RECONFIGURABLE MODULAR SNAKE ROBOT LOCOMOTION VIA LEARNING BASED HYBRID MOTION CONTROL SYSTEM ARCHITECTURE

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

İlyas KOCAER

September 2022

Reconfigurable Modular Snake Robot Locomotion via Learning Based Hybrid Motion Control System Architecture
By İlyas KOCAER
September 2022

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Ömer MORGÜL(Advisor)

Hitay ÖZBAY

İsmail UYANIK

Approved for the Graduate School of Engineering and Science:

Orhan ARIKAN
Director of the Graduate School

ABSTRACT

RECONFIGURABLE MODULAR SNAKE ROBOT LOCOMOTION VIA LEARNING BASED HYBRID MOTION CONTROL SYSTEM ARCHITECTURE

İlyas KOCAER
M.S. in Electrical and Electronics Engineering
Advisor: Ömer MORGÜL
September 2022

Snake robots propose significant advantages especially for indeterminate, chaotic environments through their robustness and versatility against unforeseen conditions and scenarios. In addition to distinct locomotion characteristics of snake robots, their redundant structure provides also fault tolerant operation capacity. However, sophisticated and versatile locomotion characteristics and redundant body structure also bring difficulty for dynamic modelling and motion control of snake robots and, for this reason, generation of snake locomotion patterns have been an ongoing challenge. To address this point; a reconfigurable, modular snake robot is designed and modelled with fundamental electromechanical structure, joint and actuation subsystem and contact force modellings based on minimal requirements which are determined by presented mathematical analysis for snake locomotion. A hybrid motion control system architecture which is constituted with state-of-the-art reinforcement learning based algorithms and a cascaded PID controller which comprises command shaper and gain scheduling components is presented. While reinforcement learning based algorithms which indicate promising potential for generation of sophisticated behaviours are employed for generation of 2D snake gait patterns with corresponding reward function terms, locomotion capabilities are expanded to 3D space with the proposed cascaded PID architecture and possible high level planners. Various experimentations that cover comparison of different reinforcement learning algorithms, individual effects of specified reward function terms, locomotion of snakes which are composed by different number of modules, fault tolerant locomotion trainings for defective snake robots and realization of 3D operation scenarios are investigated. In this content, from a holistic perspective, future directions are drawn for potential physical realization of the electro-mechanical structure, mechanical design details for self-assembly mechanisms, further improvements of the training process and curriculum, and learning based multi-robot scenarios which cover swarms of differently configured snakes to realize collaborative tasks.

Keywords: Snake Robots, Modular and Reconfigurable Robots, Robot Architecture Modelling, Contact Force Modelling, Reinforcement Learning Based Motion Control, Cascaded PID Based Motion Control, Hybrid Motion Control System Architecture.

ÖZET

ÖĞRENME TEMELLİ MELEZ HAREKET KONTROL SİSTEM MİMARİSİ İLE YENİDEN YAPILANDIRILABİLİR MODÜLER YILAN ROBOT HAREKET KABİLİYETİ

İlyas KOCAER
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Ömer MORGÜL
Eylül 2022

Yılan robotlar, gürbüzlükleri ve çok yönlü operasyon kabiliyetleri sayesinde bilhassa belirsiz, kaotik ortamlara ve öngörülemeyen senaryolara yönelik kayda değer avantajlar sunmaktadırlar. Birbirinden farklı karakteristikte gerçekleyebildikleri hareket kontrol kabiliyetlerine ek olarak, minimum gereklerden fazlasını barındıran mimarileri sayesinde olası hatalara ve arızalara karşı dirençli şekilde operasyon gösterme kapasitesine sahiptirler. Elektromekanik mimarileri açısından yılan robotların sunmakta olduğu söz konusu üstünlüklere rağmen komplike ve çok yönlü hareket karakteristikleri dolayısıyla hareket kontrolü güncel zamana değin süregelen bir problem olarak ortaya çıkmaktadır. Bu noktayı ele almak üzere, yılan robot lokomasyonuna yönelik matematiksel analizler temel alınarak minimum gerekler belirlenmiştir. Belirlenen gerekler temelinde yeniden yapılandırılabilir modüler bir yılan robot tasarımı ve ilgili tasarıma yönelik temel elektromekanik mimarinin, tahrik bileşenlerinin ve sürtünme kuvvetinin modelleme işlemleri gerçeklenmiştir. Pekiştirmeli öğrenme temelli algoritmalar ve komut şekillendirme, kazanç planlama bileşenlerini barındıran katmanlı PID kontrolcü tasarımı ile bir hibrit hareket kontrol sistem mimarisi sunulmaktadır. Bu kapsamda, iki boyuttaki sofistike yılan hareket karakteristikleri pekiştirmeli öğrenme temelli algoritmalar ile elde edilirken, üst seviye planlama algoritmaları ve katmanlı PID kontrolcü mimarisi ile hareket kabiliyeti üç boyutlu uzaya taşınmıştır. Gerçekleştirilen incelemeler kapsamında farklı pekiştirmeli öğrenme algoritmalarına dair kıyaslamalar, ödül fonksiyonunu oluşturan bileşenlerin hareket karakteristikleri üzerine bireysel etkileri, farklı sayıda modüler bileşenden oluşan yılan robotların hareket karakteristikleri, arızalı bileşenler içeren robotların hata toleranslı eğitimleri ve üç boyutta olası operasyon

senaryoları ele alınmaktadır. Ek olarak, bütüncül bir bakış açısı ile, ilgili robotun elektromekanik mimarisinin fiziki olarak gerçeklenmesine, kendiliğinden birleşme mekanizmalarının mekanik tasarım detaylarına, eğitim sürecine ve izlencesine ve farklı işlevler özelinde konfigüre edilmiş işbirlikçi robot sürülerine yönelik öğrenme temelli çoklu robot senaryolarına yönelik gelecek güzergahlar çizilmiştir.

Anahtar sözcükler: Yılan Robotlar, Modüler ve Yeniden Yapılandırılabilir Robotlar, Robot Mimarisi Modelleme, Temas Kuvveti Modellemesi, Pekiştirmeli Öğrenme Temelli Hareket Kontrolü, Katmanlı PID Temelli Hareket Kontrolü, Melez Hareket Kontrol Sistem Mimarisi.

Acknowledgement

First and foremost, I would like to state my sincere gratitude to my advisor Prof. Dr. Ömer Morgül for his approach which provides freedom for exploration and understanding attitude that helps to overcome experienced difficulties. Thanks to the chance he has given to me, I could have opportunity to proceed one step further on the route that I have devoted my life.

I also would like to remark my special appreciation to my managers in Aselsan Inc., Dr. İhsan Özsoy for his visionary perspective, equitable approach and detailed evaluations, İlhan Başçuhadar for his visionary and instructive perspective, Mutlu Çanga for his supportive, thoughtful attitude and inspiring ideas and Abbas Şahin for his supportive approach in the favor of employees.

Besides, I would like to thank my colleagues from Aselsan Inc. and Bilkent University; Nusrettin Güleç, Tolga Yüksel, Serkan Zobar, Ergin Kurtulmuş, Alper Gül, Ali Nuri Güneşdoğdu, Emre Yazıcı, Mustafa Oğuz Yeğin, Rıdvan Çelik, Alper Topuz and Oğuz Altunkaş for their supports and irreplaceable companionship. I also would like to offer my deepest appreciation to Ali Nail İnal for his solidarity and permanent support whenever needed, independent from existing conditions.

Last but not least, I also would like to express my sincere appreciation to Prof. Dr. Mehmet Koyutürk for his encouraging and heartening support, Asst. Prof. Dr. Kenneth Weisbrode for all the invaluable subjects that I learned during his courses and discussions, Prof. Dr. Hitay Özbay and Asst. Prof. Dr. İsmail Uyanık for their constructive evaluations in the thesis committee, Cenk Demir for his friendship and Vedat Yazıcı for his enlightening guidance and confidence in me to discover misty horizons.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Snake robots have significant advantages especially in terms of terrain adaptability, diversity of possible operation duties and mediums, robustness and flexibility against unexpected, undesired scenarios among the robotics implementations which have a monotonically increasing impact in both scientific research and industrial area in addition to daily life. Variety and comprehensiveness of snake robot applications for distinct tasks arise from their potential to realize sophisticated locomotion characteristics and electromechanical characteristics. However, generation of the corresponding snake locomotion characteristics is an ongoing challenge for the aimed operation concept. In the scientific research area, it is taken into consideration in terms of different perspectives which can be put in order as electro-mechanical architecture design, motion control system design for generation of snake gait patterns and perception and planning capabilities.

For these reasons, in the scope of the thesis, fundamental focal points are determined as the implementation of learning based algorithms, presentation of a motion control system architecture which provides operation capability in both 2D and 3D environments via presented hybrid motion control system architecture which is followed by an investigation on mathematical background for snake locomotion and a simple design of modular, reconfigurable snake robot architecture with corresponding modelling operations.

1.1 Advantages and Disadvantages of Snake Robots

Snake robots whose architectures and capabilities can potentially reach out even beyond of biological diversity boundaries provide advantages especially for operations in constrained, indeterminate environments with their high adaptability and versatility potential through distinct locomotion characteristics. In addition to the diverse gait patterns, redundant body structures of snake robots which can also be designed in a modular configuration ensures fault tolerance and different capabilities which can be required depending on diversified operation conditions and aimed tasks. For a specific objective, sensors and payloads which are carried by the modules can be re-determined and the snake robot can be re-configured in accordance with the intended purpose thanks to its modular structure. Furthermore, in this way, collaborative snake robot swarms can also be formed with differently configured robots which is customized to realize a specific task. Additionally, as snake robots can be designed to operate in a 2D surface or 3D space, the medium where robot operates can also be diversified. A snake robot can potentially have amphibious characteristics and can operate both on loose or rigid solid surfaces and in liquid mediums. On the other hand, several disadvantages can be put in order for snake robots especially comparing with wheeled and legged robots. The primarily noticed incompetence is the velocity limits which inherently emerge due to electro-mechanical structure. For this reason, in relatively unconstrained, regular terrains and wide spaces, wheeled or legged robots can possibly predominate snake robots in terms of effectiveness of operational capabilities. As the other drawback, even though it also extends the capabilities and flexibility of snake robots, high number of degrees of freedom characteristic also brings difficulty in dynamics modelling, motion planning and control as this challenge will be addressed in Chapter 3. Additionally, although snake robots can be equipped with various sensors and different configurations can be practically composed depending on the specific requirements through a reconfigurable modular architecture design, payload capacity of snake robots is limited in terms of both weight and size.

1.2 Application Fields of Snake Robots

Snake robots, as aforementioned, provide significant advantages for operations in chaotic environments. They are especially used with exploration and inspection purposes in environments which can include constrained routes with different kinds of obstacles, narrow passages, unpredictable medium and surface conditions. In this content; caves, debris zones [27], radioactive or poisonous sites, accident or disaster regions [28] and even body of a living organism [29] can be operation environment for snake robots. In these operation environments, application range of snake robots extends from search and rescue operations to military purposes. High versatility and robustness of snake robots provide them to operate in challenging environmental conditions where wheeled/crawler or even legged robots are not able to function properly.

1.3 Operation Concepts of Snake Robots

Operation concepts of snake robots differentiate in terms of robotic gait patterns and operation environment properties with aimed mission in addition to actuation characteristics and body structure. Some snake robots are designed to operate on a 2D surface whereas some others are designed for operation in a 3D medium. Depending on the electro-mechanical structure and locomotion characteristics of the designed snake robot, operable environments are also implicitly determined. Based on the operation mission, characteristics of medium and surface can enclose one or several of the environmental conditions which can be put in order as; smooth surfaces, slope surfaces, rough terrains, corridors, pipes, regular or irregular pole-like structures and various obstacles where the operation medium can be loose or rigid solid, liquid or combination of solid and liquid elements. For example; irregular rocky areas, trees and their branches, poles and pipes with various diameters, debris zones, human-made obstacles like wire netting or underwater operations are possible challenges that cannot be overcome easily in an intended task through legged, wheeled/crawler or flying robots. Snake robots can potentially impersonate climbing [2] [4], floating [13], crawling [9] and rolling [6] [7] [14] vehicles. Furthermore, they can be used as robotic manipulators on a static or dynamic platform as well [3]. Modular and redundant structure of snake robots provides reconfiguration potentials and fault tolerance against malfunction possibilities as these properties can provide beneficial operation concepts especially in hazardous environments. Furthermore, operation potentials of snake robots, which already provides significant opportunities, can be extended with the progress in material science especially by evaluating in a soft robotics concept. In this way, diversity and strength of snake robots in locomotion characteristics can also be reflected to their electro-mechanical architectures.

1.4 Snake Robot Examples From Industry and Academy

Some notable snake robots with distinct capabilities from both academic and industrial applications are presented in Figure 1.1. *OriSnake* is a continuously deformable snake robot which is formed by origami-inspired continuum modules [1]. Modular snake robot which is shown in image *b* can climb inside of a pipe [2]. *ETR* robot is developed especially for search and rescue tasks. It can be used as a robotic manipulator on other vehicles to collaboratively realize an intended task [3]. *Unified Snake Robot* has 3D gait capabilities and can climb poles, pipes, trees and their branches. It can also keep its existing position without consuming any power through its bistable brake system [4]. *S5* snake robot is developed for search and rescue purposes. It holds passive wheels on the underside and has high number of modules which corresponds to high degrees of freedom and redundancy [5]. *PolyBot* is a modular, self-reconfigurable robot which provides versatility and robustness through redundancy and self-repair properties [6] [7]. Pipe inspection modular robot which is shown in image *g* can move throughout different pipes and elbows. It can also detect obstacles and walls [8]. *MOIRA2* is also developed as an inspection robot for rescue missions but with a different architecture. It is composed by four body modules and holds crawler at the faces to improve its locomotion ability in the rubble [9]. *OmniTread OT-4* robot comprises seven segments which are covered by motorised drive tracks in all sides and pneumatic bellows for actuation of the joints. It aims to climb obstacles which is higher than the robot height, propel itself inside pipes and operate in challenging terrain conditions [10]. The presented robots in images from j to m are different generations of the same *Active Cord Mechanism* snake robot family which are progressively developed on their predecessors. Gradually, both their electro-mechanical architecture and locomotion capabilities with various gait types are improved and diversified. Even though there are also intermediate forms in the development process, at this point, only *ACM III* [11], *ACM-R3* [12], *ACM-R5* [13] and *ACM-R7* [14] are introduced as the representatives of fundamental milestones. *ACM III* is one of the oldest ancestors of snake robots which are developed since 1972 [30]. Glide propulsion approach is realized with *ACM III* and further capabilities, and new degrees of freedom, different gait patterns are proposed with the subsequent designs [11] *ACM-R3* snake robot is equipped with large passive wheels that wraps its body and different gait patterns are examined along with it [12]. *ACM-R5* is an amphibious robot which can operate both in water and on solid terrain. It comprises passive wheels on its six sides of modules for locomotion on solid surfaces [13]. *ACM-R7* robot can take loop form and correspondingly a loop gait which is named as serpenoid oval is proposed for its locomotion [14]. Snake robot with toroidal skin drive aims to improve speed specifications beyond the existing limits via continuous propulsive force which is provided by entire surface of the robot. Therefore, novel locomotion techniques as a combination of skeletal actuation and skin drive are obtained [15]. *Slim Slime Robot* is a pneumatically driven snake robot which has a flexibly deformable mechanical structure [16]. *Wheeko* and *Kulko* share the same internal electro-mechanical structure. *Wheeko* comprises twelve passive wheels which encircles its outer diameter. *Kulko* is covered with a spherical shell and holds contact force sensors. It is specifically developed for the purpose of obstacle-aided locomotion in uneven and cluttered surface conditions [17].

Figure 1.1: Snake robot examples with distinct architectures and capabilities. (a) OriSnake Origami Inspired Snake Robot [1], (b) Modular Snake Robot Climbing the Inside of a Pipe [2], (c) USAR-ETR Urban Search and Rescue Elephant-Trunk Like Snake Robot [3], (d) Unified Snake Robot [4], (e) S5 Snake Robot [5], (f) PolyBot Self-Reconfigurable Snake Robot [6] [7], (g) Pipe Inspection Modular Micro Snake Robot [8], (h) MOIRA2 Inspection Snake Robot [9], (i) OmniTread OT-4 Serpentine Robot [10], (j) ACM III [11], (k) ACM-R3 [12], (l) ACM-R5 Amphibious Snake Robot [13], (m) ACM-R7 Loop Forming Snake Robot [14], (n) Snake Robot with Toroidal Skin Drive [15], (o) Slim Slime Robot (SSR) Flexibly Deformable Snake Robot [16], (p) Wheeko Snake Robot [17], (q) Kulko Snake Robot for Uneven and Cluttered Environments [17].

1.5 Purpose and Contributions of the Thesis

For snake robots, motion control approaches can be classified under two essential categories which are constructed on dynamics and kinematics characteristics and morphological characteristics. In this content; central pattern generators, sliding mode controllers and PID controllers are proposed for snake-like robots which are developed in academic and industrial applications including [1] - [17]. Even though reinforcement learning is a prominent candidate which can cope with the generation of diversity and sophisticated characteristics of snake locomotion, it is not sufficiently investigated for snake robots until the present time and scope of the current researches and applications are quite limited. In contrary to legged robots and especially quadrupedal robots where reinforcement learning based motion control methodologies are investigated and practiced relatively in more detail especially after 2018, realized researches for snake robots are in the emergence phase in terms of both quantitatively and qualitatively. The purpose and some of the major contributions of the thesis concentrate on this deficient point.

In the scope of the thesis;

1. A reinforcement learning based motion control architecture is proposed for generation of gait patterns in 2D environment for snake robots which are formed by different number of modules.

2. For realization of locomotion through reinforcement learning, novel reward functions are designed for the desired gait patterns and effects of each term on locomotion characteristics are investigated.

3. Functionality of the proposed and constructed learning based motion control architecture is verified with 5, 7 and 9 modules snake robots by realizing commanded motion control objectives.

4. As a fault tolerant locomotion training, robustness of proposed motion control architecture is also tested and verified with a defective snake robot which includes various malfunctions in its actuation joints.

5. Functionality of snake robot locomotion, with trained reinforcement learning algorithms, is also verified with unseen surface conditions. Therefore, generalizability of the proposed motion control system is validated for operations in different conditions.

6. Operation capabilities of the snake robots are expanded to 3D space by a cascaded PID architecture with command shaper and gain scheduling components where different high level planners can also be integrated conveniently.

7. Consequently, a hybrid novel motion control system architecture is presented for snake robots which is versatile for realization of various high level motion plannings and robust against possible faults and dysfunctionalities.

Afterwards, as the finalizing marks, the future directions are drawn in a holistic view which covers different aspects of snake robot applications including multi-robot scenarios that cover collaborative behaviours and reconfiguration tasks. In this content, learning based swarm robotics applications are also evaluated.

1.6 Overview of the Thesis

Chapter 2 describes anatomy and locomotion characteristics of biological snakes and mathematical analysis for snake robot locomotion with corresponding controllability analyzes and propulsive forces synthesis. In this content, minimal requirements to realize proper snake locomotion are discussed and determined.

Chapter 3 portrays fundamental features of electro-mechanical architecture of trained snake robot and modelling of re-configurable modular structure, actuation and joint mechanisms, contact force modelling with passive wheeled structure to obtain anisotropic friction.

Chapter 4 clarifies the reasoning and justification for the use of reinforcement learning to generate snake locomotion behavior and implemented reinforcement learning algorithms among existing options. Subsequently, design details of algorithmic architecture, observation and action spaces, hyperparameter settings and reward function with individual roles of corresponding components are described.

Chapter 5 presents obtained training results for two different reinforcement learning algorithms and snake robots which are constructed by 5, 7, 9 modules and 11 modules which include defective components. In this content, resulting snake locomotion characteristics are separately illustrated for each case and investigations with trained robots are extended with experimentations on unexperienced surface conditions. In addition to reinforcement learning motion control scheme which corresponds to generation of 2D gait patterns for proper snake locomotion, motion control system design which extends the operation capability to 3D medium is presented and several sample scenarios are realized with 7 and 11 modules snake robots. Consequently, overall hybrid motion control system architecture which includes *Twin Delayed Deep Deterministic Policy Gradient (TD3)* and *Cascaded PID* with a *command shaper*, *gain scheduling* and *notch filter* components which will operate based on the generated commands by a higher level planner are presented.

Chapter 6 concludes the thesis with evaluations and discussions for obtained outcomes by taking into account both positive and negative aspects. Reasonings for resulting outcomes are underlined and the points which are open for further improvements are indicated.

Chapter 7 draws the future directions which are aimed to realize for simulation environment, training process, high level motion planners for 3D tasks and multi-robot scenarios which cover collaborative behaviours and reconfiguration tasks to take the obtained results one step further.

Chapter 2

Snake Locomotion Investigation

The snake locomotion characteristics differentiate depending on the corresponding snake structure in both biological snakes and electro-mechanical snake robots. Although there are similarities in fundamental ruling features among biological snakes and between biological and electro-mechanical snakes, locomotion patterns and gait types are indirectly determined depending on the corresponding architecture. The essential variables at this point can be put in order as the joint structures and numbers, contact properties between snake body and locomotion surface with corresponding limitations and degrees of freedom, properties of the material that covers and shapes the snake body and motion control and planning system architecture. In snake locomotion, the propulsive force which carries forward the snake body is obtained via repeating motions. In biological snakes, required coordinated repeating motions is generated through neural impulses and activated local muscle groups which bend the snake body whereas, in electro-mechanical snake robots, corresponding complex motions are realized via applied torque commands to the integrated joints of the snake architecture.

In this content, through the following subsections, essential features are defined and requirements are determined for a proper snake locomotion in terms of structure of the snake, fundamental snake locomotion patterns, contact properties between surface and snake body and generation of propulsive forces.

2.1 The Anatomy and Locomotion Behaviour of Biological Snakes

Typical skeletal structure of a snake consists of a skull, 130 to 500 vertebrae and ribs which are attached to each one of vertebrae as it can be seen from Figure 2.1. Vertebrae components corresponds to movable joints of a snake where relative rotation between consecutive components are approximately between 10° and 20° in vertical axis and limited with only a few degrees in horizontal axis [18]. However, through high number of consecutive vertebrae components, resultant sum of the limited rotations results in remarkable flexibility and mobility capabilities of snakes. Body movements such as bending, extension or contraction are realized through the muscles which are attached to ribs and their contraction or relaxation determines the characteristics of locomotion [18].

The other fundamental component in the anatomy of snakes is scales which cover the body of snakes as it can be seen in the Figure 2.2. In addition to physical protection purpose of the scales, they also provide *anisotropic friction* characteristics by creating higher friction coefficient in transversal direction comparing with in tangential direction of the body.

Even though there exist anatomical differences between snake species, which ultimately leads differences in locomotion types, most common locomotion types can be investigated under four major categories based on the illustrated anatomy. They can be put in order as *lateral undulation*, *concertina*, *rectilinear crawling* and *sidewinding* [18] [28]. However, four primary locomotion types can be extended further with *burrowing*, *jumping*, *sinus-lifting*, *skidding*, *swimming*, *climbing* gait types [28]. Moreover, *lateral undulation* and *concertina* can be respectively divided into five and four sub-categories based on diversifying motor patterns. Lateral undulation can be analyzed with *terrestrial lateral undulation*, *forward aquatic lateral undulation*, *backward aquatic lateral undulation*, *lateral undulation with a ventrolateral keel* and *arboreal lateral undulation* locomotion types while *concertina* types can be put in order as *flat-surface concertina*, *tunnel concertina*, *arboreal concertina with alternate bends* and *arboreal concertina*

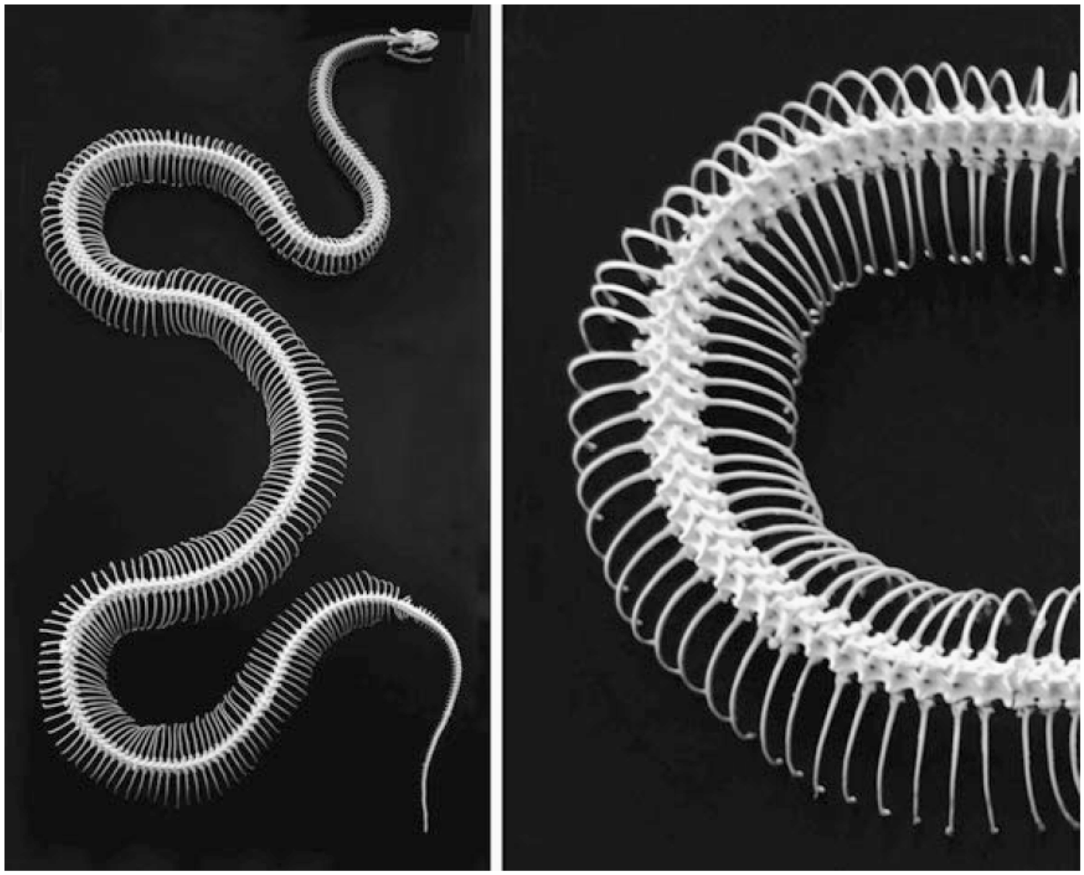

Figure 2.1: Skeleton of a snake consisting of vertebrae and ribs [18].

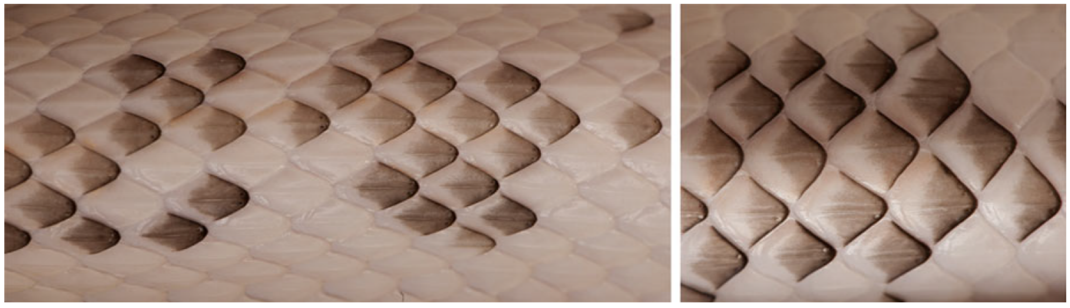

Figure 2.2: Snake skin which is covered by scales [18].

with helical wrapping [31]. In addition to these locomotion characteristics, depending on the designed robotic architecture, unlimited gait types which are not observed in biological snakes can be generated for aimed objective. Corresponding gait types can be definite based on a specifically dictated behavior or can be indefinite by focusing only on the objective and leaving required characteristics to be determined by the implemented algorithm.

For snake locomotion, as stated, most common four locomotion characteristics can be described as below and illustrated in Figure 2.3;

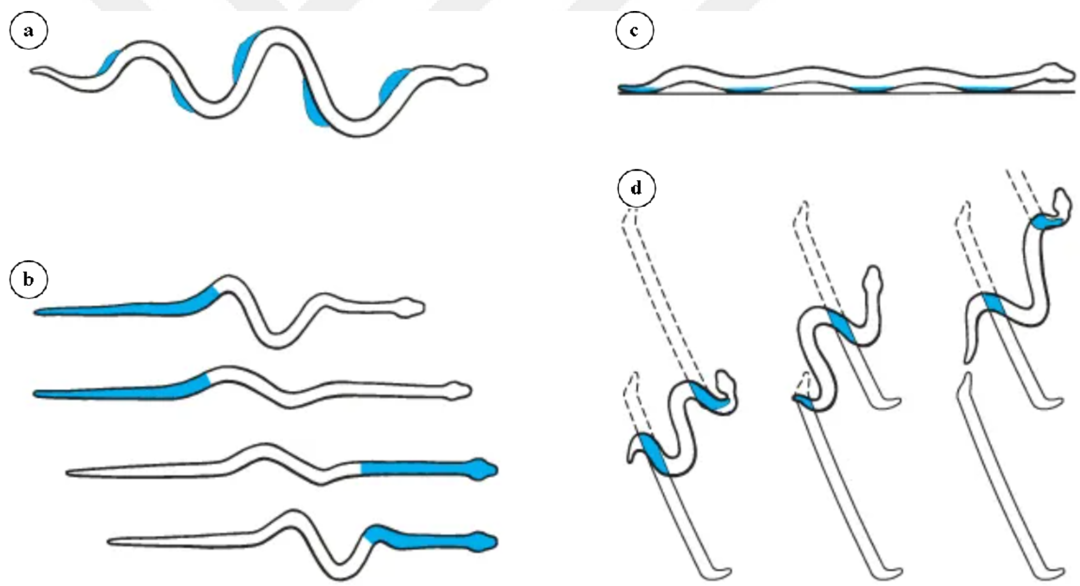

Figure 2.3: Fundamental snake locomotion patterns [19].
(a) Lateral undulation, (b) Concertina, (c) Rectilinear crawling, (d) Sidewinding.

Lateral Undulation: For both biological and electro-mechanical snakes, *lateral undulation* is the most common locomotion type since it covers a broad functionality for various cases and environments with relatively smooth and higher velocity values. During lateral undulation, continuous sinuous waves are generated by the entire body of the snake to propagate resultant forward propulsion effect. Lateral undulation locomotion form is not appropriate for exceedingly smooth, low-friction slippery contact surfaces whereas it operates most efficiently in rough ground contact surfaces.

Concertina: *Concertina* locomotion is especially required for narrow spaces, although it is an inefficient mode of movement, where lateral undulation cannot be realized properly due to limited motion range. In concertina gait, back and front parts of snake body are subsequently used as an anchor for the narrow environment and the free part of the body is the extended to proceed forward.

Rectilinear Crawling: *Rectilinear crawling* is a substantially slow form of locomotion and it is generated by biological snakes as reciprocating the muscles from the ribs attached to the skin where the edges of scales located on the underside of the body is used as the anchor points to proceed forward approximately on a straight line.

Sidewinding: *Sidewinding* locomotion type is a continuous transverse oscillation which makes movement possible on uncluttered and low shear surfaces where motion generation by other gait types is inefficient or completely impossible. Through sidewinding gait, long distance travelling becomes possible for snakes even on loose terrain. To realize this gait, head part and rest of the body consecutively act as anchor on the ground and pulling component for the body.

2.2 Controllability Analysis of Snake Robots

In the following steps, controllability with resultant propulsive force analyzes are performed and correspondingly minimum requirements are determined in terms of contact friction properties and number of modules which forms the architecture for realization of proper snake locomotion. The analyzes are realized for planar surface locomotion since it determines the minimal requirements for generalized terrain conditions and with the assumption of viscous contact friction since it provides sufficient approximation and corresponding analyzes take a more feasible form to conclude corresponding analyzes [17, 18]. Friction characteristics which depends on both contact surface of the snake and the terrain properties is one of the fundamental components which directly affects locomotion characteristics. As it is shown for planar surface locomotion in article [32] and inclined surface locomotion in article [33], optimal snake behavior significantly changes depending on the tangential and normal components of friction force.

The notation for the corresponding analyzes in the following sections are presented with required definitions in Table 2.1.

Definition	Notation
Number of modules	N
Mass of each module	m
Link length	$2l$
Angle between module i and the global x axis	θ_i
Global coordinates of the module i center of mass	(x_i, y_i)
Global coordinates of the robot center of mass	(p_x, p_y)
Viscous friction coefficient	c
Ground friction force on module i	$(\mathbf{f}_{R,x,i}, \mathbf{f}_{R,y,i})$

Table 2.1: Snake Robot and Locomotion Characterization Parameters

2.2.1 Controllability with Isotropic Viscous Friction and Anisotropic Viscous Friction

For a modular snake robot, the isotropic viscous friction which affects on the module i can be stated as the following equation.

$$\mathbf{f}_{R,i} = -c \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = -c \begin{bmatrix} \dot{p_x} - \sigma_i \mathbf{S}_\theta \dot{\theta} \\ \dot{p_y} + \sigma_i \mathbf{C}_\theta \dot{\theta} \end{bmatrix} \tag{2.1}$$

where;

$$\begin{aligned}
\mathbf{S}_\theta &= \text{diag}(\sin \boldsymbol{\theta}) \\
\sin \boldsymbol{\theta} &= [\sin \theta_1, ..., \sin \theta_N]^T \\
\mathbf{C}_\theta &= \text{diag}(\cos \boldsymbol{\theta}) \\
\cos \boldsymbol{\theta} &= [\cos \theta_1, ..., \cos \theta_N]^T \\
\sigma_i &= [a_1, a_2, ..., a_{i-1}, \frac{a_i + b_i}{2}, b_{i+1}, b_{i+2}, ..., b_N] \\
a_i &= \frac{l(2i-1)}{N} \\
b_i &= \frac{l(2i-1-2N)}{N}
\end{aligned}$$

The equation which describes acceleration of the center of mass can be stated as;

$$\begin{bmatrix} \ddot{p_x} \\ \ddot{p_y} \end{bmatrix} = \frac{1}{Nm} \begin{bmatrix} \mathbf{e}^T \mathbf{f}_{R,x} \\ \mathbf{e}^T \mathbf{f}_{R,y} \end{bmatrix} = \frac{1}{Nm} \begin{bmatrix} \sum_{i=1}^N \mathbf{f}_{R,x,i} \\ \sum_{i=1}^N \mathbf{f}_{R,y,i} \end{bmatrix} \tag{2.2}$$

where;

$$\mathbf{e} = [1, 1, ..., 1]^T$$

When equation 2.1 is inserted into equation 2.2, center of mass acceleration of the robot is obtained as;

$$\begin{bmatrix} \ddot{p_x} \\ \ddot{p_y} \end{bmatrix} = \frac{c}{Nm} \begin{bmatrix} -N\dot{p_x} + (\sum_{i=1}^{N} \sigma_i)\mathbf{S}_\theta \dot{\theta} \\ -N\dot{p_y} - (\sum_{i=1}^{N} \sigma_i)\mathbf{C}_\theta \dot{\theta} \end{bmatrix} = -\frac{c}{m} \begin{bmatrix} \dot{p_x} \\ \dot{p_y} \end{bmatrix} \tag{2.3}$$

Through σ definition in Equation 2.1, since it can be shown that $\sum_{i=1}^{N} \sigma_i = 0$, Equation 2.3 indicates that center of mass acceleration is proportional to center of mass velocity. For this reason, in the cases of snake robot starts from the stationary conditions, it is not possible to obtain a non-zero acceleration for center of mass. Consequently, it is concluded that a snake robot on a planar surface is not controllable with applied torque commands to its joints if the effective friction force reflects isotropic characteristics. In other words, to control the position of the snake robot through a targeted route, center of mass of the snake robot must be accelerated. However, since center of mass acceleration is proportional to the center of mass velocity as indicated in Equation 2.3, when the robot starts from zero velocity initial condition, it is not possible to create non-zero center of mass acceleration [18] [34–36].

On the other hand, under anisotropic viscous friction, two viscous friction coefficients are needed to be defined as c_t and c_n which describe the friction force in the tangential and normal directions of the corresponding link. In anisotropic viscous friction characteristics, friction coefficients are not equal in tangential and normal directions $c_t \neq c_n$ whereas their equality $c_t = c_n$ reduces the friction characteristics to isotropic viscous friction. The viscous friction force on link i in the local link frame $\mathbf{f}_{R,i}^{link,i}$ can be stated as in Equation 2.4 [18].

$$\mathbf{f}_{R,i}^{link,i} = -\begin{bmatrix} c_t & 0 \\ 0 & c_n \end{bmatrix} \mathbf{v}_i^{link,i} \tag{2.4}$$

where $\mathbf{v}_i^{link,i}$ is the link velocity.

In the global frame, by re-arranging Equation 2.4, it can be presented as in Equation 2.5 [18].

$$\mathbf{f}_R = \begin{bmatrix} \mathbf{f}_{R,x} \\ \mathbf{f}_{R,y} \end{bmatrix} = - \begin{bmatrix} c_t {\mathbf{C}_\theta}^2 + c_n {\mathbf{S}_\theta}^2 & (c_t - c_n)\mathbf{S}_\theta \mathbf{C}_\theta \\ (c_t - c_n)\mathbf{S}_\theta \mathbf{C}_\theta & c_t {\mathbf{S}_\theta}^2 + c_n {\mathbf{C}_\theta}^2 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{Y}} \end{bmatrix} \tag{2.5}$$

However, equation of motion of the snake robot becomes significantly more complex with anisotropic friction characteristics. For this reason, Liljebäck *et al.* [18] evaluates controllability through partial feedback linearization of the model and computing Lie brackets of the system vector fields at an equilibrium point [34–36]. Since the intermediate steps of derivation operations are not in the scope of the thesis, they are not explicitly presented and left to personal initiative for investigation through indicated references. As the main results of the employed operations, it is stated that the realized analysis is valid when the snake robot has $N \geq 4$ links and it is concluded that a snake robot with $N \geq 4$ links which are affected by anisotropic ground friction force on a planar surface is locally strongly accessible from any equilibrium point.

Consequently, Liljebäck *et al.* [18] points out that a snake robot, on a planar surface, which is composed from $N \geq 4$ links and affected by anisotropic friction is controllable.

2.3 Propulsive Forces Under Anisotropic Friction

Total resultant force which acts on the snake robot where the forward direction is defined as the global positive x-axis, can be stated as the following equation when anisotropic viscous friction on each link and dynamics of the snake robot are included into the analysis. In the equation, θ_i corresponds to the angle between the global x-axis and link $i \in \{1, 2, 3...N\}$.

$$F_{prop} = -\sum_{i=1}^{N} F_x(\theta_i)\dot{\mathrm{x}}_i - \sum_{i=1}^{N} F_y(\theta_i)\dot{\mathrm{y}}_i \tag{2.6}$$

where $F_{\mathrm{x}}(\theta_i)$ and $F_{\mathrm{y}}(\theta_i)$ are defined with equations 2.7 and 2.8 respectively.

$$F_{\mathrm{x}}(\theta_i) = c_t \cos^2\theta_i + c_n \sin^2\theta_i \tag{2.7}$$

$$F_{\mathrm{y}}(\theta_i) = (c_t - c_n)\sin\theta_i \cos\theta_i \tag{2.8}$$

The stated two components of F_{prop} corresponds to the forward and normal directions of motion. As it can be interpreted from the individual equations of components, $F_{\mathrm{x}}(\theta_i)$ does not contribute to the forward propulsion of the robot. On the contrary, it causes a counter effect on the motion since it always creates positive outcomes which result in as a negative factor in the resultant force F_{prop} whereas propulsive force which ensures the forward locomotion is generated by the $F_{\mathrm{y}}(\theta_i)$ term (when $c_n > c_t$ for ordinary undulatory gait pattern) [18] [36].

Following this point that is sufficient to determine the minimum requirements which should be taken into consideration in design and experimentation phases, synthesis for propulsive snake motion based on an aimed gait pattern and modellings of the corresponding snake robot with significantly complex operations which arise due to sophisticated nature of snake locomotion and, followingly, motion controller design can be realized. However, as stated in the purposes of the thesis, reinforcement learning is employed to realize snake locomotion in consideration of its potential to overcome complicated problems. Therefore, necessity and burden of problem specific further analyzes are eliminated.

Chapter 3

Architecture and Modelling of the Designed Snake Robot

A modular and reconfigurable snake robot is designed for investigations and various experimentations of snake robot locomotion. Correspondingly, to realize aimed analyzes by implementing described algorithms in Chapter 4, electromechanical architecture of the snake robots with different number of modules are modelled in the simulation environment [37] with required sensory operations, actuation systems, filtering operations and contact force modelling. In the scope of design and modelling steps, fundamental points with corresponding backgrounds are described under the following subsections.

3.1 Modular and Reconfigurable Structure

The snake robot architecture which is employed in 2D and 3D locomotion experimentations is designed in a modular approach where each joint module is identical with each other but the sensors and payloads they carry can be differentiated depending on the specific requirement. Its outer structure is designed so that it can be manufactured through 3D printing and in a lightweight approach where, depending on the used material, it can also be manufactured in a deformable manner to gain advantages of soft structures. However, it should be noted that internal structure and corresponding electro-mechanical components are not taken into account in detail since the focus of this thesis is to investigate snake locomotion and learning based algorithms for locomotion patterns. On the other hand, for consistency of the realized operations and to provide infrastructure for future studies which can potentially comprise hardware components, especially the robots which are presented in [13], [15], [16] and [17] are taken into account as a source of inspiration in terms of mass and dimensional features. For anisotropic friction requirement, as stated in Chapter 2, each module of the snake robot is covered with passive wheels. The resulting specifications of the designed snake robot is stated in Table 3.1 below.

Snake Robot Specifications		
Specification	**Value**	**Unit**
Mass of a Module	0.700	kg
Diameter of a Module	0.130	m
Yaw Joint Motion Range Limit	±42.5	deg
Pitch Joint Motion Range Limit	±42.5	deg
Actuation Torque Limit	6.5	Nm
Yaw Joint Speed Limit	65	deg/s
Pitch Joint Speed Limit	65	deg/s
Degrees of Freedom of the Robot	2 x ModuleNumber + 6	N/A

Table 3.1: Specifications of the Designed Snake Robot

3.1.1 Electro-Mechanical Modelling

For electro-mechanical modelling, CAD assembly models [38] are realized in the simulation environment [37] as a simplified model in terms of visual features and unrelated components for simulations of physical interactions to accelerate the simulation process and consequently training process. The single reason for the executed simplifications is related about the solving times which are encountered as the major challenge and limitation during experimentations. However; mass, inertia and dimensional properties are preserved and properly reflected to the physical modelling as stated in Table 3.1 for convenience of the realized experimentations. Additionally, as described in the following subsections, mechanical modellings are realized for actuation of joint mechanisms and representations of contact forces between the passive wheels and the locomotion platform.

3.1.2 Actuation and Joint Mechanisms

Each module of the snake robot includes two joints which correspond to degrees of freedom in yaw and pitch axes. Both axes are in the identical characteristics and, depending on the current posture, the snake robot can operate by interchanging the axes with proper switchings in the motion planning and control architecture. In the instantaneous state, the corresponding joints that provide motion in yaw axis forms 2D snake gait patterns where the joints provide motion in pitch axis extends the motion capabilities to 3D. Joint mechanisms in each module of the snake robot are represented with revolute joints which are modelled based on spring-damper force law in the simulation environment [39] and their actuation is realized via provided torque inputs. The modelling operations of joints require two sets of stiffness and damping parameters where one of them specifies the joint characteristics in the operation range, the other determines characteristics in the limits of motion range. Although various parameter sets are employed throughout the modelling trials, the eventual parameter sets for operational and limit mechanics of the revolute joints are presented in Table 3.2.

Mechanical Modelling Parameters of Revolute Joints		
Operational Mechanics	**Value**	**Unit**
Spring Stiffness	0	Nm/rad
Damping Coefficient	1.1	Nm/(rad/s)
Operation Range Limit Mechanics	**Value**	**Unit**
Spring Stiffness	10^4	Nm/rad
Damping Coefficient	2	Nm/(rad/s)
Transition Region Width	0.0175	rad

Table 3.2: Mechanical Modelling Parameters of Revolute Joints

3.2 Passive Wheeled Structure for Anisotropic Friction

Anisotropic friction, as indicated in Chapter 2, is an indispensable requirement for controllability of a snake robot with stationary initial conditions on a planar surface that points out to the extreme circumstances. As aforementioned, anisotropic friction can be obtained via passive (or, depending on a specific application, through active, semi-active) wheels or mechanical processing of the contact surface. For designed snake robot in the scope of the thesis, passive wheeled structure approach is adopted and correspondingly realized contact force modelling operations are described in the following section.

3.2.1 Contact Force Modelling

Contact force modelling between passive wheels of the snake robot and operation environment is realized based on the approach which is represented in Figure 3.1 where base frame and follower frame correspond respectively the locomotion surface and passive wheels. In this approach, contact frame moves around the specified geometry depending on corresponding motions of contact point as long as the contact between two frames is kept.

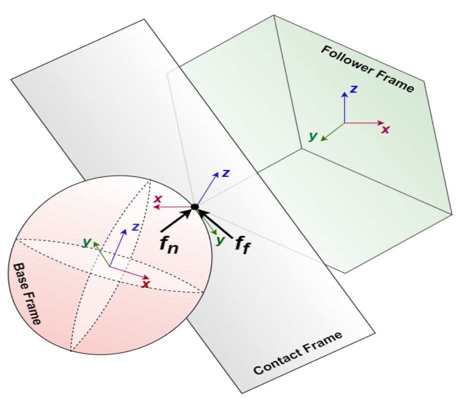

Figure 3.1: Spatial Contact Force Modelling with Base and Follower Geometries

The normal force f_n and the frictional force f_f are two fundamental components which determine characteristics of the contact force model. By employing low and high level tools of *Simscape Multibody* [39] for contact force modelling, f_n and f_f are respectively evaluated as in Equation 3.1 and 3.2.

$$f_n = s(d)(kd + bd') \tag{3.1}$$

where;
f_n: Normal force in the opposite direction with the same magnitude to each contact geometries
d: Penetration depth between geometries which are in contact
d': First time derivative of d
k: Specified stiffness value for the normal force
b: Specified damping value for the normal force
$s(d)$: Implicitly operated smoothing function

In addition to the stated parameters, the transition region width which will be denoted as w is also specified for contact force modelling. In the cases when penetration depth is smaller than the transition region width, $d < w$, normal force f_n is scaled with the implicit smoothing function $s(d)$ where it leads continuous and monotonically increasing characteristic in the interval $[0, w]$. Therefore, it equals to 0 when $d = 0$ and equals to 1 when $d = w$.

Frictional force f_f is correlated with normal force f_n as Equation 3.2 where μ is the friction coefficient which varies depending on the magnitude of relative velocity of contact surfaces.

$$|f_f| = \mu |f_n| \tag{3.2}$$

To determine effective friction coefficient based on the relative velocities of contact points, a smoothed stick-slip behaviour is implicitly employed. Effective μ gradually increases with increasing relative velocity until predetermined critical velocity to upper limit value of the coefficient μ_{static} and it converges to $\mu_{dynamic}$ value by gradually decreasing with increasing relative velocity beyond critical velocity.

Based on the presented contact force modelling approach above, even though various parameter sets are implemented and different tests are realized to verify proper snake locomotion under different contact models which consequently corresponds different surface characteristics, to obtain consistently comparable outcomes, all the trainings which are presented in Chapter 5 are realized with the values stated in Table 3.3 and, followingly, realized experimentations are extended to unexperienced surface conditions by defining new parameter sets and robustness of trained snake robots against variations of locomotion surface characteristics are tested. Before determination of contact force modelling parameter set for trainings, experimented values for stiffness and damping parameters of normal force specification respectively range from 10^2 to 10^6 N/m and from 30 to 10^3 $N/(m/s)$ while 10^{-3}, 10^{-4} and 10^{-5} m values are implemented for transition region width. For static and dynamic friction coefficients of frictional force specification, values between 0.5 to 8.0 are employed and tested while 10^{-3} and 10^{-4} m/s values are implemented for critical velocity parameter. It can also be noted that contact force modelling parameters can be randomized or varied in a planned way in a predetermined training curriculum to reflect characteristics of different contact properties.

Contact Force Modelling Parameters		
Normal Force	**Value**	**Unit**
Stiffness	10^4	N/m
Damping	40	N/(m/s)
Transition Region Width	10^{-4}	m
Frictional Force	**Value**	**Unit**
Method	Smooth Stick-Slip	N/A
Static Friction Coefficient	3.2	N/A
Dynamic Friction Coefficient	3.0	N/A
Critical Velocity	10^{-3}	m/s

Table 3.3: Parameters of the Contact Force Model

Chapter 4

Reinforcement Learning for Snake Robot Locomotion

Motion controller architecture design for the purpose of robotic locomotion which can preserve its robustness under various types of disturbances and environments is a continuing challenge in the present times for different types of robots and unlimited possibilities of external conditions. The designed motion controller architecture provides the movement capability to the robot whereas it also determines the boundaries of this capability. In the content of robotic locomotion, implemented motion control components can be categorized essentially with three fundamental approaches which are deterministic, adaptive and stochastic/learning based control systems. Reinforcement learning based approaches indicate promising potential to realize sophisticated behaviors through their evolutionary characteristics and to reach an adequate robustness level which can overcome distinctive environmental conditions and disturbances. For this reason, reinforcement learning is employed for generation of snake gait patterns in the proposed hybrid motion control system architecture.

4.1 Reinforcement Learning and Deep Reinforcement Learning

Reinforcement learning can be defined as the extraction of mapping which corresponds to cases-to-actions with the purpose of predefined reward function maximization. In accordance with this purpose, an optimal policy is learned through trial and error without explicit specification of the followed methodology. Even though its premise concept dates back a relatively long time, in the recent past, reinforcement learning algorithms have advanced and made significant progress while various algorithmic structures have been proposed and their successes have been proven with applications on distinct problems and tasks from social sciences to engineering challenges. State-of-the-art reinforcement learning algorithms, as described step by step in the following section, mostly originate from the *Q-Learning* [40] basics and certain inspirations from *State-Action-Reward-State-Action (SARSA)* [41] approach. Applicability of reinforcement learning algorithms to different tasks and problems which approximates real-world complexity has gradually extended with sequential improvements and especially with *Deep Q-Network* [42] which incorporates deep neural networks into *Q-Learning* concept as it results in the emergence of deep reinforcement learning term.

4.2 Description and Comparison of State-of-the-Art Reinforcement Learning Algorithms

Majority of state-of-the-art reinforcement learning algorithms originate from *Deep Q-Network* [42] foundations. Even though numerous variations have been proposed around the fundamental algorithms, as essential milestones, *Deep Q-Network (DQN)* [42], *Deep Deterministic Policy Gradient (DDPG)* [43], *Trust Region Policy Optimization (TRPO)* [44], *Proximal Policy Optimization (PPO)* [45] with *Distributed Proximal Policy Optimization (DPPO)* [46] variation, *Soft Actor-Critic (SAC)* [47] [48] and *Twin Delayed Deep Deterministic Policy Gradient (TD3)* [49] can be put in order. In the presented order of neural based algorithms, a gradual development is experienced in terms of sample efficiency, training stability and comprehensiveness of applicability for different problems from discrete domains to continuous domains and from low dimensional observation and action space cases to high dimensional possibilities. Even though *DQN* creates human-level performances in many tasks and even above human-level performance for some tasks [42], it experiences inadequacies especially in continuous action and large observation and action space tasks. With *DDPG* algorithm, significant improvements are realized in continuous control tasks [50] but it still suffers from training instability problems and bears the risk of insufficiency to reach a convergence in relatively large observation and action space cases. Moreover, it is seen that training process generally shows a noisy characteristic and it is quite brittle to hyperparameter settings.

In addition to the neural based reinforcement learning algorithms, as an algorithm which does not include any neural structure, *Augmented Random Search (ARS)* [51] also provides competitive outcomes in terms of sample efficiency and training stability especially comparing with *DDPG*, *TRPO* and *PPO*. It is constructed on the basic random search basis with proposed three main augmentations. They can be pointed out as scaling operation of update steps by the standard deviation, normalization of the states and using top performing directions for the optimization task.

In terms of sample efficiency, training stability and capability in relatively complex tasks which include high dimensional observation and action spaces, *Soft Actor-Critic (SAC)* and *Twin Delayed Deep Deterministic Policy Gradient (TD3)* algorithms can be put forward as state-of-the-art algorithms and experienced outcomes in different tasks, agents and environment settings also verifies the presented benchmark results [47] - [49], [52]. Both *SAC* and *TD3* algorithms are developed on *DDPG* fundamentals with several regulations and additions to eliminate experienced vulnerabilities as described step by step in the following section. Therefore, since *SAC* and *TD3* algorithms ensure an obvious superiority over the other alternatives as presented with corresponding mathematical backgrounds and benchmark outcomes whereas they may outperform over each other depending on the corresponding task and hyperparameter tunings, they both employed to train a 9-modules snake robot and resulting outcomes are compared as presented in Chapter 5. In this content, definitions of the notations in the following equations which describe and legitimize adopted methodologies in *SAC* and *TD3* algorithms to eliminate the aforementioned deficiencies are indicated in Table 4.1.

Definition	Notation
Time step	t
State	s
State space	S
Action	a
Action space	A
Reward	r
Termination conditions	d
Policy	π
Optimal policy	π^*
Marginals of the trajectory distribution	ρ
Temperature parameter	α
Discount factor	γ
Q-function	Q
Q-function parameters	θ
Policy parameters	ϕ
Replay buffer	$\mathcal{D}$
Batch of transitions	B
Desired minimum expected entropy	$\mathcal{H}$
Value function	V
Objective funtion	J
Gradient	∇
Update proportion	τ
Temporal difference residual	δ
Noise	ϵ

Table 4.1: Definition of Terms for Reinforcement Learning Algorithms

4.3 Soft Actor-Critic (SAC) Algorithm

Soft Actor-Critic approach [47] [48] aims to overcome two main challenges, where they are frequently experienced with previously developed algorithms in especially complicated tasks, that can be stated as sample inefficiency, instabilities in training process which is also related with brittleness to hyperparameter settings. In this way, applicability of model-free reinforcement learning framework in relatively complex systems and tasks in both simulation and real-world domains becomes possible.

In *Soft Actor-Critic* approach, the actor aims to maximize expected return and entropy while, in this framework, entropy can be evaluated as exploration. Optimization of the corresponding stochastic policy is performed in an off-policy manner.

In general, previously developed and widely implemented reinforcement learning algorithms until the present time intends to maximize expected sum of rewards which can be formulated as in Equation 4.1 below, where π, r s_t and a_t respectively corresponds to employed policy, reward term, states and actions in time step t as stated in Table 4.1.

$$\pi^* = \arg\max_{\pi} \sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[r(s_t, a_t)] \tag{4.1}$$

In *Soft Actor-Critic* approach, a maximum entropy objective is sought and the standard objective is augmented with an entropy term so that optimal policy intends to maximize its entropy for each corresponding state as indicated in Equation 4.2 where α and $\mathcal{H}$ correspond to *temperature parameter* and *desired minimum expected entropy* respectively.

$$\pi^* = \arg\max_{\pi} \sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[r(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t))] \tag{4.2}$$

where, *temperature parameter* α determines the relative dominance between entropy and reward components. As another interpretation, α directly controls *exploration* versus *exploitation* trade-off where increase in α indicates a more predominant exploration attitude. Therefore, it consequently appoints stochasticity level of the policy.

The stated objective function in Equation 4.2 can be extended for infinite horizon problems by introducing a discount factor, γ , to assure sum of expected rewards and corresponding entropy values are finite. In this content, the objective function is re-composed as;

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \left[\sum_{t}^{\infty} \gamma^t \Big((r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)) \Big) \right] \tag{4.3}$$

For soft policy iteration process, the soft Q-function parameters are trained with the objective of soft Bellman residual minimization as presented in Equation 4.4.

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t)\sim D} \left[\frac{1}{2} \Big(Q_\theta(s_t, a_t) - (r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}\sim p}[V_{\bar{\theta}}(s_{t+1})]) \Big)^2 \right] \tag{4.4}$$

where the implicitly parameterized soft state value function, through soft Q-function parameters, is stated in equation 4.5.

$$V(s_t) = \mathbb{E}_{a_t\sim\pi} \left[Q(s_t, a_t) - \alpha \log \pi(a_t|s_t) \right] \tag{4.5}$$

and the optimization equation form with stochastic gradients is attained as presented below with Equation 4.6.

$$\begin{aligned} \widehat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta(a_t, s_t)(Q_\theta(s_t, a_t) - (r(s_t, a_t) + \\ \gamma(Q_{\tilde{\theta}}(s_{t+1}, a_{t+1}) - \alpha \log(\pi_\phi(a_{t+1}|s_{t+1})))) \end{aligned} \tag{4.6}$$

Consequently, corresponding equation which dictates learning process of policy parameters is obtained as;

$$J_\pi(\phi) = \mathbb{E}_{s_t\sim D} \left[\mathbb{E}_{a_t\sim\pi_\phi}[\alpha \log(\pi_\phi(a_t|s_t)) - Q_\theta(s_t, a_t)] \right] \tag{4.7}$$

Since the policy should more dominantly explore where optimal action is uncertain and should remain more deterministic where distinction between the desired and undesired actions are clear, forcing the entropy to a fixed, static value is not an effective solution. For this reason and to eliminate manual adjustment of optimal *temperature parameter* which is a non-trivial operation, learning objective is re-arranged where entropy is treated as a constraint. As the result, for automation of entropy adjustment, dual variable α_t^* is optimized based on the objective which is presented in Equation 4.8.

$$\alpha_t^* = \arg\min_{\alpha_t} \mathbb{E}_{(a_t \sim \pi_t^*)} \left[-\alpha_t \log \pi_t^*(a_t|s_t; a_t) - \alpha_t \bar{\mathcal{H}} \right] \tag{4.8}$$

However, due to practical considerations for the implementation process, gradient values for α, is computed with the objective function given in Equation 4.9 below.

$$J(\alpha) = \mathbb{E}_{a_t \sim \pi_t} \left[-\alpha \log \pi_t(a_t|s_t) - \alpha \bar{\mathcal{H}} \right] \tag{4.9}$$

The overall *SAC* algorithmic structure, which is assembled from the fundamental components and key equations based on [47] and [48], is presented as following;

Algorithm 1: Soft Actor-Critic (SAC)

Initialize:
  Policy parameters ϕ,
  Q-function parameters θ_1, θ_2
  Replay buffer $\mathcal{D}$
Set target parameters to main parameters:
  $\theta_1^{'} \leftarrow \theta_1$
  $\theta_2^{'} \leftarrow \theta_2$
  while $Convergence \neq true$ **do**
    Observe state $\boldsymbol{s}$ and take action $\boldsymbol{a \sim \pi_\theta(.|s)}$
    Observe following state $\boldsymbol{s'}$, reward $\boldsymbol{r}$ and termination conditions $\boldsymbol{d}$
    Store $\boldsymbol{s, a, r, s', d}$ in replay buffer $\boldsymbol{\mathcal{D}}$
    if $s' \in d$ **then**
      Reset environment
    else
      if $s' \notin d$ & $Time - to - Update = true$ **then**
        for $i \leq UpdateNumber$ **do**
          Randomly sample batch of transitions, $B = \{(s, a, r, s', d)\}$ in $\mathcal{D}$
          Compute targets for the Q-functions:

$$y(r, s', d) = r + \gamma(1-d)\Big(\min_{i=1,2} Q_{\theta_{targ,i}}(s', \tilde{a}')) - \alpha \log \pi_\phi(\tilde{a}'|s')\Big)$$

$$\text{where:} \quad \tilde{a}' \sim \pi_\phi(.|s')$$

          Update Q-functions through gradient descent:

$$\nabla_{\theta_i} \frac{1}{|B|} \sum_{(s,a,r,s',d)\in B} \Big(Q_{\theta_i}(s,a) - y(r,s',d)\Big)^2 \qquad i = 1,2$$

          Update Q-functions through gradient descent:

$$\nabla_\phi \frac{1}{|B|} \sum_{s\in B} \Big(\min_{i=1,2} Q_{\theta_i}(s, \tilde{a}_\theta(s)) - \alpha \log \pi_\phi(\tilde{a}_\phi(s)|s)\Big)$$

$$\text{where:} \quad \tilde{a}_\phi(s) \in \pi_\phi(.|s)$$

          Update target networks:

$$\theta_i^{'} \leftarrow \tau\theta_i + (1-\tau)\theta_i^{'} \qquad i = 1,2$$

        end
      end
    end
  end

4.4 Twin Delayed Deep Deterministic Policy Gradient (TD3) Algorithm

Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm [49] which holds two critic networks and single actor network aims to overcome similar problems with *Soft Actor-Critic (SAC)* algorithm, which can be re-stated as sample inefficiency and instabilities in training process. To eliminate existing vulnerabilities such as brittleness to hyperparameter tuning and drastic overestimation problem of Q-values in *DDPG* algorithm, three fundamental modifications are introduced as they can be put in order as *clipped double Q-learning*, *delayed policy updates* and *target policy smoothing*. Via these modifications, training instability problems are eliminated for especially continuous action tasks which are formed by a high dimensional observation and action spaces, training process reflects considerably less noisy characteristics and sample efficiency is significantly improved.

In *TD3* algorithm, to address overestimation problem, a *Clipped Double Q-Learning* approach is proposed. In this way, it is aimed to eliminate overestimation bias problem whereas underestimation bias can possibly take place. However, in contrary to overestimation bias, underestimation occurrence in actions is preferable since it is not propagated through the policy updates. Consequently, the target update rule with *clipped double Q-learning* algorithm is given in the Equation 4.10 where s' corresponds to the subsequent state value and $Q_{\theta'}$ corresponds to a secondary frozen network.

$$y_1 = r + \gamma \min_{i=1,2} Q_{\theta'_i}\Big(s', \pi_{\phi_1}(s')\Big) \tag{4.10}$$

In addition to overestimation bias, through *TD3* algorithm, high variance estimates problem is also aimed to be eliminated since it causes noisy gradient for policy updates and consequently ends up with reduction in learning performance. For this purpose, several fundamental modifications are proposed for the learning process with *TD3* algorithm to reduce the experienced variance problem.

Since estimation errors can potentially accumulate due to temporal difference update, it can result in overestimation bias and suboptimal policy updates. Because Bellman Equation is not strictly satisfied, some amount of residual temporal difference error occurs after each update as pointed out in Equation 4.11 below.

$$Q_\theta(s,a) = r + \gamma \mathbb{E}[Q_\theta(s', a')] - \delta(s,a) \tag{4.11}$$

where (s', a') and δ respectively corresponds to subsequent state-action pair and temporal difference residual.

Furthermore, the value estimate converges to expected return minus expected discounted sum of future temporal difference errors instead of learning an estimate of the expected return. In this content, error accumulation is tackled as in Equation 4.12.

$$\begin{aligned} Q_\theta(s_t, a_t) &= r_t + \gamma \mathbb{E}[Q_\theta(s_{t+1}, a_{t+1})] - \delta_t \\ &= r_t + \gamma \mathbb{E}[r_{t+1} + \gamma \mathbb{E}[Q_\theta(s_{t+2}, a_{t+2}) - \delta_{t+1}]] - \delta_t \\ &= \mathbb{E}_{s_i \sim \rho_\pi, a_i \sim \pi} \left[\sum_{i=t}^{T} \gamma^{i-t}(r_i - \delta_i) \right] \end{aligned} \tag{4.12}$$

It is also claimed that the use of fast updating target networks causes divergent behaviors. Therefore, it is suggested that the policy network should be updated at a lower frequency comparing with the value network to reach a minimized error before policy update. For this reason, to obtain lower variance in value estimates, updates of the policy (and target networks) are delayed as presented in Equation 4.13 below where θ' indicates to target networks and τ is employed for parameterization of update rate and takes values between 0 and 1.

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \tag{4.13}$$

Finally, a target policy smoothing is implemented, which performs as a regularizer to prevent sharp changes in actions that ultimately causes brittleness of the training process. Therefore, obtaining similar values for similar actions aim is approximated over actions through the added clipped noise to the target policy which is averaged over mini batches. The added random noise is clipped by pre-determined limiting upper and lower borders as defined in Equation 4.14 to keep the target close to the original action.

With aforementioned modifications, modified target update approach takes the following form;

$$
\begin{gathered}
y = r + \gamma \min_{i=1,2} Q_{\theta'_i}\Big(s', \pi_{\phi'}(s') + \epsilon\Big) \\
\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c)
\end{gathered}
\tag{4.14}
$$

The overall *TD3* algorithmic structure, which is assembled from the fundamental components and key equations based on [49], is presented as following;

Algorithm 2: Twin Delayed Deep Deterministic Policy Gradient (TD3)

Initialize:

Policy parameters ϕ; Q-function parameters θ_1, θ_2; Replay buffer $\mathcal{D}$

Set target parameters to main parameters:

$\phi^{'} \leftarrow \phi$; $\theta_1^{'} \leftarrow \theta_1$; $\theta_2^{'} \leftarrow \theta_2$

while *Convergence* $\neq$ *true* **do**

  Observe state $\boldsymbol{s}$ and take action $\boldsymbol{a} = \mathbf{clip}(\boldsymbol{\mu_\phi(s)} + \boldsymbol{\epsilon}, \boldsymbol{a_{low}}, \boldsymbol{a_{high}})$

  Observe following state $\boldsymbol{s'}$, reward $\boldsymbol{r}$ and termination conditions $\boldsymbol{d}$

  Store $\boldsymbol{s}$, $\boldsymbol{a}$, $\boldsymbol{r}$, $\boldsymbol{s\prime}$, $\boldsymbol{d}$ in replay buffer $\boldsymbol{\mathcal{D}}$

  if $s' \in d$ **then**

    Reset environment

  else

    if $s' \notin d$ & $Time - to - Update = true$ **then**

      for $j \leq UpdateNumber$ **do**

        Randomly sample batch of transitions, $B = \{(s, a, r, s', d)\}$ in $\mathcal{D}$

        Compute targets actions:

$$a'(s') = \text{clip}\Big(\mu_{\phi'}(s') + \text{clip}(\epsilon, -c, c), a_{low}, a_{high}\Big); \quad \epsilon \sim \mathcal{N}(0, \sigma)$$

        Compute targets:

$$y(r, s', d) = r + \gamma(1 - d) \min_{i=1,2} Q_{\theta_i'}(s', a'(s'))$$

        Update Q-functions through gradient descent:

$$\nabla_{\theta_i} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \Big(Q_{\theta_i}(s, a) - y(r, s', d)\Big)^2 \qquad i = 1, 2$$

        if $(j \mod PolicyDelay = 0)$ **then**

          Update policy through gradient ascent:

$$\nabla_\phi \frac{1}{|B|} \sum_{s \in B} Q_{\theta_i}(s, \mu_\phi(s))$$

          Update target networks:

$$\theta_i^{'} \leftarrow \tau\theta_i + (1 - \tau)\theta_i^{'} \qquad i = 1, 2$$

$$\phi^{'} \leftarrow \tau\phi + (1 - \tau)\phi^{'}$$

        end

      end

    end

  end

end

The presented algorithmic architectures take observation space terms, summed reward function value and checked termination conditions as inputs while applied commands which form the action space as its outputs. The objective of the algorithm is to maximize the value of the reward function by learning a policy which extracts a mapping from observation set to outcomes of the actions. As a motion control system implementation; observation space and action space terms respectively correspond to control system inputs and outputs where reward function corresponds to the baseline assessment criteria for the optimization problem. In this content, corresponding components are described and presented in the following sections with required details.

4.5 Observation Space and Action Space Determination for Snake Robot Locomotion

Observation space comprises 6 sets of data where three of them include N components and the other three of them include N - 1 components while action space comprises N - 1 term command signals for an N-modules snake robot. Table 4.2 presents definitions of notations which are denoted in the observation space and action space terms.

Definition	Notation
Module number	i
Joint number	j
Time step	t
Commanded direction	x
Perpendicular direction to the command	y
Joint torque	u
Position information	p
Linear velocity information	v

Table 4.2: Definition of Notations in Observation Space and Action Space Terms

Observation space for the N-module snake robot includes;

1. Velocity information of each module in the commanded direction in the current time step, $v_{i,x,t}$: [N x 1]
2. Velocity information of each module in the perpendicular direction to the commanded direction in the current time step, $v_{i,y,t}$: [N x 1]
3. Position information of each module in the perpendicular direction to the commanded direction in the current time step, $p_{i,y,t}$: [N x 1]
4. Position information of the yaw axis joints of each module in the current time step, $p_{j,t}$: [(N-1) x 1]
5. Velocity information of the yaw axis joints of each module in the current time step, $v_{j,t}$: [(N-1) x 1]
6. Applied torque values to the yaw axis joints of each modules from the previous time step, $u_{j,t-1}$: [(N-1) x 1]

where the signals which have certain limit values in the observation set is normalized between [-1, 1] interval to equalize the range of different components for a proper learning process while the others that are not limited with certain values are approximately normalized between [-1, 1] interval based on emerging experimental outcomes by taking into account their upper and lower values and by leaving a sufficient level of margin.

As the output components of the algorithm, action space for the N-modules snake robot corresponds to the applied torque commands to the joints. Corresponding torque command values are generated in [-1, 1] interval and scaled with torque limit value on the verge of applying to the joints.

Therefore, observation space comprises 6N - 3 components while action space includes N - 1 components whose mapping is created by the implemented reinforcement learning algorithm and N components which are determined by the non-stochastic motion control algorithms.

4.6 Reward Function Design

Reward function is composed as the weighted sum of 8 different terms where each of them intends to realize a specific feature for snake locomotion. One by one, they can be described and investigated with their individual purposes as below where corresponding definitions of notations in reward function terms are presented in Table 4.3.

Definition	Notation
Module number	i
Joint number	j
Number of total modules	N
Number of total considered joints	K
Diameter of a module	M_d
Time step	t
Sampling time	T_s
Initial time step	T_i
Initial time step	T_f
Commanded direction	x
Perpendicular direction to the command	y
Joint torque	u
Position information	p
Linear velocity information	v
Acceleration of a joint	$\dot{v}$

Table 4.3: Definition of Notations in Reward Function Terms

The first component is the control effort punishment (or torque penalty), denoted as T_p, which aims to realize commanded motion with minimal torque values. Therefore, applied torque values in the joints are punished based on Equation 4.15 below.

$$T_p = -w_T \sum_{j}^{N} u_{j,t-1}^2 \tag{4.15}$$

The second term is the reward of linear velocity of the robot in the commanded direction where it is denoted as V_r. For a modular snake robot, linear velocity values of the geometric center of each module are taken into account. Although considering the velocity value of only single module such as the leading module or the central module (in the case of snake robot is constituted an odd number of modules) may seem sufficient, it is experienced that training process may tend to converge to an undesired local minima and disruptions which hinder continuous snake locomotion can occur. In this content, the reward term which evaluates linear velocities of the modules is presented in Equation 4.16.

$$V_r = w_V \sum_{i}^{N} v_{i,x} \tag{4.16}$$

The third term compares the orders of consecutive modules. If consecutive modules are not in the expected order in terms of intended direction of movement, a constant penalty with a relatively high weight is applied in each sample time for each module which disrupts the expected sequential order. Corresponding term is denoted as C_p and given in Equation 4.17 as below.

$$\begin{gathered} C_p = -w_C \sum_{i=1}^{N-1} c_{i,x} \\ c_{i,x} = \begin{cases} 1, & p_{i,x} < p_{(i+1),x} \\ 0, & \text{otherwise} \end{cases} \end{gathered} \tag{4.17}$$

The fourth component aims to keep indispensable deviation from the commanded direction due to undulation motion in minimal level by applying proportional punishments with the perpendicular distances from the commanded direction. Defined component is denoted as U_p and presented in Equation 4.18 as below.

$$U_p = -w_U \sum_{i=1}^{N} |p_{i,y}| \tag{4.18}$$

The fifth component checks the summed lateral deviation from the commanded direction by also defining a dead zone to particularly punish excessive deviations while allowing freedom for undulation and proper actuation of curved path commands. It should be noticed that it differentiates from the term which is stated in Equation 4.18 since it does not introduce any punishment due to ordinary undulation motion by considering summed deviation value with a dead zone boundary. In other words, it takes into account the cases when whole snake body deviates from the commanded route instead of undulation pattern on the commanded line. It is denoted as L_p and introduced as in Equation 4.19.

$$\begin{aligned} L_p &= -w_L l_p \\ l_p &= \begin{cases} 0, & \left|\sum_{i=1}^{N} p_{i,y}\right| \leq 2/9 \times N \times M_d \\ \left|\sum_{i=1}^{N} p_{i,y}\right|, & \text{otherwise} \end{cases} \end{aligned} \tag{4.19}$$

The sixth term takes into account the number of zero crossings along commanded direction for each module and the accompanying velocity value. It intends to maximize product of averaged velocity of the modules and total number of zero crossings of each module throughout the commanded direction. It is denoted as Z_r and formulated as in Equation 4.20.

$$\begin{aligned} Z_r &= w_Z \frac{1}{N} \sum_{i=1}^{N} v_{i,x} \left(\sum_{t=T_i+1}^{T_f} \sum_{i=1}^{N} n_{zc}(p_{i,x,t}, p_{i,x,t-1}) \right) \\ n_{zc}(f_1, f_2) &= \begin{cases} 1, & f_1 f_2 < 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{4.20}$$

The seventh component aims to minimize sharp, instant changes in velocity characteristics of the module joints to provide more natural, smooth behaviour to the snake locomotion. It is denoted as A_p and presented in Equation 4.21 as below.

$$A_p = -w_A \sum_{j=1}^{K} \|\dot{v}_j\|^2 \tag{4.21}$$

The eighth and last term is the duration reward which indicates the elapsed time interval until one of the training termination conditions occurred. In other words, it takes into account that how long the robot has been operating. It is denoted as D_r and given in Equation 4.22.

$$D_r = w_D \frac{T_s}{T_f} \tag{4.22}$$

Consequently, the overall reward function is composed as Equation 4.23 with the proposed eight terms.

$$\begin{aligned} R = &-w_T \sum_{j}^{N} u_{j,t-1}^2 + w_V \sum_{i}^{N} v_{i,x} - w_C \sum_{i=1}^{N-1} c_{i,x} - w_U \sum_{i=1}^{N} |p_{i,y}| - w_L l_p \\ &+ w_Z \frac{1}{N} \sum_{i=1}^{N} v_{i,x} \left(\sum_{t=T_i+1}^{T_f} \sum_{i=1}^{N} n_{zc}(p_{i,x,t}, p_{i,x,t-1}) \right) - w_A \sum_{j=1}^{K} \|\dot{v}_j\|^2 + w_D \frac{T_s}{T_f} \end{aligned} \tag{4.23}$$

In the process of premise trainings, weights of the defined eight terms are tuned separately based on the experienced outcomes. The finalized values for weights of the each term are stated in Table 4.4.

Weight Term	Value
Torque Penalty Weight: w_T	0.01
Velocity Reward Weight: w_V	5
Inconsecutive Order Punishment Weight: w_C	50
Undulation Penalty Weight: w_U	1.5
Lateral Deviation Punishment Weight: w_L	0.5
Zero Crossing Reward Weight: w_Z	1
Sharp Velocity Change Punishment Weight: w_A	10^{-4}
Duration Reward Weight: w_D	5

Table 4.4: Determined Weights for Components of Reward Function

Therefore, with weighted terms, the overall reward function which is stated in Equation 4.23 takes finalized form as in Equation 4.24 to employ in the snake locomotion trainings.

$$
\begin{aligned}
R = &-0.01\sum_{j}^{N} u_{j,t-1}^2 + 5\sum_{i}^{N} v_{i,x} - 50\sum_{i=1}^{N-1} c_{i,x} - 1.5\sum_{i=1}^{N} |p_{i,y}| - 0.5 l_p \\
&+ \frac{1}{N}\sum_{i=1}^{N} v_{i,x}\left(\sum_{t=T_i+1}^{T_f}\sum_{i=1}^{N} n_{zc}(p_{i,x,t}, p_{i,x,t-1})\right) - 10^{-4}\sum_{j=1}^{K}\|\dot{v}_j\|^2 + 5\frac{T_s}{T_f}
\end{aligned}
\tag{4.24}
$$

where;

$$
c_{i,x} = \begin{cases} 1, & p_{i,x} < p_{(i+1),x} \\ 0, & \text{otherwise} \end{cases}
$$

$$
l_p = \begin{cases} 0, & \left|\sum_{i=1}^{N} p_{i,y}\right| \leq 2/9 \times N \times M_d \\ \left|\sum_{i=1}^{N} p_{i,y}\right|, & \text{otherwise} \end{cases}
$$

$$
n_{zc}(t_1, t_2) = \begin{cases} 1, & t_1 t_2 < 0 \\ 0, & \text{otherwise} \end{cases}
$$

4.7 Determination of Termination Conditions

Training process includes four conditions which terminate the current episode and leads to continuation with the next episode. Three of the four conditions form the early stopping criteria while one of them is the termination of the episode reaching after the maximum number of steps. The current episode is terminated before predetermined maximum number of steps if;

1. Central module of the robot moves perpendicularly more than 1.5 meters from the commanded direction.

2. Minimum number of modules which are in contact with the ground is less than 4/9 of total number of modules.

3. Absolute velocity value of the central module in the commanded direction is less than 0.15 m/s for 2.5 seconds.

Chapter 5

Motion Control System Architecture and Training Results

Motion control system architecture for designed modular and reconfigurable snake robot is presented at this stage with realized experimentations that cover different robotic architectures, fault scenarios, 2D locomotion characteristics and 3D operation concepts. The realized scenarios with implementation details are illustrated in the corresponding sections and obtained outcomes with both positive and negative aspects are evaluated and discussed for the related cases. Realized experimentations are based on the mathematical analysis backgrounds which are described in Chapter 2, robotic architecture design and modelling specifications which are stated in Chapter 3 and algorithmic design details in Chapter 4. Consequently, the proposed hybrid motion control system architecture for snake robot locomotion is portrayed.

5.1 Snake Robot Locomotion via Reinforcement Learning

The designed and modelled snake robot, as corresponding details are presented in Chapter 3, is trained with *Soft Actor-Critic (SAC)* and *Twin Delayed Deep Deterministic Policy Gradient (TD3)* reinforcement learning algorithms due to the reasonings which are offered in Chapter 4. The employed reward function in the training process with corresponding design details and justifications, observation space and action space of the trained snake robots are also described in Chapter 4. Based on the illustrated perspective, as one by one presented in the following sections, a 9-modules snake robot is trained with *SAC* and *TD3* algorithms at the initial phase with same reward function, contact force model and electro-mechanical model characteristics to investigate comparative performances of algorithms. Afterwards, trainings are realized with determined algorithm for 5, 7, 9 modules snake robots and 11 modules snake robot which includes 3 distinct defective modules where further regulations and tunings are performed on reward function terms and hyperparameters to improve the performance and efficiency on the verge of stated experimentations. Afterwards, realized experimentations are extended to locomotion on unexperienced surface conditions and three dimensional operation concepts.

Evaluation criteria for the realized motions during training episodes are implicitly evaluated based on termination conditions and resulting reward value as described in Chapter 4. Therefore, if one of the termination conditions takes place where they take into account lateral deviation of central module, minimum number of modules which are in contact with the ground and minimum absolute velocity value of the central module in the commanded direction for a predetermined time interval, the adopted attitude by the snake robot is evaluated as unsuccessful in corresponding episode and the episode is terminated. On the reverse case, the current episode is evaluated as successful and its grade of success is associated with the resulting reward value as higher reward values indicate to improved performance based on the predetermined criteria. In other words, for a snake locomotion which can be accepted successful, a continuous movement capability on the commanded direction by keeping the (minimal) contact points with the ground and staying inside the maximum deviation borders is defined as the indispensable requirement.

During the realized trainings, initial states of yaw joints of each module are independently randomized in the intervals [-15, 15] degrees for angular positions and [-60, 60] degrees per second for angular velocities of the joints with 0.5 probability and zero initial states are set for velocity and position values with 0.5 probability at the beginning of each episode. The path following should be achieved both with zero initial states of the joints and the randomized initial states which corresponds to random deviation of the robot posture from the commanded line with non-zero initial joint angular velocities. In other words, the snake robot should follow the commanded path even if its heading points out another direction and its posture and joint velocities are in a random state for recently commanded direction by appropriately arranging its posture, position and velocity states to aimed locomotion line. In addition to randomization operation; for exploration purpose, *Ornstein-Uhlenbeck (OU)* noise model [53] is employed for all the trainings. Also, a relatively aggressive learning process is intentionally followed and learning rates are set relatively higher than common values to overcome repetitive behaviours which are encountered during various trials along high number of episodes. As a consequence of these factors, relatively noisy characteristics are observed in the resulting reward functions as they can be investigated in the following sections.

As it is aforementioned, snake robot locomotion gait patterns are determined by mainly two factors which can be restated as the electro-mechanical architecture of the robot and capabilities of the implemented motion control and planning system. For scope of the investigations of this thesis, essential determinant factor is the architecture of the robot. Therefore, commanded locomotion operations are aimed to be realized with undulation gait pattern both in straight and curved path commands. In addition to the underlined points, purpose specific motion control objectives and the scope of realized investigations are described under the corresponding subsections.

5.1.1 Comparative Training Results of SAC and TD3 with 9 Modules Snake Robot Locomotion

To compare the performances of *SAC* and *TD3* algorithms, trainings are realized with a 9 modules snake robot with separately tuned parameters for each algorithm and under equal conditions. At this comparison stage, randomization of initial states is not applied and zero initial conditions are set for yaw joint position and velocity values. As the motion control objective, a straight path following is aimed by maximizing the forward velocity and keeping the lateral deviations limited with the inevitable undulation patterns. Consequently, since there are clear differences between obtained outcomes which indicate *TD3* superiority for this specific case as can be seen in Figure 5.1, the following operations which include further hyperparameter tunings, arrangements in reward function terms and experimentations with different snake robots whose number of modules varies and may include defective components are realized with *TD3* algorithm.

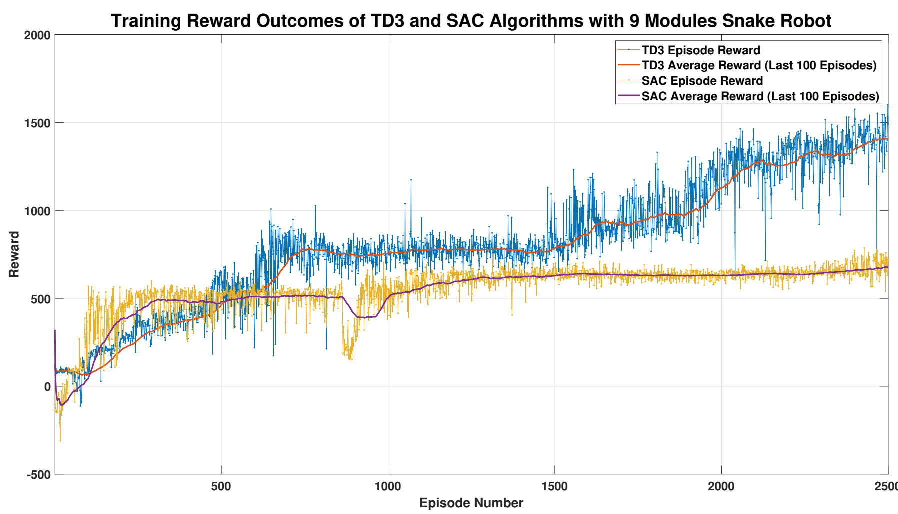

Figure 5.1: Comparative reward outcomes of SAC and TD3 algorithms for 9 modules snake robot training

5.1.2 Training Results for 5 Modules Snake Robot

At this point, 5 modules snake robot is trained with *TD3* algorithm with finalized parameter settings after its comparison with *SAC* algorithm and subsequent various trials by evaluation of emerging outcomes. For motion control objective, a straight path following purpose is pursued independent from the initial conditions. In other words, the snake robot should follow the commanded path with both zero initial states and independently randomized initial states in the intervals [-15, 15] and [-60, 60] respectively for angular position and velocities. Actually, the case of randomized initial states corresponds to commanding the robot to another target while the snake robot is already realizing a previously applied different command which contradicts and is not compatible with newly commanded straight path. In this content, lateral deviations from the commanded line are aimed to be prevented except inevitably required lateral undulations and the forward velocity in the commanded direction is intended to be maximized. Consequently, reward values for 5 modules snake robot training with *TD3* algorithm through 5000 episodes are presented in Figure 5.2.

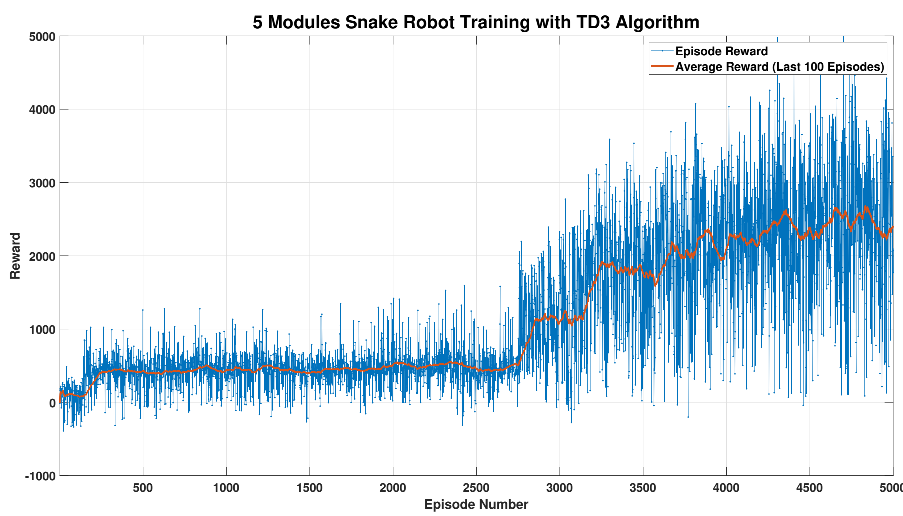

Figure 5.2: Reward values for training of 5 modules snake robot

For 5 modules snake robot locomotion patterns, position and velocity values of each module are respectively presented in Figure 5.3 and Figure 5.4 with zero initial states and in Figure 5.5 and Figure 5.6 with random initial states.

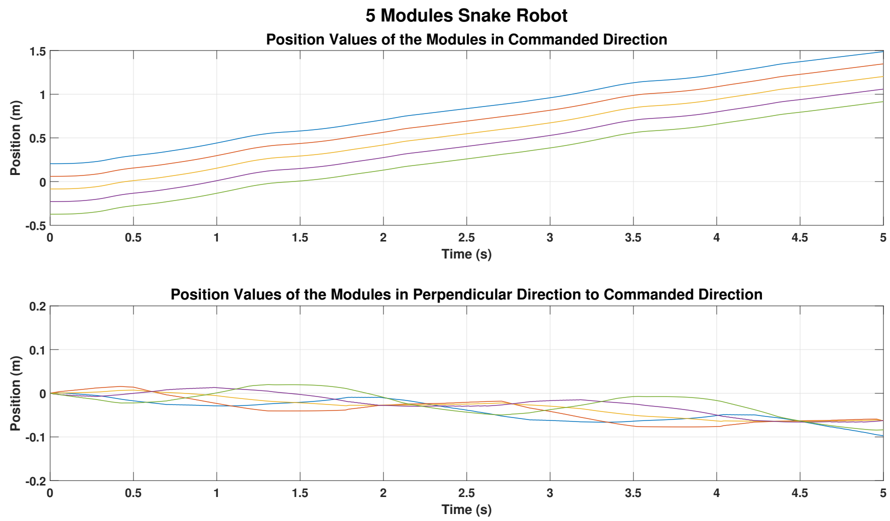

Figure 5.3: Module positions of 5 modules snake robot.

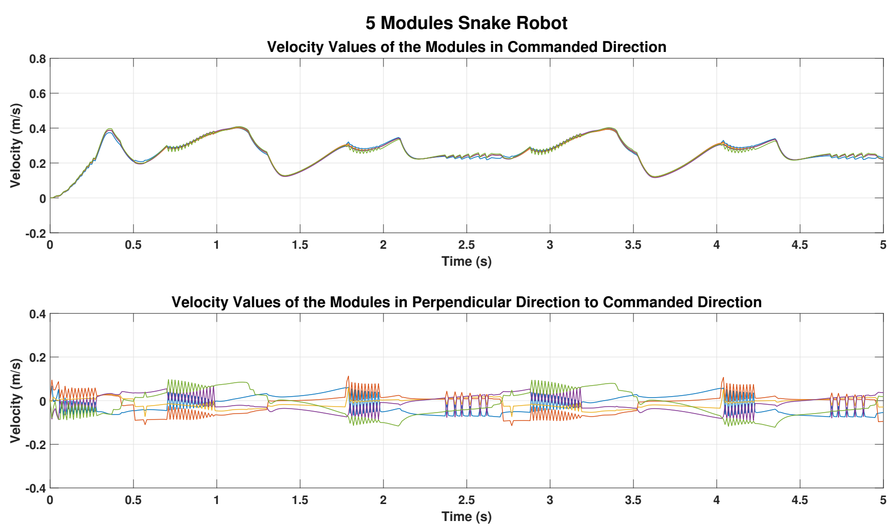

Figure 5.4: Module velocities of 5 modules snake robot.

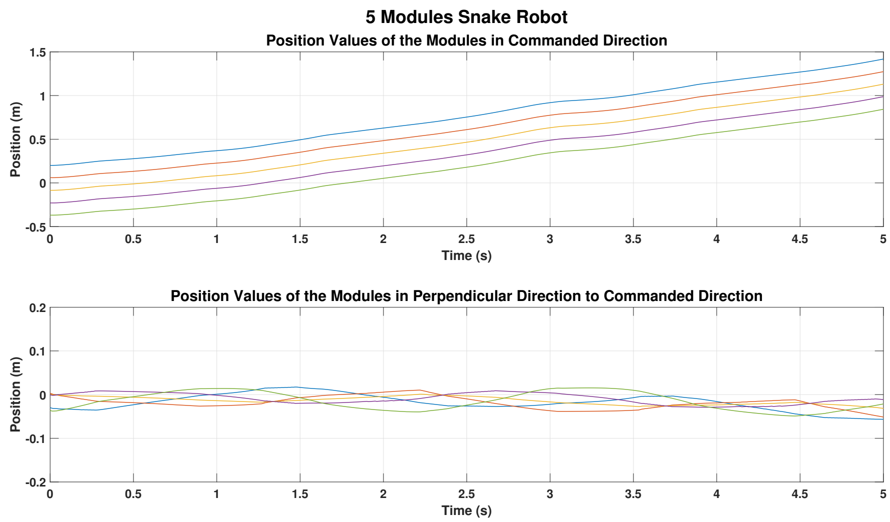

Figure 5.5: Module positions of 5 modules snake robot when initial states are randomized.

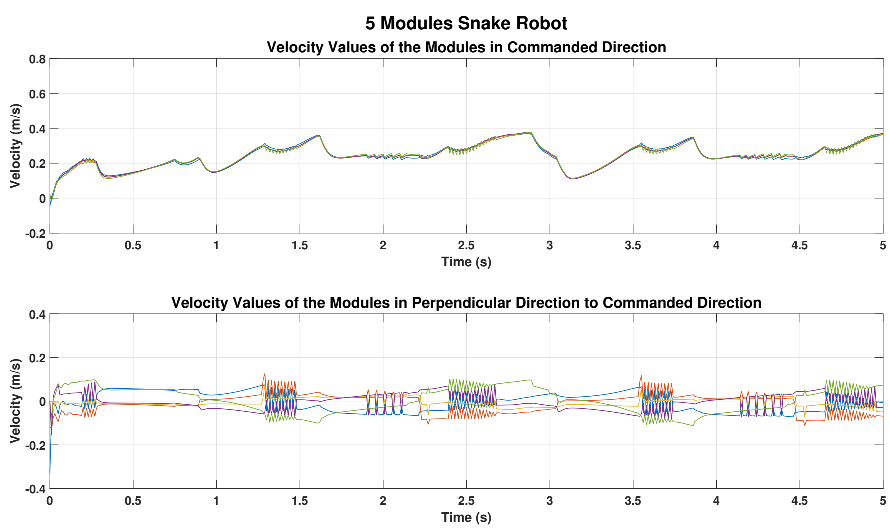

Figure 5.6: Module velocities of 5 modules snake robot when initial states are randomized.

5.1.3 Training Results for 7 Modules Snake Robot

The 7 modules snake robot is also trained with *TD3* algorithm with same hyperparameter settings for straight path following objective with both zero initial states and randomized initial states of yaw joints of the modules. Therefore, in the case of randomized initial states, the robot should appropriately re-arrange its direction, posture and module velocities and execute convenient sequences to realize the latest command. Reward outcomes of training with 5000 episodes and randomized initial conditions of yaw axis joints is presented in Figure 5.7. However, it should be note that reward values are not directly comparable or interpretable between the snake robots whose module numbers are different due to stochastic nature and implicit changes in reward function terms depending on the number of modules.

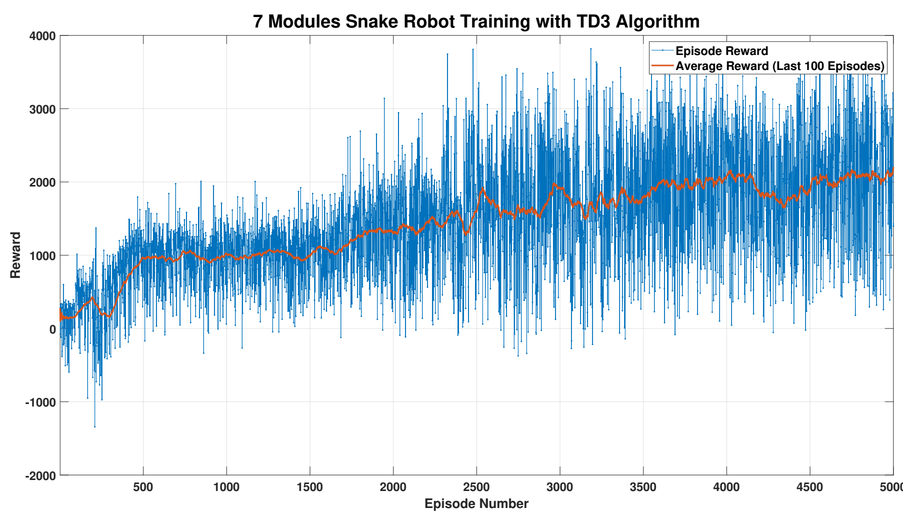

Figure 5.7: Reward values for training of 7 modules snake robot

In this content, position and velocity values of each module in the commanded direction are respectively presented in Figure 5.8 and Figure 5.9 for the specific case of zero initial states for module joints where resulting outcomes for randomized initial states are given in Figure 5.10 and Figure 5.11.

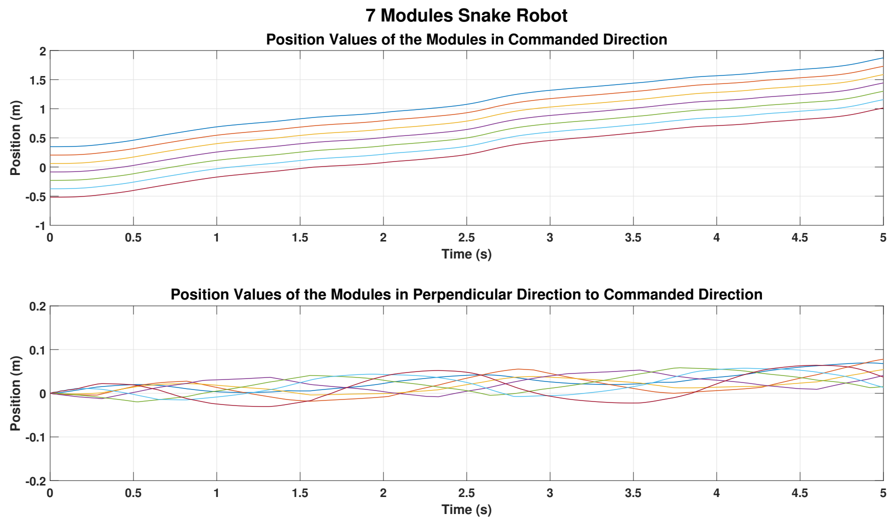

Figure 5.8: Module positions of 7 modules snake robot.

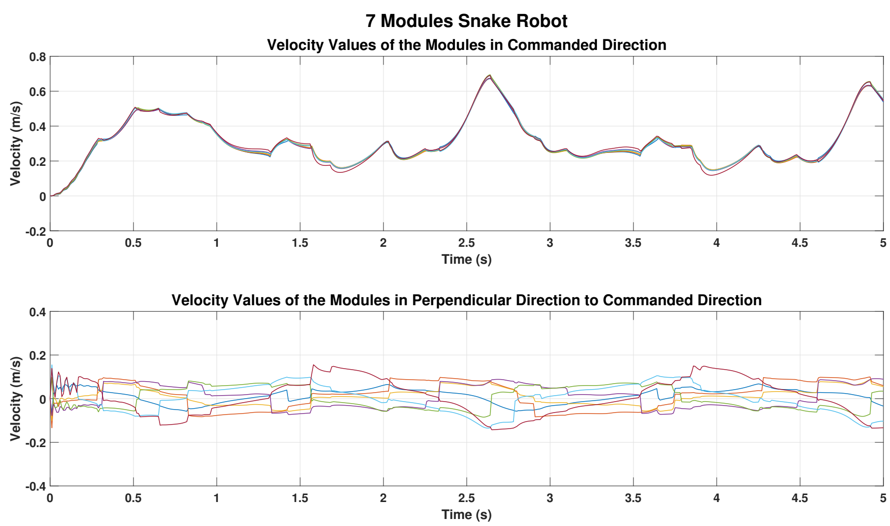

Figure 5.9: Module velocities of 7 modules snake robot.

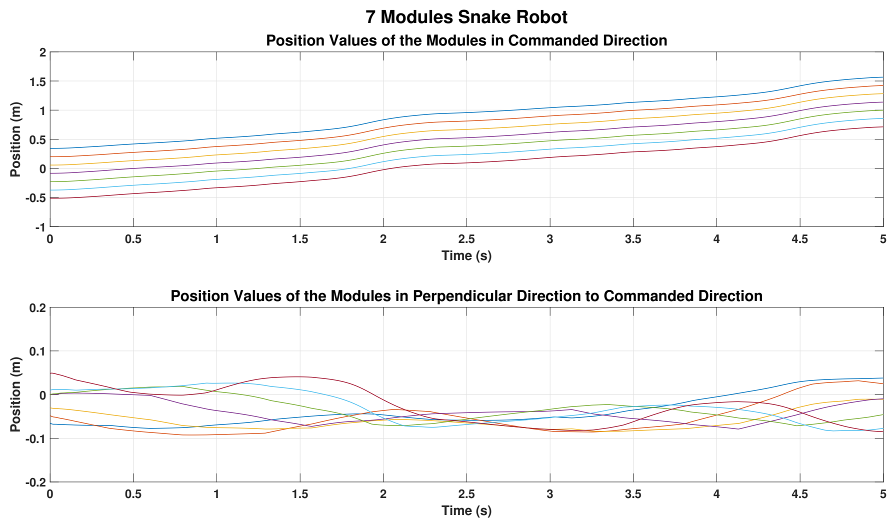

Figure 5.10: Module positions of 7 modules snake robot when initial states are randomized.

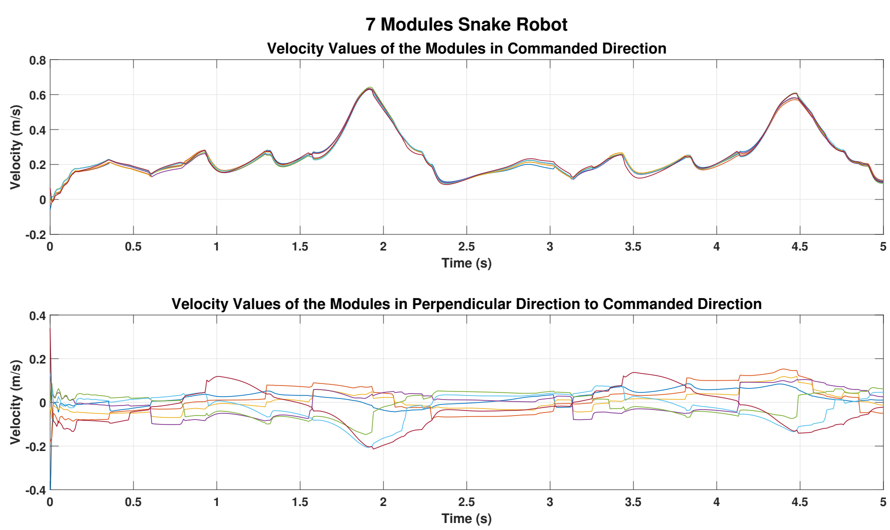

Figure 5.11: Module velocities of 7 modules snake robot when initial states are randomized.

5.1.4 Training Results for 9 Modules Snake Robot

The 9 modules snake robot is also trained via previously described approach along 5000 episodes and corresponding reward values are presented in Figure 5.12. As it can be seen from the resulting outcomes of 5, 7 and 9 modules snake robot trainings, there are minor imperfections which remain with 5000 episodes training like chatterings in lateral velocities of 5 module snake robot or minor deviations from the commanded line of 9 modules snake robot, which can be simply eliminated with further trainings. However, in contrary to instinctual anticipation, a smooth sinusoidal locomotion pattern may or may not be optimal attitude for snake locomotion depending on the surface friction characteristics. Accordingly, minor or major divergences from smooth sinusoidal pattern should not be interpreted as a deficiency or imperfection in snake locomotion which emerges through trainings. It should be noticed that although the realized snake motions after limited number of training episodes are not optimal, optimal snake locomotion is dependent on the contact surface characteristics and the optimal wave patterns significantly differentiate with various friction characteristics [32].

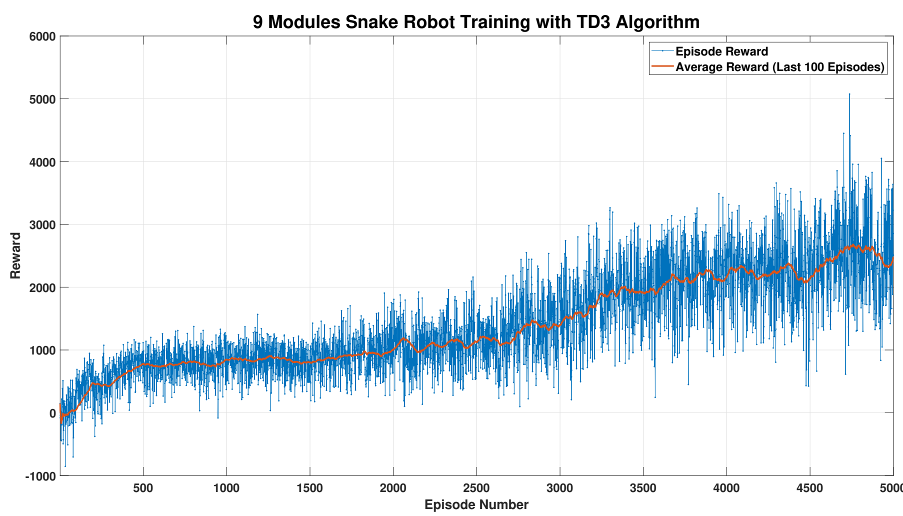

Figure 5.12: Reward values for training of 9 modules snake robot

Resulting position and velocity characteristics of nine modules snake robot is presented in Figure 5.13 and Figure 5.14 for the case of starting motion with zero initial states and in Figure 5.15 and Figure 5.16 for a random initial states case.

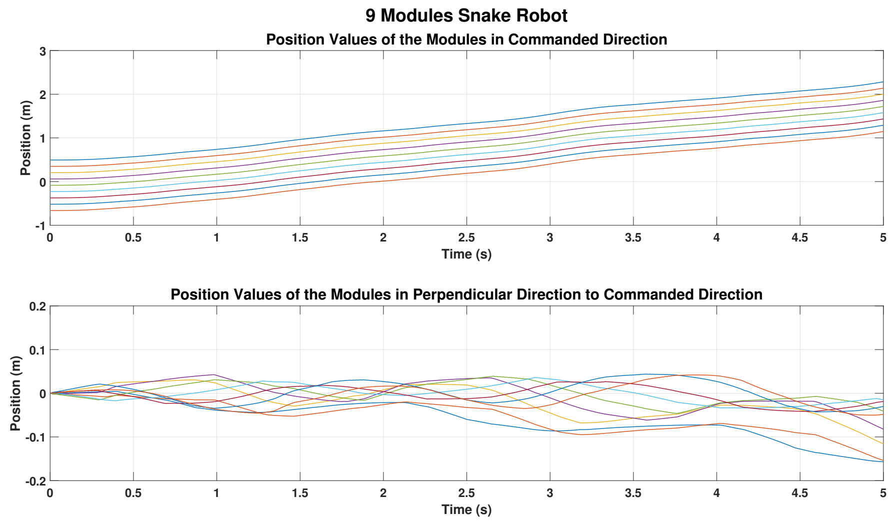

Figure 5.13: Module positions of 9 modules snake robot.

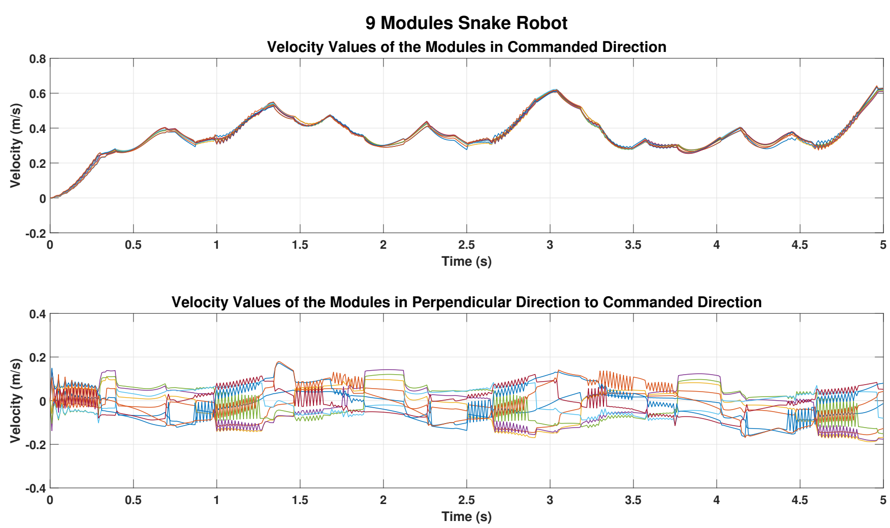

Figure 5.14: Module velocities of 9 modules snake robot.

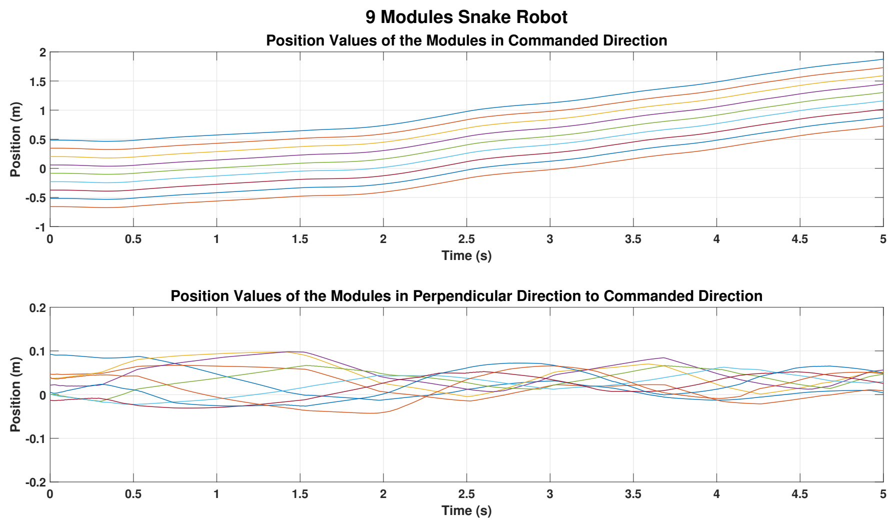

Figure 5.15: Module positions of 9 modules snake robot when initial states are randomized.

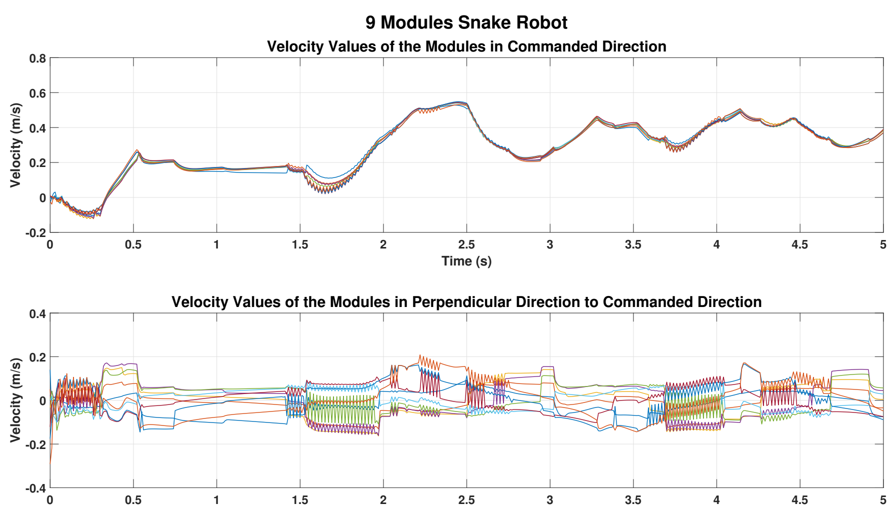

Figure 5.16: Module velocities of 9 modules snake robot when initial states are randomized.

5.1.5 Fault Tolerant Locomotion Training for a Defective Snake Robot

In this scenario, a defective 11 modules snake robot is created where corresponding defects exist in the third, fifth and seventh modules. In the third module, the yaw axis joint gets stuck and cannot actuate properly despite appropriately produced torques. It can move only in the interval $[-0.05, 0.05]$ degrees around zero degree position. In the fifth module, there is not a case of being stuck and the joint can move freely in the defined interval. However, torque commands cannot be generated and the joint cannot be commanded appropriately. Consequently, it is not actuated and it is directly open to be affected from external disturbances. Lastly, the seventh module is also dysfunctional so that it cannot produce torque commands and gets stuck in the interval 6.8 ± 0.02 degrees by breaking the symmetric posture of the snake robot. Since it would create a contradictory case, randomization operation is not employed for states of third and seventh module joints to realize stuck scenario properly. However, for the fifth module, initial states are randomized at the beginning of each episode since only torque is not generated properly but it can freely move in the defined interval. Corresponding outcomes after the training can be seen in Figures 5.17 - 5.19.

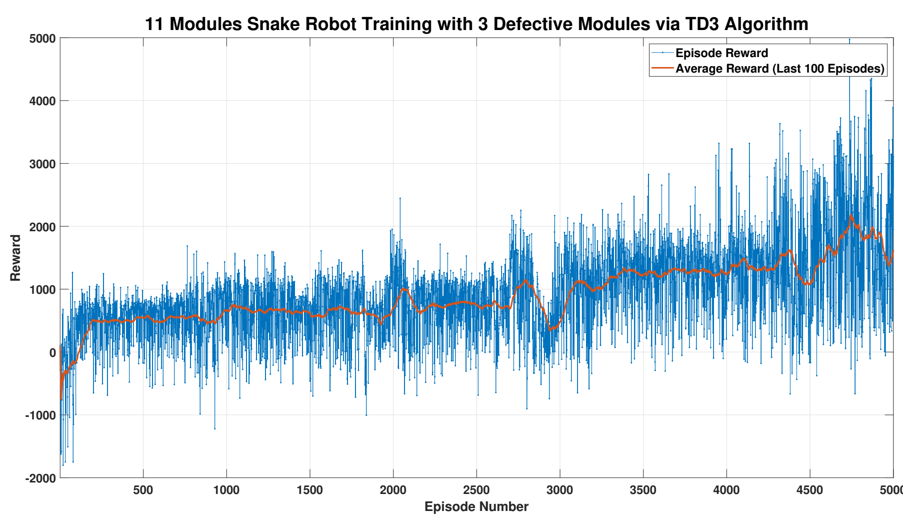

Figure 5.17: Reward values for training of 11 modules snake robot with 3 defective modules

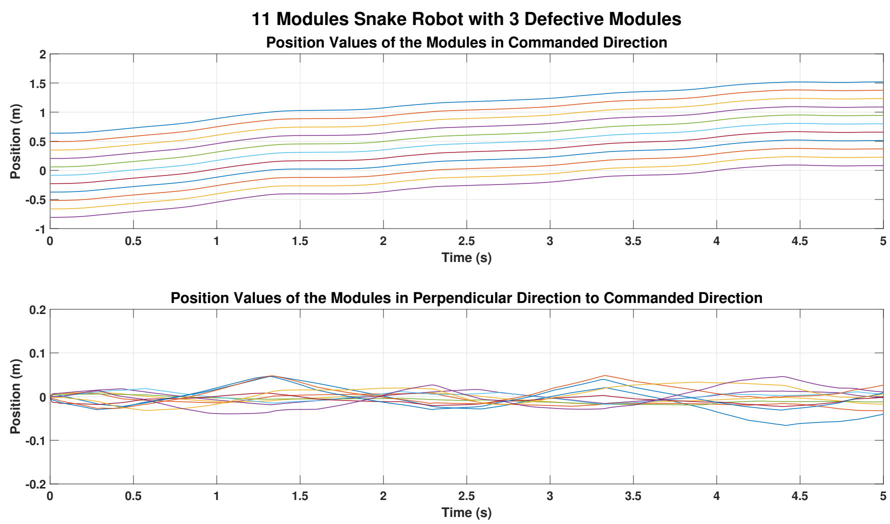

Figure 5.18: Module positions of 11 modules snake robot with 3 defective modules.

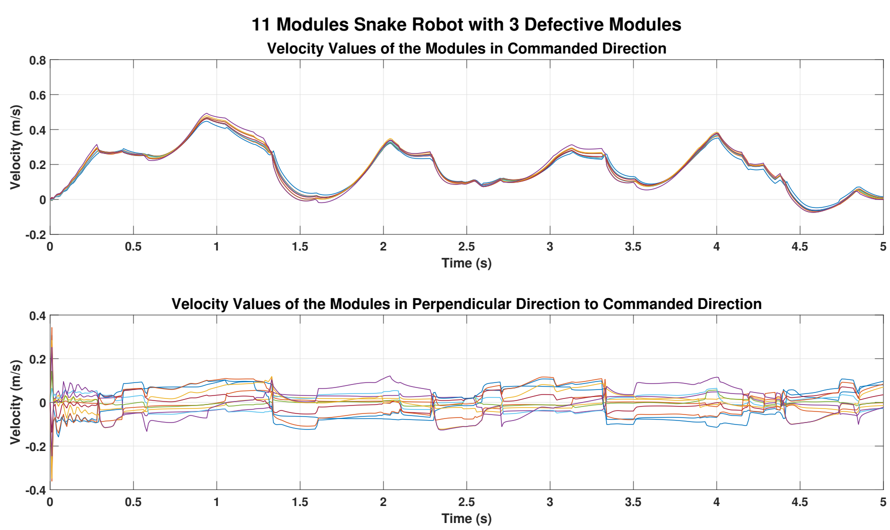

Figure 5.19: Module velocities of 11 modules snake robot with 3 defective modules.

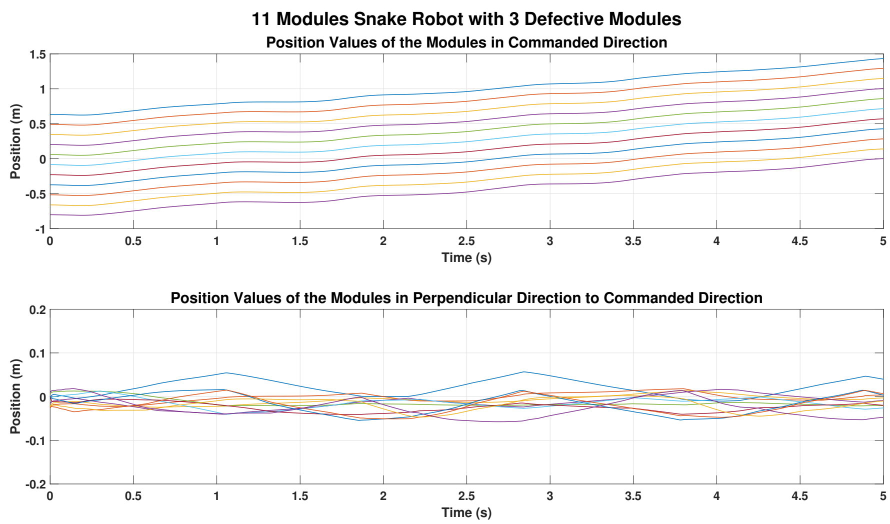

Figure 5.20: Module positions of 11 modules snake robot with 3 defective modules when initial states are randomized.

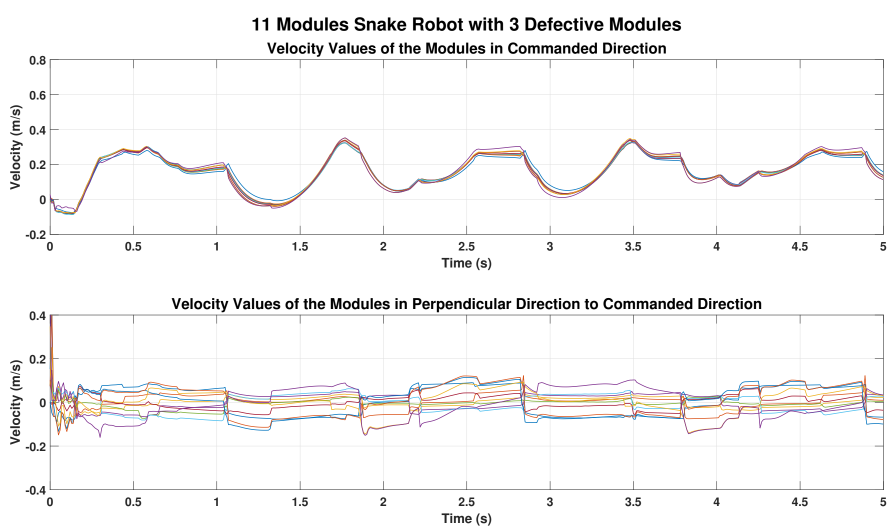

Figure 5.21: Module velocities of 11 modules snake robot with 3 defective modules when initial states are randomized.

5.1.6 Snake Robot Locomotion on Unexperienced Surface Conditions

The presented trainings and corresponding experimentations until this point, as it is underlined, are realized with modelling characteristics described in Chapter 3 and parametric values which describe contact properties as provided in Table 3.3. In this stage, the snake robot behaviours are investigated under unexperienced, unseen surface conditions. Therefore, generalizability of the resulting policy which is obtained after trainings with unchanged contact surface properties is examined. The snake locomotion objective is defined as following a straight path with possible maximum velocity and minimum lateral deviation. In this content, experimentations are realized with both zero initial states and randomized initial velocity and position states of the yaw joints.

The experimentation for unexperienced surface conditions which correspond to locomotion in different terrain properties from the snake robot has been trained is realized with 7 modules snake robot with three different contact force modelling parameter sets as presented in Table 5.1.

First set of contact force modelling parameters corresponds to a high frictional surface with a significantly increased scale of static and dynamic friction coefficients. Also, stiffness is decreased whereas damping value is increased. The resulting locomotion characteristics with zero and randomized initial states are respectively presented in Figures 5.22 - 5.25.

Second set of parameters reflect a slippery surface characteristics with considerably decreased static and dynamic friction coefficients whereas stiffness is increased and damping value is kept the same with the training set. Corresponding snake locomotion behaviors are illustrated in Figures 5.26 - 5.29.

Additionally, a third set of contact parameters is determined as another intermediate set where static and dynamic friction coefficients are specified between first and third set where stiffness value is significantly decreased and damping value is increased. The resulting outcomes are presented in Figures 5.30 - 5.33.

Contact Force Modelling Parameter Sets		
Training Parameter Set		
Normal Force	**Value**	**Unit**
Stiffness	10^4	N/m
Damping	40	N/(m/s)
Frictional Force	**Value**	**Unit**
Static Friction Coefficient	3.2	N/A
Dynamic Friction Coefficient	3.0	N/A
Unexperienced Parameter Set - 1		
Normal Force	**Value**	**Unit**
Stiffness	3×10^4	N/m
Damping	57	N/(m/s)
Frictional Force	**Value**	**Unit**
Static Friction Coefficient	5.8	N/A
Dynamic Friction Coefficient	5.5	N/A
Unexperienced Parameter Set - 2		
Normal Force	**Value**	**Unit**
Stiffness	4×10^4	N/m
Damping	40	N/(m/s)
Frictional Force	**Value**	**Unit**
Static Friction Coefficient	0.9	N/A
Dynamic Friction Coefficient	0.8	N/A
Unexperienced Parameter Set - 3		
Normal Force	**Value**	**Unit**
Stiffness	2×10^3	N/m
Damping	64	N/(m/s)
Frictional Force	**Value**	**Unit**
Static Friction Coefficient	1.6	N/A
Dynamic Friction Coefficient	1.45	N/A

Table 5.1: Contact Force Model Parameter Sets for Realization of Unexperienced Surfaces

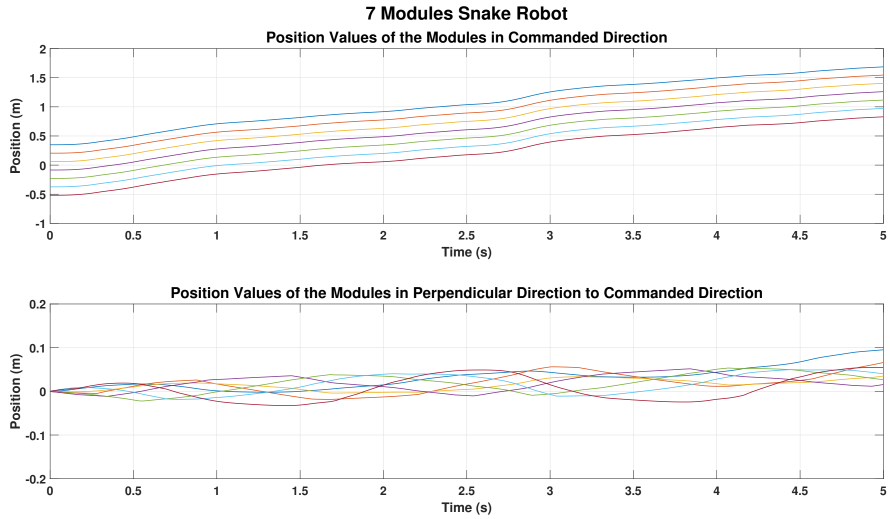

Figure 5.22: Module positions of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set one.

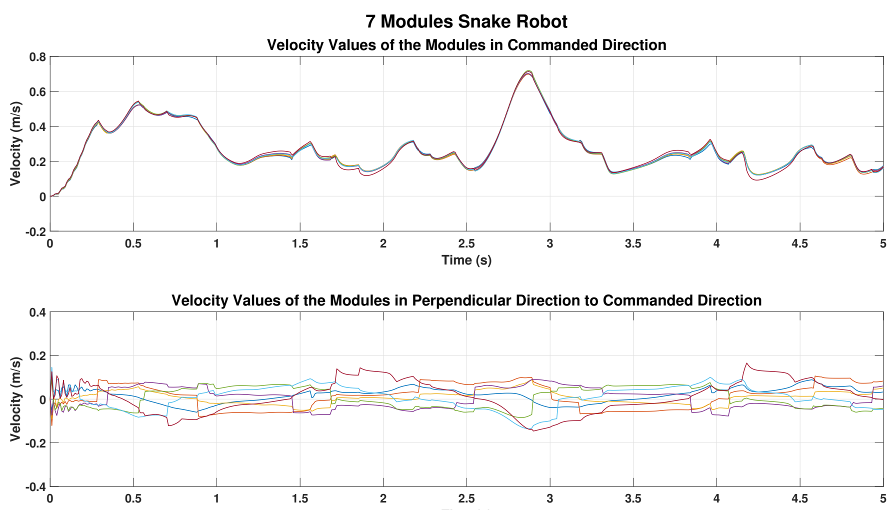

Figure 5.23: Module velocities of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set one.

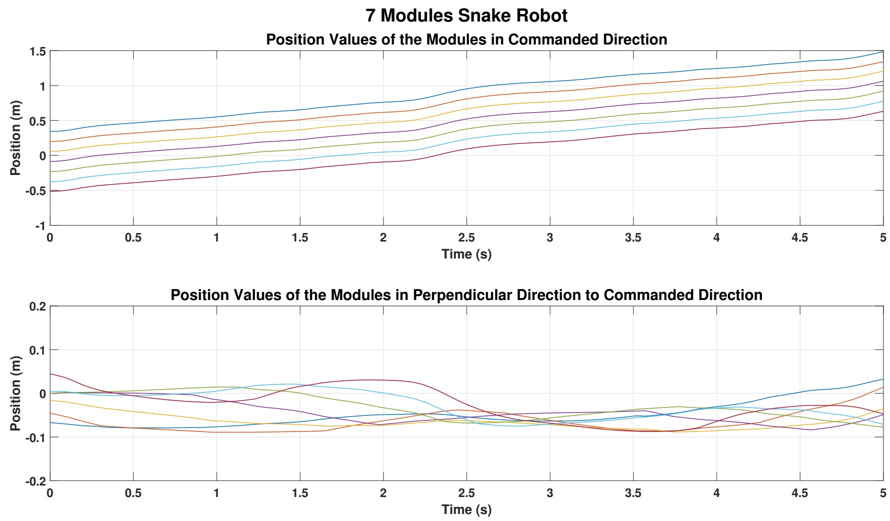

Figure 5.24: Module positions of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set one when initial states are randomized.

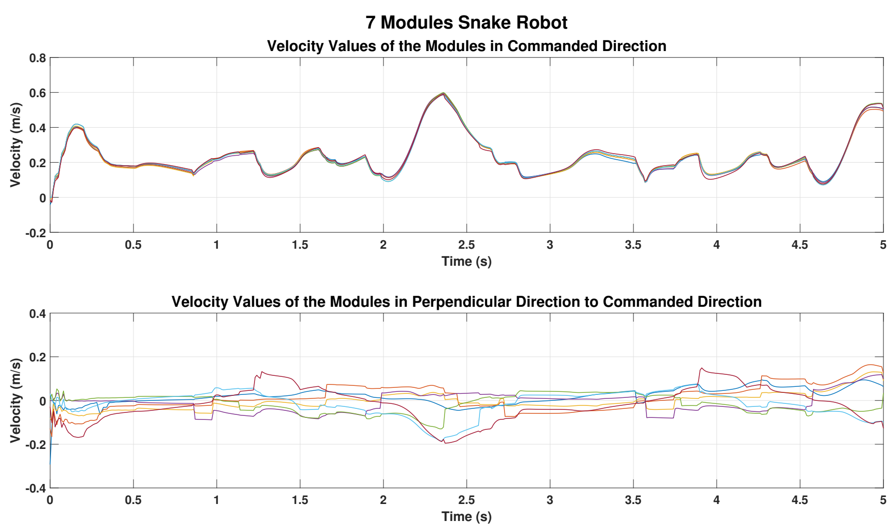

Figure 5.25: Module velocities of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set one when initial states are randomized.

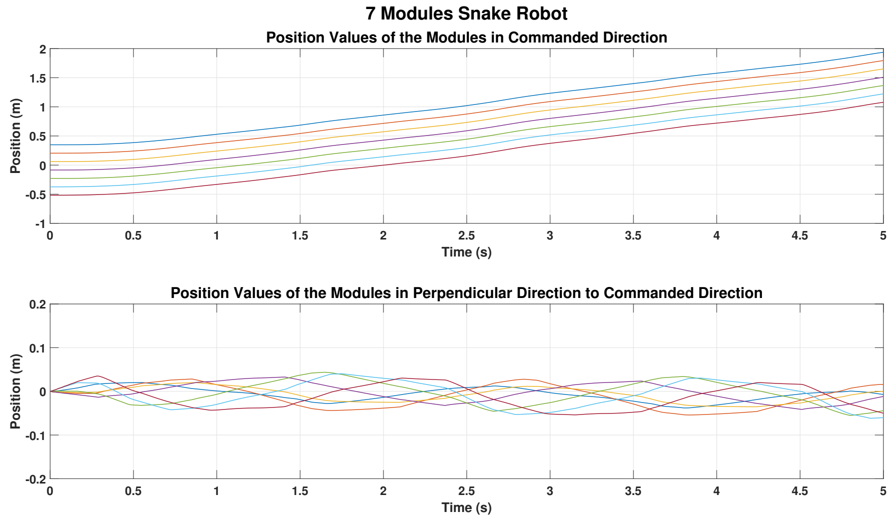

Figure 5.26: Module positions of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set two.

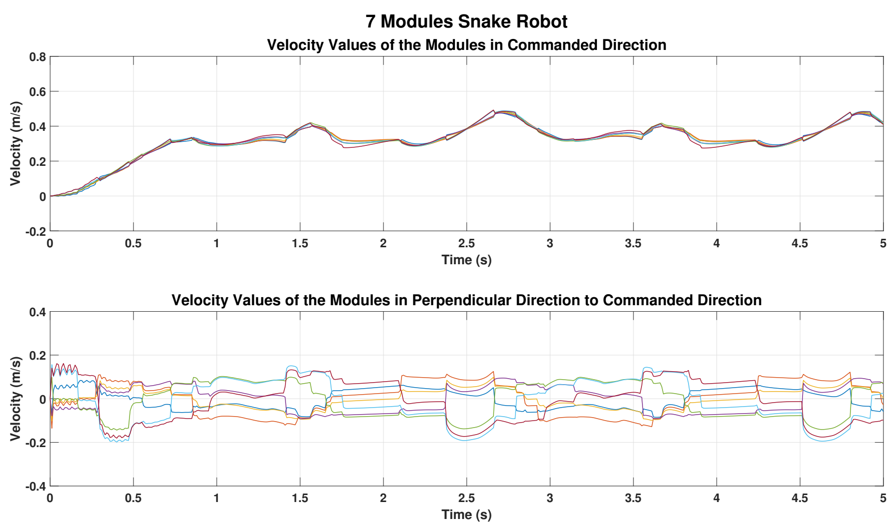

Figure 5.27: Module velocities of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set two.

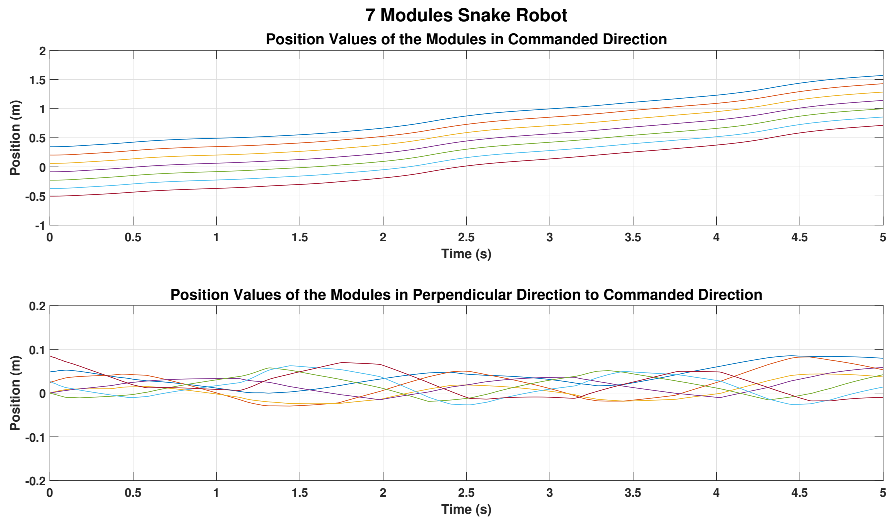

Figure 5.28: Module positions of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set two when initial states are randomized.

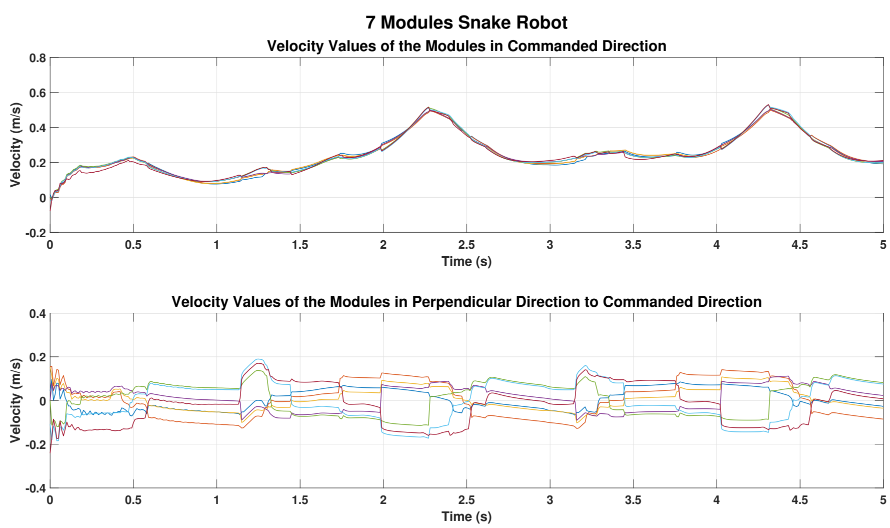

Figure 5.29: Module velocities of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set two when initial states are randomized.

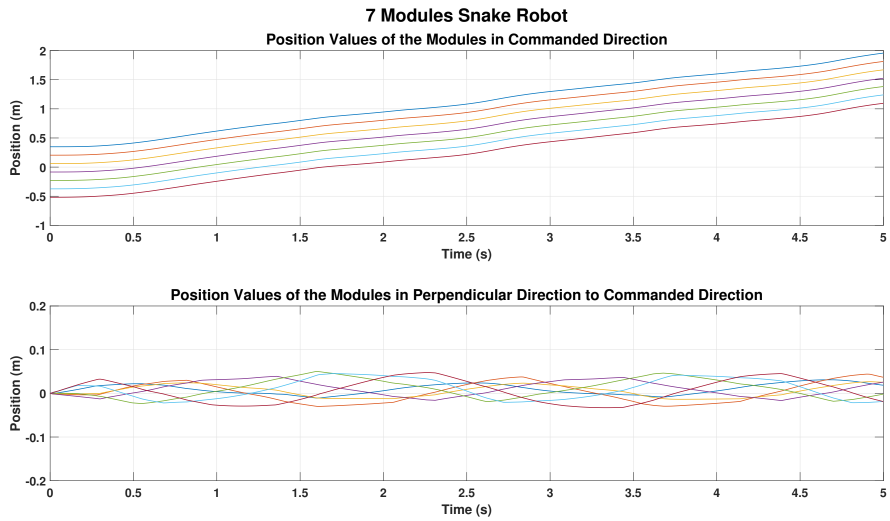

Figure 5.30: Module positions of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set three.

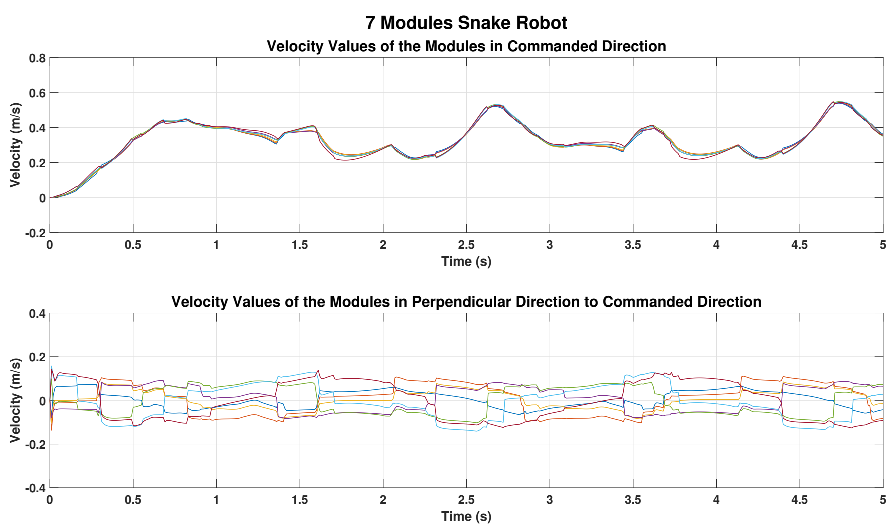

Figure 5.31: Module velocities of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set three.

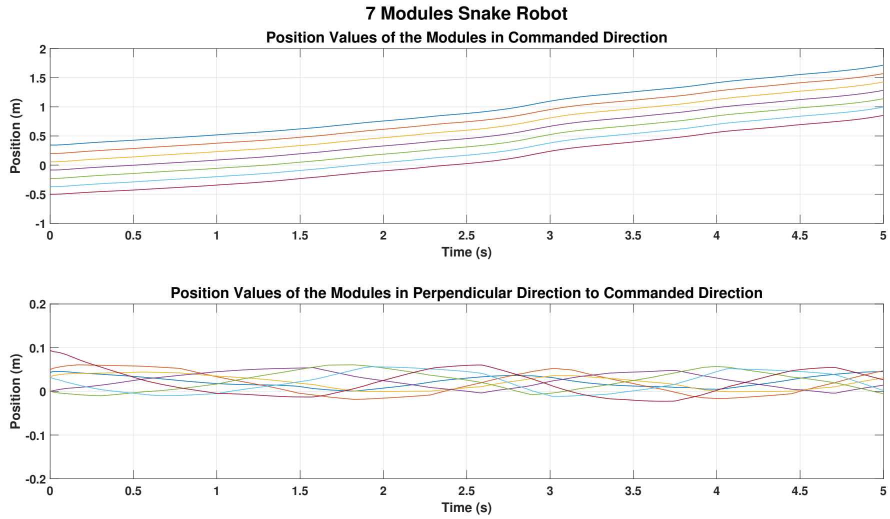

Figure 5.32: Module positions of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set three when initial states are randomized.

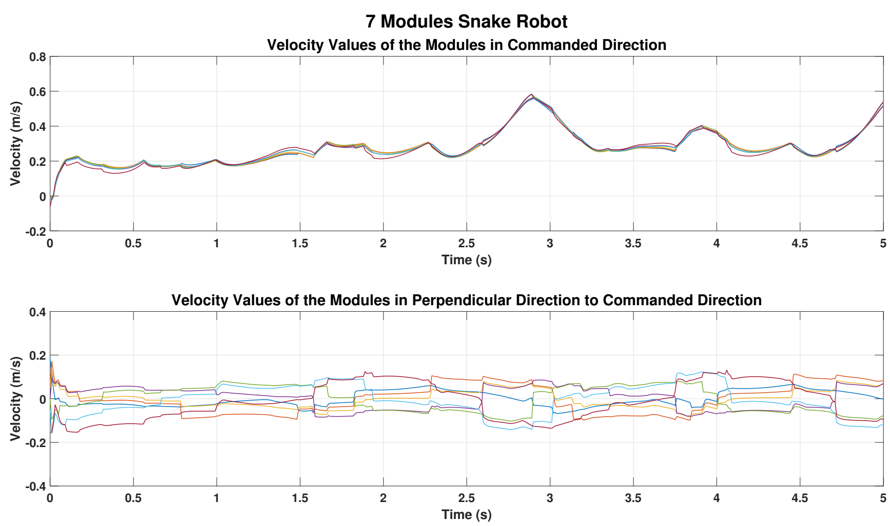

Figure 5.33: Module velocities of 7 modules snake robot with unexperienced contact surface conditions scenario as described contact force modelling parameter set three when initial states are randomized.

5.2 Snake Robot Locomotion in Three Dimensions

In the scope of previous sections, snake locomotion patterns are generated through *TD3* algorithm in 2D environments for distinct cases and robots which are formed with different number of modules where they might also include defective components. At this stage, snake locomotion is extended to 3D environment with a cascaded PID architecture which comprises command shaper and gain scheduling components where cascaded structure provides possibility for operations in both speed control and position control depending on the aimed purpose.

Therefore, snake locomotion patterns are generated via trained *TD3* algorithm to realize locomotion in commanded direction and motion capabilities are extended to 3D with the cascaded PID architecture and high level planner possibilities. General motion control system framework is presented in Figure 5.34 and it should be noted that, depending on current posture of the snake robot, reinforcement learning based and cascaded PID based components with high level planners in the proposed hybrid motion control system can be inter-switched to appropriately drive the corresponding axes.

In the cascaded PID structure, the command shaper component re-forms the applied step commands to realizable S-shape commands in position and triangular-like shapes in speed controller layers by taking into account current states, acceleration and speed limit characteristics of the system. To illustrate, several applied and shaped command examples are provided in Figure 5.35 for the cases the system is commanded from zero initial states and it does not reach to predetermined speed limit. However, it should be emphasized that, the proposed command shaper is also capable to handle with non-zero initial conditions and for the commands which leads to operation in the limit values of the system as several examples are provided in Figure 5.36 for this case.

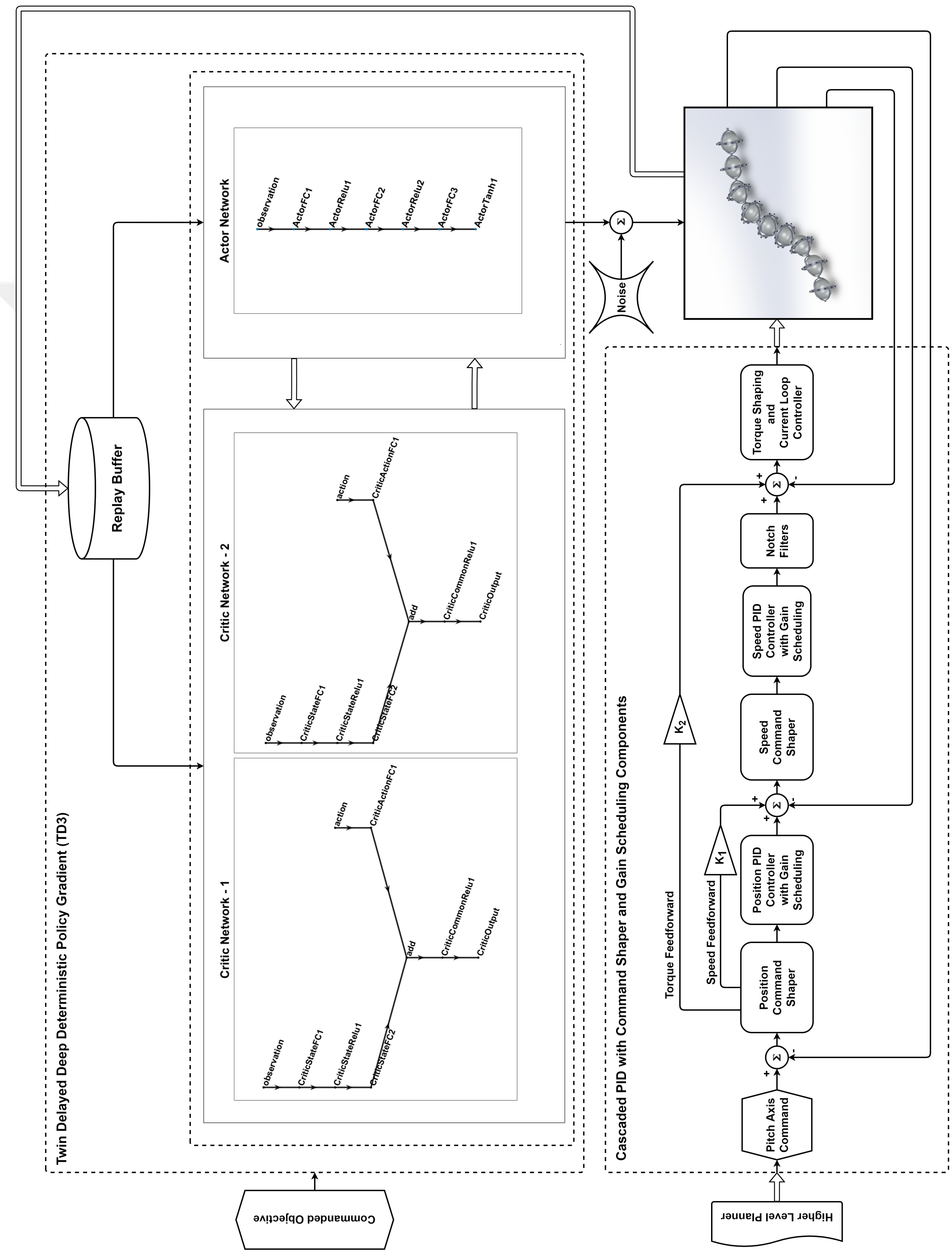

Figure 5.34: Hybrid Motion Control System Framework for Snake Locomotion.

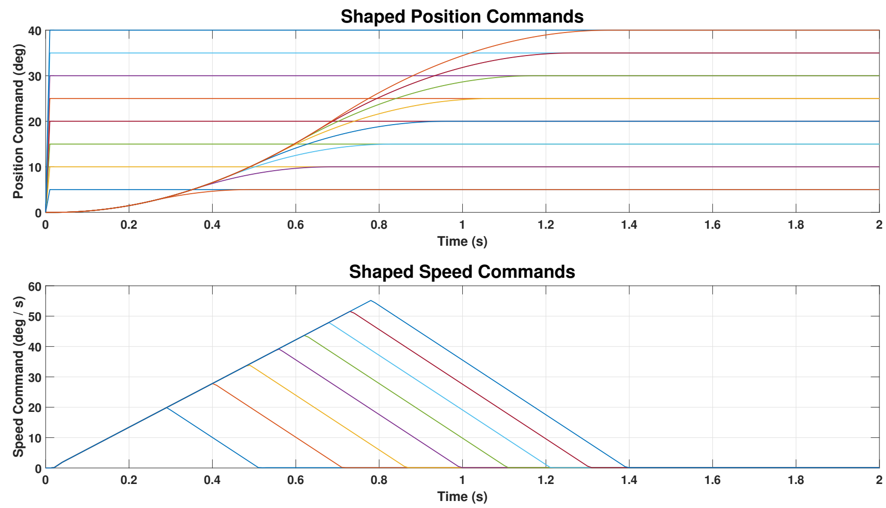

Figure 5.35: Shaped position and speed layer commands for the cases system does not reach predetermined speed limit.

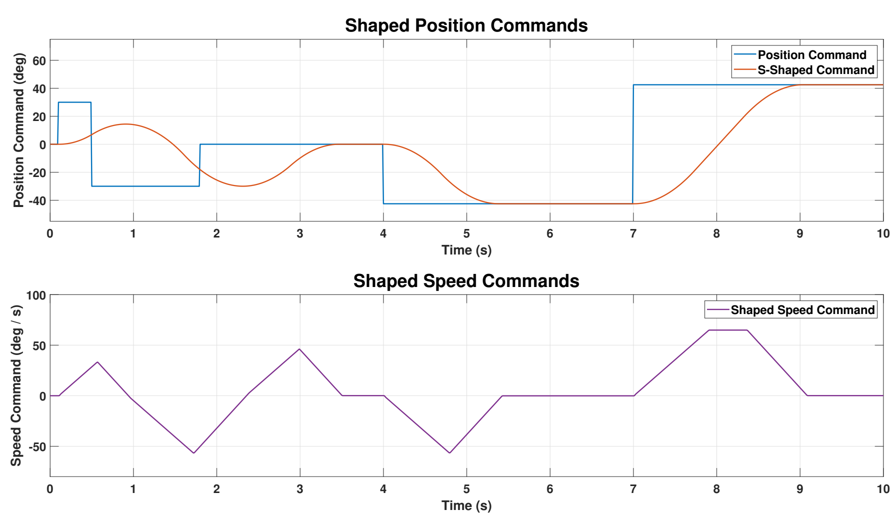

Figure 5.36: Shaped position and speed layer commands for the cases when new commands are applied before previous command is not completely realized, when new commands are applied after previous command is completely realized and when system reaches predetermined speed limit value while realizing applied command.

As the other employed component in the cascaded PID structure, gain scheduling blocks simultaneously change the gain values in an exponential manner between predetermined minimum and maximum borders depending on the instant error value as it is defined in Equation 5.1 and illustrated with arbitrary parameters in Figure 5.37. Gain scheduling is implemented in both proportional and integral components of PID controllers but derivative components are used with a significantly lower gain value and not scheduled for snake locomotion. The implemented gain scheduling component is formulated as in Equation 5.1 below.

$$G_s = G_{min} + \frac{(G_{max} - G_{min})}{e^p}$$
$$p = \begin{cases} 0, & K|\varepsilon| \leq 0 \\ K|\varepsilon|, & 0 < K|\varepsilon| \leq c \\ c, & K|\varepsilon| > c \end{cases} \tag{5.1}$$

where;

ε: Instant control error value

c: Saturation limit for power value

G_{min}, G_{max}, K: Parametric values for gain scheduling

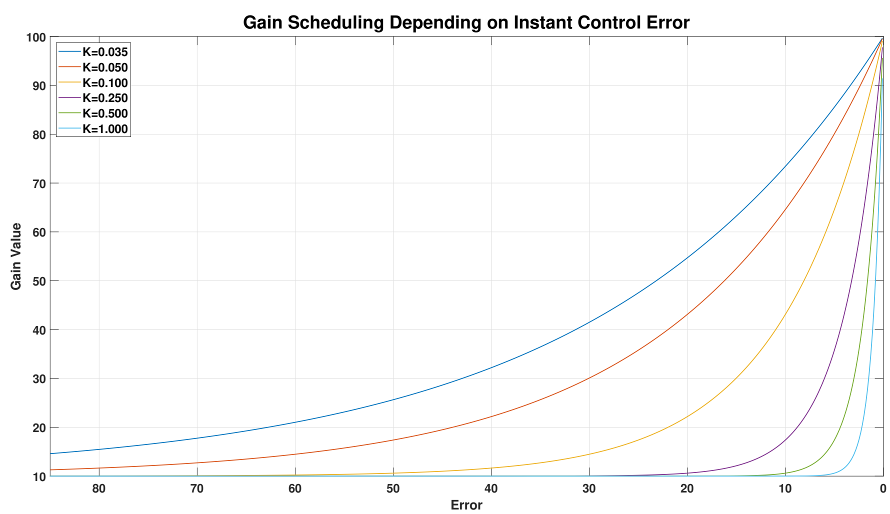

Figure 5.37: Scheduled gain curves based on exponential manner between arbitrarily predetermined lower and upper values.

With presented motion control system framework, snake locomotion is realized in 3D environment with mainly 2 different snake robot postures where they can be diversified further depending on the purpose.

In the first case, an *L-shape* posture is realized where corresponding position and speed information of joints with corresponding commands are presented in Figure 5.38 and the resulting snake posture in three different time steps is illustrated in Figure 5.39.

In the second case, an *arch shape* posture is realized. Position and velocity values of pitch joints of each module are presented in Figure 5.40 while gradual formation of arch posture is depicted in Figure 5.41.

In both scenarios, the joint position commands are applied gradually based on the foresight prediction of expected position values of the joints by the command shaper. The subsequent command is applied after the previous one is expected to settle in ± 1 degree interval around targeted point. In *L-shape* realization, position and speed commands are shaped with $1/3$ of the predetermined acceleration value of previous joint since the moved inertia gradually increases whereas in *arch shape* scenario, gradually applied commands are shaped with equal parametric values.

At this point, although two example scenarios are described and corresponding outcomes are presented, 3D locomotion cases can be diversified depending on the aimed purposes with correspondingly required high level planning operations.

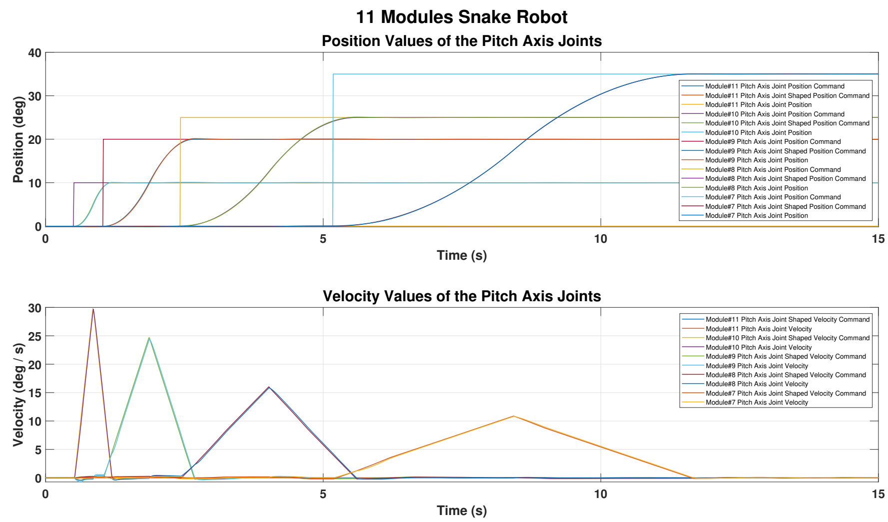

Figure 5.38: Position and velocity values of pitch axis joints for L-shape snake robot realization case.

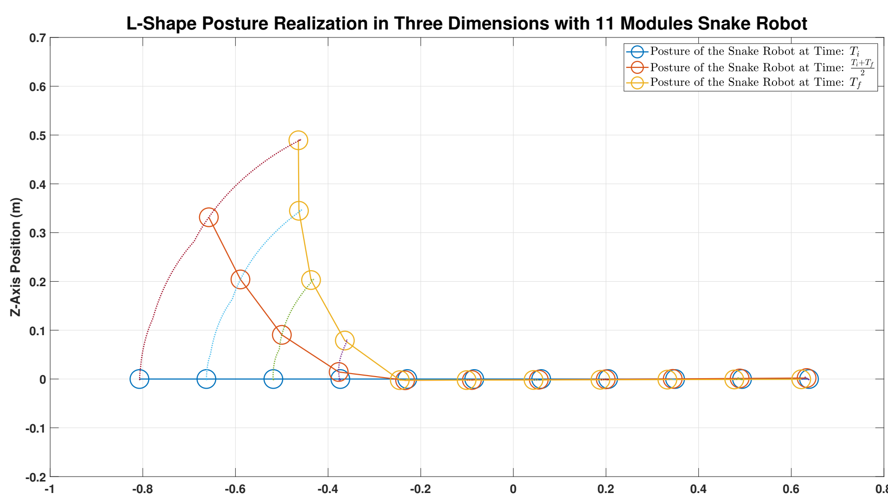

Figure 5.39: L-shape snake robot posture at initial, intermediate and final time steps with followed module paths.

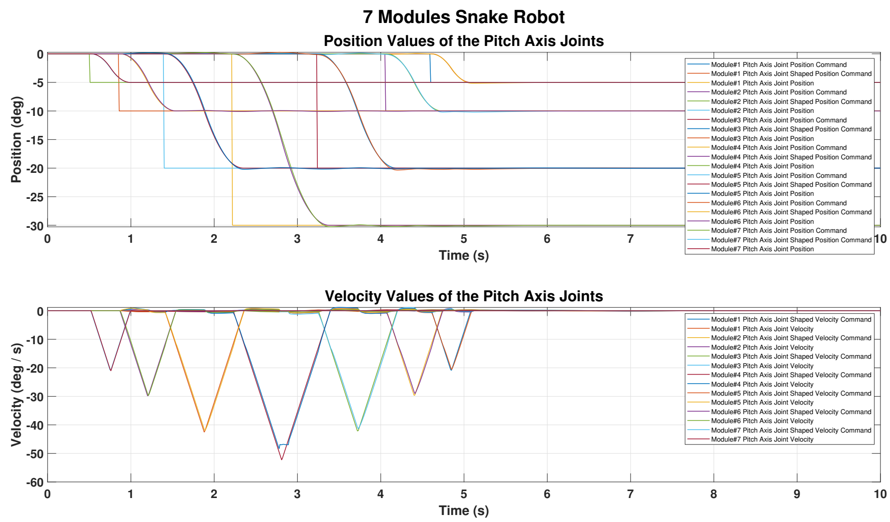

Figure 5.40: Position and velocity values of pitch axis joints for arch shape snake robot realization case.

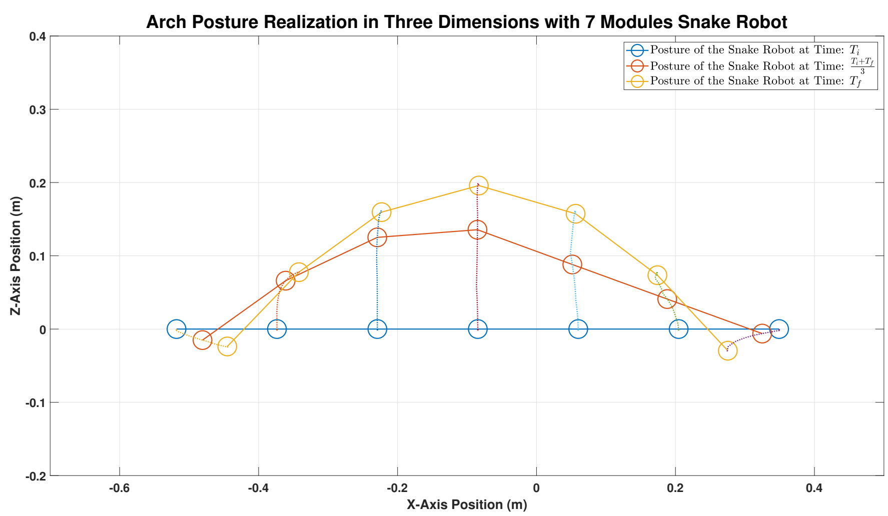

Figure 5.41: Arch shape snake robot posture at initial, transition and final time steps with followed module paths.

Chapter 6

Conclusions and Discussions

In the scope of the thesis, mathematical background essentials for snake robots are investigated and fundamental requirements are presented. Differences and similarities between biological snakes and electromechanical snake robots are discussed. Advantages and disadvantages of snake robots are evaluated with possible application fields and operation concepts.

For implementation and realization of the proposed hybrid motion control system architecture with different cases which comprises snake robots with different number of modules which might also include defective components, various surface characteristics, two dimensional and three dimensional tasks; a reconfigurable modular snake robot is designed and corresponding modelling operations are realized including contact force modelling of passive wheeled structure which satisfies anisotropic friction requirement for proper controllability of the snake robot. For snake robot locomotion, two different state-of-the-art reinforcement learning algorithms which are *Soft Actor-Critic (SAC)* and *Twin Delayed Deep Deterministic Policy Gradient (TD3)* are constructed with corresponding tuning operations of the hyperparameters. For primary snake locomotion gait patterns in 2D, corresponding reward function terms are composed and weights for each components are tuned based on the experiences of snake locomotion characteristics that have occurred during the training trials. With the finalized reward function in terms of components which form the overall reward function and corresponding weight terms for the each component, the snake robots which are formed by 5, 7 and 9 modules are trained to realize appropriate snake locomotion patterns. In addition to these, a defective 11 modules snake robot whose certain joints have become dysfunctional are also trained with the same algorithmic structure and reward functions as a fault tolerant locomotion trainings. In this content, functionality of the designed algorithm and proposed reward function are verified for the aimed purpose. Thereafter, the operational cases are extended to three dimensional scenarios with cascaded PID architecture which can be integrated with various high level planners and verifying outcomes of corresponding experimentations are presented. In this content, proposed cascaded PID architecture for pitch axis joints provides broad application infrastructure for different kind of high level planners via its operation capacity both in speed and position modes independently and instantaneously for each joint. Additionally, command shaper component provides mainly two significant advantages. Since it takes into account current states and electro-mechanical capacity of the system for determination the command in the following time steps, command tracking performance is significantly improved with minimized overshoot and undershoot effects. Also, future steps of the system becomes accurately foreseeable for a specific sample time interval.

In this context, the fundamental contributions of the thesis can be put in order as;

1. A reinforcement learning based motion control architecture is proposed for generation of gait patterns in 2D for snake robots which are formed by different number of modules.

2. For realization of locomotion through reinforcement learning, novel reward functions are designed for the desired gait patterns and effects of each term on locomotion characteristics are investigated.

3. As a fault tolerant locomotion training, functionality of implemented reinforcement learning motion control architecture is also verified with defective

snake robots which include various malfunctions through realization of corresponding experimentations.

4. In addition to electro-mechanical modelling operations, modelling of contact forces with various parameter sets which represent different locomotion terrain characteristics are also realized. Thereafter, functionality of snake robot locomotion, with previously trained algorithms, is also verified with unseen surface conditions. Therefore, generalizability and robustness of the proposed motion control system and trained algorithms are validated for distinct operation conditions.

5. For realizations of operations which span 3D medium, a cascaded PID architecture with command shaper and gain scheduling components is proposed where it can be conveniently integrated with different high level planners. Accordingly, sample operation scenarios are defined, realized and evidential outcomes are presented.

6. Consequently, a hybrid novel motion control system architecture is presented for snake robots which is versatile for implementation in different snake robots and for realization of various high level operation duties where it is also robust against faults and dysfunctionalities that can possibly take place during operation time.

In addition to the obtained and presented successful outcomes, some specific points can also be evaluated and discussed to improve resulting outcomes further. Although the proposed reward function fulfills the existing deficient point for snake locomotion through reinforcement learning, it is still open for further improvements in terms of faster convergence and increased efficiency of robot locomotion. Especially, the components like L_p and Z_r which may produce sparse values depending on the initial conditions of the robot, parameter initializations of the networks and gained locomotion behaviors during training process, can be arranged further to increase efficieny. Also, hyperparameter tunings for employed algorithms are open for further improvements. Even though *TD3* clearly outperforms SAC for designed and modelled snake robot with separately tuned hyperparemeter sets, in a specific setting in terms of both determined hyperparameters by tuning and characteristics of the trained robot, *SAC* algorithm can reach or even outperform *TD3* in contrary to obtained results in the thesis. Therefore, it should be emphasized that, for the specific architecture of snake robot which is employed in the scope of the thesis, *TD3* has created clearly superior results compared to *SAC* with detailed tuning operations as practically possible for both of the algorithms and, for this reason, following investigations and experimentations have been realized through *TD3* algorithm. However, it does not point out to *TD3* superiority over *SAC* in a generalized aspect.

Throughout the investigations and experimentations, simulation and consequently training times emerged as the main problem and fundamental limiting factor. As it can be seen in the outcomes which are presented in Chapter 5, minor level of imperfections remained with trainings of 5000 episodes. However, it is foreseen that, they can be eliminated only by increasing the number of episodes or by further repetition of the trainings. With advanced hardwares and/or more efficient simulation environments, number of episodes can be extended by keeping the training times in an acceptable level and corresponding imperfections can be eliminated as well. In this way, minor deviations in the position values of drived snake robot from the commanded route and minor differences between velocity curves between different time intervals diminish. Similarly, chatter-like undesired behaviors which are observed in velocity curves of 5 modules and 9 modules snake robots can be eliminated as well.

In the training process, as aforementioned, initial angular position and velocity states of the yaw axis joints are randomized. Although the main reason for randomization of states is to prevent possible memorization of snake locomotion characteristics which require zero initial conditions for a successful outcome, it is also experienced that randomization is also beneficial to eliminate potential problems of convergence to a local minima which prevents continuous snake locomotion. In other words, randomization is also useful for exploration of proper gait patterns. In this way, the snake robot becomes capable to realize appropriate gait patterns independent from the initial conditions which can deviate from zero initial states especially in position values as it is inevitable in physical realizations and it can reach significantly more desirable local minima points with realized trainings where it consequently leads superior locomotion performance. However, it is seen that possibility of randomization, which is kept as 0.5 during the trainings, can be increased as the snake robot gains fundamental locomotion attitudes. In this way, risk of failure or degraded performance in rarely occurring extreme cases in terms of position and speed values can be decreased. It can also be underlined that, as expected, randomization also leads to a noisier characteristics in reward values which points out learning process as it can be realized by comparing presented outcomes in Figure 5.1 and Figures 5.2, 5.7, 5.12, 5.17. In addition to randomization process of initial states of the robot, gradual diversification and randomization of contact surface properties can also be integrated into the training curriculum even though operation capacity under unexperienced conditions is verified with corresponding experimentations as presented in Chapter 5. Also, to state as a finalizing remark, a relatively aggressive exploration and learning attitude is adopted in the trainings in terms of both exploration noise level, update and learning parameters. For this reason, an aggressive learning attitude emerges as an additional factor which elevates noisy characteristics in reward values. However, it should be noted that an aggressive attitude is required to escape from undesired local minimas and it does not cause an instability problem in the learning process.

Chapter 7

Targets of the Future

The snake robot locomotion which reflects different patterns which can overcome posssible challenges in distinct environments with their robust and versatile potential capabilities is taken into account in a holistic view in this study. In this content, a number of further steps are determined where they can be described with underlying reasons one by one as below.

First of all, *MATLAB Simulink* [37] is employed as the simulation environment for training process and corresponding electro-mechanical and locomotion environment modellings are realized via low and high level tools of *Simulink*. However, despite the advantages which *Simulink* provides especially for electro-mechanical modelling, two significant drawbacks can be stated as solving time of simulations and infeasibility of environmental modellings. To reflect complex, real-world like environmental conditions and variations in surface properties, it is aimed that the robotic architecture will be transferred and trained in *Isaac Gym* [54] or *MuJoCo* [55] simulation environments. They also provide significant advantages in terms of solving times and consequently required time for training processes comparing with *MATLAB Simulink* while sufficient physical accuracy is preserved.

Secondarily, parallelization of the training process is another future target which will accelerate the learning process and correspondingly shorten the required training time drastically. *Isaac Gym* provides an end-to-end GPU based physics simulation for robot learning in a parallelized manner [54]. In this way, for training process, thousands of robots are used simultaneously and obtained experiences are collected to update the policy. In [56], it is shown that *ANYmal-C* quadrupedal robot learns to walk in under 20 minutes training by employing 4096 robots simultaneously in *Isaac Gym* environment and on *NVIDIA RTX A6000* GPU [57] where it is capable to walk in physical environment directly after training in the simulation environment while proposed parallel training approach is also validated with different industrial robot models which are *Unitree A1* and *Cassie* in the simulation environment. In this content, training of the presented modular and reconfigurable snake robot is aimed to be trained with different number of modules in an environment where different surface properties and obstacles are introduced.

In the scope of the thesis, mainly *lateral undulation* gait is taken into account with their derivatives which emerge during the training processes. In addition to these, *sidewinding* or *lateral rolling* based gait patterns for lateral motion capabilities and *concertina* locomotion patterns for especially cluttered environments as obstacle-aided locomotion will also be investigated. In a case when parallel training is employed, since the required times for trainings will be significantly decreased and therefore major practical concern will be eliminated, investigations of different snake characteristics will become much more feasible.

In the scope of thesis, electromechanical design details for physical realization possibility in the future is not considered in detail as required for manufacturing phase. At this point, possible opportunities especially in soft materials will be evaluated for the mechanical frame. In [58], Hawkes *et al.* presents a vine-like growing soft robot which evokes a snake robot concept by its architecture and operation purposes. A similar approach can be adapted to the modular snake robot design to acquire re-shaping capability and to enhance motion capability over narrower regions than the module diameter of snake robot. Relatedly, mechanical design details of self-assembly mechanisms are not considered for a self-reconfiguration capability instead of human intervention requirement. In literature; PolyBot [6], FreeBOT [20], M-Blocks [21], [59], [60], SMORES [22], Roombots [23], ATRON [61], [24], UBot [25] and Sambot [26], [62] represents remarkable robotic systems which holds self-assembly, self-reconfiguration capabilities as presented in the Figure 7.1. In this content, not only snake robots with different number of modules can be designed but also defined or undefined robotic structures different than snakes can also be constructed and corresponding locomotion capabilities can be searched over emergent architecture.

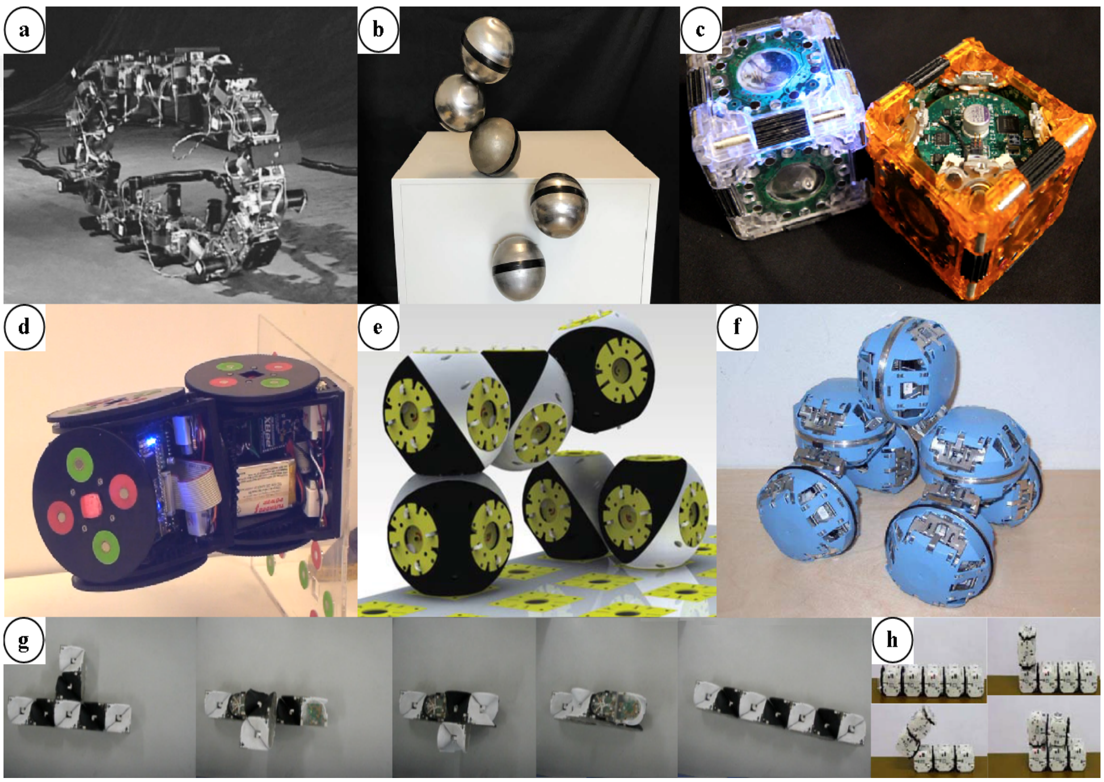

Figure 7.1: Self-reconfigurable modular robotic systems.
(a) PolyBot [6], (b) FreeBOT [20], (c) M-Blocks [21], (d) SMORES [22], (e) Roombots [23], (f) ATRON [24], (g) UBot [25], (h) Sambot [26].

In a multi-agent case which is composed by snake robots with same or different number of modules; a learning based coordinated, collaborative swarm behaviors and realization of common assignments like simultaneous localization and mapping (SLAM) in a collaborative manner will be investigated. Scope will be gradually expanded in terms of both capabilities of individual robots and realized tasks in multi robot cases. Consequently, realization of a more comprehensive learning based framework which comprises robotic locomotion and higher level decision making and planning is aimed in a more efficient manner.

Bibliography

[1] M. Luo, R. Yan, Z. Wan, Y. Qin, J. Santoso, E. H. Skorina, and C. D. Onal, “Orisnake: Design, fabrication, and experimental analysis of a 3-d origami snake robot,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1993–1999, 2018.

[2] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hatton, and H. Choset, “Design of a modular snake robot,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2609–2614, IEEE, 2007.

[3] A. Wolf, H. B. Brown, R. Casciola, A. Costa, M. Schwerin, E. Shamas, and H. Choset, “A mobile hyper redundant mechanism for search and rescue tasks,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 3, pp. 2889–2895, IEEE, 2003.

[4] C. Wright, A. Buchan, B. Brown, J. Geist, M. Schwerin, D. Rollinson, M. Tesch, and H. Choset, “Design and architecture of the unified modular snake robot,” in *2012 IEEE international conference on robotics and automation*, pp. 4347–4354, IEEE, 2012.

[5] G. Miller, “13 snake robots for search and rescue,” *Neurotechnology for Biomimetic Robots*, p. 271, 2002.

[6] M. Yim, D. G. Duff, and K. D. Roufas, "Polybot: a modular reconfigurable robot," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1, pp. 514–520, IEEE, 2000.

[7] M. Yim, Y. Zhang, K. Roufas, D. Duff, and C. Eldershaw, "Connecting and disconnecting for chain self-reconfiguration with polybot," *IEEE/ASME Transactions on mechatronics*, vol. 7, no. 4, pp. 442–451, 2002.

[8] A. Brunete, E. Gambao, J. E. Torres, and M. Hernando, "A 2 dof servomotor-based module for pipe inspection modular micro-robots," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1329–1334, IEEE, 2006.

[9] R. Haraguchi, K. Osuka, S. Makita, and S. Tadokoro, "The development of the mobile inspection robot for rescue activity, moira2," in *ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, pp. 498–505, IEEE, 2005.

[10] J. Borenstein, M. Hansen, and A. Borrell, "The omnitread ot-4 serpentine robot—design and performance," *Journal of Field Robotics*, vol. 24, no. 7, pp. 601–621, 2007.

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

[12] M. Mori and S. Hirose, "Development of active cord mechanism acm-r3 with agile 3d mobility," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 3, pp. 1552–1557, IEEE, 2001.

[13] H. Yamada, "S. development of amphibious snake-like robot acm-r5," in *the 36th International Symposium on Robotics (ISR 2005), Tokyo*, 2005.

[14] T. Ohashi, H. Yamada, and S. Hirose, "Loop forming snake-like robot acm-r7 and its serpenoid oval control," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 413–418, IEEE, 2010.

[15] J. C. McKenna, D. J. Anhalt, F. M. Bronson, H. B. Brown, M. Schwerin, E. Shammas, and H. Choset, "Toroidal skin drive for snake robot locomotion," in *2008 IEEE International Conference on Robotics and Automation*, pp. 1150–1155, IEEE, 2008.

[16] H. Ohno and S. Hirose, "Design of slim slime robot and its gait of locomotion," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 2, pp. 707–715, IEEE, 2001.

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

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

[19] Britannica, "Encyclopedia Britannica." Available at https://www.britannica.com (2021/10/12).

[20] G. Liang, H. Luo, M. Li, H. Qian, and T. L. Lam, "Freebot: A freeform modular self-reconfigurable robot with arbitrary connection point-design and implementation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6506–6513, IEEE, 2020.

[21] J. W. Romanishin, K. Gilpin, S. Claici, and D. Rus, "3D M-Blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1925–1932, IEEE, 2015.

[22] J. Davey, N. Kwok, and M. Yim, "Emulating self-reconfigurable robots-design of the SMORES system," in *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 4464–4469, IEEE, 2012.

[23] A. Sproewitz, A. Billard, P. Dillenbourg, and A. J. Ijspeert, "Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture," in *2009 IEEE international conference on robotics and automation*, pp. 4259–4264, IEEE, 2009.

[24] D. Brandt, D. J. Christensen, and H. H. Lund, “ATRON robots: versatility from self-reconfigurable modules,” in *2007 International Conference on Mechatronics and Automation*, pp. 26–32, IEEE, 2007.

[25] J. Zhao, X. Cui, Y. Zhu, and S. Tang, “A new self-reconfigurable modular robotic system UBot: Multi-mode locomotion and self-reconfiguration,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 1020–1025, IEEE, 2011.

[26] H. Wei, Y. Chen, J. Tan, and T. Wang, “Sambot: A self-assembly modular robot system,” *IEEE/ASME transactions on mechatronics*, vol. 16, no. 4, pp. 745–757, 2010.

[27] J. Whitman, N. Zevallos, M. Travers, and H. Choset, “Snake robot urban search after the 2017 mexico city earthquake,” in *2018 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, pp. 1–6, IEEE, 2018.

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

[29] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, “Continuum robots for medical applications: A survey,” *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[30] S. Hirose and M. Mori, “Biologically inspired snake-like robots,” in *2004 IEEE International Conference on Robotics and Biomimetics*, pp. 1–7, IEEE, 2004.

[31] B. C. Jayne, “What defines different modes of snake locomotion?,” *Integrative and comparative biology*, vol. 60, no. 1, pp. 156–170, 2020.

[32] S. Alben, “Optimizing snake locomotion in the plane,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 469, no. 2159, p. 20130236, 2013.

[33] X. Wang, M. T. Osborne, and S. Alben, “Optimizing snake locomotion on an inclined plane,” *Physical Review E*, vol. 89, no. 1, p. 012717, 2014.

[34] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A simplified model of planar snake robot locomotion," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2868–2875, IEEE, 2010.

[35] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Controllability analysis of planar snake robots influenced by viscous ground friction," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3615–3622, IEEE, 2009.

[36] P. Liljeback, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Controllability and stability analysis of planar snake robot locomotion," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1365–1380, 2010.

[37] MathWorks, "Simulink: Simulation and Model-Based Design." Available at https://www.mathworks.com/help/simulink/ (2021/10/09).

[38] DassaultSystemes, "SOLIDWORKS: 3D CAD Design Software." Available at https://www.solidworks.com (2021/09/08).

[39] MathWorks, "Simscape Multibody: Model and simulate multibody mechanical systems." Available at https://www.mathworks.com/help/physmod/sm/ (2021/10/09).

[40] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3, pp. 279–292, 1992.

[41] H. Van Seijen, H. Van Hasselt, S. Whiteson, and M. Wiering, "A theoretical and empirical analysis of expected sarsa," in *2009 ieee symposium on adaptive dynamic programming and reinforcement learning*, pp. 177–184, IEEE, 2009.

[42] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[43] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*, pp. 387–395, PMLR, 2014.

[44] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, pp. 1889–1897, PMLR, 2015.

[45] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[46] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. Eslami, *et al.*, "Emergence of locomotion behaviours in rich environments," *arXiv preprint arXiv:1707.02286*, 2017.

[47] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.

[48] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[49] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, pp. 1587–1596, PMLR, 2018.

[50] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[51] H. Mania, A. Guy, and B. Recht, "Simple random search provides a competitive approach to reinforcement learning," *arXiv preprint arXiv:1803.07055*, 2018.

[52] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, "Learning to walk via deep reinforcement learning," *arXiv preprint arXiv:1812.11103*, 2018.

[53] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Physical review*, vol. 36, no. 5, p. 823, 1930.

[54] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[55] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033, IEEE, 2012.

[56] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*, pp. 91–100, PMLR, 2022.

[57] NVIDIA, "NVIDIA RTX A6000." Available at https://www.nvidia.com/en-gb/design-visualization/rtx-a6000/ (2022/06/06).

[58] E. W. Hawkes, L. H. Blumenschein, J. D. Greer, and A. M. Okamura, "A soft robot that navigates its environment through growth," *Science Robotics*, vol. 2, no. 8, p. eaan3028, 2017.

[59] J. W. Romanishin, K. Gilpin, and D. Rus, "M-blocks: Momentum-driven, magnetic modular robots," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4288–4295, IEEE, 2013.

[60] J. J. W. Romanishin, *M-Blocks: Three dimensional modular self-reconfigurable robots.* PhD thesis, Massachusetts Institute of Technology, 2018.

[61] M. W. Jorgensen, E. H. Ostergaard, and H. H. Lund, "Modular ATRON: Modules for a self-reconfigurable robot," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 2, pp. 2068–2073, Ieee, 2004.

[62] H. Wei, Y. Cai, H. Li, D. Li, and T. Wang, “Sambot: A self-assembly modular robot for swarm robot,” in *2010 IEEE International Conference on Robotics and Automation*, pp. 66–71, IEEE, 2010.