Flinders UNIVERSITY
ADELAIDE • SOUTH AUSTRALIA

A Unified Guidance Framework for AUV Docking Operations

by

Amir Mehdi Yazdani

Thesis
Submitted to Flinders University
for the degree of

Doctor of Philosophy
College of Science and Engineering
July 2017

Contents

List of Figures

List of Tables

Abstract

In the recent years, autonomous underwater vehicles (AUVs) have become a key tool for underwater surveys and explorations. A plethora of successful underwater missions and task allocations performed by AUVs such as detection, sampling, mapping, surveillance, and reconnaissance have been documented. The expanding mission scope for AUVs highlights the need for a long-endurance operational capability, which mainly depends on propulsion system efficiency and battery capacity. While there have been some significant improvements in battery energy density, the capacity to conduct long-term missions extended over many days to months is still not possible. This restriction imposes an extra cost of manned deployment and recovery to extend mission duration. The use of submerged docking systems permitting battery recharges and data downloads/uploads, is therefore essential to enabling persistence while reducing operational costs and hazards.

This thesis develops a systematic approach for analyzing underwater docking operations from the optimal control theory standpoint, and offers a robust and efficient docking guidance framework that can address the limitations of existing docking guidance solutions. This unified framework is established upon formulation of the two-point boundary-value problem and utilizes well-known direct methods of calculus of variations to transcribe the conventional docking problem into an equivalent nonlinear programming problem so that to generate optimal and tractable docking trajectories. These methods include the inverse dynamics in the virtual domain and several pseudospectral computational techniques. The developed framework provides advanced flexibility and effectiveness enabling an AUV to compute not only open-loop docking guidance solutions with no uncertainties, but also closed-loop (continuously-updated) control solutions with respect to situational awareness of operating environments and uncertainties associated with docking station poses, or docking with a moving station. The framework combines both homing and docking phases in one operation, and enables smooth and stable approach of an AUV into a docking station while satisfying all realistic vehicular and environmental constraints and using minimum thrust and/or mission time in comparison with other existing docking algorithms.

The overall performance of the proposed guidance framework and particularly the capabilities and main features of the direct methods employed as the trajectory generator engines, are investigated through a series of docking scenarios in operating environments comprised of realistic currents and no-fly zone areas with respect to a priori known and unknown poses of docking station. The feasibility and robustness of trajectories from the standpoint of their realization on a real AUV are verified using a high-fidelity software-in-the-loop simulation platform and Monte Carlo simulations.

The new guidance framework developed in this thesis contributes to the cause of improving AUV autonomy by enabling longer mission durations while assuring reliable and cost-efficient docking operations.

Nomenclature

t	Time
t_0	Initial time
t_f	Final time
X	State vector
U	Control vector
$\Re^n$	The n-dimensional state space
$\Re^m$	The m-dimensional control space
υ	Flat output vector
$\hbar$	Bijective flat transformation
U^*	Optimal controls' vector
Δ	Unmodeled dynamics and uncertainties
t_{sample}	Computational time in RHC schemes
$t_{horizon}$	Horizon time in RHC schemes
t_i	The ith time instance of time interval
$\{n\}$	The world frame
$\{b\}$	The AUV body frame
η	The AUV position vector
V	The AUV velocity vector
x	The AUV position(center of gravity) in x-direction in the {n}-frame
y	The AUV position(center of gravity) in y-direction in the {n}-frame
z	The AUV position(center of gravity) in z-direction in the {n}-frame
ϕ	The AUV roll angle in the {n}-frame
θ	The AUV pitch angle in the {n}-frame
ψ	The AUV yaw (heading) angle in the {n}-frame
u	The AUV surge velocity in the {b}-frame
v	The AUV sway velocity in the {b}-frame
w	The AUV heave velocity in the {b}-frame
p	The AUV angular velocity in x-direction in the {b}-frame
q	The AUV angular velocity in y-direction in the {b}-frame

r	The AUV angular velocity in z-direction in the {b}-frame
${}_{b}^{n}R$	Euler translational velocity transformation matrix
${}_{b}^{n}W$	Euler angular velocity transformation matrix
M	The AUV mass and inertia matrix
$C(V)$	The AUV Coriolis and centripetal matrix
$D(V)$	The AUV hydrodynamic damping matrix
G	The AUV vector of gravitational and buoyancy
τ_c	The AUV external force and torque input vector
W_e	The AUV vector of environmental disturbances
M_{RB}	The AUV rigid body mass matrix
M_A	The AUV hydrodynamic added mass matrix
I	The AUV inertia matrix
m	The AUV mass
$C_{RB}(V)$	The AUV rigid body Coriolis and centripetal matrix
$C_A(V)$	The AUV Coriolis-like matrix
D_L	The AUV linear drag component
D_Q	The AUV quadratic drag term
W	The AUV gravity forces
B	The AUV buoyancy forces
CG	The AUV center of gravity vector
CB	The AUV center of buoyancy vector
L_c	The AUV actuators mapping matrix
c_x ,c_y	The northerly and easterly components of the current velocity
X_u, Z_w, N_r	The linear drag terms
$X_{u\|u\|}$, $Z_{w\|w\|}$,$N_{r\|r\|}$	The quadratic drag terms
I_z	The AUV’s inertia in the z-direction
T_u,T_w,T_r	The AUV control inputs in surge, heave and yaw directions
$\Phi(.)$	The Mayer cost function
$\chi(.)$	The Lagrange cost function
$\lambda(t)$	Co-states vector
μ	Vector of Lagrangian multipliers
C^L,C^U	The lower and upper bounds of constraint functions
γ_d	Position vector associated with the DS sector in the {n}-frame
ψ_d	Orientation of the DS sector with respect to the North
α_d	The entrance angle of the DS cone
h_d	The DS sector length
R_d , r_d	The outer and internal radii of the DS sector
X_0	Vector of initial boundary conditions
X_f	Vector of final boundary conditions
η_f	The AUV final pose vector

J	Performance index
H	Hamiltonian function
T_u^{max}	The maximum thrust on the AUV forward thrusters
T_w^{max}	The maximum thrust on the AUV vertical thrusters
T_r^{max}	The maximum thrust on the AUV horizontal thrusters
r^{max}	The maximum AUV yaw rate
x_{nf},y_{nf},z_{nf}	The 3D origin of the NFZ in the {n}-frame
r_{nf}	The radius of the NFZ
$\lambda^*(t)$	Optimal co-states' vector
$\mathrm{X}^*(t)$	The optimal trajectory generated the SITLSP
$\hat{\mathrm{X}}(t_k)$	The Estimated trajectory at the t_k time instance in the SITLSP
$\mathrm{X}^{ref}(t_k)$	The reference trajectory in the SITLSP
$\delta\mathrm{X}(t_k)$	The SMC input trajectory in the SITLSP
$\mathrm{U}^{cmd}(t_k)$	The command controls in the SITLSP
x^f_{SITLSP}	The final AUV position in x-direction generated by the SITLSP
y^f_{SITLSP}	The final AUV position in y-direction generated by the SITLSP
z^f_{SITLSP}	The final AUV position in z-direction generated by the SITLSP
u^f_{SITLSP}	The final AUV surge velocity generated by the SITLSP
w^f_{SITLSP}	The final AUV heave velocity generated by the SITLSP
r^f_{SITLSP}	The final AUV yaw rate generated by the SITLSP
$\Delta\eta,\Delta\psi$	The position and heading terminal errors associated with the docking-enabling conditions
$\Delta u,\Delta w,\Delta r$	The surge, heave, and yaw rate terminal errors associated with the docking-enabling conditions
$\varepsilon_u,\varepsilon_w,\varepsilon_r$	The tolerance corresponding to surge, heave, and yaw rate terminal errors
$\upsilon_u,\upsilon_w,\upsilon_r$	The new controls in the 10-state formulation
$\upsilon_u^{max},\upsilon_w^{max},\upsilon_r^{max}$	The maximum values of the new controls in the 10-state formulation
$\bar{t}$	The scaled time parameter
$\phi(\bar{t})$	The Lagrange interpolating polynomial
$L_N(\bar{t})$	The Legendre polynomial
w_k	Quadrature weights
$T_N(\bar{t})$	The Chebyshev polynomial
D_{ki}	The PS methods differentiation matrix
ε_d	The accuracy tolerance related to discretized dynamics and path constraints in the hp-AR method
κ	Trajectory curvature
r_k	Trajectory curvature ratio
r_{max}	Maximum limit on the trajectory curvature ratio

$e^{(k)}_{max}$	The maximum value of error associated with dynamic or path constraints
t_{CPU}	Computation time
$\wp$	Random sample
ℓ^b,u^b	Upper and lower bounds of random samples
V_c	Vector of current disturbances
τ	Virtual domain parameter
$\lambda(\tau)$	Speed factor
n_p	The minimum degree of polynomial
d_0, d_f	The highest-orders of time derivative of the reference function at the initial and terminal points
τ_f	The length of the virtual arc
N	Number of nodes
Δt	Time step
k_i	Weighting coefficient
D_r	Range from the true DS position

List of Abbreviations

ADCP	Acoustic Doppler Current Profiler
APF	Artificial Potential Field
A/S	Air to Surface
AUV	Autonomous Underwater Vehicle
CFD	Computational Fluid Dynamic
CG	Chebyshev Gauss
CGL	Chebyshev Gauss Lobatto
CPS	Chebyshev Pseudospectral
DoF	Degree of Freedom
DS	Docking Station
FSM	Finite State Machine
HBVP	Hamiltonian Boundary Value Problem
hp-AR	hp-Adaptive Radau
IDVD	Inverse Dynamics in the Virtual Domain
ILOS	Integral Line of Sight
KKT	Karush-Kuhn-Tucker
LBL	Long Baseline
LG	Legendre Gauss
LGL	Legendre Gauss Lobatto
LGR	Legendre Gauss Radau
LOS	Line of Sight
LPS	Legendre Pseudospectral
MC	Monte Carlo
NFZ	No-Fly Zone

LIST OF ABBREVIATIONS

NLP	Nonlinear Programming
OCP	Optimal Control Problem
PDF	Probability Density Function
PMP	Pontryagin Maximum Principle
PS	Pseudospectral
QPSO	Quantum Particle Swarm Optimization
ROV	Remotely Operated Vehicle
RPS	Radau Pseudospectral
RHC	Receding Horizon Control
SA	Situational Awareness
SGR	Speed Regulated Guidance
SITLSP	Software –In-The-Loop Simulation Platform
SMC	Sliding Mode Controller
TPBVP	Two-Point Boundary-Value Problem
UCAV	Unmanned Combat Aerial Vehicle
USBL	Ultra-Short Baseline
WHOI	Wood Hole Oceanographic Institute
WP	Waypoint

Declaration

I certify that this thesis does not incorporate without acknowledgment 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.

Amir Mehdi Yazdani
15 March 2017

Amir Mehdi Yazdani

Acknowledgements

Firstly, I would like to thank my advisor A/Prof. Karl Sammut for his guidance and support over the years of my PhD career. Without his guidance and dedication, this thesis would not have been possible. His pursuit of excellence in research and his constant desire to push the limits were very inspiring.

My special thanks goes to Prof. Oleg Yakimenko, my co-supervisor, who has devoted a great deal of attention to my work and directed me in a right path with his immense knowledge and experiences. He provided me with a wealth of insights and ideas that led to significant advances in my research.

My appreciation extends to Dr. Andrew Lammas, my co-supervisor, for his continuous attention to my research, his patience and unconditional support, and his valuable technical guidelines broadened my knowledge and horizon to this particular area of research. My gratitude also is devoted to Dr. Youhong Tang, my co-supervisor, for his ultimate encouragement to my PhD research.

I would like to thank to all my former and current colleagues in the Centre for Maritime Engineering, Control, and Imaging (CMECI) at the Flinders University. In particular, my sincere thanks goes to my colleague Somaiyeh MahmoudZadeh for her endless support, attention, inspiration, and encouragement during my PhD career. I would like to thank my former colleague Dr. Zheng Zeng for his encouragement, accompany, support, and friendship from early stages of my PhD career. In addition, I am thankful to my friends Asghar, Minh, Wenjin, and Rowan for the moments, stories, and experiences we had.

Last but not the least, I am eternally grateful to my family for supporting me spiritually throughout my life.

To my parents,
for their unconditional love and support…

Chapter 1

Introduction

1.1 Research Background

In recent years, oceanographic communities, oil and gas industries, and military services have shown significant attentions to employ unmanned systems for a diverse range of undersea missions. In general, unmanned undersea systems are categorized into two classes: tethered and untethered submersible vehicles [1]. Remotely operated vehicles (ROVs) are members of the former class while autonomous underwater vehicles (AUVs) belong to the latter one. These vehicles are able to perform various missions such as mapping, inspection, maintenance, detection, data collection, surveillance, and intervention [1-5].

ROVs are controlled remotely by human operators through a video link. They are usually connected to a mother ship via a communication and power supply tether. While ROVs are generally not limited in power supply, their range of operations are relatively constrained by the length of the tether. Besides, due to the nonlinear behavior of the tether, control of these vehicles are highly difficult [5].

AUVs, on the other hand, have the capability to undertake a broad range of underwater missions autonomously without the operator intervention. This feature makes AUVs potential candidates to perform tasks that ROVs are not able to carry out. Nowadays, different classes of AUVs varying in size and design configuration such as REMUS, Bluefin, Hugin, and Explorer (see Figure 1.1) are frequently used in underwater scientific research/military/commercial operations [6-9]. The operational capabilities of AUVs especially for long-term missions depends on:

- Mechanical design of the vehicle hull and propulsion systems;
- Vehicle payloads such as on-board computer, navigation sensors, and power supply package;
- Vehicle guidance, navigation and control (GNC) architecture.

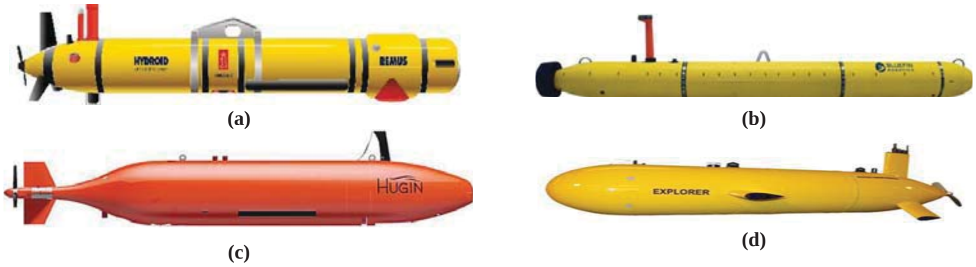


Figure 1.1. (a) REMUS-600 [6]; **(b)** Bluefin-12 [7]; **(c)** Hugin-1000 [8]; **(d)** Explorer [9].

In this regard, the on-board power supply is a key element that allows AUVs to be equipped with power-hungry sensors such as mapping sonar that impose a high hotel load (the total power used for all non-propulsion devices, i.e., computers, instrumentation, guidance, and communication devices). A larger power supply supports the propulsion system to operate for longer hours. Nowadays, significant breakthroughs in battery design, allows AUVs to utilize advanced sensor suites and payloads to perform more extensive missions [10]. However, because of constraints associated with cost and mechanical design, most AUVs commonly rely on cost-effective battery structures with limited capacity such as lithium ion, lithium polymer, nickel metal hydride, and silver zinc, which are not able to support the vehicle for a long time, and therefore long-term missions. Additionally, in persistence type AUV operations, the capacity to update missions and deliver previously collected data without requiring the vehicle to surface is highly beneficial.

Consequently, the question emerged: how do AUVs' persistence operations become possible?

1.2 Motivations

In the past and even in some recent oceanographic applications, persistent AUV operations have highly relied on supporting vessels and their technical crews to provide deployment, piloting and recovery tasks for AUVs. In these circumstances, the AUV autonomy level was constrained with much of the higher-level decision making processes made by operators of supporting vessels. Because of that, recently there have been growing attention to develop and leverage performance capabilities enabling AUVs to operate independent from supporting vehicles such as submarine, ships, autonomous surface vehicles and other AUVs [11-13]. More particularly, efforts have been devoted to boost the endurance capability of AUVs that is a key factor for long-term underwater operations. Of course, long-term operations are a function of hardware payloads particularly battery packages. There exists a series of vehicles with a high endurance operation capacity, categorized as long-range AUVs, such as Autosub Long Range and Tethys AUVs [3]. However, powering the medium and short range AUVs (and even long-rage AUVs for further operational endurance) by rechargeable batteries such as lithium ion, lithium polymer, and nickel metal that are limited in capacity, cannot conduct long-term operations of AUVs over many days to months. Besides, for any update of mission operations and tasks, it is required for AUVs to hover and then updated through a recovery process accomplished by human operators. In such a situation, also, human safety factors, when sea conditions become deteriorated, and cost of manned deployment and recovery to extend mission duration, must be severely considered. Thus, to circumvent mentioned limitations, taking advantage of hardware/ software equipment for AUVs' automating launch and recovery process are promising.

A docking system, is a submerged enabling technology which leverages the endurance capability of AUVs for long-term operations while reducing operation cost and hazards (see Figure 1.2). To this end, a docking station (DS) provides facilities for AUVs' battery recharge and mission data download and upload processes without needing to continuously recover AUVs to a ship. For instance, for AUVs with power consuming payloads like mapping sonar which usually should be recovered by ships on a daily schedule for servicing, a DS offers facilities for continuous use of high power consumption payloads while the vehicle remain deployed for extended mission periods. In addition, a DS is usually treated as a safe anchorage place once an AUV terminates the mission and to wait for retrieving mission commands. In this situation, AUVs can save accumulated mission data on a DS to protect information from possible risks of subsequent missions. It is become more attractive if the DS integrated on mooring or attached to cabled observatory networks, which allow the vehicle, become a part of the shore networks and provide connectivity advantages for direct data transferring/processing and vehicle monitoring from the shore [14-17].

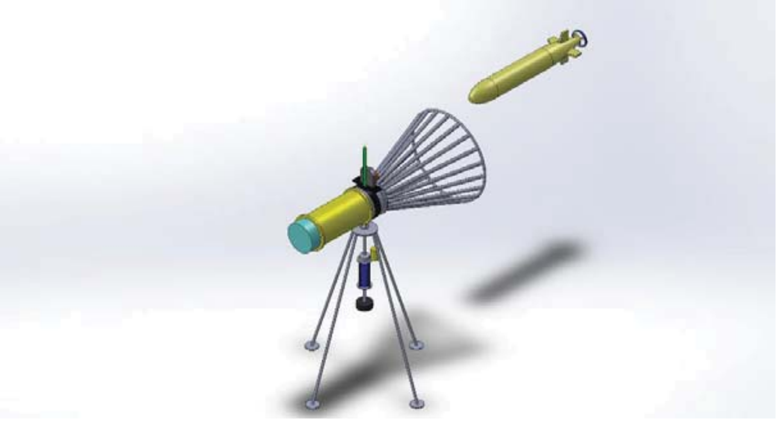


Figure 1.2. Pictorial representation of a DS and an AUV docking operation.

To benefit from the facilities a DS provides, an AUV should be able to perform a broad range of fully autonomously, efficient, and reliable docking maneuvers considering the constraints associated with operating environments, vehicles' dynamics, and DS geometry. The main motivation of this research is to attempt to extend the level of autonomy that an AUV required for such docking operations in practice. This leads to develop a systematic and universal guidance framework or equivalently a trajectory generator engine enabling an AUV to perform optimal homing/docking operations in diverse operating environments. The guidance system takes into account all related realistic constraints comprising dynamics of the vehicle's actuators, current disturbances, and no-fly zone (NFZ) areas, and supports the AUV docking maneuver with respect to a wide distribution of DS poses.

1.3 The Existing Challenges in Autonomous Docking Operations

While AUVs are capable of performing diverse autonomous tasks today, autonomous docking is still challenging and controversial. The documented research in the autonomous docking state-of-the art reflect these difficulties particularly limitations of the existing autonomous docking approaches in range and applications. Most experiments represent docking operations in a close vicinity of the DS including series of assumptions on constraints associated with the vehicle, DS geometry, and impact of current disturbance and obstacles. Meanwhile, there have not been much attention paid to support safe, stable and smooth transition/arrival of the vehicle into the DS. Consequently, this question still remains unanswered: *what are the necessary conditions to allow docking to become a reliable fully-autonomous operational capability*?

The level of AUV's autonomy for underwater missions and more specifically for autonomous underwater docking operations mainly depends on the performance of navigation, guidance and control systems. Navigation system is responsible for perceiving the surrounding environment and providing information about the DS pose. The docking guidance system, in general, is responsible for bringing an AUV from any arbitrary starting point into a destination that is the DS in our context. The control system is in charge of tracking reference commands generated by the guidance system. Usually, the navigation system and controller designed for diverse AUV missions can be exploited for docking operations as well, but this universality is not exactly the case with the guidance system. Regarding the demands of the docking process, the design of guidance system can comprise diverse levels of sophistication. These levels should address requisites associated with reliable and efficient docking operations.

Table 1.1 shows an example list of docking trials with the employed guidance systems, extracted from the literature of docking operations. There exist common features in the listed docking experiments as follows:

- Constant AUV forward velocity and acceleration;
- Fixed position and orientation of the DS;

- Applicable for the final docking stage that is terminal phase;
- The vehicle starts docking operations from front of the DS;
- Ignorance or passive compensation of current disturbances;
- No controllability for smooth and collision-free arrival into the DS cone;
- Do not consider energy expenditure/ time of docking operations.

With the clarity gained by hindsight from considering mentioned features, it is obvious that the reported guidance systems rely on a series of assumptions and simplifications; these guidance systems try to minimize the drift and miss distance during the terminal phase. Even though these approaches are relatively simple to implement, they are limited in that they usually work in a controlled operating environment under impact of negligible current disturbances, and operate based only on the geometric relationship and AUV's kinematics. Neither of these approaches can provide a closed-form solution assuring a collision-free unsaturated-control motion. Satisfaction of the terminal conditions is under the mercy of limiting the components of the final speed and acceleration, or a fixed-time arrival. These approaches might be useful at the very last stages of docking operations when an AUV is within reach and aligned with a funnel-shaped DS, but arriving to this point should use a different approach.

Table 1.1. Examples of AUV docking trials with a funnel-shaped station.

Vehicle Type	DS status	Guidance System	Environment	Reference
Odyssey llb	Non-stationary	Pure pursuit	Sea	[18]
REMUS	Fixed	Path following	Sea	[19]
REMUS	Fixed	Path following	Sea	[20]
Odyssey llb	Fixed	Pure pursuit	Sea	[21]
REMUS	Fixed	Path following	Sea	[22]
ISiMI	Fixed	Pure pursuit	Basin	[23]
Dorado	Fixed	Path following	Sea	[15]
ISiMI	Fixed	Linear terminal	Basin	[24]
KOLABOT	Fixed	Pure pursuit	Pool	[25]
REMUS	Non-stationary	Linear	Sea	[26]
HUG	Fixed	Pure pursuit	Pool	[27]

As a result, there exists lack of a systematic and universal docking guidance framework with capabilities to generate feasible and tractable trajectories providing a broad range of maneuverability for the AUV to travel into the DS considering different constraints, while minimizing the vehicle’s energy expenditure/operation time and allowing capacity to re-plan (update trajectory) when operating environment or objectives change. In practice, however, this level of autonomy has not been used in docking AUVs so far [17].

1.4 Research Objectives

It is important for a torpedo-shaped AUV to perform a wide spectrum of optimal, controllable, and safe docking maneuvers which are not restricted in range, dimension, environment, and applications. To this end, it is essential for the AUV to be equipped with a reliable and efficient docking guidance framework.

The main objectives of this thesis is to conduct a systematic study of the docking problem and to develop a universal and robust docking guidance framework enabling the AUV to perform a broad range of autonomous docking operations in a realistic sense.

From the operational standpoint, the guidance framework should satisfy a series of requirements; it should:

- provide a significant degree of freedom to control initial, midcourse, and final vehicle’s position, velocities, accelerations, etc.
- cover a vast distribution of the DS poses either static or time varying, no matter what is the initial pose of vehicle
- consider constraints associated with the DS geometry once the vehicle approaches the DS entrance
- provide smooth and stable departure from the starting point and arrival maneuvers of the vehicle into the DS
- consider the constraints associated with operating environments such as impact of current disturbances, obstacles, and NFZs

- consider the physical limitations over the states and actuators of vehicle
- optimize the vehicle energy expenditure, docking operation time, and/or any arbitrary performance index
- compute docking maneuvers in a real-time manner
- perform docking maneuvers' refinement (online maneuver computations) with respect to situational awareness of underlying environments.

In order to meet the main objective, this thesis commences by researching the problem of guidance system design for optimal docking operations in an offline mode. For this purpose, a new docking problem formulation in the context of the optimal control theory is proposed. In this regard, a geometric model of a funnel-shaped DS comprising arrival constraints, together with a high-fidelity model of the AUV that is being developed at the Flinders University, are utilized. Subsequently, the work is extended to address the problem of developing an efficient and real-time guidance system that can be operated in a closed-loop guidance configuration. This feature, enables the AUV to adapt and regenerate its trajectory during docking operations using continuously updated profiles of the operating environments and DS pose captured from the vehicle's on-board navigation sensors. Thus, autonomous docking operations associated with uncertainties, unmodeled dynamics, pose varying DS (e.g., moving/floating DS), and dynamic operating environments become practical.

1.5 Scope of Investigations

As mentioned in Section 1.3, the docking problem can be investigated from different perspectives. For instance, one investigation could consider the design and development of a particular navigational system to provide more accurate perception about position and orientation of both AUV and DS together with updates of current disturbance variations and obstacle movements. This thesis, however, is focused solely on the docking problem from the guidance perspective and tries to address the existing challenges of conventional docking guidance systems.

In Chapter 3, the integrated guidance framework problem is formulated with two assumptions. First, an offline map of operating environments together with the acoustic navigation system working based on the ultra-short baseline (USBL) positioning system are assumed to provide information about the DS pose, NFZ areas, and current vector field. It is further assumed that the USBL field of view can fully capture acoustic pings information received from the DS transponders, and that uncertainties associated with inaccuracy of the USBL positioning system and/or offline map are bounded. Second, regarding the dimension of operating environments, it is assumed that current vector fields are known, irrotational, and time-invariant.

In Chapters 4 and 5 different numerical approaches are used to address the AUV docking guidance with the aforementioned assumptions. Later on in the same chapters, the developed methods are also tested to prove that they are robust enough which allows lifting the aforementioned assumptions.

1.6 Statement of Contributions and Publications

This thesis addresses the problem of developing a systematic and universal guidance framework for a torpedo-shaped AUV intending to perform a broad spectrum of both offline and online homing/docking operations with respect to a funnel-shaped recovery station. For this purpose, this thesis establishes the foundation of the guidance framework on the context of optimal control theory. The main contributions of this thesis are threefold:

First, the docking problem is formulated using a high-fidelity model and a new approach in which the conventional docking problem is transcribed into a new form of a two-point boundary-value-problem (TPBVP). This problem formulation, practically, encapsulates a series of realistic constraints associated with the vehicle’s actuators dynamic, impact of current disturbance, and geometry of the DS while it takes into account minimization of energy expenditure or docking operation time. In this regard, the Pontryagin maximum principle (PMP) is utilized to analyze and synthesize the docking TPBVP.

Second, the proposed TPBVP is solved numerically using direct methods of optimal control theory – these methods comprise three distinct pseudospectral (PS) methods. While these methods have been extensively used in the aerospace realm of research, they have been rarely used in the underwater research areas and more particularly have not been used for the underwater docking problem. The PS methods used in this thesis are Legendre, Chebyshev, and hp-adaptive Radau methods and can provide benchmark solutions for the docking TPBVP. They offer spectral accuracy with exponential convergence rate for smooth optimal control problems. Additionally, they are tailored for finding optimal solutions very close to the true optimal solutions and offer a rigorous post-optimality facility as they estimate co-states and Hamiltonian trajectories. By using the PMP developed, the solutions obtained by the PS methods are analyzed and validated.

Third, a real-time solution generator is developed, that accounts for the design requirements and can be implemented on-board a real vehicle, using the inverse dynamics in the virtual domain (IDVD) method. The IDVD method, on the other hand, is capable of fast prototyping of near-optimal trajectories in a real-time scale as it exploits the concept of differential flatness to significantly reduce the dimension of optimization problem. It features smooth (differentiable) time histories of states and controls together with smooth departure and arrival of the vehicle to the DS and provides a viable structure for closed-loop docking configurations.

By using the PS and IDVD methods, which apply parametrization and then discretization on an optimal control problem, the docking TPBVP is transcribed into a nonlinear programming problem (NLP) and then by using commercial off-the-shelf NLP solvers the problem is solved. The consequence of this approach is to form a trajectory generator engine generating a set of trajectories that enables the AUV to perform a range of efficient and reliable docking maneuvers. The common features of this trajectory generator engine are as follows:

- Combine both homing and docking stages together to overcome the range limitation of the point-to-point docking guidance systems;
- Allow both 2- and 3-dimensional docking maneuvers as opposed to the conventional docking guidance mostly designed for 2-dimensional operations;
- Involve both dynamics and kinematics of the AUV as opposed to the conventional docking guidance approaches;

- Capable to handle the impact of current disturbances particularly cross-current disturbances;
- Enable the vehicle to perform docking operations in a cluttered environment with obstacles and/or no-fly zone areas;
- Allows smooth and stable arrival of the vehicle into the DS entrance;
- Able to satisfy the tolerance accuracy for a safe and successful operation in the terminal phase of docking operations;
- Generate feasible and tractable trajectories for on-board implementation on the vehicle;
- Able to cover minimum-energy , minimum-time, or simultaneous minimum time-energy docking operations;
- Robust to a reasonable bound of uncertainties associated with the vehicle's pose, DS pose, and current disturbance variations;
- Able to generate real-time trajectories for on-board implementation;
- Able to be implemented in a closed-loop configuration for online trajectory generation suitable for unknown or dynamic environments or against unforeseen changes and uncertainties.

The overall performance of proposed guidance framework and particularly the capabilities and main features of the direct methods employed as the trajectory generator engines, are investigated through a series of docking scenarios in operating environments comprised of realistic currents and NFZ areas with respect to a priori known and unknown poses of the DS. The feasibility and robustness of trajectories from the standpoint of their realization on a real AUV are verified using the software-in-the-loop simulation platform and Monte Carlo simulations.

The new guidance framework developed in this thesis contributes to the cause of improving AUVs' autonomy by enabling longer mission durations while assuring reliable and cost efficient docking operations.

The work developed in this thesis have been presented in the following international conferences and peer review journals:

- A comprehensive survey on the past, current, and future trends of docking guidance systems [P.1];
- A thorough study on the docking TPBVP , the IDVD and the PS methods' performances for both offline and online docking operations [P.2, P.3, P.4];
- Detailed analysis of the PS methods' performance for optimal docking operations including details on post-optimality process, tractability assessment using the software-in-the-loop simulation platform, and robustness analysis [P.4, P.5].

[P.1] A.M. Yazdani, K. Sammut, O.A. Yakimenko, A. Lammas, and Y. Tang, "A Survey on Underwater Docking Guidance System", ***ISA Transaction***, Under Review.

[P.2] A. M. Yazdani, K. Sammut, A. Lammas, and Y. Tang, "Real-time quasi-optimal trajectory planning for autonomous underwater docking", ***2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)***, 2015, pp. 15-20.

[P.3] A.M. Yazdani, K. Sammut, O. Yakimenko, A. Lammas, S. MahmoudZadeh, and Y. Tang, "Time and energy efficient trajectory generator for autonomous underwater vehicle docking operations", ***OCEANS 2016 MTS/IEEE Monterey***, 2016, pp. 1-7.

[P.4] A.M. Yazdani, K. Sammut, O.A. Yakimenko, A. Lammas, Y. Tang, and S. MahmoudZadeh, "IDVD-Based Trajectory Generator for Autonomous Underwater Docking Operations", ***Robotics and Autonomous Systems***, vol. 92, pp. 12–29, 2017.

[P.5] A.M. Yazdani, K. Sammut, O.A. Yakimenko, A. Lammas, and Y. Tang, "An Efficient Underwater Docking Guidance System Using the hp-Adaptive Radau Pseudospectral Method", ***Ocean Engineering***, Under Review.

Further papers have been written during the course of the candidature. These papers, are indirectly related to this thesis and are hence listed separately below:

[P.1] S. MahmoudZadeh, D. Powers, K. Sammut, and A.M. Yazdani, "Biogeography-based combinatorial strategy for efficient autonomous underwater vehicle motion planning and task-time management," ***Journal of Marine Science and Application***, vol. 15, pp. 463-477, 2016.

[P.2] S. M. Zadeh, D. M. Powers, K. Sammut, and A. M. Yazdani, "A Novel Versatile Architecture for Autonomous Underwater Vehicle's Motion Planning and Task Assignment," ***Soft Computing,*** pp.1-24, 2016.

[P.3] S. MahmoudZadeh, D. M. Powers, K. Sammut, and A.M.Yazdani, "Toward efficient task assignment and motion planning for large-scale underwater missions," ***International Journal of Advanced Robotic Systems***, vol. 13, pp.1-13, 2016.

[P.4] S. MahmoudZadeh, D. M. Powers, and A.M. Yazdani, "A novel efficient task-assign route planning method for AUV guidance in a dynamic cluttered environment," ***Evolutionary Computation (CEC)***, ***2016 IEEE Congress on***, 2016, pp. 678-684.

[P.5] S. Mahmoudzadeh, K. D. Powers, Sammut, A. Lammas, and A. M. Yazdani, "Optimal Route Planning with Prioritized Task Scheduling for AUV Missions," ***2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)***, 2015, pp. 7-14.

[P.6] S. M. Zadeh, A. M. Yazdani, K. Sammut, and D. M. Powers, "AUV rendezvous online path planning in a highly cluttered undersea environment using evolutionary algorithms," ***Applied Soft Computing***, Under Review.

1.7 Thesis Outline

The thesis is organized as follows. Chapter 1 provides a brief review of the existing AUVs for diverse underwater missions and then shows the exigency in using underwater docking systems. Detailed information and background relating to challenges of underwater docking operations and the need for a systematic and universal docking guidance system are also presented.

Chapter 2 considers the elements of docking operations in details and review the related works, particularly from the guidance point of view. Then it provides a thorough investigation about existing methods for docking trajectory optimization together with implementable closed-loop configurations for docking trajectory generation.

Chapter 3 constructs the foundation of this thesis. It first introduces the Flinders AUV, its specifications, and mathematical representation. This chapter then shows the use of Pontryagin maximum principle (PMP) and TPBVP formulation within two minim-energy and minimum-time docking scenarios to provide the unified framework for the underwater docking guidance design. Chapter 3 lastly introduces a high-fidelity software-in-the-loop simulation platform (SITLSP) to investigate the realization and tractability of trajectories generated.

Chapter 4 provides a comprehensive investigation on main features of PS methods in generating numerical optimal solutions for the docking TPBVP developed in Chapter 3. For minimum-energy docking operations, Chapter 4 investigates performances of various configurations of Legendre, Chebyshev, and hp-adaptive Radau PS methods in different operating environments. The post-optimality analysis is provided for the solutions achieved by the PS methods based on the information of co-states and Hamiltonian functions. The tractability of solutions is evaluated using the SITLSP. At the end, Chapter 4 presents robustness analysis of PS methods with respect to uncertainties associated with the vehicle pose, DS pose, and current disturbances.

Chapter 5 employs the IDVD method to solve the docking TPBVP and to address all objectives defined in Chapter 1. This chapter, at first describes detailed theoretical illustrations of the IDVD method. Then in the subsequent sections, Chapter 5 provides detailed discussions and analysis of the IDVD method's performance for both minimum-time and minimum-energy scenarios. The ultimate goal of developing the closed-loop guidance system or equivalently online trajectory generation for autonomous docking operations is addressed in the last section of Chapter 5. The effectiveness, tractability, and robustness of IDVD method for all possible docking scenarios, including offline and online applications, are proven based on the results of computer simulations, the SITLSP, and Monte Carlo trials.

Chapter 6 presents a summary and the overall conclusions obtained from the research and then concludes with recommendations and future direction of this research.

Chapter 2

Literature Review

This chapter presents a detailed investigation on required components for autonomous docking operations and then concentrates on the state-of-the art in underwater docking systems. It then directs the discussion to consider different aspects of developing guidance systems from the perspective of optimal control theory. The applicable numerical methods of optimal control theory for docking trajectory generation are introduced and the rationale to select the more appropriate methods are discussed. At the end, the idea for closed-loop docking guidance configuration is presented.

2.1 Elements of AUV Docking Systems and Related Works

AUVs are treated as highly complex systems with the significant potential to perform diverse ranges of underwater missions and tasks using maximum autonomy. Due to several practical difficulties, such as limited on-board power supply; limited memory storage capacity; and tedious, risky, and expensive human-based launch and recovery systems, AUVs are not yet being readily used for underwater tasks. Improvements in underwater docking technologies, obviate most if not all the practical difficulties that prevent persistent AUV operations. The docking system allows AUVs to recharge the on-board battery, upload the data collected during missions, download instructions to perform new missions, and consequently extend the vehicles' endurance capability. In the following, the main elements of docking systems are described.

2.1.1 Mechanical Configuration of Docking Stations

The chief philosophy behind design mechanism of docking structure is to simplify the docking operation. This, in general, comprises reduction of constraints on both approach direction and attitude provided by large horizontal and vertical capture apertures. In this regard, basically, two common forms of docking configurations namely unidirectional and omnidirectional have been developed [28], [29].

A. Unidirectional Docking Stations

The unidirectional DS is the most common configuration usually used for the torpedo-shaped AUV's recovery process. In this regard, funnel/cone-shaped stations are normally designed to provide a large cross section for the AUV capture mechanism. They robustly protect the AUV against biofouling or other environmental hazards when the latching phase completes. Besides, the funnel-shaped stations are able to support and minimize the number of additional payloads and hardware equipment that the AUV should carry on.

To perform docking with a funnel-shaped station, it is required for the vehicle to approach and accomplish the terminal phase of docking process along the axis of the funnel. This faces the vehicle with a constraint of knowing the orientation of the funnel to align itself with the centerline of the funnel at the final approach. Thus, an accurate navigational system is required. Figure 2.1 shows examples of funnel-shaped docking systems and their associated AUVs including REMUS [22], Eurodocker [30], Bluefin [31], and Dorado [15].

B. Omnidirectional Docking Stations

The omnidirectional docking stations are usually designed as pole dock platforms that are vertical structures composed of rigid poles or cables under tension, enabling the vehicle latching using particular latching devices attached on the nose of vehicle. In the pole-shaped DS, the vertical aperture is a function of the length of docking segment on the pole whereas the horizontal aperture is related to the width of the capture mechanism on the vehicle. This configuration is a truly omnidirectional station and gives the chance of docking an AUV by using a simple navigational system. It is applicable both in the region near the surface and close to the sea floor. Utilization of the pole-shaped DS is limited for a certain class of AUV since final latching to this structure needs a complex mechanical device implemented on the nose of the AUV. In addition, because of the moving parts of the pole dock and the fact that the attachment of the vehicle is not rigidly possible, assembling power and data connectors on the station is laborious [32], [33]. Figure 2.2 illustrates the pole-shaped docking station developed as part of the AOSN project for the Odyssey vehicle [34].

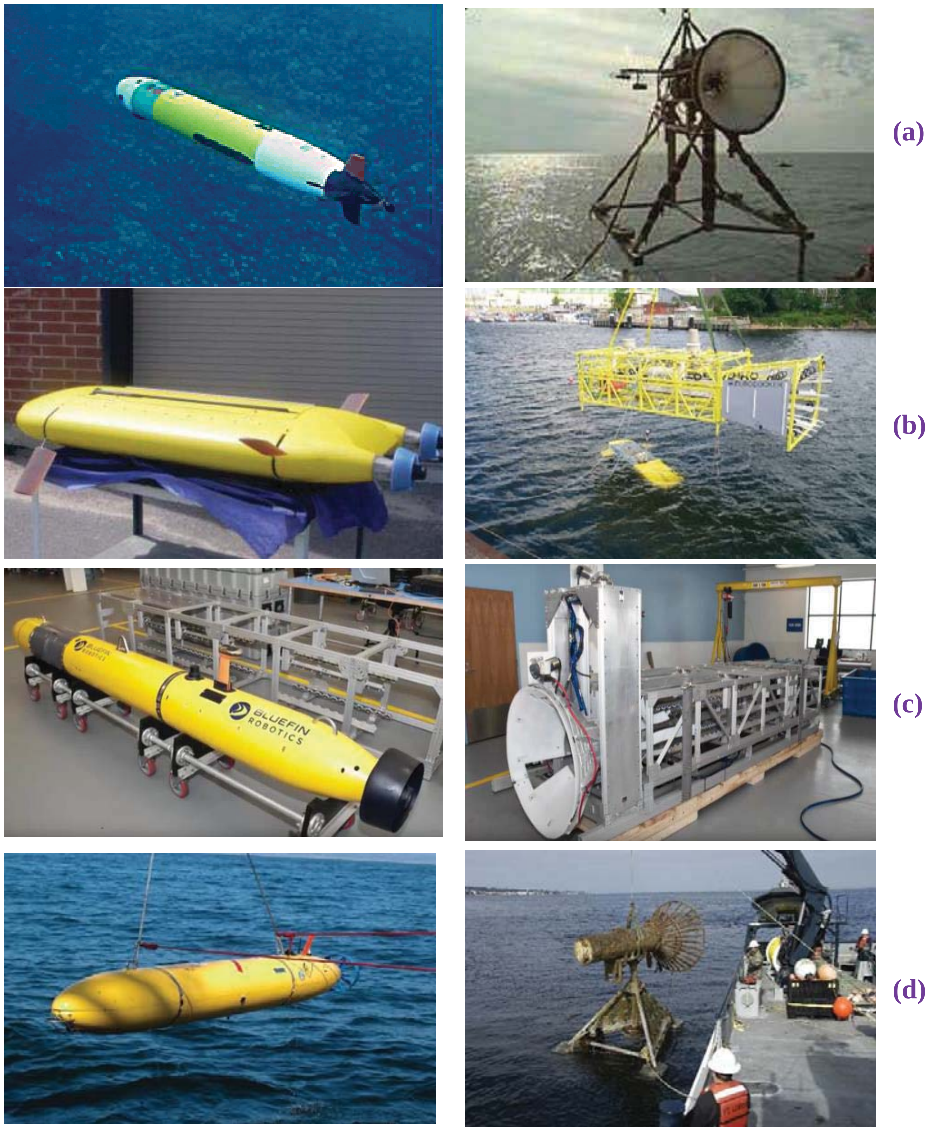


Figure 2.1. **(a)** REMUS docking system; **(b)** Eurodocker docking system; **(c)** Bluefin docking system; **(d)** Dorado docking system.

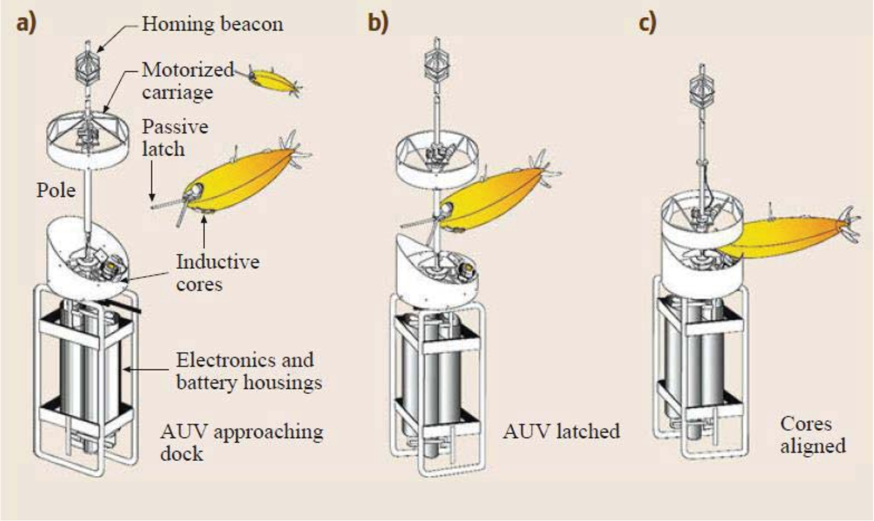


Figure 2.2. Three stages of the Odyssey IIb AUV docking process with the pole-shaped station [17].

2.1.2 Docking Navigational Systems

Docking process, basically, demands contacts between a vehicle and a DS in sequences of time intervals and therefore an accurate navigational system is required. The docking navigational system comprises navigational sensors installed and operate on the DS, providing relative position and orientation (that all together called pose) information of the DS to the vehicle equipped with related navigation sensors/systems. In general, three types of navigational systems are used for docking operations which are acoustic, optical, and electromagnetic systems.

A. Acoustic Docking System

Docking acoustic navigational systems operate based on propagation of acoustic signals over long distances between the DS and vehicle in an underwater environment. In docking operations, the common transponding configuration is employed in which the vehicle emits an interrogating ping and the DS sends a reply message. Then, the round trip travel time of the acoustic signal is used to determine the range index. For example, the ultra- short baseline (USBL) acoustic system operates based on the fact that a transponder (installed on the DS) receives the emitted acoustic signal into water and then sends a response for the transducer receivers (installed on the vehicle). The slant range, which is a distance between the transducer and transponder, is calculated based on the total signal traveling time and known speed of the sound in water. The target azimuth and elevation angles are determined with respect to the small differences of arrival time between the respective receivers. The USBL is a common and applicable navigation system for a wide spectrum of AUV docking operations [15, 20, 22, 34-36].

B. Optical Docking System

Optical docking systems are composed of multiple lights on the DS and a vison system on the AUV. The vision system can be a simple quadrant detector or a camera and associated image processing algorithms allowing more sophisticated docking operations. The relative position and orientation of the vehicle can be obtained either in a dock coordinate system or in other frames considering configuration of lights. This provides a certain flexibility for the vehicle in a situation that the approach phase should be accomplished from a particular direction. Compared to the acoustic systems, the acquisition range of optical systems is considerably limited in ocean environments. Besides, the performance of optical systems for docking is highly under impact of scattering and absorption of light in seawaters [37]. Nevertheless, there exist several documented studies employed optical systems for docking purposes [11, 13, 38, 39].

C. Electromagnetic Docking System

Unlike the optical docking system whose performance is subjected to a non-turbid and clear environment, the electromagnetic docking system is a robust and accurate navigation system under almost all oceanographic phenomena. This system consists of coils on the DS generating a magnetic field and coils on the vehicle sensing the docking signals. As a result, the bearing angle of the vehicle relative to the DS is provided. It was confirmed in a field trial experiment that accuracy of electromagnetic docking system is less than 20 cm but its operation is limited in the range of 25-30 m [21].

2.1.3 Docking Guidance Systems

More technically, the docking procedure can be divided into two major stages; the first stage is called *homing* in which the guidance system directs the vehicle from a far distance (approximately 1km~100 m) to a close vicinity of the DS (roughly 20 m~ 10 m). The second stage is named *docking* in which the guidance system drives the vehicle from the close vicinity of the DS to the DS entrance and finally latching between the vehicle and DS is performed. It is important that the guidance method is capable of handling both homing and docking stages properly. The following two categories of guidance methods are typically utilized for homing and docking procedures.

The first category contains *point-to-point* guidance laws that are usually simple in design and operate by pointing the vehicle directly into the DS. These methods normally are employed in the docking stage (vicinity of 20~10 m to the DS) and more suitable for docking operations that can be approached from any direction, for instance docking with a pole-shaped DS. The simplest of these are the classic guidance laws such as line-of-sight (LOS) [38], linear terminal guidance [39], and pure pursuit guidance [40].

The second category encompasses *trajectory generation*-based guidance laws. They are capable of being employed for both homing and docking stages. The raison d'etre for this group of guidance systems is ability to generate a geometric 3D path together with associated time histories of the vehicle's states such as velocity and acceleration commands to provide a broad range of maneuverability, unlike the point-to-point guidance systems, for AUV homing/docking operations. The major concern with these approaches, is computational burden that makes them less suitable for real-time docking operations. Clear examples of this group are AUV path planning using evolutionary algorithms [41], [42] and trajectory generation methods stem from optimal control theory [43], [44].

2.1.4 Review of the Related Works

Documented studies on employing possible space docking technologies for the underwater docking problem indicated that solutions to the underwater docking problem are still in infant stages when compared with aerospace counterparts [45]. The seminal works on the autonomous underwater docking date back to 1997 conducted by Oceanographic systems laboratory at Wood Hole Oceanographic Institute (WHOI) [30]. The REMUS AUV and the REMUS funnel-shaped DS were used to conduct the docking experiment. The experiment completely encapsulated all aspects of the docking problem such as mechanical design and configuration of the DS including charging and communication circuitry, the AUV navigation system, and the docking guidance system. The AUV docking maneuverer in this experiment was divided into following sequences; the first segment of the REMUS maneuver included straight motion and turning into a position 50 m away from the DS so that the AUV can point into the DS centerline. Then, the AUV attempted to follow the reference path along the DS centerline using a path following guidance law. When the AUV reached to the threshold of 2 m in front of the DS, the vehicle’s controller straightened out the angle of control fins and the vehicle continued with a constant thrust for 15 seconds to touch the DS’ cone center. This experiment used the USBL navigation system with tolerance accuracy of 1-3^o and supporting range of 2000 m. None of the AUV dynamic model, DS geometrical model, impact of current disturbances were considered in this experiment. To minimize the impact of the current disturbances, direction of the DS pointed into the direction of current disturbances. This experiment recorded success rate of 21% , however it was improved up to 91% due to the improvement of overall system in the next year’s experiments [20], [22].

The docking experiment coined as *optical terminal guidance* was conducted by the Odyssey IIB AUV equipped with a camera on the nose and a DS with a mounted light [18]. The pure pursuit guidance system directed the vehicle with respect to the source of light emitted from the DS. The decoupled horizontal and vertical control loops operated by PID controllers were used for the depth and heading control. This experiment reported 20-28 m acquisition range given the water turbidity rate. The docking experiment obtained an accuracy in order of 1 cm at the vehicle’s constant advance speed of 1-1.5 m/s. The effect of current disturbances was not considered, as current disturbances were negligible. The main drawback with the proposed approach was direct impact of the water turbidity on the range variations of the docking operation. Meanwhile, the proposed approach was susceptible to the sunlight in shallow waters.

A docking experiment conducted by the Odyssey AUV and a funnel-shaped DS equipped with an electromagnetic navigation system, was recorded [21]. In this experiment, the coils installed on the DS emitted horizontal and vertical magnetic fields and mutually the receiver coils on the vehicle detected the magnetic fields. The pure pursuit guidance system and a decoupled PID control loop directed and controlled the vehicle maneuver to the DS, respectively. The experiment was conducted in way that the AUV performed a 60 s straight maneuver ended with an 180° turn from the launch location in order to point into the DS centerline. The AUV continued to travel into the DS using dead reckoning until it reached the active range of the magnetic fields. A precision up to 20 cm to the DS was achieved by the proposed docking system. However, the operating range of the system was limited to 25-30 m. The failure also was recorded when there was more than 30° discrepancy in AUV alignment with the DS centerline.

Unlike the previous unidirectional docking experiments, an omnidirectional docking trial was conducted using a pole-shaped DS [34]. The authors of this study claimed that their approach could be tailored for both homing and docking stages using simple LOS guidance laws. The LOS guidance system worked based on nullifying the bearing to the DS calculated upon the information (azimuth and elevation angles) obtained by an USBL navigation system. An inner-outer loop PID was used to enable the AUV to track the reference heading angle. The authors mentioned about the versatility of their approach as the homing stage could be performed independent of initial bearing of the DS to the vehicle. To this end, they used a technique to distinguish the miss docking occasions and then it could provide a chance for the AUV to retry the homing with a higher possibility of success. They developed a layer-based hierarchical control architecture using a high-level finite-state machine (FSM) model to monitor and supervise the autonomous docking process. The work presented for the omnidirectional AUV docking was interesting as the authors practically covered every aspect of the docking problem. The major disadvantage with the proposed docking system was the mechanical complexity in design and establishment of the DS and the need to install a mechanical latching tool on the nose of vehicle for passive latching with the pole.

The Monterey Bay Aquarium Research Institute (MBARI) conducted a docking experiment using a Dorado/Bluefin 54-cm-diameter type AUV and a companion funnel-shaped DS in an open sea environment [15]. The docking approach in this experiment encapsulated both homing and docking stages in consecutive sequences as follows:

1) Locate and home to the DS;

2) Compute a position fix;

3) Travel to start of the final approach path;

4) Execute final approach;

5) Latching process.

In this experiment, the Dorado AUV used USBL and Doppler velocity log (DVL) as the navigation system and the pure pursuit law for the guidance system in the homing stage. The path following scheme using PID controller was adopted in the docking stage in which the PID tried to keep the vehicle traveling with constant speed of 1 m/s along the centerline of the DS by minimizing the cross-track between the reference and actual heading angles. Neither the pure pursuit nor the path following guidance systems did consider the compensation of the current disturbance and to minimize the impact of this phenomenon the DS direction was set.

A control solution for AUV homing and docking sequences was proposed [46], [47]. It worked based on the conventional artificial potential filed (APF) method for path generation supplied with the long baseline(LBL) navigation sensor, the LOS guidance law for computing the reference heading, and a decoupled sliding mode controller (SMC) for the vehicle heading and depth control. A valid reduced dynamic model of the AUV used in this study considering passive control of the roll direction. The results of simulation in the MATLAB environment indicated the effectiveness of the proposed docking solution.

A primary work on docking experiment based on a visual navigation system accompanied with an image processing algorithm was conducted to direct the ISiMI AUV in the terminal phase of docking stage into a stationary funnel-shaped DS equipped with five lights [23].The pure pursuit guidance system was used to generate reference heading angle. The decoupled PID control loop was employed to track the reference heading and depth parameters. The compensation for current disturbances was not considered in this study. Later they extended the work by introducing a modified linear terminal guidance system applied a sideslip angle for AUV approaching to the DS in presence of ocean currents [23, 24, 39, 48]. In these studies, the authors assumed a priori known pose of the DS; they further used both known uniform distributed current disturbances and time-varying current fields monitored by an ocean current observer. They introduced a set of constraints regarding the DS geometry to evaluate the capability of their approach in the final sequence of the docking process. The chief philosophy in these studies was to generate an intentional cross-track error to compensate the impact of current disturbances and then to compute the reference heading angle enabling the vehicle to align itself with the DS centerline in the terminal phase of docking process.

A two-step docking approach based on the integration of guidance and control strategies for an intervention type AUV was developed [49]. The control law at the first step was derived by assuming an underactuated dynamic model for the vehicle operating at higher speed. Then, it provided the AUV steering maneuver toward the final docking path led to zero convergence for the errors associated with almost all initial conditions. In the second step in which the vehicle operations were performed at lower speed and used for final approaching to the DS, a fully actuated dynamic model was assumed to provide more efficient thrust to generate transversal force and also more accurate docking performance. In this stage, an adaptive control law was proposed to account for uncertainty in hydrodynamic parameters and to guarantee desired accuracy for the vehicle docking profile. This two-step strategy was developed for the horizontal plane (2D) without considering direct impact of current disturbances. The authors however claimed the potential of the proposed approach to extend for 3D docking operations.

A three-layer hierarchical control architecture was developed for a docking problem with respect to a mobile DS that was mounted on a ROV [50]. The higher level employed hybrid automata to supervise the whole docking operation. This included interacting discrete events and continuous states represented by nonlinear ordinary differential equations. The supervisor layer also was in charge of switching the maneuvers. The medium level was responsible to derive the necessary reference state, path following, and the roundabout maneuver. Potential fields were also used for obstacle avoidance system. The lower level controller used sliding mode theory to support reference tracking in the presence of parametric uncertainties and external disturbances. The simulation results in MATLAB environment showed the effectiveness of proposed controller.

More recently, a control law based on range-only measurements was developed for robust homing of the MARES AUV to a beacon [51]. The proposed approach intended to minimize both computational complexity and sensor/equipment requirements. The objective behind this approach was not to use any state estimation technique to drive the vehicle into a small neighborhood of the beacon position. A guidance law emanated from Lyapunov theory was derived to direct the AUV toward the beacon without requiring initialization. The results of the homing experiment conducted in the Douro River demonstrated the asymptotical convergence of the vehicle to the reference beacon.

Similarly, the Lyapunov stability theory was utilized for designing a guidance controller for docking operations that were performed in the pool lab of Harbin Engineering University [52]. The visual positioning system was used in this study to support the guidance controller. The proposed guidance system showed capability to generate the reference heading and crabbing angle to compensate for horizontal and vertical deviation during pool testing of docking operations. The idea was to align the vehicle with the centerline of the DS. The proposed docking approached was applied for the terminal phase of the docking process. The explicit impact of current disturbances was not considered in this study. The experimental results showed that the system achieved 80% successful docking rate.

A study on developing guidance system for a hybrid underwater glider to perform docking operation with respect to a stationary funnel-shaped DS was presented [27]. Unlike the previous docking experiments, in this study the DS was an active structure with actuators to adjust the DS direction for the final vehicle approaching. The authors claimed the benefits of this active structure that allowed the vehicle to approach the DS from most range of directions. Additionally, the active DS was equipped with an acoustic Doppler current profilers (ADCP) sensor to measure and then send the current information to the vehicle guidance system in order to compensate the current disturbance impact. In this study, both USBL and visual navigation systems were employed. A modified pure pursuit guidance with current compensation and tailored for the terminal phase of docking process was designed. The guidance system just concentrated to direct the vehicle at the dock position as exact as possible and the suitable heading for the final sequence of docking was provided by the rotatable DS. The experiment was conducted in a swimming pool and the results indicated the feasibility and effectiveness of the proposed cooperative docking approach.

As a final example in this section, it is mentioned to the development of a hybrid docking guidance system enabling the AUV to dock with a funnel-shaped DS in the presence of cross-currents without applying a crab angle [53]. In this study, the terminal docking phase is divided into *approaching* and *sliding* paths. During the approaching path, the vehicle was supported by an integral line-of-sight guidance (ILOS) law to follow a straight path parallel to the DS centerline. Once the vehicle reached to the sliding path, an integration of the ILOS and speed regulated guidance (SRG) laws directed the vehicle to the DS entrance without using a crab angle. The proposed guidance system derived based on a simple AUV kinematic model in the horizontal plane. It provided the vehicle heading parallel to the DS entrance together with desired surge speed in the arrival to the DS, in the presence of cross-currents.

By considering the common features in the mentioned studies, it is obvious that most of the proposed guidance systems are derived based on a series of assumptions and simplifications. These guidance systems try to minimize the drift and miss distance during the terminal phase. Even though these approaches are relatively simple to implement, they are limited in range and applications. In other words, they are limited in that they usually work in a controlled operating environment where under impact of no or negligible current disturbances, and operate based only on the geometric relationship and AUV's kinematics. Neither of these approaches can provide a closed-form solution assuring a collision-free unsaturated-control motion. Satisfaction of the terminal conditions is under the mercy of limiting the components of the final speed and acceleration, or a fixed-time arrival. These approaches might be useful at the very last stages of docking operations when an AUV is within reach and aligned with a funnel-shaped DS, but arriving to this point should use a different approach.

2.2 Optimal Control Theory

As indicated in the previous section, a series of studies on underwater vehicle docking have been published. However, studies on the optimal control of underwater vehicles and particularly autonomous underwater docking operations have been rare. Compared to the aerospace realm of research, the optimal control theory is still a very underdeveloped tool in underwater research areas and there exist relatively few research works in which the principles of optimal control theory have been adopted. For instance, in [54] time and energy efficient trajectory planning and collision avoidance using optimal control framework is developed and a numerical solution is provided by the nonlinear programming approach. In [43], an efficient trajectory is generated based on developing a control strategy that minimizes the energy consumption of an underwater vehicle along the desired path. The limitations of thrusters are taken into account to ensure implementable trajectories on a real vehicle. The resulting optimal solution is achieved by a significant reduction of multiple rapid switching of the thrusters. In [44], an analytical time optimal trajectory solution for depth control purpose is achieved in a closed-form using explicit second-order differential equations of depth motion. Depending on the robust behavior of a tracking controller, the proposed trajectory can provide the shortest maneuvering time for the proposed underwater vehicle. This solution, in essence, is suitable for maneuvering the vehicle over a relatively short distance in an operating environment. In [55], energy minimization trajectory planning for a stable underwater vehicle using analytically derived relationships between energy consumption and number of thrusters is developed and eventually in [56], the optimal control framework is applied to derive the time optimal trajectories for a fully actuated underwater vehicle subject to constraint on input force.

The optimal control theory is indeed an influential tool for design, synthesis and analysis of complex nonlinear systems. It is able to provide a universal framework for developing a reliable and efficient trajectory optimization (generation) engine or equivalently guidance system. The goal of optimal control theory in general is to optimize, either minimize or maximize, a specified performance index (or cost/objective function) which is subjected to a series of constraints such as system dynamics, boundary conditions (endpoint constraints), path constraints, box constraints (including bounds on states and controls), and linkage constraints (known as phase continuity constraints). In the context of trajectory optimization (generation), it is desired to generate a trajectory (solution) enabling a system (in our context an AUV) to transverse from the starting point to the destination point while all physical (vehicular) and environmental constraints are taken into account and a specified performance index (time, energy, time-energy, etc.) is minimized. However, obtaining an analytical optimal solution for most aerospace and underwater guidance problems is typically very difficult if not impossible. Instead, an optimal control problem (OCP) is usually being solved numerically.

Indirect methods, offer numerical solutions for an OCP, using the calculus of variations and PMP to construct the first-order optimality conditions and then converting an original OCP into a Hamiltonian boundary-value problem (HBVP) [57], [58]. The term indirect stems from the fact that numerical solutions are indirectly obtained by solving the equivalent HBVP of an OCP. Indirect methods are able to provide the most accurate solutions for an OCP while ensuring the first-order optimality conditions.

Indirect shooting and collocation methods are the most common branches of indirect methods. In the indirect shooting method, at first initial guesses are provided for the unknown boundary conditions at one end of the interval of the underlying OCP. Using the proposed guesses together with the known initial conditions, the HBVP is integrated either in a forward or backward manners to the other end of the interval. The resulting terminal conditions, then, are compared with the known terminal conditions, which are derived from first-order necessary conditions, and the difference with respect to the accuracy tolerance is calculated. In a case that the obtained terminal conditions are not within the maximum tolerance, it is required to adjust the unknown initial conditions and then repeat the integration process until the tolerance accuracy is met.

The indirect shooting method has shown considerable numerical difficulties in dealing with *hypersensitive* OCPs and that are mainly due to ill-conditioning of the Hamiltonian dynamics [59-61]. This ill-conditioning stems from the fact that the divergence of the flow of trajectories should be zero and by performing the integration process in either direction of time, errors due to the unknown boundary conditions are amplified [58]. To alleviate the mentioned difficulties, the indirect multiple-shooting methods has been developed [62]. The chief philosophy of this method is to divide the time interval $[t_0,t_f]$ into smaller intervals and then apply the shooting method to each individual time interval. This approach is accompanied with the continuity conditions to enforce the continuity at the interface of each subinterval. By employing the indirect multiple-shooting method, the sensitivity issues of the indirect shooting method are addressed to a considerable extent; however, as a result of adding extra variables (values of states and adjoints at interior intervals), the size of the problem is increased that impose a computational cost. Additionally, if a proper set of initial guesses of the co-states is not provided, the indirect multiple-shooting method can also face difficulties.

The indirect collocation method is another branch of indirect methods' family in which an OCP is solved based on the state and control parametrization techniques. For this purpose, usually piecewise polynomials are employed and the OCP is transcribed into a root-finding problem in which unknown coefficients of the polynomials are decision variables. Then using an appropriate root-finding technique such as Newton method the new system is solved. The indirect collocation method is suitable to be applied on multipoint boundary value problems, for instance simple trajectory optimization problems. However, like the previous indirect methods, they cannot solely be used without solving the costate differential equations.

In essence, while indirect methods provide the most accurate numerical solution for OCPs, however, they are more often than not impractical in dealing with constrained OCP. Firstly, analytical derivations of cost function and constraints are required to derive a HBVP, that is heart of indirect methods. This process is highly cumbersome and tedious for complicated problems. Secondly, finding an accurate solution for a constrained HBVP is not an easy task and is associated with a series of practical difficulties. For instance, the initial guess of co-state variables must be properly provided to start the iterative methods. With respect to the non-physical nature of co-states, it is cumbersome to determine a suitable initial guess and an improper one can result in a non-converged or non-optimal solution. There is also a possibility that even relying on a reasonable guess, the adjoint equations turn out to be an ill-conditioned case. Finally, an insight into the switching functions of problems with active path constraints is required [58], [63]. Therefore, direct methods are used to circumvent these difficulties.

Therefore, in the subsequent section, principles, pros and cons of direct methods are reviewed to obtain insight and ultimately show the rationales for design and developing a systematic and universal docking guidance framework using popular direct methods. This approach is new and in the state-of-the-art of autonomous underwater docking operations, there is no similar work to the best of the author knowledge.

2.3 Numerical Solutions Using Direct Methods

Direct methods, are basically, different in nature with indirect methods as they transcribe the original OCP into an NLP and then directly approximate states and/or controls in an appropriate manner. The proposed NLP can be solved either by gradient-based techniques such as sequential quadratic programming or by evolutionary-based techniques, for instance, genetic and simulated annealing algorithms [58].

In general, direct methods are categorized into two major branches; the first one is called *control parametrization method* in which controls are parametrized using specific functions and the state equations are kept untouched. Whereas, in the second one called *control-state parametrization method* both states and controls are parametrized simultaneously based on particular basis functions in which the unknown coefficients are considered as NLP variables.

In the following, the most popular and applicable direct methods employed for trajectory optimization problems are briefly reviewed.

2.3.1 Direct Shooting Methods

The direct shooting method is a control parametrization method and is established upon an implicit integration of a trajectory optimization problem. In this method, controls are computed based on piecewise, linear, or polynomial approximations with unknown coefficients in a series of small time intervals. The states with unknown initial values are also treated as optimization variables. The constraints associated with system dynamics are satisfied by using time-marching integration algorithms. The cost function, additionally, is approximated using a quadrature rule corresponding to the numerical integrator applied for system dynamics' constraints. In other words, the NLP arising from the direct shooting method optimizes the cost function subjected to possible path and interior-point constraints. With an iterative optimization process, solutions for the optimization variables (NLP variables) related to controls and the initial values of states with unknown initial values are obtained.

Compared to the indirect shooting method, the direct shooting method does not require for extraction and implementation of the analytical process of the first-order necessary conditions and co-states definition. This great advantage nominates the direct shooting method as an easy-to-use candidate for users. This feature, also, allows developing a universal computer coding of this method as implemented in POST [64] and GTS [65] software.

There exist two major drawbacks with the direct shooting method; the first is the need for implicit integration to evaluate the gradient of NLP variables in each iteration that imposes a high computational cost; and the second is practical difficulties in solving a large NLP and particularly sensitivity to unknown initial conditions.

The direct multiple-shooting method is employed to alleviate the sensitivity issue of the direct shooting method to unknown initial conditions [58]. In this method, in a manner similar to that of indirect multiple-shooting method, the time interval $[t_0 ,t_f]$ is divided into several subintervals, and as a result a more detailed NLP is produced. In this regard, the NLP variables comprise the unknown coefficients in the control parametrization and the values of states at the beginning of each time subinterval. While this formulation yields a much larger NLP compared to that of in a simple direct shooting method, however it shows a direct impact on mitigating the sensitivity issue. This is because of the fact that integration is applied over smaller intervals as opposed to one large interval in the direct shooting method.

2.3.2 Direct Collocation Methods

Direct collocation methods are common and applicable state and control parametrization techniques in which states and controls are approximated and represented within and by a set of discrete number of variables [57], [66]. In fact, by employing direct collocation methods, differential/integral constraints of a system are transformed into algebraic constraints and consequently an infinite continuous OCP is transcribed into a finite NLP. To do so, first, the time interval $[t_0 ,t_f]$ is subdivide into a number of small intervals based on a set of equally or unequally spaced collocation nodes upon which both states and controls are discretized. Then using polynomials - usually low-order polynomials such as piecewise cubic Hermite, or cubic Spline- the discretized states and controls are approximated. In this approach, the polynomial coefficients are treated as NLP variables and the proposed NLP can be solved using efficient and commonly used NLP solvers such as sparse nonlinear optimizer (SNOPT)

[67], SPRNLP [68], KNITRO [69], and interior point optimizer (IPOPT) [70].

The proposed collocation method is more specifically called *local collocation* in the literature, as it has shown local support along with numerical robustness to the initial guess but with a low algebraic convergence rate as opposed to *global collocation* methods with spectral accuracy. Besides, the approach local collocation methods are adopted to solve an OCP, results in a very large NLP problem. This is problematic once real-time trajectory optimization is desired [57, 58, 71]. In order to significantly reduce the NLP size emerged from direct collocation methods, the differential flatness theory has been employed.

The differential flatness theory was first introduced in the context of differential algebraic property of nonlinear systems [72]. Since then, more attention has been devoted to this theory for trajectory generation of nonlinear systems [73-78]. In this regard, the salient point is to map a trajectory generation problem into a space called flat space that conserves the nonlinear dynamic characteristics of systems and allows representation of system's states and controls in terms of functions of flat outputs and their derivatives. Mathematically speaking, a general nonlinear system

$$\dot{\mathrm{X}} = f(\mathrm{X},\mathrm{U}) \;\; , \mathrm{X} \in \Re^{\mathrm{n}} , \, \mathrm{U} \in \Re^{\mathrm{m}} \qquad (2\text{-}1)$$

is said to be differentially flat if there exists a flat output vector of a form

$$\upsilon = h(\mathrm{X},\mathrm{U},\dot{\mathrm{U}},...,\mathrm{U}^{(p)}) \qquad (2\text{-}2)$$

such that the states and controls can be determined from equations of a form

$$(\mathrm{X},\mathrm{U}) = \hbar(\upsilon,\dot{\upsilon},...,\upsilon^{(l)}) \qquad (2\text{-}3)$$

The flat outputs in (2-2) transfer the original nonlinear system (2-1) into a new space called flat space. It means that optimal solutions taking into account boundary conditions and path constraints are achieved in the flat space and then lifted back to the original state space as shown in Figure 2.3. By using this property, the constraints associated with the system dynamics are completely removed (or alleviated to a great extent in the case of partially flat systems). As a result, the dimension of optimization problem is reduced significantly as a few number of variables are participants; and consequently, real-time computations become possible.

Numerous studies have been conducted by researchers to set necessary and sufficient conditions for system's differential flatness [79-81]. For fully differentially flat systems, no dynamic constrains remain in the transformed OCP as the system's dynamic information is implicitly encapsulated by the flat outputs. This property results in dealing with just algebraic constraints rather than dynamic constraints in trajectory generation problems and consequently converting it into a simple algebraic interpolation/collocation problem with a great computational efficiency [82]. While it has been proven that simplified dynamic models of aircrafts, helicopters, quadrotors, towed vehicles, trailers, remotely operated vehicles (ROV), and AUVs are differentially flat, however, more realistic vehicle models are in general non-differentially flat [83-90]. Even in this circumstance, the concept of differential flatness can be employed to transfer the underlying system into the possible lowest dimensional space and adding the remaining dynamic as constraints to the OCP.

In proper dynamical systems particularly unmanned vehicles, determining flat outputs, is typically based on a trial-and-error process. The spatial coordinates, are usually good candidate for flat outputs as by having vehicle's physical path, the rest of states and control variables can be computed. In essence, there is no need to develop a particular algorithm to determine the flat outputs while one uses the concept of differential flatness and relying on spatial coordinates, most of the time, are beneficial.

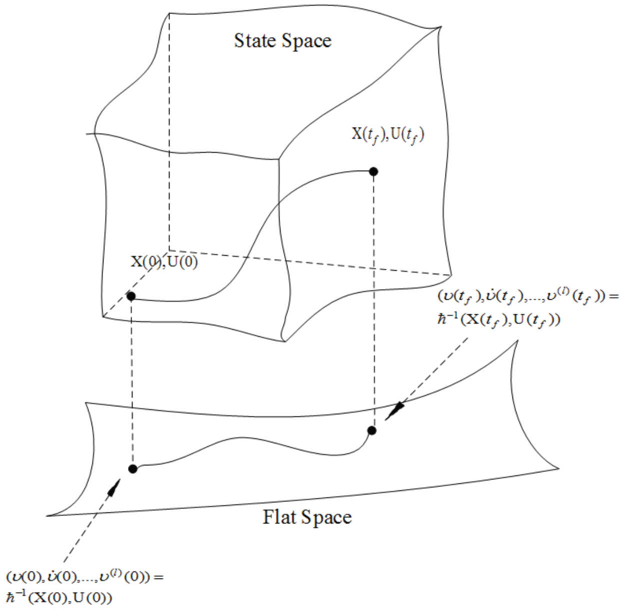

State Space
$X(t_f),U(t_f)$
$X(0),U(0)$
$(\upsilon(t_f),\dot{\upsilon}(t_f),...,\upsilon^{(l)}(t_f)) =$
$\hbar^{-1}(X(t_f),U(t_f))$
Flat Space
$(\upsilon(0),\dot{\upsilon}(0),...,\upsilon^{(l)}(0)) =$
$\hbar^{-1}(X(0),U(0))$

Figure 2.3 Transformation between the state space and flat space based on flat outputs; note that the map q is bijective [75].

2.3.3 Pseudospectral Methods

Pseudospectral (PS) methods in the early years of development were employed for solving computational fluid dynamics (CFD) problems [91], [92]. Since 1990, the optimal control community has devoted a great deal of attention to PS methods for solving a wide range of optimal control problems [93], [94]. Particularly, a plethora of aerospace trajectory optimization using PS methods have been reported in the literature [95-99], but these approaches have been seldom employed in the realm of underwater trajectory optimization problems [100-103].

The foundation of PS methods can be described in a way that the approximation of states and controls are achieved by using a finite basis of an orthogonal polynomial, like the Chebyshev or the Lagrange polynomial, at a set of discretization points. These points, called collocation points, are normally selected as roots of the of Legendre polynomials (called Legendre-Gauss or LG nodes), roots of Chebyshev polynomials (called Chebyshev-Gauss or CG nodes), extrema of Legendre polynomials (called Legendre-Gauss-Lobatto or LGL nodes), extrema of Chebyshev polynomials (called Chebyshev-Gauss-Lobatto or CGL nodes), and roots of linear combinations of

Legendre polynomials (called Legendre-Gauss Radau or LGR nodes). Unlike local collocation methods using the equally-spaced nodes, the PS methods take advantage of unequally-spaced distributions of orthogonal nodes (global collocation) in the time domain. This property provides higher accuracy of interpolation functions relying on a fewer number of nodes, as opposed to local collocation methods.

The approach PS methods adopt to transcribe an OCP to an equivalent NLP is as follows. The defect constraints, for approximating the dynamics of the system, are defined by taking derivatives of the interpolant polynomial and then setting them to be equal to the right-hand-side dynamic equations at all or a sub-set of collocation points. The path and boundary constraints are imposed on the intermediate and terminal collocation points, respectively. Finally, the integral cost function is approximated by means of highly accurate Gaussian quadrature rule [58, 71, 97]. For mainstream PS methods such as Gauss, Legendre, Chebyshev and Radau there exists a viable transformation between the Karush–Kuhn–Tucker (KKT) conditions of NLP and the continuous differential equations of co-states. Unlike the practical difficulties of indirect methods to estimate co-states trajectories, PS methods are able to efficiently provide co-states information. Having co-states information, the post-optimality process that is an investigation on the feasibility/optimality of solutions, becomes possible. Using PS methods for estimating costates is indeed useful as in most cases a good guess to the co-states is not available when indirect methods are employed [58, 71].

Using the orthogonal collocation points enables PS methods to show spectral convergence at an exponential rate for a group of smooth OCPs. However, this is not the fact in dealing with non-smooth OCPs, as PS methods show extremely slow convergence rate (that is normally involved with high computational burden), even by employing a high-degree basis functions and in some cases converging to feasible solutions is not possible [104-106]. To accommodate these difficulties, and to simultaneously improve accuracy and computational efficiency of PS methods, adaptive mesh refinement approaches have been developed [107], [108].

2.3.4 Inverse Dynamics in the Virtual Domain Method

The inverse dynamics in the virtual domain (IDVD) method is another branch of direct methods that exploits the concept of differential flatness to significantly reduce the dimension of optimization problem and thus enables fast prototyping of feasible trajectories [109], [110]. The trajectory computations occur in the output space instead of the control space. The reference functions are normally combinations of any function such as orthogonal, monomial, and trigonometric. Then, by exploiting the inverse dynamics, state and control vectors are represented as functions of the output. The IDVD method performs optimization in a virtual domain as opposed to time domain that is essentially beneficial for decoupling space and time parametrizations. Comparing to PS methods, the real-time version of IDVD method does not exploit co-states estimation in generating optimal solutions and therefore trajectories generated are of a near-optimal form [111]. However, unlike other direct methods which require extensive computational power because of using many optimization variables (large NLP size), the IDVD method uses a few number of decision parameters (usually less than 10), enabling this method to be implemented in an online form within a closed-loop configuration. Additionally, the IDVD method is also easy to modify and code, offering suitable flexibility to operators to adjust it with respect to mission scenarios.

The effectiveness and computational tractability of the IDVD method have been verified in different realms of research areas. For instance, AUVs rendezvous and path planning [90], [112], collision-free trajectory planning of quadcopters [113], experimental implementation of the planar maneuver of a chaser spacecraft docking with a rotating target [114], real-time trajectory optimization of an unmanned combat aerial vehicle (UCAV) performing agile air-to-surface (A/S) attack [115], terminal guidance of autonomous parafoils [116], and minimum-time aircraft maneuvers [117]. In all abovementioned studies, several important properties are indicated for suitability of the IDVD method that are:

- The boundary conditions comprising higher-order derivative terms are satisfied a priori;

- The controls generated are smooth and physically realizable;
- The method is not sensitive to initial guesses and a set of intuitively generated initial guesses is applicable;
- Fast convergence rate is provided as the optimization routine uses a few decision parameters.

It has been mentioned in the literature that the computational speed of the IDVD is more than an order of magnitude faster than for example PS methods, at a small loss of optimality[115], [117]. Thus, it becomes a good candidate to mitigate disturbances and unmodeled dynamics by multiple times updating trajectory during missions.

2.4 Closed-loop Solution for Trajectory Generation

More often than not, solutions generated by direct methods are considered as open loop solutions since they just depend on time but not directly on state variables. To build up a state-dependent guidance law, the concept of online trajectory generation (optimization) should be used.

The ultimate goal of the online trajectory generation is to provide a set of trajectories for a vehicle during the mission not a priori, unlike offline applications, but based on the situational awareness of the operation environment, and updates of the vehicle mission objectives. In this regard, the vehicle does not follow a pre-determined trajectory or path but indeed it computes and tracks an optimized trajectory considering variations of a mission. In these circumstances, the optimized trajectory is not just generated based on the mathematical representations of the vehicle's model and pre-determined boundary conditions but also considers the impact of uncertainties.

A series of attempts has been performed to equip unmanned aerial and underwater vehicles with diverse online path/trajectory generation systems. For instance, dynamic path planning using hierarchical approach was developed for underwater vehicles operating in smoothly variable ocean currents [118], [119]. In these studies, the search space is decomposed into several levels of resolution, in that the higher resolution is assigned to immediate current position of the vehicle, and the re-planning system generates an optimal path from the current position of the vehicle to the target of interest. This approach is applicable in situations that the rate of changes is slow and smooth but not suitable in dynamic environments where current vector fields are continuously varying, as it is computationally expensive regarding the decomposition process. In [120], [121] another dynamic re-planning scheme was proposed in which a new path is the essence of previously corrected path based on the update of operating environment. The chief philosophy of this re-planning scheme is that in facing with threats or obstacles, only the colliding path segments require to be corrected. An online dynamic path re-planning system for an AUV operating in a spatiotemporal cluttered, and uncertain environment was proposed in [122]. The proposed re-planning system works based on a quantum behaved particle swarm optimization (QPSO) algorithm to regenerate optimal path based on the continuously updated information of current profiles obtained from on-board horizontal acoustic Doppler velocity logger sensor. The simulation results showed the ability of this re-planning system in operating in dynamic environment. In all abovementioned underwater re-planning systems, however, the refinement process is undertaken in the level of path planning not trajectory generation. In other words, the dynamics of vehicle is not involved in re-planning system. As examples of online trajectory generation and closed-loop implementation, it can be mentioned to spacecraft docking maneuver proposed and experimentally validated in [114] and missile interceptor problem formulated and solved using combinatory online trajectory optimization scheme in a closed-loop form [100].

A recent and systematic approach to generate optimized trajectories based on instantaneous conditions of a vehicle is that the trajectory generation process be implemented within a control loop [75, 123, 124]. In this loop, the trajectory generation block plays a role of a nonlinear controller which receives feedbacks of state variables. This controller generates optimal control commands (U^*) based on feedbacks of instantaneous states of the vehicle (X). Figure 2.4 shows this configuration.

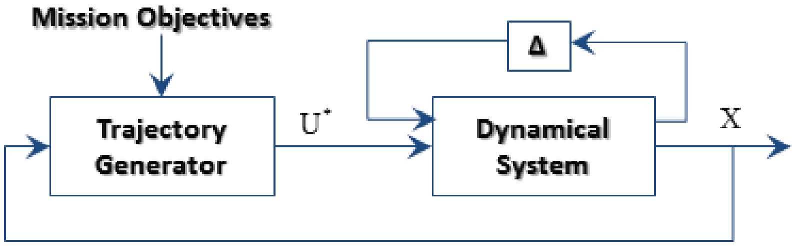


Figure 2.4 Closed-loop configuration used for the online trajectory generation; delta (Δ) represents unmodeled dynamics and uncertainties.

Respecting the fact that states' updates usually happen continuously at a fast rate, the optimized trajectory should be generated instantaneously that is impossible in reality as the nonlinear controller computational time should be taken into account. Therefore, if one can reduce the computational time of trajectory generation process significantly, then for short time intervals, the optimized trajectory can be achieved upon on instantaneous conditions of vehicle.

Basically for realization of the concept of online trajectory generation, two major parts should be taken into account:

- The computational time of trajectory generation process;
- A closed-loop trajectory generation process.

In fact, trajectory generation within a closed-loop configuration is considered as a new control approach, in which controls are generated based on trajectory generation. This approach is similar to what is referred to as model predictive control [125-128].

In model predictive control context, the current control commands are computed based on solutions of a finite horizon open loop OCP achieved at a certain sampling instant. In this approach, the current states of system are used as the initial states. The proposed optimization process yields to an optimal control sequence in which the first control in this sequence is applied to the system. Today, this approach is known usually as receding horizon control (RHC) since the optimal control sequences are generated for future time horizons. The repetitive nature of RHC leads to a state dependent control law in which the physical limitations on states and controls are incorporated as hard or soft constraints. In this approach, frequent changes of cost functions and constraints are possible, as the OCP is solved in a sequence of short time intervals [127]. The closed-loop stability in RHC has been discussed in several documented research in which various Lyapunov control functions are employed to ensure stabilizing RHC approach [127], [129].

RHC has shown successful performances in controlling industrial processes such as chemical processes because of their relatively slow dynamics nature [130]. The RHC algorithm requires a considerable computational process (and thus computationally is expensive) and results in non- or poor-converged stability if implemented improperly. These difficulties have hampered the use of RHC algorithms for nonlinear systems with fast dynamic nature in the past decades [131].

Nowadays, with the advent of affordable and powerful computational tools and also better understanding about the stability of RHC, this approach has again attracted the attention of the control community. The use of RHC for aerospace applications have been recorded in several documented research [123, 132, 133].

Several timing schemes for implementing RHC and constructing the control loops based on the trajectory generation (optimization) process have been proposed [123], [131]. In the following, the RHC scheme adopted form [131] is considered.

In this scheme, with respect to Figure 2.4, the trajectory generation problem is solved online for a finite time horizon $t_{horizon}$ based on the current conditions of state variables. If the computational time in this scheme is t_{sample}, then the optimal control U***** for the computed trajectory is applied to the system in the next t_{sample}. At this point (second t_{sample}), a new trajectory based on current conditions of states is computed and the optimal control of the trajectory is applied to the system in the third t_{sample}. By repeating this process, the vehicle can take advantage of the online trajectory generation to complete its mission. Figure 2.5 graphically demonstrates the proposed scheme. As can be seen from the figure, control commands (U^*_{i-1}, U^*_i, U^*_{i+1}) are computed for a $t_{horizon}$ time interval but just a t_{sample} time interval of control commands is utilized. For example, U^*_i is computed within [t_i, t_i+ $t_{horizon}$] but just part of this trajectory in [t_{i+1},

t_{i+1}+ t_{sample}] is applied to the system. By doing this process, a set of valid controls whose starting points are coincide roughly with the previous trajectory is generated.

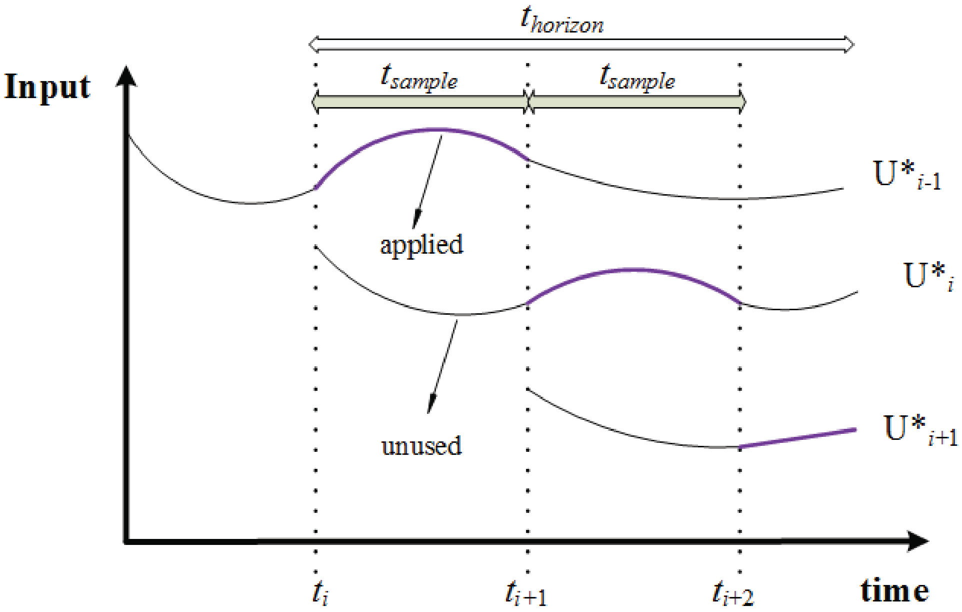


Figure 2.5 Online trajectory generation based on the RHC scheme.

In RHC implementation, $t_{horizon}$ is generally determined proportional to the dynamical nature of a system and mission. On the other hand, t_{sample} is corresponding to the computational time in obtaining a new trajectory. It can be either set as a constant or a variable value meaning that a new trajectory computation is triggered once the previous one has finished. It is noteworthy to mention that to take advantage of online computations (generating trajectory based on instantaneous changes), t_{sample} should be smaller than system time constant.

The next point in RHC implementation that should be taken into account is the possibility of non-converged trajectory computations. In reality, not all trajectory computations are guaranteed to converge since the initial guesses provided based on the last computed trajectory can be improper. In such a circumstance, a simple approach is to let the last reliable trajectory continue until the next computation is trigged.

2.5 Chapter Summary

In this chapter, details on necessary elements for underwater docking operations including mechanical structures of the DS, navigation tools, and docking guidance systems are considered. By reviewing the existing guidance solutions for the docking problem, common features containing the solutions' properness and limitations are extracted. Through this investigation, the need for a new docking problem formulation employing applicable guidance methods from the optimal control theory perspective is identified. Hence, the principle of optimal control theory and its features for the guidance system design or equivalently for developing the trajectory generator engine are considered. Then, the applicable numerical methods, such as PS and IDVD methods, together with viable closed-loop configurations to solve the docking TPBVP and ultimately to develop a systematic and universal docking guidance framework are introduced. In the next chapter, mathematical details on the docking TPBVP are presented.

Chapter 3

Formulation of Docking Problem Using Pontryagin’s Maximum Principle

This chapter encapsulates a systematic problem formulation of the underwater docking problem in the context of optimal control theory. To this end, the Pontryagin’s Maximum Principle (PMP) is employed. Using PMP allows us to transcribe the docking problem into a standard OCP in terms of finding a dynamically feasible set of state and control trajectories that direct the vehicle from its initial conditions to a target DS while minimizing a given performance index. The docking PMP representation provides the possibility of checking the feasibility and optimality of solutions relying on its necessary conditions of optimality.

The proposed PMP representation incorporates the kinematic and dynamic model of AUV together with respective physical limitations of vehicle. For this purpose, first, the AUV model and its specifications used in this thesis are introduced. Moreover, the geometric features of the DS are introduced. Accordingly, the PMP representation is developed in which also the impact of current disturbances, NFZs, and the constraint associated with geometric features of the DS are taken into account. At the end of this chapter, viable approaches for solving the proposed PMP representation together with a software-in-the-loop simulation platform (SITLSP) used to investigate the realization of PMP-based solutions in a realistic sense, are introduce

3.1 AUV Modelling

It deems significant to study the procedure used to derive a general representation AUV's equations of motion even though the focus of this thesis is not towards modelling an AUV. The purpose of this section is to describe a general state space model of an AUV and later, to specify the model and characteristics of the Flinders AUV utilized in this thesis for docking operations.

3.1.1 A General 6-DoF AUV Model

Basically, the first step in developing guidance and/or control system is to provide the AUV system model. State space representation is recommended to facilitate the design phase. The AUV model, in general, comprises two major parts: kinematics and dynamics.

The kinematic equations of AUV are representation of state transformation between the world frame indicated by {n} and body frame depicted by {b}. This transformation is undertaken via Euler angles. Let us define first the vehicle state vector as follows: $\mathrm{X} = [\eta \quad V]^T$ including $\eta = [x \ y \ z \ \phi \ \theta \ \psi \ ]^T$ and $V = [u \ v \ w \ p \ q \ r]^T$ where *x*, *y*, and *z* introduce 3D position of vehicle in {n}; ϕ, θ, and ψ represent the roll, pitch and yaw angles respectively in {n} ; *u*, *v*, and *w* are surge, sway and heave velocities (translational velocities) in {b}; finally *p*, *q*, and *r* represent the angular velocities in *x* , *y* and *z* directions respectively, in {b}.

Now the following presentation is used for vehicle's kinematics

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J(\eta) \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \tag{3-1}$$

where

$$J(\eta) = \begin{bmatrix} {}_b^nR & 0 \\ 0 & {}_b^nW \end{bmatrix} \tag{3-2}$$

and

$${}_b^nR = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{3-3}$$

$${}_b^nW = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \tag{3-4}$$

where ${}_b^nR$ and ${}_b^nW$ are called transformation matrices.

To prevent the singularity problem in the transformation matrices, an AUV is restricted to following Euler angles:

$$\begin{aligned} &-\pi \le \phi \le \pi \\ &-\frac{\pi}{2} < \theta < \frac{\pi}{2} \\ &0 \le \psi \le 2\pi \end{aligned} \tag{3-5}$$

The general dynamic model of an AUV is represented in a form of 6-degree of freedom (DoF) nonlinear equations derived from the Newton-Euler equation of motion [134]

$$M\dot{V} + C(V)V + D(V)V + G = \tau_c + W_e \tag{3-6}$$

where M is a mass and inertia matrix, $C(V)$ represents a Coriolis and centripetal matrix, $D(V)$ is a hydrodynamic damping matrix, G represents a vector of gravitational and buoyancy, τ_c expresses the external force and torque input vector or equivalently vector of control inputs via transformation, W_e is a vector of environmental disturbances (wind, waves, and currents), and finally V is the velocity state vector introduced before.

In the AUV dynamic model, the mass and inertia matrix M is defined as

$$M = M_{RB} + M_A \tag{3-7}$$

where M_{RB} is representative of rigid body mass and inertia, and M_A introduces hydrodynamic added mass elements. The body mass and inertia matrix is defined as follows provided that the body coordinate frame {b} is located at the vehicle's center of gravity

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \tag{3-8}$$

where *m* and *I* represent the mass and inertia of the vehicle. The added mass matrix *M_A* is defined as

$$M_A = \begin{bmatrix} X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{\dot{r}} \end{bmatrix} \tag{3-9}$$

The Coriolis and centripetal matrix is defined as

$$C(V) = C_{RB}(V) + C_A(V) \tag{3-10}$$

where $C_{RB}(V)$ represents the rigid body Coriolis and centripetal matrix associated with M_{RB} defined as

$$C_{RB}(V) = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & mw & -mv & 0 & I_z r & -I_y q \\ -mw & 0 & mu & -I_z r & 0 & I_x p \\ mv & -mu & 0 & I_y q & -I_x p & 0 \end{bmatrix} \tag{3-11}$$

and $C_A(V)$ expresses Coriolis-like matrix induced by M_A and defined as

$$C_A(V) = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix} \tag{3-12}$$

It is noteworthy to mention that matrices in (3-8)-(3-12) are the simplest forms generally used for AUV modeling. For further details, one can refer to [136].

Generally, for a 6-Dof AUV operating at high speed, the hydrodynamic damping matrix is highly nonlinear and coupled. However, a rough approximation of non-coupled motion can be applied for low-speed motions. This results in a diagonal structure for the damping matrix including a linear drag component indicated by D_L and a quadratic drag term shown by D_Q; the diagonal damping matrix is obtained as

$$D(V) = -diag\{D_L + D_Q|V|\} \tag{3-13}$$

that is

$$\begin{aligned} &D(V) = -diag\{X_u + X_{u|u|}|u|, Y_v + Y_{v|v|}|v|, Z_w + Z_{w|w|}|w|, \\ &K_p + K_{p|p|}|p|, M_q + M_{q|q|}|q|, N_r + N_{r|r|}|r|\} \end{aligned} \tag{3-14}$$

To obtain the vector of gravitational and buoyancy (G), let us first remind that for a neutrally buoyant underwater vehicle gravity forces (W) are equal to buoyancy forces (B). To provide a simpler representation of G, it is assumed that the center of gravity introduced by $CG= [x_g, y_g, z_g]^T$ and the center of buoyancy shown by $CB = [x_b, y_b, z_b]^T$ are vertically positioned at the z-axis , that is $x_g=x_b$ and $y_g=y_b$. Therefore, G vector is represented by

$$G = \left[0,0,0, B(z_g - z_b)W\cos\theta\sin\phi, B(z_g - z_b)W\sin\theta, 0\right]^T \tag{3-15}$$

The external force and torque input vector is generated by the vehicle's actuators ($T_1, ..., T_n$) represented as $\tau_c = L_c\mathrm{U}$ where U represents the control vector (vector of thrust) and defined as

$$\mathrm{U} = [T_1, T_2, ..., T_n]^T \tag{3-16}$$

and L_c is a mapping matrix that uses U to find the overall forces and moments applied on a vehicle.

Consequently, the 6-DoF model of an underwater vehicle can be presented in two matrix equations as:

$$\begin{aligned} \dot{\eta} &= J(\eta)V \\ M\dot{V} + C(V)V + D(V)V + G &= \tau_c + W_e \end{aligned} \tag{3-17}$$

It is noteworthy to mention that the model obtained in (3-17) is highly nonlinear and coupled [134].

3.1.2 The Flinders University AUV

Figure 3.1 depictures a controlled scheme of the AUV that is currently under development at the Flinders University. This AUV features a typical torpedo-shape cylindrical body with ellipsoidal head and conical tail with about 120-cm total length and 20-cm diameter. The control scheme shown in Figure1 represents a portion of the overall control scheme envisioned to allow AUV maneuvering at low speeds to enforce precise operations. Three actuators enable independent motion in surge (bow/stern), heave (up/down) and yaw (side-to-side) directions. Specifically, a main propeller provides a forward motion, two lateral thrusters yield the yaw control and two vertical thrusters enable the depth control. The traditional submarine control surfaces, rudder and stern planes (not shown in Figure 3.1), that serve as the primary controls at the high speeds, are still used in low-speed operations assuring roll and pitch stabilization. As a result, the yaw and heave motions for the configuration shown in Figure1 are decoupled from roll and pitch motion.

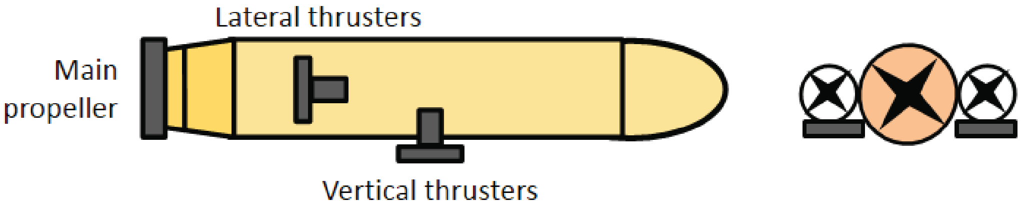


Figure 3.1 Starboard view and rear view of Flinders AUV showing thruster locations.

Due to the existence of a series of complex nonlinearity and coupling in the general 6-

DoF AUV model representation, the design phase for the guidance/control system is severely difficult. However, a series of valid assumptions, which are commonly used in AUV guidance/control system design [135], can be considered to facilitate the design phase. The following assertions on the general AUV model are considered to provide a valid simplified model:

- The vehicle motion in roll and pitch is negligible, as a result of passively controlled roll and pitch operations;
- The decoupling between the degrees of freedom can be assumed; this assumption stems from the fairly symmetrical feature of vehicle about its three planes, negligible impact of hydrodynamic damping coupling at low speed, and negligible impact of off-diagonal elements in the dynamic model in comparison with diagonal elements;
- The sway movement can be considered negligible analogous to roll and pitch. This assumption can be still valid even in presence of the bounded current disturbances which may result in a small sideslip angle.

Now, relying on the abovementioned assumptions and the general state space representation of an AUV mentioned in Section 3.1.1, a simplified 4-Dof model of the AUV is presented as follows:

$$\begin{aligned}\dot{x} &= u\cos(\psi) + c_x \\ \dot{y} &= u\sin(\psi)) + c_y \\ \dot{z} &= w\end{aligned} \tag{3-18}$$

$$\begin{aligned}m\dot{u} - (X_u + X_{u|u|}|u|)u &= T_u \\ m\dot{w} - (Z_w + Z_{w|w|}|w|)w &= T_w\end{aligned} \tag{3-19}$$

$$\dot{\psi} = r \tag{3-20}$$

$$I_z\dot{r} - (N_r + N_{r|r|}|r|)r = T_r \tag{3-21}$$

where *x*, *y*, and *z* are the coordinates of the AUV's center of gravity in the $\{n\}$; *u* and *w* are surge, and heave velocities components in $\{b\}$, relative to the water; ψ is the yaw angle; c_x and c_y are the northerly and easterly components of the current velocity; X_u, Z_w, and N_r are the linear drag terms ; $X_{u|u|}$, $Z_{w|w|}$, and $N_{r|r|}$ are the quadratic drag terms; *m* represents the mass of vehicle and I_z is the vehicle's inertia in the z-direction; finally, T_u, T_w, and T_r are the control inputs in surge, heave and yaw directions, respectively. The values for these parameters were adopted from [135, 136].

3.2 Bolza Optimal Control Problem

The general optimal control problem (OCP) in Bolza form is defined as follows: minimize the cost functional

$$J = \Phi(t_0, \mathrm{X}(t_0), t_f, \mathrm{X}(t_f)) + \int_{t_0}^{t_f} \chi(t, \mathrm{X}(t), \mathrm{U}(t))\, dt \qquad (3\text{-}22)$$

subject to dynamic constraints

$$\frac{d\mathrm{X}}{dt} = f(\mathrm{X}(t), \mathrm{U}(t), t) \qquad (3\text{-}23)$$

the inequality path constraints

$$C(\mathrm{X}(t), \mathrm{U}(t), t) \le 0 \qquad (3\text{-}24)$$

and the boundary conditions

$$\varphi(\mathrm{X}(t_0), t_0, \mathrm{X}(t_f), t_f) = 0 \qquad (3\text{-}25)$$

where $\mathrm{X}(t) \in \Re^n$ and $\mathrm{U}(t) \in \Re^m$ are the state and control vectors respectively, $\Phi(.)$ and $\chi(.)$ represent Mayer and Lagrange cost respectively, and *t* is time.

Now, let us employ PMP and define the Hamiltonian as a scalar function $H : [t_0, t_f] \times \Re^{nx} \times \Re^{nu} \times \Re^{nx} \to \Re$ that is

$$H(t, \mathrm{X}(t), \mathrm{U}(t), \lambda(t)) = \chi(t, \mathrm{X}(t), \mathrm{U}(t)) + \lambda^T(t)\, f(t, \mathrm{X}(t), \mathrm{U}(t)) \qquad (3\text{-}26)$$

where $\lambda(t)$ is the co-state vector derived based on the necessary conditions of optimality. The first-order necessary conditions of optimality are constructed using the set of differential equations for states, co-states, and controls. The state equation (3-23) in the context of PMP can be represented as

$$\dot{\mathrm{X}} = \frac{\partial H}{\partial \lambda} \tag{3-27}$$

The adjoint (co-state) dynamic equations are defined by

$$\dot{\lambda} = \frac{-\partial H}{\partial \mathrm{X}} \tag{3-28}$$

and the control expression is provided by maximizing the Hamiltonian (3-26) with respect to the controls, which results in solving

$$0 = \frac{\partial H}{\partial \mathrm{U}} = \frac{\delta \chi}{\partial \mathrm{U}} + \frac{\partial f^T}{\partial \mathrm{U}} \lambda \tag{3-29}$$

The transversality conditions associated with the first-order necessary conditions are defined as

$$[\frac{\partial \Phi}{\delta \mathrm{X}} - \lambda(t)]_{t_f} \delta \mathrm{X}_f + [H + \frac{\partial \Phi}{\partial t}]_{t_f} \partial t_f = 0 \tag{3-30}$$

Considering possible constraints on the states and controls of a form (3-24), the Hamiltonian in (3-26) is transformed into an augmented form of

$$\overline{H} = H + \mu^T C(\mathrm{X}, \mathrm{U}, t) \tag{3-31}$$

where μ is the vector of Lagrangian multipliers. By using (3-31), the optimal control trajectories are obtained using

$$0 = \frac{\partial H}{\partial \mathrm{U}} = \frac{\delta \chi}{\partial \mathrm{U}} + \frac{\partial f^T}{\partial \mathrm{U}} \lambda + \mu \frac{\delta C}{\delta \mathrm{U}} \tag{3-32}$$

where the Lagrangian multipliers are determined by

$$\mu_i \begin{cases} \leq 0 & C_i(\mathrm{X},\mathrm{U},t)=C_i^L \\ =0 & C_i^L \leq C_i(\mathrm{X},\mathrm{U},t) \leq C_i^U \\ \geq 0 & C_i(\mathrm{X},\mathrm{U},t)=C_i^U \\ \text{unrestricted} & C_i^L = C_i^U \end{cases} \tag{3-33}$$

where C^L and C^U represent the lower and upper bounds of constraint functions. The equation (3-33) is called complementarity conditions determining the switching structure of the optimal controls. In this regard, the co-state equations are changed to

$$\dot{\lambda}=\frac{-\partial H}{\partial \mathrm{X}}=\begin{cases} \frac{\delta f^T}{\delta \mathrm{X}}\lambda+\frac{\delta \chi}{\delta \mathrm{X}}+\mu\frac{\delta C}{\delta \mathrm{X}}, & C_i(\mathrm{X},\mathrm{U},t)=C_i^U \\ \frac{\delta f^T}{\delta X}\lambda+\frac{\delta \chi}{\delta X}, & C_i^L \leq C_i(\mathrm{X},\mathrm{U},t) \leq C_i^U \end{cases} \tag{3-34}$$

3.3 PMP Formulation of Underwater Docking Problem

The docking operation is said to be performed once the transition of the vehicle from its initial conditions, including the vehicle initial pose, to the funnel-shaped entrance of the DS is completed. This transition of the AUV at the terminal stage should conclude to meeting the pose of the DS within an acceptable bound of errors. Now, before proceeding to the PMP representation for the docking problem, let us introduce geometrical features of the DS. The DS proposed in this thesis, is a stationary funnel-shaped DS permitting unidirectional approach of the vehicle. This type of DS, can be modeled using several geometric features, shown in Figure 3.2. Considering the cone sector of DS in Figure 3.2 that is expressed using following parameters:

- $\gamma_d=[x_d,y_d,z_d]^T$ representing the position of the sector in the {n}frame;
- ψ_d ($-\pi \leq \psi_d < \pi$) showings the orientation (heading) of the sector with respect to the North;
- The entrance angle of the cone, parametrized by α_d ($0< \alpha_d \leq 2\pi$);
- The sector length, labeled by h_d;

- R_d and r_d indicating the outer and internal radii of the sector, respectively.

The specific DS parameters used in this thesis were adopted from [137].

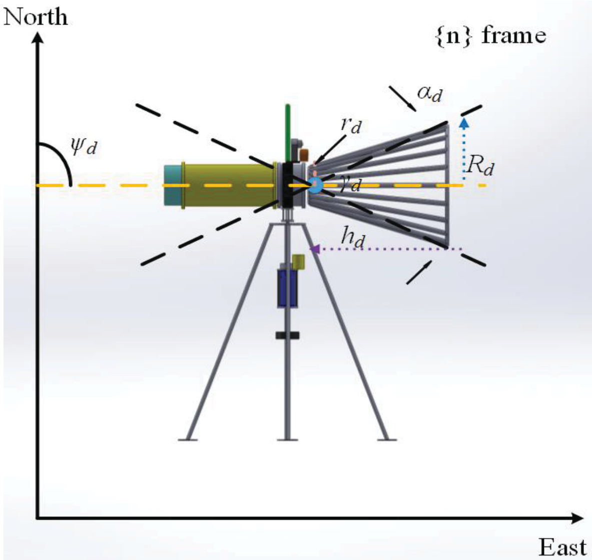


Figure 3.2 Graphical representation of the static docking station.

Now, a general representation of the docking problem in the context optimal control theory is defined. Let us consider the vehicle follows the model introduced in (3-18)-(3-21) in which the utilized state and control vectors are introduced by X=$[x\ y\ z\ \psi\ u\ w\ r]^T$ and U=$[T_u\ T_w\ T_r]^T$, respectively. Using the control vector, it is desired to bring the AUV from some initial conditions introduced by $X_0=[x_0\ y_0\ z_0\ \psi_0\ u_0\ w_0\ r_0]^T$ to docking-enabling conditions described by matching final pose of the vehicle and the DS pose. The final pose of the vehicle is introduced by the vector of $\eta_f=[x_f\ y_f\ z_f\ \psi_f]^T$ stemming from the final conditions of the vehicle $X_f=[x_f\ y_f\ z_f\ \psi_f\ u_f\ w_f\ r_f]^T$. The proposed match at docking-enabling conditions, besides, can be associated with some specified velocity components u_f, and w_f to guarantee the docking completion in presence of current disturbances. While the vehicle is transiting to docking-enabling conditions, it is desired to minimize some performance indices such as

$$J=\int_{t_0}^{t_f}(T_u^2+T_w^2+T_r^2)dt \qquad (3\text{-}35)$$

which represents the energy expenditure or

$$J = \int_{t_0}^{t_f} dt \tag{3-36}$$

representing the time expenditure. There also exists an equivalent formulation for the minimum-time performance index by employing the endpoint cost in the Mayer form that is $J=t_f$ rather than using Lagrange cost mentioned in (3-36). The choice of the specific performance index strictly depends on the details and objectives of the docking missions, conditions of the underlying operating environment, or unforeseen events that might happen during the vehicle operation.

3.3.1 Minimum-Energy Problem

The Hamiltonian for the system (3-18)-(3-21) and the performance index (3-35) can be written as

$$\begin{aligned} H &= -(T_u^2 + T_w^2 + T_r^2) + \lambda_x(u\cos(\psi) + c_x) + \lambda_y(u\sin(\psi) + c_y) \\ &+ \lambda_z w + \frac{\lambda_u}{m}(X_u u + X_{u|u|}|u|u + T_u) + \frac{\lambda_w}{m}(Z_w w + Z_{w|w|}|w|w + T_w) \\ &+ \lambda_\psi r + \frac{\lambda_r}{I_z}(N_r r + N_{r|r|}|r|r + T_r) \end{aligned} \tag{3-37}$$

where the vector λ=[λ_x, λ_y, λ_z, λ_ψ, λ_u, λ_w, λ_r] is the co-state vector associated with the vehicle's state vector. In this regard, the co-states' dynamic equations are obtained by

$$\begin{cases} \dot{\lambda}_x = \dfrac{-\partial H}{\partial x} = 0 \\ \dot{\lambda}_y = \dfrac{-\partial H}{\partial y} = 0 \\ \dot{\lambda}_z = \dfrac{-\partial H}{\partial z} = 0 \end{cases} \tag{3-38}$$

$$\begin{cases} \dot{\lambda}_u = \dfrac{-\partial H}{\partial u} = -\lambda_x \cos(\psi) - \lambda_y \sin(\psi) - \dfrac{\lambda_u}{m}(X_u + X_{u|u|}(|u| + u\,sign(u))) \\ \dot{\lambda}_w = \dfrac{-\partial H}{\partial w} = -\lambda_z - \dfrac{\lambda_w}{m}(Z_w + Z_{w|w|}(|w| + w\,sign(w))) \end{cases} \tag{3-39}$$

$$\begin{cases} \dot{\lambda}_{\psi} = \dfrac{-\partial H}{\partial \psi} = \lambda_x u \sin(\psi) - \lambda_y u \cos(\psi) \\ \dot{\lambda}_r = \dfrac{-\partial H}{\partial r} = -\lambda_{\psi} - \dfrac{\lambda_r}{I_z}(N_r + N_{r|r|}(|r| + r\, sign(r))) \end{cases} \tag{3-40}$$

The control expressions are obtained by taking the derivate of Hamiltonian (3-37) with respect to the controls and imposing the derivatives to be zero, that is

$$\begin{cases} T_u^{\ *} = \dfrac{-\lambda_u}{2m} \\ T_w^{\ *} = \dfrac{-\lambda_w}{2m} \\ T_r^{\ *} = \dfrac{-\lambda_r}{2I_z} \end{cases} \tag{3-41}$$

Now assume specified bounds on the controls and states, representing limitations on the vehicles' actuators, shown by

$$\begin{aligned} C_1 &= |T_u| - T_u^{\max} \le 0 \\ C_2 &= |T_w| - T_w^{\max} \le 0 \\ C_3 &= |T_r| - T_r^{\max} \le 0 \\ C_4 &= |r| - r^{\max} \le 0 \end{aligned} \tag{3-42}$$

and possibility of a path constraint of the form

$$C_5 = -((x - x_{nf})^2 + (y - y_{nf})^2 + (z - z_{nf})^2) + r_{nf}^{\ 2} \le 0 \tag{3-43}$$

that expresses forbidden regions on the state space, obstacles, or NFZs. The Hamiltonian in (3-37) is changed to the augmented form of (3-31). Moreover, the co-states dynamic equations in (3-38)-(3-40) are transformed to the new form of (3-34) and the control expressions in (3-41) change to

$$T_i^* = \begin{cases} sign(\lambda_i)\min(\dfrac{|\lambda_i|}{2m}, T_i^{\max}) & for\, i = u, w \\ sign(\lambda_i)\min(\dfrac{|\lambda_i|}{2I_z}, T_i^{\max}) & for\, i = r \end{cases} \tag{3-44}$$

Besides, the tangential condition associated with the path constraint (3-43) must be satisfied while a trajectory is on the path constraint (C_5=0) that is shown by $dC_5 / dt = 0$. In regions where the path constraint is active, a jump condition or discontinuity is observed in the co-states and Hamiltonian profiles [138, 139].

Finally, the transversality conditions for the minimum-energy expenditure problem are determined by

$$\begin{cases} \mathrm{X}(t_0) = \mathrm{X}_0, \mathrm{X}(t_f) = \mathrm{X}_f & \textit{for the fixed} - \textit{final states} \\ \mathrm{X}(t_0) = \mathrm{X}_0, [\dfrac{\partial \Phi}{\delta \mathrm{X}} - \lambda(t)]_{t_f} = 0 & \textit{for the free} - \textit{final states} \end{cases} \tag{3-45}$$

3.3.2 Minimum-Time Problem

Let us first develop the Hamiltonian function for the system (3-18)-(3-21) and the equivalent Mayer form of time performance index shown in (3-36) as

$$\begin{aligned} H &= 1 + \lambda_x (u\cos(\psi) + c_x) + \lambda_y (u\sin(\psi) + c_y) \\ &+ \lambda_z w + \frac{\lambda_u}{m}(X_u u + X_{u|u|}|u|u + T_u) + \frac{\lambda_w}{m}(Z_w w + Z_{w|w|}|w|w + T_w) \\ &+ \lambda_\psi r + \frac{\lambda_r}{I_z}(N_r r + N_{r|r|}|r|r + T_r) \end{aligned} \tag{3-46}$$

The part of the Hamiltonian in (3-46) associated with the switching functions in the underlying minimum-time problem is

$$\begin{aligned} H^* &= \frac{\lambda_u}{m}(X_u u + X_{u|u|}|u|u + T_u) + \frac{\lambda_w}{m}(Z_w w + Z_{w|w|}|w|w + T_w) \\ &\frac{\lambda_r}{I_z}(N_r r + N_{r|r|}|r|r + T_r) \end{aligned} \tag{3-47}$$

and therefore, the control expressions are obtained as

$$T_i^* = -T_i^{\max} sign(\lambda_i) \quad for\; i = u, w, i \tag{3-48}$$

The adjoint dynamic equations for the proposed unconstrained form of the minimum-time OCP are similar to those in (3-38)-(3-40).

Considering the constraints on controls and state in (3-42) and the path constraint in (3-43), the Hamiltonian in (3-46) is changed to the augmented form of (3-31), the co-states dynamic equations are transformed to the form of (3-34), and the control expressions in (3-47) follow the new form of (3-32).

The transversality conditions for the minimum-time problem in which the final time *t*

$$\begin{cases} \mathrm{X}(t_0)=\mathrm{X}_0, \mathrm{X}(t_f)=\mathrm{X}_f, [\overline{H}+\frac{\partial \Phi}{\partial t}]_{t_f}=0 & \textit{for the fixed} - \textit{final states} \\ \mathrm{X}(t_0)=\mathrm{X}_0, [\frac{\partial \Phi}{\delta \mathrm{X}}-\lambda(t)]_{t_f}=0, [\overline{H}+\frac{\partial \Phi}{\partial t}]_{t_f}=0 & \textit{for the free} - \textit{final states} \end{cases} \tag{3-49}$$

3.3.3 Additional Necessary Conditions

For the proposed constrained docking PMP representations, two additional necessary conditions are defined as follows [138]:

1) in dealing with the minimum-energy problem in which the final time t_f is fixed and the Hamiltonian function does not explicitly depend on time t, the Hamiltonian is constant along the optimal trajectory, i.e.,

$$\overline{H}(\mathrm{X}^*(t), \mathrm{U}^*(t), \lambda^*(t)) = const \quad \forall t \in [t_0, t_f] \tag{3-50}$$

2) in dealing with the minimum-time problem in which the final time t_f is free or not specified priori and the Hamiltonian function does not explicitly depend on time t, the evolution of the Hamiltonian along the optimal trajectory is constantly zero, i.e.,

$$\overline{H}(\mathrm{X}^*(t), \mathrm{U}^*(t), \lambda^*(t)) = 0 \quad \forall t \in [t_0, t_f] \tag{3-51}$$

3.3.4 The Docking TPBVP

By using PMP and developing the first-order necessary conditions, the docking trajectory generation problem is consequently converted into a TPBVP (also called as HBVP in the literature). In other words, the corresponding state, co-state and control equations are solved along with the given initial and final condition vectors. The TPBVP representation of docking is conceptualized as:

> *Starting from* $X_0 = [x_0, y_0, z_0, \psi_0, u_0, w_0, r_0]^T$ *with* $U_0 = [T_{u;0}, Tw_{;0}, Tr_{;0}]^T$*, it is desired to bring the AUV to some equilibrium state* $X_f = [x_f, y_f, z_f, \psi_f, u_f, w_f, r_f]^T$ *(at DS) with corresponding control* $U_f = [T_{u;f}, T_{w;f}, T_{r;f}]^T$ *while obeying constraints (3-42) (and probably path constraint (3-43)) and minimizing the performance index (3-35) or (3-36).*

As explained in Chapter 2, indirect and direct methods are employed to numerically solve the docking TPBVP in order to determine candidate optimal trajectories. However, finding a solution for the docking TPBVP using an indirect method is not an easy task and is associated with a series of practical difficulties. For instance, the initial guess on co-state variables must be properly provided to start the iterative methods. With respect to the non-physical nature of co-states, it is indeed a cumbersome procedure to determine suitable initial guess and an improper one can result in a non-converged or non-optimal solution. There is also a possibility that relying on reasonable guesses, the adjoint equations may turn out to be an ill-conditioned case. In the fortunate case of finding a solution, more often than not, one must accept a huge computational burden that makes this procedure computationally intractable [58, 63, 139].

Therefore, in spite of a small loss of the optimality, direct methods such as PS and IDVD methods are employed in the next chapters of this thesis to solve the docking TPBVP or equivalently to develop a guidance system for the docking purpose. Using the proposed methods, the docking TPBVP is transcribed into a NLP and then an optimization routine, such as *fmincon* or IPOPT, is used to solve the NLP. It has been proven in several documented research that the solution obtained via direct methods are significantly accurate approximations of the solution obtained by indirect ones with the advantage that the former offer more computationally tractable procedures [58, 71, 139].

3.4 Validation of Docking TPBVP Solutions Using SITLSP

To evaluate the solution derived from the docking TPBVP, a post-optimality analysis is undertaken. This is basically achieved by propagating the trajectory generated by direct methods (simulated solution) through a system of ordinary differential equation (ODE) to obtain the system performance (actual solution). Then, the realization of trajectory is checked based on the tractability and particular norms of errors, such as Euclidean norm or L1-norm, between the simulated and actual solutions.

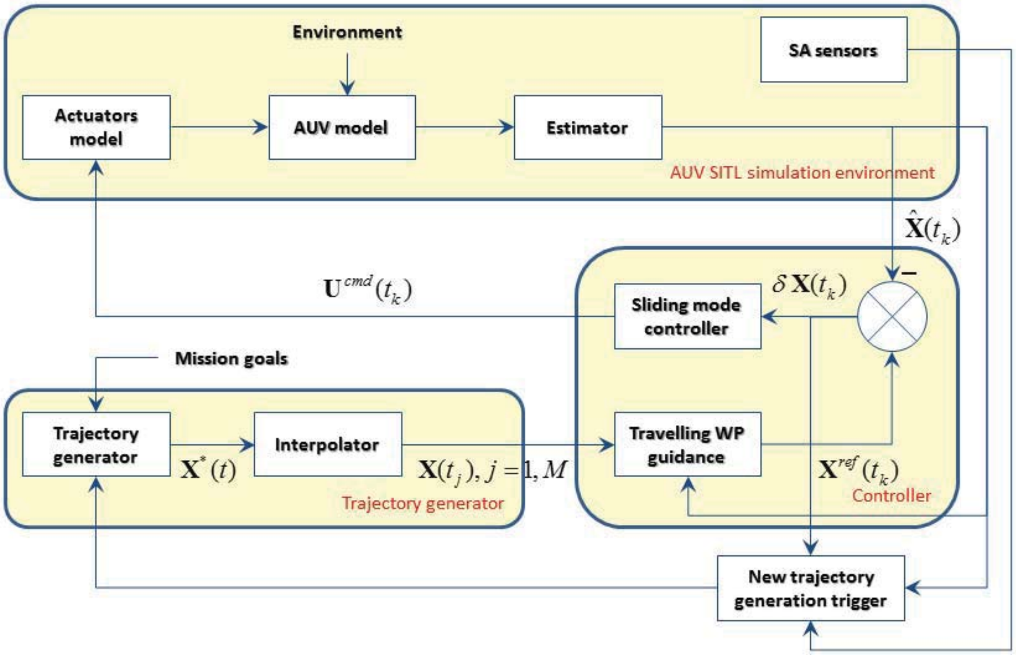


Figure 3.3 The SITLSP for the Flinders AUV.

For this purpose, and more specifically to check the realization of trajectories generated by the docking guidance system in a realistic sense, in this thesis, a SITLSP developed in the MATLAB/Simulink environment is utilized. The overall architecture of SITLSP is shown in Figure 3.3 that has been developed at the Flinders University to test and evaluate different guidance strategies for the Flinders' AUV. This simulation environment incorporates realistically modeled components of the vehicle and emulates a real vehicle behavior while operating in a maritime environment [41, 140, 141]. As shown in Figure3.3, the SITLSP includes three major components. First, it includes the high-fidelity AUV model adopted from [134] featuring fully-coupled six-degree-of-freedom dynamics of the vehicle and test-verified hydrodynamics coefficients including linear and nonlinear drag and lift, hydrodynamic added mass and inertia, and Coriolis force. It also includes the first-order-dynamics controller models and state estimation block. On top of that, it allows emulating additional sensors providing situational awareness (SA) information. In the context of this thesis, SA information includes the updates about the relative position of DS and potential threats (mines, nets, kelp forest, and sunken-ship debris) that need to be avoided on the way to the DS.

Second, the SITLSP provides a testbed to test and verify different trajectory generation engines. This Trajectory generator module is required to be able to generate feasible and trackable reference trajectories, $\mathbf{X}^{*}(t)$, based on the mission goals, known or predicted operation environment. In the case of the mission goals change or change in perception of environmental elements and events with respect to time or space (triggered by the SA sensors or growing discrepancies between the current and expected AUV state) a new trajectory should be generated. Since a trajectory generator usually produces a trajectory that has a relatively small number of not-evenly-spaced (in time) nodes, an Interpolator might be needed to adapt to the actual rate of a controller and produces an evenly-spaced set of *M* nodes, $\mathbf{X}(t_j)$, *j=1,…,M.*

The estimate of the current state at the time instant t_k, $\hat{\mathbf{X}}(t_k)$, and the interpolated reference trajectory are passed into the Controller module. This module mimics the controller that is intended to be used on board the Flinders AUV. This controller is based on the Traveling Waypoint (WP) guidance that computes a synthetic WP [142], which travels along the reference trajectory generated by the Trajectory generator module. In this specific implementation, the location of the Traveling WP, $\mathbf{X}^{ref}(t_k)$, is set to be one meter ahead of $\hat{\mathbf{X}}(t_k)$ along $\mathbf{X}(t_j)$, *j=1,…,M.* This is achieved by creating a 1m radius sphere around the vehicle (corresponding to approximately 80% of the vehicle length) and then generating reference states based on the intersection of the sphere with the interpolated states. The difference $\delta\mathbf{X}(t_k) = \mathbf{X}^{ref}(t_k) - \hat{\mathbf{X}}(t_k)$ is fed into the sliding-mode controller (SMC). The SMC converts this difference into the commanded controls, $\mathbf{U}^{cmd}(t_k)$, necessary to achieve $\mathbf{X}^{ref}(t_k)$. The details of SMC can be found in [141].

For evaluating the tractability of the docking guidance system quantitatively in a realistic sense, the docking-enabling conditions are defined as follows

$$\begin{aligned}
&\Delta\eta = \sqrt{(x_{SITLSP}^{f} - x_f)^2 + (y_{SITLSP}^{f} - y_f)^2 + (z_{SITLSP}^{f} - z_f)^2} \le h_d \\
&\Delta\psi = \left|\psi_{SITLSP}^{f} - \psi_f\right| \le 0.15\alpha_d
\end{aligned} \tag{3-52}$$

$$\begin{aligned}
&\Delta u = \left|u_{SITLSP}^{f} - u_f\right| \le \varepsilon_u \\
&\Delta w = \left|w_{SITLSP}^{f} - w_f\right| \le \varepsilon_w \\
&\Delta r = \left|r_{SITLSP}^{f} - r_f\right| \le \varepsilon_r
\end{aligned} \tag{3-53}$$

where final values of the states obtained by the SITLSP are distinguished by subscript "*SITLSP*" and superscript "*f*". The $\Delta\eta$, and $\Delta\psi$ in (3-52) are final position and direction errors respectively indicating that to what extent the generated controls enable the achievement of the final docking approach. Additionally, Δu, Δw, and Δr in (3-53) represent the terminal errors on surge, heave and yaw rate respectively. These errors should satisfy the tolerance ε_i (i=u,w,r) that are in order of 10^{-3} m/s for surge and heave terminal errors and 6 ^{o}/s for terminal yaw rate error.

3.5 Chapter Summary

In this chapter, the underwater docking problem was formulated from the standpoint of optimal control theory based on PMP. This PMP formulation provides design guidelines for developing a docking guidance system. Meanwhile, it offers insight for feasibility and optimality assessment of the solution. Two forms of problem called minimum-energy and minimum-time docking operations are developed in the context of PMP. A simplified 4-DoF Flinders AUV model together with all vehicular and environmental constraints (path and DS geometry constraints) associated with docking operations are incorporated in the PMP representation. In the sequel, the docking PMP representation leads to a TPBVP. At the end of this chapter, a high- fidelity SITLSP used to validate the tractability and realization of trajectories for real docking trials was introduced.

Considering the practical difficulties in analytically/accurately solving the corresponding TPBVP, direct methods are employed to provide a fairly close approximation of solution. In the next chapter, PS methods, that can provide numerical benchmark solutions, are utilized to solve the proposed TPBVP of docking. The PS methods’ effectiveness and robustness are also analyzed through a series of experiments.

Chapter 4

Generating Optimal Docking Trajectories Using Pseudospectral Methods

This chapter presents benchmark numerical solutions for the docking TPBVP introduced in Chapter 3. For this purpose, three well-known PS methods called LPS, CPS, and hp-AR methods are employed. The major rational to use the PS methods in this chapter is that they explicitly employ the first-order optimality conditions and co-state estimates to approximate the TPBVP solution; as a result, if convergence is achieved, the solution is very close to the true optimal solution. This provides us a valuable insight in developing an optimal docking guidance system in situations where an analytical solution is not achievable. The proposed three methods are different in nature and structure and therefore the solution achieved is not identical. The aim is to investigate comprehensively the main characteristics of each PS method in terms of optimality, computational efficiency and tractability for a real-world implementation in order to know whether they are capable to fulfil the objectives of docking guidance system design mentioned in Chapter 1. For this purpose, a series of representative docking simulations, including docking in a controlled environment, docking in a cluttered environment, and docking in uncertain environments, are designed. In all of the mentioned scenarios, the SITLSP is used to evaluate the realization of trajectories. As part of this work, an enhanced formulation utilizing 10 states rather than just 7, is introduced. That enables satisfying endpoint bounds on the controls, which was not possible with the 7-state formulation.

4.1 Basics of PS Methods

In recent years, the growth in using PS methods for solving OCPs in a wide spectrum of research realms have increased significantly [97,143-146]. PS methods show a spectral accuracy in solving smooth OCPs. This spectral accuracy is to a great extent due to the particular properties of the collocation nodes such as Legendre-Gauss (LG), Legendre-Gauss-Lobatto (LGL), Chebyshev-Gauss-Lobatto (CGL), and Legendre–Gauss–Radau (LGR) nodes and the approach adopted to approximate the differentiation [104,147]. Unlike the local collocation methods (time marching methods) such as the trapezoidal method, PS methods use an efficient discretization approach and sparsity pattern to transcribe a large-scale optimization problem into a sequence of smaller-scale problems [143, 148]. They take advantage of the first-order optimality conditions of Pontryagin principle (but not directly such as indirect methods) to solve an OCP by using co-state/co-vector mapping between the Karush–Kuhn–Tucker (KKT) multipliers of the underlying NLP and the continuous co-state representation of the TPBVP. This feature provides a close approximation of the optimal solution in the case that the solution converges. The solution convergence basically depends on the initial guesses, number of collocation nodes, and the numerical precision of approximated solution. Meanwhile, the computational efficiency of PS methods is a function of the number of collocation nodes, number of mesh intervals, sparsity pattern, and optimization solver.

Basically, PS methods are established on two distinct configurations: p- and h-schemes. In the p-scheme, the corresponding NLP is defined in a single mesh interval and by increasing the degree-of-approximation polynomial, convergence can be achieved. This approach presents an exponential convergence rate at a small number of collocation points for the group of infinitely smooth OCPs. In the h-scheme, on the other hand, the mesh interval is divided into several subintervals and by using a fixed low degree-of- approximation polynomial, the NLP is solved. In this configuration, by increasing the number of mesh intervals, the convergence can be achieved and in order to meet a specified solution accuracy, the mesh refinement is applied on the region of the trajectory where the errors are largest [106]. There are several limitations with individually using the p- or h-schemes. First, in using the p-scheme, an accurate approximation may be obtained by using a large degree global polynomial even for smooth OCPs; conversely, when faced with non-smooth problems, the p-scheme shows an extremely slow convergence rate leading to a poor approximation even when a high-degree polynomial is employed. Second, the p-scheme becomes computationally intractable in dealing with OCPs in which an excessively large-degree polynomial is required. This artefact is due to the dense blocks of the global PS differentiation matrix affecting the NLP density growth. Even though the h-scheme is more computationally tractable than the p-scheme, a large number of mesh intervals may, however, be required to obtain the desired accuracy tolerance as the exponential convergence is lost when the h-scheme is adopted [105, 106].

To overcome the aforementioned difficulties and in order to simultaneously improve accuracy and computational efficiency, the hp-adaptive algorithm which combines the best features of both the h- and p-schemes is employed [107, 108]. In other words, the hp-adaptive algorithm encompasses the advantages of the accuracy of the p-scheme and the computational efficiency of the h-scheme by algorithmically determining the number of mesh intervals, the mesh interval widths, and the polynomial degree in each mesh interval. The desired accuracy tolerance is achieved, either by increasing the degree of polynomial in a mesh interval or by increasing the number of mesh intervals [105, 107].

In this chapter, different configurations of the PS methods for designing and developing an efficient docking guidance system are applied. More specifically, four configurations containing 15, 25, 35, and 50 nodes of the p-scheme for both the LPS and CPS methods together with the hp-AR method for the proposed minimum-energy docking problem, are used.

The simulation results of the proposed configurations are analyzed and compared in detail to find out the main features of them for developing the docking guidance system. For the LPS and CPS methods, the built-in *fmincon* function is used as a solver in which the Jacobian matrix of constraints and gradient vector of objective function is automatically generated based on numerical procedures. Meanwhile, the hp-AR method is implemented within the GPOPS-II package [149] and uses the IPOPT solver [70].

More importantly, in order to provide smooth departure of the AUV from its initial conditions and arrival into the DS, a novel problem formulation in which three new states are introduced and added to the original 7-state TPBVP is offered. Conceptually, by doing this, the endpoint conditions are enforced on the controls to provide a smooth transition of the vehicle to the DS. The simulation results obtained from this formulation analyzed and compared with the 7-state formulation to find out the benefits of each formulation.

4.2 The 10-State Formulation

Given the AUV system dynamics in (3-18)-(3-21) and the 7-state PMP formulation for the minimum-energy docking problem presented in Section 3.3.1, the following boundary conditions (at $t = 0$ and $t = t_f$) is considered:

$$\begin{bmatrix} x \\ y \\ z \\ \psi \end{bmatrix}_{t=0} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \psi_0 \end{bmatrix}, \begin{bmatrix} u \\ w \\ r \end{bmatrix}_{t=0} = \begin{bmatrix} u_0 \\ w_0 \\ r_0 \end{bmatrix} \tag{4-1}$$

$$\begin{bmatrix} x \\ y \\ z \\ \psi \end{bmatrix}_{t=t_f} = \begin{bmatrix} x_f \\ y_f \\ z_f \\ \psi_f \end{bmatrix}, \begin{bmatrix} u \\ w \\ r \end{bmatrix}_{t=t_f} = \begin{bmatrix} u_f \\ 0 \\ 0 \end{bmatrix} \tag{4-2}$$

The first order-derivatives of the states at the starting and terminal points then become

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix}_{t=0} = \begin{bmatrix} u_0 \cos(\psi_0) + c_x \\ u_0 \sin(\psi_0) + c_y \\ w_0 \\ r_0 \end{bmatrix}, \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{r} \end{bmatrix}_{t=0} = \begin{bmatrix} m^{-1}(X_u + X_{u|u|}|u_0|)u_0 + m^{-1}T_{u;0} \\ m^{-1}(Z_w + Z_{w|w|}|w_0|)w_0 + m^{-1}T_{w;0} \\ I_z{}^{-1}(N_r + N_{r|r|}|r_0|)r_0 + I_z{}^{-1}T_{r;0} \\ \end{bmatrix} \tag{4-3}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix}_{t=t_f} = \begin{bmatrix} u_f \cos(\psi_f) + c_x \\ u_f \sin(\psi_f) + c_y \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{r} \end{bmatrix}_{t=t_f} = \begin{bmatrix} m^{-1}(X_u + X_{u|u|}|u_f|)u_f + m^{-1}T_{u;f} \\ m^{-1}T_{w;f} \\ I_z^{-1}T_{r;f} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (4\text{-}4)$$

In order to show a smooth departure from the initial state and arrival into the DS, the following conditions must be upheld:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\psi} \end{bmatrix}_{t=0} = \begin{bmatrix} (m^{-1}(X_u + X_{u|u|}|u_0|)u_0 + m^{-1}T_{u;0})\cos(\psi_0) - u_0 r_0 \sin(\psi_0) \\ (m^{-1}(X_u + X_{u|u|}|u_0|)u_0 + m^{-1}T_{u;0})\sin(\psi_0) + u_0 r_0 \cos(\psi_0) \\ m^{-1}(Z_w + Z_{w|w|}|w_0|)w_0 + m^{-1}T_{w;0} \\ I_z^{-1}(N_r + N_{r|r|}|r_0|)r_0 + I_z^{-1}T_{r;0} \end{bmatrix} \quad (4\text{-}5)$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\psi} \end{bmatrix}_{t=t_f} = \begin{bmatrix} (m^{-1}(X_u + X_{u|u|}|u_f|)u_f + m^{-1}T_{u;f})\cos(\psi_f) \\ (m^{-1}(X_u + X_{u|u|}|u_f|)u_f + m^{-1}T_{u;f})\sin(\psi_f) \\ m^{-1}T_{w;f} \\ I_z^{-1}T_{r;f} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4\text{-}6)$$

Resolving (4-5) and (4-6) with respect to the controls yields

$$\begin{bmatrix} T_u \\ T_w \\ T_r \end{bmatrix}_{t=0} = \begin{bmatrix} m(\ddot{x}_0 \cos(\psi_0) + \ddot{y}_0 \sin(\psi_0)) - (X_u + X_{u|u|}|u_0|)u_0 \\ m\ddot{z}_0 - (Z_w + Z_{w|w|}|w_0|)w_0 \\ I_z\ddot{\psi}_0 - (N_r + N_{r|r|}|r_0|)r_0 \end{bmatrix} \quad (4\text{-}7)$$

$$\begin{bmatrix} T_u \\ T_w \\ T_r \end{bmatrix}_{t=t_f} = \begin{bmatrix} -(X_u + X_{u|u|}|u_f|)u_f \\ 0 \\ 0 \end{bmatrix} \quad (4\text{-}8)$$

Equation (4-8) defines the control values at $t = t_f$ required for the smooth arrival (enforcing the last equations in (4-4) and (4-6)) and equation (4-7) defines the control values at $t = 0$ required for the smooth departure from the initial conditions (which is especially important when the reference trajectory needs to be recomputed while executing the previously generated solution).

Now, considering the normalized form of the performance index (3-35) that is

$$J = \frac{1}{t_f (T_u^{\max})^2} \int_{t_0}^{t_f} (T_u^2 + T_w^2 + T_r^2) dt \tag{4-9}$$

The PMP equations are reconstructed as follows:

$$\begin{aligned} H = &\frac{1}{t_f (T_u^{\max})^2} (T_u{}^2 + T_w{}^2 + T_r{}^2) + \lambda_x (u \cos(\psi) + c_x) + \lambda_y (u \sin(\psi) + c_y) \\ &+ \lambda_z w + \frac{\lambda_u}{m} (X_u u + X_{u|u|} |u| u + T_u) + \frac{\lambda_w}{m} (Z_w w + Z_{w|w|} |w| w + T_w) \\ &+ \lambda_\psi r + \frac{\lambda_r}{I_z} (N_r r + N_{r|r|} |r| r + T_r) \end{aligned} \tag{4-10}$$

$$\begin{aligned} \dot{\lambda}_x &= 0 \\ \dot{\lambda}_y &= 0 \\ \dot{\lambda}_z &= 0 \end{aligned} \tag{4-11}$$

$$\begin{aligned} \dot{\lambda}_\psi &= \lambda_x u \sin(\psi) - \lambda_y u \cos(\psi) \\ \dot{\lambda}_u &= -\lambda_x \cos(\psi) - \lambda_y \sin(\psi) - \frac{\lambda_u}{m} (X_u + X_{u|u|} (|u| + u\, sign(u))) \end{aligned} \tag{4-12}$$

$$\begin{aligned} \dot{\lambda}_w &= -\lambda_z - \frac{\lambda_w}{m} (Z_w + Z_{w|w|} (|w| + w\, sign(w))) \\ \dot{\lambda}_r &= -\lambda_\psi - \frac{\lambda_r}{I_z} (N_r + N_{r|r|} (|r| + r\, sign(r))) \end{aligned} \tag{4-13}$$

$$\begin{bmatrix} T_u{}^* \\ T_w{}^* \\ T_r{}^* \end{bmatrix} = \frac{t_f (T_u^{\max})^2}{2} \begin{bmatrix} \lambda_u m^{-1} \\ \lambda_w m^{-1} \\ \lambda_r I_z{}^{-1} \end{bmatrix} \tag{4-14}$$

On top of the control structure, (4-14) allows a singular control arc when $\lambda_r \equiv 0$ (singular control arc complicates a numerical solution even further). This implies $\dot{\lambda}_r \equiv$

0 and from the last equation in (4-13) $\lambda_\psi \equiv 0$. The latter leads to $\dot{\lambda}_\psi \equiv 0$ in (4-12) and therefore

$$\lambda_x \sin(\psi) \equiv \lambda_y \cos(\psi) \tag{4-15}$$

Knowing that λ_x and λ_y are constants (see (4-11)), equality (4-15) results in $\psi \equiv const$, hence $\dot{\psi} \equiv 0$ and $r \equiv 0$ (from the AUV dynamics equation (3-20)), $\dot{r} \equiv 0$, and $T_r^* \equiv 0$ (from (3-21)). Thus, a singular arc represents a straight motion.

Now, even if the augmented AUV system (3-18)-(3-21), (4-11)-(4-13) could somehow be solved, it would still not satisfy the controls' boundary conditions expressed mathematically in (4-7) and (4-8). In order to satisfy the boundary conditions on controls, one needs to convert them into the new states and uses the new controls. As a result, the original system (3-18)-(3-21) should be augmented with

$$\begin{aligned} \dot{T}_u &= \upsilon_u \\ \dot{T}_w &= \upsilon_w \\ \dot{T}_r &= \upsilon_r \end{aligned} \tag{4-16}$$

By constructing (4-16), constraints (3-42) are incorporated into the augmented Hamiltonian (3-31). The bounds on the new controls would be established as

$$|\upsilon_u| \le \upsilon_u^{\max}, |\upsilon_w| \le \upsilon_w^{\max}, |\upsilon_r| \le \upsilon_r^{\max} \tag{4-17}$$

The Hamilton (4-10) will now have three additional terms

$$H^* = H + \lambda_{Tu}\upsilon_u + \lambda_{Tw}\upsilon_w + \lambda_{Tr}\upsilon_r = const \tag{4-18}$$

with three more differential equations for the new co-states

$$\begin{aligned} \dot{\lambda}_{Tu} &= 2(t_f)^{-1}(T_u^{\max})^{-2}T_u - \lambda_u m^{-1} \\ \dot{\lambda}_{Tw} &= 2(t_f)^{-1}(T_u^{\max})^{-2}T_w - \lambda_w m^{-1} \\ \dot{\lambda}_{Tr} &= 2(t_f)^{-1}(T_u^{\max})^{-2}T_r - \lambda_r I_z^{-1} \end{aligned} \tag{4-19}$$

Differentiating (4-18) with respect to the three new controls yields the bang-bang optimal control structure

$$\begin{bmatrix} T_u^{\ *} \\ T_w^{\ *} \\ T_r^{\ *} \end{bmatrix} = \begin{bmatrix} \upsilon_u^{\max} sign(\lambda_{Tu}) \\ \upsilon_w^{\max} sign(\lambda_{Tw}) \\ \upsilon_r^{\max} sign(\lambda_{Tr}) \end{bmatrix} \tag{4-20}$$

with a possibility of a singular control arc when $\lambda_{Tu} \equiv 0$, $\lambda_{Tw} \equiv 0$, and $\lambda_{Tr} \equiv 0$.

In the next section, first the fundamentals of the proposed PS methods are introduced and then in the subsequent sections of this chapter, numerical solutions obtained by the PS methods are presented for both 7- and 10-states docking TPBVP.

4.3 Legendre PS Method

The LPS method transcribes the proposed TPBVP into an NLP based on both state and control parameterization scheme [148, 150]. The states and controls are discretized over the N collocation nodes and their time histories are approximated using the Lagrange interpolating polynomials

$$\begin{aligned} \mathrm{X}(\bar{t}) &= \sum_{i=0}^{N} \mathrm{X}(\bar{t}_i)\,\phi_i(\bar{t}) \\ \mathrm{U}(\bar{t}) &= \sum_{i=0}^{N} \mathrm{U}(\bar{t}_i)\,\phi_i(\bar{t}) \end{aligned} \tag{4-21}$$

where $\bar{t} = (2t - t_f - t_0)/(t_f - t_0) \in [-1,1]$ is the scaled time, $\mathrm{X}(\bar{t})$,$\mathrm{U}(\bar{t})$ are state and control time histories, and $\phi(\bar{t})$ represents the Lagrange interpolating polynomial

$$\phi_i(\bar{t}) = \prod_{\substack{i=1 \\ j\neq i}}^{N} \frac{\bar{t} - \bar{t}_j}{\bar{t}_i - \bar{t}_j} \tag{4-22}$$

where in the LPS method is of a form

$$\phi_i(\bar{t}) = \frac{1}{N(N+1)L_N(\bar{t})} \frac{(\bar{t}^2 - 1)\dot{L}_N(\bar{t})}{\bar{t} - \bar{t}_i} \tag{4-23}$$

Here $\dot{L}_N(\bar{t})$ represents differentiation of $L_N(\bar{t})$ with respect to the argument $\bar{t}$, and $L_N(\bar{t})$ denotes the Legendre polynomial of order N

$$L_N(\bar{t}) = \frac{1}{2^N N!}\frac{d^N}{d\bar{t}}(\bar{t}^2 - 1)^N \tag{4-24}$$

It is noteworthy to mention that in the LPS method the discretization and collocation nodes are the same and performed by LGL nodes ($\bar{t}_i\ for\ i = 0, \dots, N$).

$$\bar{t}_i = \begin{cases} -1 & if\ i = 0 \\ i_{th}\ root\ of\ \ \dot{L}_N(\bar{t}) & if\ i = \{1, 2, \dots, N-1\} \\ 1 & if\ i = N \end{cases} \tag{4-25}$$

The dynamics of the system in (3-18) - (3-21), is satisfied by imposing the constraint on the LGL nodes. In other words, the continuous differential equations of the system are transcribed into the following algebraic equations, using collocation procedure:

$$\begin{aligned} &\dot{\mathrm{X}}(\bar{t}_k) = \sum_{i=0}^{N} \dot{L}_i(\bar{t}_i)\mathrm{X}(\bar{t}_i) = \sum_{i=0}^{N} D_{ki}\mathrm{X}(\bar{t}_i); \qquad (k = 0, \dots, N) \\ &\sum_{i=0}^{N} D_{ki}\mathrm{X}(\bar{t}_i) - \frac{t_f - t_0}{2} f(\mathrm{X}(\bar{t}_i), \mathrm{U}(\bar{t}_i)) = 0 \end{aligned} \tag{4-26}$$

where $f(\mathrm{X}(\bar{t}_i), \mathrm{U}(\bar{t}_i))$ represents the right hand side expression of the corresponding state equation, and D_{ki} is a differentiation matrix of a size (N+1)× (N+1)

$$D_{ki} = \begin{cases} \dfrac{L_N(\bar{t}_k)}{L_N(\bar{t}_i)}\dfrac{1}{(\bar{t}_k - \bar{t}_i)} & if\ \ k \neq i \\ -\dfrac{N(N+1)}{4} & if\ \ k = i = 0 \\ \dfrac{N(N+1)}{4} & if\ \ k = i = N \\ 0 & otherwise \end{cases} \tag{4-27}$$

The boundary conditions are imposed as the equality constraints on the first and last collocation nodes $\bar{t}_0 = -1$, $\bar{t}_N = 1$, respectively. The final step is to approximate the objective function (4-9) containing the integral term. Using the Gauss quadrature weights, the integral of any polynomial such as $P(\bar{t})$ of degree ≤ $2N$-1 over the $\bar{t} \in [-1,1]$ can be exactly computed as

$$\int_{-1}^{1} P(\bar{t})\, d\bar{t} = \sum_{k=0}^{N} P(\bar{t}_k) w_k \qquad (k = 0, \ldots, N) \tag{4-28}$$

where the quadrature weights w_k are given by

$$w_k = \frac{2}{N(N+1)} \frac{1}{(L_N(\bar{t}_k))^2} \qquad (k = 0, \ldots, N) \tag{4-29}$$

The LPS method uses the first-order optimality conditions of Pontryagin principle to solve an OCP by using co-state/co-vector mapping between the KKT multipliers of the underlying NLP and the continuous co-state representation of the TPBVP. This feature provides a close approximation of the optimal solution in the case that the solution converges. The details on how the LPS method handles the first-order KKT necessary conditions for a solution in nonlinear programming to be optimal can be found in [151].

4.4 Chebyshev PS Method

The CPS method, shown in (4-21), in a similar fashion discretizes the states and controls using N collocation points in the mapped interval of $\bar{t} = (2t - t_f - t_0)/(t_f - t_0) \in [-1,1]$ to transcribe the TPBVP into a NLP [147, 152]. The Lagrange interpolating polynomials used to approximate the time histories of states and controls are similar to (4-22) but they are defined at the CGL nodes

$$\phi_i(\bar{t}) = \frac{(-1)^{i+1}}{N^2 c_i} \frac{(1-\bar{t}^2)\dot{T}_N(\bar{t})}{\bar{t} - \bar{t}_i} \qquad \mathit{for} \;\; i = 0, \ldots, N \tag{4-30}$$

where c_i is

$$c_i = \begin{cases} 2 & i = 0, N \\ 1 & 1 \le i \le N-1 \end{cases} \tag{4-31}$$

and $\dot{T}_N(\bar{t})$ represents differentiation of $T_N(\bar{t})$with respect to the argument $\bar{t}$; $T_N(\bar{t})$ denotes the Chebyshev polynomial of order *N*

$$T_N(\bar{t}) = \cos(N\cos^{-1}(\bar{t})) \tag{4-32}$$

The CGL nodes are defined at the extrema of the *N*th order Chebyshev polynomial and can be conveniently computed by

$$\bar{t}_i = -\cos(\frac{\pi i}{N}) \qquad \mathit{for}\ i = 0,...,N \tag{4-33}$$

The system dynamics constraints are enforced at the CGL nodes to form the defect constraints

$$\begin{gathered}\dot{\mathrm{X}}(\bar{t}_k) = \sum_{i=0}^{N} \dot{T}_i(\bar{t}_i)\mathrm{X}(\bar{t}_i) = \sum_{i=0}^{N} D_{ki}\mathrm{X}(\bar{t}_i); \qquad (k = 0,...,N) \\ \sum_{i=0}^{N} D_{ki}\mathrm{X}(\bar{t}_i) - \frac{t_f - t_0}{2} f(\mathrm{X}(\bar{t}_i), \mathrm{U}(\bar{t}_i)) = 0\end{gathered} \tag{4-34}$$

In (4-34), the CPS method uses a distinct differentiation matrix of D_{ki} of size (*N*+1)× (*N*+1) defined by

$$D_{ki} = \begin{cases} \dfrac{c_k}{c_i}\dfrac{(-1)^{k+i}}{(\bar{t}_k - \bar{t}_i)} & \mathit{if}\ \ k \neq i \\ -\dfrac{\bar{t}_k}{2(1-\bar{t}_k{}^2)} & \mathit{if}\ \ 1 \le k = i \le N-1 \\ \dfrac{2N^2+1}{6} & \mathit{if}\ \ k = i = 0 \\ -\dfrac{2N^2+1}{6} & \mathit{if}\ \ k = i = N \end{cases} \tag{4-35}$$

Similar to the LPS method, the boundary conditions are imposed as the equality constraints on the first and last collocation nodes $\bar{t}_0 = -1$, $\bar{t}_N = 1$, respectively and the path constraints are satisfied at intermediate collocation points. To approximate the objective function (4-9) which similarly follows (4-28), the Clenshaw-Curtis quadrature weights are utilized – the weights are exactly accurate for polynomial up to degree N and represented by

$$w_k = \frac{d_k}{N}\left(1 - \sum_{j=1}^{N/2} \frac{b_j}{4j^2 - 1} \cos(2j\bar{t}_k)\right) \qquad (k = 0, ..., N) \tag{4-36}$$

where

$$b_j = \begin{cases} 1 & if \;\; j = N/2 \\ 2 & if \;\; j < N/2 \end{cases} \quad , \quad d_k = \begin{cases} 1 & if \;\; k = \{0, N\} \\ 2 & otherwise \end{cases} \tag{4-37}$$

The principle of the CPS method is also established on the first-order KKT conditions to provide equivalent continuous co-states and Hamiltonian trajectories for the underlying TPBVP. The mathematical details on primal-dual closure conditions that provide a complete discretization of the continuous necessary conditions for the CPS method are found in [152].

4.5 hp-Adaptive Radau PS Method

The hp-adaptive algorithm is basically developed in the context of multiple-interval OCPs by using the Radau PS (RPS) method [153]. The key advantage in using RPS for the hp-adaptive algorithm is providing continuity conditions that are easily implementable. The RPS is developed based on LGR points which can lie along the interval $\bar{t} \in (-1,1]$ or $\bar{t} \in [-1,1)$, referred to as flipped and standard forms, respectively. The discretization in the RPS method is performed by using N nodes comprising N-1 flipped or standard LGR points and the initial point $\bar{t}_0 \equiv -1$ or terminal point $\bar{t}_0 \equiv 1$. Regarding the asymmetric feature of LGR points, it is beneficial for multiple mesh interval OCP to use LGR points that link the endpoint and starting points of two subsequent mesh intervals [154]. In this thesis, the standard form of LGR points are utilized.

To show mathematically how the RPS works, suppose the time interval $[t_0, t_f]$ is divided into k mesh intervals, $0 = t_0 < t_1 < \cdots < t_k = t_f$ and in each interval for $t \in [t_{k-1}, t_k]$ the affine transformation $\bar{t} = (2t - t_k + t_{k+1})/(t_k - t_{k-1})$, $t_{k-1} < t < t_k$

is valid. The state approximation is defined based on the N_k LGR point and a terminal point

$$\mathrm{X}^{(k)}(\bar{t}) = \sum_{i=1}^{N_k+1} \mathrm{X}^{(k)}(\bar{t}_i)\,\phi_i^{(k)}(\bar{t}); \quad \phi_i^{(k)}(\bar{t}) = \prod_{\substack{i=1 \\ j\neq i}}^{N_k+1} \frac{\bar{t}-\bar{t}_j^{(k)}}{\bar{t}_i-\bar{t}_j^{(k)}} \tag{4-38}$$

where $\left(\bar{t}_1^{(k)},\dots,\bar{t}_{N_k}^{(k)}\right)$ introduce the LGR collocation points in the mesh interval k and the terminal point of the mesh interval k is defined by $\bar{t}_{N_k+1}^{(k)}$.

The control in mesh intervals $k \in [1,\dots,K-1]$ is defined solely at the LGR points with one less point than the state approximation and approximated by using the N_k th degree Lagrange polynomial

$$\mathrm{U}^{(k)}(\bar{t}) = \sum_{i=1}^{N_{k+1}} \mathrm{U}^{(k)}(\bar{t}_i)\,\hat{\phi}_i^{(k)}(\bar{t}), \quad \hat{\phi}_i^{(k)}(\tau) = \prod_{\substack{i=1 \\ j\neq i}}^{N_k+1} \frac{\bar{t}-\bar{t}_j^{(k)}}{\bar{t}_i-\bar{t}_j^{(k)}} \tag{4-39}$$

To approximate the control at the final point t_f, that is a non-collocated point of mesh interval K, the following N_{k-1} th –degree Lagrange polynomial is used:

$$\mathrm{U}^{(k)}(\bar{t}) = \sum_{i=1}^{N_k} \mathrm{U}^{(k)}(\bar{t}_i)\,\tilde{\phi}_i^{(k)}(\bar{t}), \quad \tilde{\phi}_i^{(k)}(\bar{t}) = \prod_{\substack{i=1 \\ j\neq i}}^{N_k} \frac{\bar{t}-\bar{t}_j^{(k)}}{\bar{t}_i-\bar{t}_j^{(k)}} \tag{4-40}$$

It is noteworthy to mention that in (4-39) similar to (4-38) the support points of the Lagrange polynomial $\hat{\phi}_i^{(k)}(\bar{t})$ are N_k LGR points plus the terminal point in mesh interval $k \in [1,\dots,K-1]$ whilst the support points used in (4-40) are only the N_k LGR points.

The defect constraints are formed by differentiating $\mathrm{X}^{(k)}(\bar{t})$ with respect to $\bar{t}$ and then collocating the dynamic equation of system at N_k LGR points in each mesh interval

$$\sum_{i=1}^{N_k+1} D_{hi}^{(k)} \mathrm{X}^{(k)}(\bar{t}_i) - \frac{t_k - t_{k-1}}{2} f(\mathrm{X}^{(k)}(\bar{t}_i), \mathrm{U}^{(k)}(\bar{t}_i), \bar{t}_i^{(k)}) = 0 \quad \textit{for}\ h = 1,\dots,N_k \tag{4-41}$$

where

$$D_{hi}^{(k)} = \dot{\phi}_i(\bar{t}_h^{(k)}) = \begin{cases} \dfrac{\dot{g}(\bar{t}_h^{(k)})}{(\bar{t}_h^{(k)} - \bar{t}_i^{(k)})\dot{g}(\bar{t}_i^{(k)})} & if \quad h \neq i \\ \dfrac{\ddot{g}(\bar{t}_i^{(k)})}{2\dot{g}(\bar{t}_i^{(k)})} & if \quad h = i \end{cases} , 1 \leq h \leq N_k \, , 1 \leq i \leq N_k + 1 \qquad (4\text{-}42)$$

and

$$g(\bar{t}_h^{(k)}) = (1 + \bar{t}_h^{(k)})[P_{N_k}(\bar{t}_h) - P_{N_{k-1}}(\bar{t}_h)] \qquad (4\text{-}43)$$

where $P_{N_k-1}(\bar{t})$and $P_{N_k}(\bar{t})$are the $(N-1)^{\text{th}}$ and N^{th} degree Legendre polynomials, respectively; $D_{hi}^{(k)}$represents $N_k \times (N_k + 1)$ non-square RPS differentiation matrix in mesh interval k.

The boundary conditions are imposed as the equality constraints on the starting and terminal points of the first and the K th mesh intervals, respectively; that is $\mathrm{X}_1^{(1)}, t_0, \mathrm{X}_{N_k+1}^{(K)}, t_K$. Meanwhile, the path constraints in each mesh interval are satisfied at N_k LGR points. The objective function (4-9) is approximated similarly by the quadrature rule (4-28) in which the LGR weights are expressed as

$$w_i^{(k)} = \begin{cases} \dfrac{2}{{N_k}^2} & i = 1 \\ \dfrac{1}{(1-\bar{t}_i)[\dot{P}_{N_k-1}(\bar{t}^k_i)]^2} & i = 2, ..., N_k \end{cases} \qquad (4\text{-}44)$$

Finally, when using multiple-interval RPS, it is essential to keep the continuity in the discretized state variables; this condition is provided by

$$\mathrm{X}_{N_K+1}^{(k)} = \mathrm{X}_1^{(k+1)} , (k = 1, ..., K - 1) \qquad (4\text{-}45)$$

Similar to the LPS and CPS methods, the RPS method also offers its own co-state mapping function to provide equivalent continuous co-states and Hamiltonian trajectories for the underlying TPBVP. The mathematical details that provides a complete discretization of the continuous necessary conditions for the RPS method are found in [107, 154].

In the following section, the basic criteria used in forming the hp-adaptive algorithm are described.

4.5.1 Assessment of Approximation Error in a Mesh

The essence of the hp-adaptive algorithm is to improve the solution accuracy in a computationally efficient manner either by increasing the degree of the polynomial or by further sub-dividing the mesh interval to attain a desired accuracy tolerance of an OCP.

To this end, firstly, ε_d is defined as an accuracy tolerance used for the discretized dynamic and path constraints. Secondly, in each mesh interval, a set of L supporting points $\left(\hat{t}_1^{(k)}, \dots, \hat{t}_L^{(k)}\right) \in [-1,1]$ are defined to estimate the error in the discretized dynamic and path constraints. At this set of support points (L), the differential-algebraic constraints (discretized dynamic and path constraints) are evaluated and yield

$$\left| \dot{\mathrm{X}}^{(k)}(\hat{t}_l^{(k)}) - \frac{t_k - t_{k-1}}{2} f^{(k)}(\mathrm{X}_l^{(k)}, U_l^{(k)}, \hat{t}_l^{(k)}; t_{k-1}, t_k) \right| = e_l^{(k)} \tag{4-46}$$

$$C_l^{(k)}(\mathrm{X}_l^{(k)}, \mathrm{U}_l^{(k)}, \hat{t}_l^{(k)}; t_{k-1}, t_k) = b_l^{(k)} \tag{4-47}$$

where $l = (1, \dots, L)$. In a situation where every element of $e_l^{(k)}$and $b_l^{(k)}$, $l \epsilon (1, \dots, L)$, is less than ε_d , the current mesh interval is considered as an acceptable solution for the OCP. Otherwise, a modification is applied to the current mesh interval either by increasing the degree of the corresponding polynomial or by refining the mesh such that it is divided into the finer sub intervals. The selection rule is described in the next section.

4.5.2 Determination of Polynomial Degree Increase or Mesh Refinement

In a situation where the accuracy tolerance of the mesh interval k is not met, the curvature criterion is used to determine whether the degree of the approximating polynomial in the mesh interval needs to be increased or the mesh refinement process should be applied. To form the curvature criterion, let $\mathrm{X}_m^{(k)}(\bar{t})$ be defined as the component of state approximation in the mesh interval k containing the maximum value of error ($e_{\max}^{(k)}$) in terms of either $e_l^{(k)}$or$b_l^{(k)}$, $l \epsilon (1, \ldots, L)$. The curvature criterion in the mesh interval k is defined as

$$\kappa^{(k)}(\bar{t}) = \frac{\left|\ddot{\mathrm{X}}_m^{(k)}(\bar{t})\right|}{\left|[1+\dot{\mathrm{X}}_m^{(k)}(\bar{t})^2]^{3/2}\right|} \tag{4-48}$$

and subsequently the curvature ratio is defined as

$$r_k = \frac{\kappa_{\max}^{(k)}}{\bar{\kappa}^{(k)}} \tag{4-49}$$

where $\kappa_{\max}^{(k)}$ and $\bar{\kappa}^{(k)}$ are the maximum and average curvature of $\kappa^{(k)}(\bar{t})$, respectively.

Next defining parameter $r_{\max} > 0$ as a user-defined maximum limit; the curvature is considered uniform in the mesh interval k if $r_k < r_{\max}$ and in this situation the degree of polynomial should be increased to achieve a better approximation. If, however, the curvature is relatively large compared to the rest of mesh interval, that is $r_k > r_{\max}$, the mesh refinement should be applied.

4.5.3 Rule for Increasing Polynomial Degree

Let $N_k^+$and N_k^- be defined as the new and former polynomial degree, respectively, in the mesh interval *k*. The following rule is used if the polynomial degree in the mesh interval *k* needs to be increased:

$$N_k^+ = N_k^- + ceil(\log_{10}(e_{\max}^{(k)}) - \log_{10}(\varepsilon_d)) + A \tag{4-50}$$

where *A* is an arbitrary integer number for adjusting the growth of the number of collocation points in a mesh interval and *ceil* is the operator that rounds to the next larger integer.

4.5.4 Rule for Mesh Refinement

Let n_k be defined as the number of new sub-intervals if the mesh interval k needs to be refined. The following rule is used for mesh refinement:

$$n_k = \beta * ceil\ (\log_{10}(e_{\max}^{(k)}) - \log_{10}(\varepsilon_d)) \qquad (4\text{-}51)$$

where β is an arbitrary integer number used to control the growth in the number of mesh intervals. The curvature density function $\rho(\bar{t})$ is then used to determine the locations of the new mesh points

$$\rho(\bar{t}) = c\kappa(\bar{t})^{1/3} \qquad (4\text{-}52)$$

where c is a constant chosen so that

$$\int_{-1}^{1} \rho(\zeta)d\zeta = 1 \qquad (4\text{-}53)$$

and subsequently, the cumulative density function is defined as

$$F(\bar{t}) = \int_{-1}^{\bar{t}} \rho(\zeta)d\zeta \qquad (4\text{-}54)$$

Accordingly, the new mesh points are located so that

$$F(\bar{t}_i) = \frac{i-1}{n_k}, \quad 1 \le i \le n_k + 1 \qquad (4\text{-}55)$$

and (4-55) represents the fact that the density of the subinterval of mesh interval k is proportional to $F(\bar{t})$ as the i^{th} subinterval starts at $\bar{t}_{i\text{-}1}$ and the $(i+1)^{\text{th}}$ subinterval starts at $F(\bar{t}_i)$- $F(\bar{t}_{i\text{-}1})$=1/n_{k} .

4.5.5 The hp-adaptive Algorithm

The hp-adaptive algorithm is initialized by configuring a coarse mesh in which the polynomials' degree are fixed on each mesh interval; then, the proposed OCP problem is transcribed into an NLP problem based on the RPS method. Considering the user-specified accuracy tolerance (ε_d) and comparing the curvature ratio (r_k) with the maximum limit ($r_{\max}$), either the polynomial degree can be increased (as per (4-50)) or the mesh can be refined (as per (4-51)). Figure 4.1 demonstrates the basic structure of the hp-adaptive algorithm.

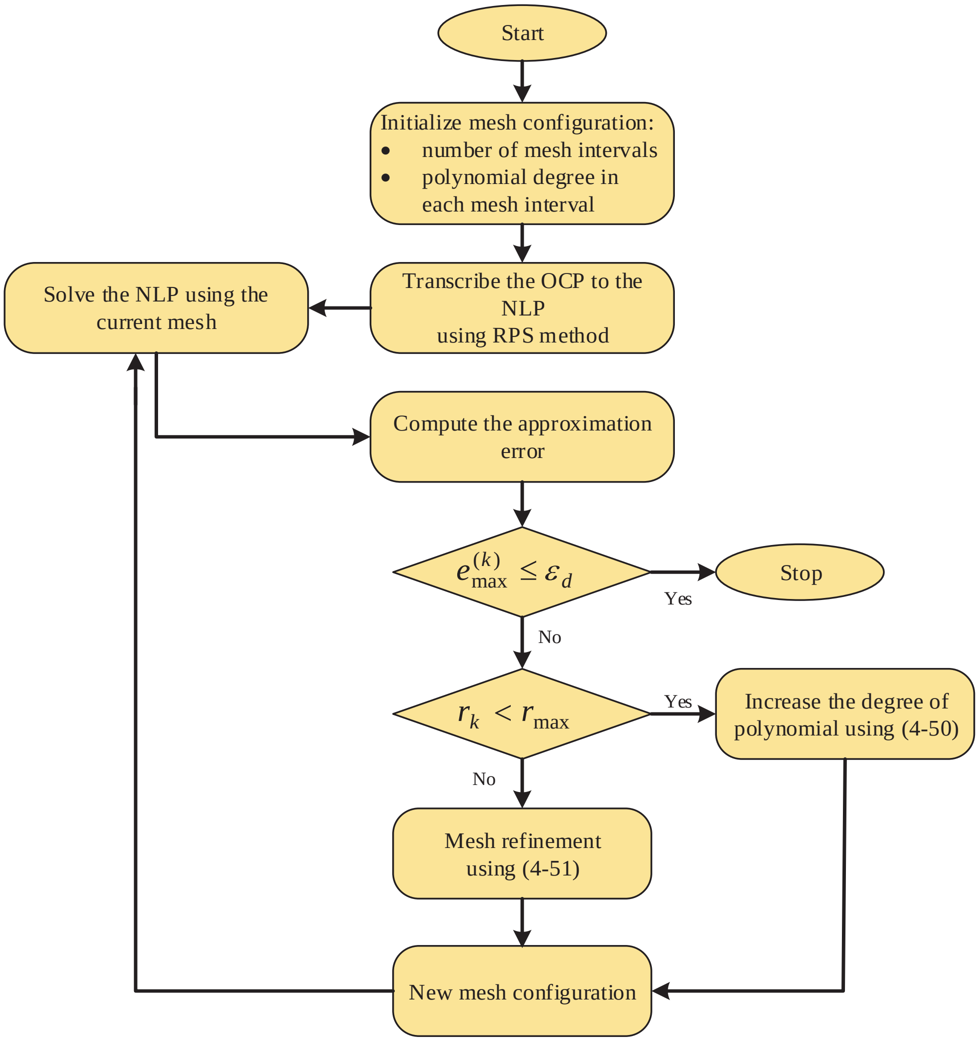

Start
Initialize mesh configuration:
• number of mesh intervals
• polynomial degree in each mesh interval
Transcribe the OCP to the NLP using RPS method
Solve the NLP using the current mesh
Compute the approximation error
$e_{\max}^{(k)} \leq \varepsilon_d$
Yes
Stop
No
$r_k < r_{\max}$
Yes
Increase the degree of polynomial using (4-50)
No
Mesh refinement using (4-51)
New mesh configuration

Figure 4.1 Flowchart of the hp-adaptive algorithm.

The goal of the hp-adaptive algorithm is to utilize as many low-degree mesh intervals as possible and to employ large-degree polynomials only in those intervals where the solution is smooth. In essence, the key advantages of the hp-adaptive algorithm can be listed as follows: first, relying on the low-degree approximation polynomials in each mesh interval, the resulting NLP problem takes advantage of a remarkably sparse configuration enabling the NLP solver to easily and efficiently compute the solution.

Second, due to the exponential convergence of the RPS method in the mesh intervals where the solution is smooth, an accurate solution can be achievable using only a relatively small number of collocation points. As a result, the simultaneous improvement in terms of solution accuracy and computational efficiency can be achieved by the hp-AR method.

4.6 Simulation Results for the 7-State Formulation

In this section, the docking simulation results showing performance with minimum control effort (minimum energy expenditure) are presented. Mathematically speaking, the minimum-energy docking performance is abstracted in a form of performance index (3-35). In this section, however, the normalized performance index (4-9) is utilized. With this performance index, the quantity $ES = 100\left(1 - \sqrt{J}\right)\%$ provides an estimate of the saved control input expenditure compared to the case when the maneuver is performed at control bounds ($T_u(t) = T_u^{mx}$).

The minimum-energy docking maneuvers are performed in an environment where the DS pose is known a priori and the operating environment is free of any NFZ or obstacles (controlled environments). In this environment, a 2D current disturbance of magnitude c_x=0.25 m/s, c_y=0.25 m/s with respect to the North and East respectively in the $\{n\}$ frame, is considered. The AUV's initial pose is defined by a vector of $\eta_0 = [x_0, y_0, z_0, \psi_0]^T$ in which the initial position of the vehicle in the $\{n\}$ frame comprises x_0=50 m, y_0=50 m, z_0=5 m and the initial heading angle with respect to the true North is defined as ψ_0=10˚. The rest of initial conditions are defined as follows: u_0=0.3m/s, w_0=0 m/s, and r_0=0˚/s. The DS pose is introduced by $\eta_d = [x_d, y_d, z_d, \psi_d]^T$ in which the position of DS in the $\{n\}$ frame is defined by x_d =150 m, y_d =75 m, z_d=10 m and the orientation of the DS with respect to the true North is defined by ψ_d=225°.Therefore, the spatial coordinates of the vehicle at the terminal docking phase defined by the equilibrium state vector $\eta_f = \left[x_f, y_f, z_f, \psi_f\right]^T$ must meet the DS position; besides, the vehicle needs to show a final heading of 45˚in the$\{n\}$ frame to be aligned with the centerline of DS for the final docking approach. The rest of final conditions are: x_f=0.4m/s, w_f=0 m/s, and r_f=0˚/s. The constraints defined by the physical limitations of the AUV's actuators and its yaw rate (3-42) are $T_u^{max} = T_w^{max} = 20$N , $T_r^{max} = 20$Nm, and r^{max}=15°/s.

In this scenario, four different configurations of the LPS and CPS methods with 15, 25, 35, and 50 nodes are implemented. The rationale of using these configurations of different node numbers is that it allows investigation of the key distinctions in terms of solution optimality, tractability and performance effectiveness. The configuration used for the hp-AR method, on the other hand, comprises four collocation points per mesh and two segments in each mesh (hp-(2,4)); thus, it includes 8 collocation points in initialization phase. Meanwhile, the maximum number of mesh iteration is set to 10. All computational processes were performed on a desktop computer with an Intel i7 3.40 GHz quad-core processor using MATLAB®R2015a development environment. The accuracy tolerance of 10^{-6} is set for the discretized dynamic and path constraints within the *fmincon* solver for the LPS and CPS methods and the IPOPT solver for the hp-AR method. The initial guesses used to assure and speed up the convergence solutions of all three PS methods are obtained using the IDVD method, which is discussed in the next chapters.

Figures 4.2-4.5, 4.6-4.9 and 4.10-4.13 illustrate the simulation results obtained by the LPS, CPS and hp-AR methods respectively. As can be seen in Figures 4.2, 4.6 and 4.10, the vehicle maneuvers smoothly from the starting point and is able to steer through the pre-specified DS position. This steering concludes with the vehicle aligning itself with the centerline of the DS entrance as a result of having a final heading angle equal to 45˚ (in the $\{n\}$ frame) as indicated in Figures 4.3,4.7 and 4.11(and thus satisfying the transversality conditions (3-45)). Meanwhile, the yaw rate trajectories generated by the LPS, CPS and hp-AR methods respect the constraint associated with limitation of the vehicle in instantaneous turn during the docking motion. In Figures 4.4, 4.8 and 4.12, one can observe the evolutions of the surge and heave velocities during the docking operation, achieved by the LPS, CPS and hp-AR methods respectively. It is obvious that, in all configurations, the boundary conditions corresponding to the surge and heave velocities (associated with the transversality conditions of (3-45)) are thoroughly satisfied. Besides, the PS methods' based docking guidance system offers enough flexibility to handle variations in the vehicle's velocity components, as opposed to most of the other existing methods reported in the literature [21, 24, 26]. The key advantage of having variable velocity in docking operation is that it allows more degree of freedom in using total range of the vehicle's maneuverability. Besides, this behavior is better suited for the complex environment in which there exist current disturbances and uncertainties associated with the operating environment. The docking maneuverability is provided by the smooth, non-saturated trajectories that roughly resemble a bang-singular-bang control, as indicated in Figures 4.5, 4.9 and 4.13. These controls offer the best possible solution for minimum-energy docking maneuvering in presence of the current disturbances. However, a smooth arrival to the DS requires a relax condition on the final value of the vertical and horizontal control efforts. In other words, the final boundary condition on the vertical and horizontal controls should be zero but the PS methods cannot encapsulate this condition based on the 7-state TPBVP formulation. Because of this issue, the 10-state problem formulation was offered in Section 4.2.

It is noteworthy to highlight that obtaining feasible solutions based on the PS methods is largely resorted to initializing the PS methods with a proper set of initial guesses. It was found through a series of simulation trials that feeding the PS methods with improper initial guesses resulted in infeasible solutions. These solutions are often obtained with a drastic increase in computational time. Figure 4.14 shows an example of infeasible solutions achieved based on zero initial guesses for some of the states when the 15-node LPS method is employed. This is one of the major difficulties associated with employing the PS methods. Either the designer must have an enough knowledge about the underlying problem space or a systematic initial guess generator should be used. As mentioned before, the IDVD method used as the initial guess generator engine to assure and speed up the convergence solutions of all three PS methods.

As the solution structure of the proposed TPBVP is not known a priori, the hp-AR method allows for polynomial degree, number and location of mesh intervals to be variable during optimization process. Thus, by using 4 collocation points per mesh and after 6 iterations, it converges to the same value of performance index as achieved by the LPS and CPS methods but with a higher computational efficiency. The mesh refinement history of hp-AR method is presented in Figure 4.15 (see Appendix A) together with distribution of collocation points for all mentioned LPS and CPS configurations. As can be seen from part (c) of the figure, between the 4th and the 6th iterations, the density function near the start and end of the trajectories becomes large and therefore to achieve the desired tolerance considerable subdivisions of intervals must be applied. The convergence to the optimal solution is achieved at the 6th iterations that is associated with a light computational burden. As a result, the docking operation can be accomplished successfully in a computationally efficient manner while the vehicle expends the least control effort.

Table 4.1 quantitatively compares the performance of all PS method configurations based on number of nodes (*NoN*), normalized performance index (*J*), computation time (t_{CPU},*s*), and energy saving percentage (*ES%*). As can be seen from the table, approximately 54% energy saving during docking operation is achieved by the PS methods' based guidance system. It can be inferred from the table that increasing the number of collocation nodes has direct impact on decreasing the computational efficiency in the LPS and CPS methods. The number of collocation points recorded for the hp-AR method in Table 4.1 is the total number of collocation points used in the final mesh. Regarding the details of the hp-adaptive algorithm explained in Section 4.5, this number of collocation nodes is the result of the initial mesh provided, the mesh error tolerance, and the mesh refinement technique used in the hp-AR method.

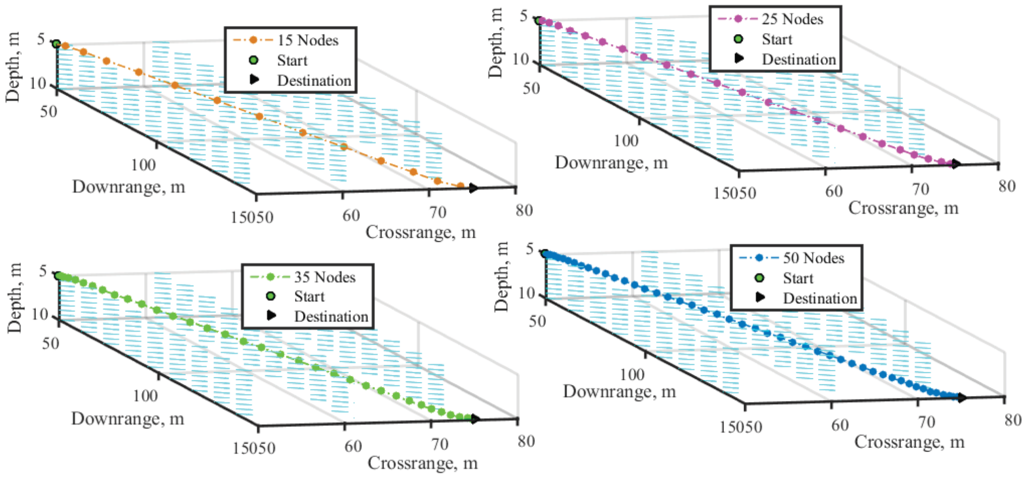


Figure 4.2 3D path for transiting AUV from initial position to DS using the LPS method.

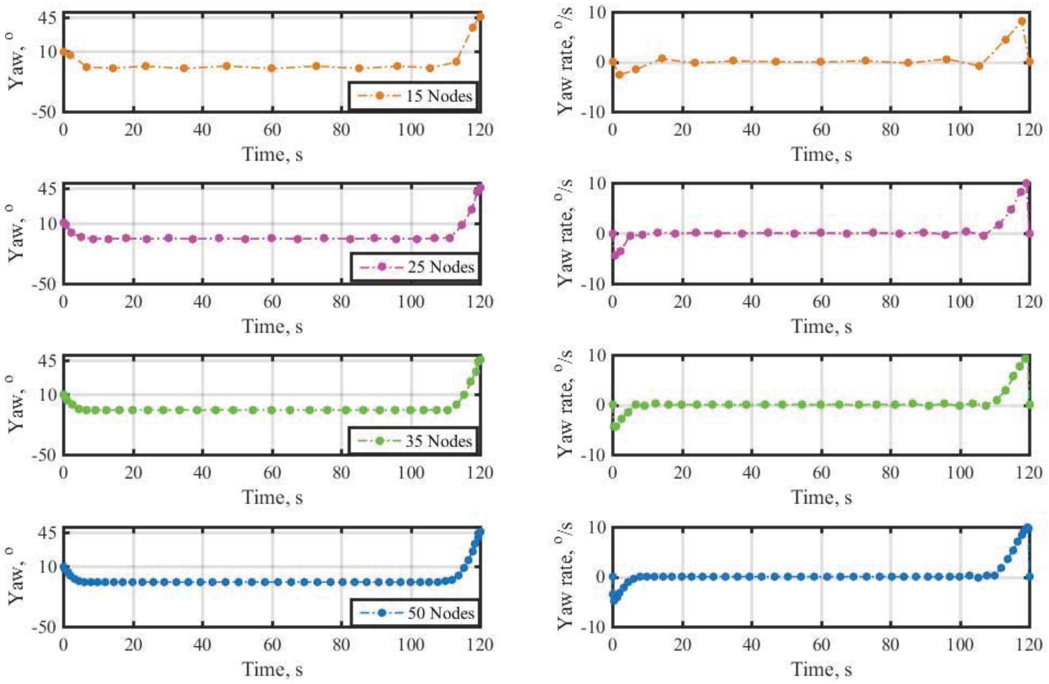


Figure 4.3 Time history of yaw and yaw rate using the LPS method.

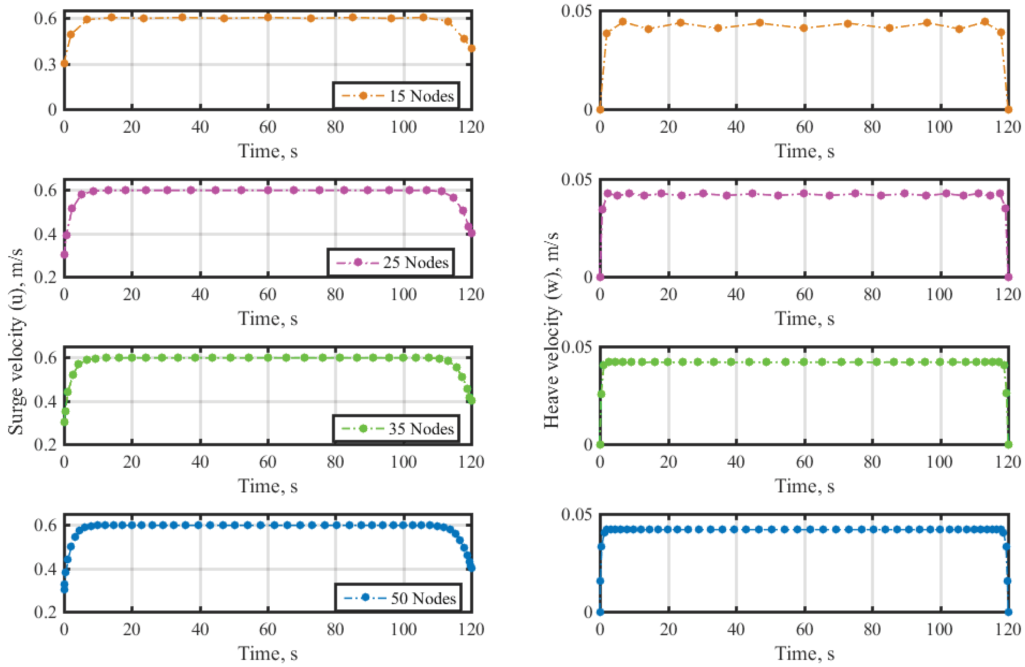


Figure 4.4 Time history of surge and heave velocities using the LPS method.

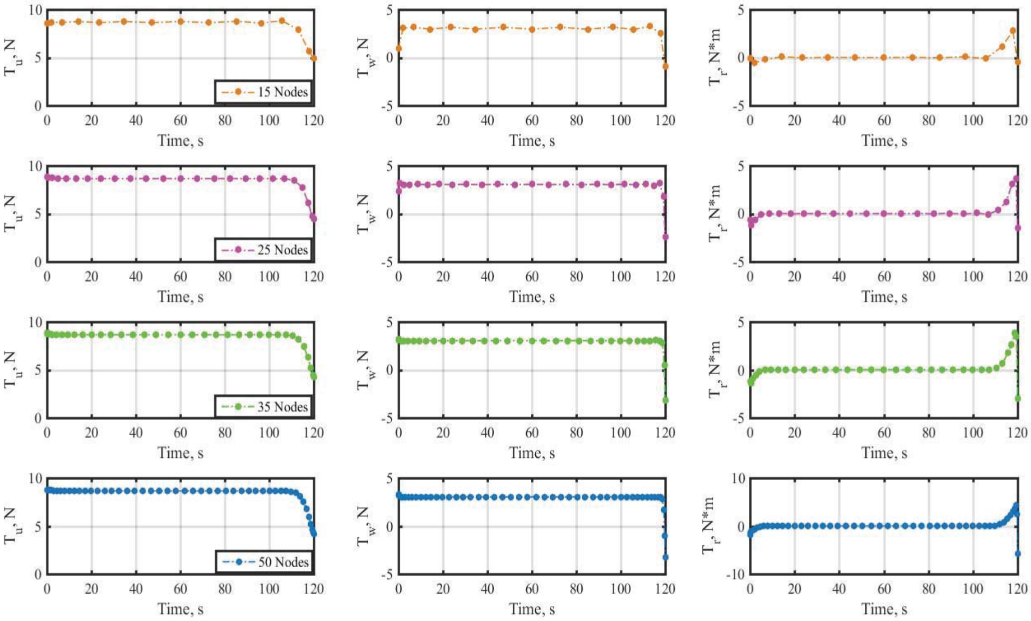


Figure 4.5 Evolution of controls using the LPS method.

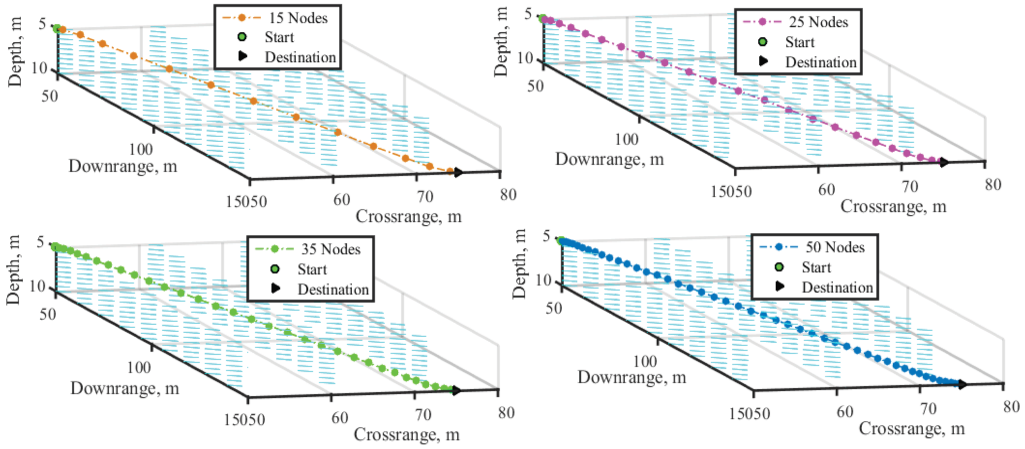


Figure 4.6 3D path for transiting AUV from initial position to DS using the CPS method.

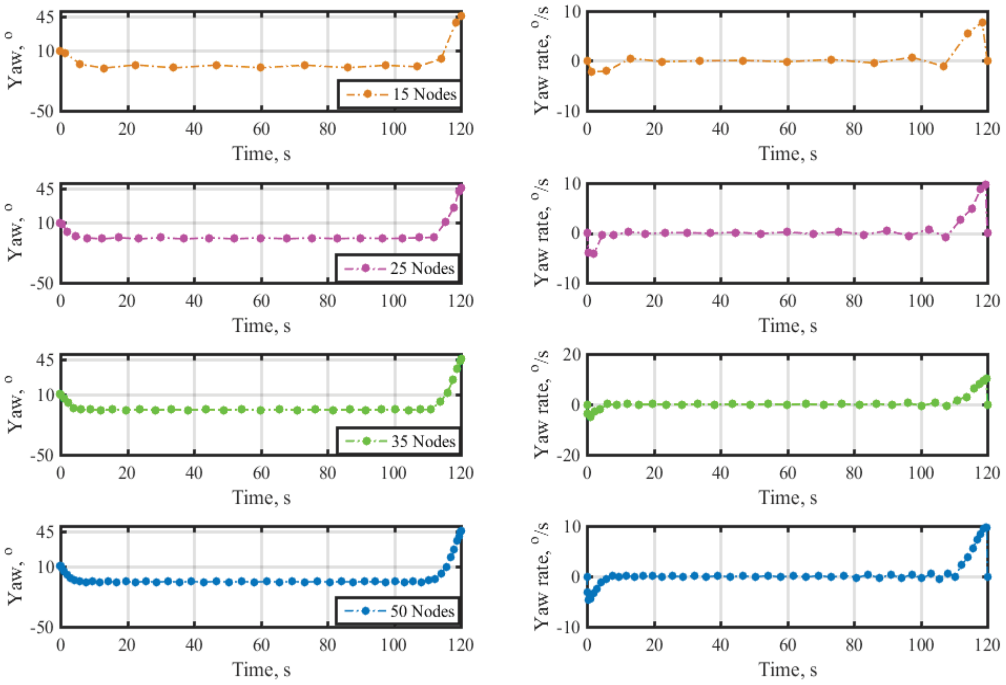


Figure 4.7 Time history of yaw and yaw rate using the CPS method.

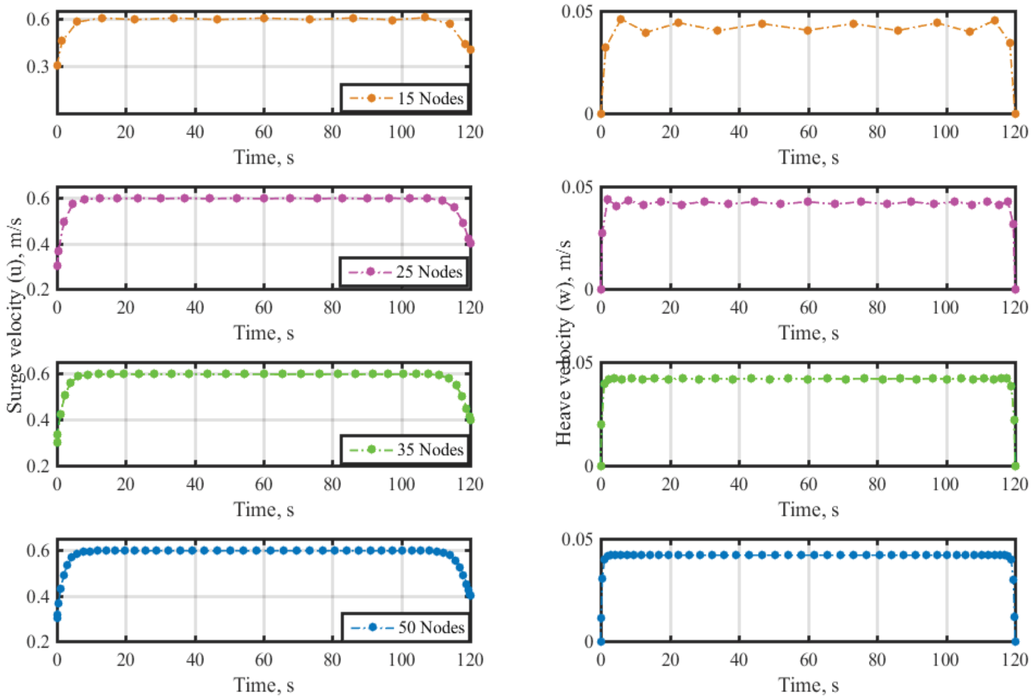


Figure 4.8 Time history of surge and heave velocities using the CPS method.

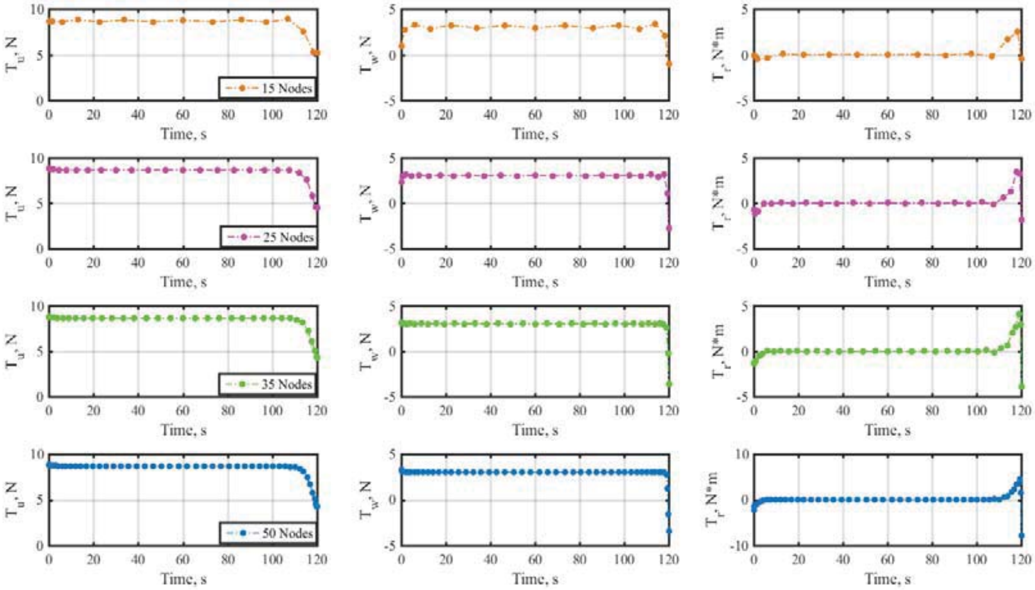


Figure 4.9 Evolution of controls using the CPS method.

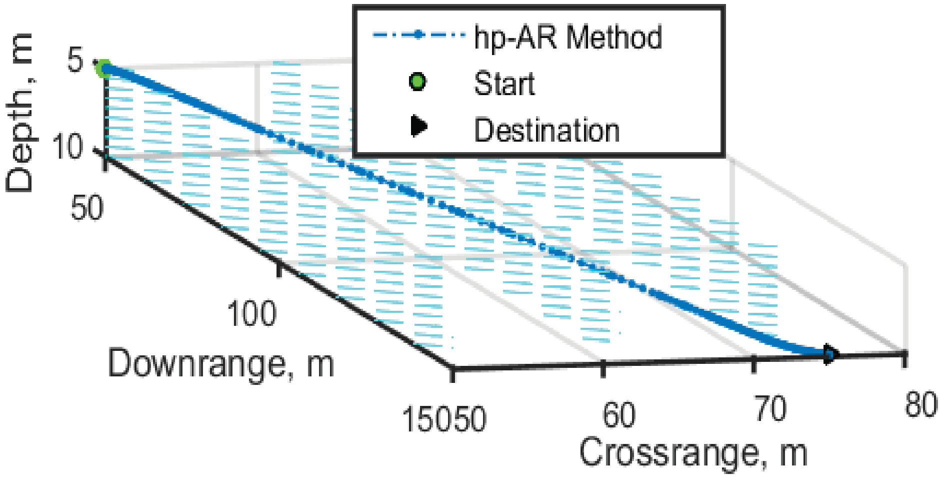


Figure 4.10 3D path for transiting AUV from initial position to DS using the hp-AR method.

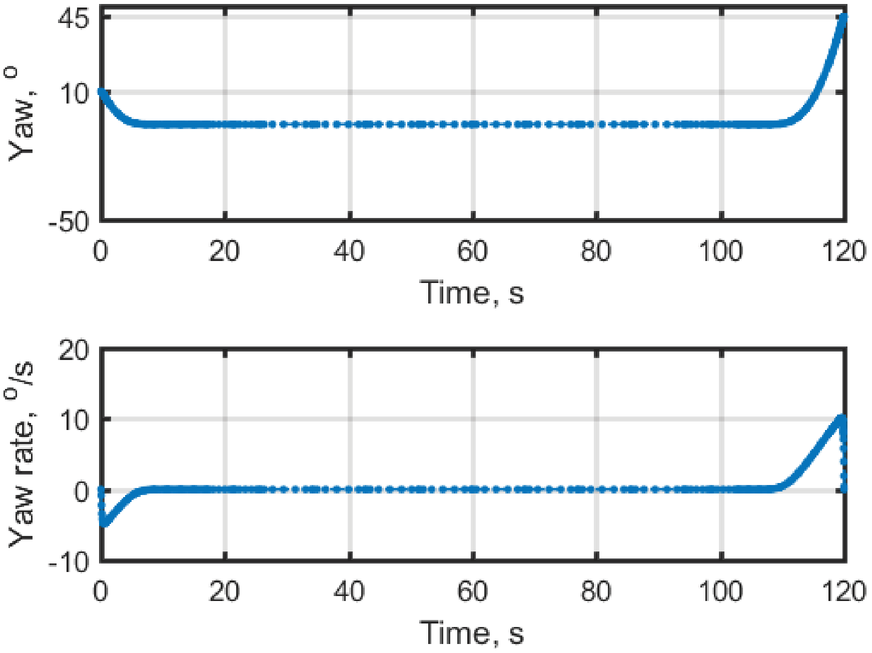


Figure 4.11 Time history of yaw and yaw rate using the hp-AR method.

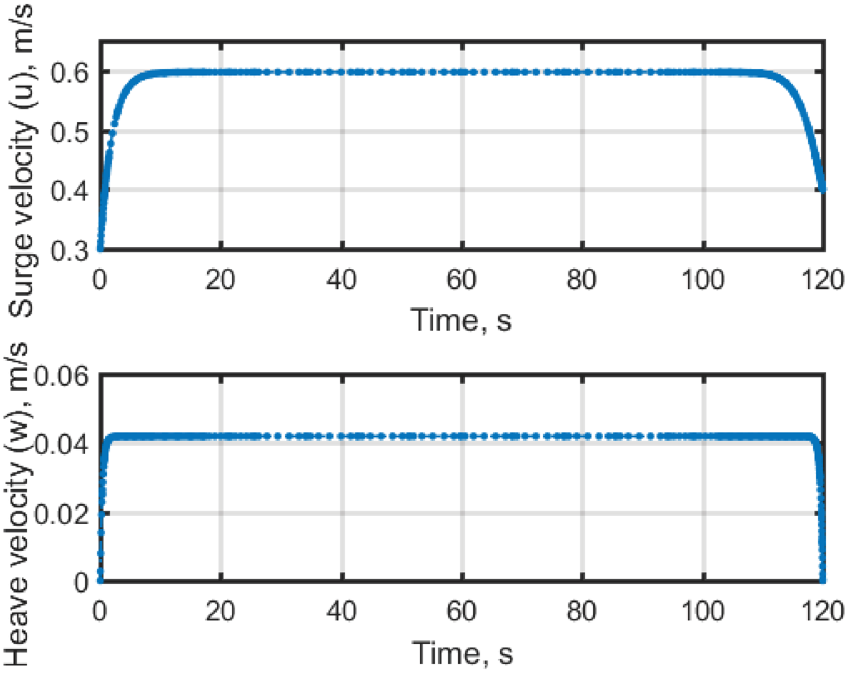


Figure 4.12 Time history of surge and heave velocities using the hp-AR method.

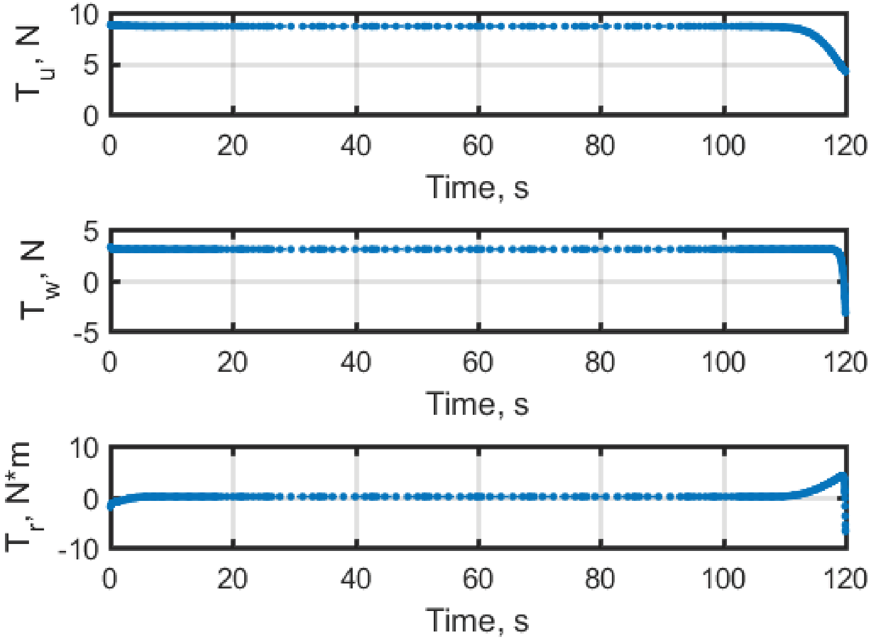


Figure 4.13 Evolution of controls using the hp-AR method.

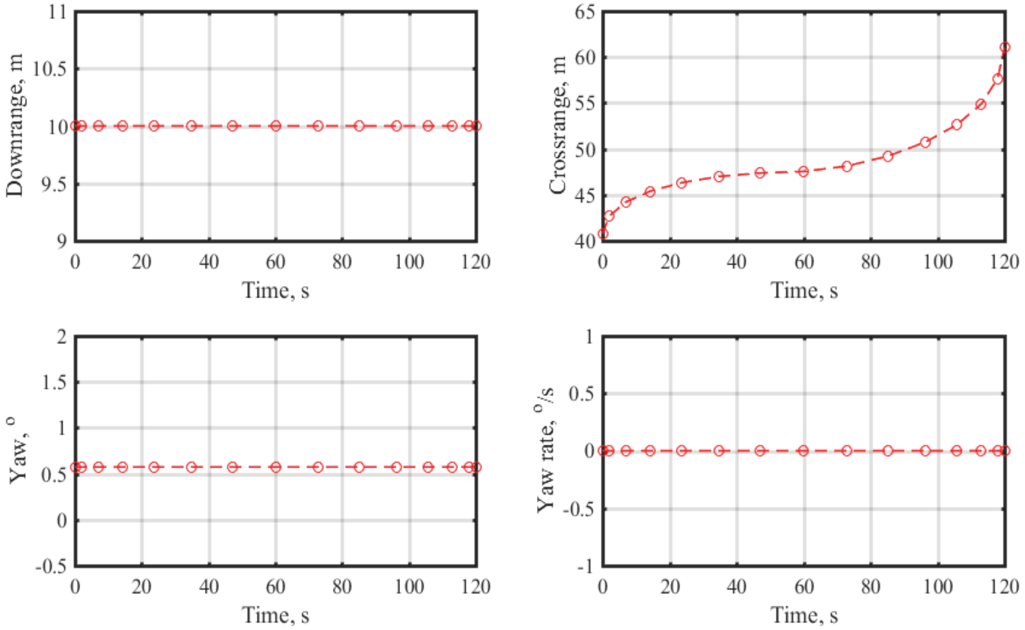


Figure 4.14 An example of infeasible solutions generated by the 15-node LPS method relying on improper initial guesses.

Table 4.1 Quantitative expression of the PS methods' performance.

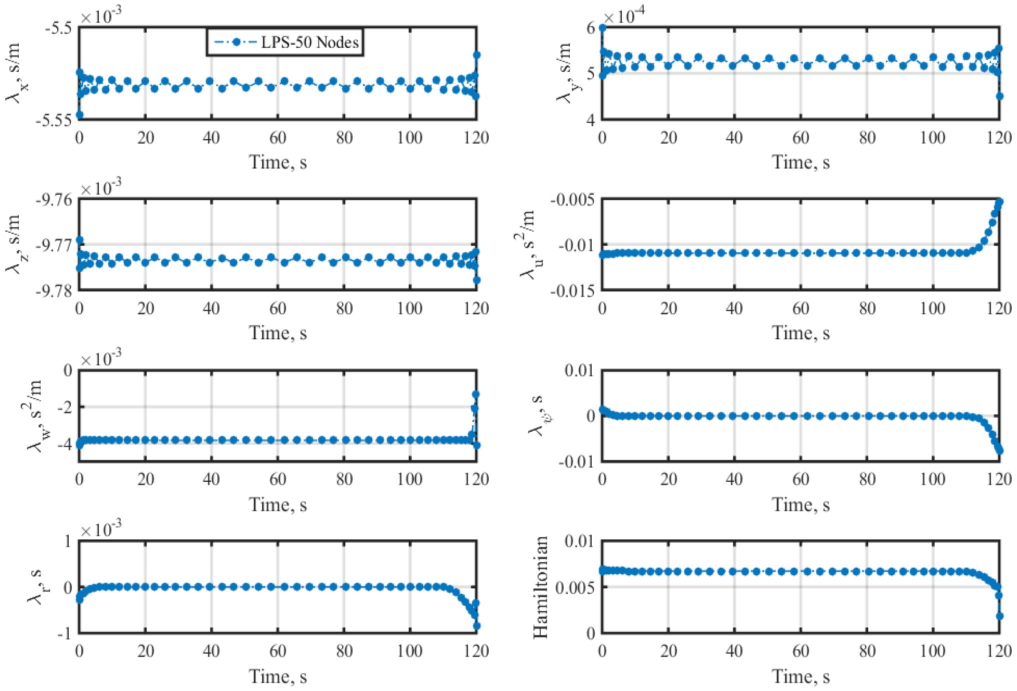
Method/Metric	***NoN***	***J***	***t_{CPU},s***	***ES%***
LPS	15	0.2069	3.88	54.51
CPS	15	0.2067	4.24	54.53
LPS	25	0.2061	15.71	54.60
CPS	25	0.2060	16.40	54.60
LPS	35	0.2060	36.96	54.61
CPS	35	0.2060	40.30	54.61
LPS	50	0.2060	125.17	54.61
CPS	50	0.2060	127.31	54.61
hp-AR	86	0.2060	2.84*	54.61

* The hp-AR solution utilizes a compiled IPOPT solver as opposed to the *fmincon* interpretative solver.

4.6.1 Validation of Solution Optimality

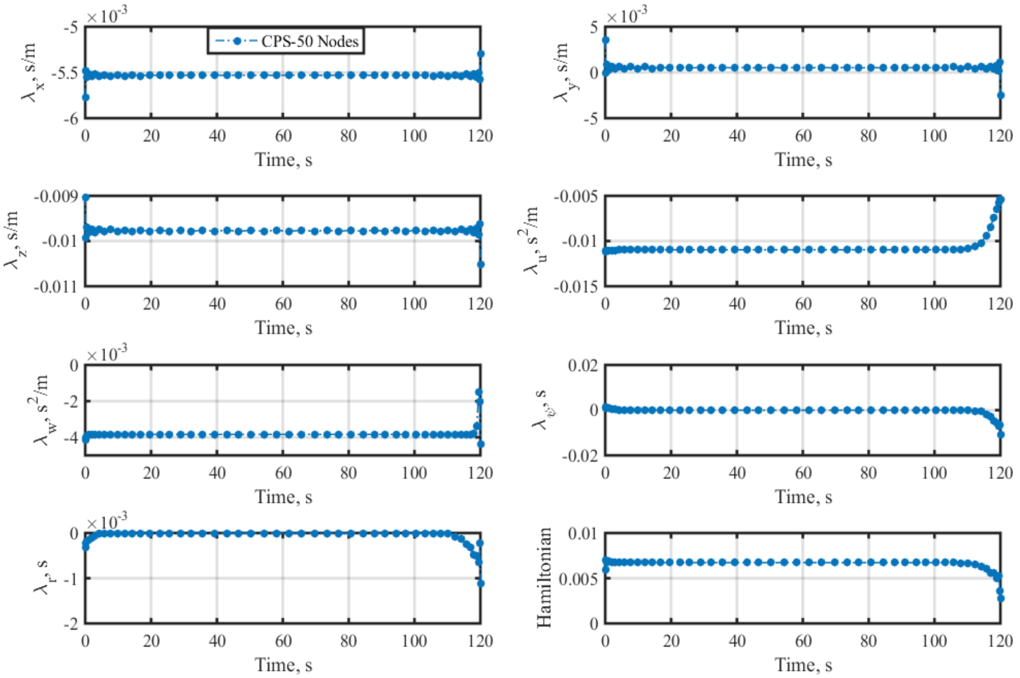
As mentioned in Sections 4.3-4.5, the PS methods enforce the first-order optimality conditions (3-38)-(3-41) to solve the docking TPBVP. They use particular mapping between the KKT multipliers of the underlying NLP and the continuous co-state representation of the TPBVP. This feature provides a close approximation of optimal solution in the case that solution converged. By having co-state and Hamiltonian trajectories, insight for feasibility/optimality verification of solution is provided without resorting to solve the TPBVP arising from the calculus of variations.

Figures 4.16(d), 4.17(d), and 4.18 demonstrate time histories of co-states and Hamiltonian for the 50-node LPS, 50-node CPS, and hp-AR methods respectively (see Appendix A for 15-, 25-, and 35-node configurations of both LPS and CPS methods). The trajectories associated with the spatial co-states λ_x, λ_y, and λ_z in Figures 4.16-4.18 indicate that differential equations (3-38) hold in the numerical sense. More precisely, it can be inferred from the results of the LPS and CPS methods that when using a lower number of nodes, 15, 25 and 35 nodes, the trajectories are not smooth, and comprise chattering and oscillations specially in the LPS solution. Increasing the number of nodes to 50 makes the quality of the solution better in both LPS and CPS methods (therefore, this configuration for both LPS and CPS methods is used in the next section). In contrast, the hp-AR spatial co-states show a stable behavior with a good numerical precision in holding (3-38). The co-states λ_ψ and λ_r in Figures 4.16-4.18 indicate that most of the time the solution features a singular arc control (4-15). Regarding the additional necessary conditions mentioned in (3-50), the Hamiltonian is explicitly independent of time and therefore should be constant. Figures 4.16 and 4.17 show that it is the case along the singular arc with LPS and CPS solutions. However, the optimization routine has difficulty in numerically solving the problem beyond that. In contrast, the Hamiltonian associated with the hp-AR solution in Figure 4.18 shows a stable and constant behavior along the optimal trajectory except some variations in order of 10^{-5} at the beginning and end of the Hamiltonian.



(d)

Figure 4.16 Evolution of co-states and Hamiltonian using the LPS method based 50-node configuration.



(d)

Figure 4.17 Evolution of co-states and Hamiltonian using the CPS method based 50-node configuration.

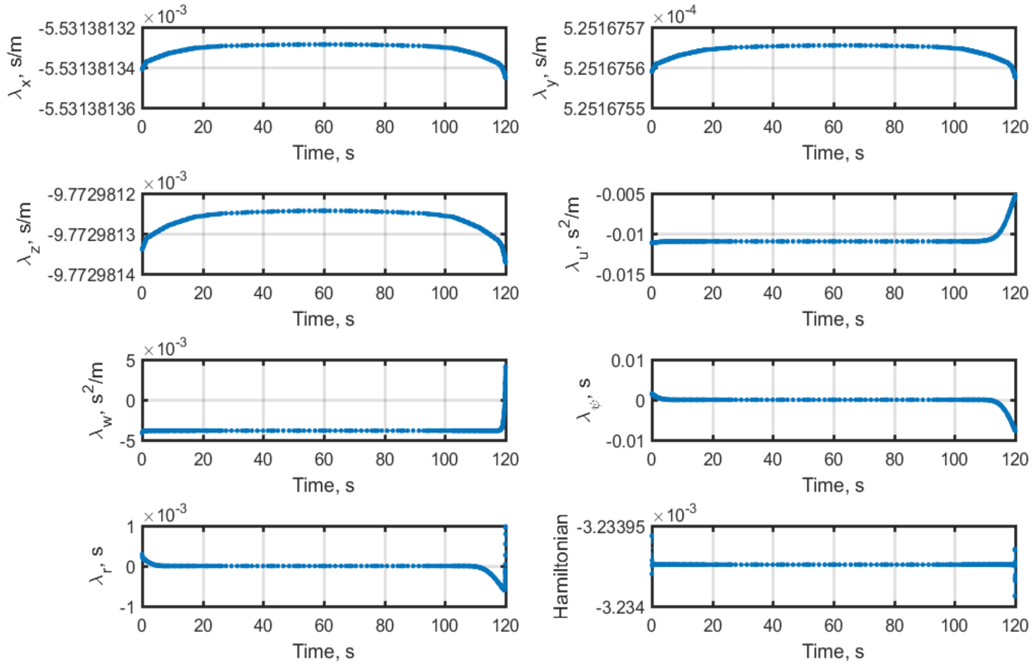


Figure 4.18 Evolution of co-states and Hamiltonian using the hp-AR method.

By comparing the co-states and Hamiltonians of all three PS methods one can conclude that the hp-AR solution is more close to the true optimal solution as it is capable of satisfying the optimality conditions in the more precise numerical sense. This nature provides faster convergence rate of the states and controls achieved by the hp-AR method as compared with the LPS and CPS methods. In essence, these differences between the PS methods in converging to the optimal solution stem from the facts that firstly, the differentiation matrix of the hp-AR method is rectangular and full rank whilst the LPS and CPS differentiation matrices are square and singular. As a result, the hp-AR method can be written equivalently in either differential or implicit integral form, whereas the LPS and CPS methods do not have an equivalent implicit integral form. Secondly, the hp-AR discrete co-state transformation matrix is full rank while the LPS and CPS counterparts are rank-deficient. As a result, the hp-AR co-state approximations show exponential convergence while the LPS and CPS ones demonstrate potential non-converged solutions. Meanwhile, the oscillatory behavior observed in the LPS and CPS co-states are due to oscillatory nature of the null space of the LPS and CPS discrete co-state transformation matrices [154].

4.6.2 Performance Evaluation Using the SITLSP

The aim of this section is to investigate the tractability and realization of the solutions obtained by the PS methods in the previous section. For this purpose, the trajectories generated by the PS methods are propagated to the SITLSP, previously introduced in Chapter 3. In this regard, the fidelity of trajectories in meeting the docking-enabling conditions, introduced in Chapter 3, is also considered. In this scenario, the 50-node configuration of the LPS and CPS are utilized.

Figure 4.19-4-22 show the SITLSP results corresponding to the minimum-energy solutions generated by the LPS, CPS, and hp-AR methods. In these figures, the results generated by the SITLSP are shown by black dashed-lines. Figure 4.19 illustrates that the AUV is able to follow the 3D path generated by all PS methods with a slight error of tracking in the midcourse part and eventually reaches to the pre-specified DS position. Figure 4.20 shows the correspondence between the vehicle heading and heading rate generated by the PS methods and results of the SITLSP. The heading rate still respects the vehicle limitation in instantaneous turn. The final alignment of the vehicle heading with the DS orientation is achieved as a result of meeting the corresponding tolerance of the docking-enabling conditions. Figure 4.21 demonstrates the correspondence between the velocity components generated by the PS methods and those achieved by the SITLSP. Consequently, Figure 4.22 provides a comparison between the evolution of controls in a realistic sense.

As seen from the figures, despite the intentionally introduced discrepancies in controls at the beginning of trajectory and discrepancies in the model used in optimization (3-18)-(3-21) and a full-degree of freedom model of the AUV used within the SITLSP, the controller does a good job that is tracking the interpolated reference trajectory. This result verifies the tractability of PS methods' trajectories by the SITLSP and hence it is possible to implement the solutions on-board a real vehicle for a similar docking operation. A peculiarity of the implementation of the SMC controller with the Traveling WP block is that it has about 2-second lead time (1m distance divided by the average speed of about 0.5m/s) as compared to the interpolated reference trajectory,

- so to accommodate the traveling WP beyond the final point the final approach leg is extended along the DS centerline. That allows terminating simulation at exactly 120s.

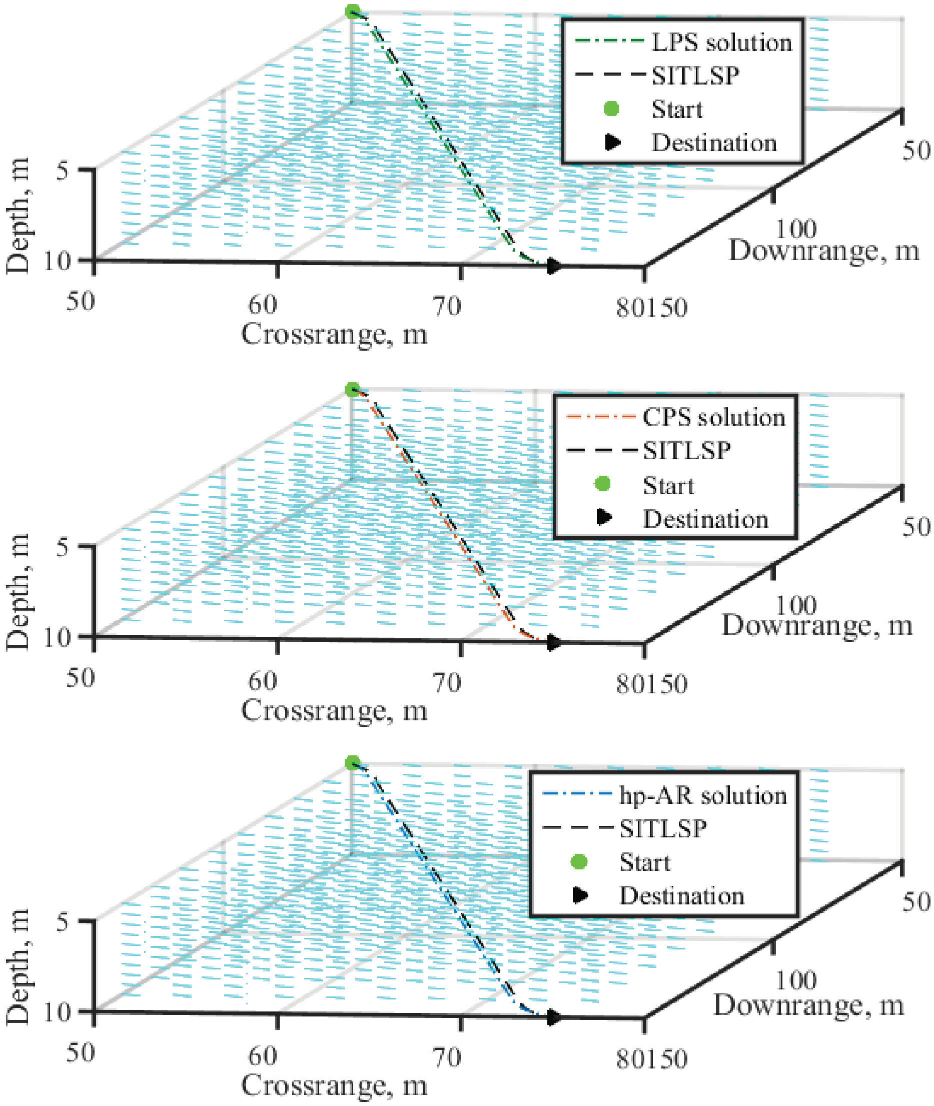


Figure 4.19 3D paths within the SITLSP.

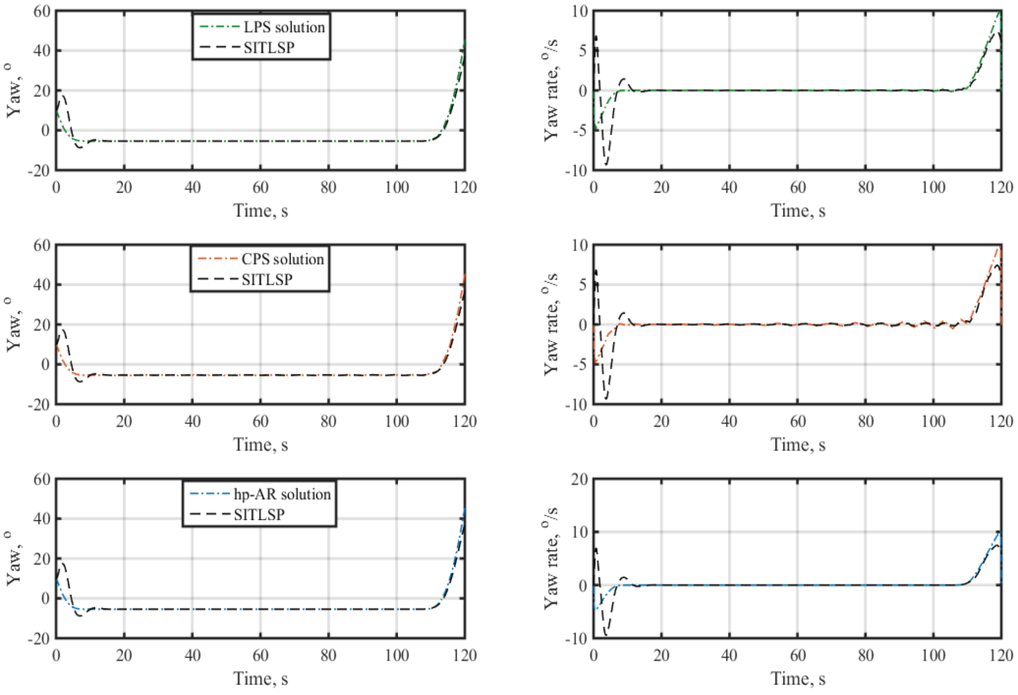


Figure 4.20 Yaw and yaw rate within the SITLSP.

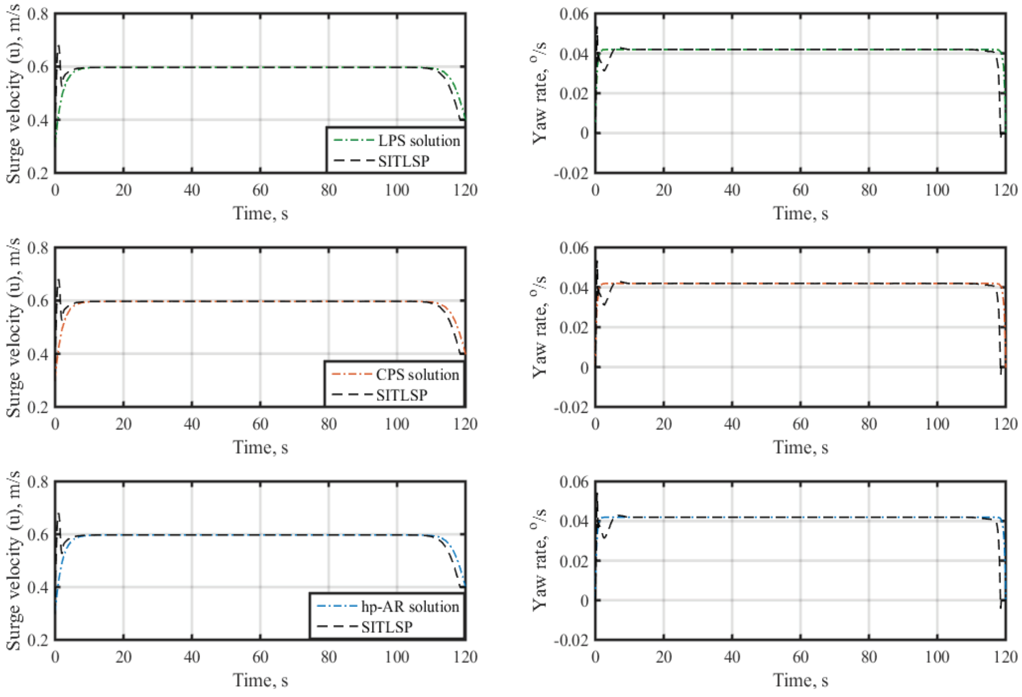


Figure 4.21 Surge and heave velocities within the SITLSP.

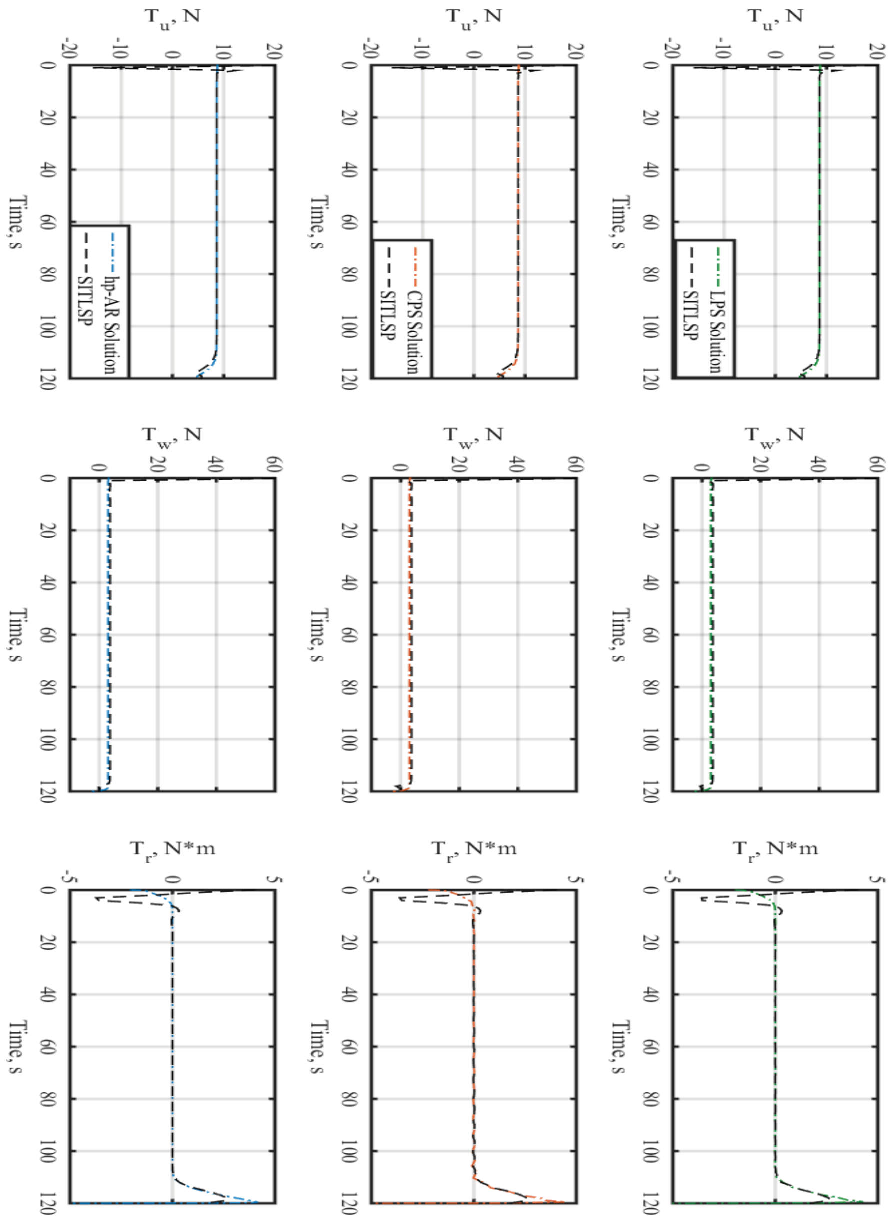


Figure 4.22 Controls within the SITLSP.

Table 4.2 Quantitative evaluation of the PS methods' performance for docking-enabling conditions using the SITLSP.

Method/Metric	***Δη*,m**	***Δψ*,°**	***Δu*,m/s**	***Δw*, m/s**	***Δr*,°/s**
LPS	0.21	8.2	9.87e-05	2.47e-04	6
CPS	0.21	8.35	1.01e-04	2.85e-04	6
hp-AR	0.2	7.7	9.17e-05	6.66e-05	5.9

The occurrence of spicks and phase shifts should be noted in the SITLSP output. These stem from functionality of the Traveling WP guidance block. The proposed block computes the location of the 1-meter-ahead traveling WP and formed an input vector for the SMC block to produce the actual controls. Therefore, the controller tries to minimize the tracking error by accelerating the vehicle beyond what anticipated by the Trajectory generator block (here developed based on the PS methods). This acceleration produces transient differences between desired velocities and those produced by the controller. These artefacts can be compensated by replacing a more advanced Traveling WP block in the SITLSP (is not part of the scope of this thesis).

Table 4.2 quantitatively demonstrate the capability of the PS methods' based guidance system in directing the vehicle from the initial boundary conditions to the docking-enabling conditions. The tolerances corresponding the final errors defined for a successful docking operation are completely met using all three PS methods. In this case, the hp-AR method shows a slightly superior performance comparing to the LPS and CPS methods.

4.6.3 Adding Path Constraints

The aim of this section is to evaluate the docking algorithms with a more challenging scenario. For this purpose, a path constraint of the form (3-43) is added to the TPBVP. From different perspective, this path constraint represents existence of a NFZ in the underlying operating area. In this scenario, three NFZs are defined with the following coordinates and radii:

$$\begin{aligned}
&\mathrm{NFZ}_1:\begin{cases}[x_{nf},y_{nf},z_{nf}]^T=[65,55,6]^T,\mathrm{m}\\ r_{nf}=1.5,\mathrm{m}\end{cases}\\
&\mathrm{NFZ}_2:\begin{cases}[x_{nf},y_{nf},z_{nf}]^T=[101,62,7]^T,\mathrm{m}\\ r_{nf}=2,\mathrm{m}\end{cases}\\
&\mathrm{NFZ}_3:\begin{cases}[x_{nf},y_{nf},z_{nf}]^T=[106,70,7]^T,\mathrm{m}\\ r_{nf}=2.5,\mathrm{m}\end{cases}
\end{aligned} \tag{4-56}$$

In this scenario, the settings for the hp-AR method are kept unchanged (with respect to the previous scenario) but the LPS and CPS methods are configured using 80 collocation points. Even though the 50-node configuration provides a feasible solution in the both LPS and CPS methods, it was, however, observed from the simulation results that the AUV maneuverability was confined within rigid operating margins. Thus, in order to have more flexibility and to make more use of the AUV's full range, the number of collocation point was increased to 80.

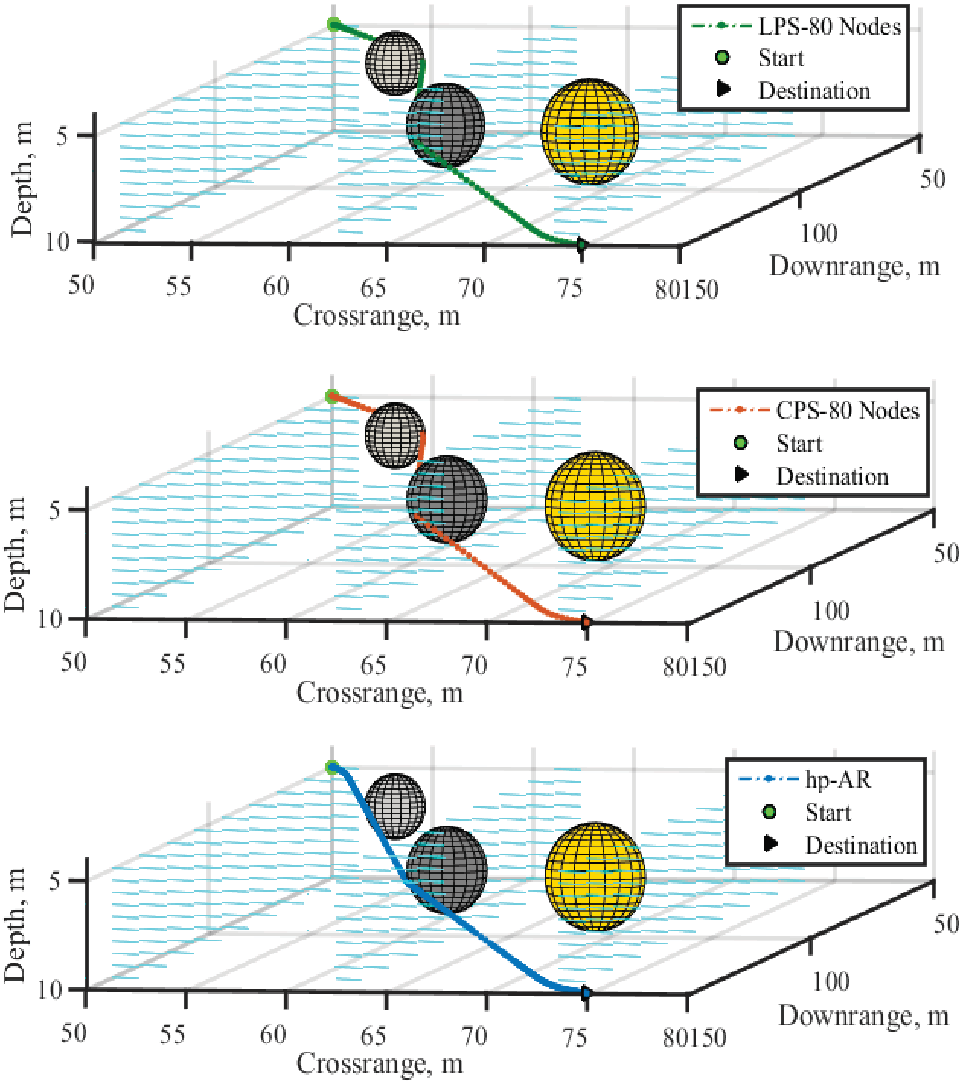


Figure 4.23 Comparison between AUV 3D paths using the PS methods in a cluttered environment.

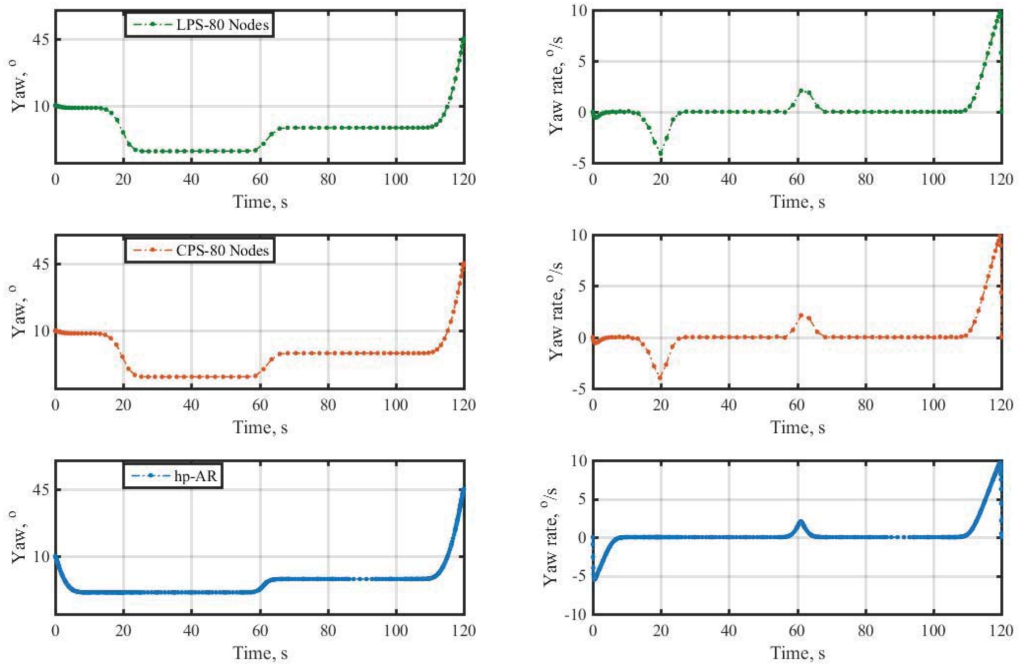


Figure 4.24 Comparison between time histories of yaw and yaw rate using the PS methods in a cluttered environment.

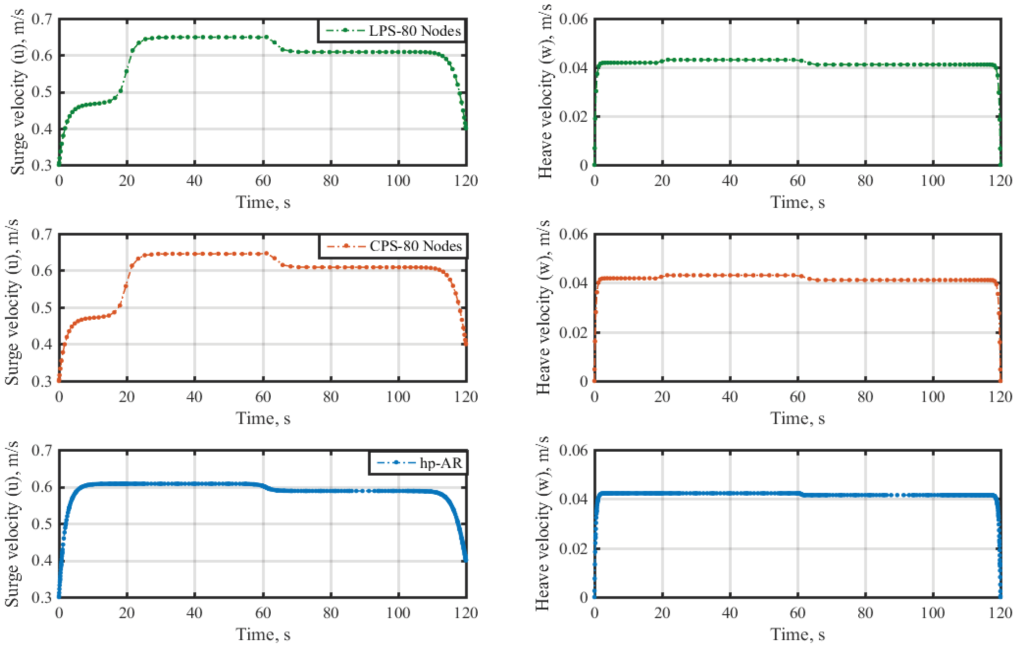


Figure 4.25 Comparison between time histories of surge and heave velocities using the PS methods in a cluttered environment.

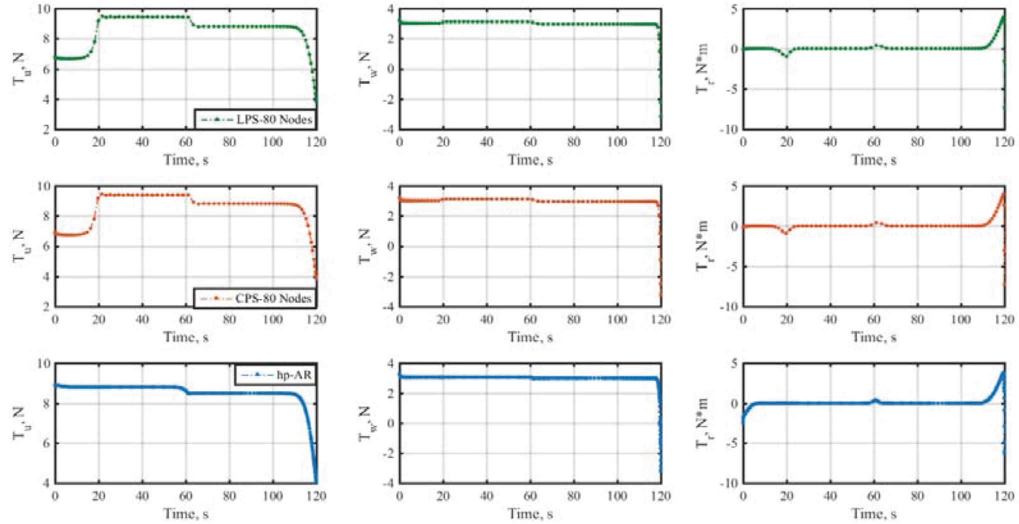


Figure 4.26 Comparison between controls generated by the PS methods in a cluttered environment.

Table 4.3 Quantitative expression of the PS methods' performance.

Method/Metric	***J***	***t_{CPU},s***	***ES%***
LPS	0.2113	516.24	54.03
CPS	0.2111	499.34	54.05
hp-AR	0.2067	8.94*	54.53

* The hp-AR solution utilizes a compiled IPOPT solver as opposed to the *fmincon* interpretative solver.

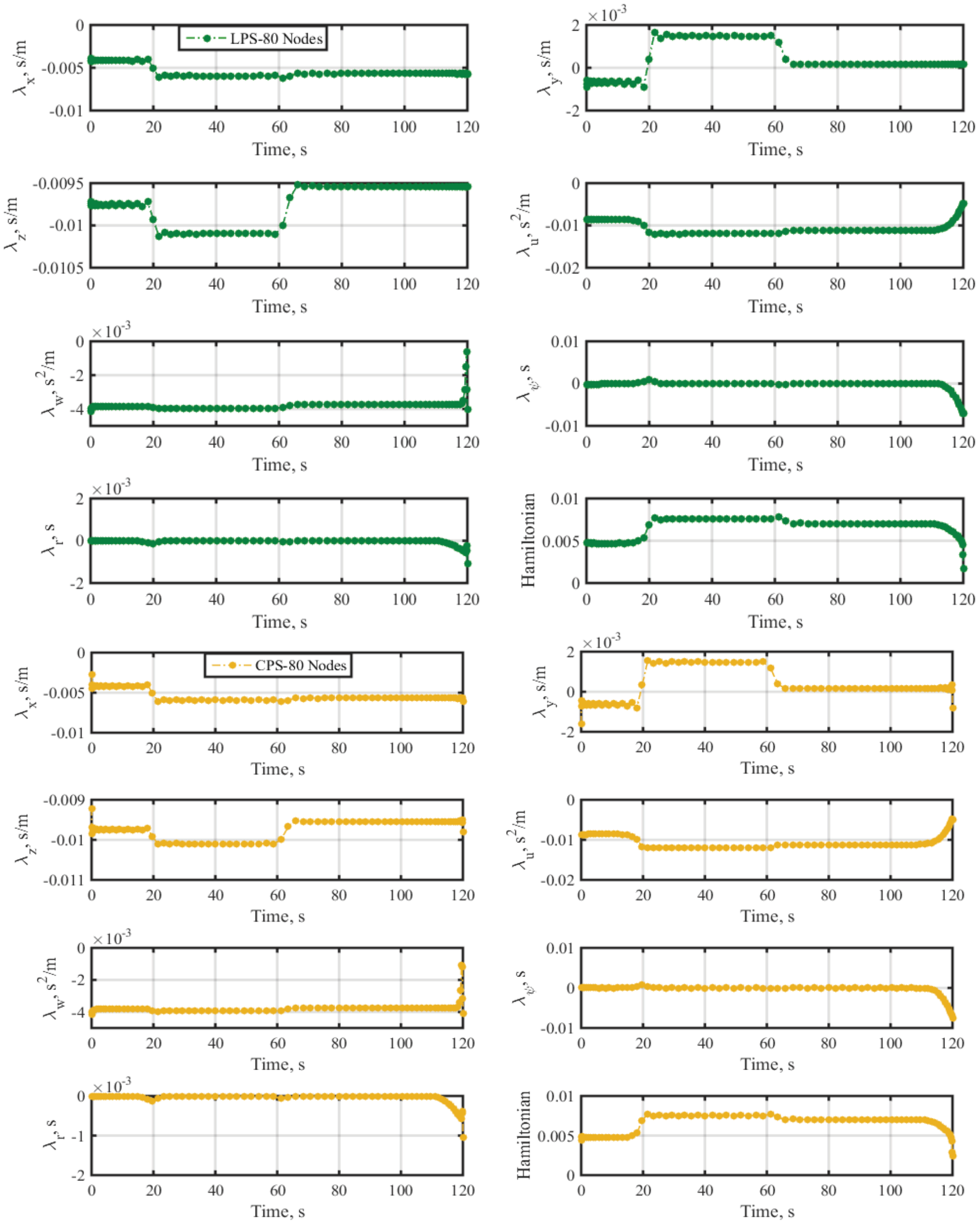
Figure 4.23 shows the collision-free path generated by the PS methods enabling the vehicle to safely transit from its initial position to the DS position. As can be seen from the figure, the solutions generated by the LPS and CPS methods are similar but different with the hp-AR method. This fact can be also seen from Figure 4.24 representing the time history of the yaw and yaw rate in presence of the path constraints. In such a cluttered environment, the PS methods are still able to generate the heading trajectory that is smooth and can satisfy the requirement of the final alignment with the DS orientation (reaching final heading angle 45^{o}). The yaw rate trajectory of all three PS methods does not exhibit any violation along the optimal solution. In Figure 4.25, one can realize how the surge and heave velocities support the collision-free maneuver of the vehicle in presence of the NFZs. Still in presence of the NFZs, the final boundary conditions related to surge and heave are satisfied perfectly. Here the transitions over the surge and heave velocities of the LPS and CPS methods are more than the hp-AR counterparts regarding the different nature of solutions achieved.

Finally, in Figure 4.26 the evolution of controls can be observed. Unlike the bang-bang nature of the minimum-energy performance index, the controls of all three PS methods are not saturated at any time instant leading to considerable save of energy expenditure. In this regard, the controls are still smooth and realizable and do not violate the limitations related to the vehicle’s thrusters.

Table 4.3 provides a quantitative comparison between the solution optimality and computational efficiency of the PS methods for the minimum-energy docking operation in the cluttered environment. As can be seen from the table, using the proposed guidance system, the vehicle operation can be performed with about 54% saving of energy expenditure compared to the situation that the vehicle relies on the maximum thrust values. The hp-AR method shows better performance comparing to the LPS and CPS methods. It is obvious that the better solution quality is obtained with approximately 1.7% computational time used by the LPS and CPS. This stems from the nature of hp-AR method that is able to simultaneously improve accuracy and computational efficiency of the solution. Even though this computational efficiency is to somehow under mercy of the IPOPT solver, however regarding the variable structure of hp-AR method, there exists a proper computational sparsity pattern in its NLP structure and thus the solution can be generated in a computationally efficient manner.

4.6.4 Validation of Solution Optimality

Figure 4.27 demonstrates the time histories of co-states and Hamiltonians for LPS, CPS, and hp-AR methods. In presence of the path constraint, in addition to holding the necessary conditions, the tangential condition associated with the path constraint (3-43) must be held while a trajectory is on the path constraint. It means that in regions where the path constraint is active, a jump condition or discontinuity may be observed in the time history of co-states. This phenomenon is particularly more highlighted in the behavior of spatial coordinates in Figure 4.27. Again, regarding the different nature of hp-AR method, the time history of spatial co-states contains three jumps in all coordinates while for the LPS and CPS five jumps are observable. The λ_ψ and λ_r time histories still show most of the time the solution features a singular arc control (4-15) including jumps as the consequence of the path constraint. The Hamiltonian should show a constant behavior including jump conditions along the optimal trajectory regarding the additional necessary condition of (3-50) and impact of the path constraint respectively. Relying on having a full rank discretized co-state transformation matrix, the hp-AR Hamiltonian trajectory in Figure 4.27 shows more stable behavior compared to the LPS and CPS counterparts and therefore the hp-AR numerical approximation of the TPBVP is more close to the true optimal control solution.



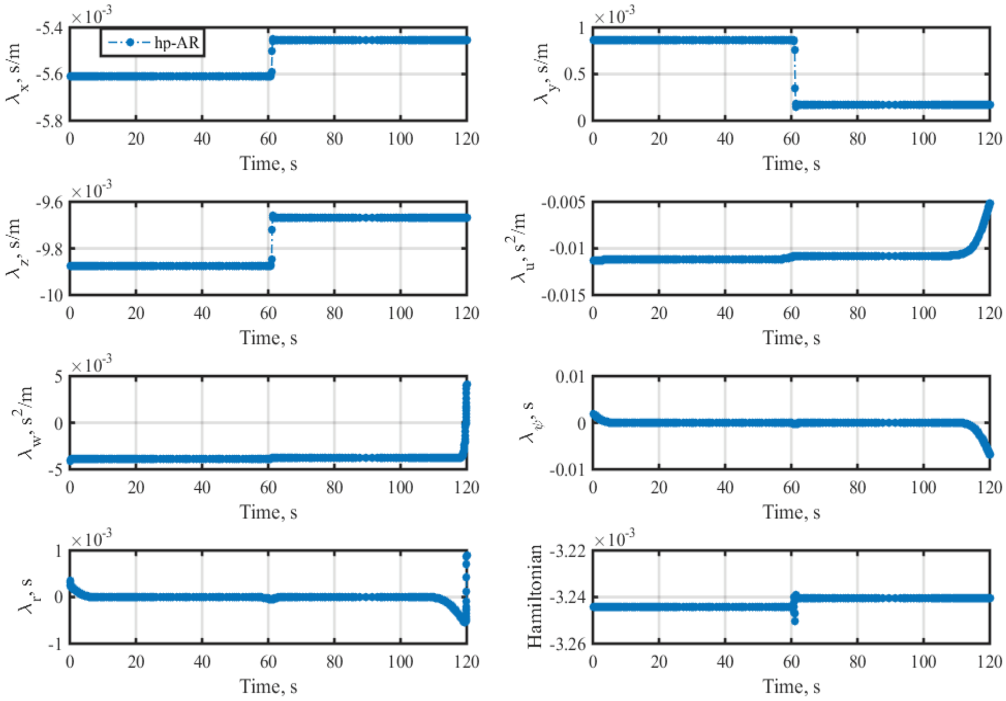


Figure 4.27 Comparison between co-states and Hamiltonians of the LPS, CPS, and hp-AR methods in a cluttered environment.

4.6.5 Validation of Solution Using SITLSP

To check the tractability of the PS method based solutions subjected to the path constraints in a realistic sense, the generated trajectories are again applied to the SITLSP. Figures 4.28-4.31(see Appendix A) demonstrate the results of this test. As shown in Figure 4.28, a collision-free spatial maneuver of the AUV toward the DS position is provided by the STILSP for all three PS methods even though there exists a slight tracking error between the PS method's solutions and the SITLSP ones. Figure 4.29 shows a significant correspondence between the simulated and actual solutions even in presence of the NFZs. This correspondence, makes the vehicle able to employ a smooth steering motion during the docking operation and finally align its head with the DS orientation without violating the yaw rate trajectory constraint. Given the close match between the SILSP and simulated performances in Figure 4.30, one can conclude that the surge and heave velocities generated by the PS methods are realizable for the vehicle. The spike at the beginning and the phase shift at the end of these trajectories are the consequence of the Traveling WP block as described before.

Figure 4.31 illustrates the realization of controls for activation of the vehicles' main propeller, lateral and vertical thrusters. The smooth and non-saturated controls guarantee the minimum-docking performance in practice. It is noteworthy to mention that the spike of about 40 N at the beginning of vertical thruster profile is still acceptable for the vehicle. This stems from the fact that the TPBVP is subjected to more conservative bounds on the vehicle actuators while in the SITLSP these bounds are larger (approximately twice larger). This design always provides implementable controls for the vehicle maneuverability to suit the different circumstances.

Table 4.4 quantitatively indicates that solution obtained by the PS methods enables successful docking operation as all metrics meet the tolerances of docking-enabling conditions. In this situation, the hp-AR method provides the smallest error with respect to the final DS position when compared with the LPS and CPS ones. The rest of the errors are approximately of the same scales.

Table 4.4 Quantitative evaluation of the PS method performances for docking-enabling conditions in a cluttered environment using the SITLSP.

Method/Metric	$\Delta\eta$, m	$\Delta\psi$, °	Δu, m/s	Δw, m/s	Δr, °/s
LPS	0.39	6.03	6.24e-05	2.45e-05	5.18
CPS	0.38	6.11	6.35e-05	3.04e-06	5.21
hp-AR	0.19	7.44	8.40e-05	8.62e-05	5.68

4.7 Simulation Results for the 10-State Formulation

In Section 4.6, it was realized that despite the optimal performance of the PS methods in minimum-energy docking operations, one of the major requirements of this thesis that is smooth departure and arrival of the vehicle to the DS was not fulfilled based on the 7-state solutions. In Section 4.2, it was shown that in order to provide smooth departure of the AUV from the initial conditions and arrival into the DS, or equivalently to set the boundary conditions on the controls, the 10-state problem formulation can be used. To do so, the original system is expanded to include three more equations converting the original controls to the three new states (4-16). In this realization, all boundary conditions and constraints corresponding to the vehicles' actuators and state are kept unchanged and bounds on the new controls are established as $\upsilon_u^{\max} = \upsilon_w^{\max} = 0.5$ N/s and $\upsilon_r^{\max} = 0.5$ Nm/s. In this scenario, also, the initial guesses used to assure and speed up the convergence of the 10-state solution was obtained using the IDVD method discussed in the next chapter. Meanwhile, the LPS and CPS methods are utilized in a 50-node configuration while the hp-AR is set with 20 collocation points per mesh and 4 segments in each mesh interval (hp-(4,20)).

Figures 4.32-4.36 illustrate the results of numerical solution for the same TPBVP with the 10-state problem formulation. In Figure 4.32, the 3D path enables the spatial maneuver of the vehicle from its initial position to the DS position. The hp-AR path seems more condense as a result of increasing the number of collocation points per mesh. Figures 4.33 shows the time history of yaw and yaw rate trajectories and Figure 4.34 demonstrates the vehicles forward and vertical velocity profiles. Both figures indicate that the 10-state formulation is able to satisfying of boundary conditions and respecting the vehicle's constraints. These figures besides demonstrate a big difference in the beginning and at the end of the trajectory as compared to the equivalent figures obtained in the 7-state representation (Figures 4.3-4.4, 4.7-4.8, and 4.11-4.12 for the LPS, CPS, and hp-AR methods respectively). Figure 4.35 confirms that the boundary conditions (4-7), (4-8) can now be satisfied assuring a smooth departure from the initial point and smooth arrival to the DS. The time histories of the new controls υ_u , υ_w, and υ_r, are shown in Figure 4.36. Indeed, according to (4-19)-(4-20) they feature a bang-singular-bang optimal control. This is more obvious in the new controls generated by the hp-AR method.

Table 4.5 quantitatively demonstrate performance of the PS methods in the minimum-energy docking operation based on the 10-state formulation. As seen from the table, an attempt to satisfy the boundary conditions on the PS methods' controls by increasing the number of states inevitably leads to increase of the performance index value. Comparing to the 7-state solution, a relative 1% decrease in the energy saving is observed with all three PS methods. As the number of decision parameters is increased (up to 30%), the required CPU time is increased significantly and therefore the computational efficiency decrease to a great extent even by using the compiled

IPOPT solver in the hp-AR method. This makes real-time operations and consequently closed-loop guidance implementation on-board a real-vehicle impractical.

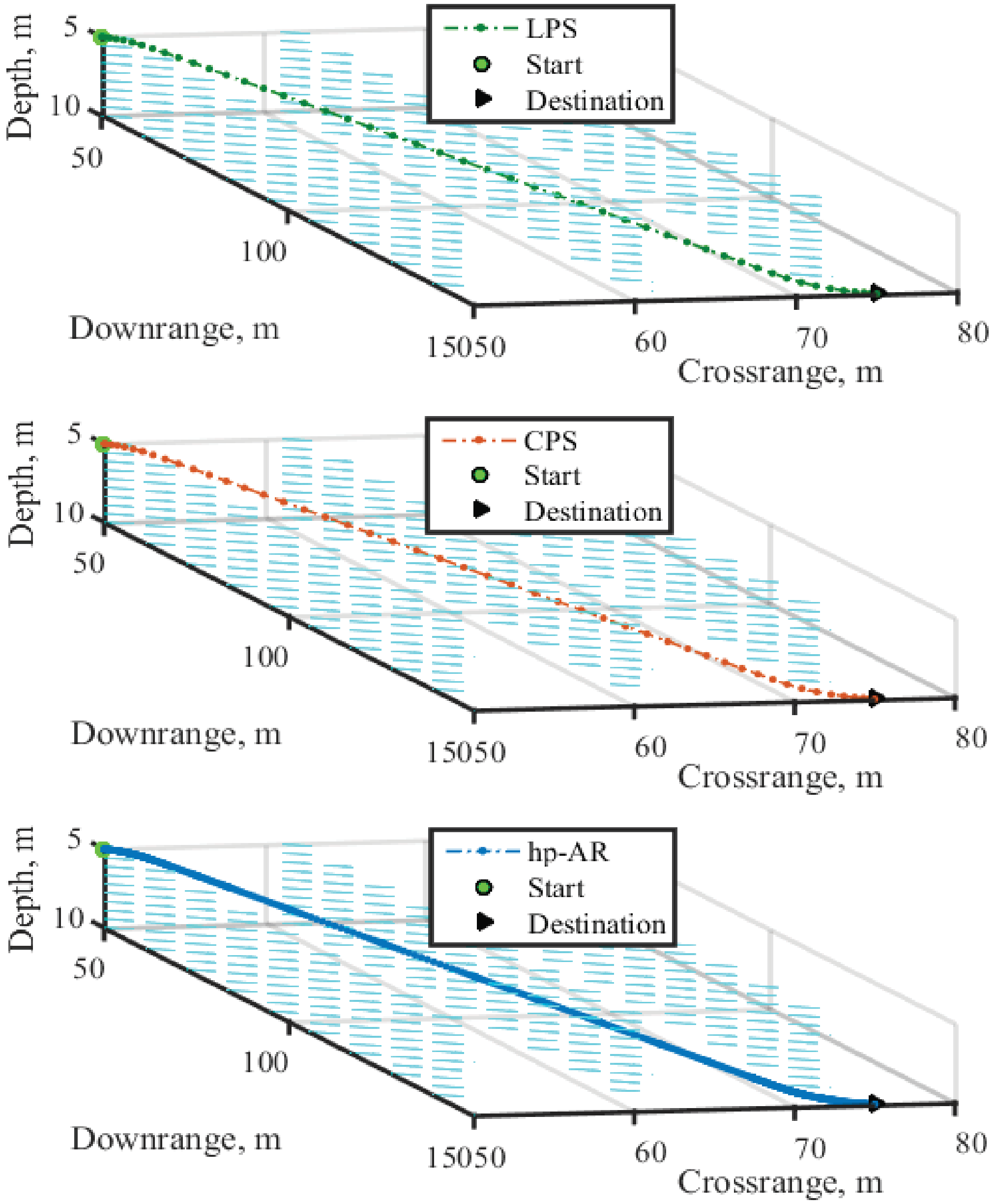


Figure 4.32 Comparison between the PS methods' 3D paths based on the 10-state solution.

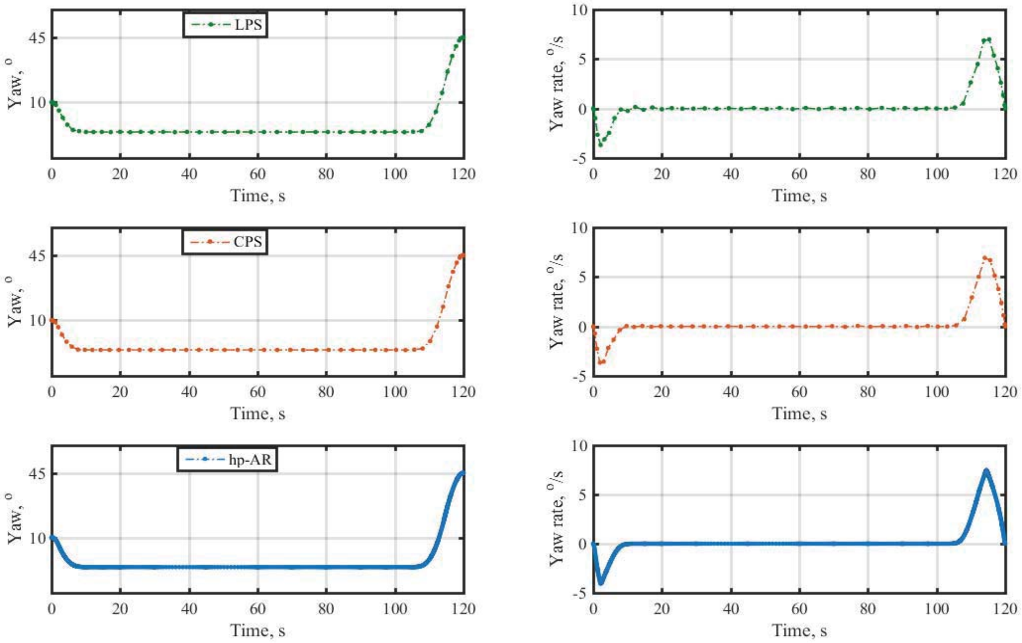


Figure 4.33 Comparison between the PS methods' time history of yaw and yaw rate based on the 10-state solution.

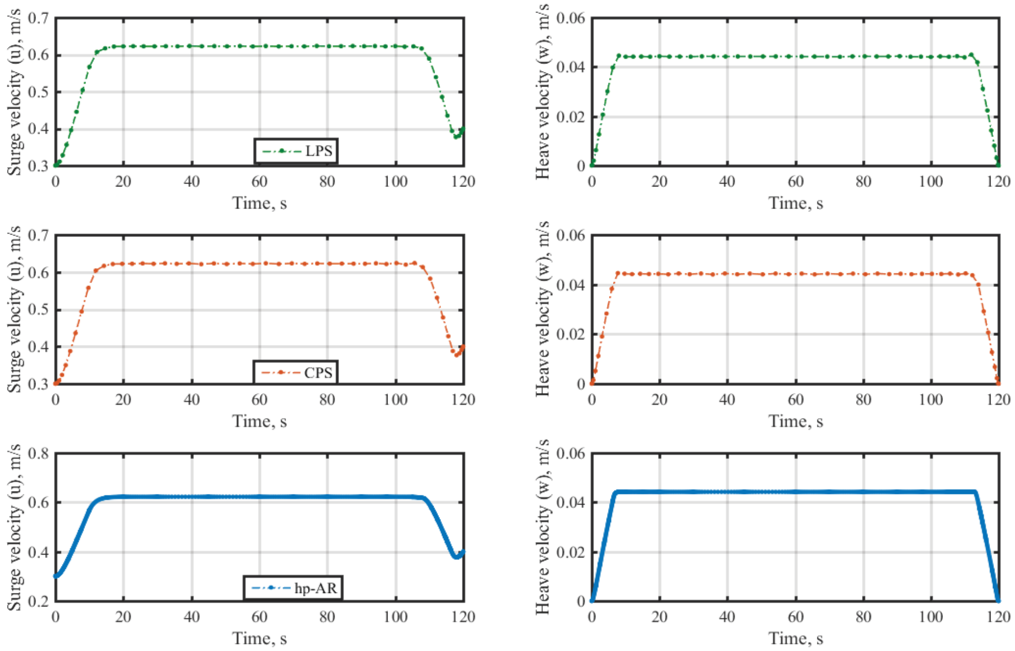


Figure 4.34 Comparison between the PS methods' surge and heave velocities based on the 10-state solution.

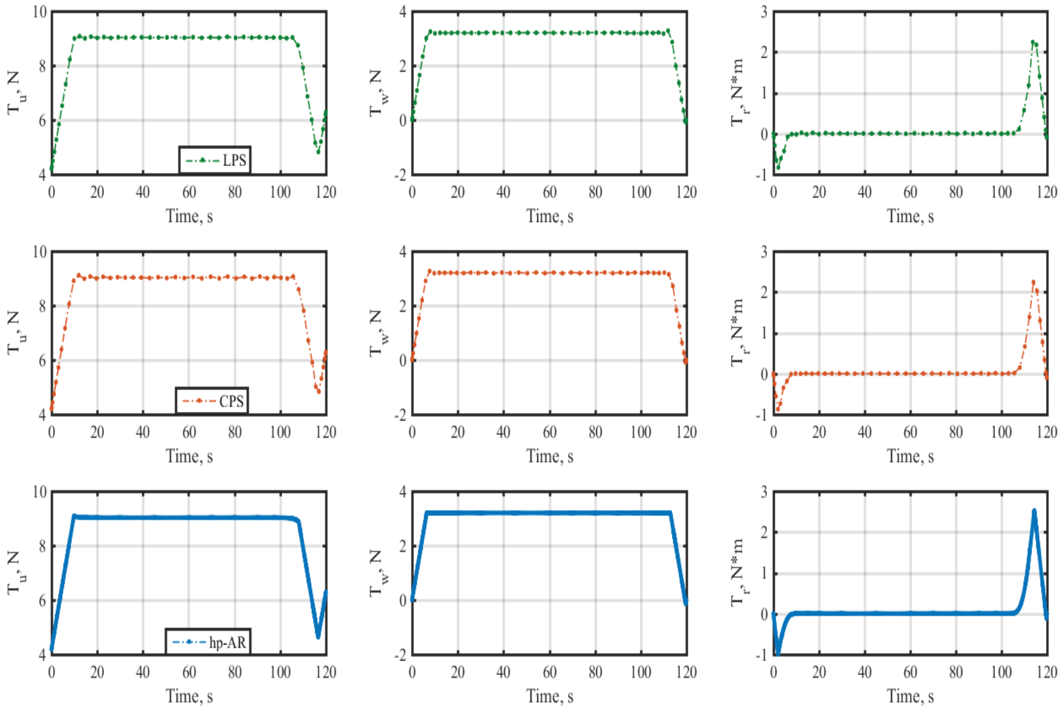


Figure 4.35 Comparison between the PS methods' new states based on the 10-state solution.

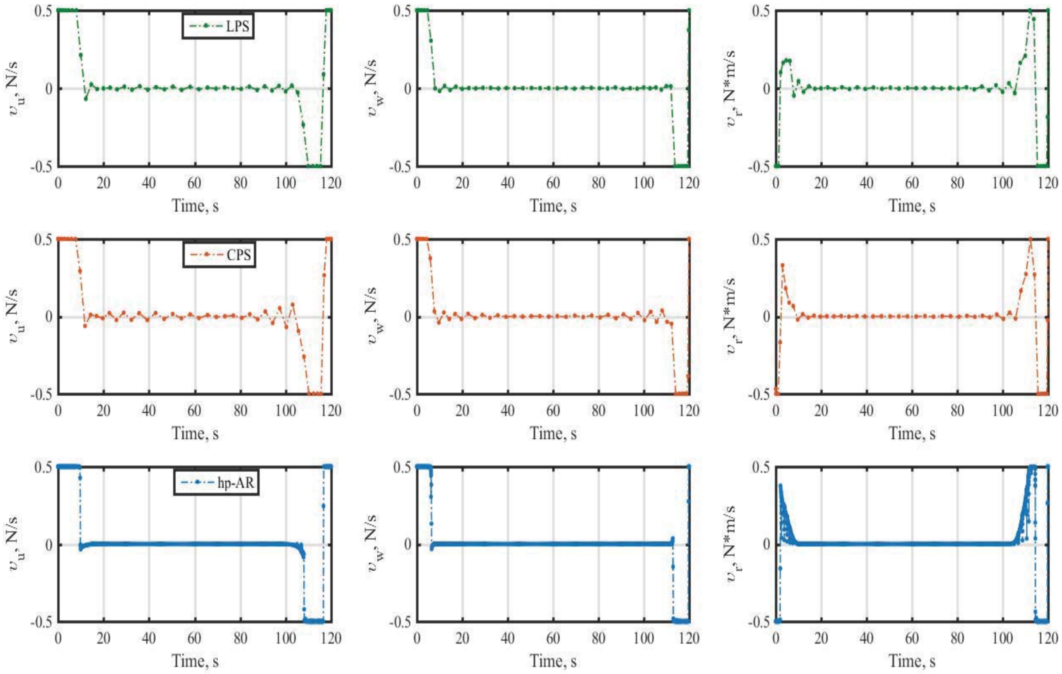


Figure 4.36 Comparison between the PS methods' new controls based on the 10-state solution.

Table 4.5 Quantitative expression of the PS methods' performance based on the 10-state formulation.

Method/Metric	***J***	t_{CPU}*,s*	***ES%***
LPS	0.2107	1160	54.09
CPS	0.2107	733.68	54.09
hp-AR	0.2108	39.72*	54.08

* The hp-AR solution utilizes a compiled IPOPT solver as opposed to the *fmincon* interpretative solver.

4.7.1 Validation of Solution Optimality

For evaluating the optimality of solutions obtained based on the 10-state formulation, the co-state and Hamiltonian trajectories are derived. Figures 4.37-4.39 show the time histories of co-states including three new co-states introduced in (4-19) and Hamiltonian. Similar to the 7-state co-state trajectories, the spatial co-states λ_x, λ_y, and λ_z in Figures 4.37-4.39 indicate that differential equations (3-38) hold in the numerical sense. More precisely, the spatial co-states of the LPS method in this realization show a more oscillatory behavior around the optimal solution as compared with the CPS and hp-AR methods. This is due to LPS deficiency in mapping between the KKT and first-order optimality conditions. In contrast, the hp-AR spatial co-states show a stable behavior with a good numerical precision in holding (3-38). The co-states λ_ψ and λ_r in Figures 4.37-4.39 indicate that most of the time the solution features a singular arc control (4-15). The new co-states λ_{Tu}, λ_{Tw}, and λ_{Tr} should hold (4-19) in a numerical sense. In this case, the LPS and CPS methods generate roughly the same trajectories but different with the hp-AR ones. In Figures 4.37 and 4.38 the Hamiltonian is constant along the singular arc with LPS and CPS solutions. Comparing to the 7-state Hamiltonian, the LPS and CPS Hamiltonians depict more variations along the optimal trajectory. In this case also, the optimization routine has difficulty in numerically solving the problem beyond that. In contrast, the Hamiltonian associated with the hp-AR solution shows a stable and constant behavior along the optimal trajectory except a jump in order of 10^{-4} at end of the Hamiltonian.

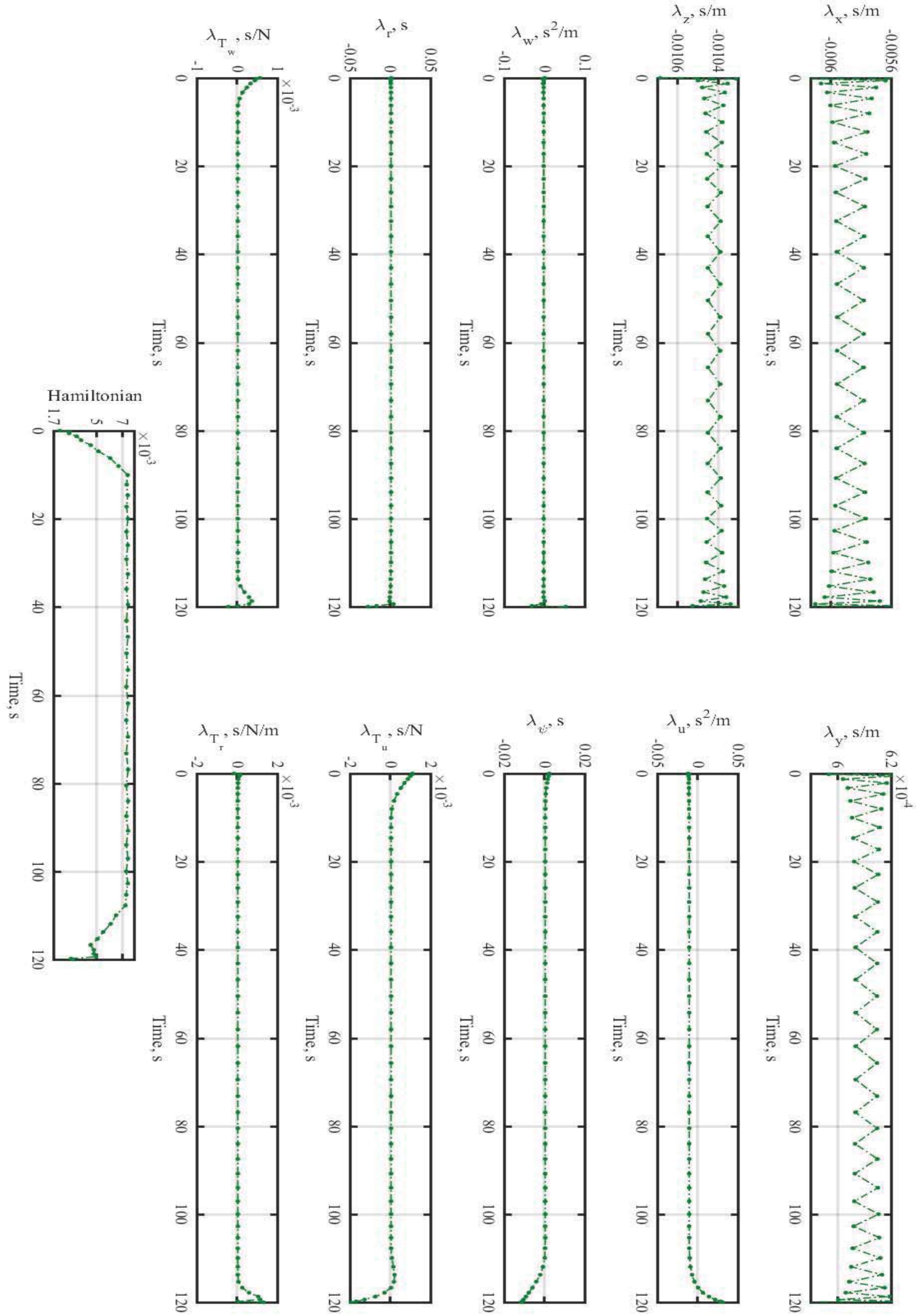


Figure 4.37 Evolution of co-states and Hamiltonian based on the LPS 10-state solution.

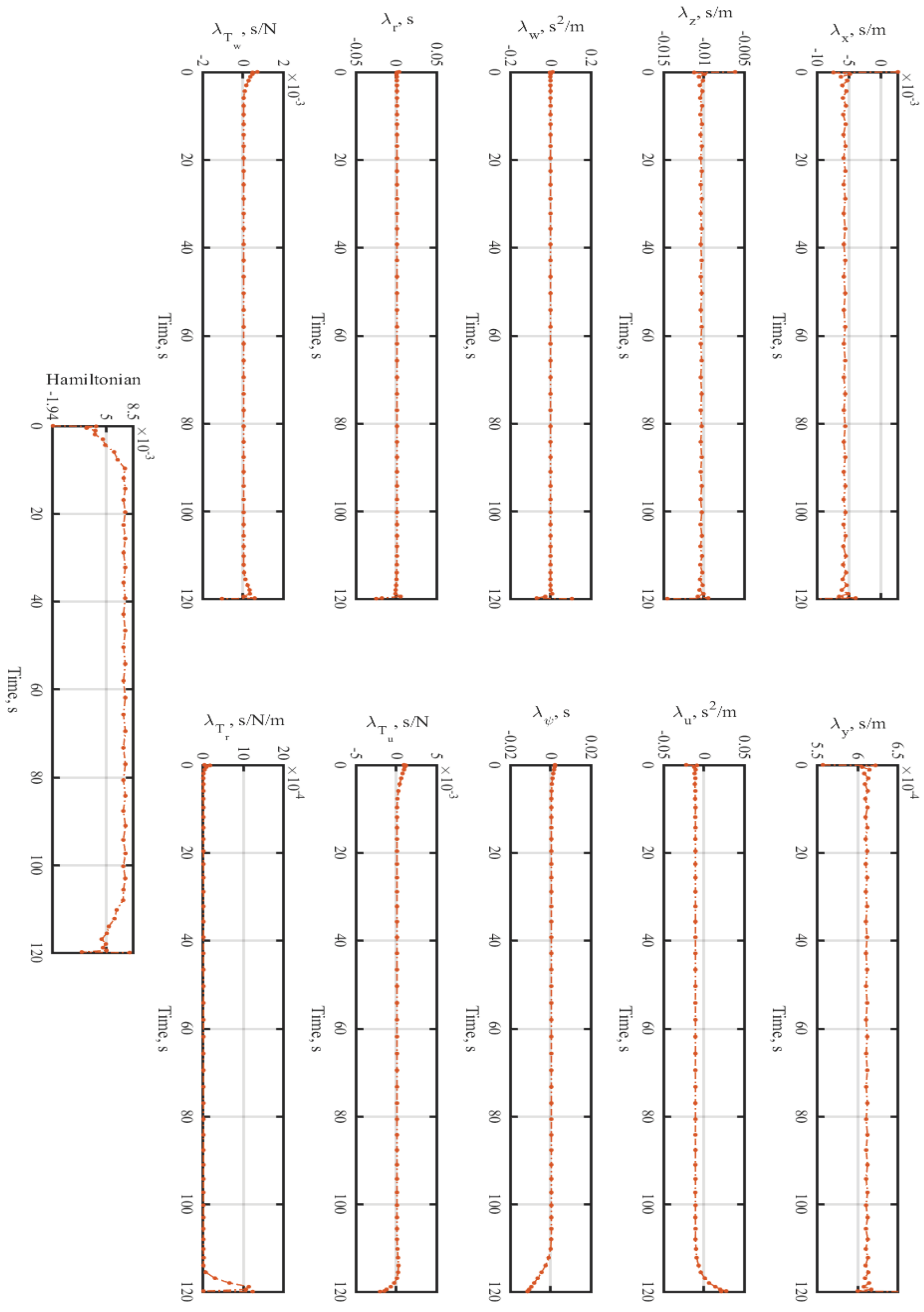


Figure 4.38 Evolution of co-states and Hamiltonian based on the CPS 10-state solution.

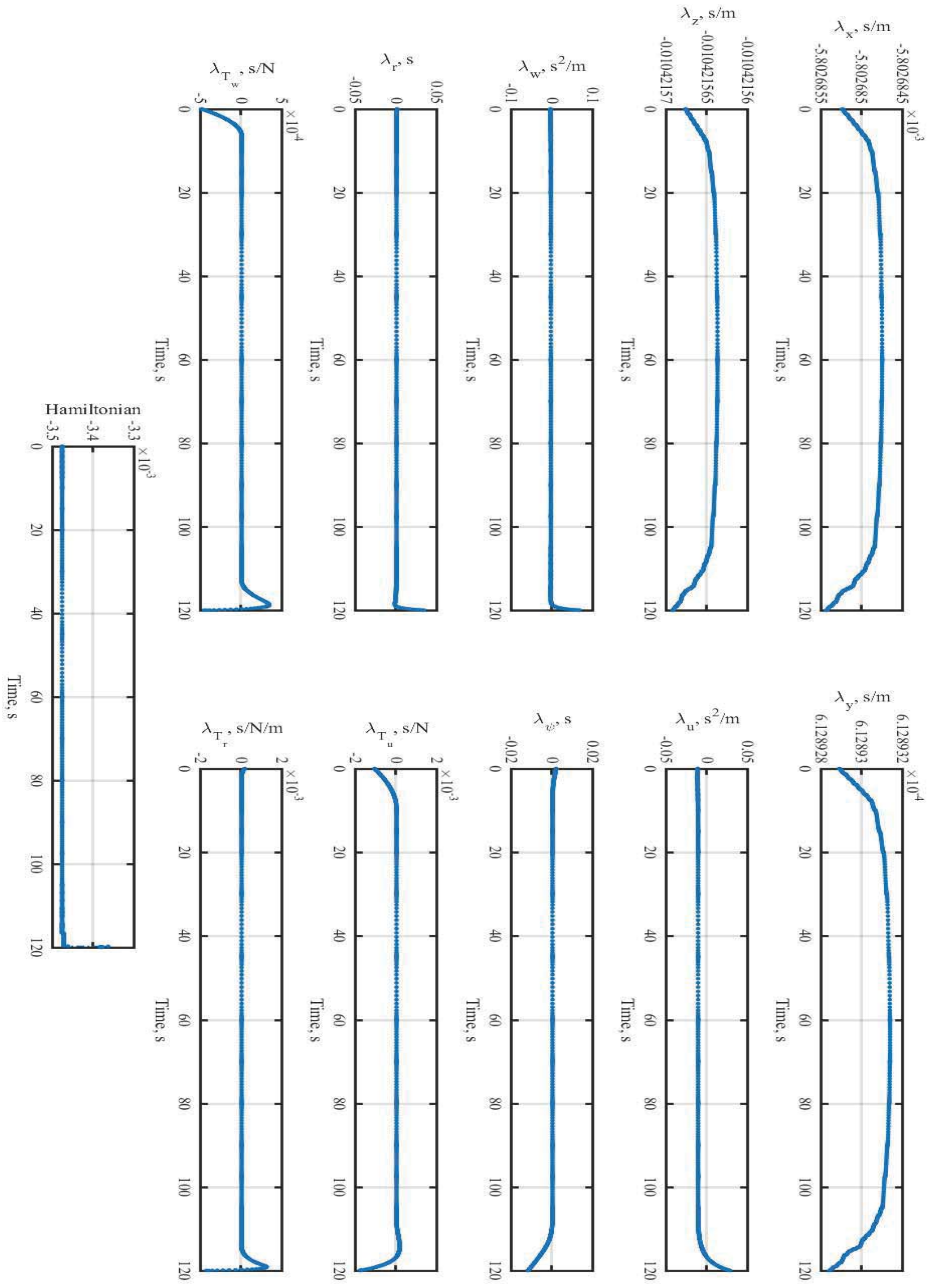


Figure 4.39 Evolution of co-states and Hamiltonian based on the hp-AR 10-state solution.

In the case of the 10-states formulation, the co-states and Hamiltonians of the hp-AR method are more close to the true optimal solution as it is capable of satisfying the optimality conditions in the more precise numerical sense. This nature provides faster convergence rate of the states and controls achieved by the hp-AR method as compared with the LPS and CPS methods. These differences between the PS methods in converging to the optimal solution stem from the facts that the hp-AR discrete co-state transformation matrix is full rank while the LPS and CPS counterparts are rank-deficient. As a result, the hp-AR co-state approximations show exponential convergence while the LPS and CPS ones demonstrate potential non-converged solutions.

4.7.2 Validation of Solution Using SITLSP

In this part, the realization of 10-state solution obtained by the PS methods is investigated using the SITLSP. Figures 4.40-4.43(see Appendix B) show the results obtained by the SITLSP. As indicated in the figures, there exists good correspondence between the simulated and actual solutions, except the jumps and phase shifts that are due to the Traveling WP block performance. The new states (4-16) are tractable for the SMC controller as a result of good match between the SITLSP controls and simulated solutions in Figure 4.42.

To check the realization of solution quantitatively, the results corresponding to the docking-enabling conditions are tabulated in Table 4.6. As inferred from the table, the final position error of all three PS methods are equal but considerably larger comparing to the 7-state solution. Although there exists a significant improvement on the final heading and yaw rate errors of all three PS methods comparing to the 7-state counterparts, nevertheless, to meet the final boundary conditions on the controls, the 10-state solution scarifies some performance indices such as final position error and computational time.

Table 4.6 Quantitative evaluation of the 10-state PS method performances for docking-enabling conditions using the SITLSP.

Method/Metric	$\Delta\eta,_{\mathrm{m}}$	$\Delta\psi,^{\mathrm{o}}$	$\Delta u,_{\mathrm{m/s}}$	$\Delta w,_{\mathrm{m/s}}$	$\Delta r,^{\mathrm{o/s}}$
LPS	0.68	0.51	2.12e-06	3.28e-05	1.81
CPS	0.68	0.53	2.22e-06	3.56e-05	1.83
hp-AR	0.68	0.24	9.03e-07	1e-04	1.56

4.8 Robustness Assessment using Monte Carlo Simulations

To examine the performance of the PS method based docking guidance system under uncertain conditions, Monte Carlo (MC) simulations of 200 randomly generated test samples are conducted and analyzed. To generate the random samples, the uniform distribution is utilized

$$\wp \sim unif\,(\ell^b, u^b), \quad \wp \in [\ell^b, u^b] \tag{4-57}$$

$$f(\wp) = \begin{cases} (u^b - \ell^b)^{-1} & if \;\; \ell^b \le \wp \le u^b \\ 0 & otherwise \end{cases} \tag{4-58}$$

here $\wp$ is the random sample, ℓ^b and u^b are the lower and upper bounds, respectively, that determine the pre-specified area of sample distribution. Equation (4-58) represents the corresponding probability density function (PDF) shown by $f(\wp)$.

In this section, the 7-state formulation is utilized for the PS methods. The 50-node configuration of the LPS and CPS methods are utilized; the hp-AR method is configured using 4 collocation points per mesh and two segments in each mesh interval.

4.8.1 Robustness Assessment with Respect to Random Initial Pose

In the first test, for a set of 200 MC trials, the initial pose of the vehicle $\eta_0 = [x_0, y_0, z_0, \psi_0]^T$ is changed to $\eta_{\wp_0} = [\wp^{x_0}, \wp^{y_0}, \wp^{z_0}, \wp^{\psi_0}]^T$. A 10% uncertainty is applied to the hydrodynamic coefficients used in the SITLSP while the other conditions are kept unchanged. Four performance metrics of normalized cost (J), execution time (t_{CPU}), final position error ($\Delta\eta$) and final heading error ($\Delta\psi$) are utilized; based on these metrics, the impact of uncertain initial pose conditions on the docking guidance system is considered. In this test, the amount of uncertainty applied to the initial pose vector is described by the following relations: $\wp^{x_0} \in x_0 \pm 5$ m, $\wp^{y_0} \in y_0 \pm 5$ m, $\wp^{z_0} \in z_0 \pm 2$ m, $\wp^{\psi_0} \in \psi_0 \pm 20^\circ$.

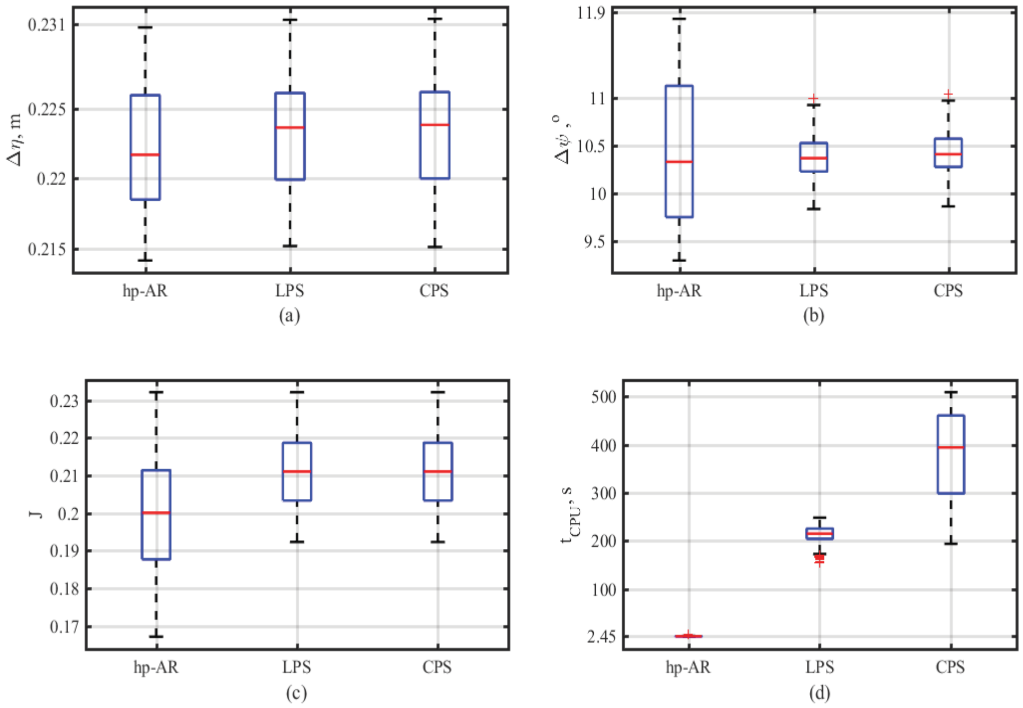


Figure 4.44 The PS methods' performance based on the Monte Carlo simulation under initial pose variations.

Figure 4.44 shows the results of the first test. The PS methods in general demonstrate robust performance in the presence of initial pose uncertainty and achieve good consistency in terms of final position and heading errors when compared with the results presented in Tables 4.2. For all 200 MC trials, the docking-enabling conditions are satisfied and the vehicle is able to complete the docking operation successfully.

More specifically, the hp-AR method shows a bit better performance in terms of position error comparing to the LPS and CPS methods whereas it shows more variations on the bound of the heading error comparing to the LPS and CPS methods.

From the variation of *J* in Figure 4.44, it can be inferred, however, that the LPS and CPS methods expend more control effort than the hp-AR method to achieve a marginally reduced final heading error. The heuristic behind the hp-AR method, however, provides a balanced compromise between minimizing energy expenditure and meeting the tractability metrics. From the variation of *J* associated with the hp-AR method, it can be inferred that the hp-AR method is more capable of finding global optima than the LPS and CPS methods. More importantly, despite the initial pose uncertainty, the execution time for the hp-AR method is significantly small making real-time implementation much more feasible than with the LPS and especially the CPS methods.

4.8.2 Robustness Assessment with Respect to Random DS Pose

In the second test, uncertainty is applied to the DS pose vector $\eta_d = \left[x_d, y_d, z_d, \psi_d\right]^T$ and the rest of the conditions kept unchanged. Similar to the first experiment, 200 random samples are generated using (4-57) and $\eta_{\wp_d} = \left[\wp^{x_d}, \wp^{y_d}, \wp^{z_d}, \wp^{\psi_d}\right]^T$ is constructed. The amount of uncertainty applied on the DS pose parameters is defined as follows: $\wp^{x_f} \in x_f \pm 5$ m, $\wp^{y_f} \in y_f \pm 5$ m, $\wp^{z_f} \in z_f \pm 2$ m, $\wp^{\psi_f} \in \psi_f \pm 20$.

Figure 4.45 illustrates the performance results of the PS methods for variations of the DS pose vector across all 200 MC runs. From a quick inspection of the performance results, it can be observed that all three PS methods are capable of directing the vehicle to the DS so that the tolerance associated with the final position error is met. However, for some final pose samples, the PS methods cannot satisfy the tolerance assigned for the final heading error; statistically speaking, 12 cases with the hp-AR, 31 cases with the LPS, and 37 cases with the CPS are reported. Referring to part (c) of the figure, one can realize that LPS method shows better robustness in terms of providing docking operations with minimum-energy expenditure when compared with the CPS and hp-AR methods. It can be observed from part (d) of the figure that while the hp-AR method is again robust in terms of execution time, both LPS and CPS methods incur a huge computational burden with respect to computing the optimal solution.

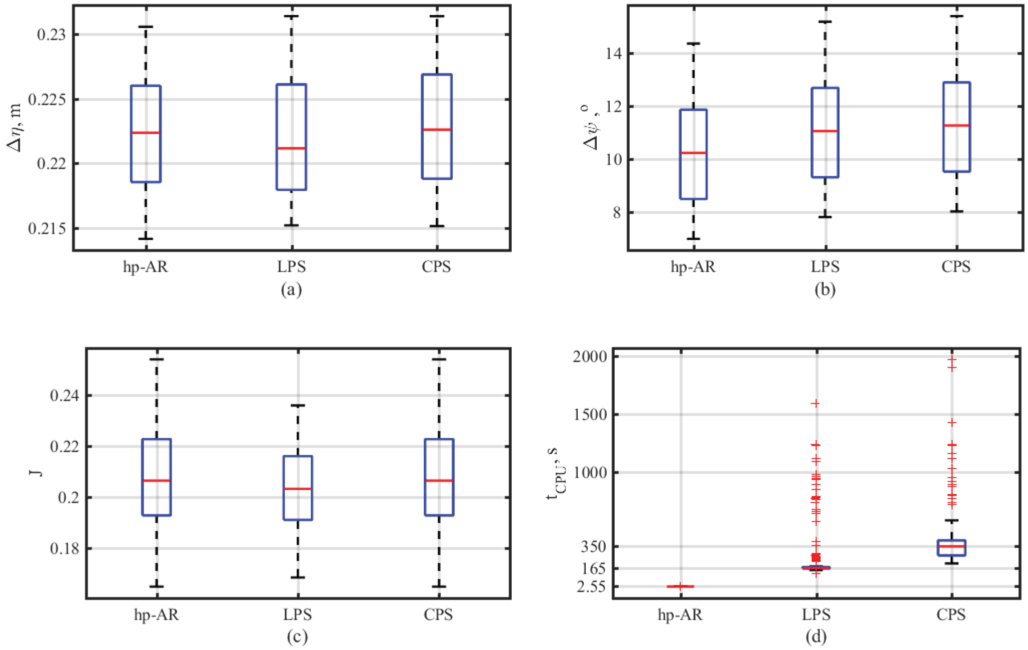


Figure 4.45 The PS methods' performance based on the Monte Carlo test under DS pose variations.

4.8.3 Robustness Assessment with Respect to Random Current Disturbances

In the third test, it is desired to investigate the impact of strong cross-currents on the overall performance of the PS methods. To this end, the current components $V_c = \left[c_x, c_y\right]^T$ (defined in Section 4.6) are changed to the $V_{c_\wp} = [\wp^{c_x}, \wp^{c_y}]^T$ where $\wp^{c_x}$=0 m/s and $\wp^{c_y}$~0.2-0.4 m/s for all MC simulations. The other conditions are kept unchanged. Figure 4.46 depicts the results of the MC simulation for this test. In this test, for all 200 MC trails, the PS methods are able to meet the docking-enabling conditions. The hp-AR method shows superior performance comparing to the LPS and CPS methods in terms of the final position and heading errors. It is noteworthy to mention that, approximately twice as much energy expenditure per second (compared with the previous test) is required to overcome the strong cross-currents for an acceptable docking operation, no matter which PS method is utilized (part (c)). Part (d) of Figure 4.46 indicates the computational efficiency of the PS method for this test. Again, the hp-AR method shows high computational efficiency and robustness with respect to the uncertain cross-currents, while the LPS and CPS methods consume several orders of magnitude more computational time than the hp-AR method. This heavy computational burden is most significant when the CPS method is utilized. From this test one can also clearly observe the direct relationship between the impact of cross-currents and low-thrust docking maneuverability. In other words, the stronger the cross-currents, the more thrust is required for a persistent docking operation.

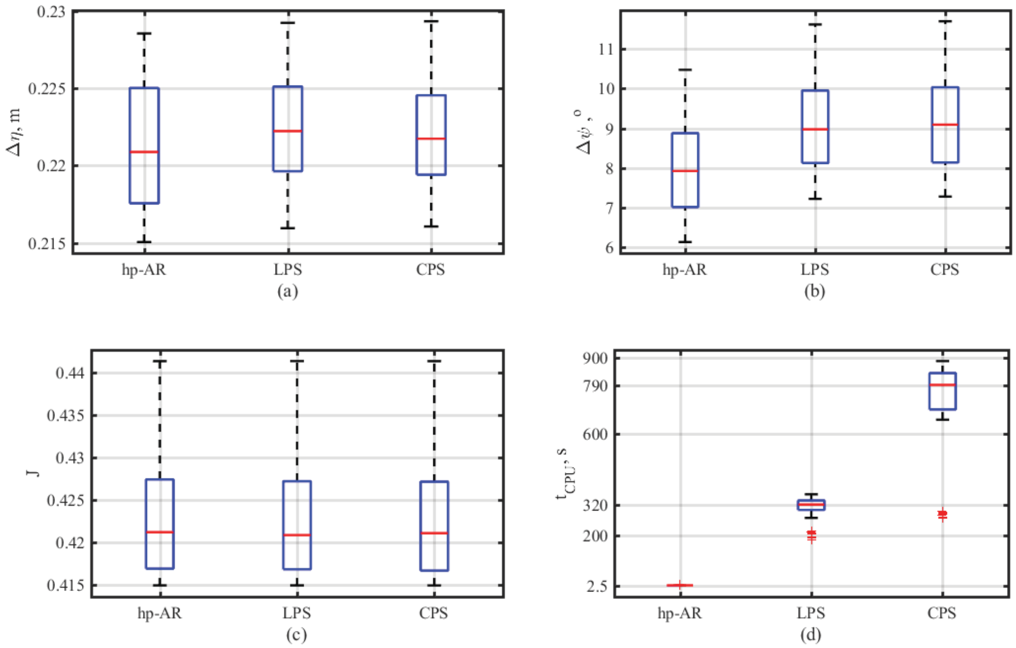


Figure 4.46 The PS methods' performance based on the Monte Carlo test under cross-current variations.

4.9 Chapter Summary

In this chapter, a comprehensive study of solutions obtained using the LPS, CPS and hp-AR methods as applied to the docking guidance system was provided. It was confirmed through a series of analysis that the PS methods can provide optimal trajectory close enough to the true optimal solution and thus can be used as benchmark solutions. This feature stems from their global collocation procedures they adopt and enforcing the first-order optimality conditions by estimating co-states and Hamiltonian trajectories upon the particular co-vector mapping functions. The solution optimality and tractability were verified both mathematically and by using the SITLSP. Regarding the objectives of guidance system design mentioned in Chapter 1, it was found that a smooth arrival of the vehicle into the DS cannot be accomplished based on the original 7-state solution. To this end, the 10-state solution was developed. While the new 10-state approach resulted in satisfying the final boundary conditions on the vehicle thrusters, however by meeting this requirement, other metrics related to the docking-enabling conditions and computational efficiency were degraded. Even using the hp-AR method and the powerful IPOPT NLP solver, a considerable amount of time was required for the solution convergence. This is because of the fact that a large number of collocation points are dictated by the 10-state solution and therefore the computational time increases significantly. Consequently, furnishing the other important requirements of this thesis such as real-time docking operations and ultimately the closed-loop docking implementation on-board the Flinders AUV become impractical.

Regarding the proposed difficulties associated with the PS methods, in the next chapter, the IDVD method's capabilities towards meeting all requirements of this thesis and consequently developing a systematic and universal docking guidance system are examined.

Chapter 5

Docking Guidance Using the *Inverse Dynamics in the Virtual Domain* Method

This chapter presents the concept, detailed theoretical explanation and computer simulations of the inverse dynamics in the virtual domain (IDVD) method for the docking problem. The overall goal is to employ the IDVD method to generate a feasible docking trajectory that satisfies endpoint requirements and vehicular/environmental constraints while enabling the vehicle to perform a relatively good maneuver with respect to assigned performance indices. The IDVD method uses the general idea of the direct methods of calculus of variations and inverse dynamics approach in generating a trajectory. In this chapter, along with considering the overall performance of the IDVD method through docking scenarios, the aim is to check three special properties. The first property is the ability to drive the AUV smoothly into the DS which ensures a safe and successful termination of the docking operation. The second property is the capability of the method to generate a real-time solution even if the solution is of a near-optimal type, i.e., loss of some optimality. Having this capability, an on-board implementation of the IDVD method on the host computer of the vehicle becomes possible. Resorting on the second property, the third property is the potential of the method to be implemented and operated in a closed-loop guidance configuration.

To this end, at the first stage, the capabilities of the IDVD method to direct the vehicle to the DS are investigated through two major offline scenarios. In the first scenario, the minimum-energy docking operation is considered while in the second scenario, the minimum-time docking maneuver is taken into account. Through both scenarios, the ability of the IDVD performance to meet the docking-enabling conditions is investigated by the SITLSP. In the second stage, the ability of the IDVD method to generate online trajectories in a closed-loop configuration through a representative minimum-energy docking maneuver is investigated. The results of this part are also validated by the SITLSP.

At the end of the chapter, the IDVD robustness tests are considered using three random conditions, namely random initial pose, random DS pose, and random cross-current disturbance.

The results of this chapter show the substantial features of the IDVD method such as solution optimality, computational efficiency, sensitivity to initial guesses, and robustness against uncertain conditions in both offline and online trajectory generation process.

5.1 Concept of IDVD Method

The idea of IDVD method is to use the differential flatness property of the system dynamics, to significantly reduce the dimension of optimization problems and thus enable fast prototyping of an optimal trajectory. This is achieved by expressing the states and controls as functions of a flat output vector of systems, which more often than not involves spatial coordinates of systems and their derivatives. Another distinguished feature of the IDVD is that it performs trajectory computations in the virtual domain as opposed to the time domain. This feature results in decoupling optimization of the 3D path and speed profile along it. Therefore, more flexibility and controllability are provided for the trajectory optimization purpose.

Several documented research confirm the effectiveness and reliability of the IDVD method for real-time trajectory optimizations [90, 111-113]. This effectiveness stems from several important properties, namely:

- the problem boundary conditions comprising higher-order derivative terms are satisfied a priori;
- the controls generated are smooth and physically realizable;
- the method is not subjected to the curse of dimensionality and does not need differentiability of the objective function and therefore any model and objective function can be utilized;
- a significant degree of robustness is provided as a result of low sensitivity to the initial guess and small variations in input variables;
- a fast convergence rate is provided as the optimization routine used only a few variable parameters.

Compared to other direct methods such as PS ones, the IDVD computational speed is more than an order of magnitude faster at the cost of a small loss of optimality. Thus, these salient features are the primary rationale behind employing the IDVD method for developing a reliable and efficient docking guidance system.

5.1.1 Problem Formulation Using IDVD Method

The essence of the IDVD method can be summarized in the following steps:

***Step* 1** Generate a reference function in the virtual domain (τ domain) that is independent of time derivative constraints.

***Step* 2** Convert the reference trajectory back into the time domain using the speed factor (λ).

***Step* 3** Employ inverse dynamics to calculate states and controls.

***Step* 4** Execute optimization routine considering boundary conditions, constraints, and performance index.

A. Generate Reference Functions

To convert the docking TPBVP (mentioned in chapter 3) into a low-dimensional NLP problem, a proper parametrization is required to perform. This parametrization is developed based on a set of reference (basis) functions. In general, the reference functions can be combination of any function such as orthogonal, monomial, or trigonometric in order to meet the requirements of the design. In other words, the general shape of an expected trajectory is a function of the reference function and its parametrization order. This order is determined via the number of boundary conditions that should be satisfied. By increasing the parametrization order, more degrees of freedom are provided. The minimum degree of the polynomial is determined according to

$$n_p = d_0 + d_f + 1 \qquad (5\text{-}1)$$

where d_0, and d_f are the highest-orders of time derivative of the reference function at the initial and terminal points, respectively.

For the proposed TPBVP, a 5th order polynomial reference function including trigonometric terms is defined for the three spatial coordinates $x, y,$ and z using an analytically-defined basis functions of an abstract scaled argument $\bar{\tau} = \tau/\tau_f \in [0; 1]$

$$\begin{aligned} x(\bar{\tau}) &= P_x(\bar{\tau}) = a_{0x} + a_{1x}\bar{\tau} + a_{2x}\bar{\tau}^2 + a_{3x}\bar{\tau}^3 + b_{1x}\sin(\pi\bar{\tau}) + b_{2x}\sin(2\pi\bar{\tau}) \\ y(\bar{\tau}) &= P_y(\bar{\tau}) = a_{0y} + a_{1y}\bar{\tau} + a_{2y}\bar{\tau}^2 + a_{3y}\bar{\tau}^3 + b_{1y}\sin(\pi\bar{\tau}) + b_{2y}\sin(2\pi\bar{\tau}) \\ z(\bar{\tau}) &= P_z(\bar{\tau}) = a_{0z} + a_{1z}\bar{\tau} + a_{2z}\bar{\tau}^2 + a_{3z}\bar{\tau}^3 + b_{1z}\sin(\pi\bar{\tau}) + b_{2z}\sin(2\pi\bar{\tau}) \end{aligned} \qquad (5\text{-}2)$$

In (5-2) the role of trigonometric terms is to increase the flexibility in varying the curvature of the resulting trajectory especially at the terminal points, and the coefficients $a_{i\eta}$ and $b_{i\eta}$ ($\eta = \{x, y, z\}$) are defined by the boundary conditions (BCs) up to the second-order derivative at τ = 0 and τ_f= 0 ($\bar{\tau} = 1$). According to (3-16) in Chapter 3, these BCs can be set as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\tau=0} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}, \quad \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}_{\tau=0} = \begin{bmatrix} u_0 \cos(\psi_0) + c_x \\ u_0 \sin(\psi_0) + c_y \\ w_0 \end{bmatrix} \tag{5-3}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\tau=\tau_f} = \begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix}, \quad \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}_{\tau=\tau_f} = \begin{bmatrix} u_f \cos(\psi_f) \\ u_f \sin(\psi_f) \\ w_f \end{bmatrix} \tag{5-4}$$

At the initial point, the second-order derivatives are described by

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix}_{\tau=0} = \begin{bmatrix} \dot{u}_0 \cos(\psi_0) - u_0 r_0 \sin(\psi_0) \\ \dot{u}_0 \sin(\psi_0) + u_0 r_0 \cos(\psi_0) \\ \dot{w}_0 \end{bmatrix} \tag{5-5}$$

where the derivatives of the velocity vector components are computed according to (3-17) and in the case of equilibrium these components are zero. At the final point the second-order derivatives should be set to zero for a smooth arrival into DS

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix}_{\tau=\tau_f} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{5-6}$$

Next, (5-2) is differentiated twice with respect to the argument $\bar{\tau}$ to obtain

$$\tau_f P'_{\tau;\eta}(\bar{\tau}) = a_{1\eta} + 2a_{2\eta}\bar{\tau} + 3a_{3\eta}\bar{\tau}^2 + \pi b_{1\eta}\cos(\pi\bar{\tau}) + 2\pi b_{2\eta}\cos(2\pi\bar{\tau}) \tag{5-7}$$

$$\tau_f^2 P''_{\tau\tau;\eta}(\bar{\tau}) = 2a_{2\eta} + 6a_{3\eta}\bar{\tau} - \pi^2 b_{1\eta}\sin(\pi\bar{\tau}) - (2\pi)^2 b_{2\eta}\sin(2\pi\bar{\tau}) \tag{5-8}$$

Equating these derivatives at the terminal points to the known BCs, (5-3) - (5-6) yields a system of linear algebraic equations to solve for coefficients $a_{i\eta}$ and $b_{i\eta}$ ($\eta = \{x, y, z\}$). For instance, for the *x*-coordinate, the set of linear algebraic equations becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & \pi & 2\pi \\ 0 & 1 & 2 & 3 & -\pi & 2\pi \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{0x} \\ a_{1x} \\ a_{2x} \\ a_{3x} \\ b_{1x} \\ b_{2x} \end{bmatrix} = \begin{bmatrix} x_0 \\ x_f \\ x'_0\tau_f \\ x'_f\tau_f \\ x''_0\tau_f^2 \\ x''_f\tau_f^2 \end{bmatrix} \tag{5-9}$$

Now, by resolving the system in (5-14), the unknown coefficients for the basis function can be obtained as

$$\begin{aligned} & a_{0x} = x_0, \quad a_{1x} = -(x_0 - x_f) + \frac{(2x''_0 + x''_f)\tau_f^2}{6} \\ & a_{2x} = \frac{x''_0\tau_f^2}{2}, \quad a_{3x} = -\frac{(x''_0 - x''_f)\tau_f^2}{6} \\ & b_{1x} = \frac{2(x'_0 - x'_f)\tau_f + (x''_0 + x''_f)\tau_f^2}{4\pi} \\ & b_{2x} = \frac{12(x_0 - x_f) + 6(x'_0 + x')\tau_f + (x''_0 - x''_f)\tau_f^2}{24\pi} \end{aligned} \tag{5-10}$$

As all of the coefficients in (5-10) are uniquely defined by the boundary conditions, τ_f is the only varied parameter used in the optimization process. To have more flexibility over the shape of trajectory, adding higher-order terms is essential (see sub-section 5.1.2).

B. Mapping from the Virtual to Time Domain

As mentioned, the reference trajectory is defined in the virtual domain not in the time domain. This feature results in decoupling optimization of the 3D path and the corresponding speed profile (or decoupling of space and time). Once the spatial coordinates are parametrized as a function of virtual domain parameter ($\bar{\tau}$), there must exist a mapping to transfer the candidate trajectories into the time domain. The mapping between the physical domain and the virtual domain is performed using the so-called speed factor $\lambda(\tau)$

$$\lambda(\tau) = \frac{d\tau}{dt} \tag{5-11}$$

Applying differentiation rule to any time-variant parameter ξ

$$\dot{\xi} = \lambda \xi', \quad \ddot{\xi} = \lambda(\lambda'_\tau \xi'_\tau + \lambda \xi''_{\tau\tau}) \tag{5-12}$$

and inverting the result yields

$$\xi'_\tau = \lambda^{-1}\dot{\xi}, \quad \xi''_{\tau\tau} = \lambda^{-2}\ddot{\xi} - (\lambda'_\tau \lambda^{-1} \xi') \tag{5-13}$$

Since the speed factor $\lambda(\tau)$ simply scales the entire problem (i.e., the higher the speed factor $\lambda(\tau)$, the larger τ_f), one may assume

$$\lambda_{0;f} = 1, \; \lambda'_{0:f} = 0 \tag{5-14}$$

$$\xi'_{\tau;0;f} = \dot{\xi}_{0;f}, \quad \xi''_{\tau\tau;0;f} = \ddot{\xi}_{0;f} \tag{5-15}$$

C. Inverting the Dynamics

Now let us describe the numerical procedure for finding the optimal solution among all candidate trajectories. First, a guesstimate of the value of the only decision parameter τ_f is provided; then the coefficients of the candidate trajectory using (5-10) with the BCs (5-3)-(5-6) are computed and converted to the virtual domain using (5-15). As an initial guess of the length of the virtual arc τ_f, a value proportional to the distance between the starting and terminal points, is considered

$$\tau_f = 1.5\sqrt{(x_f - x_0)^2 + (y_f - y_0)^2 + (z_f - z_0)^2} \tag{5-16}$$

Second, having an analytical representation of the candidate trajectory, (5-2), (5-7), and (5-8), the values of x_j, y_j, z_j, x'_j, y'_j, and $z_j{}'$, $(j = 1, \ldots, N)$ over a fixed set of N points (for instance, $N = 100$) spaced along the virtual arc $[0;\ \tau_f]$ are defined:

$$\Delta\tau = \tau_f (N-1)^{-1}, \quad (\Delta\bar{\tau} = (N-1)^{-1}) \tag{5-17}$$

so that

$$\tau_j = \tau_{j-1} + \Delta\tau \quad (\bar{\tau}_j = \bar{\tau}_{j-1} + \Delta\bar{\tau}); \quad j = 2,\ldots,N; \quad (\tau_1 = \bar{\tau}_1 = 0) \tag{5-18}$$

Third, for each node $j = 2, \ldots, N$, it is necessary to compute the time step as it is not constant. This time step is calculated based on the division of distance between two computational nodes along the arc over the speed

$$\begin{aligned}
\Delta t_{j-1} &= \frac{\sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2 + (z_j - z_{j-1})^2}}{\sqrt{u_{j-1}^2 + w_{j-1}^2 + c_x^2 + c_y^2 + 2c_x(u_{j-1}\cos(\psi_{j-1})) + 2c_y(u_{j-1}\sin(\psi_{j-1}))}} \\
\lambda_j &= \Delta\tau\Delta t_{j-1}^{-1}
\end{aligned} \tag{5-19}$$

Now, the rest of states and controls are computed by using inverse dynamics as follows

$$\begin{aligned}
u_j &= \sqrt{(\lambda_j x'_{\tau;j} - c_x)^2 + (\lambda_j y'_{\tau;j} - c_y)^2} \\
\psi_j &= \tan^{-1}\left(\frac{\lambda_j y'_{\tau;j} - c_y}{\lambda_j x'_{\tau;j} - c_x}\right) \\
w_j &= \lambda_j z'_{\tau;j} \\
r_j &= \frac{-(\lambda y' - c_y)(\lambda' x' + \lambda x'') + (\lambda x' - c_x)(\lambda' y' + \lambda y'')}{(\lambda y' - c_y)^2 + (\lambda x' - c_x)^2}
\end{aligned} \tag{5-20}$$

$$\begin{aligned}
T_{u;j} &= m\lambda_j u'_{\tau;j} - (X_u + X_{uu}|u_j|)u_j \\
T_{w;j} &= m\lambda_j w'_{\tau;j} - (Z_w + Z_{ww}|w_j|)w_j \\
T_{r;j} &= I_z\lambda_j r'_{\tau;j} - (N_r + N_{rr}|r_j|)r_j
\end{aligned} \tag{5-21}$$

Derivatives $u'_{\tau;j}$, $w'_{\tau;j}$ and $r'_{\tau;j}$ in (5-21) can be computed using finite differences or analytical expressions, by differentiating the equations in (5-20). For example, forward and vertical acceleration components of the vehicle in (5-21), can be computed using

$$\begin{aligned}
u'_{\tau;j} &= \frac{\left((\lambda'_j x'_{\tau;j} + \lambda_j x''_{\tau;j})(\lambda_j x'_{\tau;j} - c_x) + (\lambda'_j y'_{\tau;j} + \lambda_j y''_{\tau;j})(\lambda_j y'_{\tau;j} - c_y)\right)}{u_j} \\
w'_{\tau;j} &= \lambda'_j z'_{\tau;j} + \lambda_j z''_{\tau;j}
\end{aligned} \tag{5-22}$$

D. Optimization Routine

When all parameters (states and controls) are computed in each of the N points, the performance index, is computed. For example, for minimum-energy docking operations mathematically represented by the performance index (3-33), an applicable discrete form is

$$J = k_T \left(\sum_{j=1}^{N-1} \Delta t_j - t_f \right)^2 + k_u \sum_{j=1}^{N-1} (T_{u;j}^2 + T_{w;j}^2 + T_{r;j}^2) \Delta t_j \tag{5-23}$$

where t_f represents the final time and k_T , k_u are weighting factors.

The penalty function that accounts for violation of constraints on the state variable and controls is constructed as

$$\begin{aligned} \Delta = {} & k_\psi \left(\psi_N - \psi_f \right)^2 + k_r \max_j \left(0; \left| r_j \right| - r^{\max} \right)^2 + k_{T_u} \max_j \left(0; \left| T_{u;j} \right| - T_u^{\max} \right)^2 \\ & + k_{T_w} \max_j \left(0; \left| T_{w;j} \right| - T_w^{\max} \right)^2 + k_{T_r} \max_j \left(0; \left| T_{r;j} \right| - T_r^{\max} \right)^2 \end{aligned} \tag{5-24}$$

where $k_i$are the weighting coefficients allowing to balance all individual terms. Now, regarding the level of solution optimality, convergence rate and computational efficiency, the proposed low-dimensional NLP problem can be solved using numerical solvers such as the built-in *fmincon* function in the MATLAB development environment, the SNOPT, or the IPOPT.

5.1.2 Adding More Flexibility by Utilizing Multiple Parameter Variables

In sub-section 5.1.1, the virtual arc (τ_f) is the only parameter varied through the optimization routine. However, just relying on one decision parameter results in confining the flexibility of the reference functions within the set of admissible trajectories. In order to achieve greater flexibility in the reference functions, it is necessary to add more parameter variables through the optimization routine. To this end, additional fictitious boundary conditions called jerk boundary conditions are provided using higher-order derivatives, here 3rd order. As in (5-3) - (5-6) the initial and final positions, velocities, and accelerations are specified, the jerk becomes the varied parameter along with τ_f. Thus, a 7th order polynomial is employed for the reference functions (following (5-1)). For example, increasing the order of the basis functions in (5-2) leads to

$$\begin{aligned} x(\bar{\tau}) &= P_x(\bar{\tau}) = a_{0x} + a_{1x}\bar{\tau} + a_{2x}\bar{\tau}^2 + a_{3x}\bar{\tau}^3 + a_{4x}\bar{\tau}^4 + a_{5x}\bar{\tau}^5 + b_{1x}\sin(\pi\bar{\tau}) + b_{2x}\sin(2\pi\bar{\tau}) \\ y(\bar{\tau}) &= P_y(\bar{\tau}) = a_{0y} + a_{1y}\bar{\tau} + a_{2y}\bar{\tau}^2 + a_{3y}\bar{\tau}^3 + a_{4y}\bar{\tau}^4 + a_{5y}\bar{\tau}^5 + b_{1y}\sin(\pi\bar{\tau}) + b_{2y}\sin(2\pi\bar{\tau}) \\ z(\bar{\tau}) &= P_z(\bar{\tau}) = a_{0z} + a_{1z}\bar{\tau} + a_{2z}\bar{\tau}^2 + a_{3z}\bar{\tau}^3 + a_{4z}\bar{\tau}^4 + a_{5z}\bar{\tau}^5 + b_{1z}\sin(\pi\bar{\tau}) + b_{2z}\sin(2\pi\bar{\tau}) \end{aligned} \tag{5-25}$$

Differentiating (5-25) three times with respect to the $\bar{\tau}$ argument yields

$$\tau_f P'_{\tau;\eta}(\bar{\tau}) = a_{1\eta} + 2a_{2\eta}\bar{\tau} + 3a_{3\eta}\bar{\tau}^2 + 4a_{4\eta}\bar{\tau}^3 + 5a_{5\eta}\bar{\tau}^4 + \pi b_{1\eta}\cos(\pi\bar{\tau}) + 2\pi b_{2\eta}\cos(2\pi\bar{\tau}) \quad (5\text{-}26)$$

$$\tau_f^2 P''_{\tau\tau;\eta}(\bar{\tau}) = 2a_{2\eta} + 6a_{3\eta}\bar{\tau} + 12a_{4\eta}\bar{\tau}^2 + 20a_{5\eta}\bar{\tau}^3 - \pi^2 b_{1\eta}\sin(\pi\bar{\tau}) - (2\pi)^2 b_{2\eta}\sin(2\pi\bar{\tau}) \quad (5\text{-}27)$$

$$\tau_f^3 P'''_{\tau\tau;\eta}(\bar{\tau}) = 6a_{3\eta} + 24a_{4\eta}\bar{\tau} + 60a_{5\eta}\bar{\tau}^2 - \pi^3 b_{1\eta}\cos(\pi\bar{\tau}) - (2\pi)^3 b_{2\eta}\cos(2\pi\bar{\tau}) \quad (5\text{-}28)$$

and then, rewriting (5-9) for the *x*-coordinate, results in

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \pi & 2\pi \\ 0 & 1 & 2 & 3 & 4 & 5 & -\pi & 2\pi \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6 & 12 & 20 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & -\pi^3 & -8\pi^2 \\ 0 & 0 & 0 & 6 & 24 & 60 & \pi^3 & -8\pi^2 \end{bmatrix} \begin{bmatrix} a_{0x} \\ a_{1x} \\ a_{2x} \\ a_{3x} \\ a_{4x} \\ a_{5x} \\ b_{1x} \\ b_{2x} \end{bmatrix} = \begin{bmatrix} x_0 \\ x_f \\ x'_0\tau_f \\ x'_f\tau_f \\ x''_0\tau_f^2 \\ x''_f\tau_f^2 \\ x'''_0\tau_f^3 \\ x'''_f\tau_f^3 \end{bmatrix} \quad (5\text{-}29)$$

Accordingly, by resolving (5-29), unknown coefficients for the *x* basis function are achieved as follows:

$$a_{0x} = x_0$$

$$a_{1x} = \frac{\begin{pmatrix} -(24(3x'''_0 - 2x'''_f) - \pi^2(5x'''_0 + 3x'''_f))\tau_f^3 - \\ (72(7x''_0 + 3x''_f) - 12\pi^2(3x''_0 + x''_f))\tau_f^2 - \\ (12\pi^2(9x'_0 - x'_f) - 8\pi^4 x'_0)\tau_f - \\ 120(12(x_0 - x_f) - \pi^2(x_0 - x_f) \end{pmatrix}}{8(15 - \pi^2)(12 - \pi^2)}$$

$$a_{2x} = \frac{x''_0\tau_f^2}{2}$$

$$a_{3x} = \frac{\begin{pmatrix} (120x'''_0 - \pi^2(9x'''_0 + x'''_f))\tau_f^3 + \\ (3\pi^2(26x''_0 - 5x''_f) - 2\pi^4(3x''_0 - x''_f))\tau_f^2 \\ +4(15\pi^2(5x'_0 + 3x'_f) - 2\pi^4(3x'_0 + 2x'_f))\tau_f \\ +40(12\pi^2(x_0 - x_f) - \pi^4(x_0 - x_f)) \end{pmatrix}}{4(15 - \pi^2)(12 - \pi^2)}$$

$$
\begin{aligned}
a_{4x} &= \frac{\begin{pmatrix} (120(2x_0''' + x_f''') - \pi^2(19x_0''' + 11x_f'''))\tau_f^3 - \\ (360(x_0'' - x_f'') + 60\pi^2(2x_0'' - x_f'') - 4\pi^4(3x_0'' - 2x_f''))\tau_f^2 + \\ (-60\pi^2(13x_0' + 10x_f') - 8\pi^4(8x_0' + 7x_f'))\tau_f \\ -120\pi^2(12 - \pi^2)(x_0 - x_f) \end{pmatrix}}{8(15 - \pi^2)(12 - \pi^2)} \\
a_{5x} &= \frac{\begin{pmatrix} 3(x_0''' + x_f''')\tau_f^3 + 2(3 + \pi^2)(x_0'' - x_f'')\tau_f^2 + \\ 12\pi^2(x_0' + x_f')\tau_f + 24\pi^2(x_0 - x_f) \end{pmatrix}}{4(15 - \pi^2)} \\
b_{1x} &= \frac{(x_0''' - x_f''')\tau_f^3 + 6(x_0'' + x_f'')\tau_f^2 + 12(x_0' - x_f')\tau_f}{2\pi(12 - \pi^2)} \\
b_{2x} &= \frac{(x_0''' - x_f''')\tau_f^3 + 12(x_0'' - x_f'')\tau_f^2 + 60(x_0' + x_f')\tau_f + 120(x_0 - x_f)}{16\pi(15 - \pi^2)}
\end{aligned}
\qquad (5\text{-}30)
$$

with the jerks x_0''' and x_f'''being two additional parameters.

5.2 Minimum-Energy Docking Performance

The first scenario represents a situation where the AUV intends to perform docking with minimum control effort (minimum energy expenditure) and thus the performance index is of the form of (3-33). In this case, however, the normalized performance index introduced in Chapter 4 is employed

$$
J = \frac{1}{t_f (T_u^{\max})^2} \int_{t_0}^{t_f} (T_u^2 + T_w^2 + T_r^2) dt \qquad (5\text{-}31)
$$

and for the sake of clarity, the scenario conditions previously given in Chapter 4 are presented again here.

The minimum-energy docking maneuvers are performed in an environment where the DS pose is known a priori and the operating environment is free of any NFZs or obstacles. In this environment, a 2D current disturbance of magnitude c_x=0.25 m/s, c_y=0.25 m/s with respect to the North and East respectively in the $\{n\}$ frame, is considered. The AUV's initial pose is defined by a vector of $\eta_0 = [x_0, y_0, z_0, \psi_0]^T$ in which the initial position of the vehicle in the $\{n\}$ frame comprises x_0=50 m, y_0=50 m, z_0=5 m and the initial heading angle with respect to the true North is defined as ψ_0=10°. The rest of initial conditions are defined as follows: u_0=0.3m/s, w_0=0 m/s, and r_0=0 °/s. The DS pose is introduced by $\eta_d = [x_d, y_d, z_d, \psi_d]^T$ in which the position of DS in the $\{n\}$ frame is defined by x_d=150 m, y_d =75 m, z_d=10 m and the orientation of the DS with respect to the true North is defined by ψ_d=225°. Therefore, the spatial coordinates of the vehicle at the terminal docking phase defined by the equilibrium state vector $\eta_f = [x_f, y_f, z_f, \psi_f]^T$must meet the DS position; besides, the vehicle needs to show a final heading of 45°in the$\{n\}$ frame to be aligned with the centreline of DS for the final docking approach. The rest of final conditions are: u=0.4m/s, w_f=0 m/s, and r_f=0 °/s.

The IDVD method for all scenarios in this chapter uses 7 decision parameters that are$\tau_f, x_0''', y_0''', z_0''', x_f''', y_f''', z_f'''$. The initial guesses associated with these parameters are obtained from knowledge of the problem space and the IDVD structure. Moreover, the feasible solutions are also used to find the appropriate initial guesses. Compared to other direct methods (like PS methods), the IDVD method shows less sensitivity to the initial guesses and even arbitrary initial guesses offer the possibility of obtaining feasible solutions. In this scenario, 50 computational nodes (*N*=50) is used for the IDVD initialization and the *fmincon* solver tolerance accuracy is set to 10^{-6}.

Figures 5.1-5.6 demonstrate the results of simulation for the proposed scenario. In Figure 5.1, a 3D path generated in the $\{n\}$ frame is presented to drive the AUV from the starting point to the final DS position. As can be seen from Figure 5.1, the equilibrium spatial coordinates of the AUV are matched with position coordinates of the DS (indicated by a black triangle) at the terminal stage. This figure, moreover, shows that there exists a proper degree of flexibility for the vehicle path toward the DS as a result of the structure of the reference functions used in the IDVD problem formulation.

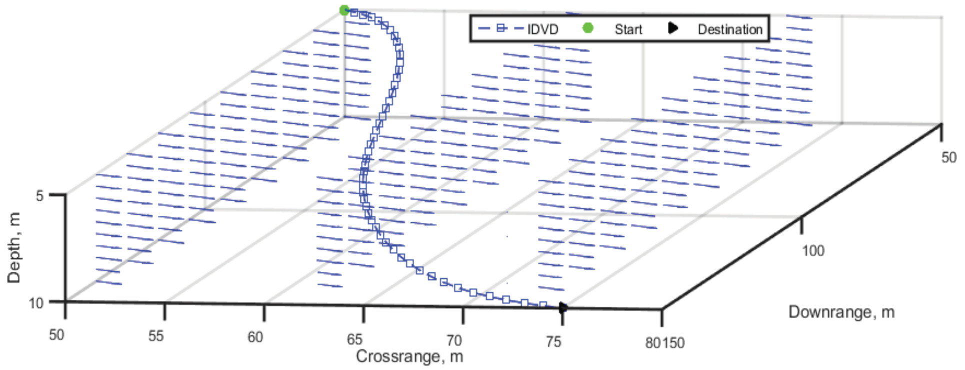


Figure 5.1 Representation of the 3D path for traveling from initial to DS position.

In Figure 5.2, the trajectories associated with the vehicle's heading and heading rate are shown and in Figure 5.3 the time history of the vehicle's forward and vertical velocities (in the {b} frame) are demonstrated. These figures indicate the satisfaction of the final boundary conditions particularly the alignment of the vehicle at the terminal stage with the DS orientation (shown by the yaw trajectory).

The controls generated by the IDVD method are depicted in Figure 5.4. The controls are smooth, meet the constraints posed by the vehicle actuators and terminal conditions, and hence realizable for practice. Due to the nature of the IDVD method, the controls always show a smooth time history instead of a bang–bang type, as they are computed as functions of the trajectory components and their derivatives through the inverse dynamic process (see (5-21)). Obviously, the controls generated with this approach are always continuous as the trajectory components are represented at least with class C_2 functions (see (5-1)). In this regard, however, the IDVD controls result in near-optimal solutions comparing to that of PS methods in Chapter 4.

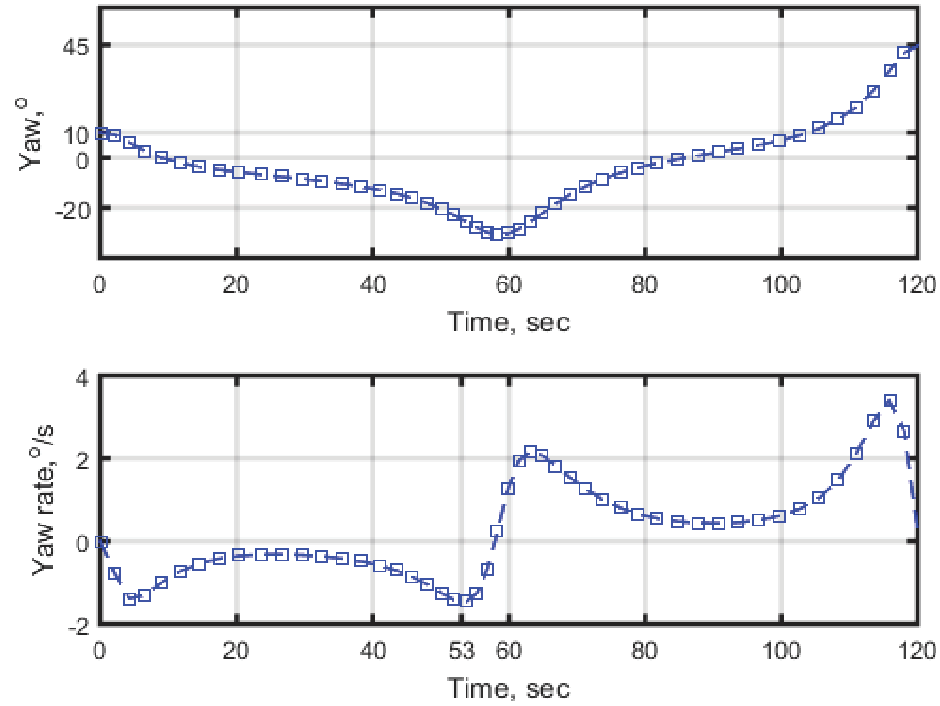


Figure 5.2 Time history of the vehicle’s heading and heading rate.

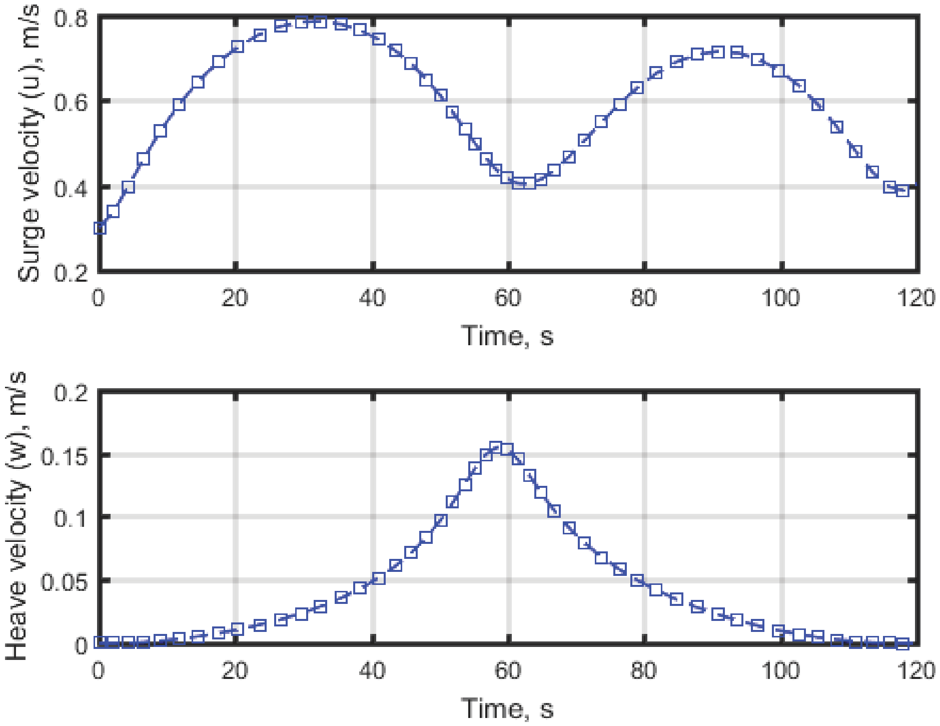


Figure 5.3 Time history of the vehicle’s surge and heave velocities.

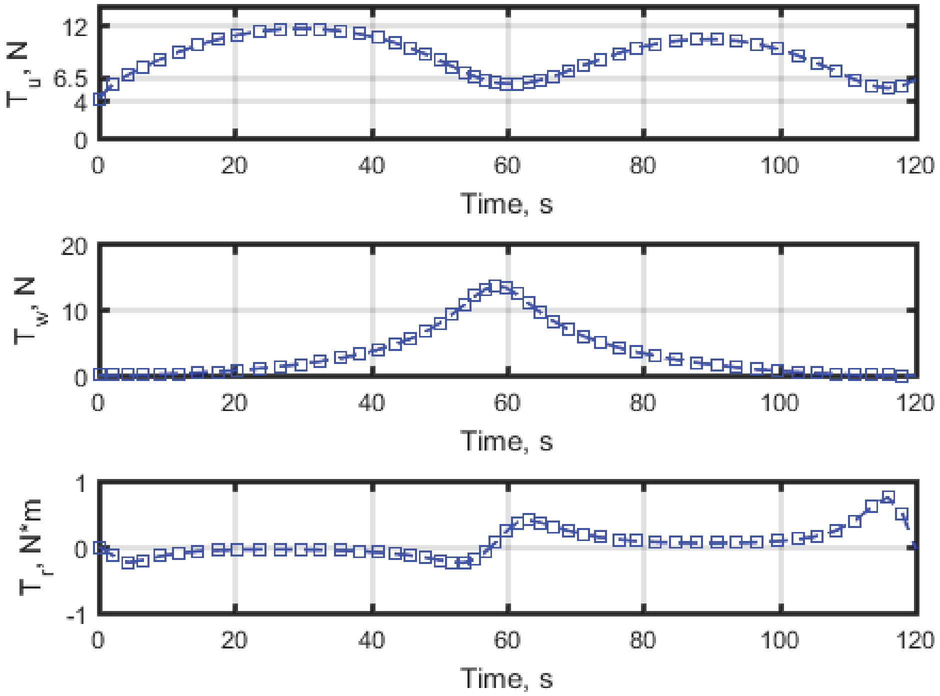


Figure 5.4 Evolution of controls during docking mission.

As can be inferred from the simulation results, in the trajectories generated by the IDVD method, the vehicle transitions at the arrival and departure time are smooth. This is due to the existence of higher order derivatives at the IDVD structure that can pictorially be observed from Figure 5.5 together with the changes of speed factor and non-linear mapping between the virtual and physical domains in Figure 5.6. The speed factor shows how fast the vehicle is moving along the spatial trajectory.

Tables 5.1 shows the changes of the initial guesses over the decision parameters after optimization process and Table 5.2 expresses the performance of the IDVD method for the proposed scenario in a quantitative manner. As can be seen from the table, a fast convergence to a near-optimal solution is achievable after only 6.7 second due to the great computational efficiency that the IDVD method offers. The IDVD method saves 48.8% control input expenditure compared to the case when the maneuver is performed at the control bounds. By providing a quantitative comparison between the cost function of the IDVD method and for example the LPS method, one can state that the IDVD controls result in a relative maximum difference of 23.8% in terms of optimal cost with the LPS solution. That is because of the fact that the IDVD method uses more control force of the main propeller and vertical thrusters than LPS method to allow smoother departure and arrival. A better solution quality is obtainable by increasing the order of the IDVD polynomials for trajectory representation, but at the cost of higher computational time. Therefore, there is always a trade-off between the solution optimality and computational efficiency that can be determined based on mission objectives.

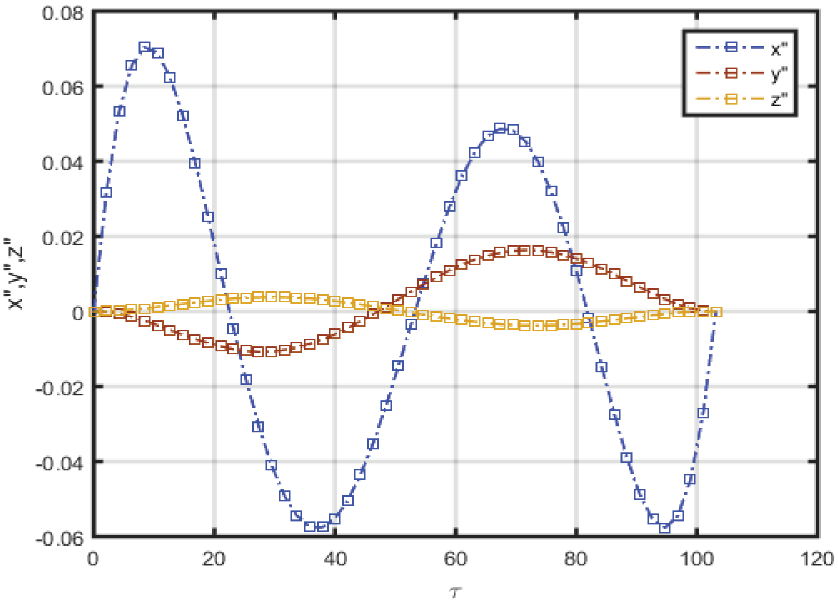


Figure 5.5 Higher order derivatives of spatial coordinates.

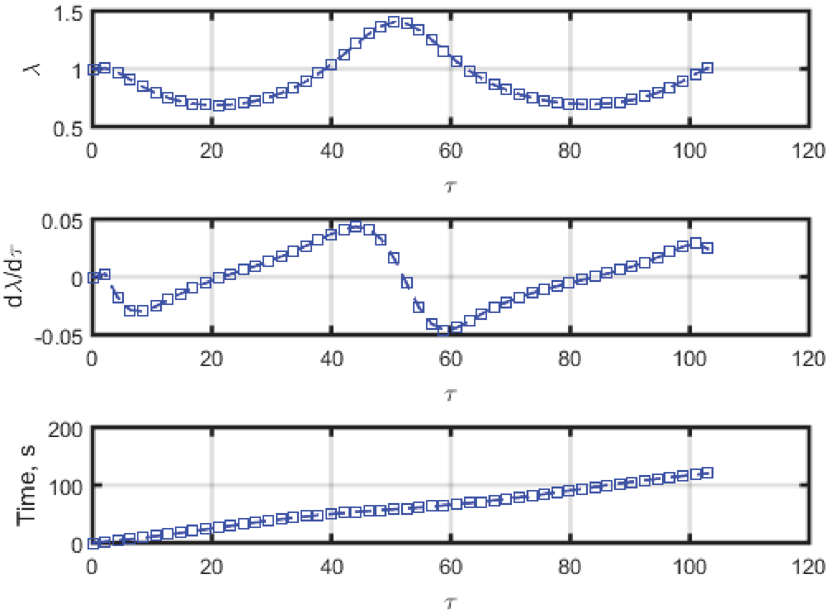


Figure 5.6 Speed factor and mapping between the virtual and physical domains.

Table 5.1 Estimated and optimized guess values of the decision parameters.

Varied Parameters	Initial guesses	Optimized guesses
Arc length	103.2	103
Initial jerk magnitude	0.010	0.017
Initial jerk horizontal orientation	-0.100	-0.058
Initial jerk vertical orientation	0.100	0.050
Final jerk magnitude	0.010	0.016
Final jerk horizontal orientation	-0.100	-0.119
Final jerk vertical orientation	0.100	0.240

Table 5.2 Quantitative expression of the IDVD performance.

Feature/Metric	*NoN*	*NoP*	*J*	t_{CPU}, *s*	*ES%*
Value	50	7	0.262	6.7	48.8

5.2.1 Performance Evaluation Using the SITLSP

This section provides a detailed analysis of the IDVD performance as applied within the realistic AUV SITLSP. By using the SITLSP, tractability, smoothness and the errors of the final conditions associated with the IDVD performance are investigated. Figures 5.7-5.10 illustrate the SITLP results corresponding to the minimum-energy solution generated by the IDVD method. In these figures, the results corresponding to the SITLSP are shown by black dashed-lines whereas the blue dashed-lines represent the IDVD simulation results mentioned in section 5.2.

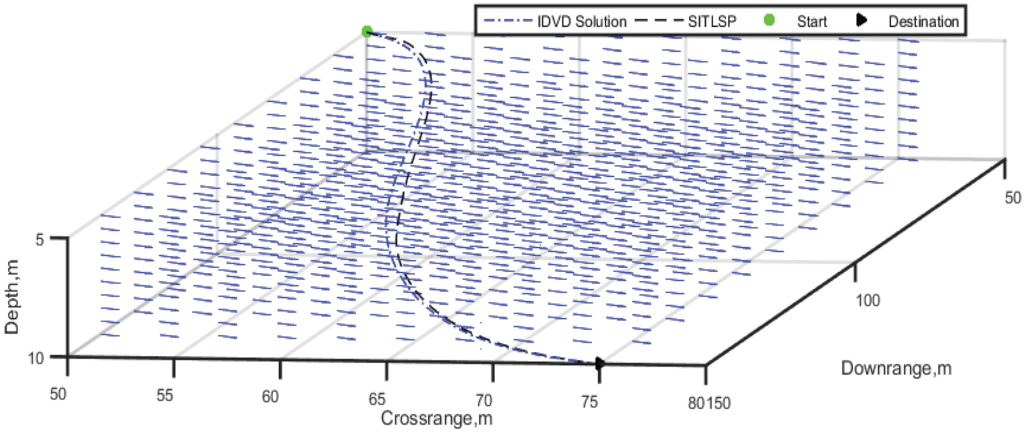


Figure 5.7 3D path evaluation within the SITLSP.

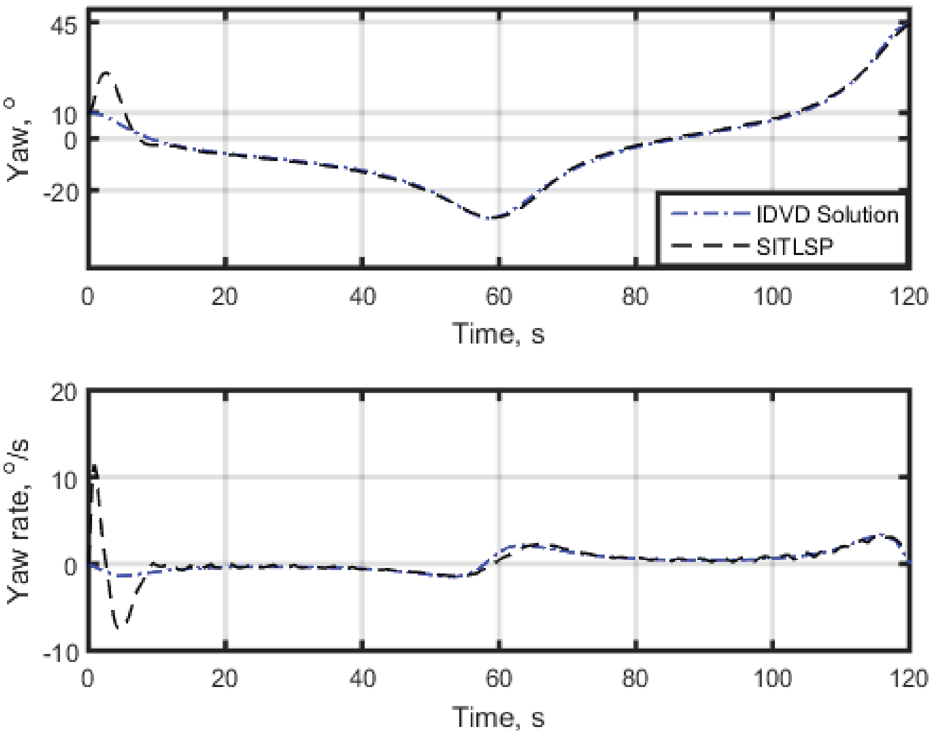


Figure 5.8 Evolution of heading and heading rate in the SITLSP.

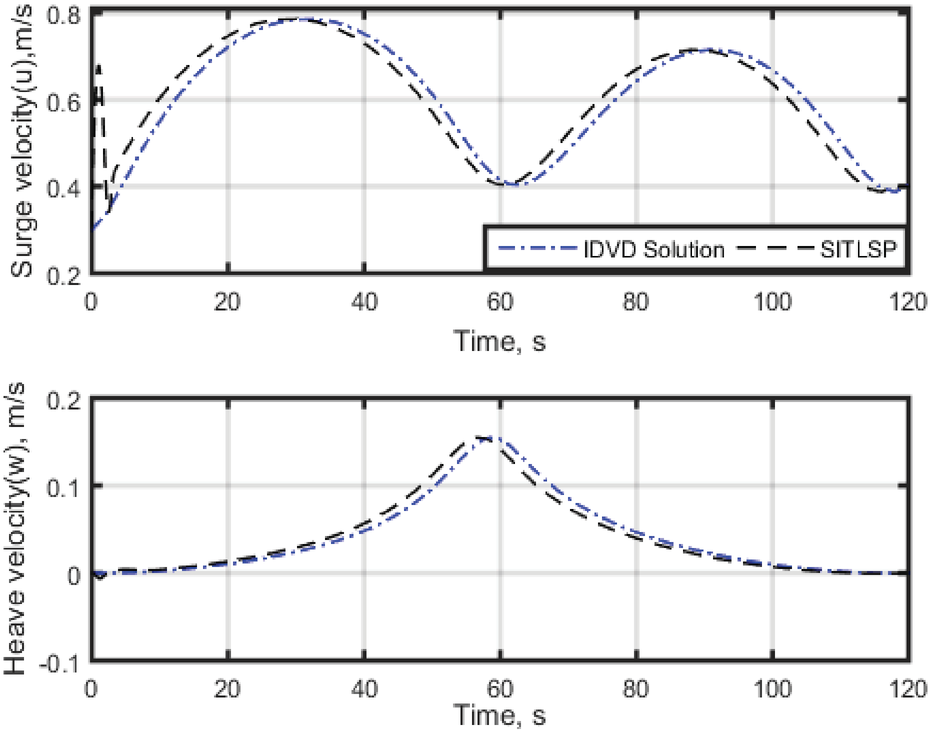


Figure 5.9 Time history of surge and heave velocities in the SITLSP.

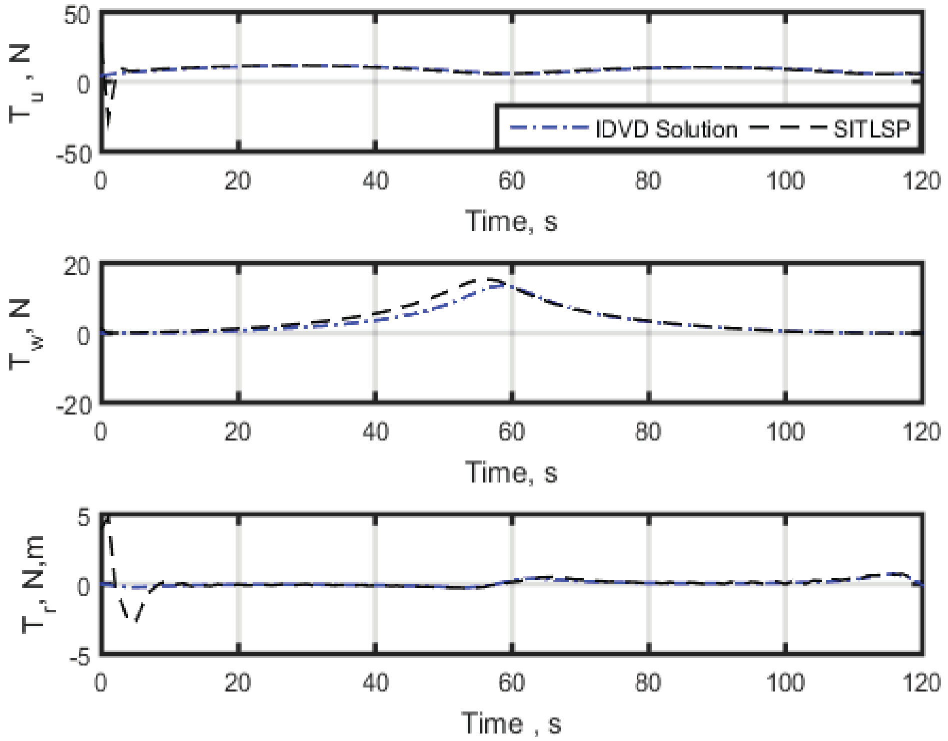


Figure 5.10 Evolution of controls in the SITLSP during docking mission.

As seen from Figure 5.7, the AUV follows the reference 3D path with a slight error of tracking in midcourse part but then at the end converges to the DS position. By referring to the yaw and yaw rate trajectories in Figure 5.8, one can assure the alignment of the vehicle heading with the DS orientation within the tolerance bound at the terminal phase. By examining the velocity components in Figure 5.9, it can be inferred that they are realizable for the real vehicle because of the correspondence between the simulation results and the SITLSP results. Figure 5.10, consequently shows the evolution of controls for this scenario. As can be seen, there exists a good match between the controls generated by the IDVD method and those generated by the SITLSP. This indicates the realization of the controls and consequently suitability of the IDVD guidance system for real implementations.

In some trajectories generated by the SITLSP, there exist spicks at the beginning or phase shift at the mid parts even though the trajectories converge to the equilibrium conditions at the end. This stems from the Traveling WP block in the SITLSP that inherently produces the reference states for the SMC controller based on 1-meter ahead distance. This behavior leads to some disjoints (jumps) at the beginning of the trajectories or phase shift in the mid parts. Consequently, the controller tries to minimize the error by accelerating the vehicle beyond what anticipated by trajectory generator (here the IDVD method). This acceleration produces transient differences between the desired velocities and those produced by the controller. It is noteworthy to mention that the spikes and tracking error can be improved by using an advanced waypoint generation block in the SITLSP; however, this design is not within the scope of this study.

Table 5.3 quantitatively shows that the IDVD method is successful in directing the vehicle form the initial conditions to the docking-enabling conditions. As can be seen, the tolerances defined for locating the vehicle within the cone of the DS and also aligning the vehicle with DS direction, are perfectly satisfied. Moreover, the required translational forward and vertical velocities together with rotational velocity supporting the final heading angle (yaw rate) are also within negligible bounds of errors.

Table 5.3 Quantitative evaluation of the IDVD performance for docking-enabling conditions using the SITLSP.

Metric	$\Delta\eta,_{m}$	$\Delta\psi,^{o}$	$\Delta u,_{m/s}$	$\Delta w,_{m/s}$	$\Delta r,^{o/s}$
Value	0.13	0.813	0.004	9.58e-06	1.4725

5.2.2 Adding Path Constraints

To make the docking performance more challenging, the path constraint of the form (4-56) is added to the proposed minimum-energy docking scenario. This path constraint represents three NFZs added to the operating environment. The NFZs are modeled in the form of spheres with the same coordinate and radius explained in Chapter 4. Here, the initial guesses over the decision parameters are similar to the previous scenario as tabulated in Table 5.4.

Figure 5.11 shows 3D path generated by the IDVD method. It is obvious from Figure 5.11 and Figure 5.12 that the path is collision-free and the vehicle can safely and smoothly transit from its initial position to the DS position. Here, in the presence of the NFZs, the IDVD method is still able to align the vehicle with the centerline of the DS (meeting 45^{o} heading angle at the endpoint) accompanied with a smooth arrival within the DS cone, indicated by final quantity of the heading rate in Figure 5.12. Moreover, from Figure 5.13, one can realize how the surge and heave velocities support the collision-free maneuver of the vehicle toward the DS; these velocities at the final boundary conditions, meet the demanded equilibrium quantities. The control solution is shown in Figure 5.14, which is smooth over the interval and always meets the bounds of the vehicle's actuators. In addition, Figure 5.14 indicates that the vehicle's actuators are not required to be saturated at any instant unlike the bang-bang nature of minimum-energy cost performance index; thus, a considerable saving in energy expenditure is achieved as indicated in Table 5.5. It is noteworthy to mention that even with added path constraint fast prototyping of the docking trajectory is obtained, due to the inherent computational efficiency of the IDVD method as indicated in Table 5.5. The rationale behind a lower computational time in this scenario comparing to the previous one (without the path constraint) is that the problem space is confined and therefore the IDVD method searches for a feasible solution in a smaller space.

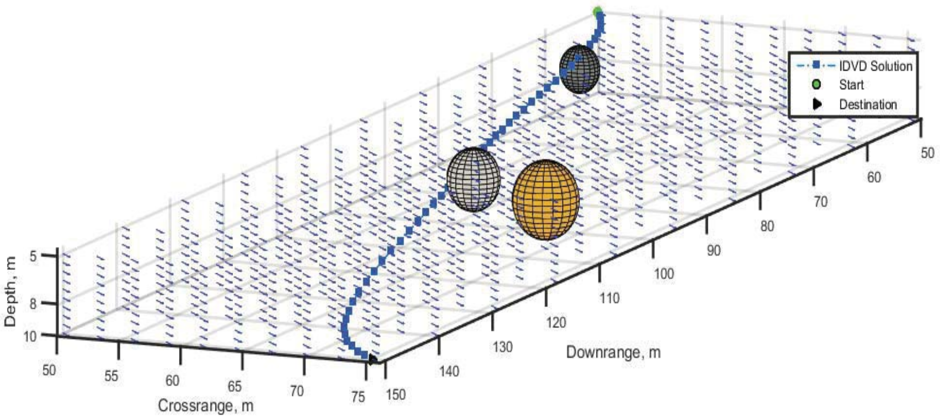


Figure 5.11 Collision-free 3D path.

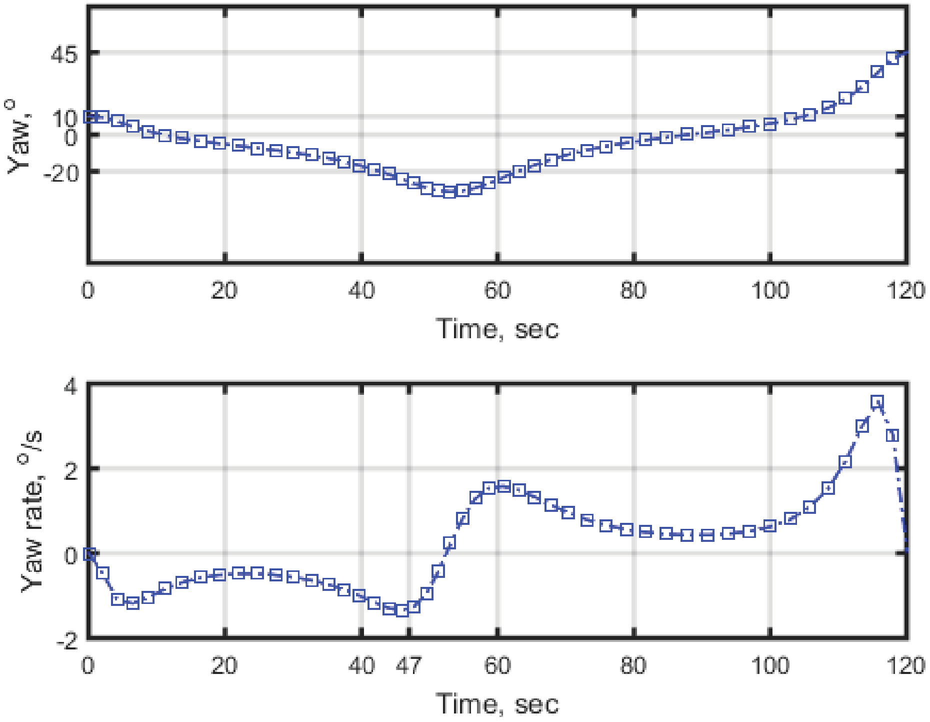


Figure 5.12 Vehicle’s heading and heading rate considering NFZs.

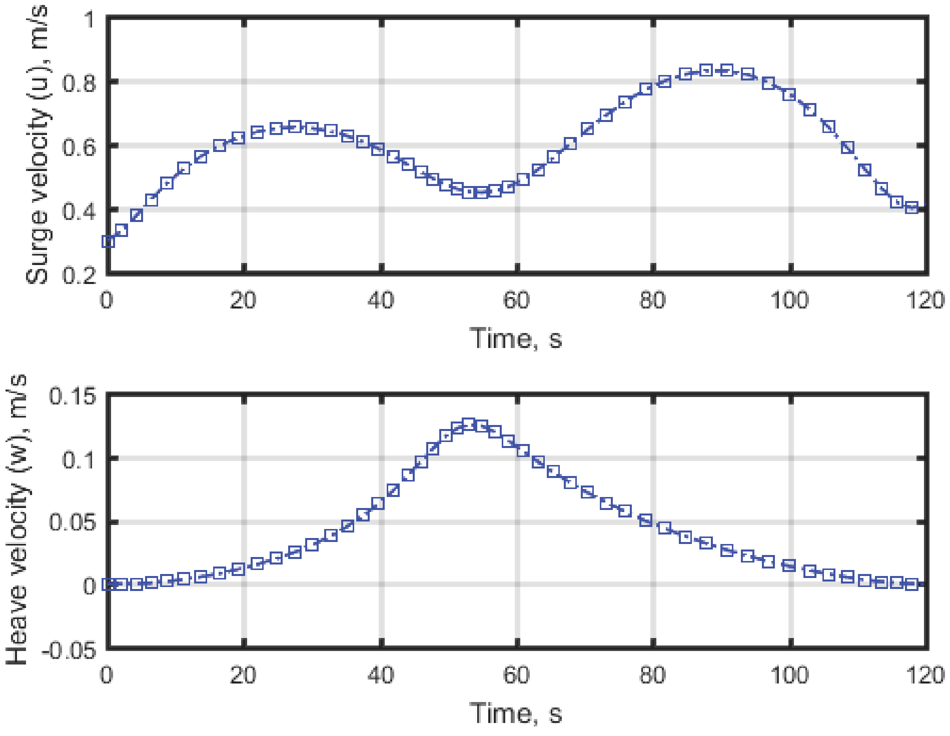


Figure 5.13 Vehicle’s surge and heave velocities considering NFZs.

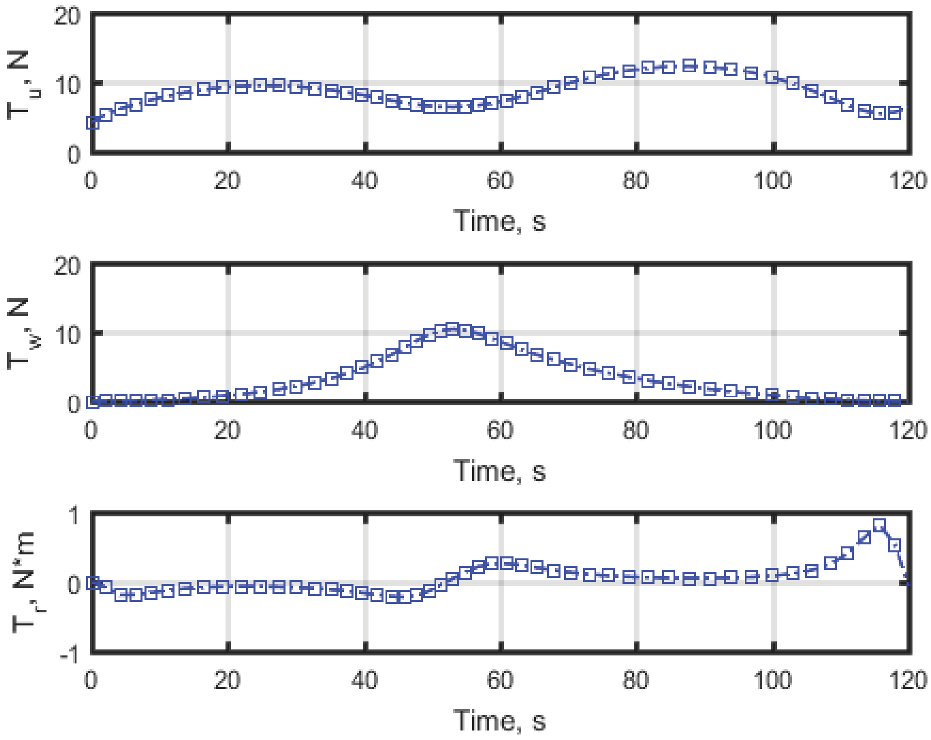


Figure 5.14 Evolution of controls considering NFZs.

Table 5.4 Estimated and optimized guess values on the decision parameters considering NFZs.

Varied Parameters	Initial guesses	Optimized guesses
Arc length	103.2	103.3
Initial jerk magnitude	0.010	0.012
Initial jerk horizontal orientation	-0.100	-0.078
Initial jerk vertical orientation	0.100	0.135
Final jerk magnitude	0.010	0.017
Final jerk horizontal orientation	-0.100	-0.052
Final jerk vertical orientation	0.100	0.072

Table 5.5 Quantitative expression of the IDVD performance considering NFZs.

Feature/Metric	***NoN***	***NoP***	***J***	**t_{CPU},s**	***ES%***
Value	50	7	0.257	4.5	49.3

5.2.3 Validation of Solution Using the SITLSP

To check the fidelity and tractability of the solution in presence of the path constraint, the desired trajectories generated by the IDVD method are applied to the SITLSP. Figures 5.15-5.18 illustrate the results of this test. As can be seen from Figure 5.15, there exists a strong correspondence between the IDVD 3D path and the one generated by the SITLSP. It shows that the 3D path is tractable for the STILSP and by using that vehicle can avoid NFZs and reach safely to the DS position. This is also supported by high tractability of the vehicle's heading in steering maneuver and aligning the vehicle with the DS orientation at the terminal phase as indicated in Figure 5.16. Again, the spikes at the beginning of these trajectories and the phase shift in the vehicle's velocity profiles in Figure 5.17 stem from the nature of the Traveling WP block in the SITLSP in generating the reference states for the SMC controller. Figure 5.18 demonstrates a strong matching between the IDVD controls and those generated by the controller block for activation of the vehicles' main propeller, lateral and vertical thrusters. This match is an evidence for proving the realization of the solution generated by the IDVD method. Meanwhile, the non-saturated and smooth nature of the controls are desirable for providing an energy efficient docking maneuver and having a smooth arrival to the docking enabling-conditions. These statements are quantitatively indicated in Table 5.6 in which all metrics of successful docking are perfectly satisfied within the defined tolerances.

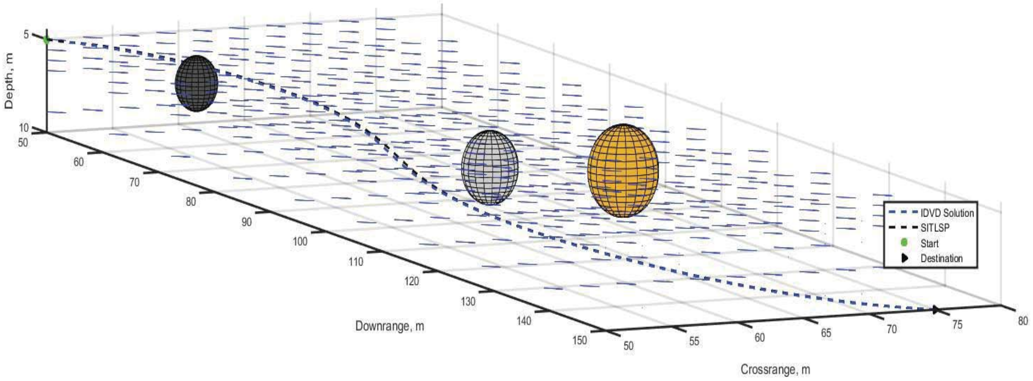


Figure 5.15 Verification of the 3D collision-free path using SITLSP.

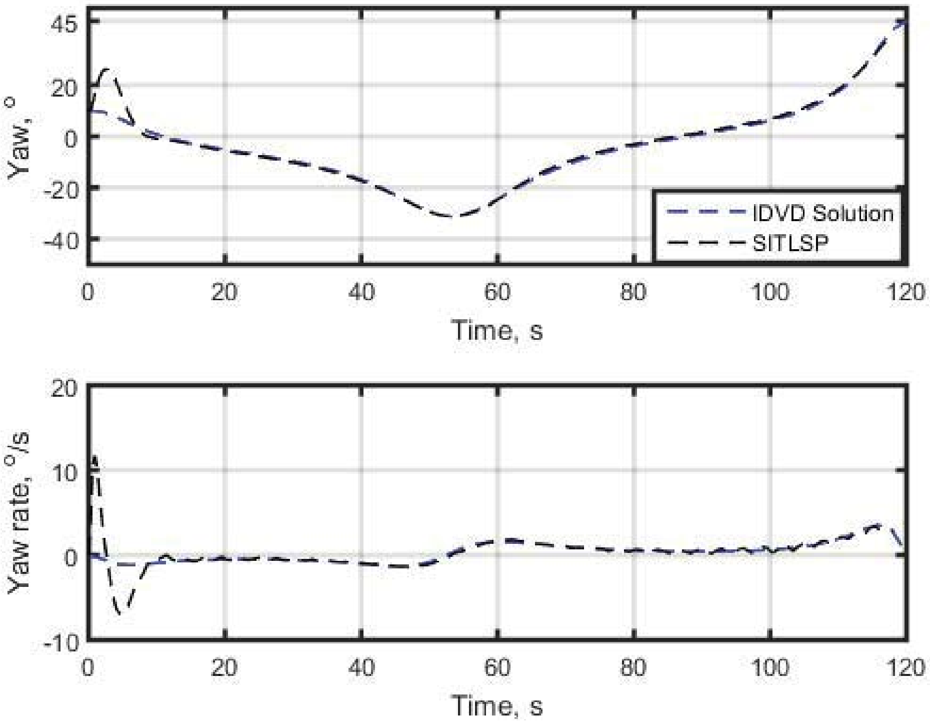


Figure 5.16 Heading and heading rate verification using SITLSP.

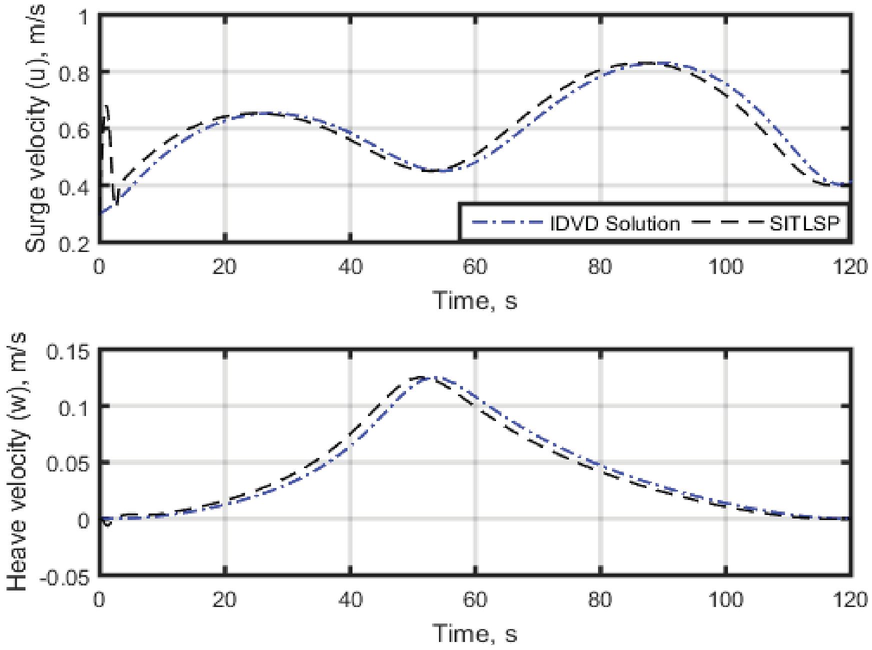


Figure 5.17 Surge and heave velocities verification using SITLSP.

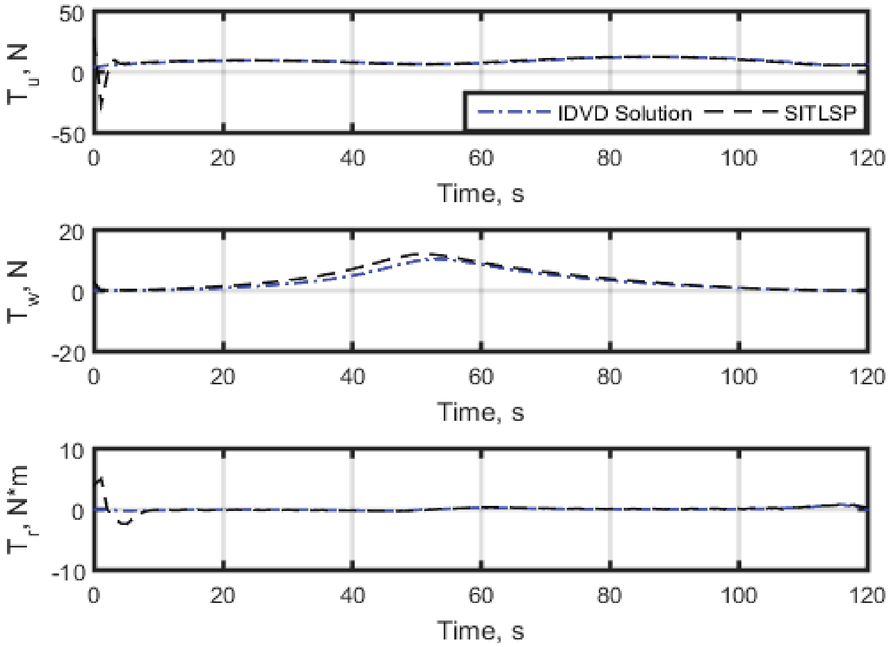


Figure 5.18 Evolution of controls in SITLSP.

Table 5.6 Quantitative evaluation of the IDVD performance for docking-enabling conditions considering path constraints.

Metric	$\Delta\eta_{,m}$	$\Delta\psi,^{o}$	$\Delta u_{,m/s}$	$\Delta w,_{m/s}$	$\Delta r,^{o/s}$
Value	0.18	0.892	0.014	1.7e-05	1.53

5.3 Minimum-Time Docking Performance

In this section, minimum-time trajectory generation for the docking problem is considered. Mathematically speaking, a general form of the performance index in this scenario is of the form (3-36) or an equivalent form of the endpoint cost in the Mayer form. In this scenario, a normalized performance index of the following form is utilized

$$J = \frac{1}{t_f (T_u^{\max})^2} \int_{t_0}^{t_f} \, dt \tag{5-32}$$

Conceptually, the goal of the IDVD method is to find the shortest travel time into the DS while considering the physical limitations of the vehicle's thrusters and possibility of the path constraint (NFZs). It is important for the IDVD method also to meet the docking-enabling conditions and provides a smooth arrival into the DS at the terminal phase. Similar to the Minimum-Energy docking scenario, here, the computational efficiency of the IDVD in generating a near-optimal solution is of great importance.

5.3.1 Simulation Results

In this scenario, the initial / final boundary conditions, constraints on the state and controls and path constraints (NFZs) are similar to the Minimum-Energy scenario except the final condition on the surge velocity that is free here. The rationale behind this is to allow the vehicle to use as much forward velocity as required to find a minimum-time solution. Meanwhile, to provide more flexibility for the vehicle maneuver, a number of 70 computational nodes are utilized. The initial guesses for the decision parameters of the IDVD method are kept unchanged as shown in Table 5.7.

Table 5.7 Estimated and optimized guess values on the decision parameters for the minimum-time docking performance.

Varied Parameters	Initial guesses	Optimized guesses
Arc length	103.2	103.2
Initial jerk magnitude	0.010	0.014
Initial jerk horizontal orientation	-0.100	-0.049
Initial jerk vertical orientation	0.100	0.110
Final jerk magnitude	0.010	0.010
Final jerk horizontal orientation	-0.100	-0.101
Final jerk vertical orientation	0.100	0.107

Table 5.8 Quantitative expression of the IDVD performance.

Feature/Metric	***NoN***	***NoP***	***J***	***t_{CPU},s***	***$T_{Arrival,}$ s***
Value	70	7	0.64	4.01	96.03

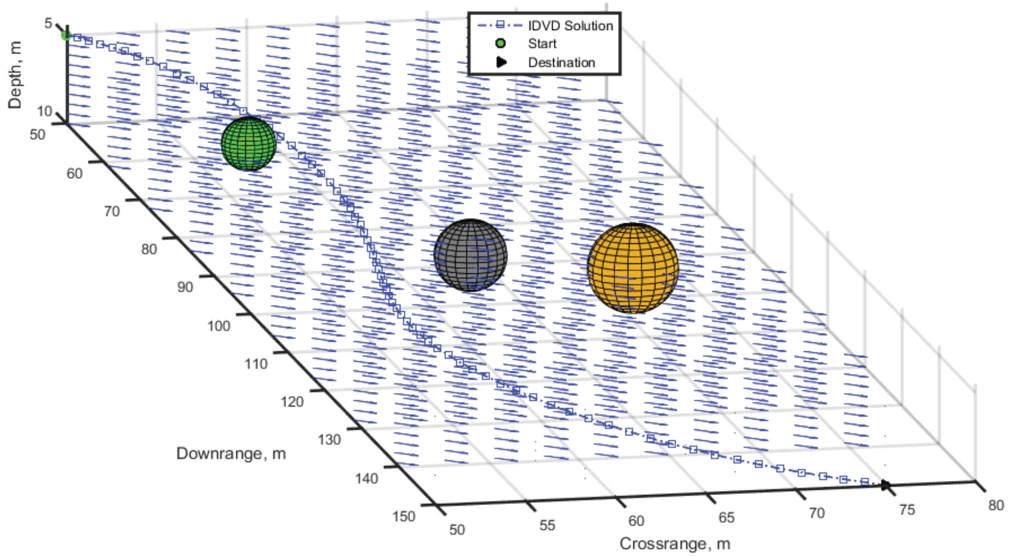


Figure 5.19 Collision-free 3D path for the minimum-time docking scenario.

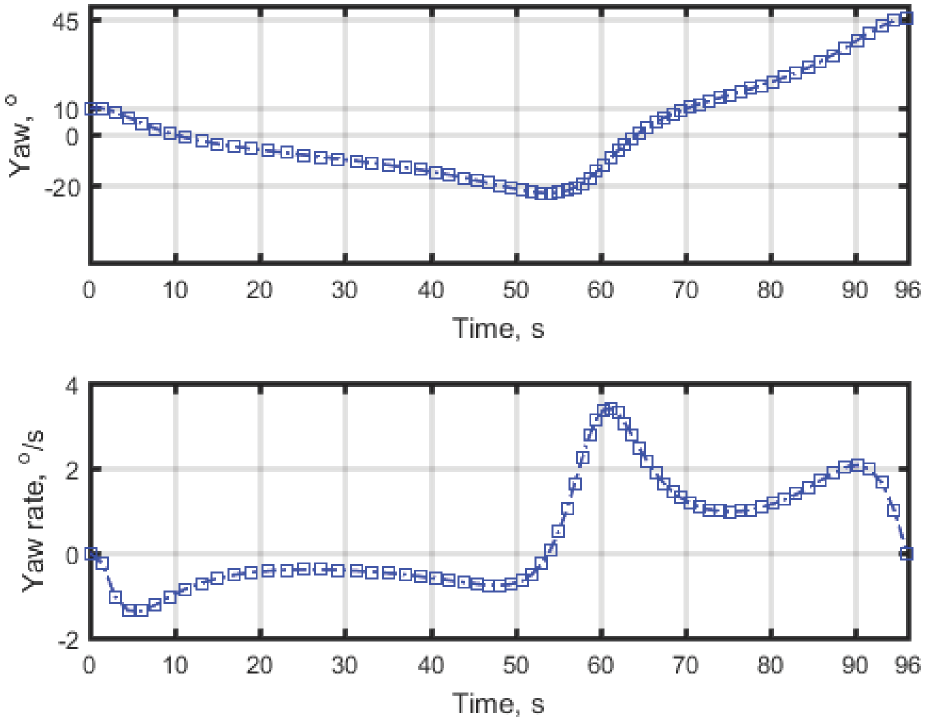


Figure 5.20 Heading and heading rate trajectories for the minimum-time scenario.

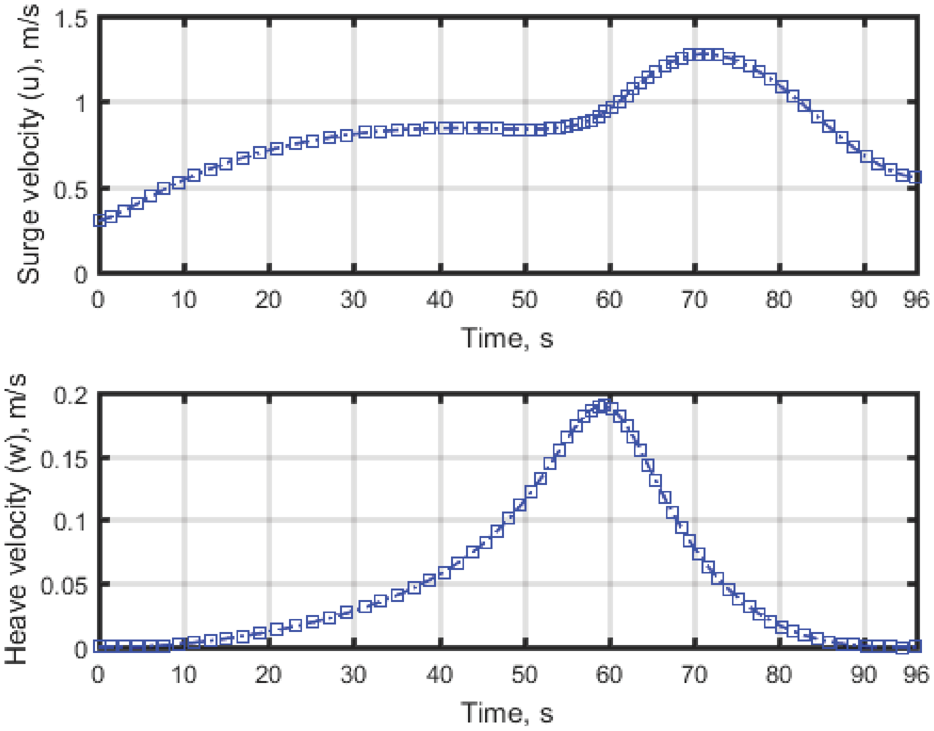


Figure 5.21 Surge and heave trajectories for the minimum-time scenario.

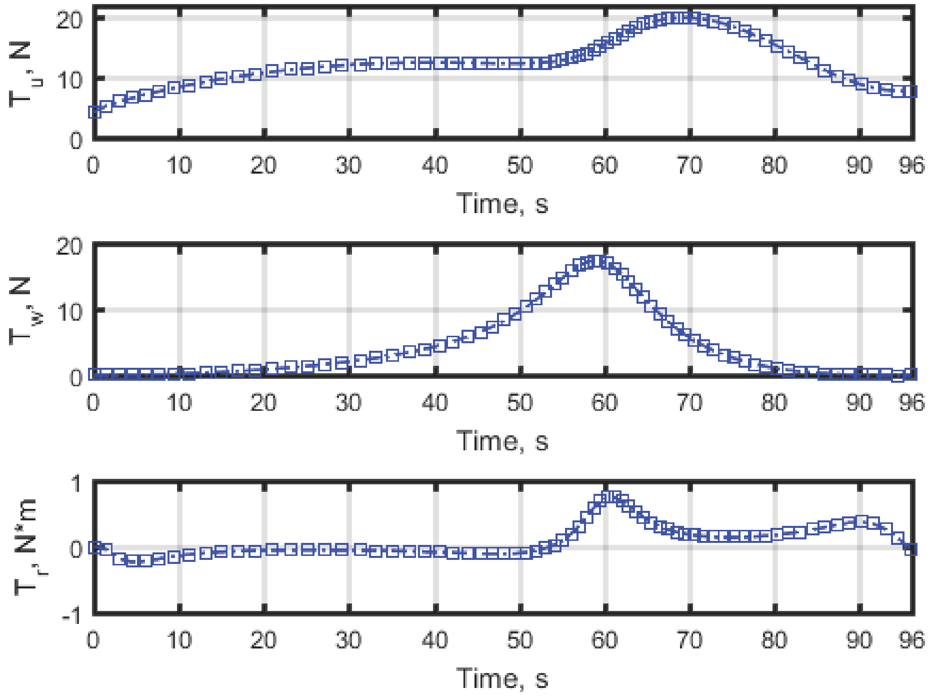


Figure 5.22 Evolution of controls for the minimum-time scenario.

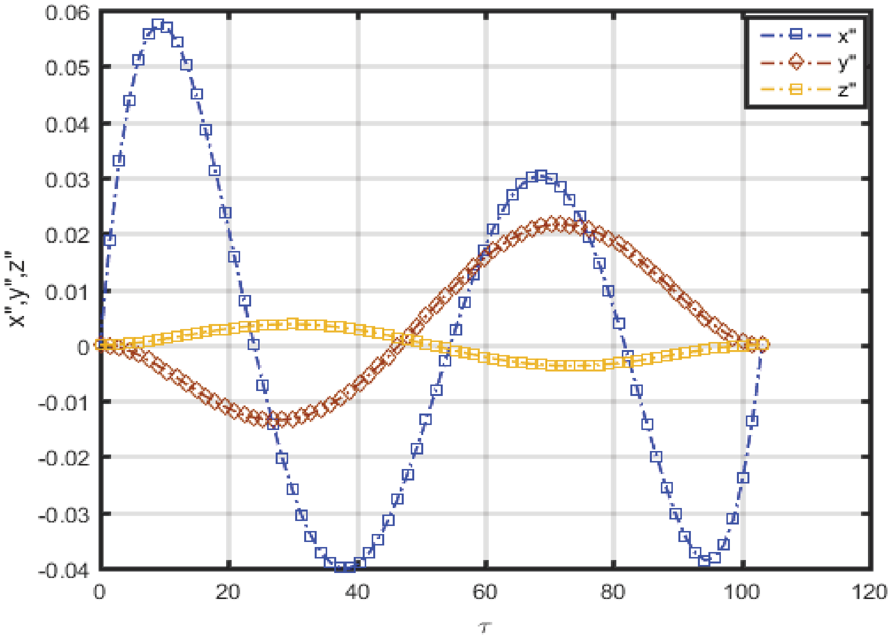


Figure 5.23 Higher order derivatives of spatial coordinates for the minimum-time scenario.

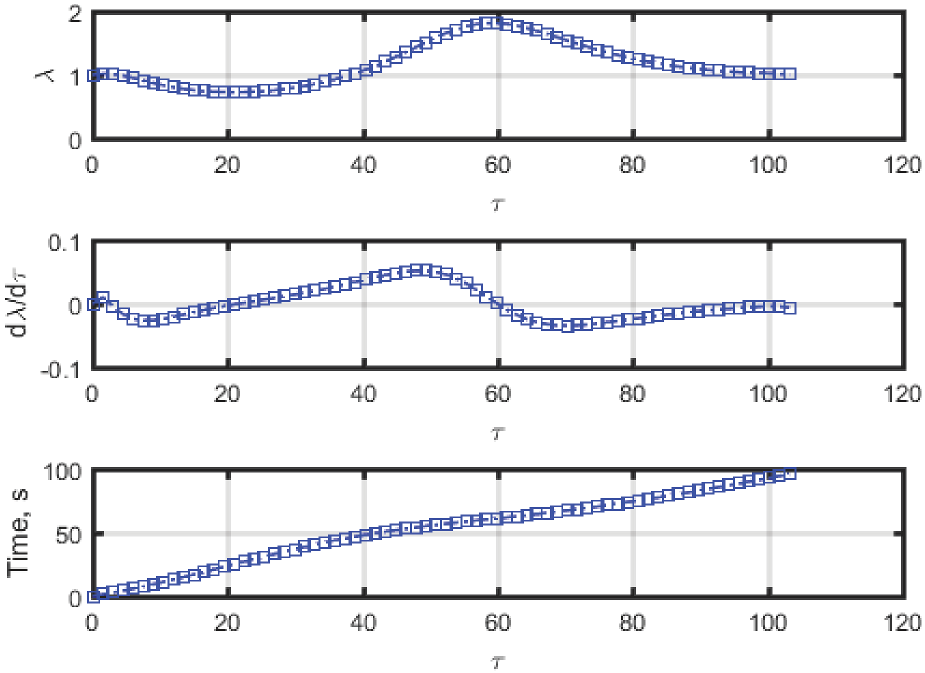


Figure 5.24 Evolution of the speed factor and mapping between virtual space and time domain for the minimum-time scenario.

Table 5.8 quantitatively shows the simulation results of this scenario. As can be observed from the table, the normalized performance index obtained is 0.64 which is equivalent to 96.03 seconds for the vehicle to arrive into the DS cone, indicated by

$T_{Arrival}$ in the table. Comparing to the Minimum-Energy docking scenario, the vehicle is able to terminate the docking operation in a relatively 19.97% shorter time while takes into account the problem's constraints. Regarding the salient feature of the IDVD method in using a few number of optimization variables (7 here), a real-time solution is achievable; this confirms by the computational time of the solution that is in the order of less than 10% of the arrival time (see t_{CPU} in the table).

Figure 5.19 shows the 3D path generated by the IDVD method for the proposed minimum-time solution. As can be seen from the figure, the vehicle is able to show a collision-free maneuver from its initial position to the DS position. In Figure 5.20, one can observe the smooth transition of the vehicle's heading accompanied with the heading rate from the starting condition toward aligning the vehicle's head with the DS direction at the terminal docking phase. Figure 5.21 shows that to achieve the minimum-time solution the vehicle needs to increase both surge and heave velocities. In this case, the quantity of the surge velocity at the terminal point is 0.55 m/s. In Figure 5.22, the evolution of the controls is demonstrated. Comparing to the Minimum-Energy docking scenario, a relatively 22.37% more control efforts are used by the vehicle to obtain a relatively 19.97% shorter docking operation time. In this situation, the IDVD method still generates a smooth control trajectory while satisfying the bounds on the vehicle's actuators. This smooth transition of the control trajectory is provided by satisfaction of the second-order terms derivatives at the terminal point as indicated in Figure 5.23. In this regard, the evolution of speed factor is of the form Figure 5.24.

5.3.2 Validation of Solution Using the SITLSP

In this section, the SITLSP is employed to check the fidelity and tractability of the minimum-time solution. Figures 5.25-5.28 show the results of this test. As can be seen from the figures, there exists consistency between the IDVD solution and the SITLSP performance. In Figure 5.25, it is observed that the vehicle is able to avoid NFZs and smoothly can transit into the DS. Figure 5.26 supports the statement that the final alignment of the vehicle with the DS is achievable even though at the first 10 second of the yaw and yaw rate trajectories a sort of disjoints can be observed. As mentioned before, these disjoints are consequences of the Traveling WP block in the SITLSP which generates waypoints 1-meter ahead of a reference trajectory. The artefact of the Traveling WP block functionality is observable in a form of phase shift in surge and heave velocity profiles as indicated in Figure 5.27. However, even in this case, the general pattern of the trajectories is analogous with what obtained by the IDVD and the strong convergence at the boundary conditions is acquirable. Finally, in Figure 5.28 the evolution of controls is demonstrated. As can be seen from the figure, there exist a perfect match between the controls associated with the main propeller and lateral thrusters. The vertical thrusters' profile in the SITLSP exceeds the boundary assigned on the optimization procedure for approximately 7 seconds. However, as mentioned in Chapter 3, the real bounds on the thrusters and propeller are considerably greater than quantities used in the optimization procedure. The purpose of this design is to have a more conservative trajectory generator system that is implementable for a real vehicle. Therefore, the SITLSP vertical thrusters' profile in Figure 5.28 is acceptable for the proposed vehicle.

Table 5.9 quantitatively presents that in the minimum-time docking performance all metrics of successful docking are perfectly satisfied within the defined tolerances.

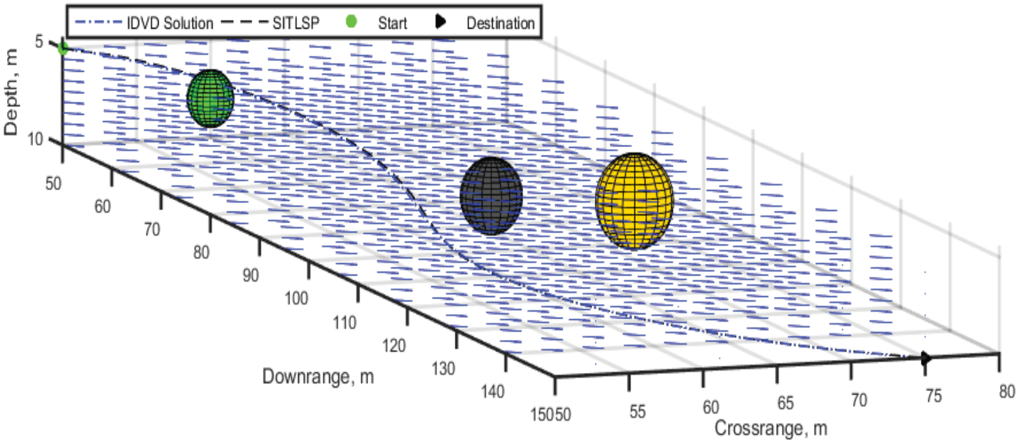


Figure 5.25 Evaluation of 3D path in the SITLSP for the minimum-time scenario.

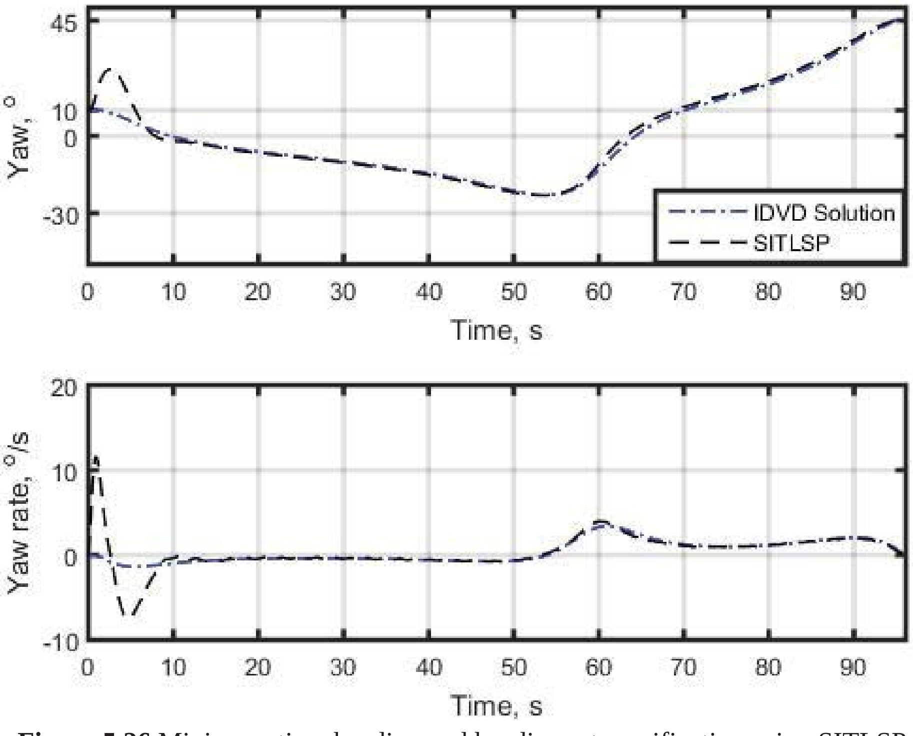


Figure 5.26 Minimum-time heading and heading rate verification using SITLSP.

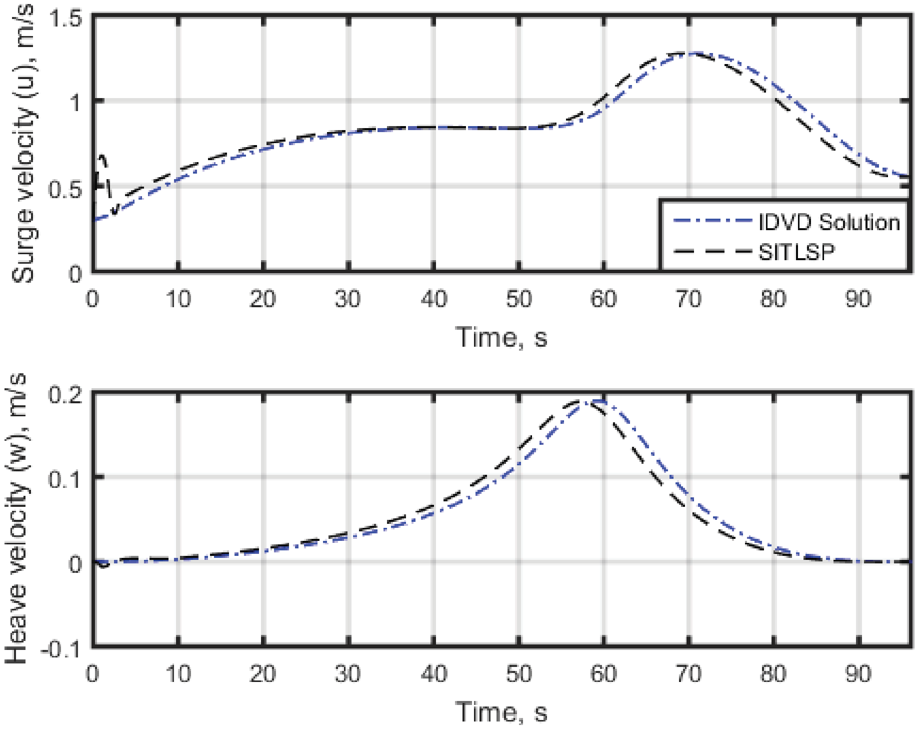


Figure 5.27 Minimum-time surge and heave velocities verification using SITLSP.

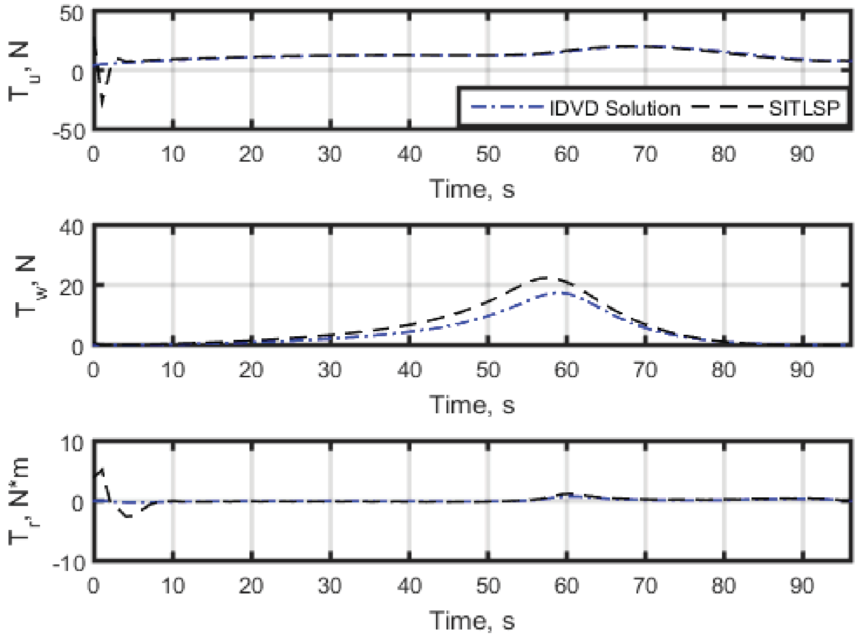


Figure 5.28 Minimum-time evolution of the controls in the SITLSP.

Table 5.9 Quantitative evaluation of the IDVD performance for docking-enabling conditions for the minimum-time docking scenario.

Metric	$\Delta\eta_{,m}$	$\Delta\psi,^{o}$	$\Delta u_{,m/s}$	$\Delta w_{,\ m/s}$	$\Delta r,^{o/s}$
Value	0.1	0.34	4.7e-05	2e-05	0.16

5.4 Online Trajectory Generation for a Realistic Docking Operation

This section introduces an online docking guidance system with the capability of trajectory refinement based on situational awareness of the AUV's operating maritime environment. The design is established based on the significant features of the IDVD method in fast prototyping of near-optimal trajectories, as thoroughly demonstrated in the previous sections. This online guidance method enables the AUV to perform docking operation in an environment where the offline map is only partially or not at all available. That comprises a reactive environment with unknown or semi-known DS pose; or docking with respect to a variable DS pose that is a moving/floating station. The concept of RHC introduced in Chapter 2 is explicitly used to establish the online system in a closed-loop configuration. Within a representative scenario and using the SITLSP, performance of the online guidance system is investigated.

5.4.1 Problem Formulation

The docking operation in a marine environment, particularly in a region where an offline map is not available or incomplete, is not an easy task as the vehicle encounters a spectrum of vehicular and environmental variabilities and uncertainties. Dynamic uncertain obstacles/NFZs, variable ocean currents, irregularly shaped terrains, malfunctions of navigational payloads, and uncertainty with sensory information are factors that can make the docking operation complex and challenging. Unlike the docking operation in a controlled environment where the DS pose is known a priori and no uncertainty exists, in a realistic environment the vehicle must be capable of performing the docking operation with respect to unknown, semi-known or variable DS pose. These conditions generalize the docking operation with static, floating, and moving stations. In details, the vehicle should be able to perform its docking operation where a rough estimate of relative DS pose is available a priori and DS pose updates are provided during the mission. This also can be treated as a representation of a docking operation with a moving station in which the vehicle is able to receive a sequence of DS pose updates in certain time instances, once it reaches to a pre-determined rendezvous point.

In aforementioned examples, an offline guidance system, cannot guarantee a successful docking operation and definitely results in failure of mission. Therefore, taking advantage of an online guidance system is essential. For developing an online guidance system, as mentioned in Chapter 2, two basic prerequisites are essential; the first is, the method employed for trajectory generation should be able to generate a feasible trajectory in a very efficient computational time, preferably in a real-time manner; the second is, a closed-loop configuration which ensures stable repetitive trajectory computations should be at hand.

Regarding the salient feature of the IDVD method in fast prototyping of feasible and near-optimal trajectories, this method together with concept of RHC is employed to develop an online docking guidance system tailored for docking with static and/or moving recovery stations.

The following docking scenario represents a situation where there exists an uncertainty in the actual position of DS. In this scenario, it is assumed that the AUV executing a docking mission gets an update about DS pose every t_{update} using an USBL sensor. The DS pose information is corrupted by both sensor and environmental noises. As the AUV approaches the DS, the impact of uncertainty is reducing, and the DS pose information becomes more accurate. The USBL acoustic positioning system used in this scenario (denoted as SA Sensors in the SITLSP) is adopted from [155]. Moreover, to make the scenario more challenging a cluttered environment with six NFZs modelled as (3-41) is simulated. In this environment, a 2D current disturbance of magnitude c_x=0.25 m/s, c_y=0.25 m/s with respect to the North and East respectively in the $\{n\}$ frame, is considered. The initial and final conditions together with the constraints over the AUV's states and controls are set as follows:

$$\begin{aligned} \mathrm{X}_0 &= [50\mathrm{m}, 50\mathrm{m}, 5\mathrm{m}, 10^{\circ}, 0.3\mathrm{m/s}, 0\mathrm{m/s}, 0\,^{\circ}/\mathrm{s}]^T \\ \mathrm{X}_f &= [x_f(D_r)\mathrm{m}, y_f(D_r)\mathrm{m}, z_f(D_r)\mathrm{m}, 45^{\circ}, u_f\ \mathrm{m/s}, 0\mathrm{m/s}, 0\,^{\circ}/\mathrm{s}]^T \end{aligned} \tag{5-33}$$

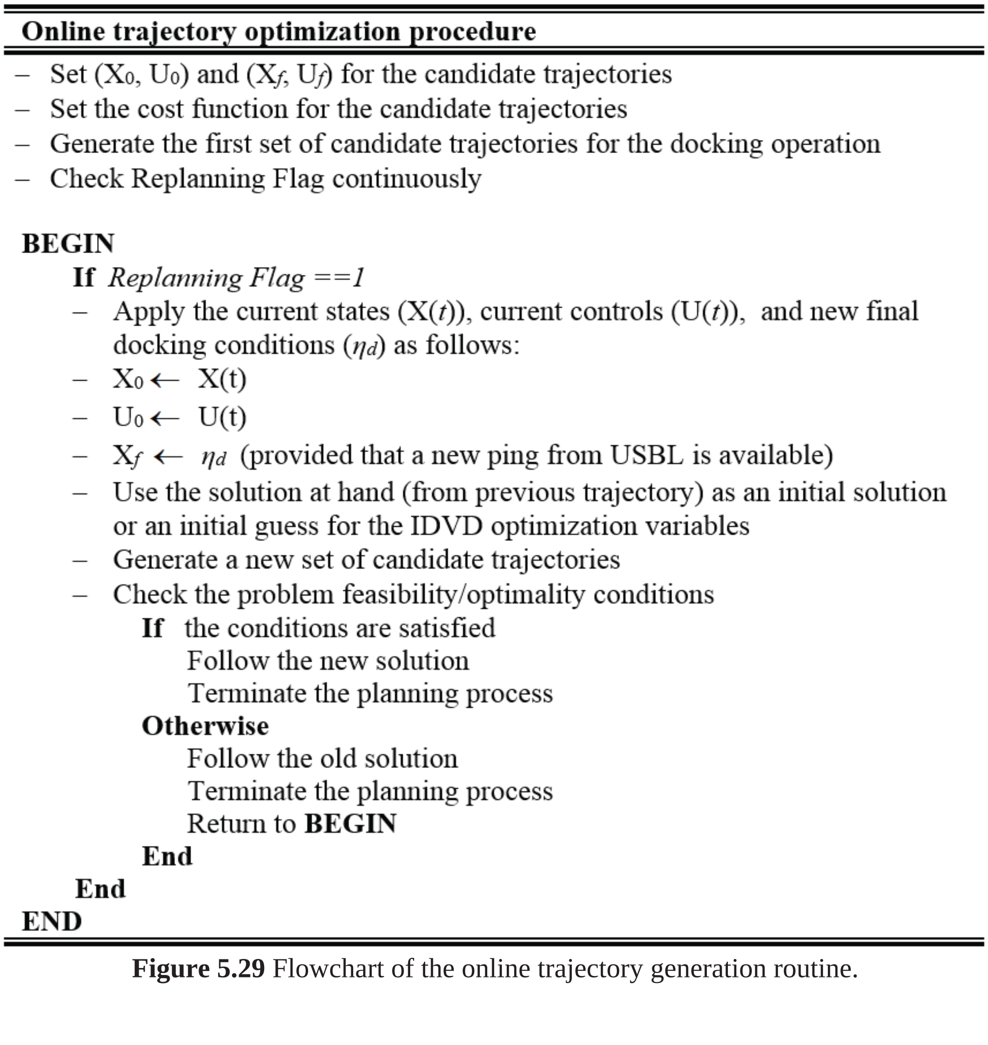
$$\begin{aligned} T_u^{\max} &= T_w^{\max} = 20\mathrm{N} \\ T_r^{\max} &= 15\mathrm{N.m} \\ r^{\max} &= 15\,^{\circ}/\mathrm{s} \end{aligned} \tag{5-34}$$

As opposed to the fixed values of final spatial coordinate (DS position) and surge velocity used in Chapter 4 and 5, the first three elements of the final state vector are dependent on the range from the true DS position, D_r, and modeled as

$$\begin{aligned} x_f(D_r) &= 180(1+\delta(D_r)), \mathrm{m} \\ y_f(D_r) &= 70(1+\delta(D_r)), \mathrm{m} \\ z_f(D_r) &= 11(1+\delta(D_r)), \mathrm{m} \end{aligned} \tag{5-35}$$

where $\delta(D_r) = N(0, \sigma(D_r)^2)$ represents the DS position with a normally distributed uncertainty ($\sigma(D_r) = 4.1 \times 10^{-6} D_r^2$). The u_f is set free to provide more flexibility for the vehicle in using its total range of maneuvering.

As a result, at each DS position update, every t_{update}, the AUV reference trajectory needs to be recomputed to account for an updated $\mathrm{X}_f\big(D_r(t)\big)$. When doing so, the trigger corresponding to new trajectory generation (it is called Re-planning Flag here) is activated; then, current AUV states and controls are used as new initial states $\mathrm{X}_0 = \mathrm{X}(t)$ and controls$\mathrm{U}_0 = \mathrm{U}(t)$. In this scenario, the concept of RHC is explicitly used in such a way that it is suitable for slow dynamics nature of the AUV and docking operation. In this regard, $t_{horizon} \equiv t_{update}$ and t_{sample} is set to termination time of one optimization run. Meanwhile, in this scenario, the docking performance is required to be performed with a minimum control effort. Therefore, the normalized cost function (5-31) is utilized. Also, the IDVD method setup is similar to what used in the Minimum-Energy scenario in Section 5.2. Figure 5.29 shows the flowchart of the utilized IDVD-based online trajectory generation (optimization) procedure.



```
Online trajectory optimization procedure
– Set (X0, U0) and (Xf, Uf) for the candidate trajectories
– Set the cost function for the candidate trajectories
– Generate the first set of candidate trajectories for the docking operation
– Check Replanning Flag continuously

BEGIN
    If  Replanning Flag ==1
    –  Apply the current states (X(t)), current controls (U(t)),  and new final
       docking conditions (ηd) as follows:
    –  X0 ← X(t)
    –  U0 ← U(t)
    –  Xf ← ηd (provided that a new ping from USBL is available)
    –  Use the solution at hand (from previous trajectory) as an initial solution
       or an initial guess for the IDVD optimization variables
    –  Generate a new set of candidate trajectories
    –  Check the problem feasibility/optimality conditions
          If  the conditions are satisfied
              Follow the new solution
              Terminate the planning process
          Otherwise
              Follow the old solution
              Terminate the planning process
              Return to BEGIN
          End
    End
END
```

Figure 5.29 Flowchart of the online trajectory generation routine.

5.4.2 Simulation Results

Figures 5.30- 5.33 demonstrate an example where a reference trajectory is updated twice based on two sequential USBL updates on the DS position. Figure 5.30, illustrates the generated path in 3D, revealing NFZs and the same current field as in the offline computation scenario of Section 5.3. Three solid circles in Figure 5.30 (including the starting point) indicate the DS position update points and three triangles show the corresponding perception of the DS position. Specifically, the Trajectory generator block in the SITLSP generates the first reference trajectory based on the DS position information available at the first solid circle, denoted as *Start*. At this point the DS position is thought to be at the location denoted as *1st destination*. With this reference trajectory, the AUV continues its motion to this destination. At the first update corresponding to the AUV position depicted as the second solid circle (*1st update*), the vehicle receives a new ping from USBL and based on the new information, the Trajectory generator block refines the reference trajectory and generates a new one leading to the *2^{nt} destination* point. The AUV keeps tracking the second reference trajectory until the next ping from the DS is received, and the 2nd update occurs. Another reference trajectory is then generated with respect to the updated estimate of the DS position denoted as the *Actual DS* in Figure 5.30 (at this point, being about 40 m away from the DS, its horizontal and vertical position is known within 1 m and 0.1 m, respectively). As seen in Figure 5.30, each of the three reference trajectories forces the AUV to maneuver around NFZs and that is where IDVD-approach capability to generate spatial non-singular arc solutions really pays off.

Figures 5.31 and 5.32 feature time histories of yaw angle and yaw rate and surge and heave velocities, corresponding to all three reference trajectories shown in Figure 5.30, respectively. Figure 5.31 clearly shows a correct final zero-yaw-rate AUV orientation aligned with that of the DS centerline even though there exists less than 1%violation on the third update of yaw rate for a fraction of seconds.

Data shown in Figure 5.33 proves that all controls are within their limits (within preset tolerances). In the particular realization shown in Figure 5.30, where the new DS position updates happen to be farther and farther away than originally expected, obtaining the new reference trajectories becomes more and more challenging. In this particular simulation, the only way to be able to arrive to the DS in exactly 120 seconds with zero acceleration is to allow the final surge velocity to vary. As seen from Figure 5.33, the surge control reaches its limit at the second update and is within 1% above its limit for the third update. The performance index value, tabulated in Table 5.10, is also indicative of this.

Generally, in online trajectory generation, it is possible that the underlying algorithm cannot find feasible solution or may converge to infeasible solutions in the estimated t_{sample} (around 12 s for the IDVD method). When doing so, vehicle should follow the previous control commands until receiving a new update and calculating a new trajectory. In this particular docking scenario, however, as the third trajectory is generated based on the last available ping of USBL and the proposed trajectory satisfies all boundary conditions, one can rely on it. In terms of improving the computational time, it is noteworthy to mention that by using more powerful and computationally efficient NLP solvers like IPOPT the IDVD computational efficiency obtained by f_{mincon} can be increased up to two orders of magnitude and therefore trajectory refinement in each cycle can be performed in a fraction of second.

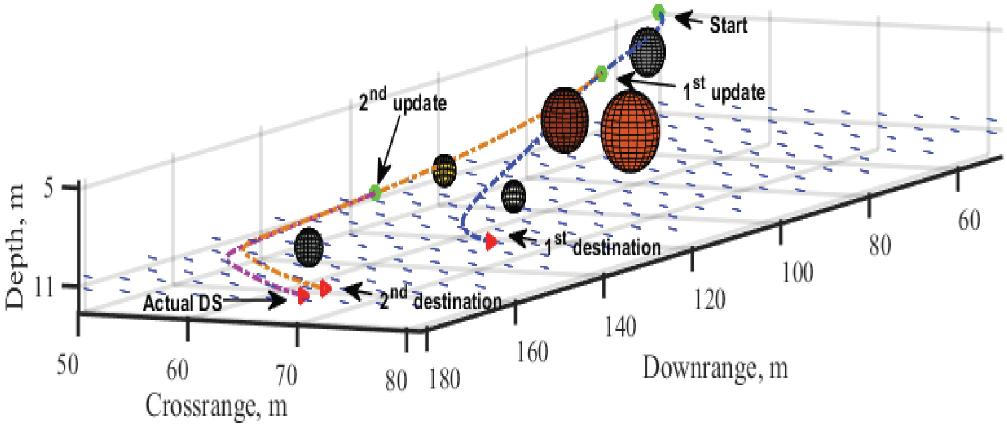


Figure 5.30 3D collision-free path re-optimized based on a better knowledge of the DS location.

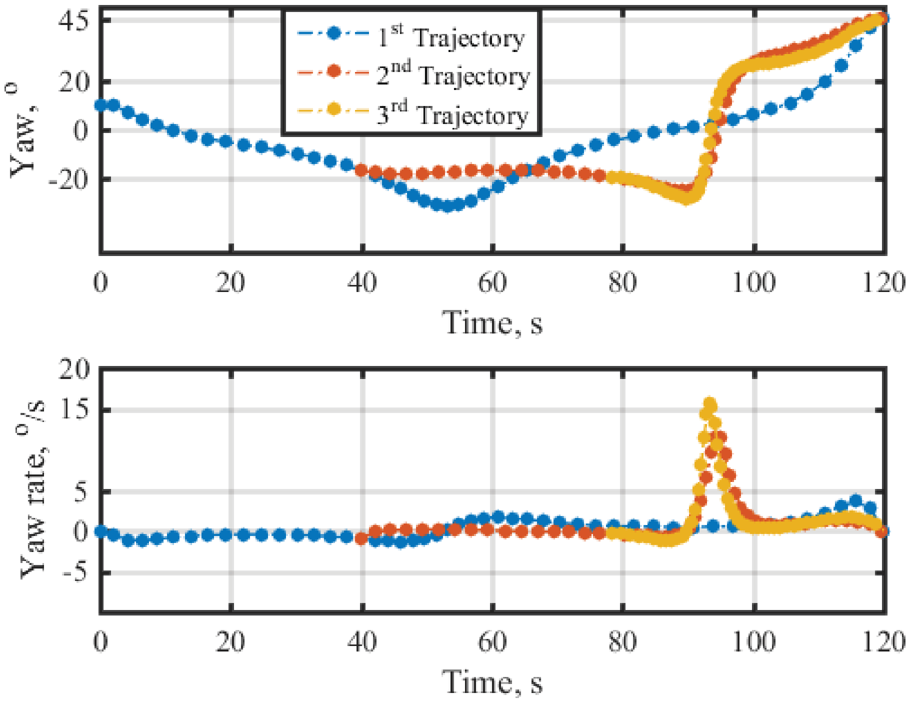


Figure 5.31 Refinement of yaw and yaw rate during the docking operation.

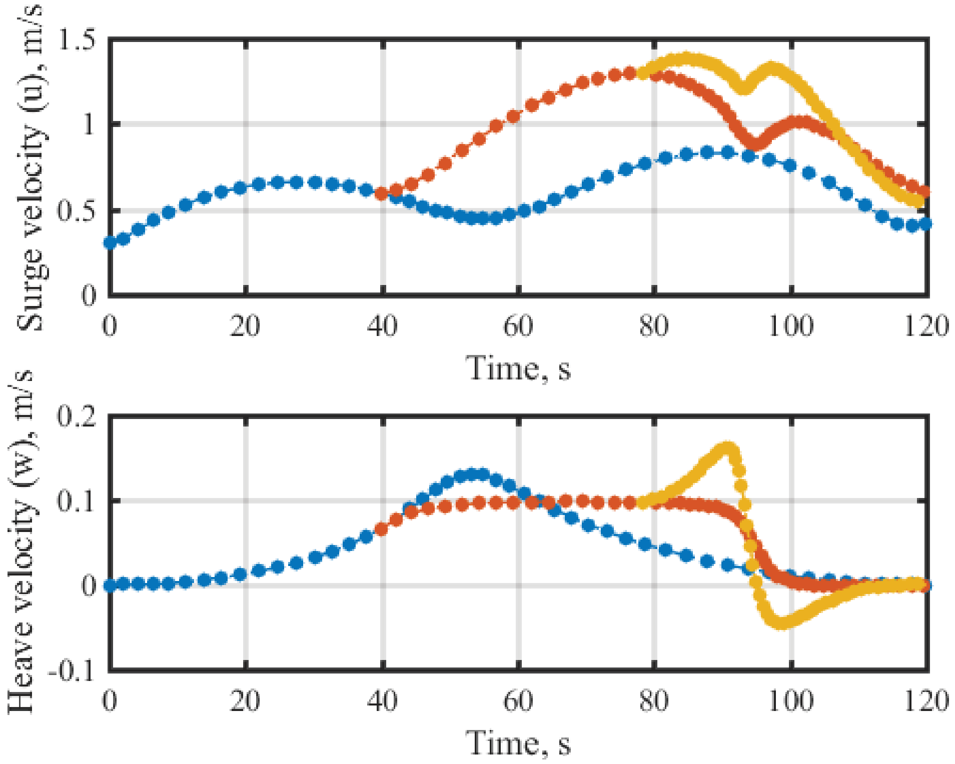


Figure 5.32 Refinement of surge and heave velocities during the docking operation.

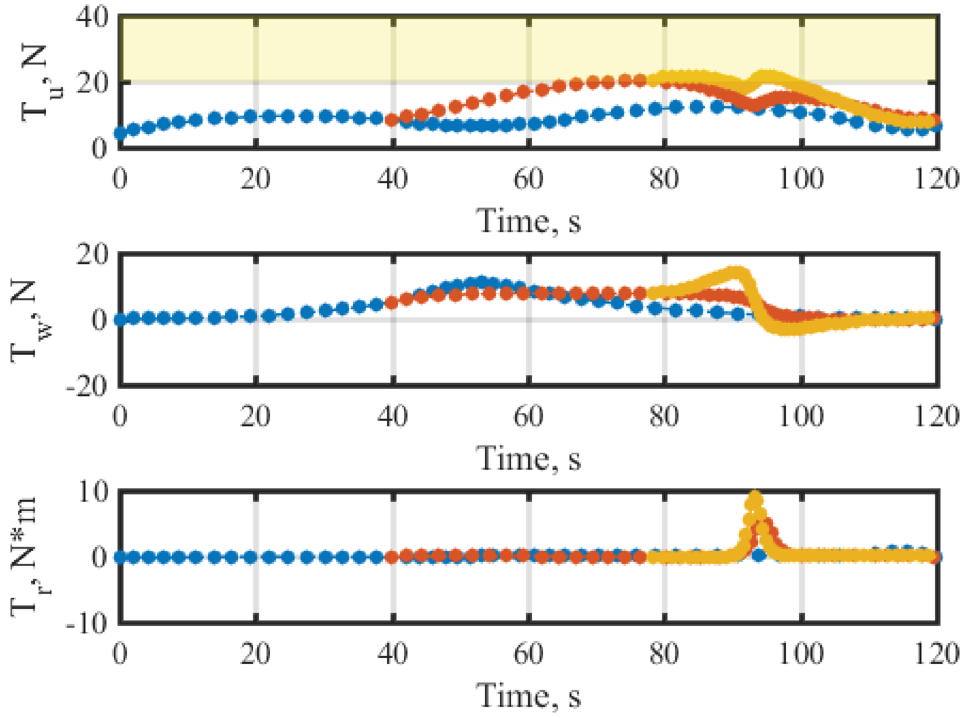


Figure 5.33 Time histories of the control inputs for online docking trajectory generation.

Table 5.10 Quantitative expression of the online trajectory generation performance.

Metric	***NoN***	***J***	***t_{CPU},s***	***ES%***
Value	50	0.6	12.5	22.54

5.4.3 Verification Using the SITLSP

In this section, tractability of the IDVD-based online guidance system is investigated using the SITLSP. Figure 5.34 indicates a correspondence between 3D collision-free path generated by the IDVD method and the SITLSP execution. In Figure 5.35, one can observe the realization of yaw and yaw rate trajectories generated during the online process for the SITLSP. The SITLSP yaw trajectory shows that the vehicle is able to align its head with the estimated DS orientation at the end of each update. Regarding the functionality of Traveling WP guidance block in the SITLSP, at the beginning of the trajectory, one can observe jumps and disjoints. This artefact is more significant at the beginning of the yaw rate trajectory in Figure 5.35 but the jump still respects the physical limitation on yaw rate. Figure 5.36 demonstrates the refinement of surge and heave velocities within the SITLSP; in this figure, the trajectories are properly tracked by the SMC controller. Finally, Figure 5.37 illustrates the time history of controls generated by the SITLSP with respect to the sequence of updates; this figure indicates the realization of solutions generated by the online guidance system for the vehicle. Table 5.11 indicates that docking-enabling conditions are met by the SITLSP trajectories and therefore the online guidance system enables the vehicle to successfully complete the docking operation even in such a cluttered and uncertain environment.

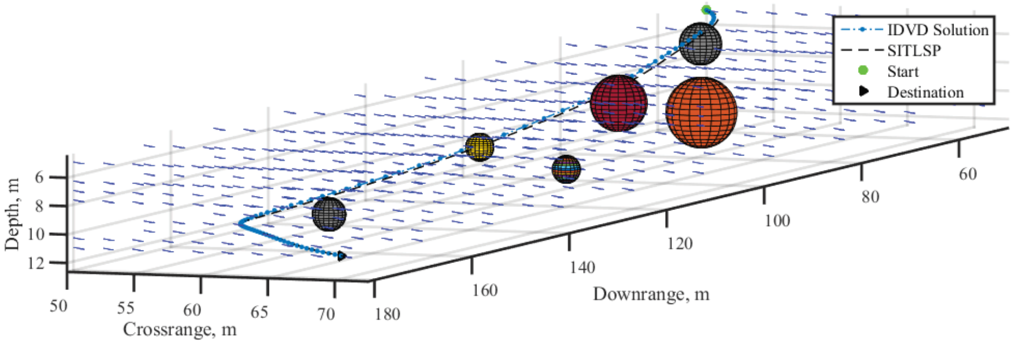


Figure 5.34 3D collision-free path re-optimized based on the SITLSP implementation.

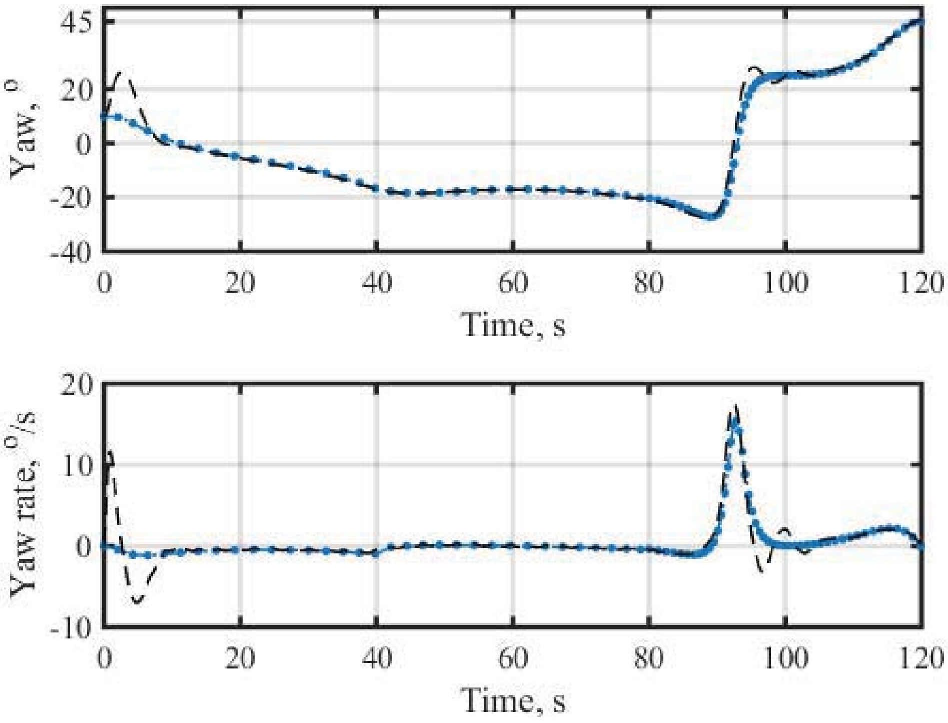


Figure 5.35 Refinement of yaw and yaw rate based on the SITLSP.

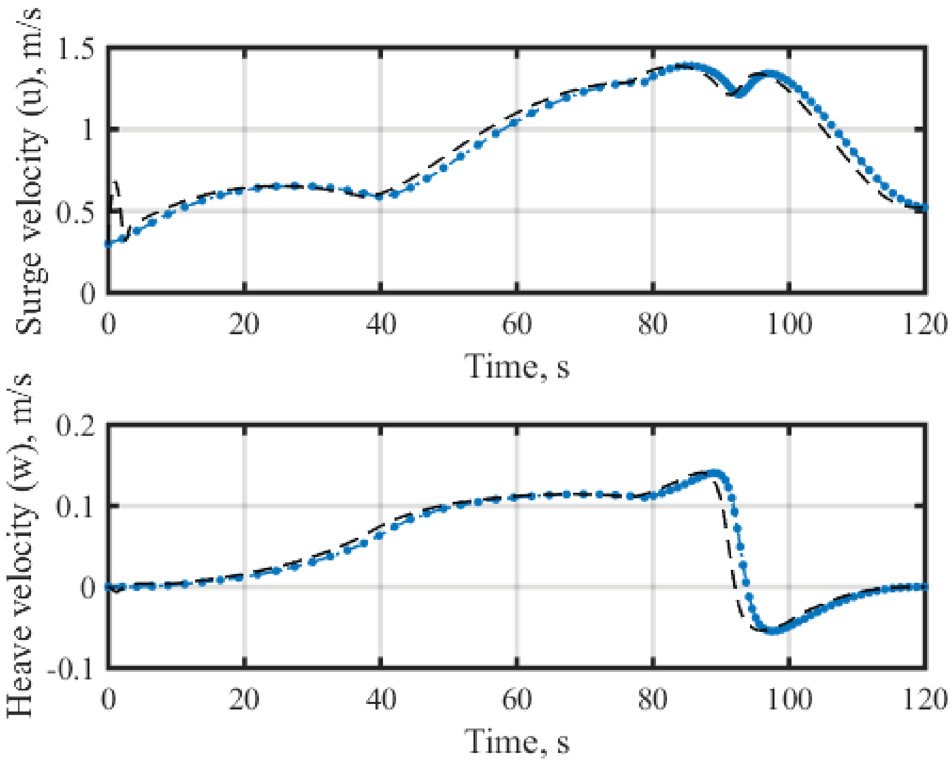


Figure 5.36 Refinement of surge and heave velocities based on the SITLSP.

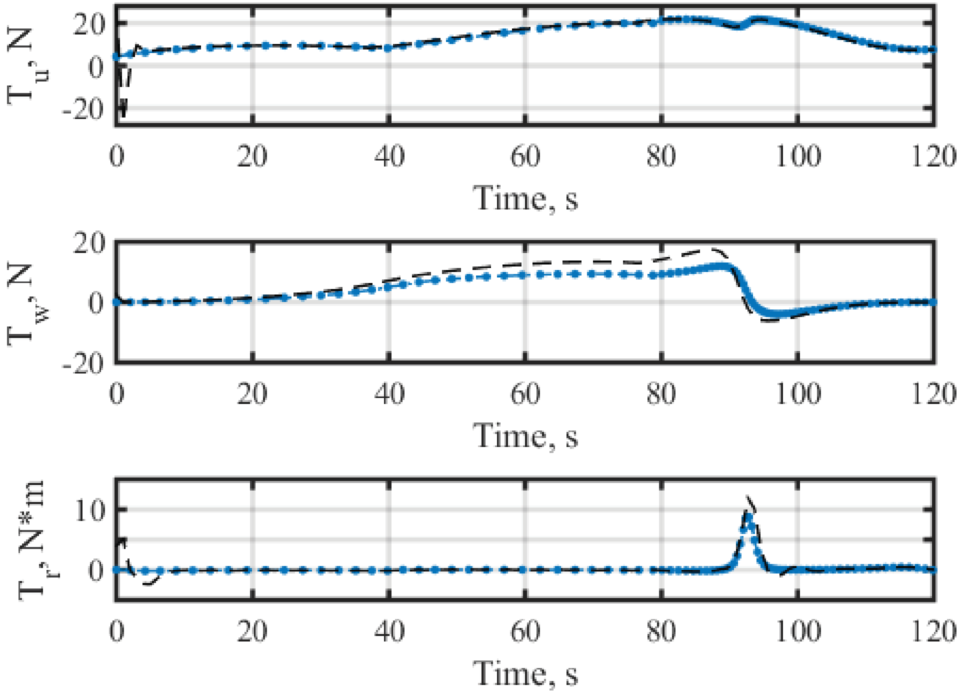


Figure 5.37 Refinement of controls based on the SITLSP.

Table 5.11 Quantitative evaluation of the IDVD performance for docking-enabling conditions using the SITLSP.

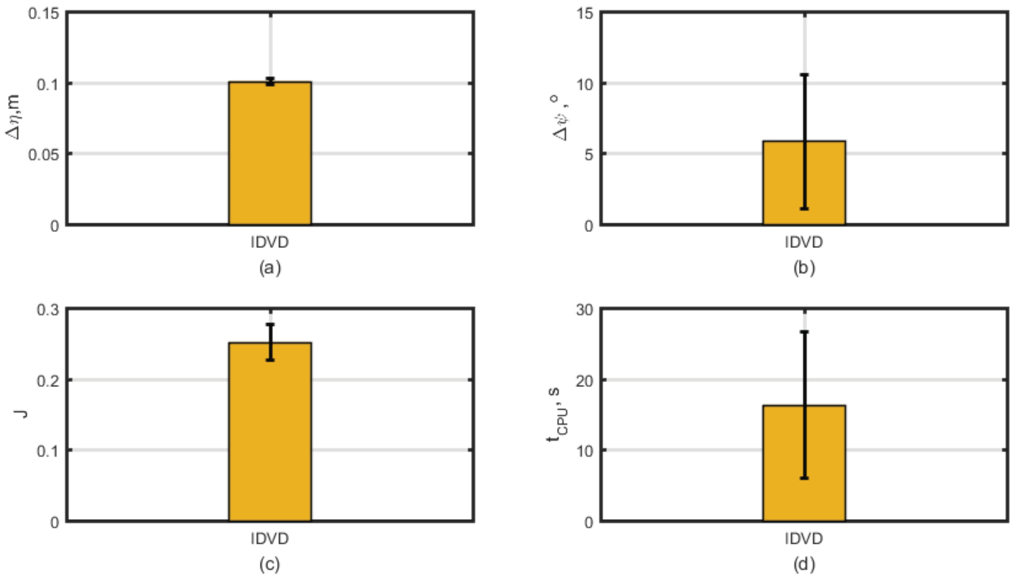
Metric	***$\Delta\eta$,$_m$***	***$\Delta\psi$,^{o}***	***Δu,$_{m/s}$***	***Δw, $_{m/s}$***	***Δr,$^{o/s}$***
Value	0.24	0.74	2.15e-6	3.9e-5	1.34

5.5 Robustness Assessment Using Monte Carlo Simulations

In this section, similar to Chapter 4, the MC simulations are implemented to investigate the robustness of the IDVD method under uncertain conditions. Here, the robustness assessment is applied for the Minimum-Energy docking scenario based on generation of 200 random samples for initial pose of the vehicle, pose of the DS, and random cross-current disturbances.

For the sake of reminding, in the first test, the initial pose of the vehicle $\eta_0 = [x_0, y_0, z_0, \psi_0]^T$ is changed to $\eta_{\wp_0} = [\wp^{x_0}, \wp^{y_0}, \wp^{z_0}, \wp^{\psi_0}]^T$ during 200 MC trials; a 10% uncertainty is applied to the hydrodynamic coefficients used in the SITLSP, while the other conditions are kept unchanged. Four performance metrics of normalized cost (J), execution time (t_{CPU}), final position error ($\Delta\eta$) and final heading error ($\Delta\psi$) are utilized to investigate the impact of uncertain conditions on the IDVD performance. In this test, the amount of uncertainty applied to the initial pose vector is described by the following relations: $\wp^{x_0} \in x_0 \pm 5$ m, $\wp^{y_0} \in y_0 \pm 5$ m, $\wp^{z_0} \in z_0 \pm 2$ m, $\wp^{\psi_0} \in \psi_0 \pm 20^\circ$.

Figure 5.38 shows the results of the first test. In this figure, the metrics of the $\Delta\eta$ and $\Delta\psi$ metrics (parts (a)-(b)), determine whether the vehicle is still able to meet the docking-enabling conditions or not. Comparing to the results of the first scenario tabulated in Table 5.3 (without uncertainty), the random initial poses impact on enlarging the average of the final position and orientation errors. However, the IDVD shows proper robustness in terms of satisfying the docking enabling-conditions as the corresponding final position and heading error satisfy the required tolerances for a successful docking operation. Under the random initial pose conditions, the IDVD is able to provide average of 49.8% with ±2.5% variations in terms of the ES, as referred to the part (c) of Figure 5.38. The overall energy expenditure of the vehicle in the uncertain conditions has a slight difference with that of tabulated in Table 5.1. The computational efficiency of the IDVD method, is measured based on the execution time, indicated in the part (d) of Figure 5.38. The average of execution time is 16.3 seconds with 10.33 seconds variations under the random initial pose conditions. The random initial pose conditions apply 41% increase on average of the computational time when it is compared with IDVD computational time mentioned in Table 5.1. However, the IDVD still shows fast convergence rate as the average of the computational time is about 13% of the mission time.



(a)

(b)

(c)

(d)

Figure 5.38 Results of Monte Carlo trials with uncertainty in the initial pose; the bars and vertical lines represent the means and standard deviations, respectively.

In the second test, the uncertainty is applied to the DS pose $\eta_d = [x_d, y_d, z_d, \psi_d]^T$and the rest of the conditions kept unchanged. Similar to the first test, 200 random samples are generated and a random DS pose vector of $\eta_{\wp_d} = [\wp^{x_d}, \wp^{y_d}, \wp^{z_d}, \wp^{\psi_d}]^T$ is constructed. The amount of uncertainty applied on the DS pose parameters is defined as follows: $\wp^{x_f} \in x_f \pm 5$ m, $\wp^{y_f} \in y_f \pm 5$ m, $\wp^{z_f} \in z_f \pm 2$ m, $\wp^{\psi_f} \in \psi_f \pm 20^{\circ}$.

Figure 5.39 illustrates the performance results of the IDVD method for variations of the DS pose through the MC runs. Compared to the previous test, here, the uncertainty in the DS pose shows more impact by increasing the variations of the final position and heading errors, as shown in Figure 5.39, parts (a)-(b). Still in this situation, the IDVD method shows robust performance in terms of meeting the corresponding tolerances to posture the vehicle with a correct alignment within the cone of the DS. The IDVD method in this test shows average of 48.1% with ±6.1% variations in terms of the ES metric. Comparing to the previous test, the variations of energy expenditure demonstrate 35% increase that indicate the greater impact of random DS pose on the performance index. Lastly, part (d) shows the average execution time of 8.33 seconds with variations of 6.2 seconds for the IDVD performance in this scenario. Comparing to the previous scenario, it can be realized that random DS pose conditions have less impact on the average of the IDVD computational time. This indicates that the IDVD method keeps its high computational efficiency even under variations of the DS pose, due to using 5% of the docking operation time on average to generate a feasible near-optimal trajectory.

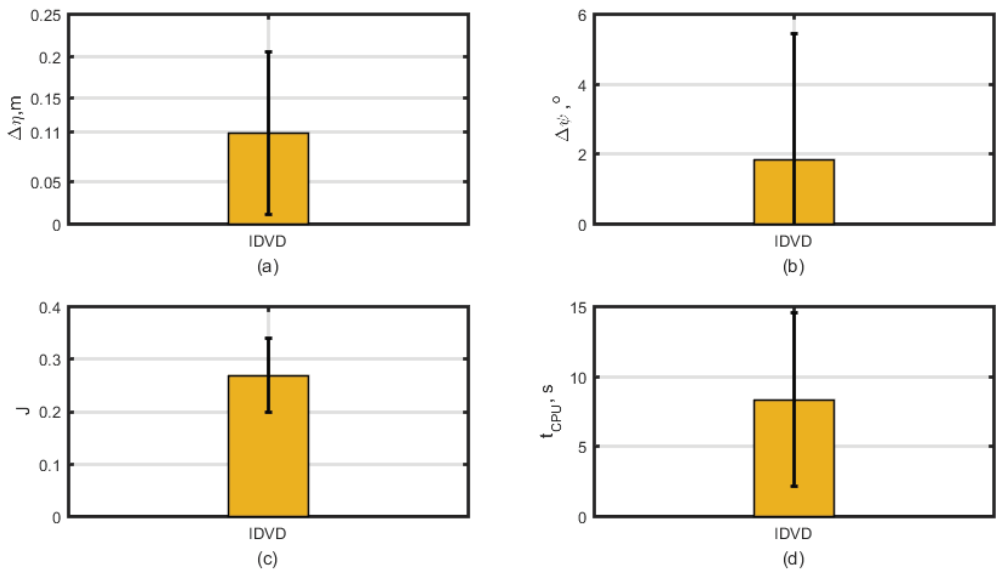


Figure 5.39 Results of Monte Carlo trials with uncertainty in the final pose.

In the last test, it is desired to investigate the impact of strong cross-currents on the overall performance of the IDVD docking guidance system. Similar to Chapter 4, the current components $V_c = \left[c_x, c_y\right]^T$ are changed to the $V_{c_\wp} = [\wp^{c_x}, \wp^{c_y}]^T$ where $\wp^{c_x}$=0 m/s and $\wp^{c_y}$~0.2-0.4 m/s for all MC simulations. The other conditions are kept unchanged. Figure 5.40 depicts the results of the MC simulation for this test. It is observed from the figure that random cross-current disturbances impose much control efforts on the vehicle for a feasible docking performance compared with two previous MC tests and similar result in Table 5.3. More specifically, in this test the IDVD method shows average of 30.7% with ±1.42% variations in terms of the ES metric. The random cross-current disturbances also show more impact on the final heading error compared with the two previous MC tests. Here, the final alignment of the vehicle with the DS centerline shows average of 8.5 ^{o} and variations of 5.45 ^{o}. One can observe that, even in this scenario, the required tolerance of the final heading error is perfectly met. This is also the case with the final position error. Accordingly, the IDVD method still shows computational efficiency with average of 12.73 second computational time for generating a near-optimal trajectory in presence of the random cross-current disturbances. This computational time is 10.6% of the docking operation time indicating the possibility of generating real-time solutions.

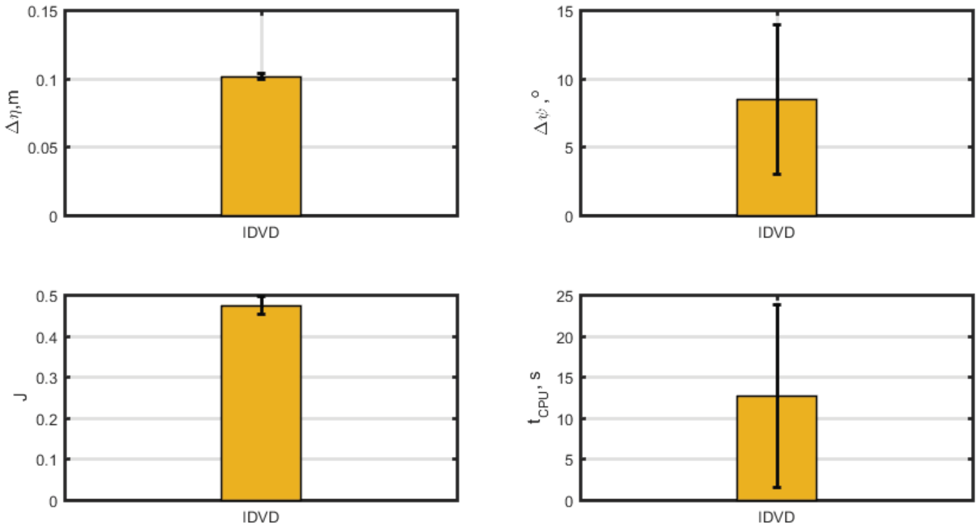


Figure 5.40 Results of Monte Carlo trials with uncertainty in the cross-current disturbance.

5.6 Chapter Summary

In this chapter, the IDVD method is employed for fast prototyping of near-optimal docking trajectory. This method uses the general idea of differential flatness to reduce the dimension of optimization problem and therefore is able to generate real-time and online solutions. Unlike the PS methods in Chapter 4, the IDVD method does not exploit the Pontryagin principle and therefore the solutions generated are near-optimal, as opposed to optimal. However, this loss of optimality is not much when compared with the PS counterparts. In contrast, it shows superior performance in terms of computational efficiency due to using only a small number of optimization variables, 7 variables in this chapter. In this chapter, through two major Minimum-Energy and Minimum-Time docking scenarios, all features of the IDVD method for both offline and online applications were investigated. Through these docking scenarios, the following salient features of the IDVD method are confirmed:

- Satisfaction of boundary conditions at every iteration even when exposed to uncertainties;
- Smooth transition of a trajectory from initial boundary conditions to the endpoint constraints;
- Satisfying arbitrary path constraints and constraints due to the vehicles' dynamics;
- Treating higher-order terms derivatives or equivalently controlling the terminal point of the control profiles;
- Remarkable heuristic in finding local optima using any arbitrary initial guesses;
- High computation efficiency regarding using a few number of optimization variables (7 variables in this chapter) and therefore tailoring for an on-board implementation.

These superiorities are verified with the SITLSP which closely matches the vehicle operating in a realistic marine environment. Using the SITLSP, it is observed that the IDVD method shows significant performance in meeting docking-enabling conditions. The tolerances associated with these conditions are perfectly satisfied especially in terms of final locating and aligning the vehicle within the DS cone.

The capability of the IDVD method to be used for a close-loop docking configuration is also investigated. To this end, the offline application of the IDVD method is extended to include continuous reshaping and regeneration of the optimal trajectories during the docking operation in an environment that the DS pose is not fully recognizable a priori. An online guidance system is developed that integrates the capability of the IDVD method in fast prototyping of feasible trajectories and an updating scheme explicitly browed from RHC concept. This online guidance system enables the vehicle to perform the docking operation with an uncertain DS (can be treated as a static or moving DS). The simulation results of a representative scenario in which the pose of DS is updated based on the USBL pings, indicate the effectiveness and fidelity of the proposed online system. The results of the SITLSP test show the fact that the vehicle is able to track the trajectory generated through the sequence of updates. The docking-enabling conditions are perfectly satisfied with respect to the estimated pose of DS. These results confirm the realization and tractability of solutions obtained by the IDVD-based online guidance system for on-board implementation on a real vehicle.

Through the MC trials, the overall robustness of the IDVD method in presence of the uncertainties is observed. It is inferred that the strong cross-current disturbances have more impact on the overall performance of the IDVD docking trajectory generation.

Chapter 6

Conclusions and Future Work

This Chapter summarizes the main components, contributions, key findings and achievements of this thesis. The direction for possible future work is also proposed.

6.1 Summary

This thesis presented a new guidance system framework for the purpose of autonomous underwater docking operations from the angle of optimal control theory. The guidance system allows combined homing and docking phases in a whole, and provides a smooth and stable arrival of the vehicle into the DS cone. Additionally, the proposed guidance system enables the vehicle to successfully accomplish a broad range of realistic docking operations in controlled, cluttered, and uncertain environments in both open- and closed-loop forms.

Chapter 1 provided a discussion on the importance and need for employing docking stations together with the associated difficulties of autonomous docking operations (Section 1.3). This chapter encapsulated the thesis objectives and contributions in Sections 1.4 and 1.6, respectively.

Chapter 2, at first, thoroughly reviewed existing configurations of docking stations and their relevant AUVs together with the state-of-the-art in docking guidance systems. Secondly, this chapter considered the importance and advantages of trajectory planning approaches as opposed to path following and way-point-based approaches for designing and developing docking guidance systems. Chapter 2, then, introduced the perspective of optimal control theory and reviewed the most recent and applicable optimal control theory-based methods for the purpose of the trajectory generation in the autonomous underwater docking problem. The chapter concluded with exigency of the closed-loop guidance together with the applicable schemes for the on-board implementation.

Chapter 3 constructed the foundation of this thesis. It first introduced the Flinders AUV, its specifications, and the mathematical representation of reduced dynamic model tailored for this AUV. This chapter, then, in Section 3.3 showed how the PMP was employed to provide the unified framework for the underwater docking guidance design within minimum-energy and minimum-time problems. The proposed framework provided a proper degree of freedom to impose desired arrival and departure conditions corresponding to any specific docking operation, all possible path constraints, and realistic constraints associated with physical limitations of the vehicles such as bounds on the states and controls. Additionally, this framework offered conditions to investigate the feasibility/optimality of trajectories generated. In the subsequent section, Chapter 3 mentioned to exigency of employing direct methods of optimal control theory to provide numerical solutions for the emerged docking TPBVP. Chapter 3 in the last section, introduced the high-fidelity SITLSP together with the docking-enabling conditions as a useful tool capable of producing performance metrics for investigating the realization and tractability of trajectories presented in the subsequent Chapters.

Chapter 4 provided a comprehensive investigation on the main features of PS methods in generating numerical optimal solutions for the docking TPBVP developed in Chapter 3. Various configurations of the LPS, CPS and hp-AR methods were implemented and their performances were investigated through a series of docking scenarios. Regarding the objective of providing smooth arrival of the vehicle into the DS, the novel 10-state formulation was presented in Chapter 4. This 10-state formulation allowed setting endpoint constraints on the controls generated by the PS methods. With the clarity gained through experimentation, it was recognized that solutions obtained by the PS methods were very close to the true optimal solutions as they employed influential information of co-states and Hamiltonian in generating trajectories. This statement was confirmed with the post-optimality process based on co-states and Hamiltonian trajectories for both controlled and cluttered operating environments. The tractability of solutions generated by the PS methods for a field test trial were confirmed by the SITLSP regarding docking-enabling conditions. Chapter 4 finally investigated the robustness of the PS methods against uncertainties based on a series of Monte Carlo simulations. The results of this part also showed inherent robustness of PS methods with respect to randomly generated conditions. More specifically, in all of the aforementioned scenarios, the LPS and CPS methods demonstrated roughly similar performances; whereas, the hp-AR method showed superior performance in terms of optimality, tractability and computational efficiency when compared with the LPS and CPS methods.

Chapter 5 employed the IDVD method to numerically solve the docking TPBVP and to address all requirements (defined in Chapter 1) of the universal docking guidance framework. This chapter, at first, in Section 5.1, described detailed theoretical illustrations and computer simulation of the IDVD method. Then in the subsequent sections, Chapter 5 provided detailed discussions and analysis of the IDVD method's performance for both minimum-energy and minimum-time scenarios. The ultimate goal of developing the closed-loop guidance system or equivalently online trajectory generation for autonomous docking operations was addressed in Section 5.4. For this purpose, a representatively realistic docking scenario- applicable for docking with static, floating and moving stations- was defined and then the online solution generated by the IDVD method was synthesized and analyzed. The effectiveness, tractability, and robustness of IDVD method for all possible docking scenarios, including offline and online applications, were proven based on the results of computer simulations, the SITLSP, and Monte Carlo trials. The highlights with the results were the superior performance of the IDVD method in fast prototyping of feasible trajectories (in a real-time manner), assuring smooth controls changes over the time histories, and easily meeting endpoint constraints, without increasing optimization variables, which results in smooth and stable arrival of the AUV into the DS for all possible docking scenarios.

6.2 Conclusions

Following the objectives of this thesis mentioned in Chapter1, the TPBVP for autonomous underwater docking was formulated in Chapter 3, followed by Chapters 4 and 5 where, four different direct methods of calculus of variations were employed to numerically solve the trajectory optimization problem.

The principle advantage of the LPS, CPS, and hp-AR methods used in Chapter 4, was to generate docking trajectories holding the first-order optimality conditions. Thus, if the PS methods converge to a solution whose feasibility/optimality can be verified by post-optimality process, the proposed trajectory is very close to the true optimal solution. Nevertheless, reaching to this point by the PS methods was largely under mercy of having a proper set of initial guesses at hand. Many efforts were made to obtain a set of proper initial guesses for the PS methods in Chapter 4 to reach into the converged solutions. It was a cumbersome and tedious process until the IDVD method was employed to assure and speed up the convergence solutions of the PS methods. This approach was a new contribution that was not used before in the literature of PS methods. Comparison between the LPS, CPS, and hp-AR method performances showed that the LPS and CPS methods have some difficulties to fully transform the continuous co-states information into the discretized version; this affect the accuracy of their co-states and Hamiltonian trajectories in a numerical sense. In contrast, the hp-AR method showed a stable and numerically acceptable performance in holding the first-order optimality conditions regarding its rectangular and full rank differentiation matrix.

It was realized that the requirement of having a smooth arrival of the AUV into the DS, could not be achieved with the original 7-state formulation. This was due to the lack of inherent capability of the PS methods to set endpoint constraints on the controls, and that is a challenging issue in the literature of PS methods. Thus, this problem was addressed based on the novel 10-state AUV model in which three new states were added to the conventional docking TPBVP. By doing this, all three PS methods were able to meet the remaining conditions at the arrival time on the associated thrusters. However, satisfying this objective led to considerable degradation of computational efficiency.

To put it concisely, any PS method can address the main objective and almost all requirements of this thesis by offering optimal solutions whose resolution is verifiable by the post-optimality process and the SITLSP. This characteristic resulted in an approximately 54.4% saving in thrust energy consumption compared to the classic docking guidance methods in the literature [38-40]. This feature is exactly what is required for possible on-board implementation of the PS methods in docking operations in which the minimization of performance index is of great importance. However, because of the large number of nodes/mesh intervals, PS methods are unable to generate real-time solutions (even by using the hp-AR method and compiled IPOPT solvers in the 10-state formulation). Hence, their applications for AUV docking operations are limited to offline and open-loop configurations.

Conversely, the IDVD method discussed in Chapter 5, allows real-time computing of feasible solutions to be used online on-board AUV during docking operations. Compared to the PS methods generating optimal solutions, trajectories generated by the IDVD method are only sub-optimal. For instance, for the minimum-energy docking scenario used in this work the value of the cost function for the IDVD solution was about 5-10% worse that obtained using the PS methods. In addition, the IDVD method does not offer post-optimality facility like PS methods as it does not hold the first-optimality conditions. Nevertheless, all trajectories generated by the IDVD method were smooth, physically realizable and always assured satisfaction of endpoint constraints without the need to use the 10-state docking formulation. This was due to the fact that the IDVD formulation allowed to set the constraints on higher-order derivatives than the level of controls. As observed, it was even possible to enforce the jerk constraints whilst the nature of control vector was acceleration. This feature was not achievable with the PS methods. For instance to set the constraints on jerk profiles, the control vector should be at least based on jerk (or probably higher-order term derivatives), but not based on acceleration, that consequently results in adding an extra level of complexity. This salient feature in the IDVD structure was accompanied with a great computational efficiency, even by using the interpretive *fmincon* solver (see

Appendix C). This was a direct result of having only seven decision parameters as opposed to hundreds for the PS methods. It is noteworthy to mention that increasing the number of computational nodes (for example, from 50 nodes in the minimum-energy scenario to 70 nodes in the minimum-time scenario) has no impact on the number of decision parameters nor computational efficiency. More importantly, for the few number of decision parameters used in the IDVD method, a set of intuitive initial guesses is used which contributes to fast convergence of the optimization routine. This is completely in contrast with the results perceived in Chapter 4. The fast prototyping of feasible solutions using the IDVD method, make the realistic implementation of the closed-loop trajectory generation possible. By closed-loop implementation, the overall robustness of docking process is reinforced and docking in various operating environments with different types of stations such as static, floating, and moving becomes possible. Accordingly, the unique characteristics of the IDVD method support it to be a general-purpose candidate for the docking trajectory generation, and thus for the universal docking guidance framework.

6.3 Future Research Directions

Based on the contributions and key findings of this thesis, the following future research directions are suggested.

6.3.1 Incorporating Concept of Differential Flatness in the hp-AR-Based Guidance System

In Chapter 5, it was shown that employing the concept of differential flatness has considerable impact on reducing the dimension of optimization problems. Regarding the optimal and reliable performance of the hp-AR docking guidance system indicated in Chapter 4, it is of interest to develop a new hybrid guidance system employing differential flatness and the hp-AR structure. In this case, one still can use the spatial coordinates of the AUV as a flat output vector and use the Lagrange polynomials as basis functions.

6.3.2 Investigation of the Impact of Different NLP Solvers on Performance of the Guidance System

As seen in Chapters 4 and 5 of this thesis, *fmincon* and IPOPT solvers show roughly the same performance in satisfying tolerance accuracy of solutions but with different computational efficiency. In recent years, a wide variety of NLP solvers have been developed and used for trajectory optimization problems. It would be valuable to conduct a deep study on impact of appropriate NLP solvers such as SNOPT, SPRNLP, KNITRO, and IPOPT on both solution optimality and computational efficiency of the proposed guidance system. Particularly, it is interesting to extend the IDVD implementation based on the complied C++ IPOPT solver instead of *fmincon* interpretative MATLAB solver.

6.3.3 Investigation of the Online Guidance System Based on Different RHC Schemes

As mentioned in Chapter 2, RHC schemes have become popular and effective feedback strategies for nonlinear systems subjected to either state or control constraints. The online guidance system developed in this thesis was inspired explicitly from one of the RHC scheme. However, it would be valuable to conduct more research on different important aspects of diverse closed-loop configurations based on the RHC schemes such as stability, robustness, and computational efficiency for the underwater applications.

6.3.4 Hardware Implementation of the Guidance System on the Flinders' AUV

In this thesis, performance of the guidance system through the simulation and SITLSP results for the series of docking scenarios were validated. The next step is to concentrate on hardware implementation of the proposed guidance system on a real AUV, potentially the Flinders' AUV under development. For that purpose, the guidance system will be implemented in a C++ environment to be executed on an on-board embedded computer. This approach will definitely increase the computational efficiency of the proposed guidance system by several orders of magnitude. By doing this, more details and insight about the effectiveness of the proposed guidance system can be obtained. In a particular case, it is interesting to conduct hardware implementation and performance investigation of the closed-loop guidance system developed in this thesis for a docking operation with respect to a moving DS towed behind a mothership or an autonomous surface vessel. In fact, the IDVD-based online trajectory generator is efficient to continuously regenerate the trajectory based on the update of the moving station pose and finally capture the station in a minimum-time or with a minimum-energy maneuvering.

6.3.5 Hardware Implementation of the Guidance System on the Flinders' WAM-V Boat

The Center for Maritime Engineering, Control, and Imaging (CMECI) at Flinders University has developed an autonomous surface vessel based on the WAM-V platform, which is used for a series of tasks such as cost effective maritime monitoring and data collection, seabed mapping, and environmental surveying. Figure 6.1 shows the Flinders' WAM-V boat [156].

The universality of the guidance system developed in this thesis allows us to easily modify it based on the WAM-V boat model and the type of maritime missions. For example, it is desired to implement the proposed guidance system and investigate its effectiveness for the Maritime RobotX boat docking scenario, in the presence of wind and current disturbances, as illustrated in Figure 6.2.

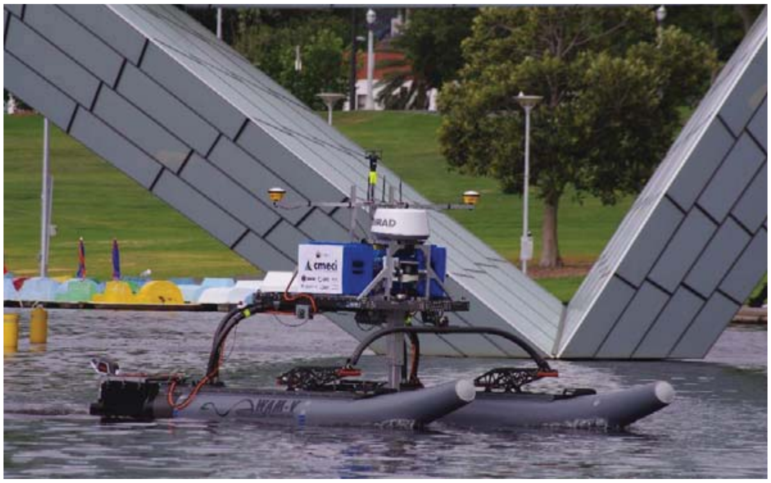


Figure 6.1 Flinders' WAM-V boat.

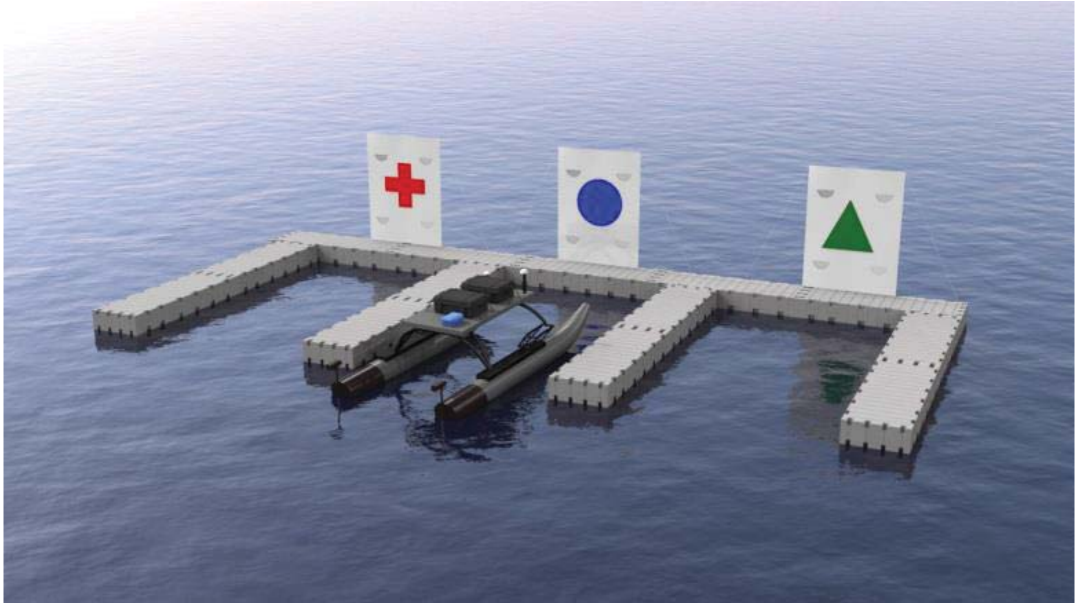
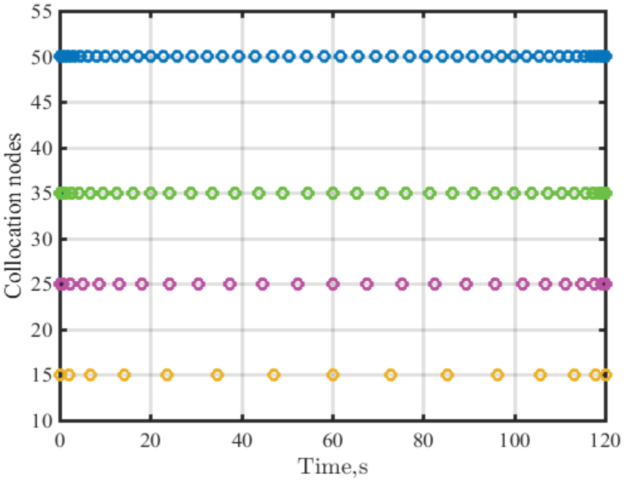


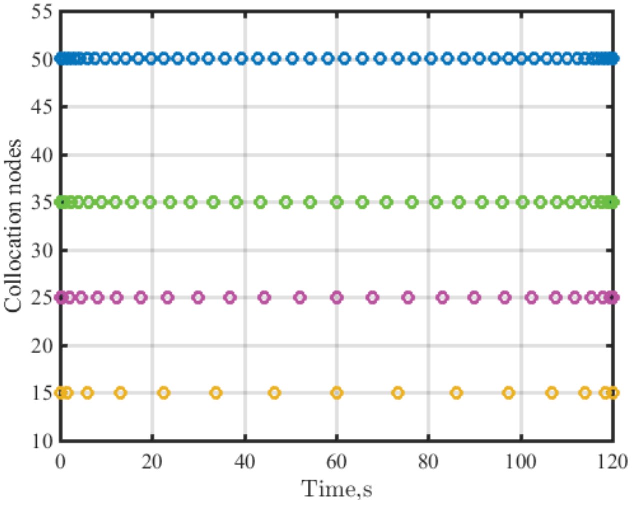
Figure 6.2 RobotX boat docking scenario (www.RobotX.org).

Appendix A

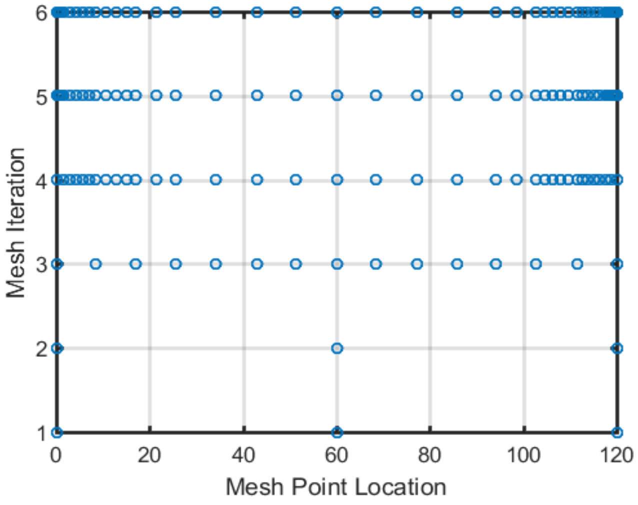
Simulation Results for the 7-State Formulation



(a)

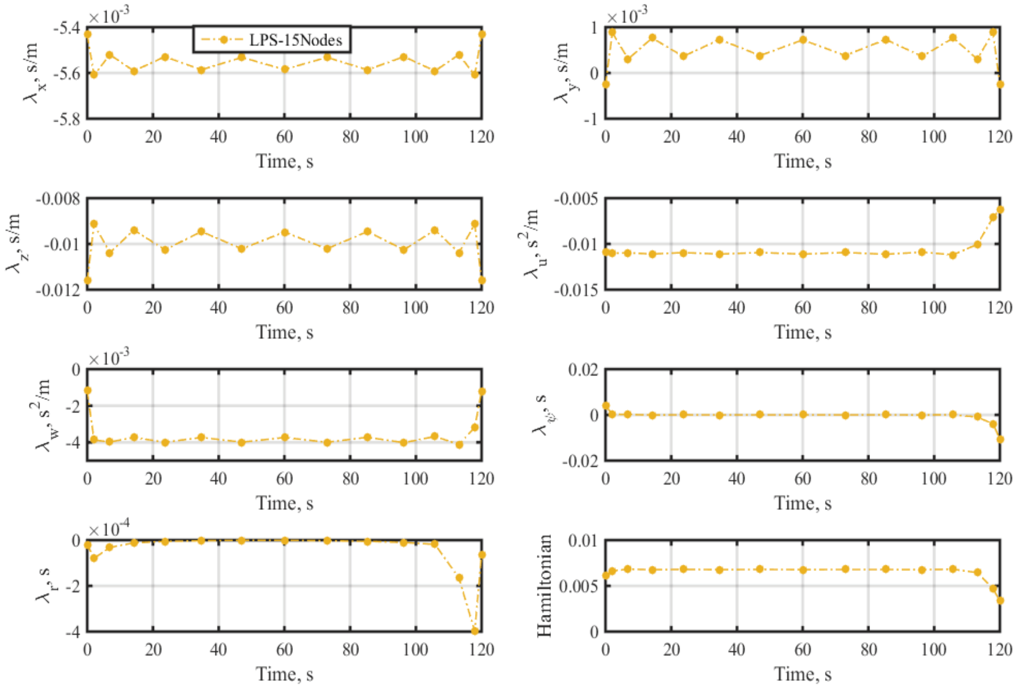


(b

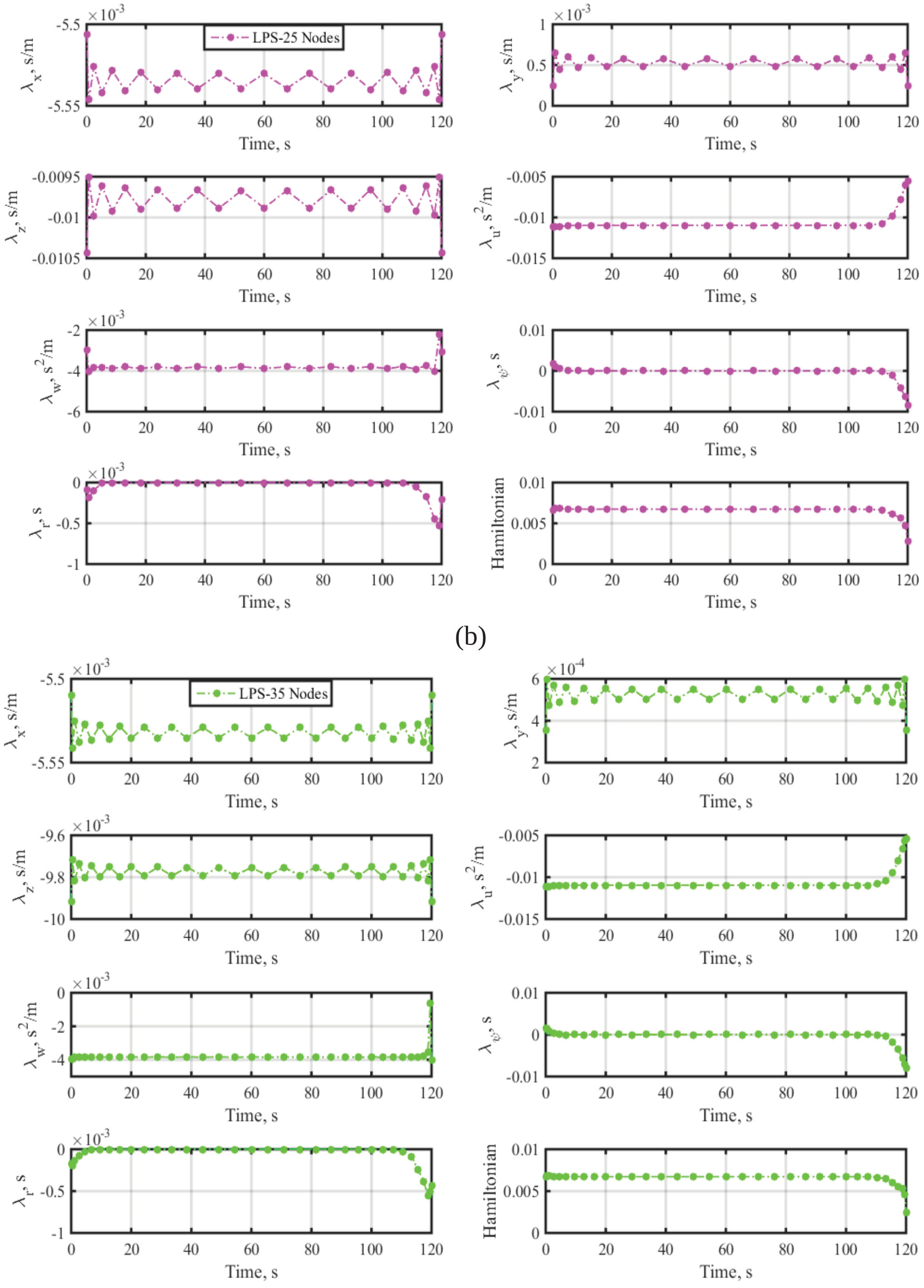


(c)

Figure 4.15 (a) Distribution of collocation points in LPS method; **(b)** distribution of collocation points in CPS method; **(c)** mesh refinement history in hp-AR method.



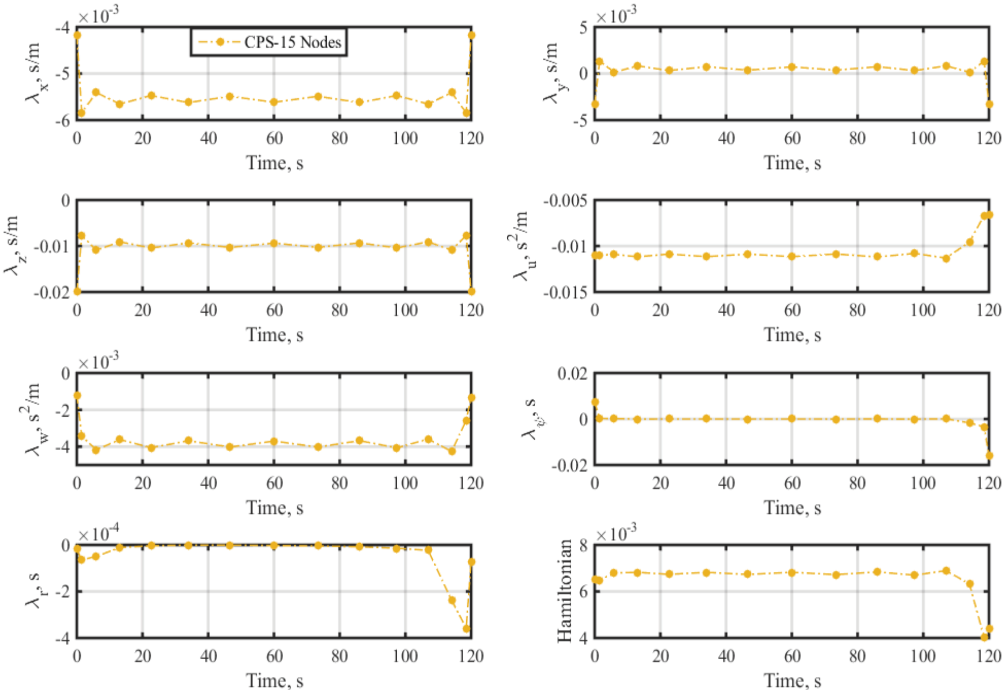
(a)



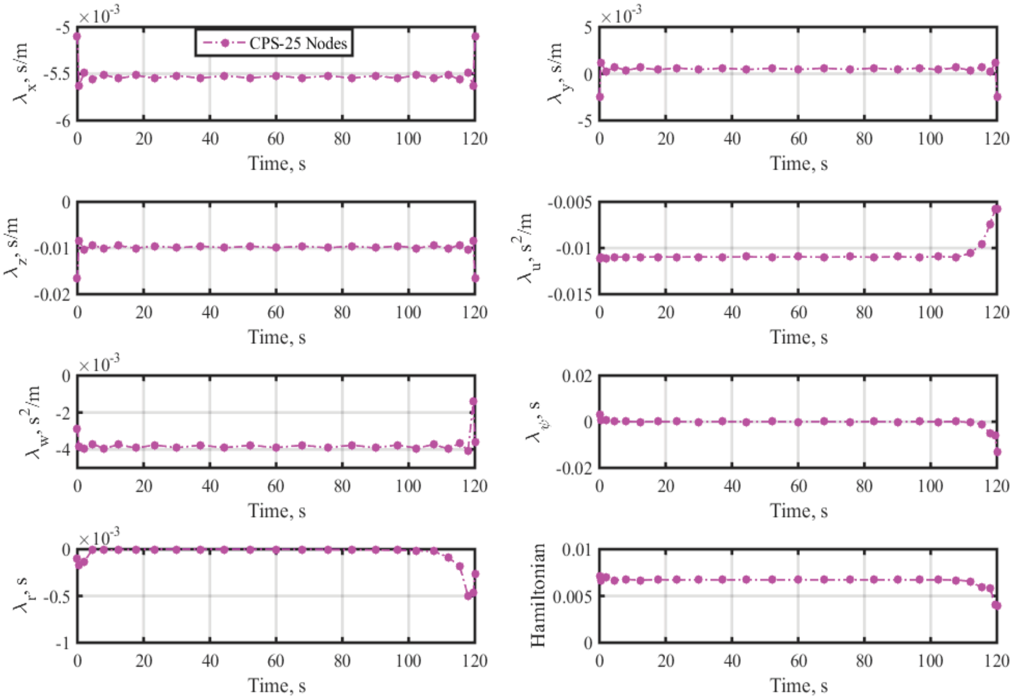
(b)

(c)

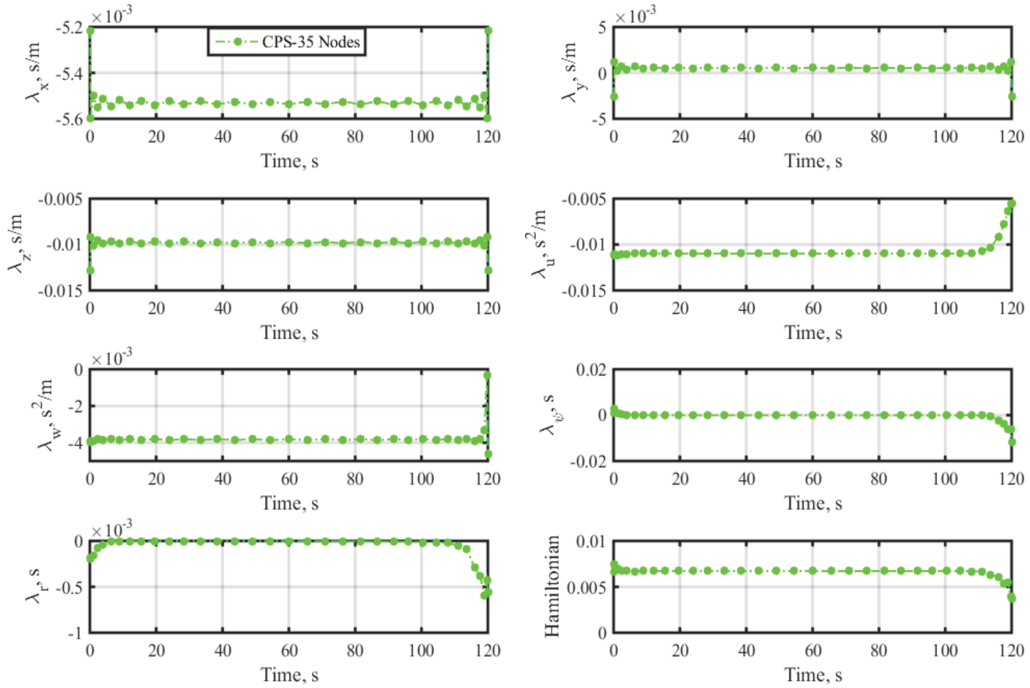
Figure 4.16 Evolution of co-states and Hamiltonian using the LPS method based 15-node configuration (a); 25-node configuration (b); 35-node configuration (c).


CPS-15 Nodes
×10^-3
λx, s/m
-4
-5
-6
λy, s/m
5
0
-5
λz, s/m
0
-0.01
-0.02
λu, s²/m
-0.005
-0.01
-0.015
×10^-3
λw, s²/m
0
-2
-4
λψ, s
0.02
0
-0.02
×10^-4
λr, s
0
-2
-4
×10^-3
Hamiltonian
8
6
4
0
20
40
60
80
100
120
Time, s

(a)


CPS-25 Nodes
×10^-3
λx, s/m
-5
-5.5
-6
λy, s/m
5
0
-5
λz, s/m
0
-0.01
-0.02
λu, s²/m
-0.005
-0.01
-0.015
×10^-3
λw, s²/m
0
-2
-4
λψ, s
0.02
0
-0.02
×10^-3
λr, s
0
-0.5
-1
Hamiltonian
0.01
0.005
0
0
20
40
60
80
100
120
Time, s

(b)



(c)

Figure 4.17 Evolution of co-states and Hamiltonian using the CPS method based 15-node configuration (a); 25-node configuration (b); 35-node configuration (c).

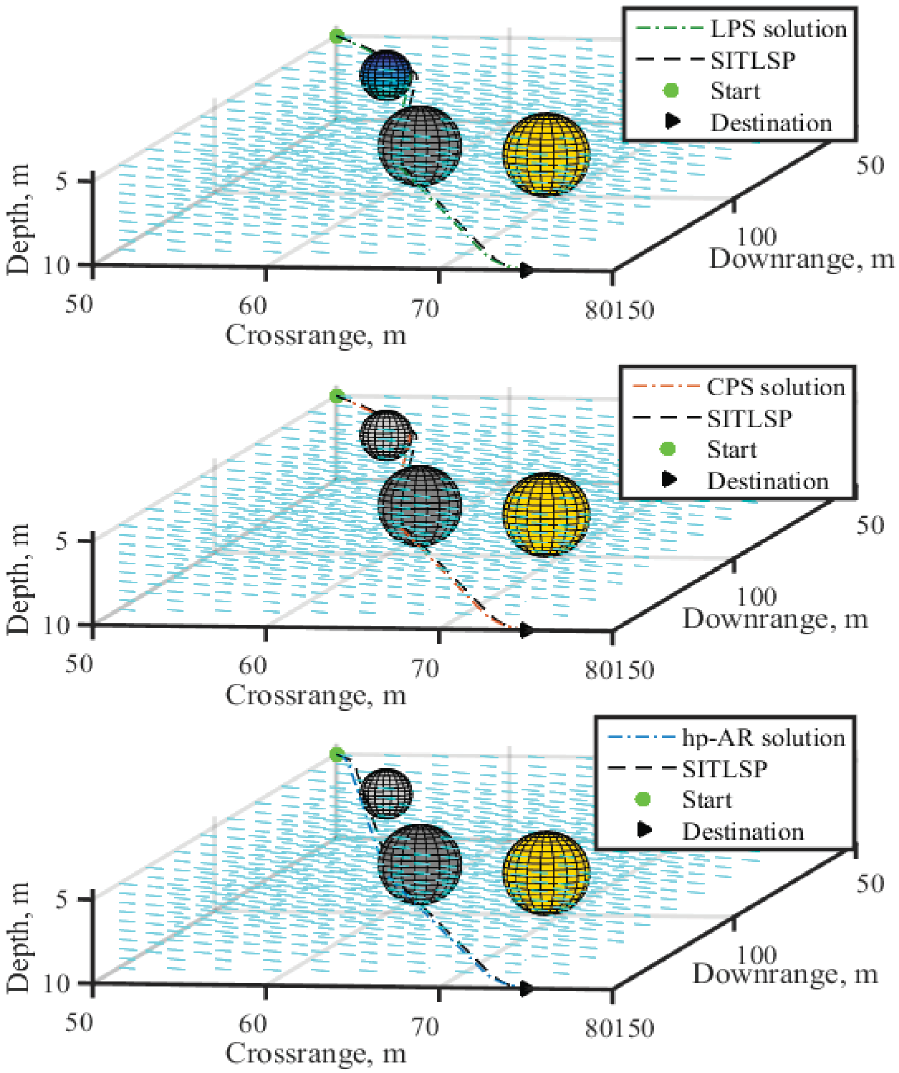


Figure 4.28 3D paths within the SITLSP considering NFZs.

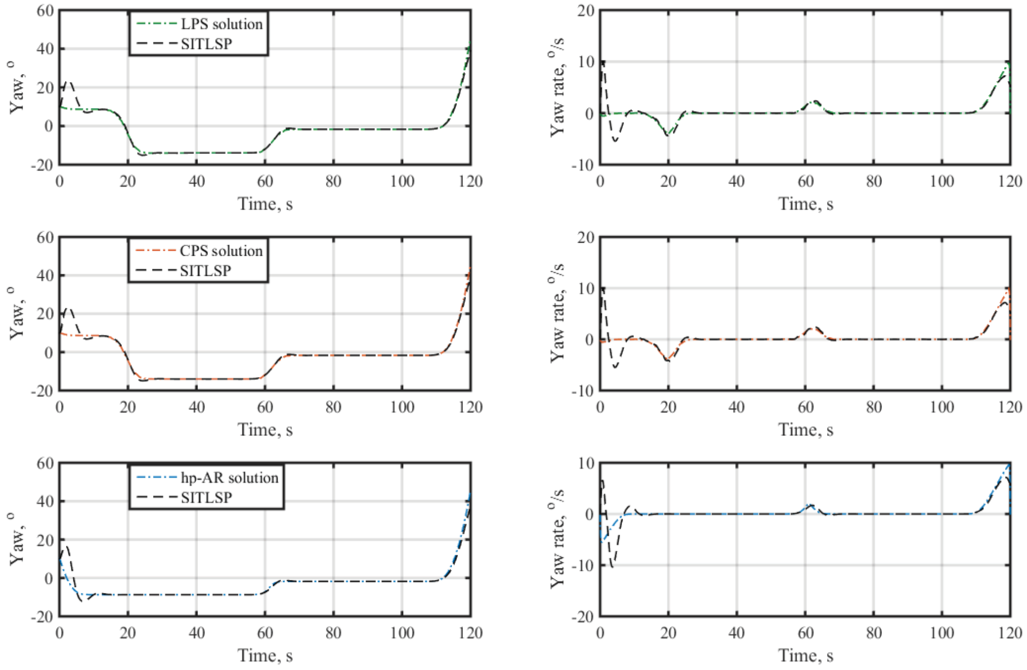


Figure 4.29 Yaw and yaw rate within the SITLSP considering NFZs.

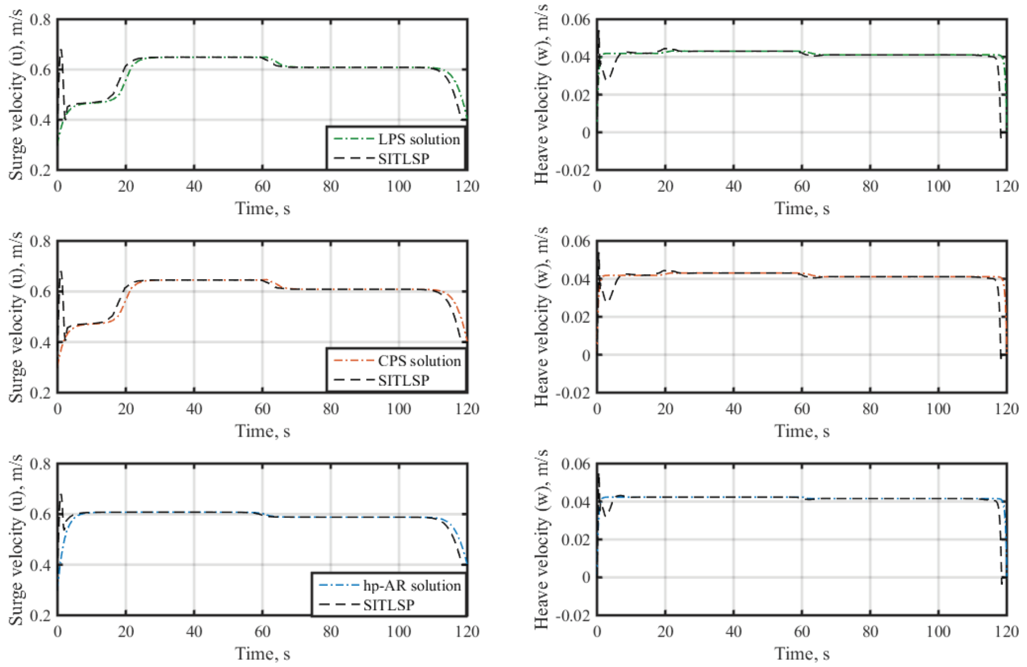


Figure 4.30 Surge and heave velocities within the SITLSP considering NFZs.

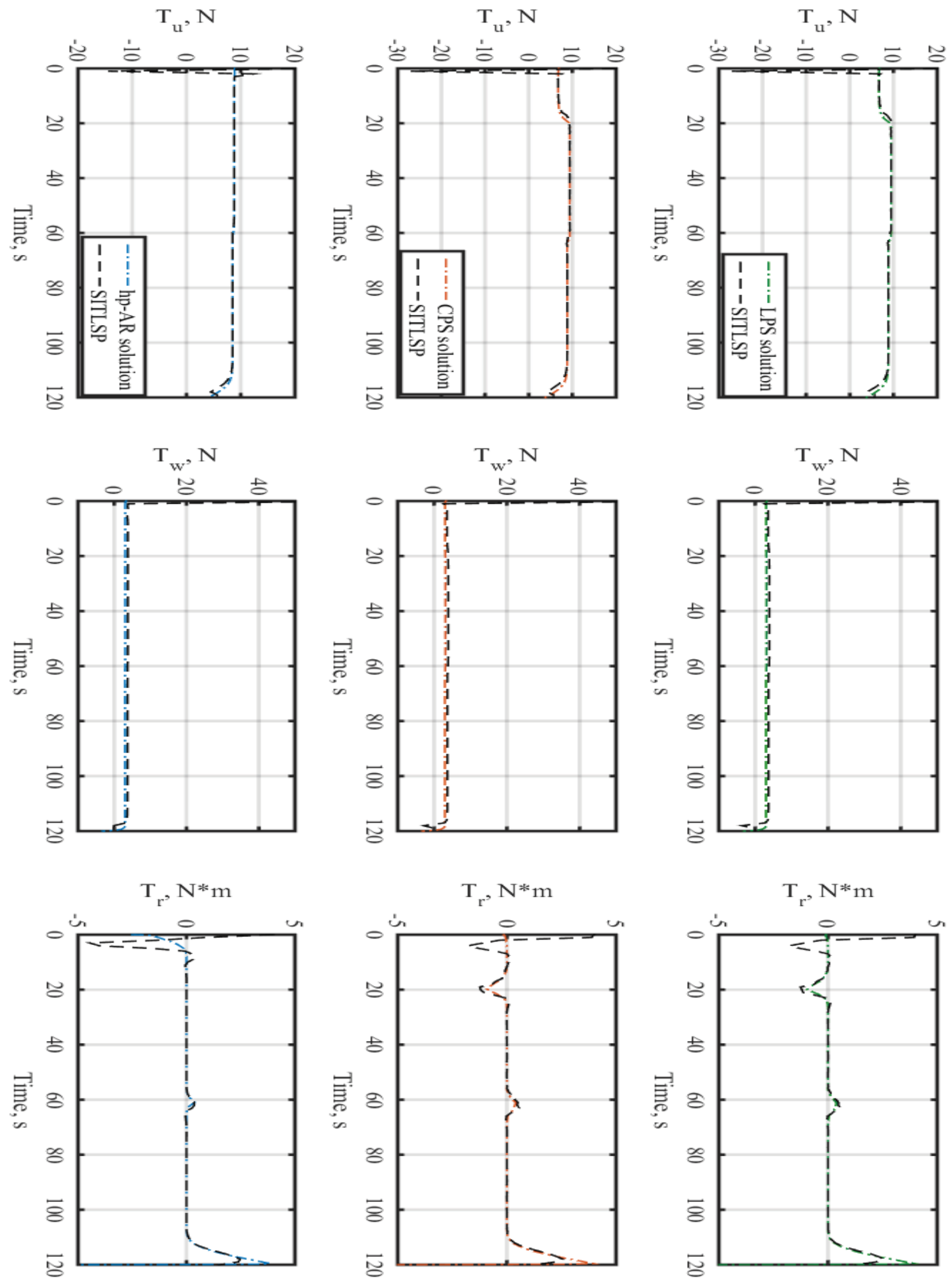


Figure 4.31 Controls within the SITLSP considering NFZs.

Appendix B

Simulation Results for the 10-State Formulation

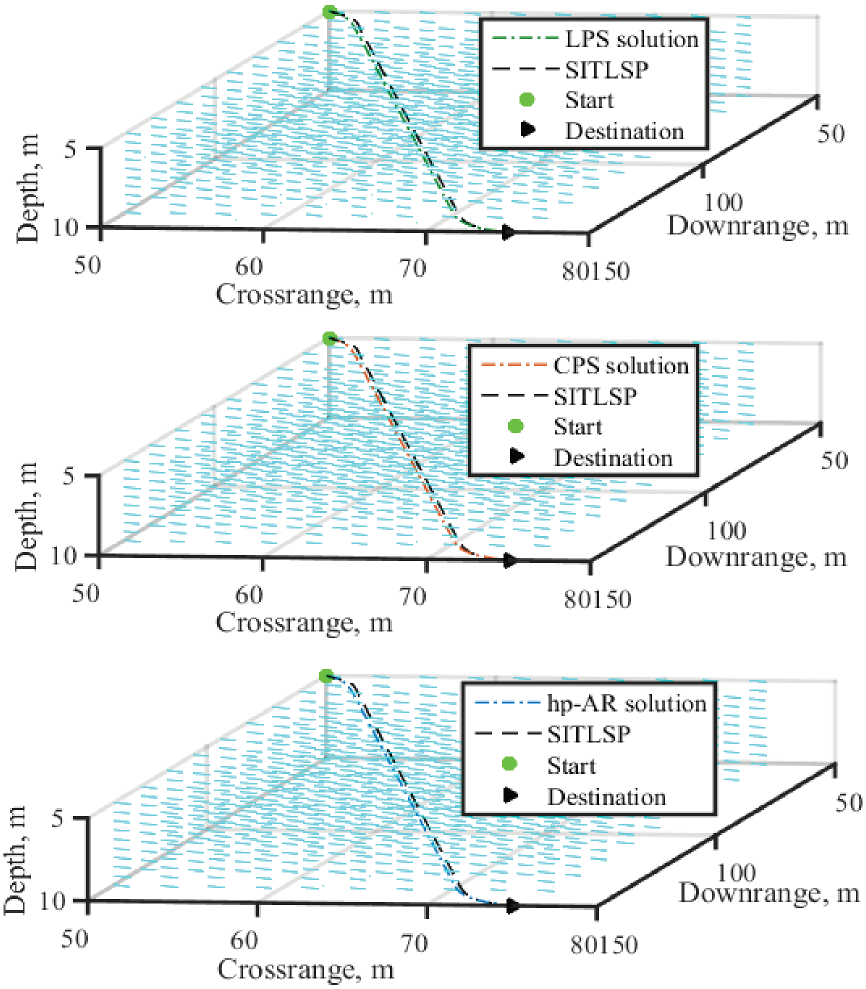


Figure 4.40 Realization of the 10-state based 3D paths within the SITLSP.

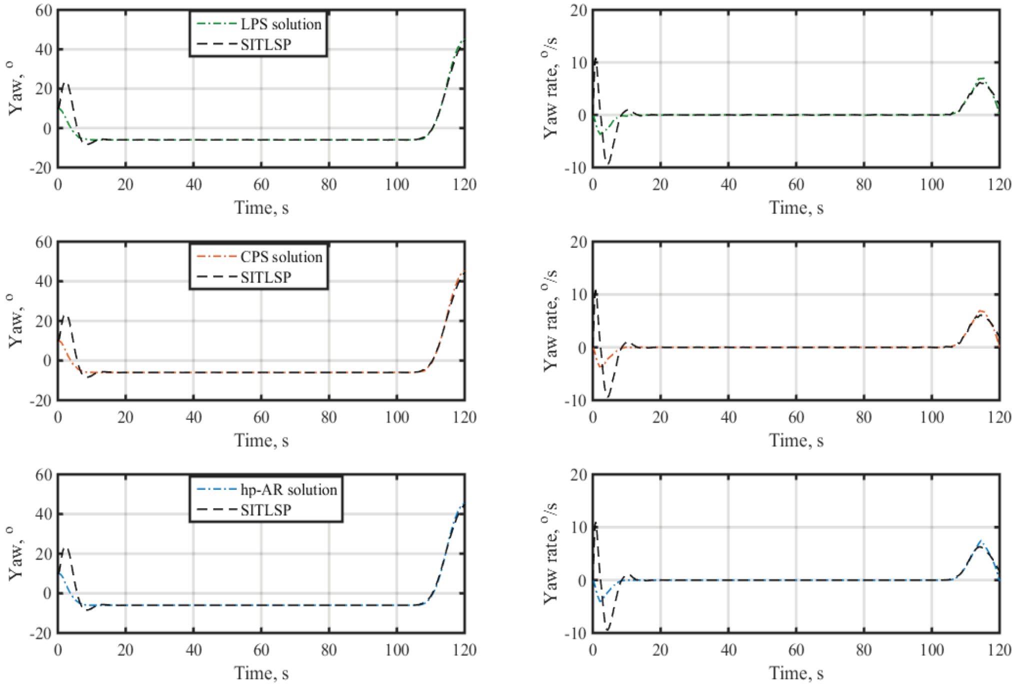


Figure 4.41 Realization of the 10-state based yaw and yaw rate solution within the SITLSP.

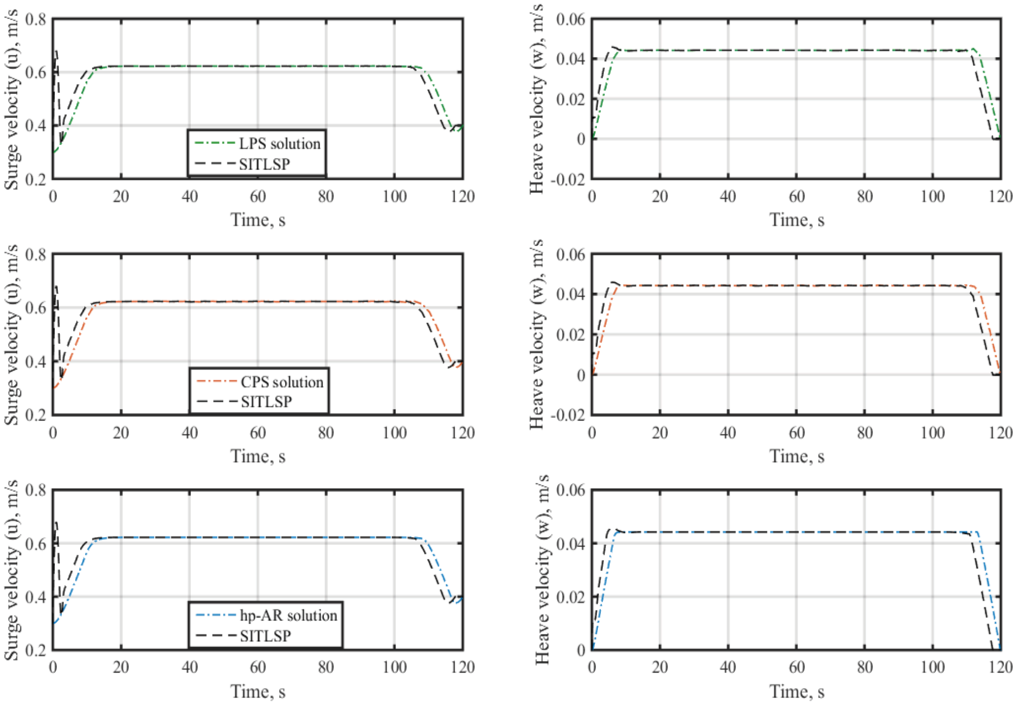


Figure 4.42 Realization of the 10-state based surge and heave velocities within the SITLSP.

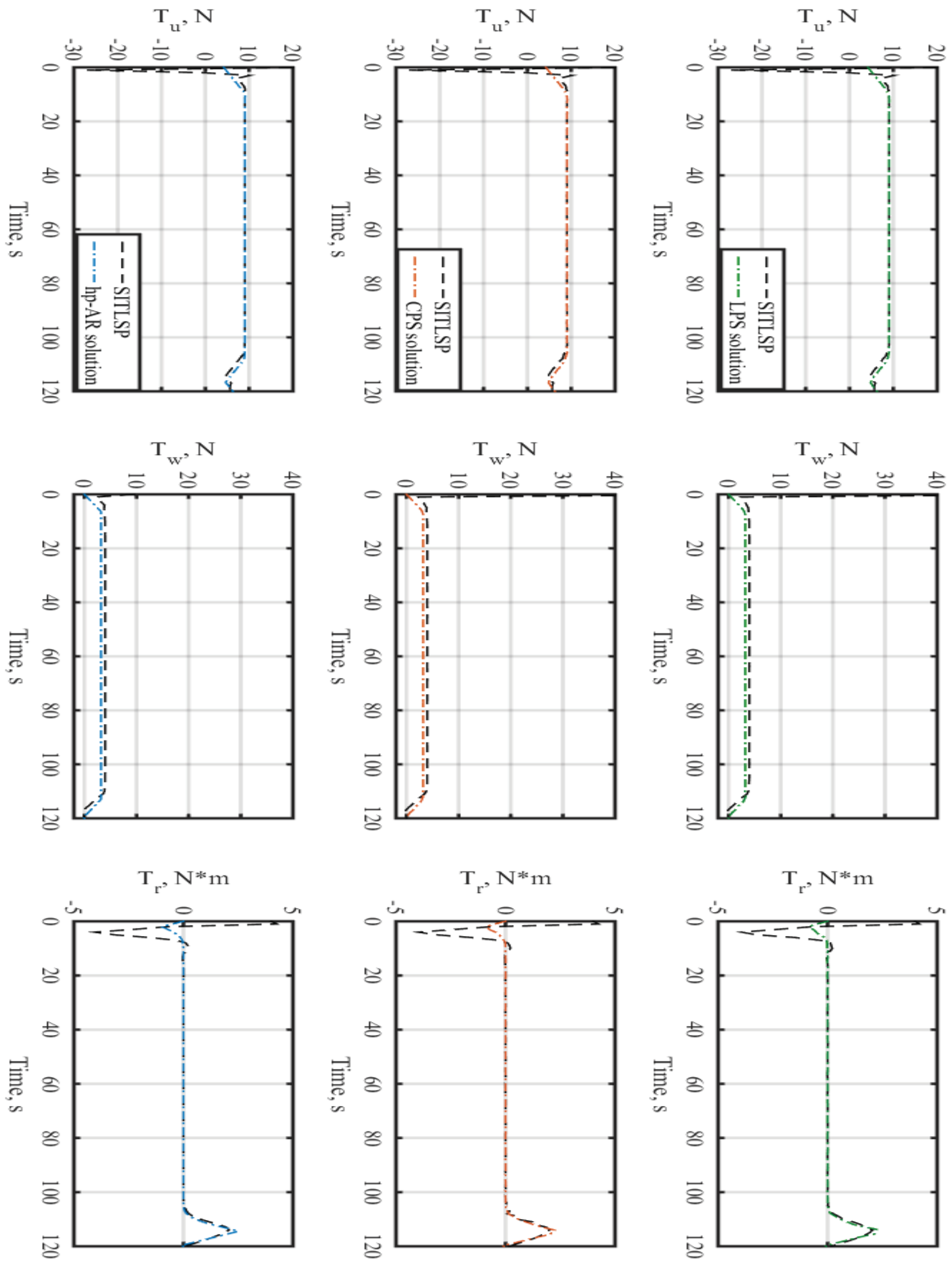


Figure 4.43 Realization of new states within the SITLSP.

Appendix C

Performance Comparison between PS Methods and IDVD Method

Table 6.1 Performance comparison between PS methods and IDVD method based on the minimum-energy docking scenario considering NFZs.

Method/Metric	*J*	t_{CPU}, *s*	*ES%*
LPS	0.2113	516.24	54.03
CPS	0.2111	499.34	54.05
hp-AR	0.2067	8.94*	54.53
IDVD	0.257	4.5	49.3

Bibliography

1. D. R. Blidberg, "The development of autonomous underwater vehicles (AUV); a brief summary," in *IEEE ICRA*, 2001.

2. S. Zhao and J. Yuh, "Experimental study on advanced underwater robot control," *IEEE transactions on robotics*, vol. 21, pp. 695-703, 2005.

3. Z. Zeng, L. Lian, K. Sammut, F. He, Y. Tang, and A. Lammas, "A survey on path planning for persistent autonomy of autonomous underwater vehicles," *Ocean Engineering*, vol. 110, pp. 303-313, 2015.

4. R. D. Christ and R. L. Wernli Sr, *The ROV manual: a user guide for remotely operated vehicles*: Butterworth-Heinemann, 2013.

5. J. Yuh, "Design and control of autonomous underwater robots: A survey," *Autonomous Robots*, vol. 8, pp. 7-24, 2000.

6. R. P. Stokey, A. Roup, C. von Alt, B. Allen, N. Forrester, T. Austin, *et al.*, "Development of the REMUS 600 autonomous underwater vehicle," in *OCEANS, 2005. Proceedings of MTS/IEEE*, 2005, pp. 1301-1304.

7. Bluefin-21ProductSheet-Bluefin Robotics,"http://www.bluefinrobotics.com/products/bluefin-21/".

8. P. E. Hagen, N. Storkersen, K. Vestgard, and P. Kartvedt, "The HUGIN 1000 autonomous underwater vehicle for military applications," in *OCEANS 2003 Proceedings*, 2003, pp. 1141-1145.

9. J. Ferguson and A. Pope, "Explorer-a modular AUV for commercial site survey," in *Underwater Technology, Proceedings of the 2000 International Symposium on*, 2000, pp. 129-132.

10. M. S. Stewart and J. Pavlos, "A means to networked persistent undersea surveillance," in *Submarine Technology Symposium, University of Washington, Tacoma, Wash*, 2006.

11. F. D. Maire, D. Prasser, M. Dunbabin, and M. Dawson, "A vision based target detection system for docking of an autonomous underwater vehicle," in *Proceedings of the 2009 Australasian Conference on Robotics and Automation*, 2009.

12. D. Pyle, R. Granger, B. Geoghegan, R. Lindman, and J. Smith, "Leveraging a large UUV platform with a docking station to enable forward basing and persistence for light weight AUVs," in *Oceans 2012*, 2012, pp. 1-8.

13. A. Murarka, G. Kuhlmann, S. Gulati, M. Sridharan, C. Flesher, and W. C. Stone, "Vision-based frozen surface egress: A docking algorithm for the ENDURANCE AUV," in *International Symposium on Unmanned Untethered Submersible Technology (UUST)*, 2009.

14. D. E. Frye, J. Kemp, W. Paul, and D. Peters, "Mooring developments for autonomous ocean-sampling networks," *IEEE journal of oceanic engineering*, vol. 26, pp. 477-486, 2001.

15. R. S. McEwen, B. W. Hobson, L. McBride, and J. G. Bellingham, "Docking control system for a 54-cm-diameter (21-in) AUV," *IEEE Journal of Oceanic Engineering*, vol. 33, pp. 550-562, 2008.

16. O. S. Board and N. R. Council, *Illuminating the hidden planet: The future of seafloor observatory science*: National Academies Press, 2000.

17. J. G. Bellingham, "Autonomous Underwater Vehicle Docking," in *Springer Handbook of Ocean Engineering*, Springer, 2016, pp. 387-406.

18. S. Cowen, S. Briest, and J. Dombrowski, "Underwater docking of autonomous undersea vehicles using optical terminal guidance," in *OCEANS'97. MTS/IEEE Conference Proceedings*, 1997, pp. 1143-1147.

19. R. Stokey, M. Purcell, N. Forrester, T. Austin, R. Goldsborough, B. Allen, *et al.*, "A docking system for REMUS, an autonomous underwater vehicle," in *OCEANS'97. MTS/IEEE Conference Proceedings*, 1997, pp. 1132-1136.

20. R. Stokey, B. Allen, T. Austin, R. Goldsborough, N. Forrester, M. Purcell, *et al.*, "Enabling technologies for REMUS docking: an integral component of an autonomous ocean-sampling network," *IEEE Journal of Oceanic Engineering,* vol. 26, pp. 487-497, 2001.

21. M. D. Feezor, F. Y. Sorrell, P. R. Blankinship, and J. G. Bellingham, "Autonomous underwater vehicle homing/docking via electromagnetic guidance," *IEEE Journal of Oceanic Engineering,* vol. 26, pp. 515-521, 2001.

22. B. Allen, T. Austin, N. Forrester, R. Goldsborough, A. Kukulya, G. Packard, *et al.*, "Autonomous docking demonstrations with enhanced REMUS technology," in *OCEANS 2006*, 2006, pp. 1-6.

23. J.-Y. Park, B.-h. Jun, P.-m. Lee, and J. Oh, "Experiments on vision guided docking of an autonomous underwater vehicle using one camera," *Ocean Engineering,* vol. 36, pp. 48-61, 2009.

24. J.-Y. Park, B.-H. Jun, P.-M. Lee, Y.-K. Lim, and J.-H. Oh, "Modified linear terminal guidance for docking and a time-varying ocean current observer," in *Underwater Technology (UT), 2011 IEEE Symposium on and 2011 Workshop on Scientific Use of Submarine Cables and Related Technologies (SSC)*, 2011, pp. 1-6.

25. H. Kondo, K. Okayama, J.-K. Choi, T. Hotta, M. Kondo, T. Okazaki, *et al.*, "Passive acoustic and optical guidance for underwater vehicles," in *OCEANS, 2012-Yeosu*, 2012, pp. 1-6.

26. F. Raspante, "Underwater mobile docking of autonomous underwater vehicles," in *OCEANS 2012 Conference*, 2012, pp. 1-15.

27. C. Yang, S. Peng, S. Fan, S. Zhang, P. Wang, and Y. Chen, "Study on docking guidance algorithm for hybrid underwater glider in currents," *Ocean Engineering,* vol. 125, pp. 170-181, 2016.

28. R. Coulson, J. Lambiotte, G. Grenon, T. Pantelakis, J. Curran, and A. An, "Development of a modular docking sub-system for 12" class autonomous

underwater vehicles," in *OCEANS'04. MTTS/IEEE TECHNO-OCEAN'04*, 2004, pp. 1745-1749.

29. J. Kim, "Dual control approach for automatic docking using monocular vision," Stanford University, 2007.

30. A. Brighenti, L. Zugno, F. Mattiuzzo, and A. Sperandio, "EURODOCKER-a universal docking-downloading recharging system for AUVs: conceptual design results," in *OCEANS'98 Conference Proceedings*, 1998, pp. 1463-1467.

31. http://www.thinkdefence.co.uk/2015/07/underwater-pit-stops/.

32. V. P. Shah, "Design considerations for engineering autonomous underwater vehicles," Massachusetts Institute of Technology, 2007.

33. B. W. Hobson, R. S. McEwen, J. Erickson, T. Hoover, L. McBride, F. Shane*, et al.*, "The development and ocean testing of an AUV docking station for a 21" AUV," in *OCEANS 2007*, 2007, pp. 1-6.

34. H. Singh, J. G. Bellingham, F. Hover, S. Lemer, B. A. Moran, K. Von der Heydt*, et al.*, "Docking for an autonomous ocean sampling network," *IEEE Journal of Oceanic Engineering,* vol. 26, pp. 498-514, 2001.

35. K. Vickery, "Acoustic positioning systems. A practical overview of current systems," in *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings of the 1998 Workshop on*, 1998, pp. 5-17.

36. G. M. de Goede and D. Norris, "Recovering unmanned undersea vehicles with a homing and docking sonar," in *OCEANS, 2005. Proceedings of MTS/IEEE*, 2005, pp. 2789-2794.

37. D. Stramski, E. Boss, D. Bogucki, and K. J. Voss, "The role of seawater constituents in light backscattering in the ocean," *Progress in Oceanography,* vol. 61, pp. 27-56, 2004.

38. P. A. Wilson, "Autonomous Homing and docking tasks for an underwater vehicle," IFAC Proceedings Volumes, vol. 42, pp. 304-309, 2009.

39. J. Y. Park, B. H. Jun, P. M. Lee, Y. K. Lim, and J. H. Oh, "Docking problem and guidance laws considering drift for an underactuated AUV," in OCEANS 2011 IEEE-Spain, 2011, pp. 1-7.

40. W. Naeem, R. Sutton, and S. Ahmad, "Pure pursuit guidance and model predictive control of an autonomous underwater vehicle for cable/pipeline tracking," in Proceedings-Institute of Marine Engineering Science and Technology Part C Journal of Marine Science and Environment, 2004, pp. 25-35.

41. Z. Zeng, A. Lammas, K. Sammut, F. He, and Y. Tang, "Shell space decomposition based path planning for AUVs operating in a variable environment," Ocean Engineering, vol. 91, pp. 181-195, 2014.

42. S. M. Zadeh, D. M. Powers, and K. Sammut, "An autonomous reactive architecture for efficient AUV mission time management in realistic dynamic ocean environment," Robotics and Autonomous Systems, vol. 87, pp. 81-103, 2017.

43. M. Chyba, T. Haberkorn, S. B. Singh, R. N. Smith, and S. K. Choi, "Increasing underwater vehicle autonomy by reducing energy consumption," Ocean Engineering, vol. 36, pp. 62-73, 2009.

44. M. B. Loc, H.-S. Choi, S.-S. You, and T. N. Huy, "Time optimal trajectory design for unmanned underwater vehicle," Ocean Engineering, vol. 89, pp. 69-81, 2014.

45. F. Maurelli, Y. Petillot, A. Mallios, S. Krupinski, R. Haraksim, and P. Sotiropoulos, "Investigation of portability of space docking techniques for autonomous underwater docking," in OCEANS 2009-EUROPE, 2009, pp. 1-9.

46. P. Jantapremjit and P. A. Wilson, "Optimal control and guidance for homing and docking tasks using an autonomous underwater vehicle," in Mechatronics and Automation, 2007. ICMA 2007. International Conference on, 2007, pp. 243-248.

47. P. Jantapremjit and P. Wilson, "Guidance-control based path following for homing and docking using an autonomous underwater vehicle," in Oceans 2008-MTS/IEEE Kobe Techno-Ocean, 2008, pp. 1-6.

48. J.-Y. Park, B.-H. Jun, K. Kim, P.-M. Lee, J.-H. Oh, and Y.-K. Lim, "Improvement of vision guided underwater docking for small AUV ISiMI," in OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges, 2009, pp. 1-5.

49. P. Batista, C. Silvestre, and P. Oliveira, "A two- step control approach for docking of autonomous underwater vehicles," International Journal of Robust and Nonlinear Control, vol. 25, pp. 1528-1547, 2015.

50. J. M. A. Braga, "Control of underwater vehicles on autonomous docking maneuvers," Porto, Portugal: Faculdade de Engenharia da Universidade do Porto, 2010.

51. B. M. Ferreira, A. C. Matos, N. A. Cruz, and A. P. Moreira, “Homing a robot with range-only measurements under unknown drifts,” Robotics and Autonomous Systems, vol. 67, pp. 3-13, 2015.

52. Y. Li, Y. Jiang, J. Cao, B. Wang, and Y. Li, “AUV docking experiments based on vision positioning using two cameras,” IEEE Journal of Oceanic Engineering, vol. 110, pp. 163-173, 2015.

53. A. Sans-Muntadas, K. Y. Pettersen, E. Brekke, and V. F. Henriksen, "A hybrid approach to underwater docking of AUVs with cross-current," in OCEANS 2016 MTS/IEEE Monterey, 2016, pp. 1-7.

54. I. Spangelo and O. Egeland, "Trajectory planning and collision avoidance for underwater vehicles using optimal control," Oceanic Engineering, IEEE Journal of, vol. 19, pp. 502-511, 1994.

55. R. P. Kumar, A. Dasgupta, and C. Kumar, "Real-time optimal motion planning for autonomous underwater vehicles," Ocean engineering, vol. 32, pp. 1431-1447, 2005.

56. M. Chyba, T. Haberkorn, R. N. Smith, and S. K. Choi, "Design and implementation of time efficient trajectories for autonomous underwater vehicles," Ocean Engineering, vol. 35, pp. 63-76, 2008.

57. J. T. Betts, “Survey of numerical methods for trajectory optimization,” Journal of guidance, control, and dynamics, vol. 21, pp. 193-207, 1998.

58. A. V. Rao, "Trajectory optimization: a survey," in Optimization and optimal control in automotive systems, Springer, 2014, pp. 3-21.

59. A. Rao, "Extension of the computational singular perturbation method to optimal control," PhD thesis, Princeton University, 1996.

60. A. V. Rao, "Application of a dichotomic basis method to performance optimization of supersonic aircraft," Journal of Guidance, Control, and Dynamics, vol. 23, pp. 570-573, 2000.

61. A. Rao and K. Mease, "Dichotomic basis approach to solving hyper-sensitive optimal control problems," Automatica, vol. 35, pp. 633-642, 1999.

62. J. Stoer and R. Bulirsch, Introduction to numerical analysis vol. 12: Springer Science & Business Media, 2013.

63. J. García-Heras, M. Soler, and F. J. Sáez, "A comparison of optimal control methods for minimum fuel cruise at constant altitude and course with fixed arrival time," Procedia Engineering, vol. 80, pp. 231-244, 2014.

64. G. Brauer, D. Cornick, and R. Stevenson, "Capabilities and applications of the Program to Optimize Simulated Trajectories (POST). Program summary document," 1977.

65. D. Meder and J. L. Searcy, "Generalized Trajectory Simulation (GTS), Volumes IV," The Aerospace Corporation. SAMSO-TR-75-255, 1975.

66. C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," Journal of Guidance, Control, and Dynamics, vol. 10, pp. 338-342, 1987.

67. P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," SIAM review, vol. 47, pp. 99-131, 2005.

68. http://tomopt.com/docs/tomlab/tomlab004.php.

69. R. H. Byrd, J. Nocedal, and R. A. Waltz, "KNITRO: An integrated package for nonlinear optimization," in Large-scale nonlinear optimization, ed: Springer, 2006, pp. 35-59.

70. A. Waechter, C. Laird, F. Margot, and Y. Kawajir, "Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT," Revision, 2009.

71. H. Mirinejad, "A radial basis function method for solving optimal control problems," University of Louisville, 2016.

72. M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," International journal of control, vol. 61, pp. 1327-1361, 1995.

73. M. B. Milam, K. Mushambi, and R. M. Murray, "A new computational approach to real-time trajectory generation for constrained mechanical systems," in Decision and Control, 2000. Proceedings of the 39th IEEE Conference on, 2000, pp. 845-851.

74. N. Faiz, S. Agrawal, and R. Murray, "Differentially flat systems with inequality constraints: An approach to real-time feasible trajectory generation," Journal of Guidance, Control, and Dynamics, vol. 24, pp. 219-227, 2001.

75. M. J. Van Nieuwstadt and R. M. Murray, "Real time trajectory generation for differentially flat systems," 1997.

76. M. van Nieuwstadt and R. M. Murray, "Outer flatness: Trajectory generation for a model helicopter," in Control Conference (ECC), 1997 European, 1997, pp. 325-330.

77. N. Petit, M. Milam, and R. Murray, "Constrained trajectory generation for micro-satellite formation flying," in AIAA Guidance, Navigation, and Control Conference and Exhibit, 2001, p. 4030.

78. M. E. Flores, "Real-time trajectory generation for constrained nonlinear dynamical systems using non-uniform rational B-spline basis functions," California Institute of Technology, 2007.

79. N. Petit, M. Milam, and R. Murray, "Constrained trajectory generation for micro-satellite formation flying," in AIAA Guidance, Navigation, and Control Conference and Exhibit, 2001, p. 4030.

80. M. E. Flores, "Real-time trajectory generation for constrained nonlinear dynamical systems using non-uniform rational B-spline basis functions," California Institute of Technology, 2007.

81. B. Charlet, J. Lévine, and R. Marino, "On dynamic feedback linearization," Systems & Control Letters, vol. 13, pp. 143-151, 1989.

82. S. Taamallah, X. Bombois, and P. M. Van den Hof, "Trajectory planning and trajectory tracking for a small-scale helicopter in autorotation," Control Engineering Practice, vol. 58, pp. 88-106, 2017.

83. F. Cazaurang, B. Bergeon, and L. L. Lavigne, "Modelling of longitudinal disturbed aircraft model by flatness approach," in AIAA Guidance, Navigation, and Control Conference and Exhibit, p. 5478.

84. W.-C. Lu, L. Duan, F.-B. Hsiao, and F. Mora-Camino, "Neural guidance control for aircraft based on differential flatness," Journal of guidance, control, and dynamics, vol. 31, pp. 892-898, 2008.

85. T. J. Koo and S. Sastry, "Differential flatness based full authority helicopter control design," in Decision and Control, 1999. Proceedings of the 38th IEEE Conference on, 1999, pp. 1982-1987.

86. A. Chamseddine, Y. Zhang, C. A. Rabbath, C. Join, and D. Theilliol, "Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle," IEEE Transactions on Aerospace and Electronic Systems, vol. 48, pp. 2832-2848, 2012.

87. I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "A prototype of an autonomous controller for a quadrotor UAV," in Control Conference (ECC), 2007 European, 2007, pp. 4001-4008.

88. R. M. Murray, "Optimization-based control, "California Institute of Technology,2009.

89. S. L. Fraga, J. B. Sousa, and F. L. Pereira, "Trajectory generation for a remotely operated vehicle," in European Control Conference (ECC), 2003, 2003, pp. 2899-2904.

90. O. A. Yakimenko and S. P. Kragelund, Real-time optimal guidance and obstacle avoidance for umvs: INTECH Open Access Publisher, 2011.

91. C. Canuto, M. Y. Hussaini, A. M. Quarteroni, and A. Thomas Jr, Spectral methods in fluid dynamics: Springer Science & Business Media, 2012.

92. C. G. Canuto, M. Y. Hussaini, A. M. Quarteroni, and T. A. Zang, Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics (Scientific Computation): Springer-Verlag New York, Inc., 2007.

93. J. Vlassenbroeck and R. Van Dooren, "A Chebyshev technique for solving nonlinear optimal control problems," IEEE transactions on automatic control, vol. 33, pp. 333-340, 1988.

94. J. Vlassenbroeck, "A Chebyshev polynomial method for optimal control with state constraints," Automatica, vol. 24, pp. 499-506, 1988.

95. P. Han, J. Shan, and X. Meng, "Re-entry trajectory optimization using an hp-adaptive Radau pseudospectral method," Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, pp. 1-15, 2012.

96. G. T. Huntington and A. V. Rao, "Optimal reconfiguration of spacecraft formations using the Gauss pseudospectral method," Journal of Guidance, Control, and Dynamics, vol. 31, pp. 689-698, 2008.

97. I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: from theory to flight," Annual Reviews in Control, vol. 36, pp. 182-197, 2012.

98. B. Wu, D. Wang, E. K. Poh, and G. Xu, "Nonlinear optimization of low-thrust trajectory for satellite formation: Legendre pseudospectral approach," Journal of Guidance, Control, and Dynamics, vol. 32, pp. 1371-1381, 2009.

99. J. Li, "Fuel-optimal low-thrust formation reconfiguration via Radau pseudospectral method," Advances in Space Research, vol. 58, pp. 1-16, 2016.

100. K. A. Baumgartner, S. Ferrari, and A. V. Rao, "Optimal control of an underwater sensor network for cooperative target tracking," IEEE Journal of Oceanic Engineering, vol. 34, pp. 678-697, 2009.

101. C. Francolin, A. Rao, C. Duarte, and G. Martel, "Optimal Control of an Autonomous Surface Vehicle to Improve Connectivity in an Underwater Vehicle Network," Journal of Aerospace Computing, Information, and Communication, vol. 9, pp. 1-13, 2012.

102. A. Adhami-Mirhosseini, A. P. Aguiar, and M. J. Yazdanpanah, "Seabed tracking of an autonomous underwater vehicle with nonlinear output regulation," in 2011 50th IEEE Conference on Decision and Control and European Control Conference, 2011, pp. 3928-3933.

103. Y. Zhuang, S. Sharma, B. Subudhi, H. Huang, and J. Wan, "Efficient collision-free path planning for autonomous underwater vehicles in dynamic environments with a hybrid optimization algorithm," Ocean Engineering, vol. 127, pp. 190-199, 2016.

104. G. T. Huntington, D. Benson, and A. V. Rao, "A comparison of accuracy and computational efficiency of three pseudospectral methods," in Proceedings of the AIAA Guidance, Navigation, and Control Conference, 2007, pp. 840-864.

105. C. L. Darby, W. W. Hager, and A. V. Rao, "A preliminary analysis of a variable-order approach to solving optimal control problems using pseudospectral methods," in AIAA/AAS astrodynamics specialist conference, 2010, pp. 2-5.

106. C. L. Darby, W. W. Hager, and A. V. Rao, "Direct trajectory optimization using a variable low-order adaptive pseudospectral method," Journal of Spacecraft and Rockets, vol. 48, pp. 433-445, 2011.

107. C. L. Darby, W. W. Hager, and A. V. Rao, "An hp- adaptive pseudospectral method for solving optimal control problems," Optimal Control Applications and Methods, vol. 32, pp. 476-502, 2011.

108. F. Liu, W. W. Hager, and A. V. Rao, "Adaptive mesh refinement method for optimal control using nonsmoothness detection and mesh size reduction," Journal of the Franklin Institute, vol. 352, pp. 4081- 4106, 2015.

109. V. Taranenko, "Experience of Employment the Ritz's, Poincare's, and Lyapunov's Methods for Solving the Problems of Flight Dynamics," Air Force Engineering Academy na Prof. N. Zhukovskiy Press, Moscow, 1968.

110. V. Taranenko and V. Momdzhi, "Direct Variational Method in Boundary Problems of Flight Dynamics," ed: Mashinostroenie Press, Moscow (1968, in Russian), 1968.

111. O. A. Yakimenko, "Direct method for rapid prototyping of near-optimal aircraft trajectories," Journal of Guidance, Control, and Dynamics, vol. 23, pp. 865-875, 2000.

112. O. A. Yakimenko, D. P. Horner, and D. G. Pratt, "AUV rendezvous trajectories generation for underwater recovery," in Control and Automation, 2008 16th Mediterranean Conference on, 2008, pp. 1192-1197.

113. I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "Direct method based control system for an autonomous quadrotor," Journal of Intelligent & Robotic Systems, vol. 60, pp. 285-316, 2010.

114. M. Wilde, M. Ciarcià, A. Grompone, and M. Romano, "Experimental Characterization of Inverse Dynamics Guidance in Docking with a Rotating Target," Journal of Guidance, Control, and Dynamics, pp. 1-15, 2016.

115. Y. Zhang, J. Chen, and L. Shen, "Real-time trajectory planning for UCAV air-to-surface attack using inverse dynamics optimization method and receding horizon control," Chinese Journal of Aeronautics, vol. 26, pp. 1038-1056, 2013.

116. N. Slegers and O. Yakimenko, "Optimal control for terminal guidance of autonomous parafoils," in 20th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, 2009, p. 2958.

117. O. Yakimenko, Y. Xu, and G. Basset, "Computing short-time aircraft maneuvers using direct methods," in AIAA Guidance, Navigation and Control Conference and Exhibit, 2008, p. 6632.

118. D. Jung and P. Tsiotras, "Multiresolution on-line path planning for small unmanned aerial vehicles," in American Control Conference, 2008, 2008, pp. 2744-2749.

119. R. V. Cowlagi and P. Tsiotras, "Multiresolution path planning with wavelets: A local replanning approach," in American Control Conference, 2008, 2008, pp. 1220-1225.

120. M. Wzorek and P. Doherty, "Reconfigurable path planning for an autonomous unmanned aerial vehicle," in Hybrid Information Technology, 2006. ICHIT'06. International Conference on, 2006, pp. 242-249.

121. M. Wzorek, J. Kvarnström, and P. Doherty, "Choosing Path Replanning Strategies for Unmanned Aircraft Systems," in ICAPS, 2010, pp. 193-200.

122. Z. Zeng, K. Sammut, A. Lammas, F. He, and Y. Tang, "Efficient path re-planning for AUVs operating in spatiotemporal currents," Journal of Intelligent & Robotic Systems, vol. 79, p. 135, 2015.

123. R. Jamilnia and A. Naghash, “Optimal guidance based on receding horizon control and online trajectory optimization,” Journal of Aerospace Engineering, vol. 26, pp. 786-793, 2011.

124. S. S. Farahani, I. Papusha, C. McGhan, and R. M. Murray, "Constrained autonomous satellite docking via differential flatness and model predictive control," in Decision and Control (CDC), 2016 IEEE 55th Conference on, 2016, pp. 3306-3311.

125. E. F. Camacho and C. B. Alba, Model predictive control: Springer Science & Business Media, 2013.

126. C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practice—a survey," Automatica, vol. 25, pp. 335-348, 1989.

127. D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," Automatica, vol. 36, pp. 789-814, 2000.

128. M. Morari and J. H. Lee, "Model predictive control: past, present and future," Computers & Chemical Engineering, vol. 23, pp. 667-682, 1999.

129. A. Jadbabaie, "Nonlinear receding horizon control: a control lyapunov function approach," PhD thesis, California Institute of Technology, Control and Dynamical Systems, 2001.

130. S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," Control engineering practice, vol. 11, pp. 733-764, 2003.

131. N. DUNBAR and R. FRANZ, "Online control customization via optimization, based control," Software-Enabled Control Inf. Technol. Dyn. Syst, p. 149, 2003.

132. T. Keviczky and G. J. Balas, "Receding horizon control of an F-16 aircraft: A comparative study," Control Engineering Practice, vol. 14, pp. 1023-1033, 2006.

133. R. Bhattacharya, G. J. Balas, M. A. Kaya, and A. Packard, "Nonlinear receding horizon control of an F-16 aircraft," Journal of Guidance, Control, and Dynamics, vol. 25, pp. 924-931, 2002.

134. T. I. Fossen, Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles, 2002.

135. L. A. Gonzalez, Design, modelling and control of an autonomous underwater vehicle, Bachelor of Engineering Honours thesis, The University of Western Australia, 2004.

136. T. T. J. Prestero, Verification of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle, Master of Science thesis, Massachusetts institute of technology, 2001.

137. J.-Y. Park, Docking problem and guidance laws considering drift for an underactuated autonomous AUV, Ph.D. dissertation, Korea Advanced Institute of Science and Technology, 2011.

138. D. S. Naidu, Optimal control systems, CRC press, 2002.

139. G. A. Boyarko, "Spacecraft guidance strategies for proximity maneuvering and close approach with a tumbling object," 2010.

140. A. Lammas, F. He, and K. Sammut, 6-DoF navigation systems for autonomous underwater vehicles, Mobile Robots Navigation, InTech, 2010.

141. M. Kokegei, F. He, and K. Sammut, Fully coupled 6 degree-of-freedom control of an over-actuated autonomous underwater vehicle, Autonomous Underwater Vehicles, InTech, 2011.

142. E. D. B. Medagoda, and P. W. Gibbens, Synthetic-waypoint guidance algorithm for following a desired flight trajectory, Journal of Guidance, Control, and Dynamics, 33(2) , pp.601-606, 2010.

143. D. Garg, W. W. Hager, and A. V. Rao, "Pseudospectral methods for solving infinite-horizon optimal control problems," Automatica, vol. 47, pp. 829-837, 2011.

144. B. Fornberg and D. M. Sloan, "A review of pseudospectral methods for solving partial differential equations," Acta numerica, vol. 3, pp. 203-267, 1994.

145. D. Gottlieb and S. A. Orszag, Numerical analysis of spectral methods: theory and applications: SIAM, 1977.

146. Q. Gong, W. Kang, N. S. Bedrossian, F. Fahroo, P. Sekhavat, and K. Bollino, "Pseudospectral optimal control for military and industrial applications," in Decision and Control, 2007 46th IEEE Conference on, 2007, pp. 4128-4142.

147. F. Fahroo and I. M. Ross, "Direct trajectory optimization by a Chebyshev pseudospectral method," Journal of Guidance, Control, and Dynamics, vol. 25, pp. 160-166, 2002.

148. F. Fahroo and I. M. Ross, “Pseudospectral methods for infinite-horizon nonlinear optimal control problems,” Journal of Guidance, Control, and Dynamics, vol. 31, pp. 927-936, 2008.

149. M. A. Patterson and A. V. Rao, "GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming," ACM Transactions on Mathematical Software (TOMS), vol. 41, p. 1, 2014.

150. I. M. Ross and F. Fahroo, “Pseudospectral knotting methods for solving nonsmooth optimal control problems,” Journal of Guidance, Control, and Dynamics, vol. 27, pp. 397-405, 2004.

151. F. Fahroo and I. M. Ross, “Co-state Estimation by a Legendre Pseudospectral Method,” Journal of Guidance, Control, and Dynamics, vol. 24, pp. 270-277, 2001.

152. Q. Gong, I. M. Ross, and F. Fahroo, "A Chebyshev pseudospectral method for nonlinear constrained optimal control problems," in Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, 2009, pp. 5057-5062.

153. D. Garg, M. A. Patterson, C. Francolin, C. L. Darby, G. T. Huntington, W. W. Hager, et al., "Direct trajectory optimization and co-state estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method," Computational Optimization and Applications, vol. 49, pp. 335-358, 2011.

154. D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," Automatica, vol. 46, pp. 1843-1851, 2010.

155. D. Pearson, E. An, M. Dhanak, K. von Ellenrieder, and P. Beaujean, High-level fuzzy logic guidance system for an unmanned surface vehicle (USV) tasked to perform autonomous launch and recovery (ALR) of an autonomous underwater vehicle (AUV): IEEE, 2014.

156. http://csem.flinders.edu.au/competitions/maritime_robotx/.