. *Revolutionizing Maritime Technology and Monitoring-Unmanned Underwater/Surface Vessel* http://www.oceanaero.us/documents/OAHandout11MetricJul222014.pdf.

BIBLIOGRAPHY

8.   Hobson, B.W., et al. *Tethys-class long range AUVs-extending the endurance of propeller-driven cruising AUVs from days to weeks*. in *2012 IEEE/OES Autonomous Underwater Vehicles, AUV 2012*. 2012.

9.   Griffiths, G. and S.D. McPhail, *AUVs for depth and distance: Autosub6000 and autosub long range.* Sea Technology, 2011. **52**(12): p. 27-30.

10.   Stokey, R.P., et al. *Development of the REMUS 600 autonomous underwater vehicle*. in *Proceedings of MTS/IEEE OCEANS, 2005*. 2005.

11.   Taylor, M. and A. Wilby. *Design considerations and operational advantages of a modular AUV with synthetic aperture sonar*. in *OCEANS'11 - MTS/IEEE Kona, Program Book*. 2011.

12.   *Bluefin-21 Product Sheet - Bluefin Robotics.* http://www.bluefinrobotics.com/products/bluefin-21/.

13.   Hagen, P.E., et al. *The HUGIN 1000 Autonomous Underwater Vehicle for military applications*. in *Oceans Conference Record (IEEE)*. 2003.

14.   *Autonomous underwater vehicle - MUNIN AUV.* http://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/48D837005CB784DFC1257C970041051F?OpenDocument.

15.   *Bluefin-9 Product Sheet - Bluefin Robotics.* http://www.bluefinrobotics.com/assets/Downloads/Bluefin-9-Product-Sheet.pdf.

16.   Incze, M.L., *Lightweight Autonomous Underwater Vehicles (AUVs) performing coastal survey operations in REP 10A.* Ocean Dynamics, 2011. **61**(11): p. 1955-1965.

17.   Nygaard, K.H., *Meet the family.* International Ocean Systems, 2014. **18**(1): p. 6-7.

18.   *Teledyne Gavia.* http://www.teledynegavia.com/index.php/product_dashboard/auvs.

19.   Hasvold, Ø., et al., *Power sources for autonomous underwater vehicles.* Journal of Power Sources, 2006. **162**(2 SPEC. ISS.): p. 935-942.

20.   Abu Sharkh, S.M. and G. Griffiths, *Energy storage systems for unmanned underwater vehicles.* Underwater Technology, 2002. **25**(3): p. 143-148.

21.   Joung, T.H., et al., *Shape optimization of an autonomous underwater vehicle with a ducted propeller using computational fluid dynamics analysis.* International Journal of Naval Architecture and Ocean Engineering, 2012. **4**(1): p. 44-56.

22.   Joung, T., et al. *A study on the design optimization of an AUV by using computational fluid dynamic analysis.* in *Proceedings of the International Offshore and Polar Engineering Conference.* 2009.

23.   Kuwata, Y., et al. *Robust constrained receding horizon control for trajectory planning.* in *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference.* 2005.

24.   Dijkstra, E.W., *A note on two problems in connexion with graphs.* Numerische Mathematik, 1959. **1**(1): p. 269-271.

25.   Carroll, K.P., et al. *AUV path planning: an A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones.* in *Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology, 1992. AUV'92.* 1992. IEEE.

26.   Hernández, E., et al. *A search-based path planning algorithm with topological constraints. Application to an AUV.* in *IFAC Proceedings Volumes (IFAC-PapersOnline).* 2011.

27.   Ferguson, D. and A. Stentz, *Using interpolation to improve path planning the field D* algorithm.* Journal of Field Robotics, 2006. **23**(2): p. 79-101.

28.   Pereira, A.A., et al. *Toward risk aware mission planning for autonomous underwater vehicles.* in *IEEE International Conference on Intelligent Robots and Systems.* 2011.

29. Pereira, A.A., et al., *Risk-aware path planning for autonomous underwater vehicles using predictive ocean models.* Journal of Field Robotics, 2013. **30**(5): p. 741-762.

30. Garau, B., et al., *Path planning for autdndmdus underwater vehicles in realistic oceanic current fields: Application to gliders in the Western Mediterranean sea.* Journal of Maritime Research, 2009. **6**(2): p. 5-21.

31. Kruger, D., et al. *estuarine environments.* in *Proceedings - IEEE International Conference on Robotics and Automation.* 2007.

32. Witt, J. and M. Dunbabin. *Go with the flow: Optimal AUV path planning in coastal environments.* in *Proceedings of the 2008 Australasian Conference on Robotics and Automation, ACRA 2008.* 2008.

33. Pêtrès, C., et al., *Path planning for autonomous underwater vehicles.* IEEE Transactions on Robotics, 2007. **23**(2): p. 331-341.

34. Pêtrès, C., et al. *Underwater path planing using fast marching algorithms.* in *Oceans 2005 - Europe.* 2005.

35. Thompson, D.R., et al. *Spatiotemporal path planning in strong, dynamic, uncertain currents.* in *Proceedings - IEEE International Conference on Robotics and Automation.* 2010.

36. Soulignac, M., *Feasible and optimal path planning in strong current fields.* IEEE Transactions on Robotics, 2011. **27**(1): p. 89-98.

37. Soulignac, M., P. Taillibert, and M. Rueher. *Adapting the wavefront expansion in presence of strong currents.* in *Proceedings - IEEE International Conference on Robotics and Automation.* 2008.

38. Lolla, T., et al. *Path planning in time dependent flow fields using level set methods.* in *Proceedings - IEEE International Conference on Robotics and Automation.* 2012.

39. Wzorek, M. and P. Doherty. *Reconfigurable path planning for an autonomous unmanned aerial vehicle.* in *Proceedings - 2006 International Conference on Hybrid Information Technology, ICHIT 2006.* 2006.

40. Wzorek, M., J. Kvarnström, and P. Doherty. *Choosing path replanning strategies for unmanned aircraft systems*. in *ICAPS 2010 - Proceedings of the 20th International Conference on Automated Planning and Scheduling*. 2010.

41. Jung, D. and P. Tsiotras. *Multiresolution on-line path planning for small unmanned aerial vehicles*. in *Proceedings of the American Control Conference*. 2008.

42. Cowlagi, R.V. and P. Tsiotras. *Multiresolution path planning with wavelets: A local replanning approach*. in *Proceedings of the American Control Conference*. 2008.

43. Kuffner Jr, J.J. and S.M. La Valle. *RRT-connect: an efficient approach to single-query path planning*. in *Proceedings - IEEE International Conference on Robotics and Automation*. 2000.

44. Zucker, M., J. Kuffner, and M. Branicky. *Multipartite RRTs for rapid replanning in dynamic environments*. in *Proceedings - IEEE International Conference on Robotics and Automation*. 2007.

45. Ferguson, D., N. Kalra, and A. Stentz. *Replanning with RRTs*. in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* . 2006.

46. Tsourdos, A., B. White, and M. Shanmugavel, *Cooperative path planning of unmanned aerial vehicles*. Vol. 32. 2010: John Wiley & Sons.

47. Fiorelli, E., et al., *Multi-AUV control and adaptive sampling in Monterey Bay*. IEEE Journal of Oceanic Engineering, 2006. **31**(4): p. 935-948.

48. Fallon, M.F., et al., *Cooperative AUV navigation using a single maneuvering surface craft*. International Journal of Robotics Research, 2010. **29**(12): p. 1461-1474.

49. Knudson, M. and K. Tumer, *Adaptive navigation for autonomous robots*. Robotics and Autonomous Systems, 2011. **59**(6): p. 410-420.

50. Alvarez, A., A. Caiti, and R. Onken, *Evolutionary path planning for autonomous underwater vehicles in a variable ocean*. IEEE Journal of Oceanic Engineering, 2004. **29**(2): p. 418-429.

51.   Rao, D. and S.B. Williams. *Large-scale path planning for underwater gliders in ocean currents*. in *Proceedings of the 2009 Australasian Conference on Robotics and Automation, ACRA 2009*. 2009.

52.   Smith, R.N., et al., *Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model*. International Journal of Robotics Research, 2010. **29**(12): p. 1475-1497.

53.   Smith, R.N., et al. *An investigation on the accuracy of Regional Ocean Models through field trials*. in *Proceedings - IEEE International Conference on Robotics and Automation*. 2013.

54.   Besada-Portas, E., et al., *Evolutionary trajectory planner for multiple UAVs in realistic scenarios*. IEEE Trans. Robotics., 2010. **26**(4): p. 619-634.

55.   Roberge, V., M. Tarbouchi, and G. Labonte, *Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning*. IEEE Transactions on Industrial Informatics, 2013. **9**(1): p. 132-141.

56.   Zeng, Z., et al. *Efficient path evaluation for AUVs using adaptive B-spline approximation*. in *OCEANS 2012 MTS/IEEE: Harnessing the Power of the Ocean*. 2012.

57.   Zeng, Z., et al., *Shell space decomposition based path planning for AUVs operating in a variable environment*. Ocean Engineering, 2014. **91**(0): p. 181-195.

58.   Zeng, Z., et al. *Optimal path planning based on annular space decomposition for AUVs operating in a variable environment*. in *2012 IEEE/OES Autonomous Underwater Vehicles, AUV 2012*. 2012.

59.   Zeng, Z., et al., *Efficient Path Re-planning for AUVs Operating in Spatiotemporal Currents*. Journal of Intelligent & Robotic Systems, 2014: p. 1-19.

60.   Zeng, Z., et al. *Path planning for rendezvous of multiple AUVs operating in a variable ocean*. in *2014 IEEE 4th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER),*. 2014. IEEE.

61. Cui, R., et al., *Leader-follower formation control of underactuated autonomous underwater vehicles.* Ocean Engineering, 2010. **37**(17-18): p. 1491-1502.

62. Alvarez, A., A. Caiti, and R. Onken, *Evolutionary path planning for autonomous underwater vehicles in a variable ocean.* IEEE J. Oceanic Eng., 2004. **29**(2): p. 418-429.

63. Nikolos, I.K., et al., *Evolutionary Algorithm Based Offline/Online Path Planner for UAV Navigation.* IEEE Trans. Syst. Man, Cybern. B, Cybern., 2003. **33**(6): p. 898-912.

64. Zheng, C., et al., *Evolutionary route planner for unmanned air vehicles.* IEEE Trans. Robotics., 2005. **21**(4): p. 609-620.

65. Saska, M., et al. *Robot path planning using particle swarm optimization of Ferguson splines.* in *IEEE Conference on Emerging Technologies and Factory Automation (ETFA '06), 2006.* 2006. Hamburg.

66. Sun, J., B. Feng, and W. Xu. *Particle swarm optimization with particles having quantum behavior.* in *Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004.* 2004.

67. Clerc, M. and J. Kennedy, *The particle swarm-explosion, stability, and convergence in a multidimensional complex space.* IEEE Transactions on Evolutionary Computation, 2002. **6**(1): p. 58-73.

68. Mikki, S.M. and A.A. Kishk, *Quantum particle swarm optimization for electromagnetics.* IEEE Transactions on Antennas and Propagation, 2006. **54**(10): p. 2764-2775.

69. Coelho, L.S., *Novel Gaussian quantum-behaved particle swarm optimiser applied to electromagnetic design.* IET Science, Measurement and Technology, 2007. **1**(5): p. 290-294.

70. Mikki, S.M. and A.A. Kishk, *Theory and applications of infinitesimal dipole models for computational electromagnetics.* IEEE Transactions on Antennas and Propagation, 2007. **55**(5): p. 1325-1337.

71.   Omkar, S.N., et al., *Quantum behaved Particle Swarm Optimization (QPSO) for multi-objective design optimization of composite structures.* Expert Systems with Applications, 2009. **36**(8): p. 11312-11322.

72.   Coelho, L.d.S., *Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems.* Expert Systems with Applications, 2010. **37**(2): p. 1676-1683.

73.   Horng, M.H., *Vector quantization using the firefly algorithm for image compression.* Expert Systems with Applications, 2012. **39**(1): p. 1078-1091.

74.   Liu, F., H. Duan, and Y. Deng, *A chaotic quantum-behaved particle swarm optimization based on lateral inhibition for image matching.* Optik, 2012. **123**(21): p. 1955-1960.

75.   dos Santos Coelho, L. and V.C. Mariani, *Particle swarm approach based on quantum mechanics and harmonic oscillator potential well for economic load dispatch with valve-point effects.* Energy Conversion and Management, 2008. **49**(11): p. 3080-3085.

76.   Sun, J., et al., *Solving the economic dispatch problem with a modified quantum-behaved particle swarm optimization method.* Energy Conversion and Management, 2009. **50**(12): p. 2967-2975.

77.   Meng, K., et al., *Quantum-inspired particle swarm optimization for valve-point economic load dispatch.* IEEE Transactions on Power Systems, 2010. **25**(1): p. 215-222.

78.   Kanayama, Y. and S.i. Yuta, *VEHICLE PATH SPECIFICATION BY A SEQUENCE OF STRAIGHT LINES.* IEEE journal of robotics and automation, 1988. **4**(3): p. 265-276.

79.   Dubins, L.E., *On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents.* American Journal of mathematics, 1957: p. 497-516.

80.   Reeds, J. and L. Shepp, *Optimal paths for a car that goes both forwards and backwards.* Pacific Journal of Mathematics, 1990. **145**(2): p. 367-393.

81. Chitsaz, H. and S.M. LaValle. *Time-optimal paths for a dubins airplane*. in *Proceedings of the IEEE Conference on Decision and Control*. 2007.

82. Hota, S. and D. Ghose, *Optimal Trajectory Generation for Convergence to a Rectilinear Path.* Journal of Intelligent and Robotic Systems: Theory and Applications, 2013: p. 1-20.

83. Techy, L. and C.A. Woolsey, *Minimum-time path planning for unmanned aerial vehicles in steady uniform winds.* Journal of Guidance, Control, and Dynamics, 2009. **32**(6): p. 1736-1746.

84. Bakolas, E. and P. Tsiotras. *Time-optimal synthesis for the Zermelo-Markov-Dubins problem: The constant wind case*. in *Proceedings of the 2010 American Control Conference, ACC 2010*. 2010.

85. Bakolas, E. and P. Tsiotras, *Optimal partitioning for spatiotemporal coverage in a drift field.* Automatica, 2013. **49**(7): p. 2064-2073.

86. Tsourdos, A., B. White, and M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles*. Cooperative Path Planning of Unmanned Aerial Vehicles. 2010.

87. Fraichard, T. and A. Scheuer, *From Reeds and Shepp's to continuous-curvature paths.* IEEE Transactions on Robotics, 2004. **20**(6): p. 1025-1035.

88. Bruyninckx, H. and D. Reynaerts. *Path planning for mobile and hyper-redundant robots using Pythagorean hodograph curves*. in *International Conference on Advanced Robotics, Proceedings, ICAR*. 1997.

89. Lekkas, A.M. and T.I. Fossen, *Integral LOS Path Following for Curved Paths Based on a Monotone Cubic Hermite Spline Parametrization.* IEEE Transactions on Control Systems Technology, 2014.

90. Harary, G. and A. Tal, *3D Euler spirals for 3D curve completion.* Computational Geometry: Theory and Applications, 2012. **45**(3): p. 115-126.

91. Lekkas, A.M., et al., *Continuous-Curvature Path Generation Using Fermat.* 2013.

92. Qu, Z., J. Wang, and C.E. Plaisted, *A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles.* IEEE Transactions on Robotics, 2004. **20**(6): p. 978-993.

93. Fossen, T.I., *Handbook of Marine Craft Hydrodynamics and Motion Control.* Handbook of Marine Craft Hydrodynamics and Motion Control. 2011.

94. Wagner, P., et al. *Motion control for robots based on cubic hermite splines in real-time.* in *IFAC Proceedings Volumes (IFAC-PapersOnline).* 2010.

95. Jolly, K.G., R. Sreerama Kumar, and R. Vijayakumar, *A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits.* Robotics and Autonomous Systems, 2009. **57**(1): p. 23-33.

96. Lapierre, L. and B. Jouvencel, *Robust nonlinear path-following control of an AUV.* IEEE Journal of Oceanic Engineering, 2008. **33**(2): p. 89-102.

97. Antonelli, G., S. Chiaverini, and G. Fusco, *A fuzzy-logic-based approach for mobile robot path tracking.* IEEE Transactions on Fuzzy Systems, 2007. **15**(2): p. 211-221.

98. Breivik, M. and T.I. Fossen. *Applying missile guidance concepts to motion control of marine craft.* in *IFAC Proceedings Volumes (IFAC-PapersOnline).* 2007.

99. LaValle, S.M. and J.J. Kuffner Jr, *Randomized kinodynamic planning.* International Journal of Robotics Research, 2001. **20**(5): p. 378-400.

100. Tovar, B., R. Murrieta-Cid, and S.M. LaValle, *Distance-optimal navigation in an unknown environment without sensing distances.* IEEE Transactions on Robotics, 2007. **23**(3): p. 506-518.

101. Hart, P.E., N.J. Nilsson, and B. Raphael, *A formal basis for the heuristic determination of minimum cost paths.* Systems Science and Cybernetics, IEEE Transactions on, 1968. **4**(2): p. 100-107.

102. Nash, A., et al. *Theta*: Any-angle path planning on grids.* in *Proceedings of the National Conference on Artificial Intelligence.* 2007.

103. Daniel, K., et al., *Theta*: Any-angle path planning on grids.* Journal of Artificial Intelligence Research, 2010. **39**: p. 533-579.

104. Soulignac, M., P. Taillibert, and M. Rueher. *Time-minimal path planning in dynamic current fields.* in *Proceedings - IEEE International Conference on Robotics and Automation.* 2009.

105. Sethian, J.A., *Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts.* Journal of Computational Physics, 2001. **169**(2): p. 503-555.

106. Warren, C.W. *A technique for autonomous underwater vehicle route planning.* in *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology.* 1990.

107. Barraquand, J., B. Langlois, and J.-C. Latombe, *Numerical potential field techniques for robot path planning.* IEEE Transactions on Systems, Man and Cybernetics, 1992. **22**(2): p. 224-241.

108. Ferguson, D. and A. Stentz. *Anytime RRTs.* in *IEEE International Conference on Intelligent Robots and Systems.* 2006.

109. Tan, C.S., R. Sutton, and J. Chudley. *Quasi-random, Manoeuvre-based motion planning algorithm for autonomous underwater vehicles.* in *IFAC Proceedings Volumes (IFAC-PapersOnline).* 2005.

110. Nikolos, I.K., et al., *Evolutionary Algorithm Based Offline/Online Path Planner for UAV Navigation.* IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2003. **33**(6): p. 898-912.

111. Besada-Portas, E., et al., *On the performance comparison of multi-objective evolutionary UAV path planners.* Information Sciences, 2013. **238**: p. 111-125.

112. Roberge, V., M. Tarbouchi, and F. Allaire, *Parallel hybrid metaheuristic on shared memory system for real-time UAV path planning.* International Journal of Computational Intelligence and Applications, 2014. **13**(2).

113. Eberhart, R.C. and Y. Shi. *Comparison between genetic algorithms and particle swarm optimization.* in *Evolutionary Programming VII.* 1998. Springer.

114. Fu, Y., M. Ding, and C. Zhou, *Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV.* IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans, 2012. **42**(2): p. 511-526.

115. Brassington, G.B., et al., *BLUElink〉 development of operational oceanography and servicing in Australia.* Journal of Research and Practice in Information Technology, 2007. **39**(2): p. 151-162.

116. Oke, P.R., et al., *The Bluelink ocean data assimilation system (BODAS).* Ocean Modelling, 2008. **21**(1-2): p. 46-70.

117. Garau, B., A. Alvarez, and G. Oliver. *AUV navigation through turbulent ocean environments supported by onboard H-ADCP.* in *Proceedings - IEEE International Conference on Robotics and Automation.* 2006.

118. Gonzalez, J.P. and A. Stentz. *Planning with uncertainty in position: An optimal and efficient planner.* in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS.* 2005.

119. Garau, B., A. Alvarez, and G. Oliver. *Path planning of autonomous underwater vehicles in current fields with complex spatial variability: An A* approach.* in *Proceedings - IEEE International Conference on Robotics and Automation.* 2005.

120. Karaman, S. and E. Frazzoli, *Sampling-based algorithms for optimal motion planning.* International Journal of Robotics Research, 2011. **30**(7): p. 846-894.

121. Lane, J.M. and R.F. Riesenfeld, *A theoretical development for the computer generation and display of piecewise polynomial surfaces.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1980. **2**(1): p. 35-46.

122. Shi, Y. and R. Eberhart. *Modified particle swarm optimizer.* in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC.* 1998.

123. Shi, Y. and R.C. Eberhart. *Empirical study of particle swarm optimization.* in *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999.* 1999.

124. Parsopoulos, K.E. and M.N. Vrahatis, *Parameter selection and adaptation in Unified Particle Swarm Optimization.* Mathematical and Computer Modelling, 2007. **46**(1-2): p. 198-213.

125. Zheng, C., M. Ding, and C. Zhou, *Real-time route planning for unmanned air vehicle with an evolutionary algorithm.* International Journal of Pattern Recognition and Artificial Intelligence, 2003. **17**(1): p. 63-81.

126. Yi, M., M. Ding, and C. Zhou. *3D route planning using genetic algorithm.* in *Proceedings of SPIE - The International Society for Optical Engineering.* 1998.

127. Nikolos, I.K., E.S. Zografos, and A.N. Brintaki, *UAV path planning using evolutionary algorithms*, in *Studies in Computational Intelligence.* 2007. p. 77-111.

128. LaValle, S.M., *Planning algorithms.* 2006: Cambridge university press.

129. Jiang, K., L.D. Seneviratne, and S.W.E. Earles, *Time-optimal smooth-path motion planning for a mobile robot with kinematic constraints.* Robotica, 1997. **15**(5): p. 547-553.

130. Hao, Y., W. Zu, and Y. Zhao. *Real-time obstacle avoidance method based on polar coordination particle swarm optimization in dynamic environment.* in *IEEE Conference on Industrial Electronics and Applications (ICIEA) 2007.* 2007. Harbin.

131. Garau, B., A. Alvarez, and G. Oliver. *Path planning of autonomous underwater vehicles in current fields with complex spatial variability: An A* approach.* in *IEEE International Conference on Robotics and Automation, ICRA 2005. .* 2005. Barcelona.

132. Lammas, A.K., K. Sammut, and F. He. *A 6 DoF navigation algorithm for autonomous underwater vehicles.* in *IEEE Oceans/MTS.* 2007. Aberdeen, Scotland.

133. Fossen, T.I., *Marine control systems: Guidance, navigation and control of ships, rigs and underwater vehicles.* 2002: Marine Cybernetics Trondheim, Norway.

134. Kokegei, M., F. He, and K. Sammut. *Fully coupled 6 degrees-of-freedom control of autonomous underwater vehicles*. in *IEEE/MTS Oceans 2008*. 2008. Quebec City, QC.

135. Kim, K. and T. Ura. *Towards a new strategy for AUV navigation in sea currents: A quasi-optimal approach*. in *Underwater Technology (UT), 2011 IEEE Symposium on and 2011 Workshop on Scientific Use of Submarine Cables and Related Technologies (SSC)*. 2011. Tokyo.

136. Smith, R.N., et al., *Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model*. Int. J. Robotics. Res., 2010. **29**(12): p. 1475-1497.

137. Galceran, E., et al., *Coverage Path Planning with Real-time Replanning and Surface Reconstruction for Inspection of Three-dimensional Underwater Structures using Autonomous Underwater Vehicles*. Journal of Field Robotics, 2014.

138. Smith, R.N., et al. *Autonomous underwater vehicle trajectory design coupled with predictive ocean models: A case study*. in *Proceedings - IEEE International Conference on Robotics and Automation*. 2010.

139. Hollinger, G.A., A.A. Pereira, and G.S. Sukhatme. *Learning uncertainty models for reliable operation of Autonomous Underwater Vehicles*. in *Proceedings - IEEE International Conference on Robotics and Automation*. 2013.

140. Fong, D.A. and N.L. Jones, *Evaluation of AUV-based ADCP measurements*. Limnology and Oceanography: Methods, 2006. **4**(MAR.): p. 58-67.

141. Ordonez, C.E., et al. *Obtaining absolute water velocity profiles from glider-mounted Acoustic Doppler Current Profilers*. in *Program Book - OCEANS 2012 MTS/IEEE Yeosu: The Living Ocean and Coast - Diversity of Resources and Sustainable Activities*. 2012.

142. Thurnherr, A.M., *A practical assessment of the errors associated with full-depth LADCP profiles obtained using teledyne RDI workhorse acoustic doppler current profilers*. Journal of Atmospheric and Oceanic Technology, 2010. **27**(7): p. 1215-1227.

143. Firing, E. *GPS attitude determination for shipboard Doppler current profiling*. in *Oceans Conference Record (IEEE)*. 1991.

144. Tam, C. and R. Bucknall, *Cooperative path planning algorithm for marine surface vessels*. Ocean Engineering, 2013. **57**: p. 25-33.

145. Yang, Y., et al., *Motion planning for multi-HUG formation in an environment with obstacles*. Ocean Engineering, 2011. **38**(17-18): p. 2262-2269.

146. Bhattacharya, S., M. Likhachev, and V. Kumar, *Topological constraints in search-based robot path planning*. Autonomous Robots, 2012. **33**(3): p. 273-290.

147. Cui, R., B. Gao, and J. Guo, *Pareto-optimal coordination of multiple robots with safety guarantees*. Autonomous Robots, 2012. **32**(3): p. 189-205.

148. Lau, B., C. Sprunk, and W. Burgard. *Kinodynamic motion planning for mobile robots using splines*. in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*. 2009.

149. Li, Z., et al., *Robust adaptive motion control for underwater remotely operated vehicles with velocity constraints*. International Journal of Control, Automation and Systems, 2012. **10**(2): p. 421-429.

150. Li, Z., et al., *Adaptive fuzzy-based motion generation and control of mobile under-actuated manipulators*. Engineering Applications of Artificial Intelligence, 2014. **30**: p. 86-95.

151. Yang, C., Z. Li, and J. Li, *Trajectory planning and optimized adaptive control for a class of wheeled inverted pendulum vehicle models*. IEEE Transactions on Cybernetics, 2013. **43**(1): p. 24-36.

152. Barnsley, M.F. and M. Frame, *The influence of benoît B. Mandelbrot on mathematics*. Notices of the American Mathematical Society, 2012. **59**(9): p. 1208-1221.

153. Rigby, P., O. Pizarro, and S.B. Williams, *Toward adaptive benthic habitat mapping using gaussian process classification*. Journal of Field Robotics, 2010. **27**(6): p. 741-758.

154. Manley, J.E. *Multiple AUV missions in the national oceanic and atmospheric administration*. in *2004 IEEE/OES Autonomous Underwater Vehicles*. 2004.

155. Yoon, S. and C. Qiao, *Cooperative search and survey using Autonomous Underwater Vehicles (AUVs)*. IEEE Transactions on Parallel and Distributed Systems, 2012. **23**(3): p. 364-379.

156. Chatzichristofis, S., et al. *The NOPTILUS project: Autonomous multi-AUV navigation for exploration of unknown environments*. in *IFAC Proceedings Volumes (IFAC-PapersOnline)*. 2012.

157. Häusler, A.J., et al. *Cooperative motion planning for multiple autonomous marine vehicles*. in *IFAC Proceedings Volumes (IFAC-PapersOnline)*. 2012.

158. Leonard, N.E., et al., *Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay*. Journal of Field Robotics, 2010. **27**(6): p. 718-740.

159. Carlson, E.A., P.P.J. Beaujean, and E. An, *Location-aware source routing protocol for underwater acoustic networks of AUVs*. Journal of Electrical and Computer Engineering, 2012.

160. German, C.R., et al. *A long term vision for long-range ship-free deep ocean operations: Persistent presence through coordination of Autonomous Surface Vehicles and Autonomous Underwater Vehicles*. in *2012 IEEE/OES Autonomous Underwater Vehicles, AUV 2012*. 2012.

161. Zebrowski, P., Y. Litus, and R.T. Vaughan. *Energy efficient robot rendezvous*. in *Proceedings - Fourth Canadian Conference on Computer and Robot Vision, CRV 2007*. 2007.

162. Marco, D.B. and A.J. Healey. *Current developments in underwater vehicle control and navigation: the NPS ARIES AUV*. in *Oceans Conference Record (IEEE)*. 2000.

163. Sanz, P.J., et al. *TRIDENT: Recent improvements about autonomous underwater intervention missions*. in *IFAC Proceedings Volumes (IFAC-PapersOnline)*. 2012.

164. Häusler, A.J., et al. *Temporally and spatially deconflicted path planning for multiple autonomous marine vehicles*. in *IFAC Proceedings Volumes (IFAC-PapersOnline)*. 2009.

165. Deng, Y., et al. *Task allocation and path planning for collaborative AUVs operating through an underwater acoustic network*. in *MTS/IEEE Seattle, OCEANS 2010*. 2010.

166. Häusler, A.J., et al. *Multiple marine vehicle deconicted path planning with currents and communication constraints*. in *IFAC Proceedings Volumes (IFAC-PapersOnline)*. 2010.

167. Litus, Y., P. Zebrowski, and R.T. Vaughan, *A distributed heuristic for energy-efficient multirobot multiplace rendezvous*. IEEE Transactions on Robotics, 2009. **25**(1): p. 130-135.

168. Bajaj, C., *The algebraic degree of geometric optimization problems*. Discrete & Computational Geometry, 1988. **3**(1): p. 177-191.

169. Cieliebak, M., et al., *Solving the robots gathering problem*, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2003. p. 1181-1196.

170. Martínez, S., et al., *On synchronous robotic networks - Part II: Time complexity of rendezvous and deployment algorithms*. IEEE Transactions on Automatic Control, 2007. **52**(12): p. 2214-2226.

171. Atashpaz-Gargari, E. and C. Lucas. *Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition*. in *2007 IEEE Congress on Evolutionary Computation, CEC 2007*. 2007.

172. Hoseini, R. and H. Salehipoor, *Optimum design process of vibration absorber via imperialist competitive algorithm*. International Journal of Structural Stability and Dynamics, 2012. **12**(3).

173. Mohammadi-ivatloo, B., et al., *Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch*. Energy, 2012. **44**(1): p. 228-240.

174. Jolai, F., M. Rabiee, and H. Asefi, *A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times.* International Journal of Production Research, 2012. **50**(24): p. 7447-7466.

175. Eichhorn, M., *Optimal routing strategies for autonomous underwater vehicles in time-varying environment.* Robotics and Autonomous Systems, 2013.

176. Latombe, J.-C., *Robot Motion Planning: Edition en anglais.* 1991: Springer.

177. Cowlagi, R.V. and P. Tsiotras, *Hierarchical motion planning with dynamical feasibility guarantees for mobile robotic vehicles.* IEEE Transactions on Robotics, 2012. **28**(2): p. 379-395.

178. Bakolas, E. and P. Tsiotras. *Multiresolution path planning via sector decompositions compatible to on-board sensor data.* in *AIAA Guidance, Navigation and Control Conference and Exhibit.* 2008.

179. Nieuwenhuisen, M., R. Steffens, and S. Behnke, *Local multiresolution path planning in soccer games based on projected intentions*, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* 2012. p. 495-506.

180. Prestero, T., *Development of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle*, in *12th International Symposium on Unmanned Untethered Submersible Technology.* 2001: University of New Hampshire, Durham, NH.

181. Kokegei, M., F. He, and K. Sammut, *Nonlinear Fully-Coupled Control of AUVs*, in *Society of Underwater Technology Annual Conference.* 2009: Perth, Australia.