A Study on Adaptive Locomotion Control of Underwater Snake Robot in Open, Cluttered, and Confined Environments

September 2024

QIU YIPING

A Study on Adaptive Locomotion Control of Underwater Snake Robot in Open, Cluttered, and Confined Environments

Graduate School of Science and Technology
Degree Programs in Systems and Information Engineering
University of Tsukuba

September 2024

QIU YIPING

Contents

List of Figures

List of Tables

Abstract

The advancement of undersea technology is becoming increasingly important, driven by the necessity for efficient exploration and maintenance in marine environments. Conventional snake robots, which mimic the movement of actual snakes, have limited flexibility to changing underwater conditions due to their reliance on predefined movement patterns when employing a Proportional–Integral–Derivative controller. To increase the flexibility, Model Predictive Control is implemented, so that the robot can change its locomotion without a predetermined reference. However, as the conventional gradient-based Model Predictive Control cannot handle nonlinear dynamics, it would face difficulties in managing collisions, especially faced with complicated underwater terrains. Furthermore, this typical Model Predictive Control approach may result in high computational costs when utilizing a complex dynamic model, which negatively affects real-time application.

In order to address these concerns, it is suggested employing a sample-based approach named Monte Carlo Model Predictive Control, which involves running forward simulations using randomly sampled input sequences. Nonlinear dynamics can be directly incorporated into the system model, so that the robot is able to effectively manage collisions. However, Monte Carlo Model Predictive Control is computationally demanding, particularly when dealing with multiple control inputs which requires a large number of samples, such as managing each joint of a snake robot. In order to deal with this issue, it is proposed that Curved Derivative Control is implemented in which the other parts of the snake robot simply follow the locomotion of the head.

This effectively reduces computational costs while improving the flexibility of snake locomotion.

This solution incorporates the interaction between the snake robot and its environment directly into the system dynamics, allowing the robot to easily overcome barriers in cluttered and confined environments. The future study will focus on the manipulation of independent sections of a snake body in order to fulfill multiple functions concurrently. To be specific, Monte Carlo Model Predictive Control can be used to create control inputs for the head and middle joints of a robot. This allows one part of the robot's body to grab items while the other section makes it forward. More importantly, it is planned that this approach can be executed on a physical underwater snake robot to verify its effectiveness in practical situations.

By demonstrating the efficacy of Monte Carlo Model Predictive Control combined with Curved Derivative Control in managing nonlinear dynamics and reducing computational costs, our approach provides a generalized framework applicable to other robotic systems operating in dynamic and unstructured environments. This research paves the way for advancements in autonomous underwater vehicles, industrial inspection robots, and search-and-rescue operations, offering a robust solution for real-time, adaptive control in complex settings.

Acknowledgments

I have faced many challenges in my life so far, but completing my PhD is considered one of the most challenging tasks. Especially the COVID-19 coronavirus that has been spreading since 2021. The severity of the virus has led to a drastic change in my daily life, and the lack of communication with others, living alone at home all day and night, has led to mental stress, and I have thought of giving up this journey in a foreign country at the most confusing moments. However, because of the help of many people around me, I was able to complete this study, and I would like to take this opportunity to express my deepest thanks to them.

First of all, I would like to thank my supervisors Associate Professor Hisashi Date, Professor Takashi Tsubouchifor and Professor Mochiyama Hiromi for the care and help with my research. Especially, I would like to thank my main supervisor Associate Professor Hisashi Date for his unwavering support, insightful guidance, and continuous encouragement. His expertise and constructive feedback have been invaluable throughout my research. I am also deeply grateful to my supervisor, Professor Takashi Tsubouchi, for his profound knowledge and guidance on my research. The feedback of him on my presentations of Yamabico Symposium during the whole PhD course helps me improve my presentation and accomplish my study. I also want to thank Professor Mochiyama Hiromi for his commends on the preliminary examination which help me improve my thesis and defense.

My sincere thanks go to Professor Akihisa Ohya, Assistant Professor Ayanori Yorozu and the members of the Intelligent Robot Laboratory, for their technical assistance and intellectual discussions. During the past few years in this lab, I have met many good friends from various countries. The Japanese students are very kind to me even I'm not good at speaking Japanese. They always invite me for lunch outside and chat with me so that I'm not alone in the lab. I also want to express a special thank to my foreign friends, who have been a source of constant support and encouragement.

I would like to extend my deepest gratitude to my best friends: SDW,YCJ,ZCZ. Your unwavering support, encouragement, and friendship have been an invaluable source of strength throughout my life.

I would also like to thank my parents and twin brother for their care and attention for more than 20 years, their unrequited contribution and giving, their encouragement when I encountered difficulties, and their courage to go forward. My family is my cozy harbour forever.

Last but not least, I would like to express my gratitude to my girlfriend, our story began when I first came to Japan. You accompanied me through the loneliness of the tough time. Whenever I feel like giving up, your love keeps me going. I would rather share one lifetime with you than face all the ages of this world alone.

Chapter 1

Introduction

1.1 Background and Motivation

Scientists have derived inspiration from the natural world and wild animals for decades. To address the complexities of robot operating environments, variable structures based on various types of terrains have been proposed, including: Designed to replicate the locomotion of a salamander, Pleurobot [1], The Hexapod Robot, as described in [2] consists of a six-legged structure resembling an arachnid and features enhanced balance capabilities. On the other hand, Boston Dynamics' Spot [3] is a four-legged robotic platform in the form of an Atlas and Wildcat that is capable of performing autonomous tasks. While these robots exhibit greater adaptability to expansive terrains compared to wheeled robots, their capabilities are still constrained by the environments in which they operate. Due to their numerous degrees of freedom (DOFs), snakes are capable of generating a variety of locomotion patterns that are adapted to their environment, which includes rugged mountains, deserts, trees, and the ocean. More emphasis is placed by researchers on the advancement of snake robots designed to function in confined and intricate environments [4]. Gray introduced the first study of snake locomotion in [5], and Hirose developed the first prototype of a snake robot in [6].

Scientists have increased their focus on underwater exploration, rescue, and other types of missions in recent years. This study examines a hypothetical situation in which the robot is tasked with conducting a search mission within a sunken ship. This mission entails navigating to the ship's location and entering through narrow apertures, including ruptured pipelines and apertures. For underwater missions involving such conditions, bio-inspired robotic systems that emulate the locomotion of fish and reptiles would be viable alternatives. The underwater snake robot exhibits greater versatility, access capability, and intervention capability in comparison to conventional underwater robots such as remotely operated underwater vehicles (ROVs) and autonomous underwater vehicles (AUVs) [7, 8, 9, 10, 11]. Torpedo-shaped, glider-like, and hovering are all characteristics of AUVs, which can range in size from human-portable to hundreds of tons [12]. AUV's energy efficiency, facilitated by its streamlined design, renders it well-suited for extended functional periods. Nonetheless, the fixed body structure prevents the AUV from interacting with its surroundings due to certain peculiarities. Conversely, remotely operated vehicles (ROVs), which are typically box-shaped and can be outfitted with manipulator limbs, are capable of interacting with their surroundings [13]. Energy consumption, however, renders the box-shaped ROV unsuitable for extended operations. Due to their fixed body structures, both AUVs and ROVs are unsuitable for operation in complex and confined spaces. The optimal utilization of underwater snake robots could significantly improve the maneuverability and efficacy of modern underwater vehicles [9, 11, 14]. The DOFs of the serpent robot's body structure enable it to generate a variety of gait patterns in response to its surroundings. This characteristic enables the serpent robot to navigate through confined spaces, and its unique body structure enables it to interact with its surroundings as a manipulator. These merits of the snake robot contribute to its superior performance in underwater exploration and rescue missions when compared to other varieties of underwater robots.

The numerous degrees of freedom of the snake robots make their modeling and control extremely complicated. When the serpent robot is in motion underwater, hydrodynamics present an additional difficulty in incorporating them into the system dynamics. Kelasidi et al. [9] utilized joint references to represent the snake robot in their

PID controller, which enables the robot to generate the intended forward locomotion. However, it is assumed that the serpent robot is adaptable to a variety of environments, which means that its locomotion could vary depending on the circumstances. Model Predictive Control (MPC) was implemented in [15] in order to generate a reference-free trajectory for the serpent robot. In this investigation, a simplified dynamic model of a serpent robot was applied. Incorporating complex hydrodynamics into the underwater serpent robot's system dynamics is essential. In addition, the lengthy computation time is a consequence of the dynamics' complexity. Furthermore, it is anticipated that the snake robot will possess the capability to actively utilize environmental factors to aid in its locomotion when traversing confined spaces or performing the operation, a process referred to as obstacle-aided locomotion [16]. Normal gradient-based MPC has difficulty incorporating discontinuous situations such as collisions into the system dynamics; therefore, an alternative type of sample-based MPC is utilized in this study, which substantially reduces the computational cost as well.

This thesis will provide a comprehensive introduction to the aforementioned problems, accompanied by corresponding solutions for each section. The primary objective of this research is to enhance the performance of an underwater snake robot while traversing a subsea environment devoid of fixed references and to enable the robot to alter its movement patterns to its surroundings by changing. Contact between the body of the serpent robot and its surroundings is another crucial aspect that must be taken into account when designing the control system. By incorporating the concept of discontinuous collisions into the MPC's prediction stages, snake robots will possess the capability to handle collisions with the surroundings and judiciously employ the forces produced by said collisions to propel themselves towards more efficient patterns. The computational time required for the control method is a significant obstacle as a result of the hydrodynamics and serpent locomotion complexities, which render the MPC procedure sluggish and unsuitable for real-time implementation. The complex dynamics of the serpent robot employed in this research will be presented in this thesis, along with the incorporation of hydrodynamics into the robot's body. Furthermore, the proposed control method cannot be implemented on the original system due to its excessive complexity; therefore, we will resolve this issue by converting the system into an alternative physical model. We will specifically demonstrate how contact dynamics are incorporated into the system and the method of collision detection that follows. A comprehensive introduction will be provided to the sample-based model predictive control that was employed to regulate the underwater serpent robot in conjunction with curvature derivative control. The advantages of these two methods of control will be demonstrated. In the event that a ship has sunk, the objective of this research is to equip the serpent robot with the capability to conduct search and rescue operations. In order to determine how the snake robot would react to altering conditions, the proposed control methods and the system will be evaluated during testing on a variety of environmental scenarios.

1.2 Biological Gait Patterns of Snake Robots

In the past few decades, numerous researchers have investigated snake robots with a focus on the implementation of nature-inspired modeling of serpent locomotion into physical robots. Snakes exhibit distinct locomotion patterns that correspond to their respective environments. Snakes will select the most efficient mode of locomotion for both land and aquatic travel in the wild. Numerous studies [7, 17, 18, 19, 20] have been devoted to the improvement and application of these properties to the design of serpent robots. The initial observation of the snake's locomotion was made by Gray et al. [21], who subsequently categorized these movements into four distinct gait patterns. This study also described the forces acting on the snake's body and how the serpent would utilize these forces as propulsion to advance. Hirose [6] conducted a biological investigation of a snake's locomotion on land and developed a model that represented it as a continuous curve capable of sideways movement without being affected by sideslip. This type of motion is referred to as the serpenoid curve, or lateral undulation, which is also the most common locomotion of snakes. Hirose specifically discovered that a planar curve whose curvature varies sinusoidally provides an approximation of the shape of a biological snake undergoing lateral undulation. The researchers in [22] reach the conclusion that serpentine curve locomotion is superior to serpenoid curve locomotion for snake robots when it comes to ground motion efficiency. Further investigation was conducted in [23] regarding the muscle activity patterns associated with the lateral undulation movement of a serpent, as well as the muscle activity required for curvature. To accomplish undulating locomotion while the snake is in motion on the ground, the frictional properties of the snake body are crucial. It is critical that the coefficient of friction between the skin of the serpent and the area of contact be directional [24]. It is necessary to acquire a low friction coefficient in the tangential direction and a high friction coefficient in the normal direction in order to propel the serpent robot forward. An illustration of this phenomenon can be found in the characteristics of lampreys and other anguilliform organisms [25].

There are four basic locomotion of the natural snake been investigated as : lateral undulation also known as serpentine locomotion, concertina, sidewinding and the rectilinear locomotion. Marvi et al. [26, 27, 28] studied the concertina locomotion and the rectilinear movement gait of snakes with the behavior of snake moving on the sand. These gait patterns were discussed in detail with also the application in snake robots. In [7], the four basic locomotion gait of biological snakes are fully introduced. Fig. 1.1 shows the various locomotion gaits of the snake and will be introduced briefly as follows. However, the biological snakes also show other kinds of gaits besides these four patterns in special situations such as sinus-lifting, rectilinear crawling, climbing and swimming, more details can be found in [29]. Sinus-lifting motion is discussed in [30] which is kind of improvement of lateral undulation by lifting part of the trunk in order to avoid lateral slippage and obtain better propulsion forces. This gait pattern was studied in detail and applied to the snake robots in [31, 32]. The snakes also have the ability to move forward in the underwater environment by using lateral undulation gait like eels [33]. Furthermore, some tree snakes can move through the open chest to achieve the sliding surface for generating optimal gait patterns [34, 35].

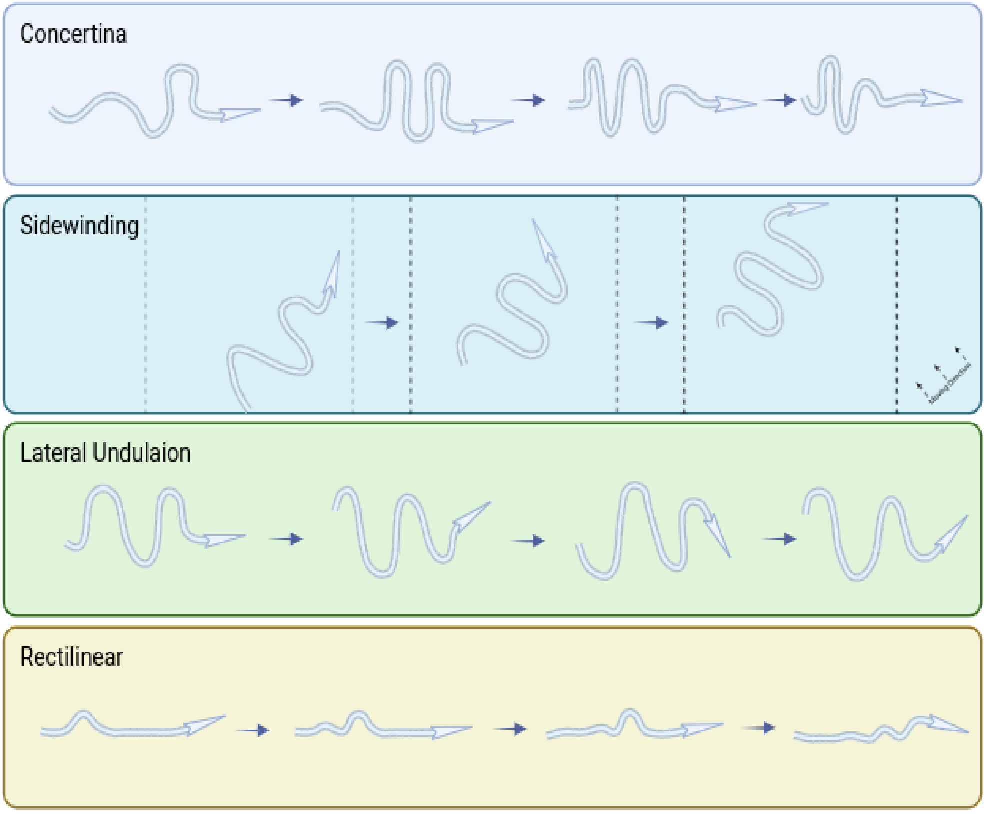


Figure 1.1: Four basic locomotions of biological snakes

1.2.1 Lateral Undulation

Lateral undulation constitutes the snake's most frequent locomotion. Furthermore, it is the only biological snake motion that does not depend on the static interaction between the snake's body and the surface of objects that are similar to it [36]. This locomotion is accomplished by the serpent's entire body moving in a continuous, sinuous pattern. Multiple points of contact between the snake's body and the surface are necessary for the frictional force between the contact areas to propel the serpent forward. When the serpent encounters an obstruction, such as a rock, it would leverage the force of contact to its advantage by striking the object. As the propulsion force that propels the snake forward, this motion produces a reaction force [37]. This phenomenon of serpent locomotion is alternatively referred to as obstacle-aided locomotion. A comprehensive explanation of this term can be found in the subsequent subsection.

1.2.2 Sidewinding Gait

As illustrated in Fig. 1.1, the sidewinding motion is a continuous transverse bending oscillation. This locomotion is characterized by only two contacts between the snake body and the surface, with no force acting in any other direction. Consequently, it minimizes slippage in comparison to lateral undulations [36, 27]. It is evident from the natural environment that desert snakes employ this sidewinding locomotion to propel themselves towards the dunes. The article presents an introduction to the biological mechanism of sidewinding and explores its implications for the design of serpent robots [38]. However, according to a study in [39], sidewinding locomotion is exclusive to certain species of snakes and is predominantly observed in arid desert environments with porous soil and sand [40].

1.2.3 Rectilinear Locomotion

Snakes advance using rectilinear gait, which is accomplished by the epidermis moving in the same direction as the bone relative to the motion. The rectilinear locomotion of biological snakes was examined by Lissmann et al. [41] in their work on the correlation between muscle activity and gait pattern. According to [42, 43], the snake acquires propulsion force by stretching muscles from the ribs that are affixed to the elastic skin in order to achieve this unique locomotion. A symmetrical form is utilized in the gait as opposed to a contraction wave. Due to the forward pulling of a portion of the abdominal epidermis, the abdominal scales develop in bundles. This type of gait requires only a minimal vertical movement throughout the entire locomotion process [44].

1.2.4 Snake Locomotion for Moving through Complex and Narrow Space

Snake robots possess the ability to operate effectively in a wide variety of environments, such as small spaces, rugged terrain, and subterranean environments, among others.

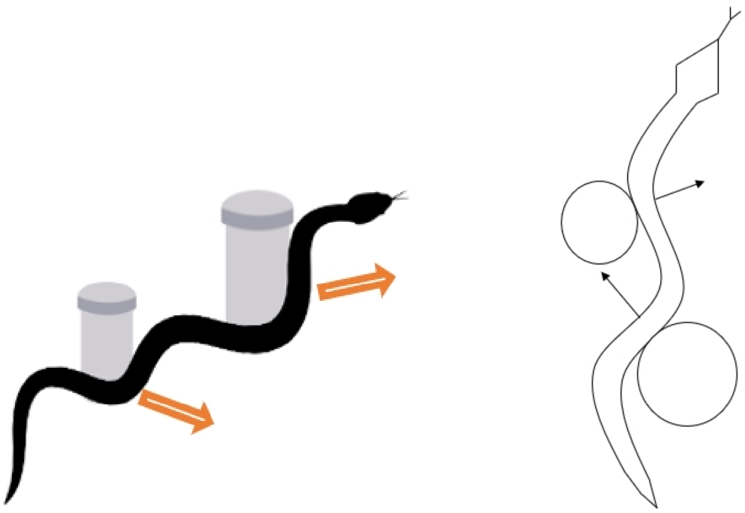


Figure 1.2: A simple schematic of the obstacle-aided locomotion.

The serpent demonstrates an assortment of locomotion patterns in reaction to changes in its surroundings. The underwater movement of the serpent robot is controlled by a lateral undulation [45, 7].

Snakes are capable of navigating through highly complex and confined environments while evading environmental interference due to their slender body structure. Snakes demonstrate environmental adaptation by employing a variety of body morphologies to traverse various environments, such as navigating obstacles and maneuvering through pipelines [46, 47]. Utilizing external propulsion forces generated by contact with mass-specific objects, such as boulders and stones, snakes advance more efficiently in the wild. This form of reaction force is commonly observed at the points of contact between the obstruction and the link. As depicted in Figure 1.2, the serpent robot generates propulsion forces by means of interaction with the obstacles; this facilitates the robot's forward motion, which is more efficient. The initial proposition to utilize the reaction force to overcome obstacles in the development of a wheeled snake robot was published in [48]. The approach cited in Nakajima et al. [49] outlined as obstacle-aided dynamics of the snake robot the generation of contact force and friction through the pressure of the snake body against the obstacles. A hybrid dynamic model was put forth by the authors of [16] to describe the interaction between a planer robot and obstacles. A coulomb friction model was employed to simulate the friction between the obstacle and the serpent body. In contrast, the contact force was determined through the solution of a linear complementarity problem (LCP).

When the snake robot is in a confined space, such as a conduit, frictional force generated by the contact pressure between the pipe's surfaces and the robot will propel it forward. In [50], the aforementioned self-locking phenomenon is implemented to optimize the concertina movement during ascent via a vertical conduit and actuator torque. The outcome illustrates that the number of connections that come into contact with the surface also has an effect. Torque is reduced as the number of contact linkages increases. In their study, Shapiro et al. [51] proposed a frictional compliance model for the serpent robot, which was designed to facilitate its ascent between two walls while ensuring stability and preventing slippage. This was achieved through the utilization of frictional contact generated between the bulk of the robot and the environment at large. The author illustrates in [52] how the performance of pipe robots is impacted when the diameter of the conduit is altered during pipe inspection operations. The snake robot is deliberately programmed with a sinusoidal gait pattern in order to facilitate control over its movements and enable it to adjust to its diverse environment. In their study, Ivan et al. [53] proposed a traveling wave that mirrored the shape of a trapezium to characterize the trajectory of a snake robot traversing a conduit. Reflexional locomotion, as opposed to the concertina locomotion utilized by snakes to navigate through confined spaces, is attained when traversing exceedingly narrow environments. The serpent robot's rectilinear locomotion through the conduit is depicted in Fig. 1.3. The concertina locomotion of the serpent is illustrated in Fig. 1.1.

1.3 Previous Studies on Underwater Robots

Nowadays underwater robots can be categorized as remotely operated underwater vehicles (ROVs), autonomous underwater vehicles (AUVs), or bottom-crawing-legged underwater robots. As the demand for underwater exploration and maintenance in

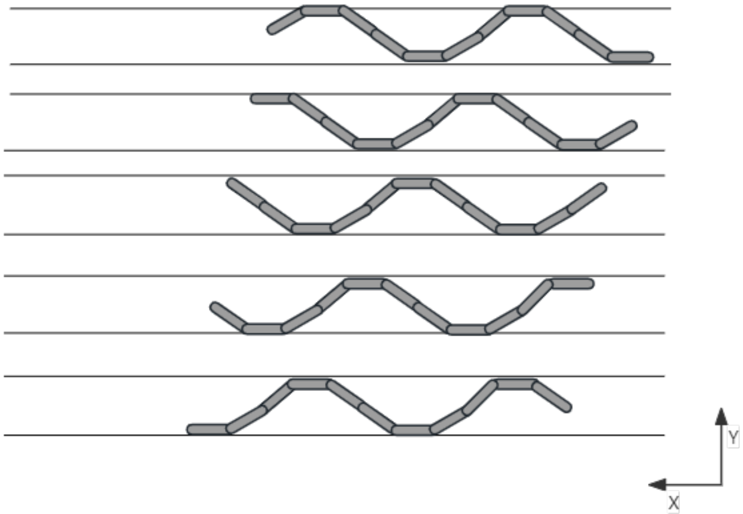


Figure 1.3: Trapezium-like traveling wave locomotion

increasingly complex and confined environments increases and robot limitations vary, scholars are beginning to concentrate on the design and control of underwater robots through the imitation of natural creature motion [54]. As previously stated, research has been conducted in the domain of underwater snake robots due to the increased flexibility exhibited by these devices when navigating diverse environments, owing to their high degrees of freedom. Additional studies have investigated the optimal modeling of eel-like robots [8, 25, 14].

In comparison to the dynamic of the body, the hydrodynamics that occur on the robot body are typically the most complex factors to account for when modeling an underwater robot. The investigations of [55] and Taylor [56] examined the dynamics of fluid forces acting on the body of an undulating automaton in motion. According to these studies, viscous forces predominate for small swimming objects at low Reynolds numbers; nevertheless, for larger machines, the added mass force becomes significant. Both the modeling of added mass force and drag forces are necessary for the underwater snake robot due to the significant role that both hydrodynamics play in the motion of underwater robotics [57]. Chen [58] proposed a model that describes the interaction between the body and the fluid of undulating swimming leeches. The hydrodynamic forces acting on the body, which is composed of rigid connections, are reactive and resistive in nature. Khali et al. [11, 59] proposed the Newton-Euler equation-based model of hydrodynamics for an underwater automaton resembling an eel. Additionally, a simplified fluid force model was introduced. An approach to representing the fluid-robot body dynamics is detailed in the article [60]. Nevertheless, in this investigation, neither the added mass force nor the drag force were accounted for in the system dynamics.

A comprehensive mathematical framework for the underwater serpent robot is expounded upon in the article [9]. The dynamic of the system takes into account fluid contact forces, which consist of added mass forces (reactive fluid forces), linear and non-linear drag forces (resistive fluid forces), and ocean currents. In addition, fluid torques are applied to each link of the serpent robot. By deriving the model in a closed form that obviates the need for numerical evaluation of drag effects, it becomes suitable for implementation in model-based control schemes. In addition, this model can be modified to represent a serpent robot traversing land by substituting ground friction for fluid friction and disregarding the impact of fluid forces. Kelasidi et al. [45, 61] additionally introduced a closed-form dynamic model of the underwater snake robot that incorporates additional effectors to produce supplementary propulsion forces. The kinematic model and equations of motion are adjusted in [62] to account for the possibility that the length and ass of each link may vary. A multitude of operations have been executed utilizing this dynamic prototype of an underwater serpent robot. The authors of [63] demonstrate that the path following controller proposed for a snake system with a sinusoidal gait pattern and an integral line-of-sight (LOS) guidance law is effective for both lateral undulation and eel-like motion.

1.4 Control Methods for Snake Robots

In recent years, numerous control strategies have been suggested for the underwater snake robot as a means of addressing the intricate control design challenges unique to snake robots. The research described in [64, 65] focuses exclusively on the turning and forward motions of bio-inspired swimming robots. The results of [66, 25] regarding the control of eel-like motion demonstrated the ability to trace both straight and curved trajectories. [67] present the outcome of implementing feedback control on an eel-like robot; two pectoral fins affixed to the cranium maintain the robot's rolling angle in a stable manner. The research was expanded upon in [68], wherein a motion control system for the three-dimensional eel-like robot was developed; without pectoral fins, the robot's three-dimensional position could be tracked and its rolling angle could be maintained. The energy utilization of an underwater snake robot's control design for extended operation is also a significant factor. The control of individual joints of the underwater snake robot is achieved using a basic PID controller, as described in [69]. The simulation outcomes demonstrated that the empirical principles put forth for determining the parameters of gait patterns effectively managed energy consumption and forward velocity while maintaining the desired level of efficiency.

Jiang et al. [70] introduced a novel decoupling control approach in recent years to address the issue of full-link trajectory tracking of an underwater serpent robot that incorporates axis-intersect joints and vector thrusters. In addition to handling full-link trajectory tracking, the proposed control method is capable of handling complex disturbances such as contact in confined space and ocean current. The new control strategy isolates the head link's movement from that of the remaining links. Thrusters with a specified reference trajectory govern the body's motion, while the remaining links are controlled in accordance with their relative angles to the head link. The research conducted by Xu et al. [71] centred on the tracking control of curved paths for a serpent robot submerged in water. A proposed approach for guiding a desired path utilizes parametric cubic-spline interpolation (PCSI) to enhance the LOS (light of sight) method. For energy efficiency, a pigeon-inspired optimization algorithm enhanced by quantum principles (QPIO) is proposed. Simulation results showed that the robot is able to accurately follow the intended curved path as long as achieving energy efficiency.

Prior investigations have identified a pre-established gait pattern for underwater snake robots, which enables the robot to effortlessly carry out its designated locomotion during operation [72]. Nevertheless, for the serpent robot to effectively navigate the diverse environments, as well as the exceedingly intricate and capricious nature of the space it explores, it must generate an extensive repertoire of motions. Recently, model predictive control (MPC) has been integrated into the control of serpent robots. MPC is an optimal control method in which, after solving a finite-horizon optimization problem, the optimal input is supplied to the system. Economic MPC is applied to the control of a simplified snake robot model in [15]. The outcomes indicate that the snake robot generates a variety of patterns without the need for a predefined gait pattern, and EMPC outperforms economic MPC in terms of forward moving speed and energy consumption. Evan et al. demonstrate the effectiveness of EMPC for obstacle-assisted serpent locomotion in their study [73]. A model of continuous compliant contact was suggested. Undulating gait patterns resulted, and the serpent robot advanced by utilizing the anisotropic ground friction. However, the computation of each optimal input in the simulations takes anywhere from 5 to 200 seconds, and no attempt was made to decrease this time.

Sample-based MPC, in contrast to gradient-based MPC, does not necessitate gradient information, enabling the direct incorporation of discontinuities into the MPC prediction stages. Our previous study employed Monte Carlo model predictive control (MCMPC), which falls under the category of sample-based MPC, to analyze the underwater serpent robot. By incorporating contact dynamics into the system, we provide evidence that the robot can effectively navigate intricate surroundings using obstacle-assisted locomotion. For the simulation, MATLAB 2021b was utilized, and each control cycle necessitates an exceptionally lengthy computation time. As a result, this study employs CUDA, a parallel computation framework that leverages GPUs, to decrease the amount of time required.

1.5 Contribution of the Thesis

This paper describes the adaptive control of the underwater snake robot that operates in open space as well as environments with constraints. A combination of two control methods are proposed in this study which enables the snake robot to move through diverse underwater environments containing various kinds of obstacles and terrains. The contributions that this thesis makes are outlined below.

Previous research has incorporated the hydrodynamics of the underwater environment into the closed-form solution of the dynamic model of the underwater snake robot. This eliminates the need for numerical analysis of drag effects, enabling the model to be applied in control schemes that rely on models. Nevertheless, simulation results indicate that this model remains highly complex to be implemented in advanced control methodologies like model predictive control. The complexity would increase significantly if contact situations were incorporated. In this study, we propose an easy approach for describing the dynamics of a system that is capable of executing complex dynamics in parallel. The external forces can be added directly to the related parts of the snake robot. Simulation results lead to a reduction in the computational cost when compared to the original forms.

For the purpose of enabling the snake robot to adapt to changing surroundings, Monte Carlo Model Predictive Control and Curvature Derivative Control are implemented in the control system. MCMPC provides the snake robot with the capability to execute a wide variety of gait patterns in the absence of a predetermined reference. Discontinuous situations, such as collisions between the body link and the environment, can be incorporated directly into the system dynamics with the assistance of this sampling-based MPC. The snake can take fully advantage of this ability to propel itself forward by employing obstacle-aided locomotion in open environments containing numerous obstacles. In this study, we tested the performance of the snake robot for moving in open space, through multiple obstacles and passing narrow pipes. The contact between the snake body and the surroundings are introduced into the system dynamics. With the help of the control method, the snake successfully handle with the various situations.

The contribution of CDC can be summarized as: First, setting each joint as the control target for MCMPC would significantly rise the complexity of calculation which causes huge computational cost for processing. With the help of CDC, the entire snake body can be controlled based on the head joint. Another benefit of CDC is that the external environment's effect on one link can be transferred to the adjacent link, thereby adapting the snake's body to the varied surroundings. In addition to merely transmitting the joint angle to subsequent joints, propulsion forces that operate on preceding components can also be transferred to the succeeding link. This implies that snake robot consistently selects the most advantageous portion of contact area to facilitate its forward motion.

1.6 Constructs of the Thesis

The conclusions of the research are detailed in the subsequent chapters of this thesis. The introductory section of the first chapter provides an overview of the context and rationale underlying this research. The subsection of the first chapter provides an introduction to the history and review of previous research that focuses primarily on the natural inspiration for the locomotion of various snakes. Four major snake gaits are described in detail, along with the snake's movements as it traverses a complex and confined space. Demonstrating various control methods proposed for each control purpose. The intricate dynamic model of the submerged snake automaton is comprehensively introduced in Chapter 2, which is devoted to hydrodynamics. The efficacy of the proposed control methods is demonstrated through simulation outcomes in Chapter 3. Chapter 4 provides simulations that illustrate the various environments in which the underwater snake robot operates. The efficacy of the snake robot in transitioning from unrestricted environments to open space is detailed. Chapter 5 contains discussions of the conclusions and prospective research, respectively.

The relationship between the content of this thesis and the author's publications will be shown here. The following publications have resulted from the research conducted during the course of the PhD program. Each publication corresponds to specific chapters or sections of this thesis, as detailed below.

1. Peer-reviewed journal article:

   Y. Qiu and H. Date, "A low computation-cost locomotion control for underwater snake robot based on monte carlo model predictive control and curvature derivative control," *Advanced Robotics*, vol. 38, no. 11, pp. 770–783, 2024.

2. Peer-reviewed international conference:

   Y. Qiu and H. Date, "Obstacle-aided locomotion for underwater snake robot using monte carlo model predictive control and curvature derivative control," in *Proceedings of the 2023 62nd Annual Conference of the Society of Instrument and Control Engineers (SICE)*, pp. 690-695, 2023.

3. Refereed international conference papers (Abstract Review):

   Y. Qiu and H. Date, "Monte carlo model predictive control for underwater snake robot locomotion," in *Preprints of the 22nd World Congress of the International Federation of Automatic Control (IFAC World Congress)*, pp. 6244-6247, 2023.

The sections of 3.5, from 4.3 to 4.6 related to the GPU parallel computing, the performances of the snake robot moving through the confined environments and the benefits of utilizing the proposed methods are based on the journal article [74].

The locomotion of the snake robot moving in open sub-sea environment described in section 4.1 are based on the [75]. And the results of the snake moving among several obstacles which realized the obstcal-aided locomotion are based on the Peer-reviewed international conference [76], respectively.

Chapter 2

Dynamic Model of Underwater Snake Robot

As stated in the introduction, the intricacies of locomotion dynamics and hydrodynamics become apparent when a robot operates in a subsea environment, potentially rendering them unsuitable for control applications. Therefore, appropriate consideration should be given to the hydrodynamics incorporated into the underwater serpent robot's locomotion model. The closed-form derivation of the model suggested in Kelasidi et al. [9] is implemented in model-based control methodologies. The fluid forces and torques, including added mass force, linear and nonlinear drag force, and fluid torques, are accounted for in this model. It is hypothesized that the underwater serpent robot, which is neutrally buoyant, follows the proposed model while moving in the virtual two-dimensional plane. The impact of ocean current is likewise incorporated into the system model in the form of the corresponding link velocity. This model takes into account the critical hydrodynamics that occur on the snake body and are the primary determinants of the snake robots' underwater locomotion. A control-oriented model, described in [77], is also put forth; it functions as a simplified iteration of the complex model. On the other hand, the translational motion of each joint is proposed in lieu of the rotational motion between two links in this straightforward model of the snake robot. While this control-oriented model is comparatively simpler and therefore better suited for control purposes, it may not fully capture the contact dynamics experienced by the snake robot as it navigates through confined spaces when the joint is in translational motion. As a result, the intricate dynamics of the robotic snake submerged are employed. With complex dynamics, the control method proposed in Chapter 3 can mitigate the drawback of computational inefficiency.

This chapter provides an introduction to the serpent robot's components, as well as the intricate equations of motion that govern its underwater operation. The system incorporates various components of hydrodynamics, such as added mass force, ocean current effect, linear and nonlinear drag force, and fluid torque. In this investigation, the complex model is employed to regulate the hydrodynamics. Nevertheless, the dynamic model's initial mathematical equations are excessively complex to be utilized in the parallel computation of the proposed method. Thus, the fundamental structure of the model is constructed independently, and the dynamics are subsequently incorporated. This modification enables the underwater serpent robot's new form of the motion model to achieve improved computation time performance; further details on this will be provided in Chapter 3.

The structure of this chapter is as follows. In Section 2.1, the fundamental structure of the underwater serpent robot's components is described. In the same section, the definitions of mathematical symbols, matrices, and vectors are also provided. Illustrated is the prototype of the underwater serpent robot. In Section 2.2, the definitions of hydrodynamics are elaborated. In Section 2.3, a summary of the chapter concludes the material.

2.1 Components of the Underwater Snake Robot

In this section, the basic structure of the underwater snake robot and the kinematics are introduced. Fig. 2.1 shows the body structure of the underwater snake robot. The robot contains n rigid links, and the distance between the endpoint and the center of

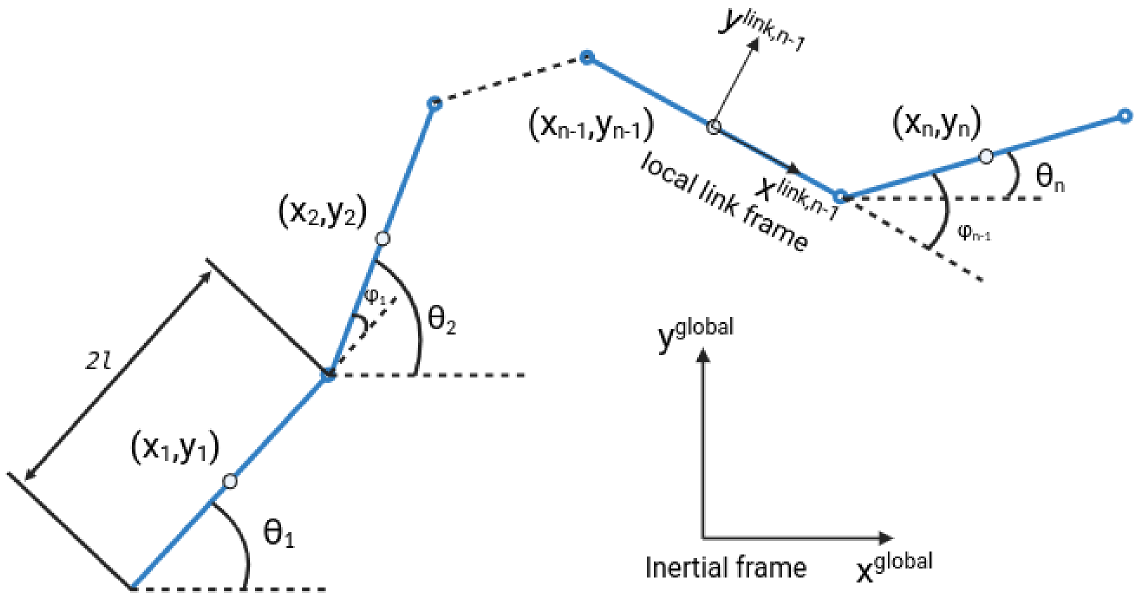


Figure 2.1: Kinematic model of the underwater snake robot

mass (CM) of each link $i \in 1, ..., n$ is equal to l so that the total length of one link is $2l$. Links are connected to each other by $n-1$ joints. The mass of each link is maintained the same at m with the moment of inertia of $J = \frac{1}{3}ml^2$. The CM position is utilised to represent the x-y coordinates of each link as x_i and y_i in the global frame. The angle between the link i and the global x-axis is represented as θ_i and the joint angle ϕ_i which is the difference between the link i and $i-1$ is defined as

$$\phi_i = \theta_i - \theta_{i-1} \tag{2.1}$$

The CM of the whole robot P_{CM} is given by

$$P_{CM} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \frac{1}{nm}\sum_{i=1}^{n} mx_i \\ \frac{1}{nm}\sum_{i=1}^{n} my_i \end{bmatrix}, \tag{2.2}$$

Moreover, the state variables are specified as $x = [\theta^T, P_{CM}^T, \dot{\theta}^T, \dot{P}_{CM}^T]$ where θ represents the link angle which is defined as the angle formed by each link with respect to the

Symbol	Description
n	The total number of links of the snake robot
l	The half length of the link, total length is thus $2l$
m	The mass of each link
ϕ_i	The ith joint angle
θ_i	The ith link angle
x_i, y_i	Center of mass of the ith link in global coordinate
p_x, p_y	Center of mass of the snake robot in global coordinate
λ_1	Coefficient of the added mass force
λ_2, λ_3	Coefficients of the drag force
$h_{x,i}, h_{y,i}$	joint constraint forces act on link i from adjacent links in x and y directions
$f_{x,i}, f_{y,i}$	externally hydrodynamic forces act on link i in x and y directions
α_i	The direction parameter determines the side of contact with the pipe
k_{msd}, d_{msd}	The spring and damping coefficients
$R_{L_i}^{global}$	The rotation matrix from the link frame to the global frame

Table 2.1: Definitions of mathematical symbols.

global x-axis. This 2-dimensional model is proposed under the assumption that the snake robot moves in a virtual horizontal and flat plane, and the whole snake body is fully submerged. The robot has $n + 2$ degrees of freedom which contain the $x - y$ position in the plane, $n - 1$ joint coordinates, and the orientation, respectively.

Table 2.1 shows the meaning of each mathematical symbols of the model of the underwater snake robot utilized in this study. The rotation matrix from the link to the global frame is defined as

$$\mathbf{R}_{link}^{global} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \tag{2.3}$$

The calculation of the CM position of each link in the global frame is shown as

$$\begin{aligned} \mathbf{X} &= -l\mathbf{K}^T \cos\boldsymbol{\theta} + \mathbf{e}p_x, \\ \mathbf{Y} &= -l\mathbf{K}^T \sin\boldsymbol{\theta} + \mathbf{e}p_y, \end{aligned} \tag{2.4}$$

Then the related velocities and accelerations of links can be further calculated through the differentiation respect to time which will be used for the modelling of the fluid forces. The equation of the velocity is shown in (2.5) and the acceleration in (2.6), respectively.

$$
\begin{aligned}
\dot{\mathbf{X}} &= l\mathbf{K}^T\mathbf{S}_\theta\dot{\boldsymbol{\theta}} + \mathbf{e}\dot{p}_x, \\
\dot{\mathbf{Y}} &= -l\mathbf{K}^T\mathbf{C}_\theta\dot{\boldsymbol{\theta}} + \mathbf{e}\dot{p}_y.
\end{aligned} \tag{2.5}
$$

$$
\begin{aligned}
\ddot{\mathbf{X}} &= l\mathbf{K}^T(\mathbf{C}_\theta\dot{\boldsymbol{\theta}}^2 + \mathbf{S}_\theta\ddot{\boldsymbol{\theta}}) + \mathbf{e}\ddot{p}_x, \\
\ddot{\mathbf{Y}} &= l\mathbf{K}^T(\mathbf{S}_\theta\dot{\boldsymbol{\theta}}^2 - \mathbf{C}_\theta\ddot{\boldsymbol{\theta}}) + \mathbf{e}\ddot{p}_y.
\end{aligned} \tag{2.6}
$$

where the definitions of the matrices and vectors are $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_n]^T \in \mathbb{R}^n$, $\dot{\boldsymbol{\theta}}^2 = [\dot{\theta}_1^2, \cdots, \dot{\theta}_n^2]^T \in \mathbb{R}^n$ $\mathbf{X} = [x_1, \cdots, x_n]^T \in \mathbb{R}^n$, $\mathbf{Y} = [y_1, \cdots, y_n]^T \in \mathbb{R}^n$ and $\mathbf{K} = \mathbf{A}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D} \in \mathbb{R}^{n \times n}$. The other expressions of the defined matrices are shown as follows,

$$
\mathbf{A} = \begin{bmatrix} 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix}, \mathbf{e} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^T \in \mathbb{R}^n,
$$

$$
\begin{aligned}
\sin\boldsymbol{\theta} &= \begin{bmatrix} \sin\theta_1 & \cdots & \sin\theta_n \end{bmatrix}^T \in \mathbb{R}^n, \mathbf{S}_\theta = \operatorname{diag}(\sin\boldsymbol{\theta}) \in \mathbb{R}^{n \times n}, \\
\cos\boldsymbol{\theta} &= \begin{bmatrix} \cos\theta_1 & \cdots & \cos\theta_n \end{bmatrix}^T \in \mathbb{R}^n, \mathbf{C}_\theta = \operatorname{diag}(\cos\boldsymbol{\theta}) \in \mathbb{R}^{n \times n}.
\end{aligned}
$$

2.2 Hydrodynamics of Subsea Environments

In the context of an underwater environment, where the Reynolds number is estimated to be between 10^4 and 10^5, it becomes imperative to incorporate resistive and reactive forces into the modeling. These forces play a significant role in generating the propulsion force that the underwater snake robot employs to advance. As stated in the preceding chapters, each link body is influenced by the additional mass force, linear and nonlinear drag forces, fluid torque, and ocean current. Within this particular segment, the hydrodynamics will be presented in isolation before being integrated into the dynamic model of the submerged serpent robot.

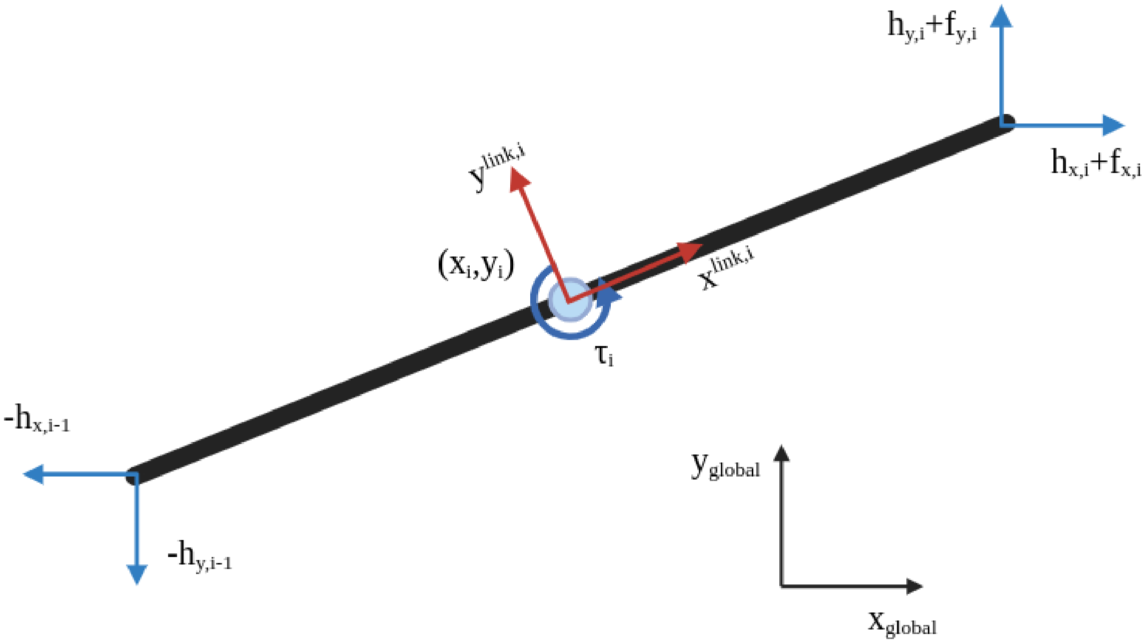


Figure 2.2: Decomposition of forces act on the link of the snake robot

Fig. 2.2 shows the forces act on each link from both the surrounding environment and the adjacent links, where h_x and h_y are the joint constraint forces from the adjacent links in x and y coordinates. f_i represents the external fluid forces on link i and τ is the fluid torque on the link. All the related forces are expressed in the global frame. Hydrodynamics are represented in a simplistic manner in the modeling due to the complexity[9]. The total hydrodynamic forces in this study include ocean current, linear and nonlinear drag forces, and added mass forces, as these types of forces significantly contribute to the propulsion of underwater movements[57]. Several assumptions regarding the dynamic model of the underwater snake robot are presented.

- The fluid is viscid, incompressible, and irrotational in the inertia frame.
- The robot is assumed to be neutrally buoyant where the gravity and the buoyancy cancels each other.
- The current expressed in the inertia frame is contact and irrotational.

It is presumed that the velocity on any part of the link is represented by the relative velocity of the centre of mass of each link. The fluid forces are included into the system model through the effects on the related velocity of each link. The drag force has effect on the link in the direction of both tangential and normal which is in the form of an anisotropic friction force. The representation of the fluid forces acting on each link as shown in Fig. 2.2 can be calculated as the combination of the sum of the added mass force and the drag force as

$$\begin{bmatrix} \mathbf{f_x} \\ \mathbf{f_y} \end{bmatrix} = \begin{bmatrix} \mathbf{f_{Ax}} \\ \mathbf{f_{Ay}} \end{bmatrix} + \begin{bmatrix} \mathbf{f^I_{Dx}} \\ \mathbf{f^I_{Dy}} \end{bmatrix} + \begin{bmatrix} \mathbf{f^{II}_{Dx}} \\ \mathbf{f^{II}_{Dy}} \end{bmatrix} \tag{2.7}$$

where f_A represents the added mass force. f_D^I and f_D^{II} show the linear and nonlinear drag forces, respectively. First, the implementation of the added mass force is introduced. Added mass force is the inertia added to the link body caused by the movement of the link which will take some volume of surrounding fluid. As introduced in [9], the added mass force is expressed as

$$\begin{bmatrix} \mathbf{f_{Ax}} \\ \mathbf{f_{Ay}} \end{bmatrix} = - \begin{bmatrix} \mu_n(\mathbf{S}_\theta)^2 & -\mu_n\mathbf{S}_\theta\mathbf{C}_\theta \\ -\mu_n\mathbf{S}_\theta\mathbf{C}_\theta & \mu_n(\mathbf{C}_\theta)^2 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{X}} \\ \ddot{\mathbf{Y}} \end{bmatrix} - \begin{bmatrix} -\mu_n\mathbf{S}_\theta\mathbf{C}_\theta & -\mu_n(\mathbf{S}_\theta)^2 \\ \mu_n(\mathbf{C}_\theta)^2 & \mu_n\mathbf{S}_\theta\mathbf{C}_\theta \end{bmatrix} \begin{bmatrix} \mathbf{V^a_x} \\ \mathbf{V^a_y} \end{bmatrix} \dot{\boldsymbol{\theta}} \tag{2.8}$$

where $\mathbf{V^a_x}, \mathbf{V^a_y}$ represent the current velocity in the inertial frame.

When a solid moves through a fluid environment a resistive force will be added to the object in the way opposite to the forward moving direction. For snake robots moving underwater, the drag force plays an important role in the propulsion of the snake locomotion. In this dynamic model of the underwater snake robot, the linear and nonlinear drag forces are shown as,

$$\begin{bmatrix} \mathbf{f^I_{Dx}} \\ \mathbf{f^I_{Dy}} \end{bmatrix} = - \begin{bmatrix} c_t\mathbf{C}_\theta & -c_n\mathbf{S}_\theta \\ c_t\mathbf{S}_\theta & c_n\mathbf{C}_\theta \end{bmatrix} \begin{bmatrix} \mathbf{V_{r_x}} \\ \mathbf{V_{r_y}} \end{bmatrix} \tag{2.9}$$

$$\begin{bmatrix} \mathbf{f^{II}_{Dx}} \\ \mathbf{f^{II}_{Dy}} \end{bmatrix} = - \begin{bmatrix} c_t\mathbf{C}_\theta & -c_n\mathbf{S}_\theta \\ c_t\mathbf{S}_\theta & c_n\mathbf{C}_\theta \end{bmatrix} \operatorname{sgn}\left(\begin{bmatrix} \mathbf{V_{r_x}} \\ \mathbf{V_{r_y}} \end{bmatrix} \right) \begin{bmatrix} \mathbf{V_{r_x}}^2 \\ \mathbf{V_{r_y}}^2 \end{bmatrix} \tag{2.10}$$

where the relative velocities are shown as,

$$\begin{bmatrix} \mathbf{V_{r_x}} \\ \mathbf{V_{r_y}} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_\theta & \mathbf{S}_\theta \\ -\mathbf{S}_\theta & \mathbf{C}_\theta \end{bmatrix} \begin{bmatrix} \dot{\mathbf{X}} - \mathbf{V_x} \\ \dot{\mathbf{Y}} - \mathbf{V_y} \end{bmatrix} \tag{2.11}$$

Another important environment factor which should be considered into the dynamic of the system is the ocean current [78]. The effect of the current is introduced into the system in the form of the velocity vector. This current velocity is added to the speed of each link and is assumed to be constant in the inertial frame. Therefore, the related link velocity can then be expressed as $v_{r,i}^{link,i} = \dot{p}_i^{link,i} - v_{c,i}^{link,i}$, where $v_{c,i}^{link,i}$ represents the current velocity in the local link frame by transforming the global velocity utilizing the transform matrix. Then the acceleration of the link is calculated as,

$$\dot{v}_{c,i}^{link,i} = \begin{bmatrix} -\sin\theta_i\dot{\theta}_i & \cos\theta_i\dot{\theta}_i \\ -\cos\theta_i\dot{\theta}_i & -\sin\theta_i\dot{\theta}_i \end{bmatrix} \begin{bmatrix} V_{x,i} \\ V_{y,i} \end{bmatrix} \tag{2.12}$$

The fluid forces are assumed to act on only the center of mass of each related link as mentioned before. Except the fluid forces, fluid torque also has impact on the CM of the links depend on the Morison's equations. The fluid torques can be calculated by the following equations as,

$$\tau = -\Lambda_1\ddot{\theta} - \Lambda_2\dot{\theta} - \Lambda_3\dot{\theta}|\dot{\theta}| \tag{2.13}$$

where the $\Lambda_1 = \lambda_1 I_n, \Lambda_2 = \lambda_2 I_n, \Lambda_3 = \lambda_3 I_n$. The definitions of the related coefficients can be found in [9, 10] for more details.

As shown in Fig. 2.2, the total forces balance act on the link are given by,

$$m\ddot{\mathbf{X}} = \mathbf{D}^T\mathbf{h}_x + \mathbf{f_x}, m\ddot{\mathbf{Y}} = \mathbf{D}^T\mathbf{h}_y + \mathbf{f_y} \tag{2.14}$$

And the torque balance of links can be expressed as,

$$\mathbf{J}\ddot{\theta} = \mathbf{D}^T\mathbf{u} - l\mathbf{S}_\theta\mathbf{A}^T\mathbf{h}_x + l\mathbf{C}_\theta\mathbf{A}^T\mathbf{h}_y + \tau \tag{2.15}$$

By introducing the related expressions of each mathematical symbols and inserting into the torque balance of the links in Eq. 2.15 gives the dynamic model of the underwater snake robot as,

$$\mathbf{M}_\theta\ddot{\theta} + \mathbf{W}_\theta\dot{\theta}^2 + \mathbf{V}_\theta\dot{\theta} + \Lambda_3|\dot{\theta}|\dot{\theta} - l\mathbf{S}_\theta\mathbf{K}\mathbf{f}_{\mathbf{D_x}} + l\mathbf{C}_\theta\mathbf{K}\mathbf{f}_{\mathbf{D_y}} = \mathbf{D}^T\mathbf{u} \tag{2.16}$$

where $\mathbf{f}_{\mathbf{D_x}} = \mathbf{f}^{\mathbf{I}}_{\mathbf{Dx}} + \mathbf{f}^{\mathbf{II}}_{\mathbf{Dx}}$ and $\mathbf{f}_{\mathbf{D_y}} = \mathbf{f}^{\mathbf{I}}_{\mathbf{Dy}} + \mathbf{f}^{\mathbf{II}}_{\mathbf{Dy}}$ represent the sum of both linear and nonlinear drag force. The definition of the matrices $\mathbf{M}_\theta$, $\mathbf{W}_\theta$ and $\mathbf{V}_\theta$ can be found in [9], respectively. The state variables of the dynamic model for the underwater snake robot are chosen as the angle of links and the position of the center of mass of the entire robot as $\mathbf{x} = \left[\theta^T, \mathbf{p}_{CM}^T, \dot{\theta}^T, \dot{\mathbf{p}}_{CM}^T\right]^T \in \mathbb{R}^{2n+4}$.

2.3 Dynamic model transformed for parallel computing

The control methods that have been suggested are implemented in this research to generate motion for the underwater snake robot across various scenarios. In Chapter 3, the control mechanisms will be described in detail. Notably, in the case of this type of sample-based model predictive control, parallel computation will be utilized to execute the predictions of the dynamics. Consequently, these methods are unsuitable for the underwater snake robot's completely complex dynamic model, resulting in a substantial computational expense. Following this, the dynamic model of a 10-link snake robot is recalculated using the MotionGenesis software, which generates a dynamic model that accurately represents the complex hydrodynamics and complete kinematics of the subsea environment and the underwater snake robot. Incorporating external forces, such as the contact force between the serpent's body and the obstacles, into the system dynamics is also possible. In Chapter 3, the efficacy of the transformed model as measured by the computation time for the MCMPC procedure will be demonstrated.

MotionGenesis is a kind of fast, compact, highly advanced symbolic manipulator with expert tools for mathematics, code generation, and forces and motion. It incorporates Newtonian physics and simulates the motion of mechanical, biomechanical, aerospace, and related systems. Both linear and nonlinear systems can be described as fast, compact and optimized codes for high-speed and low-memory in-the-loop hardware. The external forces that act on each link are designed as a changeable variable as,

$$force(a1o, fy_1 * a12 > + fx_1 * a11 >) \tag{2.17}$$

where $a1o$ represents the CM of the first link, $a12 >$ and $a11 >$ are the direction vectors of link 1 in the x and y coordinates of the local frame. The fy_1 and fx_1 are the external forces added to the link which include the hydrodynamics and contact forces. So the total external hydrodynamic forces added to the ith link of the underwater snake robot in this study can be expressed as

$$\begin{cases} fx_i = f_{Ax,i} + f^{I}_{Dx,i} + f^{II}_{Dx,i} \\ fy_i = f_{Ay,i} + f^{I}_{Dy,i} + f^{II}_{Dy,i} \end{cases} \tag{2.18}$$

The fluid torque is also changed into the form of

$$torque(a1, flutor_1 * a13 >) \tag{2.19}$$

where $a1$ represents link 1 and the $flutor1$ is the value of the fluid torque acts on the first link. $a13 >$ is the vector of link 1 in the $z - axis$ direction. After compiling the code by MotionGenesis, a simple version of the dynamics is given in C code used as the dynamics of the underwater snake robot for the parallel computation. The comparison results of different forms of the dynamic model utilized for the same control method will be shown in the next chapter.

2.4 Chapter Summary

In this chapter, the basic component of the structure for the underwater snake robot is introduced at first in 2.1 section. Fig. 2.1 shows the structure of the underwater snake robot. The robot is assumed to be consist of n links with the same length and mass. Links are connected to each other through $n-1$ joints gives the robot $n+2$ degrees of freedom. The expressions and meaning of each mathematical symbols declared for the underwater snake robot model are described in detail such as the link angle θ and joint angle ϕ. The positions, related velocities and accelerations of center of mass of each link and the whole robot are used for the calculation of the snake kinematics which are shown as (2.4), (2.5) and (2.6).

The hydrodynamics of the subsea environment are introduced in 2.2. Several assumptions have been made for the simplification of the complex and highly nonlinear dynamics for a rigid body in the underwater environment. Fig. 2.2 shows the forces act on each link of the snake robot which includes the reactive forces as the added mass effects and the resistive forces consist of linear and nonlinear drag forces. Ocean current and the constraint forces act on link i from link $i-1$ and $i+1$ are also added to the dynamic system of the snake robot. The calculation of added mass force is shown as (2.8) and the one for the linear and nonlinear drag forces are based on (2.9) and (2.10), respectively. Fluid torques act on each link are also introduced with the Eq. 2.15. The state variables for the dynamic model are chosen as the link angles and the center of mass of the whole robot with also the related velocities. Then the force balance and the torque balance are shown as (2.14) and (2.15). The fully description of the dynamic model of the underwater snake robot with the complex dynamics is finally calculated in (2.16).

Finally in section 2.3, the transformed model of the underwater snake robot is introduced. A more simple expression of the kinematics of the snake robot can be obtained while keeping the origin external forces and torques with the help of MotionGenesis. The complex equations which describe the dynamics are modified into high-speed forms for the parallel computing procedure of MCMPC. The details of the Monte Carlo Model Predictive Control and Curvature Derivative Control will be discussed in the next chapter with also the evidence of the benefits of utilizing the transformed dynamic model compared to the original complex one.

Chapter 3

Control Methods for Underwater Snake Robot

The locomotion of sea snakes, fascinating creatures of the marine world, showcases remarkable adaptations enabling them to navigate their aquatic habitats with astonishing agility and efficiency. Unlike their terrestrial counterparts, sea snakes have evolved specialized anatomical features and locomotion strategies tailored to life in the ocean's depths. Through a combination of undulating movements, precise muscle control, and hydrodynamic principles, sea snakes exhibit a unique locomotion style that allows them to thrive in diverse marine environments.

The locomotion of sea snakes is primarily characterized by undulatory motion, a serpentine movement pattern characterized by rhythmic waves propagating along the length of the body. Sea snakes employ a unique form of locomotion known as "concertina" locomotion. This mode of locomotion allows sea snakes to navigate through complex underwater environments with precision and maneuverability, enabling them to explore coral reefs, navigate through underwater vegetation, and hunt for prey in tight spaces. The ability to anchor themselves to stationary objects also serves as a defensive strategy, allowing sea snakes to maintain position against strong currents or turbulent waters.

In addition to concertina locomotion, sea snakes utilize lateral undulations of their body to generate forward propulsion and steering. By flexing their body from side to side, sea snakes create undulatory waves that propagate along their length, pushing against the surrounding water and propelling them forward. This form of locomotion is particularly effective for long-distance travel and high-speed pursuits, allowing sea snakes to chase down prey or evade predators with remarkable agility.

Thus, the suitable control methods chosen for the locomotion control become very important depends on various of designed purposes and also meet the demands of generating complex gait patterns. Different control methods have be utilized for controlling the snake robots among previous studies such as normal PID control and model predictive control. The work in [9] used a simple PID controller by setting each joint as the control inputs. With a given reference to each joint, the robot could achieve lateral undulation gait when it is moving in an open space. However, as introduced in Chapter 2, there are many kinds of gait patterns observed from the nature snake in various environments. Thus, the control purpose is that the snake robot can change and generate its own locomotion based on the surrounding environments. In [15], model predictive control was designed as the control method for each joint which generates more efficient locomotion compared to PID control without a given reference.

In this chapter, the proposed control methods for the motion control of the underwater snake robot in this study are introduced. A sample-based model predictive control is chosen as the main control method with also curvature derivative control. With the help of these two control methods, the underwater snake robot is able to move through both open spaces and environments with constraints which is introduced in detail in the next chapter. One main advantage of using MCMPC is that it can deal with discontinuous phenomena as well as reduce the computational cost.

3.1 Overview of Control Methods for Snake Robots

Snake robots, inspired by the locomotion of serpents, offer unique capabilities for traversing complex and confined environments. These versatile robots find applications in various fields, including search and rescue operations, exploration of hazardous environments, and inspection of industrial infrastructure. Achieving precise control over the motion of snake robots is essential for maximizing their effectiveness and adaptability in diverse tasks and environments. In the past few years, researchers have proposed many control methods to achieve different control purposes.

PID control is a common control method chosen for generating snake robot locomotion. In [79], modules support angular position, velocity, and torque control through cascaded PID control. It is also utilized in [9] for the control of the underwater snake robot. A standard PD-controller is used to calculate the joints' actuator torques from the given joints' reference angles. The robot is able to generate both lateral undulation and eel-like motion. The work in [80] achieves autonomous compliant behavior in snake robots, enabling adaptation to environmental changes like varying pipe diameters during climbing. Utilizing a low-dimensional gait framework for closed-loop control simplification, an extended Kalman filter (EKF) estimates gait parameters representing the robot's shape. Control parameters are chosen relative to this state, enabling generation of compliant behavior despite stiff gear ratios and absence of mechanical compliance or torque sensing in the robot's joints. This approach facilitates precise whole-body motion control, enhancing the robot's adaptability and maneuverability in dynamic environments.

Compared to other kinds of robot, snake robot is capable of moving through complex unstructured environments with the help of its special body structure gives it many degrees-of-freedom(DOFs). In order to reduce the computational cost of coordinating these high DOFs, a so called shape-based compliant control has been proposed [81]. This method utilizes shape functions to determine the coupling between degrees of freedom, enabling intuitive adaptation of robot shape through joint-level torque feedback control. Focusing on compliance in spatial frequency and temporal phase parameters of snake-like robot kinematics, researchers explore how varying spatial frequency influences degree of centralization. Experimental findings demonstrate shape-based control's superiority over central pattern generator-based methods in facilitating effective locomotion, offering potential for enhancing agility and efficiency in challenging terrains. In [82], a biologically-inspired strategy called directional compliance is proposed. The snake robot is able to adjust stiffness to conform or resist terrain. A dynamical system switches locomotion modes to handle obstacles, ensuring reliable traversal of planar peg arrays and three-dimensional rock piles. This approach empowers snake robots to navigate complex environments with precision and adaptability.

Recently, model predictive control(MPC) is utilized to control each joint of a snake robot in [15]. MPC is a control method that solves a finite-horizon optimal control problem at every sampling time instance and applies the first part of the optimal input to the real dynamic system. Different from classical PID control, no reference is needed which means that the snake robot is able to change its locomotion depends on different situations. Simulation results showed that the proposed control method maximized the robot's forward velocity and integrates the choice of the gait pattern into the closed loop. The snake robot generates a standard lateral undulation without reference while achieving constraint satisfaction. A simplified model which represents revolute joint with prismatic joint is utilized as the control model for MPC. Economic model predictive control(EMPC) is also applied to snake robot locomotion in [73]. The complex dynamic model of snake robot moving on land is set as the control model. A compliant contact model is also introduced into the system. Numerical simulation results show that the generated gait patterns are undulatory and can make use of anisotropic ground friction and obstacles.

The work in [83] introduces a novel simplified model tailored for snake robot control, applied within a path-following framework using model predictive control (MPC). Unlike previous models, joint angles are excluded via averaging, rendering it compatible with MPC, which explicitly considers inequality constraints. This model, derived without parameter identification or assumptions of straight-line movements, ensures simplicity and practicality. Alongside constraints on joint angles and motion frequency, limitations on variable change rates are imposed in MPC design. Additionally, a soft constraint mitigates approximation errors. Through extensive simulations and experiments, the effectiveness of the proposed control system, affirming its robust performance in real-world scenarios are validated.

Different from the gradient-based MPC, sample-based MPC does not require gradient information so that discontinuities can be incorporated directly into the MPC prediction steps. Monte Carlo model predictive control (MCMPC), a type of sample-based MPC, is utilised for the underwater snake robot in this study. One benefits of MCMPC is that discontinuous phenomena can be directly introduced into the system dynamics during prediction steps. This gives the snake robot ability to move through some complex environments with several constraints and make contact with the surroundings to gain propulsion forces. However, setting each joint as the control input for MCMPC extremely increase the computation costs. Thus, another control method, curvature derivative control, is used to control the rest of joints while the head joint is controlled by MCMPC. These control methods are introduced in detail below.

3.2 Model Predictive Control

Model Predictive Control (MPC), also known as receding horizon control, operates by repeatedly solving an optimization problem over a finite prediction horizon, thereby predicting future system behavior and determining optimal control inputs. This predictive nature distinguishes MPC from other control methods, allowing it to anticipate changes in the system and adjust control actions according to the value of the cost function. A sample of the procedure is shown as Fig. 3.1:

In MPC, a mathematical model of the system dynamics is utilized to predict future states based on current measurements and past inputs. The discret nonlinear model

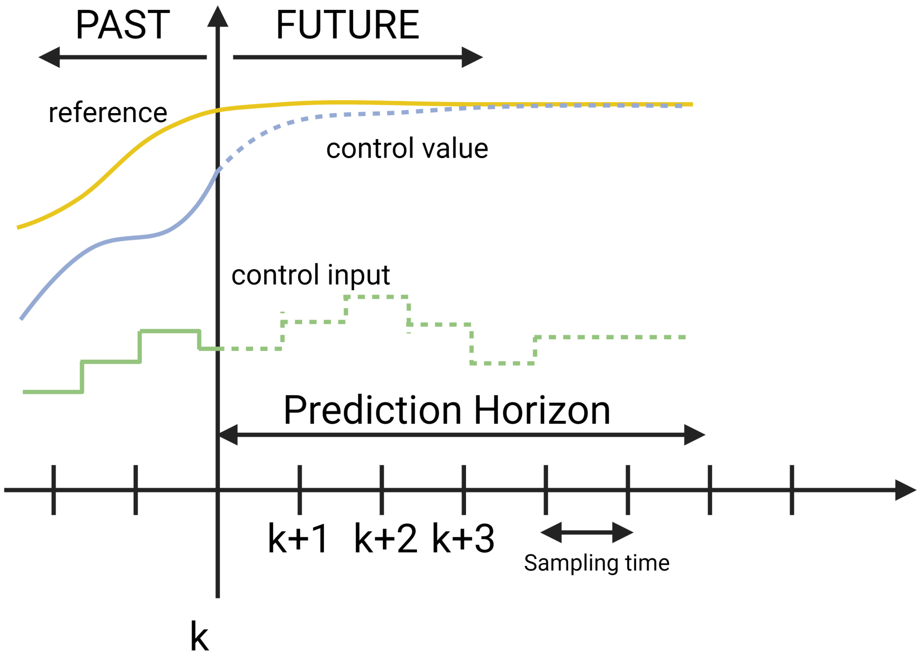


Figure 3.1: Procedure of model predictive control

can be shown as:

$$x_{k+1} = x_k + f(x_k, u_k)\Delta t \tag{3.1}$$

where k is the control time and and Δt represents the sampling time for the MPC, $f(x_k, u_k)$ shows the state dynamic of the control target model, u_k is the control input and x_K shows the current states.

By optimizing a cost function J that quantifies deviations from desired set points and control effort, MPC computes optimal control inputs that steer the system towards desired objectives while adhering to constraints on states and inputs. The cost function

J can be shown as follows,

$$
\begin{aligned}
\min_{x,u} \quad & \sum_{k=0}^{N-1} [(x_k - x_{ref})^T Q (x_k - x_{ref}) + (u_k - u_{ref})^T R (u_k - u_{ref})], \\
\textbf{s.t.} \quad & x_0 = x_{in}, \\
& x_{k+1} = f(x_k, u_k), \quad k = 0, \ldots, N-1, \\
& x_{lower} \leq x_k \leq x_{upper}, \quad k = 0, \ldots, N-1, \\
& u_{lower} \leq u_k \leq u_{upper}, \quad k = 0, \ldots, N-1,
\end{aligned}
\tag{3.2}
$$

where N represents the prediction horizon for MPC, Q and R are the weight matrices, x_{ref} and u_{ref} are the references for both the state variables and the control inputs so that the cost function is able to minimize the differences between these variables. x_{lower} and u_{lower} are the lower limits for the constraints of states and inputs while x_{upper} and u_{upper} are the upper limits.

MPC offers several advantages, including robustness to disturbances and uncertainties, explicit consideration of constraints, and flexibility in handling nonlinear, multivariable systems. These features make MPC well-suited for a wide range of applications, including process control, automotive control, robotics, and renewable energy systems. Despite its computational complexity, MPC continues to be a widely-used and effective control strategy in modern engineering practice.

3.2.1 Gradient-based Model Predictive Control

Gradient-based Model Predictive Control (MPC) is a variant of MPC that relies on gradient-based optimization techniques to solve the underlying optimization problem. In gradient-based MPC, the control inputs are adjusted iteratively in the direction that minimizes a predefined cost function while satisfying system dynamics and constraints. The optimization problem is typically formulated as a nonlinear programming (NLP) problem, where the cost function and constraints are represented as smooth, differentiable functions of the control inputs and system states.

One of the key advantages of gradient-based MPC is its computational efficiency, particularly for problems with relatively small dimensions and smooth cost functions. By leveraging gradient information, gradient-based MPC algorithms can rapidly converge to a solution, making them well-suited for real-time control applications with stringent time constraints. This efficiency is especially beneficial in embedded control systems, such as automotive engine control, aircraft flight control, and robotic manipulator control, where fast response times are crucial.

However, gradient-based MPC may face challenges in handling non-smooth cost functions, discontinuous constraints, or systems with high-dimensional state and control spaces. Additionally, gradient-based methods may struggle with local minima and numerical stability issues, particularly in highly nonlinear or ill-conditioned optimization problems. Despite these limitations, gradient-based MPC remains a powerful and widely-used control strategy in engineering applications, offering a balance between computational efficiency and control performance.

3.2.2 Sample-based Model Predictive Control

Sample-based Model Predictive Control (MPC) is an alternative approach to solving the optimization problem in MPC, which employs sampling-based techniques instead of gradient-based optimization methods. In sample-based MPC, the solution space is explored through random or pseudo-random sampling of control inputs, and the performance of each sample is evaluated based on the cost function and constraints [84]. This sampling process allows for the consideration of non-smooth cost functions, nonlinear dynamics, and complex constraints that may be challenging to represent analytically or differentiate.

One of the main advantages of sample-based MPC is its robustness to uncertainties and disturbances, as it does not rely on explicit knowledge of system gradients or analytical models [85]. Instead, sample-based MPC generates multiple candidate solutions and selects the most promising one based on performance metrics. This robustness makes sample-based MPC well-suited for applications where the system dynamics are uncertain or where disturbances are prevalent, such as in renewable energy systems, autonomous vehicles, and industrial process control.

However, sample-based MPC typically requires more computational resources compared to gradient-based methods, as it involves generating and evaluating a large number of samples to ensure adequate coverage of the solution space. Additionally, sample-based MPC may exhibit slower convergence and higher computational overhead, particularly for high-dimensional problems or problems with tight constraints. Despite these challenges, sample-based MPC offers a flexible and reliable control strategy for addressing complex and uncertain systems in practical engineering applications.

Thus, in order to come over the disadvantage of sample-based MPC which is to slow down the computational cost and also takes the advantage of being able to deal with the uncertainties such as discontinuous situations, Monte Carlo model predictive control is utilized in this study. With the help of GPU parallel acceleration, MCMPC is able to choose the best input in a very short time. The details are introduced as below.

3.3 Monte Carlo Model Predictive Control

The sample-based model predictive control been utilized in this study is called Monte Carlo Model Predictive Control (MCMPC). The procedure of MCMPC is proposed in detail in [86]. This kind of sample-based MPC takes the advantage of GPU parallel computing and is able to deal with huge amounts of threads in the same time. MCMPC has the main advantage is that it can deal with discontinuous situations such as collisions while decrease the computational expense through GPU parallel computing. Thus this control method is employed to manage contacts for various types of robotics, including collisions for the inverted pendulum swing up control on a cart in [87]. MCMPC is also used to control the quadcopter while considering collision with wall [88] and the trajectory generation of CM for Biped Robot taking contact with walls and constraints of ZMP into account [89]. There are four main procedures of MCMPC and the detail is shown as Fig. 3.2,

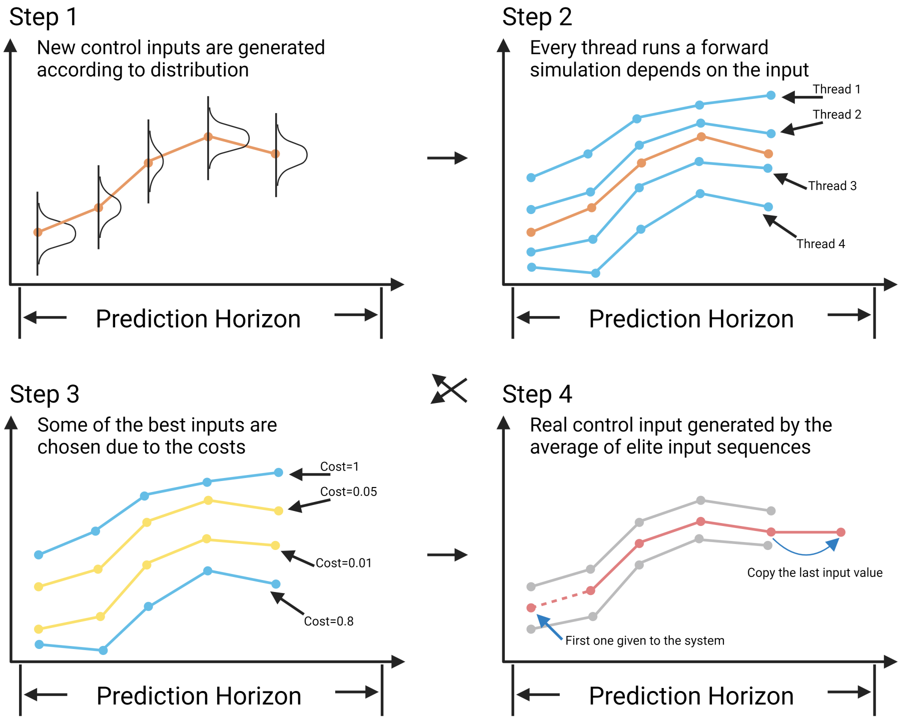

Step 1
New control inputs are generated according to distribution
Prediction Horizon
Step 2
Every thread runs a forward simulation depends on the input
Thread 1
Thread 2
Thread 3
Thread 4
Prediction Horizon
Step 3
Some of the best inputs are chosen due to the costs
Cost=1
Cost=0.05
Cost=0.01
Cost=0.8
Prediction Horizon
Step 4
Real control input generated by the average of elite input sequences
Copy the last input value
First one given to the system
Prediction Horizon

Figure 3.2: The main procedures of MCMPC: Random input sequences are generated based on the weighted mean(the orange line) from previous step; Every thread(blue lines) runs a forward simulation based on the input sequences; Elite samples are chosen depend on the cost value(yellow lines are threads with small costs while blue ones are threads with high costs); The weighted mean(red line) is then calculated based on the elite threads(grey lines).

3.3.1 Control Inputs Generation

At the outset of each control cycle, Monte Carlo random sampling is utilised to generate the control input sequences based on weighted mean(orange line) of previous step. Generally speaking, the input sequence generated at each control loop can be represented as follows,

$$u_k^n = \{u_{k|1}^n, u_{k|2}^n, \cdots, u_{k|i}^n, \cdots, u_{k|N}^n\} \tag{3.3}$$

where N is prediction horizon, n represents the index of threads, k is the number of control cycle of MCMPC which means the time, and i is the prediction step. The normal distribution $\mathcal{N}$ of Monte Carlo sampling is generated with weighted mean u_{k-1}^* and standard deviation σ as,

$$u_{\{k|i\}}^n \sim \mathcal{N}(u_{\{k-1|i+1\}}^*, \sigma^2) \tag{3.4}$$

In this study, the value of standard deviation is held constant. However, it may be subject to variation if the increase of dynamic uncertainty is taken into account. So that the proposed method is able to deal with the changing environments. The next step will be the execution of the forward simulations depend on the calculated input sequences from the previous step.

3.3.2 Execution of Forward Simulations

As shown in Fig. 3.2, every individual thread(blue lines) in the parallel computing framework executes a forward simulation of the dynamic system utilising the control inputs that were produced in the preceding phase. Following the generation of the specified number of input sequences, forward simulation of each sequence is executed concurrently on each thread of GPU parallel computing during the sampling period. The simulation is based on the state equation of the controlled system as,

$$x_{\{k|i\}}^n = f(x_{\{k|i-1\}}^n, u_{\{k|i\}}^n) \tag{3.5}$$

where x is the state variables. The designed cost function is then calculated using these state variables in order to determine which elites have the lowest cost value.

3.3.3 Selection of Elite Threads

Following the calculation of the cost value of each thread, the elite samples(yellow lines) are selected according to the cost values as shown in Fig. 3.2. The threads with the smaller values are chosen as the elite samples.

Once all forward simulations have been completed, the elite samples are selected according to a certain rate. These samples are then utilised in the computation of the weighted average of the actual input sequence.

$$u^*_{\{k|i\}} = \frac{\sum_{q=1}^{Q_t} u^q_{\{k|i\}} e^{-\frac{L^q_k}{\lambda}}}{\sum_{q=1}^{Q_t} e^{-\frac{L^q_k}{\lambda}}} \tag{3.6}$$

where L^q_k represents value of the cost function of q-th thread among elite threads during k-th control cycle, Q_t is the number of chosen threads, q is the q-th thread among elite threads and λ is constant parameter, if $\lambda \rightarrow 0$, estimated optimal input matches real optimal input based on Pincus theorem [90].

3.3.4 Inherit of Generated Control Inputs

Finally, depending on the input sequences of these elite samples(grey lines), the weighted mean(red line) is utilized for the next control loop. The first step of optimal input sequence is applied to the real system. Therefore, optimal control laws for the prediction of next control cycle are described as,

$$u^*_{\{k|2\}}, u^*_{\{k|3\}}, \cdots, u^*_{\{k|N\}}, u^*_{\{k|N\}} \tag{3.7}$$

The N+1 mean value for the prediction of next cycle is not estimated and it is expected that control inputs are usually continuous. Thus, it is simply coped as the same value as the Nth optimal control law.

3.3.5 Constraints Setting of MCMPC

One of the most important features of MPC is that constraints can be set and obtained. Therefore, constraints are established for both state variables and control inputs for the design of MCMPC. Once the input is generated by the Monte Carlo random sampling, it will then be saturated according the specific constraints as,

$$u = \begin{cases} u_{min} & u < u_{min}, \\ u & u_{min} \leq u \leq u_{max}, \\ u_{max} & u > u_{max}, \end{cases} \tag{3.8}$$

where u_{min} and u_{max} are the lower and upper limits of the control inputs. So the generated inputs will be limited to the set range of values. In this study, two variables are set as the control inputs to be generated by MCMPC. One variable is chosen as the angle of the head joint and the other one is designed as the passing speed of the curvature which will be introduced in the next section.

For the head joint, the random inputs generated by MCMPC will be evaluated and set within a specified range to satisfy the input constraint. This constraint is straightforward to establish, as the generated value will be replaced with the constraint value when it surpasses the up and down limitations. As for the state constraints, in the event that the value of state deviates significantly from the specified limits of MCMPC, a substantial value will be appended to the cost. This augmentation serves the purpose of excluding threads that surpass the limit from the elite sample selection process.

In order to be able to better guarantee the wide range of random inputs generated and in order to cope with any random possible situation, the number of samples for Monte Carlo sampling should be large enough for each input which means that the calculation for MCMPC will be complex. Thousands of samples are required for the designed control inputs which highly increase the complexity for parallel computation and slow down the efficiency of the proposed control method. Therefore, instead of using each joint as the control inputs, only the head joint is chosen as the control input to generate locomotion of the entire snake body. This is realized with the help of Curvature Derivative Control(CDC), and the details of CDC are introduced as the follow section.

3.4 Curvature Derivative Control

Following the generation of the head joint input by MCMPC, the other joints are controlled using Curvature Derivative Control (CDC). CDC is a type of control method that determines the ideal moment of bending while taking into account energy consumption of the associated joint torques. This method is proposed and utilized for the locomotion of snake robot in [91, 18]. The joint torque is calculated according to the following equation:

$$\tau^*(s) = -\frac{m\alpha}{\int_0^{L_{cdc}} \langle \kappa'(s), \kappa'(s) \rangle \, ds} \kappa'(s) \tag{3.9}$$

where $\kappa'(s) \in \mathbb{R}$ signifies the curvatures with arclength s, $\tau^*(s)$ represents the bending moment distribution, α denotes the longitudinal acceleration, and L_{cdc} is the length of body curve. Following this, the articulated model is derived from the continuum model, with the joint torque replacing the bending moment and the curvature being substituted with the joint angle. The calculation of control inputs for joints, excluding the head joint, is as follows:

$$\tau_i = k_c(\mathbf{A}\phi_{i+1} - \phi_i) - k_{cd}\dot{\phi}_i \tag{3.10}$$

where k_c and k_{cd} are control gains with given constant values, ϕ_i and $\dot{\phi}_i$ represent the related joint angle and the joint velocity. The parameter $\mathbf{A}$ is set so that the amplitude can be justified when passing through the entire body. This gives the robot more flexibility to face the various environments and generate proper locomotion depends on different situations.

By employing this control method, the external environment's effect on one link can be transferred to the adjacent link, thereby adapting the snake's body to the varied surroundings. In addition to merely transmitting the joint angle to subsequent joints, propulsion forces that operate on preceding components can also be transferred to the succeeding link. It is important to note, however, that this method of control operates under the assumption that the snake robot body does not experience any side slip. Such conditions can be maintained for an underwater environment. In [18], Fig. 10 in the paper illustrates the relationship between epaxial muscle activity and curvature, as observed in the behaviours of an actual snake.

Furthermore, it is demonstrated that advanced locomotion can be generated using CDC even in the absence of knowledge regarding the wall's geometry. This implies that snake robot consistently selects the most advantageous portion of contact area to facilitate its forward motion. This phenomenon is also observed when the robot is in motion within a restricted environment.

3.5 Computational Efficiency of Parallel Computing

As mentioned in the previous chapters and also in the introduction, one advantage of the proposed method is to decrease the computational cost for MPC. The complex dynamics of the snake model also effects a lot on the procedure of MPC due to the forward execution during the prediction step. In order to show the efficiency of the proposed method, several simulations have been tested based on different situations.

Setups of the simulation environment are introduced also with parameters for MCMPC. Simulation results are shown under different initial conditions and analyzed at the end of this section. The control method is implemented and tested in the environments built on the laptop with 11th Gen Intel(R) Core(TM) i7-11800H CPU and NVIDIA GeForce RTX 3060 Laptop GPU. Two different programming environments have also been designed for the relevant tests, and the first one is built in MATLAB2021b. $PARFOR$ function is utilized to calculate the forward simulations of MCMPC. This function uses

Table 3.1: Parameters for MCMPC

Name	Meaning
Prediction Horizon	Time interval of the prediction of MCMPC
Prediction Steps	Number of predictions for one horizon
Number of Samples	Total samples for random sampling of MCMPC
Number of Iterations	Times of repetition for each control loop
σ^2	Variance of normal distribution for input sequence generation
Input Constraints	Constraints for the inputs generated by MCMPC

the total eight-core laptop CPU to execute the parallel computation [76]. The second simulation environment is built in CUDA with the help of GPU based on C programming [74].

3.5.1 Experimental Setups for MCMPC

There are several parameters need to be determined for MCMPC and CDC and will be introduced. As mentioned in previous sections of this Chapter, constraints for both the control inputs and the state variables need to be defined. The parameters of the MCMPC are shown in Table 4.2.

The snake robot consists of 10 links with total link length $l_{all} = 1.4m$, total mass $m_{all} = 1kg$, and each link has the same length and mass. Parameters for hydrodynamics are set as $\rho = 1000\ kg/m^3$, $C_M = 0.5$, $C_f = 0.03$, $C_A = 1$ and $C_D = 2$. Current values are established in accordance with the experimental conditions. The top 1% of random inputs are chosen as the elite threads. For the experiments running in CUDA, the prediction horizon is set to $N = 100$ with a sampling time of $T_s = 0.1$ s. Constraints of MCMPC for control input and individual joint angles are denoted as $|\phi_i| \leq 1.04$. Constraint for CDC is $|u_i| \leq 0.5$ with $K_c = 0.5$ and $K_{cd} = 1.5$, respectively. As for the simulations in MATLAB, the top 1% of random inputs will be chosen as the elite threads. The prediction horizon is set to $N = 100$ with a sampling time of $T_s = 0.1$ s. Constraints of MCMPC for control input and joint velocities are identified as $|\phi_i| \leq 0.52$

and $|v_{\phi_i}| \leq 0.6$. Constraint for CDC is $|u| \leq 0.74$ with $K_c = 0.5$ and $K_{cd} = 1.5$, respectively.

3.5.2 Analysis of the Difference in Computation Time Consumed by Each Control Method

In the research of [76, 73], the computational time for either MCMPC in MATLAB or Economic MPC required a substantial amount of time to calculate. As mentioned in [73], the computation of individual optimal inputs in the simulations requires a time interval of 5 to 200 seconds. The desktop computer executes the simulation using C code generated on a virtual machine. The main difference between the dynamic model of the snake robot and the one investigated in this work is that the former operates on land, whereas the latter operates underwater. As mentioned in [92], the dynamic model between these two types of snakes is not so much different. The dynamics of underwater snake can be easily changed to the snake moving on land by setting the hydrodynamics as zeros and replace the friction forces with the land frictions. Thus, there is not much difference between the complexity of two models. Additionally, it is noted that no attempt is made to decrease the computation time in [73]. In one of the work [76] realizing MCMPC and CDC for the obstacle-aided locomotion for the underwater snake robot. The computation of a single input sequence requires between 30 and 60 seconds and is performed in MATLAB 2021b utilizing CPU parallel computing. The program operates on the identical laptop as this study.

GPU parallel computing is accomplished with CUDA, the calculation time for each control cycle of each input sequence is displayed in Table 3.2. The calculation time is significantly impacted by the number of samples, as it determines the complexity of MCMPC. By utilizing GPU parallel computation, each control input sequence is generated within the control cycle of 0.1s. The time required decreases as the number of threads decreases. However, it is evident from the results that as the number of samples increases, so does the success rate of obtaining complete performance for the

Table 3.2: Computation time for each control method.

Control Method	Time consumed for each optimal input
Economic MPC	between 5 and 200 seconds
MCMPC(CPU parallel computing)	between 30 and 60 seconds(4000 threads)
MCMPC(GPU parallel computing)	
8640 Threads	Average 0.098 seconds
5760 Threads	Average 0.056 seconds
2880 Threads	Average 0.033 seconds

locomotion of the snake robot. The 2880-thread based snake robot might be unable to reach the intended location.

3.6 Chapter Summary

In this chapter, the control methods used for the locomotion control of the underwater snake robot have been introduced. First, different control methods used by researchers to control the snake robot are shown. Various control methods optimize snake robots' locomotion in complex environments. PID control facilitates lateral undulation and eel-like motion, applied in terrestrial and underwater scenarios. Autonomous compliant behavior adapts to environmental changes via a low-dimensional gait framework and extended Kalman filter. Shape-based compliant control coordinates high degrees of freedom intuitively through joint-level torque feedback. Directional compliance adjusts stiffness for terrain conformity. Model predictive control (MPC) enables adaptive locomotion without references. Economic MPC enhances undulatory gait patterns leveraging ground friction and obstacles. Sample-based MPC like MCMPC incorporates discontinuities directly into system dynamics for complex environment navigation. Curvature derivative control balances computation costs alongside MCMPC, ensuring effective joint control.

The introductions of basic MPC is then given. The main control methods consist of MCMPC and CDC are introduced in detail. MCMPC is mainly made up of four steps as shown in Fig. 3.2 as:

1. At the outset of each control cycle, Monte Carlo random sampling is utilised to generate the control input sequences based on weighted mean(orange line) of previous step.

2. Every individual thread(blue lines) in the parallel computing framework executes a forward simulation of the dynamic system utilising the control inputs that were produced in the preceding phase.

3. Following the calculation of the cost value of each thread, the elite samples(yellow lines) are selected according to the cost.

4. Depending on the input sequences of these elite samples(grey lines), the weighted mean(red line) is utilized for the next control loop.

and Curvature Derivative Control (CDC) is utilized for controlling snake robot joints following MCMPC-generated head joint inputs. CDC determines optimal bending moments while considering joint torque energy consumption. The control method enables adaptation to varying environments by transferring effects between adjacent links, transmitting joint angles and propulsion forces. CDC facilitates advanced locomotion even without knowledge of wall geometry, allowing the robot to select optimal contact areas for forward motion. This capability is observed in both unrestricted and confined environments, as demonstrated in real snake behavior.

Finally, the efficiency of the proposed method is shown based on the simulation results of different simulation environments. The computational cost is significantly reduced by implementing both MCMPC and CDC. The simulation results of various environment from open spaces to environments with constraints will be shown in the next chapter.

Chapter 4

Performances of Underwater Snake Robot in Various Environments

Chapter 2 of the study delves into the intricacies of the underwater snake robot model, crucial for understanding its behavior and performance in various aquatic environments. This chapter introduces the hydrodynamic aspects incorporated into the system dynamics, including linear and nonlinear drag forces, added mass force, and fluid torques. These additions enhance the realism of the model, reflecting the complexities of underwater movement and interaction with the surrounding fluid medium.

The underwater snake robot's unique design with high degrees of freedom offers advantages over traditional underwater vehicles like Autonomous Underwater Vehicles (AUVs) and Remotely Operated Vehicles (ROVs). Its flexibility enables efficient and adaptive navigation through underwater terrains, showcasing the versatility derived from its ability to perform various gait patterns in response to different surroundings. By outlining the model in Chapter 2, the groundwork is laid for subsequent chapters to explore the control methods tailored to exploit these capabilities.

In Chapter 3, the focus shifts to the proposed control methods designed to optimize the underwater snake robot's performance. Two key approaches are introduced: Monte Carlo Model Predictive Control (MCMPC) and Curvature Derivative Control (CDC).

MCMPC stands out for its ability to handle discontinuous situations during prediction steps while maintaining computational efficiency. This capability is particularly valuable in navigating complex underwater environments with obstacles and varying constraints. The combination of MCMPC and CDC offers a comprehensive control strategy that leverages the robot's high degrees of freedom for efficient and adaptive locomotion.

Chapter 4 serves as the testing ground for the proposed control methods, aiming to demonstrate the benefits of utilizing MCMPC and CDC for controlling the underwater snake robot. Through simulation results, the performance of the robot is assessed across various environments, ranging from open spaces to environments with constraints. Initially, the focus lies on the robot's movement in unobstructed settings, providing a baseline for comparison.

The simulations in Chapter 4 showcase the underwater snake robot's adaptability and agility, highlighting its ability to navigate complex environments with precision. By leveraging MCMPC and CDC, the robot demonstrates efficient locomotion and effective response to environmental challenges. The integration of these control methods enables seamless transition between different gaits and facilitates obstacle avoidance in confined spaces.

Overall, Chapter 4 underscores the practical applicability of the proposed control methods in enhancing the performance of underwater snake robots. Through rigorous testing and analysis of simulation results, the advantages of utilizing MCMPC and CDC become evident, positioning the robot as a viable solution for underwater exploration and intervention tasks in diverse aquatic environments.

4.1 Locomotion of Snake Robot Moving in Subsea Environments

In open space, the underwater snake robot showcases its remarkable adaptability and efficiency in locomotion. Its design, inspired by the locomotion of serpents, grants it unparalleled flexibility and maneuverability in aquatic environments. By exploiting its high degrees of freedom and employing sophisticated control algorithms, the robot navigates through open water with precision and grace.

Utilizing a combination of Monte Carlo Model Predictive Control (MCMPC) and Curvature Derivative Control (CDC), the robot achieves efficient locomotion while maintaining stability and control. MCMPC enables the robot to anticipate and respond to dynamic changes in its surroundings, adjusting its trajectory in fast response to optimize efficiency. Meanwhile, CDC ensures smooth and coordinated movement of the robot's joints, allowing it to seamlessly transition between different gaits and adapt to varying conditions.

Through simulation and experimentation, the underwater snake robot demonstrates its capability to generate efficient locomotion in open space. By harnessing its unique body structure and leveraging advanced control strategies, the robot exemplifies the potential for agile and adaptable underwater exploration.

4.1.1 Simulation Setups of Proposed Control Methods

The setups for the designed control method such as the value of control parameters and the make up of cost function will be shown next.

The underwater snake robot was considered to have $n = 10$ links, each link has it length of $l = 0.14m$ with mass $m = 1kg$. Parameters for hydrodynamics given by [9] were set as $\rho = 1000\ kg/m^3$ $C_M = 0.5$, $C_f = 0.03$, $C_A = 1$ and $C_D = 2$. As mentioned before, current value was set to $[0.0, 0.0]$ which would not be considered in the following experiments. In this simulation, control parameters for curvature derivative control was set as $K_c = 5$ and $K_{cd} = 1.5$, respectively.

As for the design of the cost function of MCMPC, the difference between the position of center of mass of snake robot and the goal is used. Snake robot moves towards the goal because the smallest value of cost function is seen as good results and related control input sequences are inherited. The cost function is shown as

$$L = \sum_{i=1}^{N} x_{\{k|i\}}^T Q x_{\{k|i\}} \tag{4.1}$$

$$x_{\{k|i\}} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} - \begin{bmatrix} D_x \\ D_y \end{bmatrix} \tag{4.2}$$

where D_x and D_y are the coordinate of the destination we want the snake robot to move to. Q is the weight matrix for the control state as shown in

$$Q = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix} \tag{4.3}$$

Table 4.1 shows the parameters used for MCMPC, 2000 number of sample threads are randomly generated and top 10% will be chosen as the elite threads for next control cycle. Constraints for both joint angle and joint velocity are identified. As for curvature derivative control, the sampling time is set to $0.001s$.

Table 4.1: Parameters for MCMPC

Prediction horizon	5s
Simulation time	20s
Prediction step	50 step
Sampling time	0.1s
Number of samples	2000
Input constraint	$\lvert u \rvert \leq 0.52$ rad
State constraint	$\lvert v_\theta \rvert \leq 0.6$ rad/s

Due to the background that the snake robot is assumed to only move in the open environment where there is no other objects such as obstacles or pipes to enter, so the contact dynamics are not introduced into the system for the following simulations in this section.

The proposed control method was implemented and verified through simulations via Matlab 2021b by using the parallel accelerate function called $Parfor$. The parameters for the snake robot and fluid dynamic are set as shown above and the the same as [9], [10]. In the experiments, Snake robot is set to have the same physical parameters with also state and input constraints as given in Table.4.1.

4.1.2 Locomotion in Open Space

The previous researches in [15] and [73] show that the snake robots can generate efficient locomotion without a given reference. The robots are able to perform different and adaptive gait patterns depend on the changing environment.

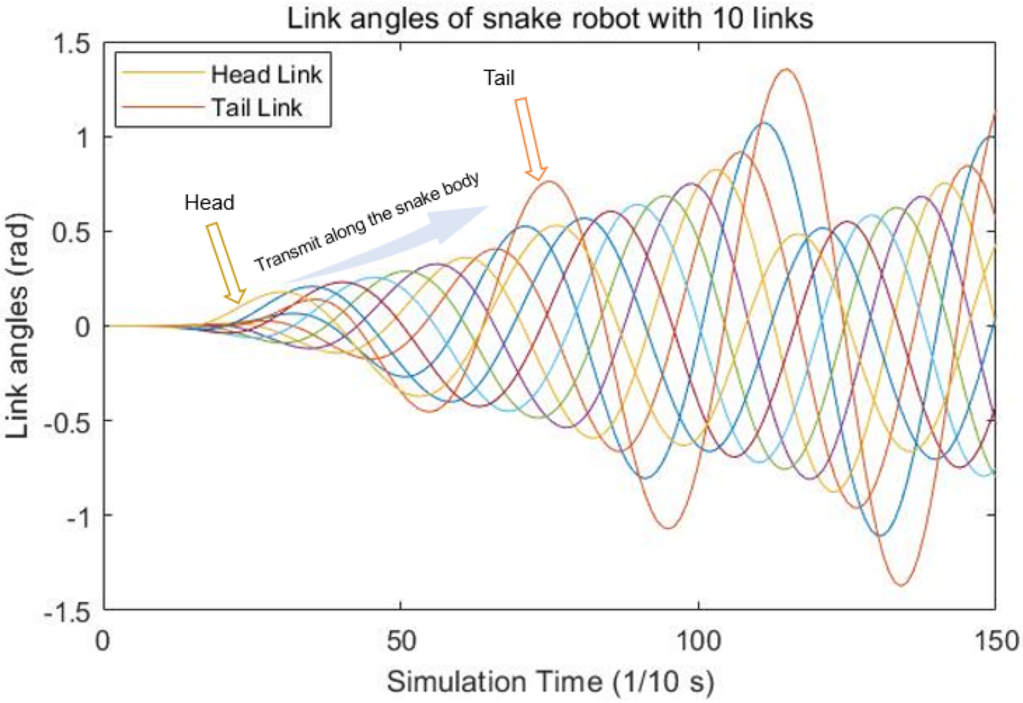


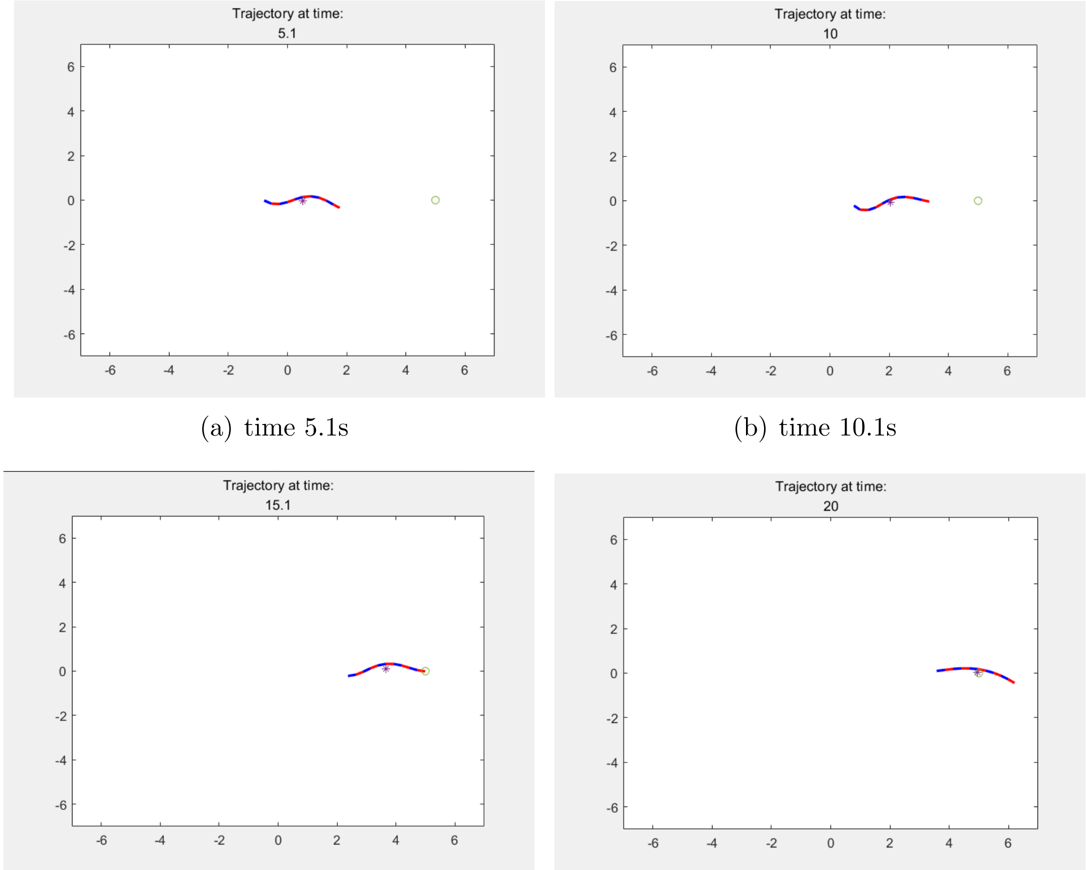
Figure 4.1: Total link angles of a 10 links snake robot

In the first experiment, we simply test the desired method for an underwater snake robot with 10 links. Fig. 4.1 shows the related result of the curves of the total link angles for the entire snake robot body. The figure illustrates how Monte Carlo Model Predictive Control (MCMPC) facilitates efficient lateral locomotion in the underwater snake robot without requiring a predefined gait pattern. By treating the transform speed of curvature as a variable, Curvature Derivative Control (CDC) allows the robot to dynamically adjust its motion patterns. This flexibility enables the robot to seamlessly adapt to diverse environmental changes, such as ocean currents encountered during underwater navigation.

In this study, the magnifying coefficient remains a constant parameter. Notably, the amplified link angle from head to tail is evident, showcasing the robot's ability to traverse through the water with fluidity. However, it's worth noting that the magnifying coefficient could also serve as a control input for MCMPC. By incorporating this additional parameter, the robot could potentially achieve even more efficient locomotion, including backward movement if necessary. This adaptive capability ensures the robot can adeptly respond to a wide range of environmental conditions, enhancing its versatility and effectiveness in underwater exploration and maneuvering.

In order to show the advantage of curvature derivative control that the proposed control method is suitable for snake robots with different number of links, the snake robots are tested with both 10 and 20 links. We set initial time as $t_0 = 0s$ and the original state of snake robot as $\phi(0) = [0, 0, 0, 0, 0, 0, 0, 0, 0]^T$rad, $v_\phi(0) = 0$ rad/s, $v_t(0) = 0$ m/s and $v_n(0) = 0$ m/s. The standard deviation for MCMPC is set as a constant with the value of 0.1.

Fig. 4.2 shows the behaviors of 10 links snake robot runs for 20 seconds where the circle point repents the destination, the color of snake body in red or blue shows the odd and even links from head to tail. The snake robot is put in the original point as the initial start position. Snake body generates such kind of lateral locomotion smoothly when approaching the destination as shown in the pictures. The cost function is made up of the difference between the robot and the point, so the robot changes its gait pattern during the experiment. The snake also stays in the same pose when it is approaching the final point and finally stops its locomotion when the coordinate of



(a) time 5.1s

(b) time 10.1s

(c) time 15s

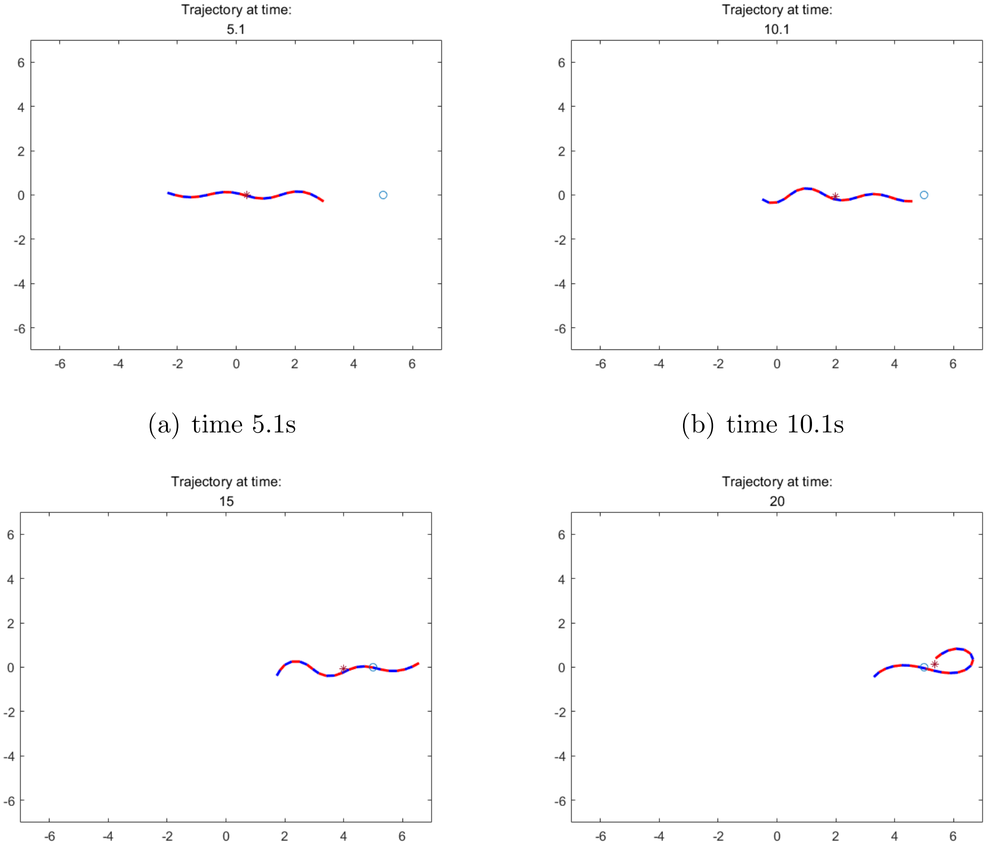
(d) time 20s

Figure 4.2: Results of 10 links snake robot movements in 20s

center of mass is almost the same as the goal.

Another experiment is tested for snake robot with 20 links. We only change the number of links, other parameters for MCMPC, CDC and dynamic model remain the same as 10 links. Fig. 4.3 are the results of related experiment. During the first part of this result, the snake robot generates the same lateral undulations which pushes the entire body towards the target point. Unlike the 10 links snake, snake with 20 links seems to turn around its orientation when nearing the destination.

This situation may be caused by the value of passing speed parameter which is the same as last experiment. Even when the center of mass of snake robot is approaching the point, the changing joint angle is still passing through the snake body which may cause a delay for the back part to follow. Tail part of the snake robot still generates pushing force around 15 second, so the head part starts to change its decision of locomotion and turn around in order to minimize the difference in desired cost function.



(a) time 5.1s

(b) time 10.1s

(c) time 15s

(d) time 20s

Figure 4.3: Results of 20 links snake robot movements in 20s

In this work, we proposed the combination of both Monte Carlo model predictive control and curvature derivative control for underwater snake robot locomotion which does not need pre-defined gait pattern to generate efficient locomotion. With the help of this control method, we proved that only one control input is needed to control the whole snake body trajectory. We also show that by setting the difference between position of snake robot and the destination into the cost function, snake robot could move to the given point and change its locomotion independently during the operation.

Only one type of cost function is tested and just head joint angle is studied as the control input. We believe that underwater snake robot could achieve many efficient gait patterns and complete many challenging tasks by choosing different control inputs and the components of cost functions.

4.2 Obstacle-aided Locomotion in Clustered Environments

Snake robots, inspired by the locomotion of their biological counterparts, have demonstrated remarkable capabilities in navigating challenging environments. However, conventional locomotion methods, such as serpentine or sidewinding gaits, may encounter limitations in clustered or cluttered environments characterized by tight spaces, obstacles, and complex terrain. To address this challenge, researchers have explored innovative approaches known as obstacle-aided locomotion, which leverage environmental features to facilitate traversal [16]. This paradigm shift in locomotion strategies opens new possibilities for snake robots to navigate confined spaces, negotiate obstacles, and overcome terrain complexities with enhanced agility and efficiency [4].

The motivation behind obstacle-aided locomotion arises from the inherent limitations of traditional locomotion methods in clustered environments. In densely packed environments with narrow passages and obstacles, snake robots may struggle to execute conventional gaits effectively, leading to inefficiencies and potential entanglement. By contrast, obstacle-aided locomotion capitalizes on the interaction between the robot and its surroundings, exploiting obstacles as anchoring points or support structures to propel forward motion [82]. This approach enables snake robots to push against obstacles, wrap around them, or use them as leverage points, effectively navigating through confined spaces and negotiating complex terrain features [93].

Central to obstacle-aided locomotion is the utilization of frictional forces, contact interactions, and mechanical constraints to facilitate locomotion [4]. By strategically leveraging environmental features, snake robots can conserve energy and minimize the risk of entanglement or obstruction while traversing clustered environments. However, this approach presents several challenges that must be addressed to realize its full potential. Robust perception and planning algorithms are essential to identify suitable obstacles and plan optimal locomotion trajectories. Furthermore, the design of snake robot hardware and control strategies must be carefully tailored to accommodate interaction with environmental features while ensuring stability, resilience, and efficiency.

Obstacle-aided locomotion holds promise for a wide range of applications across various domains. In search and rescue operations, snake robots equipped with obstacle-aided locomotion capabilities can navigate through rubble, debris, and confined spaces with greater ease, improving their effectiveness in locating and assisting survivors in disaster scenarios. In industrial settings, snake robots can explore cluttered environments, such as pipelines, ducts, and machinery, for inspection, maintenance, and repair tasks. Similarly, in urban environments, snake robots can be deployed for surveillance missions, traversing complex terrain features and navigating through densely populated areas with agility and precision.

In this section, the proposed control methods are tested for the control of an underwater snake robot which is assumed to move among several fixed obstacles. In order to show the benefits of using MCMPC and CDC, no reference is given for the control inputs. And the contact dynamics can be introduced into the system dynamics with the help of MCMPC, the effect of the contact will be passed with the curvature which has been observed through CDC in [18].

In the following subsections, the definition of the obstacles set in the open space environment will be introduced first. How the robot detect the contact between the obstacle and the related link is shown. The influence of the collision and the impact to the system dynamic are explained. Finally, the simulation results of the underwater snake robot are extracted to show the performance and generated obstacle-aided locomotion of the robot.

4.2.1 Collision Phenomenon and Contact Force Calculation

In this study, the dynamics of the underwater snake robot contains both the hydrodynamics and the contacts. A system is called a complementarity system if the flow of the states is constrained by several complementarity conditions. This constraints of the complementarity system is often be described as Linear Complementary problem (LCP). The method is used to calculate the relationship between the contact impacts and the system dynamics. This method asks the two complementary vectors x and y should meet the condition that either one of them should be zero while the other one is non-zero. LCP is often been used by researchers to describe the contact phenomena between two different objects. For two complementary vectors $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$ so that:

$$\begin{aligned} &y = a + AX \\ &x \geq 0, \quad y \geq, \quad x^T y = 0 \end{aligned} \tag{4.4}$$

where $a \in \mathbb{R}^m$ is a given vector and $A \in \mathbb{R}^{m \times m}$ represents a matrix. The solution of the related LCP is unique for a only when the A is a $P-$matrix based on the $Lemke' salgorithm$ shown in [94].

In [16], the impacts between contact links and obstacles are assumed to be inelastic which causes a problem in that nothing prevents the head link from having contact with the obstacle in its tangential direction. Thus, an external term is introduced into the cost function so that the head link will not have contact with any surroundings in this part of study. The punishment is simply set a large value to the cost function which makes it fall in the elite selection process. The dynamic model of the underwater snake robot is then transformed to:

$$M(q)\ddot{q} = f_{all} \tag{4.5}$$

where the corresponding parameters are

$$q = \begin{bmatrix} \theta \\ p_x \\ p_y \end{bmatrix} \in \mathbb{R}^{n+2}$$

$$M(q) = \begin{bmatrix} M_\theta & \begin{matrix} 0_{n\times 1} 0_{n\times 1} \end{matrix} \\ M_p L_P l K^T \begin{bmatrix} S_\theta \\ -C_\theta \end{bmatrix} & \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \end{bmatrix} \in \mathbb{R}^{(n+2)\times(n+2)}$$

$$f_{all} = \begin{bmatrix} -K_x \quad -K_y \\ M_p E^T \end{bmatrix} \begin{bmatrix} f_{Dx} \\ f_{Dy} \end{bmatrix} + \begin{bmatrix} D^T u - RHS_\theta \\ RHS_{p_{cm}} \end{bmatrix}$$

RHS represents the right-hand side pf equations Eq.**??** and Eq.4.5, the definition of matrices can be found in [10] and is not shown here for brevity.

The dynamic of the contact model in the normal acceleration of each contacted link coordinate when the number of the contacted link remains the same is:

$$\overline{C}\ddot{q} + \dot{\overline{C}}\dot{q} \geq 0, \lambda \geq 0, \lambda^T(\overline{C}\ddot{q} + \dot{\overline{C}}\dot{q}) = 0 \tag{4.6}$$

where λ is the vector of Lagrange multipliers and represents the magnitude of the constraint forces and $\overline{C}\ddot{q} + \dot{\overline{C}}\dot{q}$ is the acceleration of each link in the local normal direction. Because the contact force is assumed to be unilateral and completely inelastic which means normal velocity should be zero when λ is nonzero.

When a new link comes into contact with an obstacle, an impact happens. The start and end points of each link are calculated and evenly separated into 20 points along the link. One link is said to have contact with an obstacle if the minimum distance among these 20 points to the center of the obstacle is smaller than the radius of 0.3m. When contact occurs, the distance of the center of the obstacle transformed in the link's local coordinate is used to determine the direction of contact related to the link. Then impact is modeled in as:

$$\overline{C}\dot{q}^{+} = \overline{C}\dot{q}^{-} + \overline{C}M^{-1}\overline{C}^{T}\lambda \tag{4.7}$$

$$\overline{C}\dot{q} \geq 0, \lambda \geq 0, \lambda^{T}\overline{C}\dot{q} = 0 \tag{4.8}$$

where $\overline{C}\dot{q}$ shows the normal direction velocity of each contact link. The propulsion force of each contact link is the $-sin\theta_i$ component of the contact constraint force which represents the contact force along the forward x-axis.

4.2.2 Simulation Performance of Snake Robot Moving among Multiple Obstacles

The proposed control method is implemented in MATLAB2021b. The dynamic model of the underwater snake robot which includes hydrodynamics from the sub-sea environment and the contact influences of the collision between the snake link and the associate obstacles. The experimental setups are shown as follow.

The top 1% of random inputs will be chosen as the elite threads. The prediction horizon is set to N with a sampling time of $T_s = 0.1$ s. Constraints of MCMPC for control input and joint velocities are identified as $|\phi_i| \leq 0.52$ and $|v_{\phi_i}| \leq 0.6$. Constraint for CDC is $|u| \leq 0.74$ with $K_c = 0.5$ and $K_{cd} = 1.5$, respectively. The 2D underwater snake robot consists of 10 links with total link length $l_{all} = 1.4m$, total mass $m_{all} = 1kg$, each link has the same length and mass. Parameters for hydrodynamics are set as $\rho = 1000\ kg/m^3$ $C_M = 0.5$, $C_f = 0.03$, $C_A = 1$ and $C_D = 2$. Current values were set to $[0.0, 0.0]$. Each simulation is set to stop when the distance between the CM of the snake robot and the target is smaller than 0.05 m.

The cost function includes a position error of the CM and the destination, control input, head link punishment, and negative sum of propulsion forces is given by

$$\begin{aligned} L^{n} = \sum_{i=1}^{N} x_{\{k|i\}}^{T} Q x_{\{k|i\}} + u_{\{k|i\}}^{T} R u_{\{k|i\}} + \\ 0.1 l_p - 2 sgn(l_{prop}) l_{prop}^{2} \end{aligned} \tag{4.9}$$

$$x_{\{k|i\}} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} - \begin{bmatrix} D_x \\ D_y \end{bmatrix} \tag{4.10}$$

where $D_x = 6, D_y = 0$ are the destination, l_p is the head link punishment and l_{prop} represents the total propulsion forces of contact links. $Q = \begin{bmatrix} 2 & 0 \\ 0 & 10 \end{bmatrix}$ and $R = 0.5$.

The performances of different variables in cost functions are tested to show how each component affects the behavior of the snake robot. The control variates method is used, and four factors of the cost function are chosen as the experimental subject for MCMPC: number of samples, prediction horizon, the initial condition of the snake robot, and introducing the sum of propulsion forces as a negative part into the cost function.

One advantage of MCMPC is that discontinues can be introduced into the system, allowing the robot to deal with contact with the environment. Therefore, the control method is tested in an environment with three obstacles. The obstacles are assumed to have a circle shape of radius $0.3m$ with each center of the circle is set as $(1.5, -0.3), (2.5, 0.3), (3.5, -0.3)$. Fig. 4.4, Fig. 4.5 and Fig. 4.6 show the related performances of different parameters, all three simulations with the same initial condition. So the beginning state of the snake robot is not shown in Fig. 4.5 and Fig. 4.6. The units of the x-axis and y-axis are meters(m). The small green point represents the target point $(6, 0)$. Blue and red colors are even and odd links of the snake's body, respectively. Two initial poses of snake robots are simulated, condition 1: $x_0 =$ [1.12;0.66;0.2;-0.18;-0.41;-0.45;-0.28;0.07;0.51;0.90] and condition 2: $x_0 =$ [0.6;0.3;0.1;-0.09;-0.2;-0.25;-0.14;0.03;0.25;0.45] and related velocities are set to 0.

First, the performances of different lengths of the prediction horizon are studied. The parameters of MCMPC for two comparison experiments are set with the same number of samples as 4000, cost function, and initial condition. Different prediction horizons 50 and 100 which are related to 5s and 10s of predictable future are tested. In both cases, the snake robot generates various locomotion, takes advantage of pushing itself forward by using obstacles to obtain propulsion, and finally gets near the given

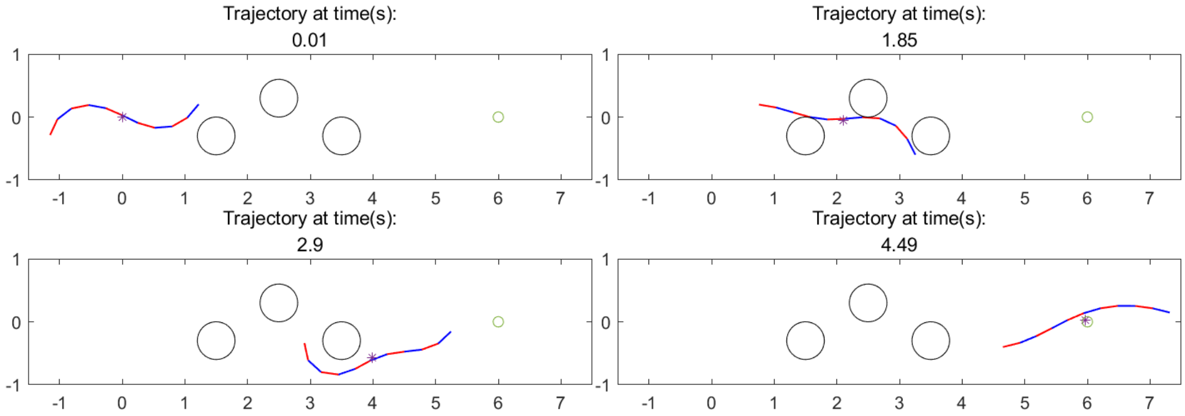


Figure 4.4: Snake locomotion of Initial condition: 1, $N = 100$, 4000 samples, cost function consists of positional error, control input, propulsion forces.

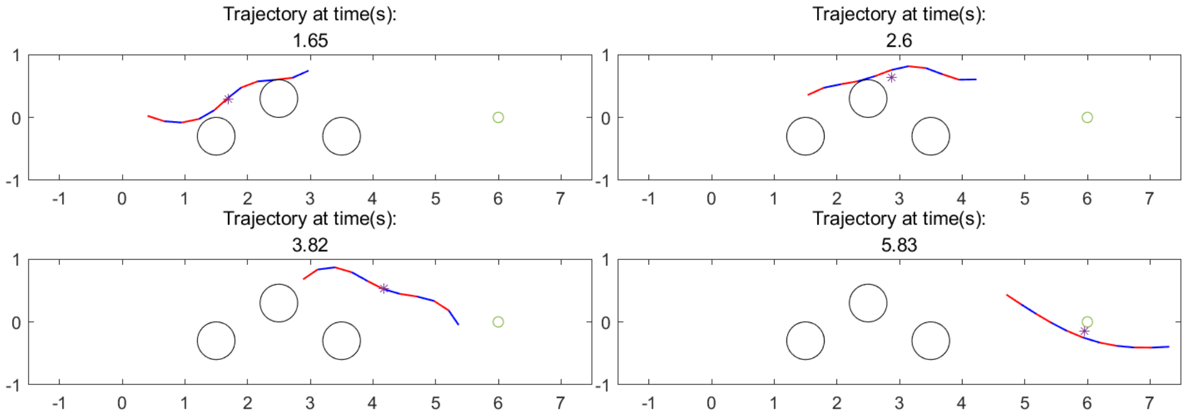


Figure 4.5: Snake locomotion of Initial condition: 1, $N = 50$, 4000 samples, cost function consists of positional error, control input, propulsion forces.

point. However, the snake robot with a 10s prediction period has better performance in terms of shorter overall simulation time and more convenient path selection. Also, in Fig. 4.5, the positional error still exists at the end of the simulation which is well matched in Fig. 4.4. It is obtained that the time for a curve passing from the head link to the tail link takes almost 9 seconds. So the 10s horizon contains the whole transformation of the curvature during prediction time which is incomplete in the 5s case.

Then, the cost function with and without the sum of propulsion forces is verified according to Fig. 4.4 and Fig. 4.6. Fig. 4.7 shows the related position of CM for each situation of Fig. 4.4 and Fig. 4.6. At the time point around 185 in the upper figure,

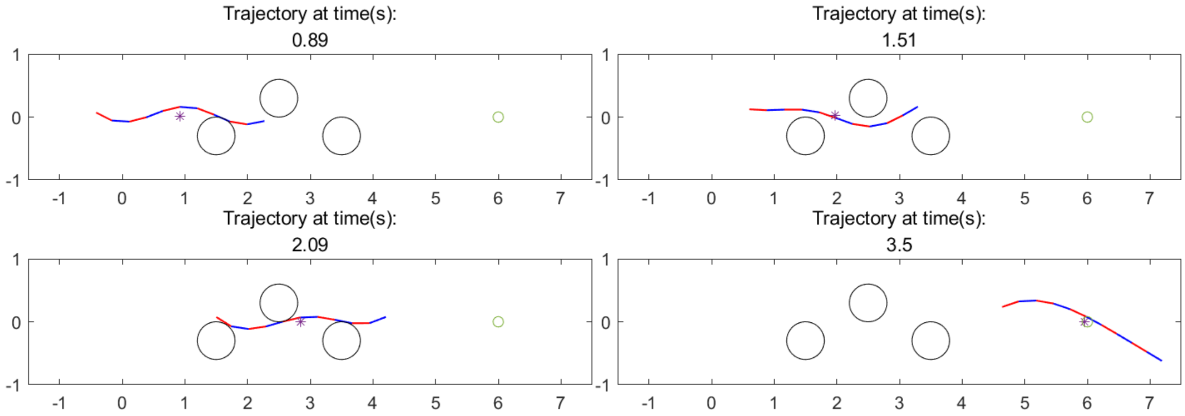


Figure 4.6: Snake locomotion of Initial condition: 1, $N = 100$, 4000 samples, cost function consists of positional error, control input.

the robot gains an increase in the x-axis position and a decrease in the y-axis position which is caused by the propulsion forces shown in Fig. 4.4.

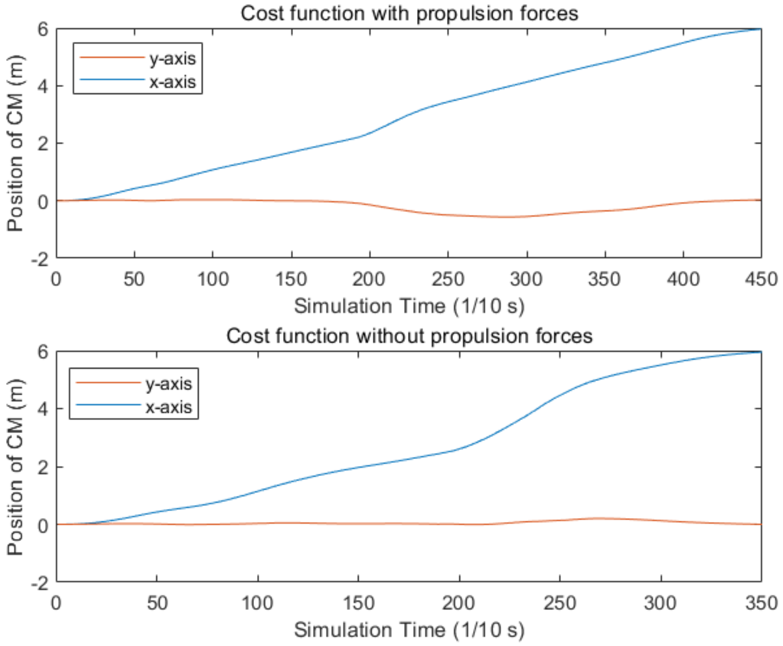


Figure 4.7: Position of CM in x and y axis related to Fig. 4.4 and Fig. 4.6

The same performance of cost without propulsion forces can be found starting near 200 for the second case. The sum of propulsion forces is introduced into the cost function as a negative part which will be maximized by the controller. Both results show that the snake robot generates propulsion forces even if it is not added to the cost function. Without the propulsion force term, the controller considers only the difference in positional error and control input. As a result, the robot prioritizes the shortest path to the destination more than generating propulsion force with a shorter simulation time but still generates propulsion during its movements. However, if propulsion is added, the snake robot will preferentially try to generate this kind of force as much as possible which is related to having more contact with obstacles. This feature may be useful for mitigating the current disturbance.

In the comparative analysis depicted in Fig. 4.8, the influence of varying initial conditions and the number of samples on the performance of the snake robot's locomotion is examined. Despite starting from different initial configurations, the simulations exhibit similar motion processes, indicating the robustness of the control strategy employed. Notably, the snake robot demonstrates consistent movement patterns across different initial conditions, suggesting the effectiveness of the control algorithm in achieving desired locomotion behaviors. Furthermore, comparison between simulations with differing sample sizes reveals that while variations in the number of samples, such as four thousand and two thousand, do not significantly impact the robot's movement patterns, increasing the sample size does lead to improvements in positional error performance. This observation underscores the importance of sample size selection in optimizing the accuracy and reliability of the control system. Overall, the findings highlight the resilience of the control approach to initial conditions and the potential benefits of increasing sample sizes for enhancing performance metrics, such as positional accuracy, in snake robot locomotion applications.

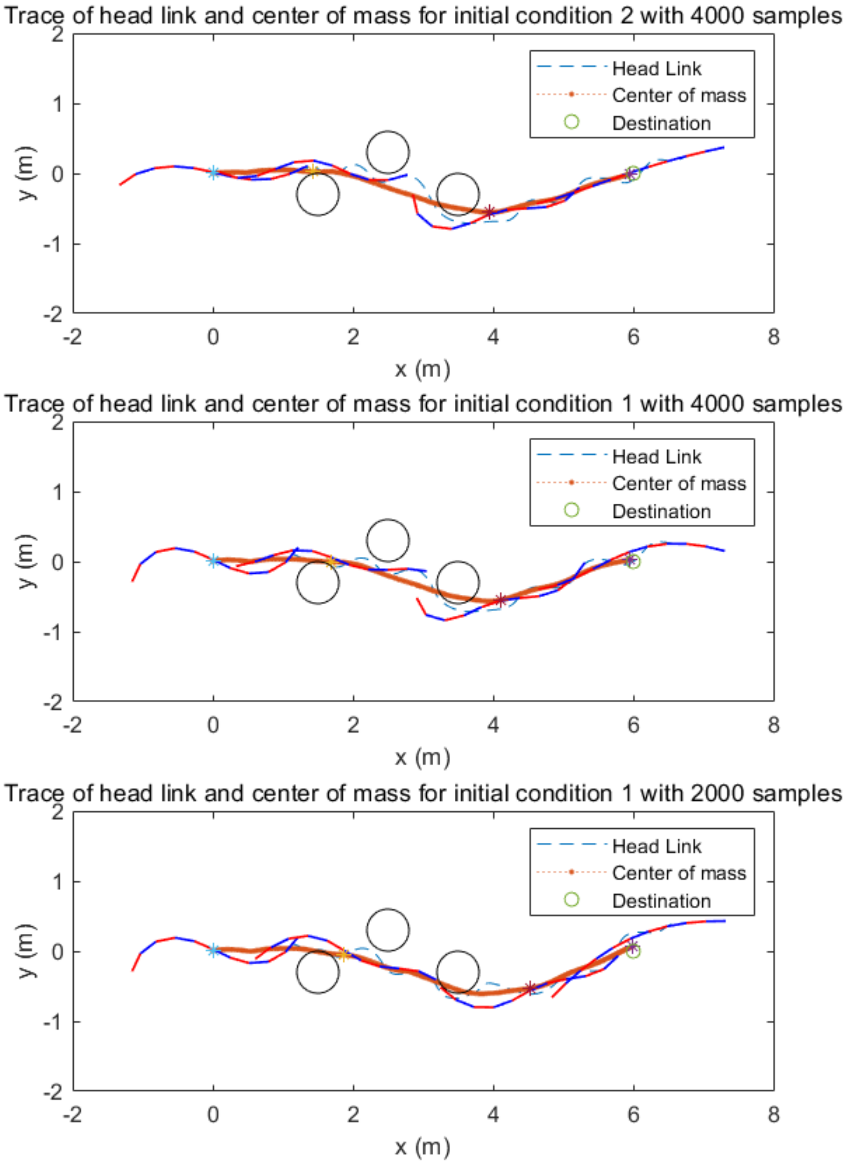


Figure 4.8: Trace of head link and center of mass for different conditions

4.2.3 Calculation performances

In all the experiments, the whole computation process runs on a laptop with 11th Gen Intel(R)Core(TM)i7-11800H. The calculation of each control circle takes about 30 to 60 seconds. The computation time makes it impossible for real-time implementation utilizing the $Parafor$ function of MATLAB. Therefore, GPU acceleration is chosen instead of CPU. Running the whole program in CUDA is tested with an underwater snake robot model with torque forces only. In this step, the complex hydrodynamics and the contact situations are not included into the system dynamics. The results showed that with the help of GPU parallel computing, for a 10 seconds simulation result under totally the same conditions, only 5 seconds is needed compared to almost one hour in MATLAB.

However, the dynamic model of the underwater snake robot does not include the fully dynamics which the snake robot may have. The calculations of the collisions are not introduced. Thus, in the next section, the contact model will be added to the system dynamic for a snake robot moving through some narrow space and also having contact with the surroundings. The hydrodynamics are also calculated based on the models shown in the Chapter 2.

4.3 Analysis of Underwater Snake Robot Moving through Narrow Space

Snake robots are uniquely suited for navigating pipes due to their flexible and modular design, inspired by the locomotion of real snakes. Their ability to generate waves along their bodies enables them to move through confined spaces with agility and precision, making them ideal for inspection and maintenance tasks in industrial pipelines, sewer systems, and other infrastructure networks. This serpentine motion allows snake robots to navigate around obstacles and through narrow passages efficiently. Additionally, snake robots can utilize anchoring and gripping mechanisms to stabilize themselves against pipe walls, facilitating traversal through vertical or inverted sections. However, deploying snake robots in pipe environments presents challenges such as developing robust perception and mapping algorithms to navigate complex pipe networks accurately. Efficient control strategies are also necessary to adapt to variable pipe diameters, surface conditions, and unexpected obstacles encountered during traversal. Despite these challenges, snake robots offer significant advantages over traditional robots in pipe environments, revolutionizing operations in industries such as oil and gas, water management, and infrastructure maintenance.

In the previous results, the underwater snake robot is tested to show the performance when moving through open space with some obstacles as the constraints. The snake robot shows the ability to moving among these obstacles and reaches the finial destination. The robot uses the environment to generate efficient gait pattern and realizes the obstacle-aided locomotion to help itself moving towards the goal.

In this section, the environment is changed to narrow space as the environmental constraints. The proposed control method is test for an underwater snake robot moving through various pipe structures. Ocean currents are also introduced into the system dynamics, and the contact between the snake link and the surroundings are described in a different way. The simulation results will show the efficiency of the proposed method through both the locomotion of the underwater snake robot and the calculation time.

The scenario of the simulation will be introduced first and then will be the different structures of the pipes. The methods used for collision detecting and the contact force calculation are shown. Finally, the simulation results of different initial conditions are given to show the advantages of both MCMPC and CDC.

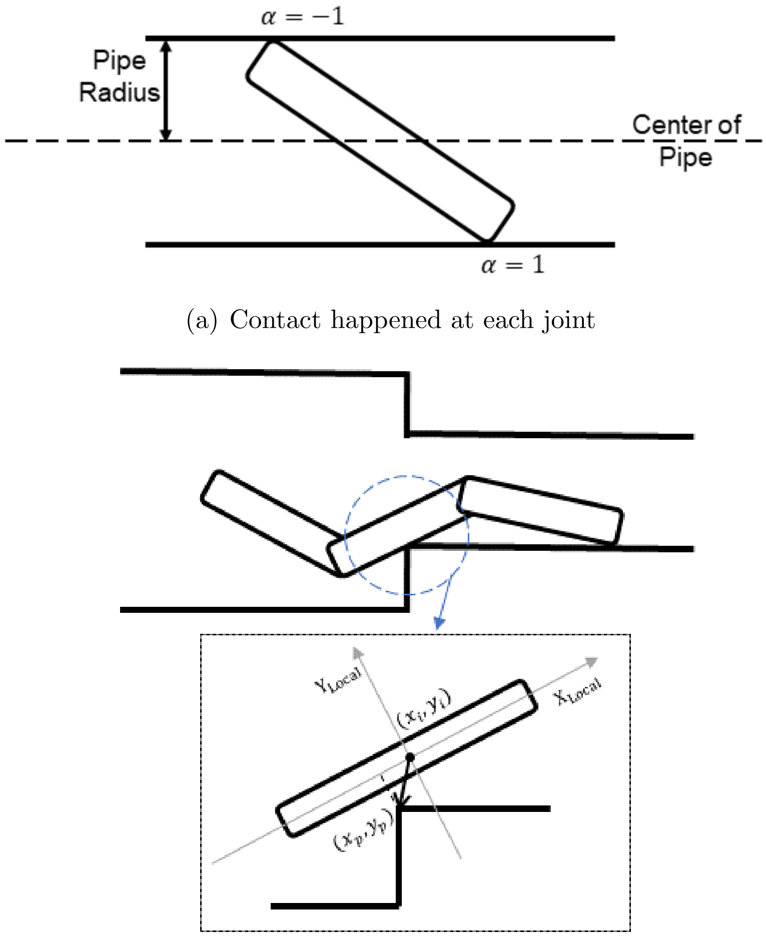
4.3.1 Assumption and Setup of the Sunk Ship Condition

The snake

4.3.2 Collision Detection and Contact forces

As previously mentioned, contact conditions will occur between the robot and the surroundings as it traverses a restricted area. MCMPC is applied to address this type of discontinuous situation. In this section, the contact dynamic model and contact detection are presented.

It is supposed that the contact force between the robot and the pipe occurs at the moment the joint initially makes contact with pipe surface. The associated forces influence the joints of the snake robot rather than the CM of the link.



(a) Contact happened at each joint

(b) Contact occurs on the body of the link

Figure 4.9: Method of determine the contact direction of the related link body.

The conditions for determining contact situation are shown in Fig. 4.9. The normal force acts on contact joint i in the y direction of local frame is determined by the distance between joint i and the center of pipe with same x coordinate. When the distance exceeds the radius of pipe, it is assumed that the contact is established. The contact parameter α_i is equal to -1 when the joint touches the upper side and 1 when having contact with the down side. After a collision takes place, the collision force is computed using a mass-spring-damper system,

$$F_{N,i} = \alpha_i \left|(-k_{msd}(D_i - R_p) - d_{msd}v_i)\right| \tag{4.11}$$

$$\begin{cases} \alpha_i = -1, \qquad if \quad D_i = y_{joint,i} - C_p > 0 \\ \alpha_i = 1, \qquad if \quad D_i = y_{joint,i} - C_p <= 0 \end{cases} \tag{4.12}$$

where D_i is the difference between the y-component $y_{joint,i}$ of joint i in the global frame and y coordinate of the related pipe center C_p aligned to x axis, R_p represents the pipe radius. k_{msd} and d_{msd} are the spring and damping coefficients. v_i is the related velocity of joint i in the y-axis of the global frame.

An instance of discontinuous diameter change in the pipe, as illustrated in (b) of Fig. 4.9, can result in the link establishing contact between the robot and the corner. It is assumed that contact force operates on the CM of the associated link. In the y-coordinate of the local frame, the distance from the corner point to the CM of the link is computed using the vector from link i to the corner.

$$Distance_i = [0,1](\mathrm{R}_{\mathrm{L_i}}^{\mathrm{global}})^T a_i \tag{4.13}$$

where $a_i \in \mathbb{R}^2$ represents the vector from link i to the closest corner in the global frame, it is then transformed into the local frame by the rotation matrix and the y-component of this vector is extracted. In this research, the thickness of the link is not considered so the link i is assumed to have contact with the corner if the absolute value of $Distance_i$ is smaller than 0.01m.

The friction force along the x-axis in the local frame between snake body and the pipe is incorporated into the system dynamics when the link makes contact with the pipe. It is demonstrated in [50] that the snake robot utilises friction force to maintain its stability and resist the weight of the robot, which is supported by wall friction. Furthermore, the research demonstrates that the friction coefficient's value significantly influences the snake robot's locomotion. In the aquatic environment, the snake could propel itself forward using the friction force against the ocean current. Sliding friction in the tangential direction of local frame is computed as:

$$f_i = -\mu_f \mathrm{sgn}(v_{t,i}) \left| F_{N,i} \right| \tag{4.14}$$

where μ_f is the friction coefficient, respectively. $v_{t,i}$ represents the tangential velocity in the local x coordinates.

4.3.3 Simulation Study of Underwater Snake Robot Locomotion in Each Type of Pipe

In this part, setups of the simulation environment are introduced also with parameters for MCMPC. Simulation results are shown under different initial conditions and analyzed at the end of this section. The control method is implemented in the CUDA v12.0. All the experiments are tested on the laptop with 11th Gen Intel(R) Core(TM) i7-11800H CPU and NVIDIA GeForce RTX 3060 Laptop GPU.

The top 1% of random inputs are chosen as the elite threads. The prediction horizon is set to $N = 100$ with a sampling time of $T_s = 0.1$ s. Constraints of MCMPC for control input and individual joint angles are denoted as $|\phi_i| \leq 1.04$. Constraint for CDC is $|u_i| \leq 0.5$ with $K_c = 0.5$ and $K_{cd} = 1.5$, respectively. The snake robot consists of 10 links with total link length $l_{all} = 1.4m$, total mass $m_{all} = 1kg$, and each link has the same length and mass. Parameters for hydrodynamics are set as $\rho = 1000$ kg/m^3, $C_M = 0.5$, $C_f = 0.03$, $C_A = 1$ and $C_D = 2$. Current values are established in accordance with the experimental conditions. The parameters of the MCMPC, CDC, and the initial conditions of CUDA are shown in Table 4.2.

The cost function includes a position error of the CM and the destination with control inputs is given by

$$L^n = \sum_{i=1}^{N} x^T_{\{k|i\}} Q x_{\{k|i\}} + u^T_{\{k|i\}} R u_{\{k|i\}} \tag{4.15}$$

Table 4.2: Parameters for MCMPC

Name	Meaning	Value
Prediction Horizon	Time interval of the prediction of MCMPC	10s
Prediction Steps	Number of predictions for one horizon	100
Number of Samples	Total samples for random sampling of MCMPC	8640
Number of Iterations	Times of repetition for each control loop	1
σ^2	Variance of normal distribution for input sequence	0.1
Input Constraints	Constraints for inputs generated by MCMPC	$-1.04 < u_i < 1.04$

$$x_{\{k|i\}} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} - \begin{bmatrix} D_x \\ D_y \end{bmatrix} \tag{4.16}$$

where D_x, D_y are the coordinates of the final destination, $Q = \begin{bmatrix} 10 & 0 \\ 0 & 50 \end{bmatrix}$ and $R = 0.5$. The values of Q are chosen for the performance of snake robot. y component is greater than x component in this study, as it is presumed that the robot will initially dive priorly in y direction and then proceed towards the entrance of along x direction. This provides the robot with additional space to adjust the proper angle of entry. (large weight converges faster)

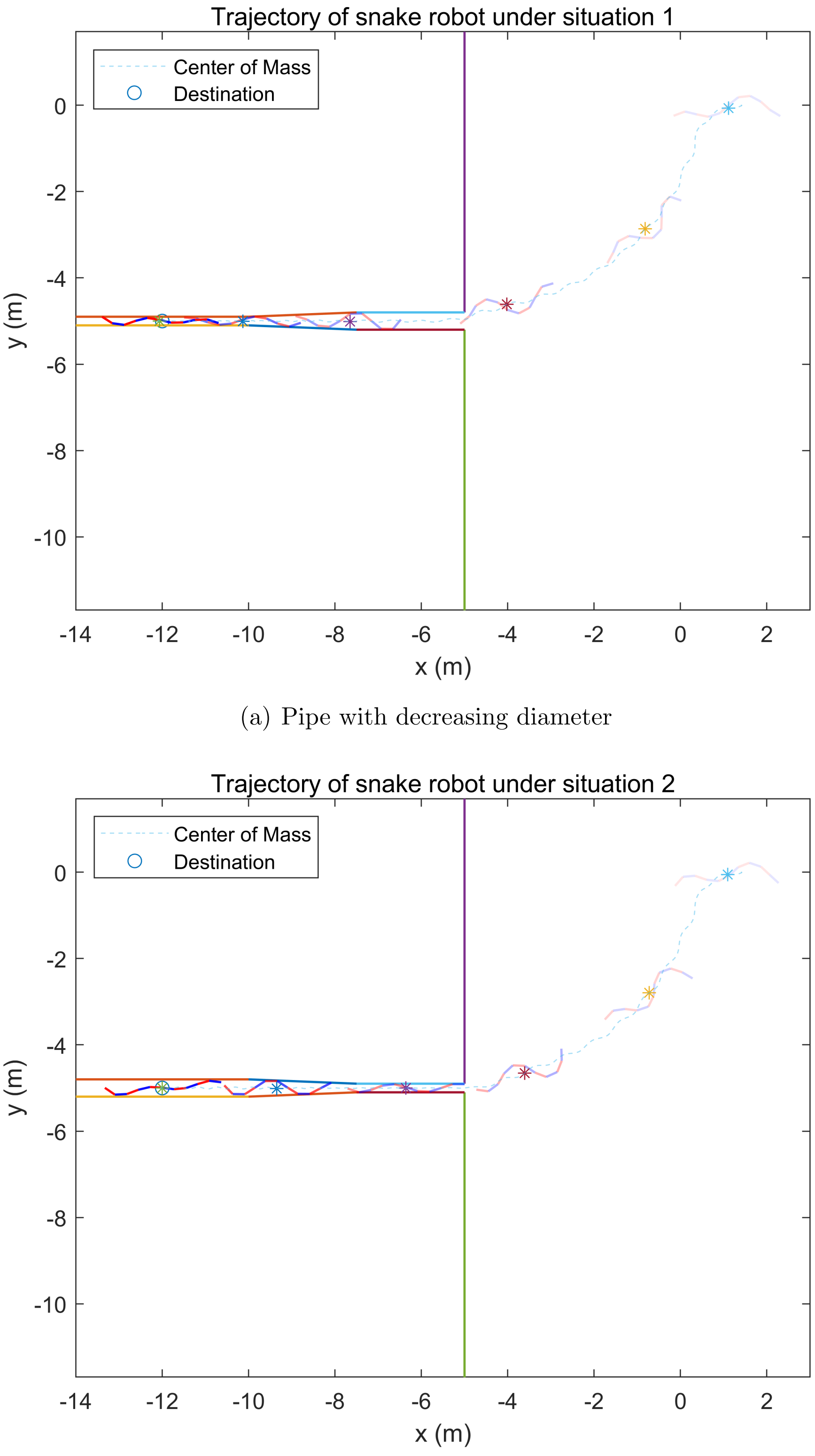
In this study, various pipe configurations are examined in order to validate the capability of the proposed control method to accommodate pipes with varying diameters and structures. The robot will identify the pipe as the entrance to access the interior and reach the destination. The initial position of the head joint of the robot is (0,0) with zero initial states. The position of the entrance and the final destination are given to the robot as external information.

The vertical alignment of the wall serving as the exterior surface of the sunken ship is established at an x-coordinate of -5. The entrance locations remain consistent across all experiments. The time progression is indicated by the varying shades of the lines in each of the resultant representations in Fig. 4.10,4.11,4.12,4.13,4.15. This suggests that the initial condition of the robot at the start of the experiment is denoted by light-colored lines, while the state after a specific length has passed is represented by dark-colored lines, respectively.

4.3.4 Continuously change of pipe diameter

As a simulation scenario, a pipe with a continuously changing radius is initially evaluated. The pipe initially holds a radius of 0.2m. However, after deviating by -7.5m, its diameter gradually reduces to 0.1m and keeps constant after $x = -10$. In the second experiment, which is designed to examine the inverse circumstance, the pipe's radius begins at 0.1m and gradually increases to 0.2m. The simulation configurations for both investigations are identical.

The results in Fig. 4.10 are illustrated. The snake-like robot, as depicted in the figure, advances from its initial position toward the entrance. Without a given reference, MCMPC and CDC generate the snake robot's locomotion pattern at random, and the robot is capable of adjusting the amplitude of the moving wave as it traverses different environments. Once the robot reaches the entrance of the pipe, it proceeds inside and executes a traveling wave locomotion resembling a trapezium, as a result of its physical constraints. Assuming that the ocean current occurs only along the global x-axis within the pipe, its velocity is held constant at 0.1 m/sec for the shorter sections and 0.2 m/sec for the larger sections. In addition, the results demonstrate that the snake robot is adaptable to both scenarios, as its locomotion amplitude would vary in accordance with the pipe's diameter.



(a) Pipe with decreasing diameter

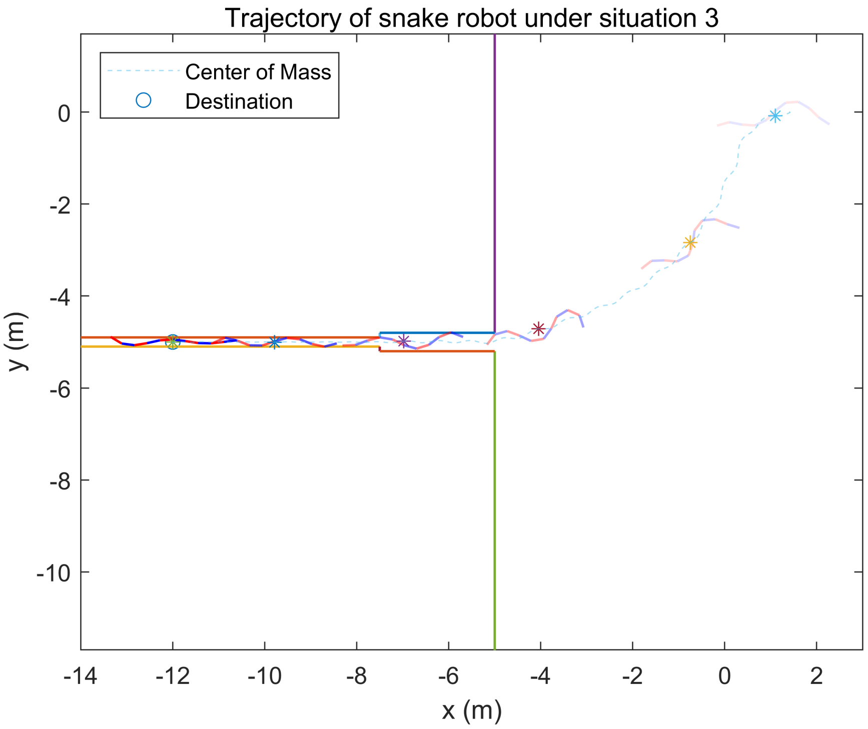
(b) Pipe with increasing diameter

Figure 4.10: Results of snake robot moving through a pipe with continuous change of the diameter.

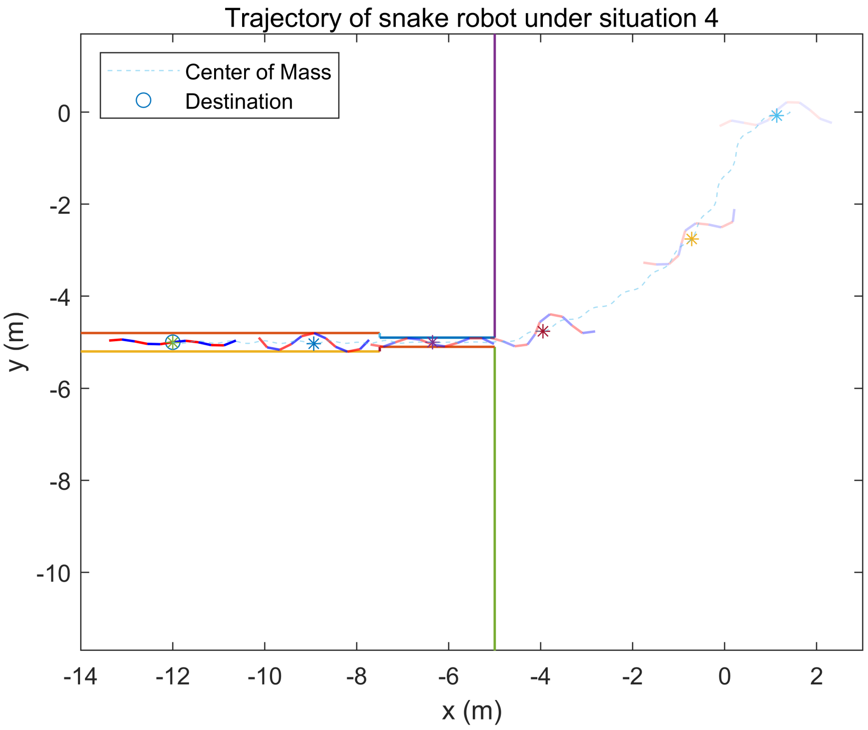
4.3.5 Discontinuous change of pipe diameter

The pipe's discontinuous section is subsequently examined by the snake robot. The initial conditions for the robot are identical to those of the preceding tests. In the first simulation of this section, the pipe's diameter is initialized at 0.4m between coordinates $x = -5$ and $x = -7, 5$. It then stabilizes at 0.2m thereafter. The pipe's center is located at $y = -5$. The second simulation examines the opposite scenario. The ocean current is maintained at the same velocity of 0.1m/sec for thinner pipe segments and 0.2m/sec for the remaining pipe segments. The link body and the stage may come into contact due to the discontinuous connection between the two sections of the pipes. As previously stated, the system dynamics takes into account this type of contact situation, and the associated contact force exerts an influence on the link's center of mass.

Fig. 4.11 shows the related results. The initial conditions for both simulations are identical. In both scenarios, the robot executes smooth locomotion in order to approach the tunnel's entrance. Throughout operation, the robot's locomotion varies in response to the physical limitations of its surroundings, demonstrating its adaptability to different environments. Even after reaching the final destination, the robot continues to generate slowly movements in order to maintain stable in the face of current influence.



(a) Pipe with decreasing diameter



(b) Pipe with increasing diameter

Figure 4.11: Results of snake robot moving through a pipe with discontinuous change of the diameter.

4.3.6 Pipe with a bend structure

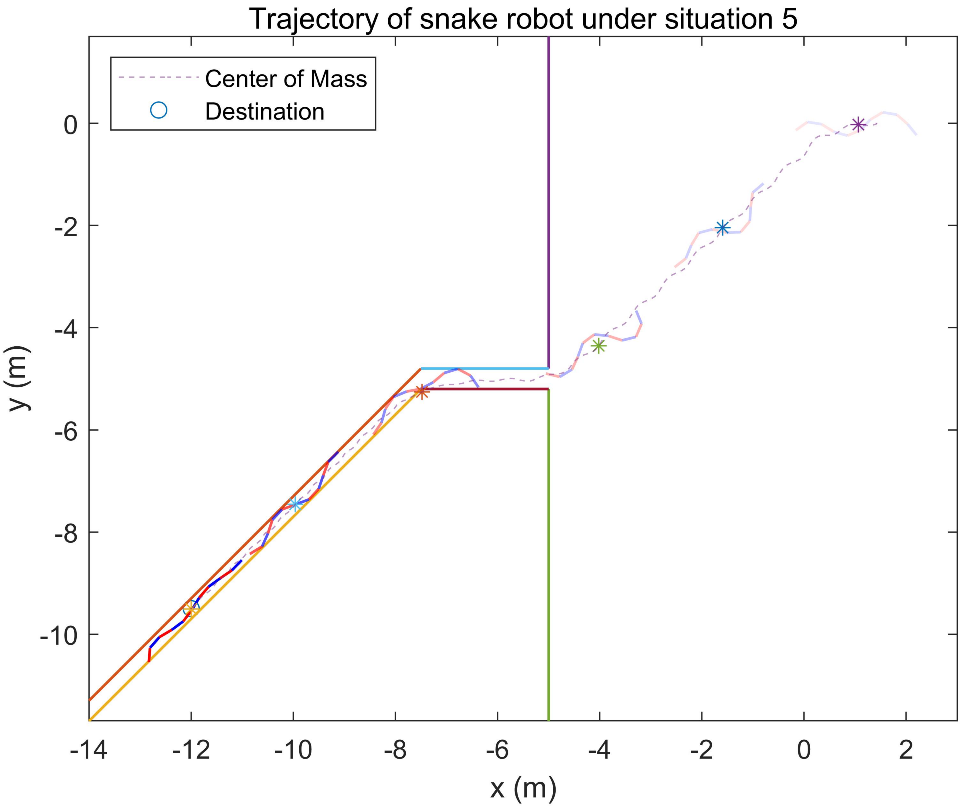


Figure 4.12: Results of snake robot moving through a pipe with a bend.

Another situation of the pipe is that it is bent at the point where it meets the horizontal section. Fig. 4.12 shows the simulation result. The pipe is characterized by a flat part with a diameter of 0.4m, a turn occurring at the point where $x = -7.5$, and a degree of 45°. Likewise, the diameter is altered to 0.28m, respectively. The ocean current exerts an influence on the y-axis of the inclined parts, causing a change of 0.1m/sec in both the x and y coordinates. The result shows that the snake robot successfully enters the pipe. When it comes to the bending part, the locomotion of the robot also changes to pass through the bend.

Additionally, the result indicates that the snake robot generates propulsion force by pressing its body against the blue portion of the pipe, thereby enabling it to move forward in the same direction as the pipe. This type of propulsion facilitated by contact exhibits the same characteristics as obstacle-assisted locomotion.

It is worth noting that the snake robot, with the assistance of the CDC, would select the essential parts from each side of the pipe in order to produce advanced locomotion in each of these scenarios. An instance of this characteristic is also noted in the study [18], where the CDC is applied to regulate a snake robot within a constricted, twisting environment.

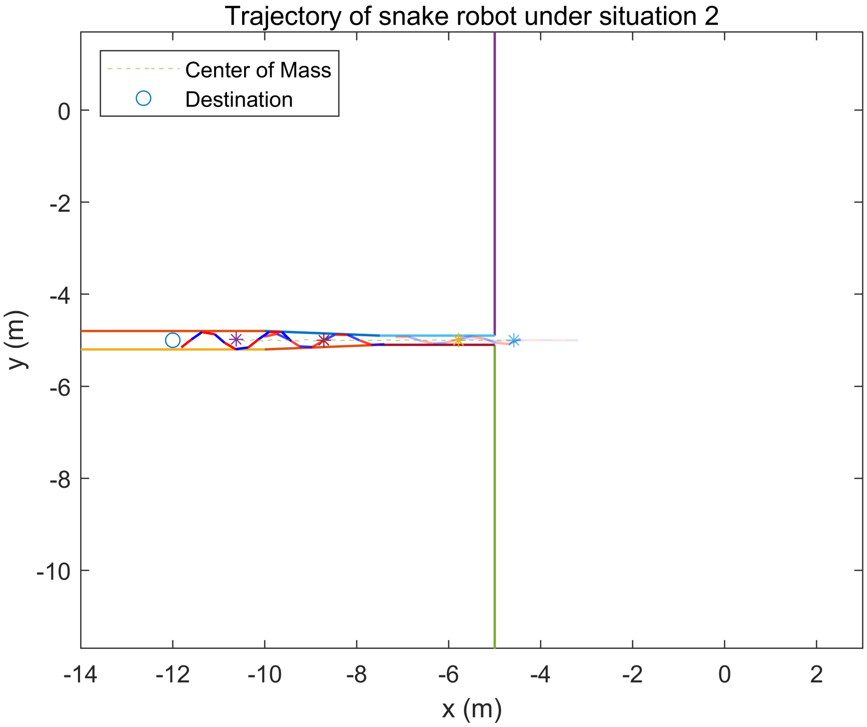
4.4 Benefits of Utilizing MCMPC for Controlling the Underwater Snake Robot

This section provides an analysis of the benefits associated with employing MCMPC for the head joint. To demonstrate the efficacy of MCMPC, the head joint of the snake robot is referenced with a sinusoidal input for comparison. The remaining joints are controlled by CDC in the same manner as the proposed method. Pipes with continually varying diameters are tested.

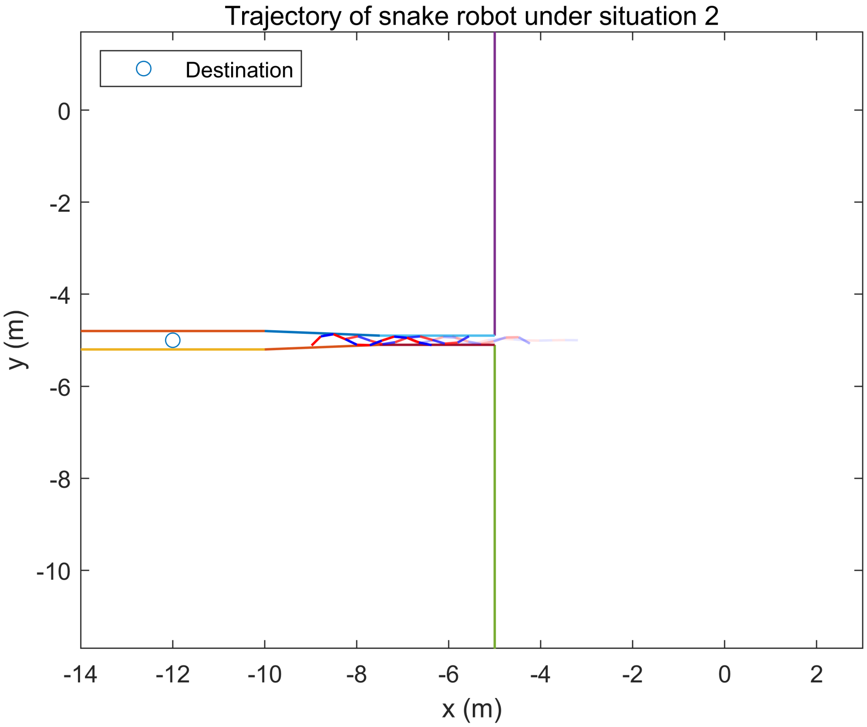
The robot is configured to begin its trajectory at the entry of the pipe. Both experiments simulate the performance of 60-second results. The following specifies the input reference for the head joint:

$$u_{ref} = 1.04 \sin(2.2 t_{ref}) \tag{4.17}$$

where t_{ref} represents the simulation time. The frequency and the amplitude are set the same as for the MCMPC.



(a) Locomotion with MCMPC for the head joint



(b) Locomotion with given reference for the head joint

Figure 4.13: Results of snake robot moving through a pipe with MCMPC and with a given joint reference both in 60 seconds

Fig. 4.13 illustrates results of the associated experiment. Both snake robots demonstrate the capacity to advance in response to a pipe diameter change. Due to the fixed reference, the snake robot was unable to change its locomotion in case (b). This results in increased extra contact with the pipe, thereby impeding the velocity of motion. In contrast, the snake robot shown in (a) is more adaptable across environments and can therefore travel further in the same amount of time due to the assistance of MCMPC.

For comparisons of both control methods, the efficiency of MCMPC is demonstrated above. However, comparing results of ”with and without CDC” is difficult. It is challenging to set up each joint as control input for MCMPC. Because the computation becomes more complicated as the quantity of inputs increases and amplifies with the complexity of dimensions. Consequently, CDC is utilized for all simulations.

4.5 Merit of Introducing Contact Model

This section demonstrates the impact of incorporating the contact model into the MCMPC prediction stages. In the prediction stages, two MCMPC prediction models are evaluated both with and without the contact dynamic. Failure to incorporate the contact dynamic into the prediction stages of MCMPC would result in the robot failing to anticipate a collision prior to its occurrence. The robot starts from the initial position of (0,0) in order for it to move in proximity to the destination. The entrance portion of the pipe is configured with a diameter of 0.2m, which is marginally inadequate for the snake robot to traverse. The investigations are simulated using three different sample amounts for MCMPC: 8640, 5760, and 2880, with each scenario being executed twenty times.

The successful rate represents the chance of the robot to move through the entrance among all the tested simulations. It is calculated according to the formula as:

$$\frac{Robot\ get\ inside}{Toatl\ number\ of\ simulations} \times 100\% \tag{4.18}$$

Fig. 4.14 illustrates experimental results associated with the given variable.

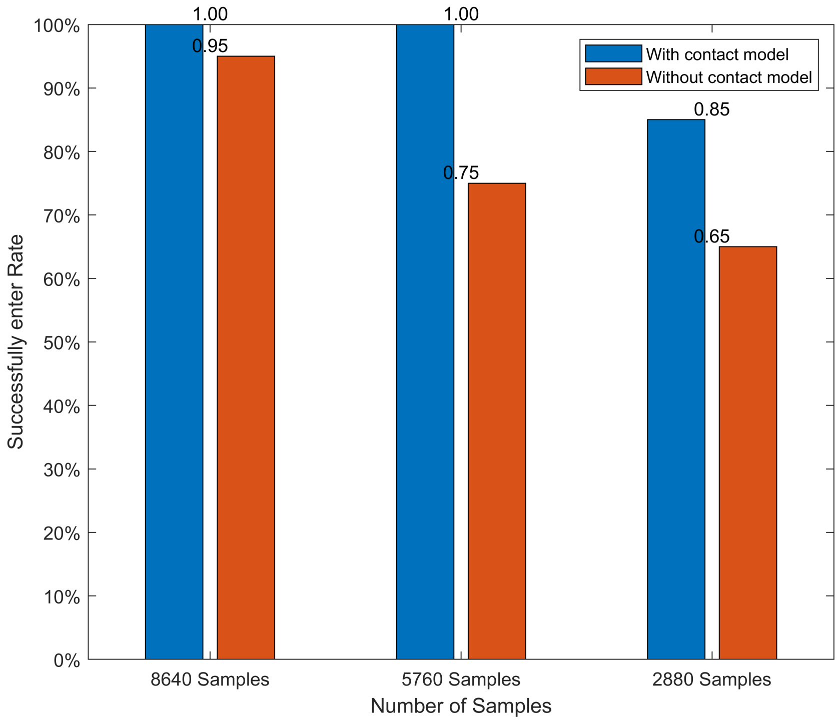
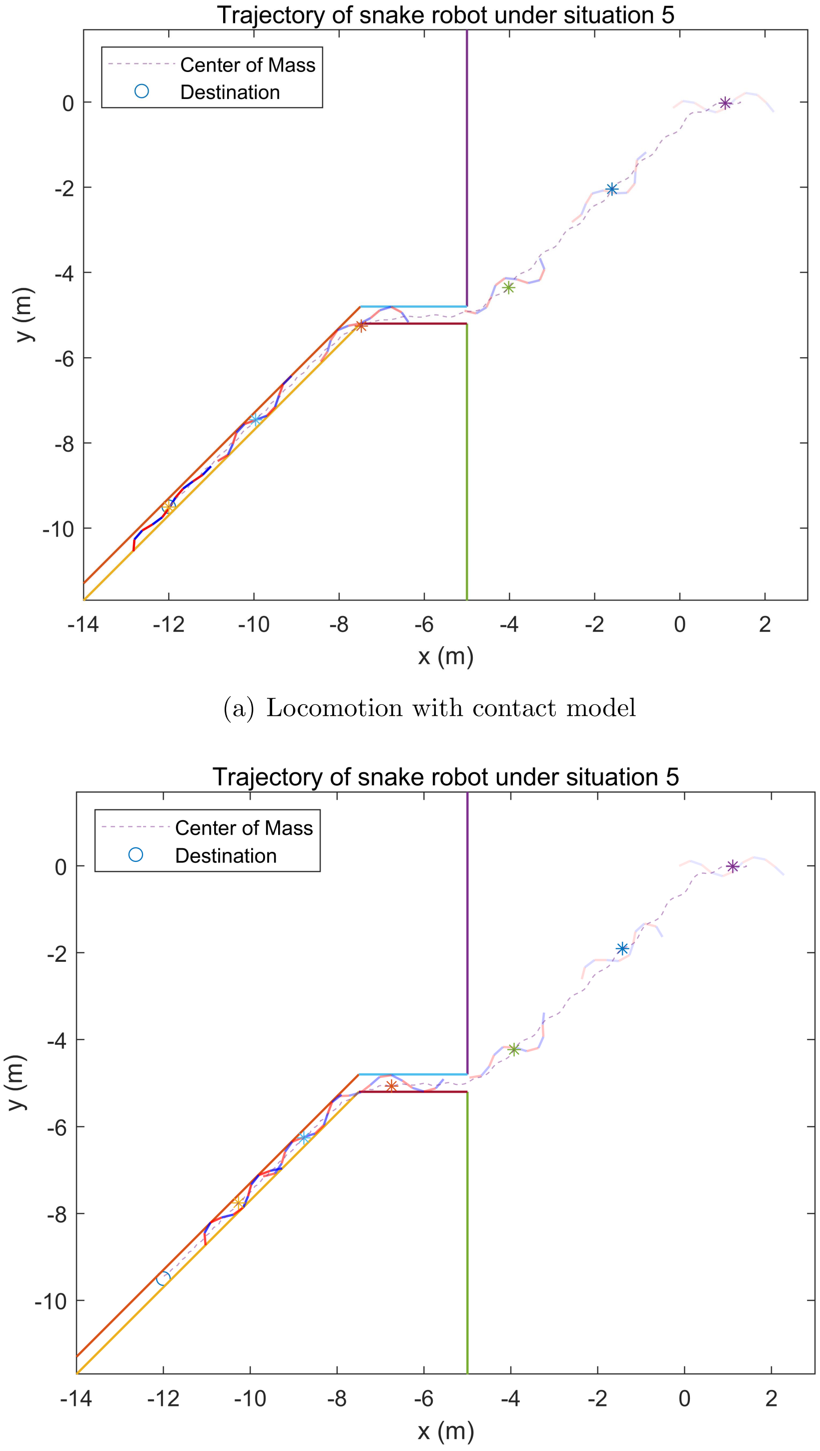


Figure 4.14: Success rate of entering the pipe for MCMPC with and without the contact dynamic during prediction steps

An increased quantity of samples results in enhanced efficacy of the robot, enabling it to enter the pipe. The circumstance in which the robot is unable to enter is due to a trajectory change caused by the head of the robot colliding with the wall. Based on the results of the experiment, MCMPC with contact model successfully entered the tube for both 8640 and 5760 samples. However, the possibility of without contact model decreased from 0.95 to 0.75, suggesting that the robot's failure probability may be higher in the absence of contact model. With a reduced number of samples(2880), both models experience a failure condition due to the diminished likelihood of diverse gait patterns. Performance is invariably enhanced when contact model is incorporated into MCMPC prediction stages. It is noteworthy to remark that the computation time for MCMPC remains almost the same whether the contact model is utilized or not.



(a) Locomotion with contact model

(b) Locomotion without contact model

Figure 4.15: Results of MCMPC with and without contact model during prediction steps under situation 5 both in 90 seconds

Fig. 4.15 shows experiment results of two situations with 8640 samples. All of the snake robots manage to traverse the pipe's entrance successfully. However, collisions occur once the robots enter the pipe, and it is observed that the model of the robot without contact makes needless contact with the pipe. Without contact information detected during the prediction phase of MCMPC, the head link of the robot would collide with the pipe. Consequently, the contact model-equipped robot arrives at the designated location within 90 seconds of simulation time, whereas another robot is still en route to the same destination. The robot without contact model depicted in (b) continues to advance along the pipe's narrow structure with the assistance of the CDC and reaches its destination in 120 seconds.

4.6 Robustness of proposed methods

Model predictive control highly replied on the accuracy of the model used for prediction. Thus it is important to verify the robustness of MCMPC. This section presents the simulation studies conducted to verify the robustness of the MCMPC for an underwater snake robot. The robustness of the control algorithm was assessed by testing the system under varying conditions of hydrodynamic coefficients since the various environment may cause the differences between the experimental data and the real values. Due to the fact that these hydrodynamic forces play an important part of the swimming capability for the underwater snake robot and will highly affect the locomotion. By running the related simulations help to ensure that the control system can maintain stability and performance despite changes in the dynamic parameters of the underwater environment.

The simulations are tested based on the scenario which including the continuously change of the pipe diameter, which is set as the situation 1. All the simulation setups are chosen as the same as the one in section 4.3.4, respectively. The initial values of the drag coefficient C_D and C_f are set as 1 and 0.015. The added mass coefficient C_A is equal to 0.5. In [95] it is shown that for a cylinders under the same conditions to the underwater snake robot, the parameter C_D changes from nearly 2 to 3 and C_A is between 0.93 and 1. In [11], it is shown that for a flow of Reynolds number of 10^5, the values of the fluid parameters are set as C_D, C_A, C_f with $1, 1, 0.01$.

In the following simulations, the underwater snake robot was modeled considering key hydrodynamic effects including linear and nonlinear drag forces and added mass effects. The nominal values of the hydrodynamic coefficients were as follows: the model for the prediction steps of MCMPC is based on the initial values for these three coefficients, when it comes to the actuated model, these parameters are set to three different conditions as:

- Initial values: $C_D = 1$, $C_A = 0.5$, $C_f = 0.015$
- Situation R-1: $C_D = 1.5$, $C_A = 1$, $C_f = 0.02$
- Situation R-2: $C_D = 2$, $C_A = 1.2$, $C_f = 0.03$
- Situation R-3: $C_D = 1$, $C_A = 1$, $C_f = 0.01$

The simulations are tested with the same time period of 120 seconds. The results of these simulations are shown as the following figures, where the title for each figure shows the situation of the related simulation with the values given above.

The simulation studies verified that the MCMPC framework is robust to variations in key hydrodynamic parameters. The control system maintained stability and acceptable performance across a wide range of added mass and drag coefficient values. These results suggest that the MCMPC can effectively handle uncertainties and changes in the underwater environment, making it a viable control strategy for underwater snake robots operating in dynamic conditions. The snake robot will reach the final destination under these three conditions and generate various locomotion according to the surrounding environment.

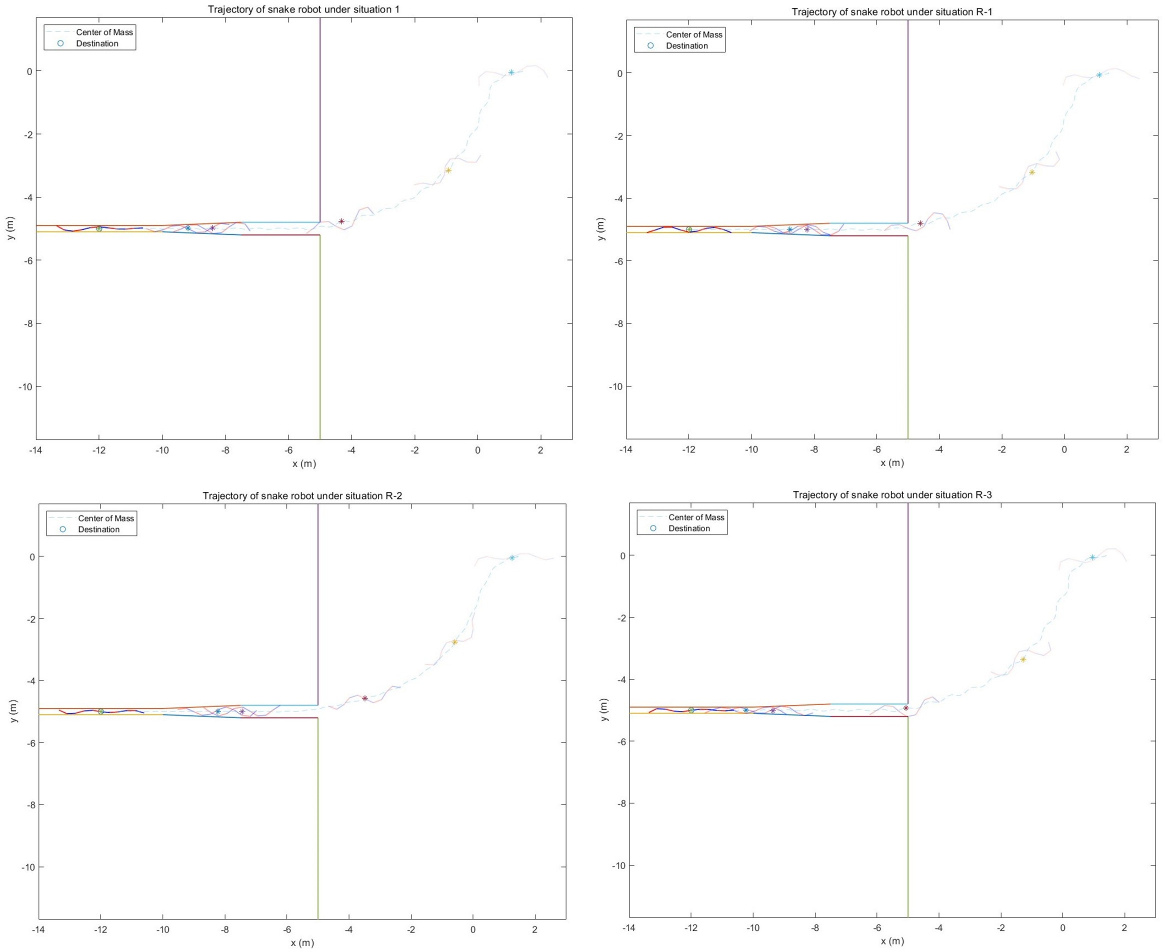


Figure 4.16: Snake locomotion of three different values for the hydrodynamic coefficients

As shown in the Fig. 4.16, each framed robot gait was captured at the same time point for all the simulation results. When the coefficient values in the real environment are smaller than in the predictive model, the snake robot is able to move further at the same point in time, implying that the real environment is sparser than expected. On the other hand, when the real underwater environment is stickier than the predictive model, this kind of error causes the robot to move a bit slower.

Another situation is the error of the current velocity. In this part of the simulation, only the difference between the current velocity is tested. Other hydrodynamic parameters such as the drag and added mass coefficients are chosen as the same for both simulations. The current velocity in x-axis direction is set as 0.1m/s for the prediction model of MCMPC while it is chosen as 0.2m/s for the actuated model which simulates the real situation. The performance of the snake body is shown as follows:

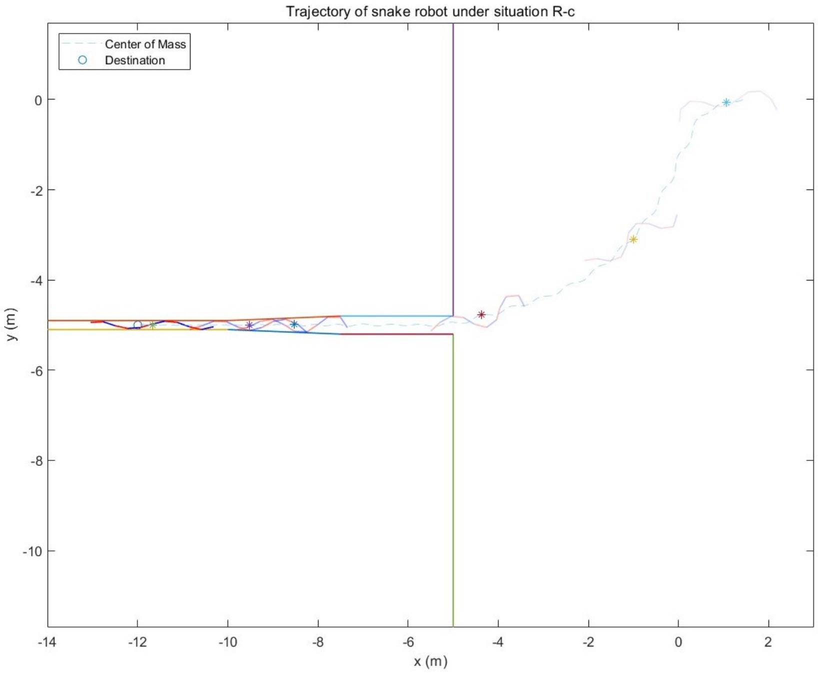


Figure 4.17: The locomotion of snake robot with different currents for prediction model and actuated model.

The performance is simulated with the same simulation time. As shown in the Fig. 4.17, the snake robot is able to perform various locomotion depend on the surrounding environments. However, due to the error between the values of current velocity, the robot may need more time for itself to reach the final destination. Due to the fact that the current is predicted to be slower than the real condition. But this error does not affect the locomotion of the snake robot.

4.6.1 Influence of Prediction horizon

This section of the study investigates the impact of varying lengths on the prediction horizon. The kinematic gait generation of the snake robot is studied as well in relation to the predicted step length. This is a critical factor for MCMPC, as a longer prediction length enables the robot to make more accurate predictions about the future. Consequently, it can respond more effectively to unforeseen circumstances and improve its judgment. However, a longer step size also results in an increased computation time, which can be a significant factor in practical applications. In contrast.

The results of simulations concerning the impact of the prediction horizon on the snake robot's performance are presented below with the simulation results of the robot performance with prediction horizon $N = 20$ and $N = 40$ shown in Fig. 4.18. There is not so much difference between the snake performance between the prediction horizon 80 and 100, thus, only the above two results are shown to highlight the effect of shorter step lengths on the kinematic morphology of snake robots.

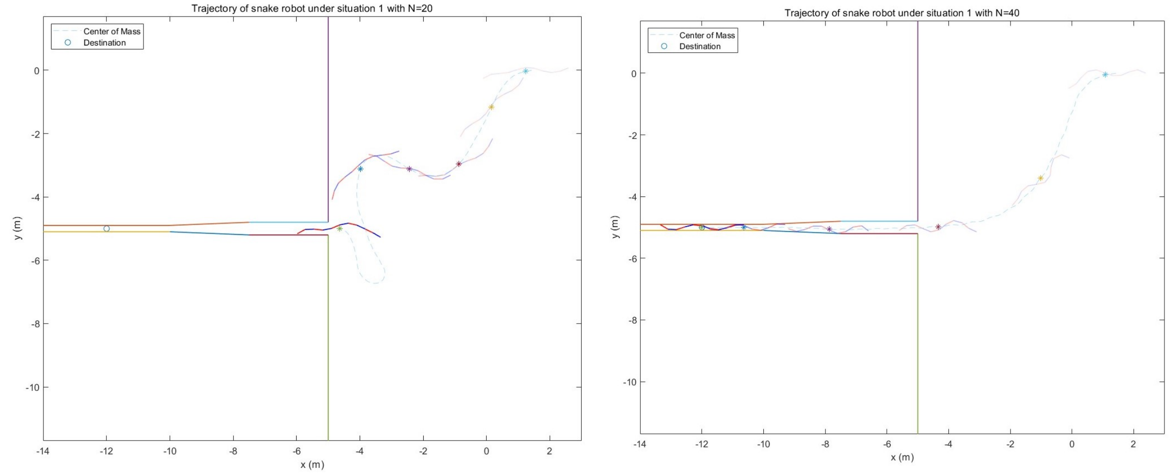


Figure 4.18: Snake locomotion performances of short prediction horizon for MCMPC

The performance of $N = 20$ demonstrates that the robot's generated locomotion is insufficient, indicating a small amplitude of the gait pattern. The entrance to the pipe is beyond the capabilities of the robot.

The performance of $N = 40$ shows that the generated locomotion of the robot is complete. The robot is capable of entering the pipe's entrance. Nevertheless, the robot's contact area with each side of the pipes is incomplete. As a result, the robot's ability to traverse the ocean current is weakened.

The efficiency of the locomotion generated by the $N = 80$ configuration enables the robot to get through the pipe. The robot has completely contact with the pipe.

According to the simulation results, the duration for one curve to pass from head to tail was average around 8.6s. The snake robot will perform better if the entire locomotion is contained within the prediction horizon. However, short prediction horizon did bring one benefit which is the computational cost for MCMPC. The calculation time for $N = 20, 40, 80$ is with the average of $0.021s, 0.04s, 0.078s$. Thus, a short prediction horizon may be chosen when the calculation time is considered more important with a normal performance of the snake robot locomotion.

4.7 Chapter Summary

The present study introduced sample-based Monte Carlo Model Predictive Control and Curvature Derivative Control as control strategies for a two-dimensional underwater snake robot. By utilizing MCMPC and CDC, the snake robot achieves more efficient locomotion in response to different kinds of environment result in a better performance than a given fixed reference. Through the implementation of GPU parallel computing, the time for the Monte Carlo random sampling procedure is drastically decreased compared to previous methods.

Simulation results show that by incorporating contact dynamics into the MCMPC prediction stages, the robot is capable of successfully passing the pipe through the entrance while the robot is located external to the wall initially. The snake robot is designed to exhibit adaptive mobility in order to navigate through diverse settings, such as varied pipe constructions with varying geometric constraints, and successfully reach its desired goal. The robot also possesses the capability to maintain stability in the presence of a current.

The robustness of the proposed method is also tested to check how the MCMPC would affect the errors in the model of the underwater physical environment. The simulations are taken based on different values of the added mass and drag coefficients between the prediction model and the actuated model. The error of the value for the current velocity is also tested. The simulation results show that the proposed control methods are able to deal with the model error.

The influence of the prediction horizon is studied. The prediction horizon is a crucial factor for Monte Carlo Model Predictive Control, as a longer prediction horizon enables the robot to make more accurate future predictions, respond more effectively to unforeseen circumstances, and improve its judgment. However, longer prediction horizons also increase computation time, which can be a significant factor in practical applications. Therefore, a shorter prediction horizon might be preferred when computational efficiency is more critical, balancing performance and computation time.

Nevertheless, proposed control approach fails to perform concertina locomotion in confined spaces. Because rest of snake's body merely amplifies the motion produced by the head. In [17], Tegotae-based control was utilized with CDC to generate autonomous gait pattern that achieved both concertina locomotion and scaffold-based locomotion. This kind of advantage will be considered to improve the performance in the future. Future works will also entail separate control of snake body components to achieve numerous objectives. As an example, MCMPC generates control inputs at the head and middle joint, allowing one part of the body to grasp the object while the other part propels the robot forward. Furthermore, design and implementation of the proposed method to an actual snake robot are future plans.

Chapter 5

Conclusions and Future Works

The conclusions of this study and the future challenges on this study will be given in this chapter.

5.1 Conclusions

Adaptive locomotion control is proposed in this study for a serpent robot capable of traversing a variety of underwater environments. Chapter 1 provides an introduction to the fundamental understanding of snake behavior in various environments through the discussion of the following biological gait patterns: concertina locomotion, rectilinear locomotion, sidewinding gait, and lateral undulation. Additional information is provided regarding prior research conducted on this subject, with particular emphasis on the control methods suggested by other scholars. The PID controller is employed to regulate the predetermined gaits produced by the serpent robot. In lieu of PID, MPC is implemented so that the serpent robot's locomotion can be altered in response to varying circumstances.

The introduction of the two-dimensional dynamic model for the underwater serpent robot occurs in Chapter 2. It is anticipated that the snake robot will comprise n connections, each possessing $n+2$ degrees of freedom. Introduced are the hydrodynamics of the subsea environment, including fluid torques, added mass forces, and drag forces.

MotionGenesis is employed to both generate a GPU-compatible model and characterize the dynamics of the complex system in order to facilitate its implementation.

In Chapter 3, the proposed control methods for the underwater serpent robot are introduced. Monte Carlo Model Predictive Control, a sample-based model predictive control, is employed to regulate the head joint. In contrast, curvature derivative control is utilized to control the remaining joints. MCMPC enables the direct introduction of discontinuous dynamics into the system dynamics, thereby equipping the robot to handle contact situations. CDC calculates optimal bending moments taking into account the energy consumption of joint torque. By transmitting joint angles and propulsion forces and conveying effects between adjacent links, the control method permits adaptation to fluctuating environments. Without requiring knowledge of wall geometry, CDC enables sophisticated locomotion by enabling the robot to choose the most advantageous contact areas for forward progress. The effectiveness of the suggested approach is demonstrated through the analysis of simulation outcomes across various simulation environments. The computational expense is substantially diminished through the simultaneous implementation of MCMPC and CDC.

The main part of the study is introduced in Chapter 4. By employing Model Predictive Control (MPC) and Centralized Distributed Control (CDC), the snake robot achieves enhanced locomotion efficiency under various environmental conditions, resulting in superior performance compared to a predetermined fixed reference. By utilizing GPU parallel processing, the Monte Carlo random sampling procedure experiences a significant reduction in time compared to earlier methodologies.

The simulation findings demonstrate that by integrating contact dynamics into the prediction stages of Model Predictive Control (MPC), the robot is able to effectively navigate the pipe into the entry even when it is originally positioned outside the wall. The snake robot is specifically engineered to demonstrate flexible and adjustable movement capabilities, enabling it to effectively traverse a wide range of environments, including complex pipe structures with different geometric limitations, and ultimately achieve its intended objective. The robot is able to retain stability even when there is a current present.

The proposed method's robustness is also evaluated to determine the impact of the MCMPC on the errors in the underwater physical environment model. The simulations are conducted using varying values of the added mass and drag coefficients between the prediction model and the actuated model. Evaluating the current velocity's error is also considered. The simulation results indicate that the proposed control methods are capable of dealing with the model error.

The impact of the prediction horizon is investigated. The prediction horizon is an essential component of Monte Carlo Model Predictive Control, as an extended prediction horizon allows the robot to make more precise future predictions, respond more effectively to unforeseen circumstances, and enhance its judgment. Nevertheless, the computation time is also increased by extended prediction horizons, which can be a substantial factor in practical applications. Consequently, a shortened prediction horizon may be preferred when computational efficiency is more critical, as it allows for a more balanced approach to performance and computation time.

5.2 Future works on Underwater Snake Robot

The initial step in the future is to develop additional functions for the underwater snake robot. The robot is capable of executing more complicated locomotion due to its many degrees of freedom. In the current research, MCMPC controls only the head joint, while the remaining joints are simply replicated from the prior joints. This may restrict the combination of various components of the snake's body. In the future, the control inputs for MCMPC may be assigned to various components of the snake body, enabling each component to perform a variety of functions. For instance, the middle joint of the snake may be selected as one of the control inputs to ensure that the half body exhibits separate locomotion in comparison to the other components. The robot can utilize one component to grasp small objects, while the other part will generate propel locomotion that draws the entire robot out of the internal space.

The dynamic model for the underwater snake robot may be relevant to the second plan. The proposed control methods are implemented in this investigation using a two-dimensional model. The robot's gravitation is assumed to be canceled in this model. Nevertheless, the three-dimensional model is necessary in real-world scenarios. The model's complexity will also increase when it is subjected to the 3D condition. This model has not been examined due to its inclusion in the future plan. Nevertheless, the author's personal assessment is that the calculation cost will not experience a substantial increase. This is due to the fact that the dimension will introduce an additional degree of freedom, which the proposed method can still manage due to the model's existing numerous degrees of freedom. The calculation time may continue to be maintained within the specified prediction horizon.

Additionally, there is a situation that necessitates consideration in the robot's dynamic model. The dynamics of the environment are also changed when the snake is moving through the pipe, where there may be a pocket of air. The snake body will no longer be affected by fluid dynamics, and the propulsion forces will be generated by the friction forces between the snake body and the pipe's surface. Consequently, it is imperative to incorporate this type of scenario into the snake's dynamic model.

The third plan relates to the actual robot's design. Relying on the simulation results, the proposed control methods have been validated for a variety of scenarios. Consequently, it is imperative to observe the efficacy of this investigation on the actual underwater snake robot. It is also important to consider the use of various sensors to detect physical dynamics, such as the contact between the snake body and the environment.

The following are the future goals of this study: the implementation of the proposed methods in a real underwater snake robot, the enhancement of the present 2D model to 3D with additional dynamics, and the development of more complex functions for the snake robot.

Bibliography

[1] K. Karakasiliotis, R. Thandiackal, K. Melo, T. Horvat, N. K. Mahabadi, S. Tsitkov, J.-M. Cabelguen, and A. J. Ijspeert, "From cineradiography to biorobots: an approach for designing robots to emulate and study animal locomotion," *Journal of The Royal Society Interface*, vol. 13, no. 119, p. 20151089, 2016.

[2] U. Saranli, M. Buehler, and D. E. Koditschek, "Rhex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001.

[3] B. Dynamics, "Spot by boston dynamics."

[4] J. Liu, Y. Tong, and J. Liu, "Review of snake robots in constrained environments," *Robotics and Autonomous Systems*, vol. 141, p. 103785, 2021.

[5] J. Gray, "Studies in animal locomotion: I. the movement of fish with special reference to the eel," *Journal of experimental biology*, vol. 10, no. 1, pp. 88–104, 1933.

[6] S. Hirose, "Biologically inspired robots," *Snake-Like Locomotors and Manipulators*, 1993.

[7] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, *Snake robots: modelling, mechatronics, and control.* Springer, 2013.

[8] F. Boyer, M. Porez, and W. Khalil, "Macro-continuous computed torque algorithm for a three-dimensional eel-like robot," *IEEE transactions on robotics*, vol. 22, no. 4, pp. 763–775, 2006.

[9] E. Kelasidi, K. Y. Pettersen, J. T. Gravdahl, and P. Liljebäck, "Modeling of underwater snake robots," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4540–4547, 2014.

[10] K. Y. Pettersen, "Snake robots," *Annual Reviews in Control*, vol. 44, pp. 19–44, 2017.

[11] W. Khalil, G. Gallot, and F. Boyer, "Dynamic modeling and simulation of a 3-d serial eel-like robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1259–1268, 2007.

[12] L. Paull, S. Saeedi, M. Seto, and H. Li, "Auv navigation and localization: A review," *IEEE Journal of oceanic engineering*, vol. 39, no. 1, pp. 131–149, 2013.

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

[14] A. Wiens and M. Nahon, "Optimally efficient swimming in hyper-redundant mechanisms: control, design, and energy recovery," *Bioinspiration & biomimetics*, vol. 7, no. 4, p. 046016, 2012.

[15] M. Nonhoff, P. N. Köhler, A. M. Kohl, K. Y. Pettersen, and F. Allgöwer, "Economic model predictive control for snake robot locomotion," in *Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 8329–8334, 2019.

[16] P. Liljeback, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Hybrid modelling and control of obstacle-aided snake robot locomotion," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 781–799, 2010.

[17] T. Kano, R. Yoshizawa, and A. Ishiguro, "Tegotae-based decentralised control scheme for autonomous gait transition of snake-like robots," *Bioinspiration & biomimetics*, vol. 12, no. 4, p. 046009, 2017.

[18] H. Date and Y. Takita, "Adaptive locomotion of a snake like robot based on curvature derivatives," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3554–3559, 2007.

[19] T. Sato, W. Watanabe, and A. Ishiguro, "An adaptive decentralized control of a serpentine robot based on the discrepancy between body, brain and environment," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp. 709–714, 2010.

[20] H. Yamada and S. Hirose, "Approximations to continuous curves of active cord mechanism made of arc-shaped joints or double joints," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp. 703–708, 2010.

[21] J. Gray, "The mechanism of locomotion in snakes," *Journal of experimental biology*, vol. 23, no. 2, pp. 101–120, 1946.

[22] S. Ma, "Analysis of snake movement forms for realization of snake-like robots," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 4, pp. 3007–3013, 1999.

[23] B. R. Moon and C. Gans, "Kinematics, muscular activity and propulsion in gopher snakes," *Journal of Experimental Biology*, vol. 201, no. 19, pp. 2669–2684, 1998.

[24] D. L. Hu, J. Nirody, T. Scott, and M. J. Shelley, "The mechanics of slithering locomotion," *Proceedings of the National Academy of Sciences*, vol. 106, no. 25, pp. 10081–10085, 2009.

[25] K. A. McIsaac and J. P. Ostrowski, “Motion planning for anguilliform locomotion,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 637–652, 2003.

[26] H. Marvi, J. Bridges, and D. L. Hu, “Snakes mimic earthworms: propulsion using rectilinear travelling waves,” *Journal of the Royal Society Interface*, vol. 10, no. 84, p. 20130188, 2013.

[27] H. Marvi, C. Gong, N. Gravish, H. Astley, M. Travers, R. L. Hatton, J. R. Mendelson III, H. Choset, D. L. Hu, and D. I. Goldman, “Sidewinding with minimal slip: Snake and robot ascent of sandy slopes,” *Science*, vol. 346, no. 6206, pp. 224–229, 2014.

[28] H. Marvi and D. L. Hu, “Friction enhancement in concertina locomotion of snakes,” *Journal of the Royal Society Interface*, vol. 9, no. 76, pp. 3067–3080, 2012.

[29] A. A. Transeth, K. Y. Pettersen, and P. Liljebäck, “A survey on snake robot modeling and locomotion,” *Robotica*, vol. 27, no. 7, pp. 999–1015, 2009.

[30] R. Ariizumi and F. Matsuno, “Dynamic analysis of three snake robot gaits,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1075–1087, 2017.

[31] S. Toyoshima and F. Matsuno, “A study on sinus-lifting motion of a snake robot with energetic efficiency,” in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, pp. 2673–2678, 2012.

[32] S. Toyoshima, M. Tanaka, and F. Matsuno, “A study on sinus-lifting motion of a snake robot with sequential optimization of a hybrid system,” *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 139–144, 2013.

[33] M. Javaheri Koopaee, C. Pretty, K. Classens, and X. Chen, “Dynamical modelling and control of snake-like motion in vertical plane for locomotion in unstructured environments,” in *Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 59292, p. V009T12A004, American Society of Mechanical Engineers, 2019.

[34] L. Rai and S. J. Kang, “Multi-thread based synchronization of locomotion control in snake robots,” in *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA’05)*, pp. 559–562, 2005.

[35] K. Walton, “The oblique compression of two elastic spheres,” *Journal of the Mechanics and Physics of Solids*, vol. 26, no. 3, pp. 139–150, 1978.

[36] B. C. Jayne, “Muscular mechanisms of snake locomotion: an electromyographic study of lateral undulation of the florida banded water snake (nerodia fasciata) and the yellow rat snake (elaphe obsoleta),” *Journal of Morphology*, vol. 197, no. 2, pp. 159–181, 1988.

[37] J. Gray and H. Lissmann, "The kinetics of locomotion of the grass-snake," *Journal of Experimental Biology*, vol. 26, no. 4, pp. 354–367, 1950.

[38] H. Marvi, D. Dimenichi, R. Chrystal, J. Mendelson, D. Goldman, D. Hu, Z. A. Collaboration, *et al.*, "Sidewinding snakes on sand," in *Proceedings of the APS Division of Fluid Dynamics Meeting Abstracts*, pp. H32–007, 2012.

[39] J. L. Tingle, "Facultatively sidewinding snakes and the origins of locomotor specialization," *Integrative and Comparative Biology*, vol. 60, no. 1, pp. 202–214, 2020.

[40] P. E. Schiebel, H. C. Astley, J. M. Rieser, S. Agarwal, C. Hubicki, A. M. Hubbard, K. Diaz, J. R. Mendelson III, K. Kamrin, and D. I. Goldman, "Mitigating memory effects during undulatory locomotion on hysteretic materials," *Elife*, vol. 9, p. e51412, 2020.

[41] H. Lissmann, "Rectilinear locomotion in a snake (boa occidentalis)," *Journal of Experimental Biology*, vol. 26, no. 4, pp. 368–379, 1950.

[42] A. H. Chang and P. A. Vela, "Shape-centric modeling for control of traveling wave rectilinear locomotion on snake-like robots," *Robotics and Autonomous Systems*, vol. 124, p. 103406, 2020.

[43] S. Manzoor, U. Khan, and I. Ullah, "Serpentine and rectilinear motion generation in snake robot using central pattern generator with gait transition," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 44, pp. 1093–1103, 2020.

[44] S. Yu, S. Ma, B. Li, and Y. Wang, "An amphibious snake-like robot with terrestrial and aquatic gaits," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pp. 2960–2961, 2011.

[45] E. Kelasidi, K. Y. Pettersen, J. T. Gravdahl, *et al.*, "Modeling and propulsion methods of underwater snake robots," in *Proceedings of the 2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 819–826, 2017.

[46] M. Inazawa, T. Takemori, M. Tanaka, and F. Matsuno, "Motion design for a snake robot negotiating complicated pipe structures of a constant diameter," in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8073–8079, 2020.

[47] T. Kano and A. Ishiguro, "Obstacles are beneficial to me! scaffold-based locomotion of a snake-like robot using decentralized control," in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3273–3278, 2013.

[48] S. Hirose and Y. Umetani, "Kinematic control of active cord mechanism with tactile sensors," *Transactions of the Society of Instrument and Control Engineers*, vol. 12, no. 5, pp. 543–547, 1976.

[49] M. Nakajima, M. Tanaka, K. Tanaka, and F. Matsuno, "Motion control of a snake robot moving between two non-parallel planes," *Advanced Robotics*, vol. 32, no. 10, pp. 559–573, 2018.

[50] F. Barazandeh, B. Bahr, and A. Moradi, "How self-locking reduces actuators torque in climbing snake robots," in *Proceedings of the 2007 IEEE/ASME international conference on advanced intelligent mechatronics*, pp. 1–6, 2007.

[51] A. Shapiro, A. Greenfield, and H. Choset, "Frictional compliance model development and experiments for snake robot climbing," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp. 574–579, 2007.

[52] A. Kuwada, S. Wakimoto, K. Suzumori, and Y. Adomi, "Automatic pipe negotiation control for snake-like robot," in *Proceedings of the 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 558–563, 2008.

[53] I. Virgala, M. Kelemen, E. Prada, M. Sukop, T. Kot, Z. Bobovskỳ, M. Varga, and P. Ferenčík, "A snake robot for locomotion in a pipe using trapezium-like travelling wave," *Mechanism and Machine Theory*, vol. 158, p. 104221, 2021.

[54] J. E. Colgate and K. M. Lynch, "Mechanics and control of swimming: A review," *IEEE journal of oceanic engineering*, vol. 29, no. 3, pp. 660–673, 2004.

[55] M. J. Lighthill, "Large-amplitude elongated-body theory of fish locomotion," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 179, no. 1055, pp. 125–138, 1971.

[56] G. I. Taylor, "Analysis of the swimming of long and narrow animals," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 214, no. 1117, pp. 158–183, 1952.

[57] C. E. Jordan, "Coupling internal and external mechanics to predict swimming behavior: a general approach," *American Zoologist*, vol. 36, no. 6, pp. 710–722, 1996.

[58] J. Chen, W. Friesen, and T. Iwasaki, "Mechanisms underlying rhythmic locomotion: body–fluid interaction in undulatory swimming," *Journal of Experimental Biology*, vol. 214, no. 4, pp. 561–574, 2011.

[59] W. Khalil, G. Gallot, O. Ibrahim, and F. Boyer, "Dynamic modeling of a 3-d serial eel-like robot," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1270–1275, 2005.

[60] K. A. Morgansen, B. I. Triplett, and D. J. Klein, “Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles,” *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1184–1199, 2007.

[61] E. Kelasidi, K. Y. Pettersen, P. Liljebäck, and J. T. Gravdahl, “Locomotion efficiency of underwater snake robots with thrusters,” in *Proceedings of the 2016 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, pp. 174–181, 2016.

[62] J. Sverdrup-Thygeson, E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl, “Modeling of underwater swimming manipulators,” *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 81–88, 2016.

[63] E. Kelasidi, K. Y. Pettersen, P. Liljebäck, and J. T. Gravdahl, “Integral line-of-sight for path following of underwater snake robots,” in *Proceedings of the 2014 IEEE Conference on Control Applications (CCA)*, pp. 1078–1085, 2014.

[64] K. A. McIsaac and J. P. Ostrowski, “Open-loop verification of motion planning for an underwater eel-like robot,” in *Proceedings of the Experimental Robotics VII*, pp. 271–280, Springer, 2001.

[65] M. Porez, F. Boyer, and A. J. Ijspeert, “Improved lighthill fish swimming model for bio-inspired robots: Modeling, computational aspects and experimental comparisons,” *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1322–1341, 2014.

[66] K. A. McIsaac and J. P. Ostrowski, “A framework for steering dynamic robotic locomotion systems,” *The International Journal of Robotics Research*, vol. 22, no. 2, pp. 83–97, 2003.

[67] M. Alamir, M. El Rafei, G. Hafidi, N. Marchand, M. Porez, and F. Boyer, “Feedback design for 3d movement of an eel-like robot,” in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp. 256–261, 2007.

[68] M. El Rafei, M. Alamir, M. Porez, N. Marchand, and F. Boyer, “Motion control of a three-dimensional eel-like robot without pectoral fins,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 750–755, 2008.

[69] E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl, “Energy efficiency of underwater snake robot locomotion,” in *Proceedings of the 2015 23rd Mediterranean Conference on Control and Automation (MED)*, pp. 1124–1131, 2015.

[70] X. Jiang, F. Yang, and S. Shi, “Design and full-link trajectory tracking control of underwater snake robot with vector thrusters under strong time-varying disturbances,” *Ocean Engineering*, vol. 266, p. 113012, 2022.

[71] B. Xu, M. Jiao, X. Zhang, and D. Zhang, "Path tracking of an underwater snake robot and locomotion efficiency optimization based on improved pigeon-inspired algorithm," *Journal of Marine Science and Engineering*, vol. 10, no. 1, p. 47, 2022.

[72] E. Kelasidi, P. Liljeback, K. Y. Pettersen, and J. T. Gravdahl, "Innovation in underwater robots: Biologically inspired swimming snake robots," *IEEE robotics & automation magazine*, vol. 23, no. 1, pp. 44–62, 2016.

[73] E. Müller, P. N. Köhler, K. Y. Pettersen, and F. Allgöwer, "Economic model predictive control for obstacle-aided snake robot locomotion," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9702–9708, 2020.

[74] Y. Qiu and H. Date, "A low computation-cost locomotion control for underwater snake robot based on monte carlo model predictive control and curvature derivative control," *Advanced Robotics*, vol. 38, no. 11, pp. 770–783, 2024.

[75] Y. Qiu and H. Date, "Monte carlo model predictive control for underwater snake robot locomotion," in *Preprints of the 22nd IFAC World Congress*, pp. 6244–6247, 2023.

[76] Y. Qiu and H. Date, "Obstacle-aided locomotion for underwater snake robot using monte carlo model predictive control and curvature derivative control," in *Proceedings of the 2023 62nd Annual Conference of the Society of Instrument and Control Engineers (SICE)*, pp. 690–695, 2023.

[77] E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl, "A control-oriented model of underwater snake robots," in *Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 753–760, 2014.

[78] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control.* John Wiley & Sons, Ltd, 2011.

[79] D. Rollinson, Y. Bilgen, B. Brown, F. Enner, S. Ford, C. Layton, J. Rembisz, M. Schwerin, A. Willig, P. Velagapudi, *et al.*, "Design and architecture of a series elastic snake robot," in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4630–4636, 2014.

[80] D. Rollinson and H. Choset, "Gait-based compliant control for snake robots," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, pp. 5138–5143, 2013.

[81] J. Whitman, F. Ruscelli, M. Travers, and H. Choset, "Shape-based compliant control with variable coordination centralization on a snake robot," in *Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 5165–5170, 2016.

[82] T. Wang, J. Whitman, M. Travers, and H. Choset, "Directional compliance in obstacle-aided navigation for snake robots," in *Proceedings of the 2020 American Control Conference (ACC)*, pp. 2458–2463, 2020.

[83] H. Fukushima, T. Yanagiya, Y. Ota, M. Katsumoto, and F. Matsuno, "Model predictive path-following control of snake robots using an averaged model," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 6, pp. 2444–2456, 2020.

[84] D. D. Dunlap, E. G. Collins Jr, and C. V. Caldwell, "Sampling based model predictive control with application to autonomous vehicle guidance," in *Proceedings of the Florida Conference on Recent Advances in Robotics*, 2008.

[85] G. Williams, B. Goldfain, P. Drews, K. Saigol, J. M. Rehg, and E. A. Theodorou, "Robust sampling based model predictive control with sparse objective information.," in *Proceedings of the Robotics: Science and Systems*, vol. 14, p. 2018, 2018.

[86] S. Ohyama and H. Date, "Parallelized nonlinear model predictive control on gpu," in *Proceedings of the 2017 11th Asian Control Conference (ASCC)*, pp. 1620–1625, 2017.

[87] S. Nakatani and H. Date, "Swing up control of inverted pendulum on a cart with collision by monte carlo model predictive control," in *Proceedings of the 2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 1050–1055, 2019.

[88] M. Kato, H. Date, S. Nakatani, and A. Ohya, "Control of quadcopter considering collision with wall utilizing monte carlo model predictive control," *Transactions of the Society of Instrument and Control Engineers*, vol. 57, no. 9, pp. 379–390, 2021.

[89] H. Ando, S. Nakatani, and H. Date, "Trajectory generation of center of mass for biped robot by monte carlo model predictive control considering upper body contact with walls and constraints of zmp," *Transactions of the Society of Instrument and Control Engineers*, vol. 59, no. 3, pp. 162–175, 2023.

[90] M. Pincus, "A closed form solution of certain programming problems," *Operations Research*, vol. 16, no. 3, pp. 690–694, 1968.

[91] H. Date and Y. Takita, "Control of 3d snake-like locomotive mechanism based on continuum modeling," in *Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 47438, pp. 1351–1359, 2005.

[92] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A review on modelling, implementation, and control of snake robots," *Robotics and Autonomous systems*, vol. 60, no. 1, pp. 29–40, 2012.

[93] M. J. Travers, J. Whitman, P. E. Schiebel, D. I. Goldman, and H. Choset, "Shape-based compliance in locomotion.," in *Proceedings of the Robotics: Science and Systems*, vol. 12, 2016.

[94] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The linear complementarity problem*. SIAM, 2009.

[95] N. Y. Gus' kova, G. Makhortykh, and M. Shcheglova, "Inertia and drag of elliptic cylinders oscillating in a fluid," *Fluid dynamics*, vol. 33, no. 1, pp. 91–95, 1998.

Appendix A

Publications

1. Peer-reviewed journal article:

   Y. Qiu and H. Date, "A low computation-cost locomotion control for underwater snake robot based on monte carlo model predictive control and curvature derivative control," *Advanced Robotics*, vol. 38, no. 11, pp. 770–783, 2024.

2. Peer-reviewed international conference:

   Y. Qiu and H. Date, "Obstacle-aided locomotion for underwater snake robot using monte carlo model predictive control and curvature derivative control," in *Proceedings of the 2023 62nd Annual Conference of the Society of Instrument and Control Engineers (SICE)*, pp. 690-695, 2023.

3. Refereed international conference papers (Abstract Review):

   Y. Qiu and H. Date, "Monte carlo model predictive control for underwater snake robot locomotion," in *Preprints of the 22nd World Congress of the International Federation of Automatic Control (IFAC World Congress)*, pp. 6244-6247, 2023.