TOWARDS FAULT DIAGNOSIS IN RECONFIGURABLE ROBOT

DISSERTATION

Karthikeyan Elangovan

Department of Advanced Multidisciplinary Engineering

Graduate School of Advanced Science and Technology

Tokyo Denki University

Supervisors:

Professor, Masami Iwase,	Tokyo Denki University
Professor, Mohan Rajesh Elara,	Singapore University of Technology and Design

April 2018

Acknowledgements

I would like to take this opportunity to extend my sincere thanks and appreciation to my supervisor, Professor Masami Iwase, Tokyo Denki University, for his constant guidance and encouragement.

I want to take this opportunity to extend my sincere thanks and appreciation to my supervisor, Professor Mohan Rajesh Elara, Singapore University of Technology and Design, for his unwavering support, collegiality, and mentorship throughout this project.

I am indebted to Professor Kristin L. Wood, Co-Director (SUTD-MIT IDC), Founding Head of Pillar (EPD) & Associate Provost of Graduate Studies, Singapore University of Technology and Design for reviewing my work.

I would like to show my appreciation to Yokhesh, Takuma, Vinu, Prathap, Harsha, and the Researchers at ROAR Lab, Singapore University of Technology and Design for their support and assistance in this project.

Finally, I am most grateful to my wife, Sivaranjani, my son, Nadhiyuga Dheeren, and my parents for their continuous motivation, strong moral support, understanding and love.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Motivation

The ascent need for service robots in the globe has been promoted due to the expanding interest for lessening the operational expenses and the time utilization because of human interference. The business is likewise seeing high development openings from the coordination segment attributable to the expanding wage rates, the capacity to build the throughput, and an exceptional yield on venture offered by these machines. Besides, the expanding interest for service robots and interest towards automation their high-level performance and precise work has let to its growth over different segments, ranging from healthcare, farming-agriculture, retail, defence and military. This progressing innovative in the field of robotics and automation has automatically promote to the business request globally. The headways of innovation in the robotics field such as IoT, machine learning, AI has made the reception of this technology more valuable and welcoming. For more enhancement in the performance of these robotics many manufacturers are endeavouring to consolidate such innovations and new developments for having an upgraded and advanced execution of work by the use of robots. To improve the general usefulness of the robot, deep learning technology and IoT has been increasingly used for advancement into the frameworks. It empowers them to make choices in the continuous in vigorous mechanical conditions and unusual circumstances, impelling the market development.

The prerequisite of high capital ventures, expanding Research and development costs, and the multifaceted nature related with programming the robots may thwart the fast market development. Besides, the surprising expense of these machines is relied upon to influence the reception rates in ventures that have budgetary limitations. Absence of awareness about these robots in the rising economies combined with the wellbeing concerns related with the utilization of such machines is constraining the market development. By certain prediction timelines it is also expected that there would be high needs for personal service robots . With increasing ageing population, there is a high need which witness the development and needs for growing demand for the personal assistance across households. The development can be ascribed to the expanding interest for individual help crosswise over family units combined with the developing maturing populace in nations including Japan and Germany. Quickly maturing populace in these nations creates a

popularity for cutting-edge therapeutic care and nursing arrangements. With the developing number of senior nationals, the quantity of caregivers figures stays stale. With the help of certain new technologies like machine learning and artificial intelligence these machines are equipped for giving great help to elderly individuals by diagnosing few simple ailments.

The robotics market has been developing recently where the increased interest and need for automation in the developing fast economies worldwide has been a catalyst for this robotic industry. Robots for labour works such as Vacuum cleaning, floor cleaning, and garden cutting robots are picking up prominence in the private segment and these are progressively used for performing huge residential assignments, for example, cleaning and wiping, considering the high expenses related with procuring labourers for such undertakings. A new entrepreneurial culture sees a huge need for logistics service robots is on the ascent in the near future. The development can be certify to the expanding organization of such robots in warehouses to get operational productivity crosswise over different business procedures, for example, packaging, bundling, load and unloading. These robots offer a good level of precision, safety, working hours, speed and productivity over a scope of utilization when contrasted with humans. The developing interest for robots in the defence and military services gives lucrative development chances to the safeguard benefit apply autonomy showcase for gathering information, surveillance, border control, certain intelligence needs, demining, etc. Nations including the U.S. are intensely putting resources into the barrier part and plan to cut back military by supplanting human troops with mechanically progressed self-sufficient or remotely-controlled robots.

A new framework with regards to automation has seen an increased boom in the robotics field, the morphology of self-reconfiguration and automation together as a framework seems to be a new beneficial innovation which work as good automatic kinematic machines. Functions such as actuating, sensing and control are found to be settled in these morphology robots, self-reconfiguring robots are additionally ready to intentionally change their own shape by modifying the network of their shape, keeping in mind the end goal to adjust to new conditions, perform new errands at various circumstance, or recuperate themselves from harm. For instance, a robot made of such parts could accept a worm-like shape to travel through a restricted pipe, reassemble into something with insect-like legs to cross uneven territory, at that point frame a third discretionary protest (like a ball or wheel that can turn itself) to move rapidly over a levelled landscape.

These robotic framework contain a couple of modules that are associated with each other using a couple of connectors. They stay in connection with one another by using various sensors, electronics, memory, power supplies where they would be able to manipulate certain things and utilize them to stay in connection and control with one another. These possess the ability and capacity of these modules to naturally interface and detach themselves to and from each other, and to frame into numerous articles or perform numerous assignments moving or controlling itself over the designated area. Mentioning "self-reconfiguring" or "self-reconfigurable" it implies that the machine is fit for using its own particular arrangement of control, for example, with actuators or stochastic intends to change its general auxiliary shape. Having the nature of being "modular" in "self-reconfiguring measured mechanical autonomy" is to state that a similar module or set of modules can be added to or expelled from the framework, rather than being nonexclusively "modularized" in the more extensive sense. The fundamental goal is to have an inconclusive number of indistinguishable modules, or a limited and moderately little arrangement of indistinguishable modules, in a work or network structure of self-reconfigurable modules.

Self-reconfiguration is not quite the same as the idea of self-replication, which is certifiably not a quality that a self-reconfigurable module or accumulation of modules needs to have. It is not necessary that a matrix of modules that should combine itself to increase the capacity into a framework to be viewed as self-reconfigurable. It is adequate for self-reconfigurable modules to be delivered at a regular plant, where devoted machines stamp or shape parts that are then gathered into a module, and added to a current framework with a specific end goal to supplement it to expand the amount or to supplant exhausted modules. A framework made up of numerous modules can separate to shape various networks with less modules, or they can consolidate, or recombine, to shape a bigger grid. A few focal points of isolating into various matrices incorporate the capacity to handle different and more straightforward undertakings at areas that are remote from each other at the same time, exchanging through obstructions with openings that are too little for a solitary bigger framework to fit through however not very little for littler network parts or individual modules, and vitality sparing purposes by just using enough modules to achieve a given assignment. A few focal points of joining different frameworks into a solitary network is capacity to shape bigger structures, for example, a stretched scaffold, more unpredictable structures, for example, a robot with numerous arms or an arm with more degrees of freedom, and expanding strength. The feature of its increasing strength and its quality is the increase in its rigidity, over a fixed or static

structure, and perform various movements itself by pushing, pulling, raising its force, pulling with another module or perform a combination of operations.

One significant contributor hindering the success of reconfigurable robots in both industrial and service domains is the occurrence of faults. Several factors including limitation in robot's hardware (sensors, actuators, etc), software, cognitive capabilities and task complexities contribute towards these faults. Even though fault diagnosis has been extensively studied in the context of fixed morphology robots. However, there are unique attributes and issues within these emergence class of reconfigurable robots in the context of fault diagnosis that has been neglected by the mainstream robotics researchers. These include,

- Self-reconfigurable robots possess the ability to navigate in more than one locomotion modes. If a specific fault occurs and curtain one of the locomotion mode. They could switch to other available valid modes of locomotion to overcome the fault. For example, in case of a wheg robot. If a fault has occurred with wheeled locomotion, the robot could switch to a legged mode to still continue to operate and accomplish the given task. However, this is only possible if they are capable of detecting the occurrence of a fault. This largely impact the performance of such robots.
- Additionally, reconfigurable robots possess an extended set of locomotion gaits as compared to fixed morphology robots. This due to the availability of alternative morphologies and associated gaits. The occurrence of faults in this case goes beyond just one mode of locomotion but also applicable to all the available set of morphologies. Also, these robots possess unique transformation gaits that enable them to transform from one form to another. The fault diagnosis studies are then required to include these new class of transformation gaits that are absent in fixed morphology platforms.
- Finally, the availability of multiple morphologies leads to a unique class of false positive and false negative faults. For example, in the false positive case of a rolling-crawling robot. The robot may believe that it has transformed from a crawling state to a rolling state but while it may still remain in the crawling state due to a specific fault. In false negative case, the robot may believe that it has not made any transformation and remains in crawling state but a transformation may have occurred due to presence of a fault transforming the robot into rolling state.

- Consequently, the ever-increasing application domains for reconfigurable robots in diverse fields and increasing need for robots to operate autonomously overshadowed by the above challenges inspire and motivate us to investigate and implement learning approaches towards fault diagnosis. This work focuses on extension of traditionally adopted learning models for reconfigurable class of robots to accommodate for the additional demand incurred due to increased complexity and features.

1.2 Objectives

The main objective of the presented thesis is to develop machine aided fault diagnosis algorithms to enhance performance, safety and reliability of various missions using unmanned systems. To tackle these issues, the proposed approach is based on the data-driven fault detection paradigm. This class of fault detectors relies on the assumption that in order to detect faults it is sufficient to gather fault-sensitive data in the target system, derive a model from it and apply the model to decide whether the system behaves normal or not. For complex systems this procedure is beneficial since the model can be learned instead of generating it manually which can be a tedious task and requires system expert knowledge. The approach proposed in this work exploits the support vector machine and neural network-based learning approaches to a wide range of faults across multiple robot platforms.

Fault detection is basically a classification task. This classification can be realized either with one or with several classes. In the former case the classifier maintains only one class which represents the system's normal behaviour and outliers to this class are regarded as faults. In the latter case the classifier possesses additional classes representing faulty states and detection takes place by deciding to which class the current system behaviour belongs to. In this work, I therefore choose to exploit both classifiers.

Another contribution made within this thesis is the implementation of the proposed fault detection approach in a modularized way which makes it easy to apply the approach to other robotic systems. Furthermore, an experimental framework was designed which facilitates the repetitive execution of experiments in a rolling-crawling reconfigurable urban reconnaissance robot and a Tetris inspired reconfigurable floor cleaning robot.

1.3 Research Methodology

This thesis focuses on investigation and diagnosis of faults that hinders task performance in reconfigurable robots using machine learning techniques. In particular, literature from the research topics of reconfigurable robotics, fault diagnosis, and machine learning are investigated. The work presented in this thesis proceeds on a theoretical level and on an experimental level.

On the theoretical level, two learning frameworks based on neural network and support vector machine are put forward and validated. In this thesis, we study new ways of detecting faults in autonomous reconfigurable robots. Discovering that a fault has occurred is the first step in the process of ensuring that a robot remains safe and dependable even if something stops functioning correctly. We first study a method that enables a shape-shifting robot to discover that a fault, such as a broken leg, has occurred in itself. There are various ways of detecting such faults: we could for instance add sensors for proprioception such as encoders. If an inertial measurement unit detects that a leg is broken when it should, we could interpret that as a symptom of a fault. Alternatively, we could build a model of how the robot is supposed to behave and compare the actual behaviour to the behaviour predicted by the model. We discuss some of the established techniques in more detail in Chapter 2. We present an alternative method for detecting faults, namely through fault injection and learning. We collect data while a robot is operating normally under the available set of morphologies and after faults have been injected. Based on the collected data, we train the robot to detect the presence of the faults. Our method has the advantage that no additional sensors are needed and we do not need to build an analytical model of how a robot should behave. Also, this thesis presents a new path planning approach based on VPH+.

On the experimental level, two unique reconfigurable robots were designed and built as part of this doctoral thesis. The first robot was a bio-inspired reconfigurable crawling-rolling platform, Scorpio. The second robot was a Tetris inspired reconfigurable floor cleaning robot, hTetro. Exhaustive experiments were run to collect data while operating under normal conditions and after faults have been induced. Based on the extracted data, we used machine learning approaches to train and identify presence of faults in robot operation.

1.4 Organization of the thesis

An overview of the research work carried out and the structure of the thesis in concise form are outlined in this section. The rest of the thesis is organized in the following fashion.

Chapter 3 describes the development of a local path planning algorithm for a bio-inspired, reconfigurable crawling robot. A detailed description of the robotic platform is first provided, and the suitability for deployment of each of the current state-of-the-art local path planners is analyzed after an extensive literature review. The Enhanced Vector Polar Histogram algorithm is described and reformulated to better fit the requirements of the platform. The algorithm is deployed on the robotic platform in crawling configuration and favourably compared with other state-of-the-art local path planning algorithms.

Chapter 4 describes the state of art in bio-inspired design, realization, and validation of a reconfigurable rolling–crawling robot. The developed platform is able to mimic Cebrennus rechenbergi, a species of huntsman spider which can crawl and roll using only its legs. Mechanical design, control architecture, and actuator selection strategies targeting platform miniaturization are presented in detail. The navigating and autonomous capabilities of the robot are examined in two facets: (1) recovery behaviors where a robot in a previously unknown state after a fall recovers autonomously to a known standing gait state using Inertial Measurement Unit (IMU); and (2) terrain perception where the robot is capable of autonomously assessing the characteristics of the terrain and chooses the appropriate morphology and locomotion mode in relation to the perceived terrain.

In Chapter 5, we focus on the issue that as robots begin to perform jobs autonomously, with minimal or no human intervention, a new challenge arises: robots also need to autonomously detect errors and recover from faults. In this study, we present a Support Vector Machine (SVM)-based fault diagnosis system for a bio-inspired reconfigurable robot named Scorpio. The diagnosis system needs to detect and classify faults while Scorpio uses its crawling and rolling locomotion modes. Specifically, we classify between faulty and non-faulty conditions by analyzing onboard Inertial Measurement Unit (IMU) sensor data. The data capture nine different locomotion gaits, which include rolling and crawling modes, at three different speeds. Statistical methods are applied to extract features and to reduce the dimensionality of original IMU sensor data features. These statistical features were given as inputs for training and testing. Additionally, the c-Support Vector Classification (c-SVC) and nu-SVC models of SVM, and their fault classification accuracies, were

compared. The results show that the proposed SVM approach can be used to autonomously diagnose locomotion gait faults while the reconfigurable robot is in operation.

Chapter 6 starts from the background that last three decades have witnessed a steep increase in the number of robots with the ability to autonomously execute jobs with little or no human intervention. This has led to the emergence of a new challenge. Robots should be able to detect their errors autonomously and correct the faults. This study presents a fault diagnosis system that is based on support vector machine (SVM) for a new Tetris-inspired reconfigurable floor-cleaning robot. Morphological errors that involve partial or incomplete transformation may diminish the area coverage performance significantly. Therefore, the diagnosis system should be able to detect as well as categorize morphological errors in all the seven valid hTetro states. As the robot was moving, faulty and non-faulty morphologies were classified through the analysis of feedback data obtained from hinged motors. We applied statistical methods in the extracting features while lowering the features of hinged motor data dimensionality. The statistical features were offered as testing and training inputs. We also compared between the SVM c-SVC (Support Vector Classification) and nu-SVC models together with their fault categorization accuracies and found out that the proposed SVM technique may accurately and autonomously examine morphological faults when the robot is in action.

In Chapter 7, it is summarized at the beginning that as robots witness a steep increase in their autonomous capabilities to undertake complex tasks without human involvement, they should also independently overcome and locate faults and errors. This chapter presents a neural network approach towards classification and detection of fault in the context of Scorpio, a bio-inspired reconfigurable robot. The design of the system was intended for the classification and identification of faults, which arose when Scorpio was in crawling and rolling locomotion. In particular, the system was capable of distinguishing faulty conditions from non-faulty ones by evaluating the IMU sensor data. This study covers 9 unique locomotion gaits, which spanned modes of rolling and crawling at 3 distinct speeds. A backpropagation neural network was adopted for performing the task of classifying faults. A set of statistical methods was utilized for mining the key characteristics and reducing the statistical features' dimensionality of the collated data from the IMU sensors. Such statistical characteristics were subsequently used as inputs during training and testing purposes.

Experimental results show that the proposed neural network approach can be used to autonomously diagnose locomotion gait faults while the reconfigurable robot is in operation.

Finally, Chapter 8 summarizes my thesis and gives concluding remarks.

2 Literature Review

Reconfigurability research in robotics falls into two broad classes: intra-reconfigurability and inter-reconfigurability. Reconfigurable robots are capable of changing their morphologies either internally in an individual scale (intra-) or by merging with other homogeneous or heterogeneous robots (inter-). The strength of such robots lies in maximizing their resilience and flexibility in a wide range of environments and tasks through self-construction, self-deconstruction, and self-repair. A large variety of intra- and inter- reconfigurable robots have been studied over the last three decades. In this chapter, we survey and analyze existing inter-, intra- reconfigurable and nested reconfigurable robots. Many applications such as surveillance, security, search and rescue involving extreme uncertainty and require reconfiguration mechanisms to overcome the challenges posed by the environment/task.

2.1 Intra-Reconfigurable Robots

An intra-reconfigurable robot can be viewed as a collection of components (sensors, actuators, mechanical parts, power, controller, etc.) acting as a single entity while having the ability to change its internal morphology without requiring any external assembly or disassembly. Intra-reconfigurability has been generally centered on functional modules namely mobility, sensing, and computing. Here, we will summarize exemplary cases of functional intra-reconfigurability.

Intra-reconfiguration for mobility allows robots the flexibility of traversing over a variety of terrains and spaces (land, air, and water). Such a robot might transform from wheeled to legged when moving from flat to uneven terrain or turn into a sphere when rolling down slopes. Figure 2.1 depicts the concept of intra-reconfigurability wherein a robot changes its morphology as an individual unit. Fujita et al [1]present an intra-reconfigurable robot based on a standard OPEN-R platform that is capable of transforming from a four-legged form to a wheeled form with a plug and play feature. Another effort towards intra-reconfigurable locomotion mechanisms for enhanced mobility is AZIMUT [2], a robot that combines legs, wheels and tracks in a novel design that also allows omnidirectional movement without changing its orientation.

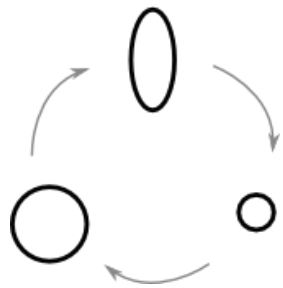


Figure 2.1 Conceptual Depiction of an Intra-Reconfigurable Robot

Sun et al [3]present versatile amphibious robots capable of intra-reconfiguration over three terrestrial and two aquatic gait mechanisms. Yoon et al [4] demonstrate an intra-reconfigurable rehabilitation robot capable of deployment for a wide variety of exercises with minimal human intervention. The developed mechanism allows for desired ankle and foot motions that involve toe and heel rising as well as ankle rotations. An intra-reconfigurable Black Ghost Knife fish was developed by Low et al [5] where an optimized fin-like propulsion mechanism is modeled with a series of connecting linkages that can be reconfigured to derive other fish movements with undulating fins/body.

He et al [11] discuss i4WD4WS, a mobile robot with independent propulsion and individual steering capable of intra-reconfigurability of mobility, sensing and pay load. The work illustrates the ability of the platform to adjust its wheelbase, wheel stance, clearance, stability margin and altitude of center of gravity.

Siddiqi et al [6] propose two frameworks for analyzing intra-reconfigurations in robotic systems. The work details a non-homogeneous Markov model approach for identifying useful configurations and a control theory approach for dynamic analysis of the reconfiguration. The results of the experimental case involving a planetary surface vehicle capable of intra-reconfiguration in response to changing terrain show that reconfigurability improves performance by approximately 30%.

Intra-reconfiguration for manipulation consists of reconfigurable mechanisms that support a variety of functional capabilities. Guowu et al [7] introduce a metamorphic robotic hand capable of intra-reconfigurable palm topologies and demonstrate the prototype for a variety of tasks including grasping, twisting and prehension. An unconventional approach to intra-reconfigurability achieved through passive joints only by changing the Denavit-Hartenberg parameters is presented

in Aghili et al [8]. The proposed prototype has passive joints that are normally locked, and are released only during reconfiguration and controllable while forming a closed kinematic chain.

A component based intra-reconfigurable robot system for manufacturing task is discussed in Chen et al [9]. The proposed robot is assembled using standard and inter-operable components like actuators, connectors and tools with arbitrary geometry and degrees of freedom. The work demonstrates a 'plug and play' robot kinematic and dynamic modeling algorithm for 3D contour following and positioning tasks. Voyles et al [10] present an intra-reconfigurable robot that allows side slipping, holonomic differential drive and serpentine behaviors while capable of switching between limb and tread locomotion.

Intra-reconfiguration for sensing enables a robot to adapt its sensor configuration to the environment/task at hand. An evolutionary approach to intra-reconfigurability of sensor morphologies in response to task/environmental variables is demonstrated in Parker et al [12]. The number of active sensors, the heading angle, range measurements from individual sensors and the control rules are reconfigured within a single robot. The paper demonstrates the efficiency of such an approach where minimal sensory and controller configuration are maintained for a given task/environment using an intra-reconfigurable hexapod robot.

Balakrishnan et al [13] propose evolutionary design techniques for perceptual intra-reconfigurability. The work shows that robots of higher fitness are produced by enabling intra-reconfigurability of sensory placement and ranges in a simulated block environment. Another evolutionary approach to optimal minimum sensory suite is that of Bugajska et al [14]. The work discusses an intra-reconfigurable autonomous micro-air vehicle capable of configuring its sensory and control morphologies for navigation and collision avoidance.

Djath et al [15] have put forward an intra-reconfigurable rule-based strategy to recognize and eliminate corrupted sensory, to improve reliability in a multi-sensor system for a mobile robot. Lichtensteiger et al [16] present intra-reconfigurability of sensory morphologies over varied task experimented with a robot that dynamically reconfigures individual facets of its artificial compound eye.

Intra-reconfiguration for computing allows robots to reconfigure its logic in response to environmental/task demands. FPGA-based intra-reconfigurable computing and electronic

interfacing for an autonomous sailing boat is presented by Alves et al [17]. Such an approach is shown to minimize the energy consumption by reconfiguring the computer control logic circuits in response to navigational parameters such as wind and sea conditions.

Spinka et al [18] study an open-source low-cost intra-reconfigurable autopilot framework for small unmanned aerial vehicles control system. The intra-reconfiguration in this case is found in software; in the event that the primary node of the control system fails, another node can take over the critical vehicle control functionality, at the expense of less important tasks.

A framework for implementing dynamically reconfiguring avionics and computing system for unmanned aerial vehicles is presented by Rawashdeh et al [19]. The framework takes a 'graceful degradation' approach where any hardware/software failure causes quality drop of the control system but does not result in overall system failure.

Kim et al [20] developed a practical framework, SHAGE (Self-Healing, Adaptive and Growing SoftwarE) to realize intra-reconfigurable software in home-based service robots. Experiments validated the ability of SHAGE to enable runtime intra-reconfiguration of software architecture when a service robot witnesses unexpected scenarios or human interactions. Table 2.1summarizes the exemplary approaches to intra-reconfigurability from literature across mobility, sensing and computing functional types.

Table 2.1Exemplary Approaches to Intra-Reconfigurability

Intra-r types	Exemplary approaches
Mobility	Legged to wheeled to tracks; sailing to underwater; hovering to jumping; flying to crawling; finger gripper to driller;
Sensing	Sensory placement; sensory specifications; collection of sensors/type
Computing	Dynamic logic components; Self-repair and graceful degradation

2.2 Inter-Reconfigurable Robots

An inter-reconfigurable robot consists of an assembly of intelligent component robots capable of forming a variety of morphologies through an ongoing assembly and disassembly

process involving a set of robotic modules. Figure 2.1presents a conceptual depiction of inter-reconfigurable robots. Such robots are gaining widespread popularity due to the possibility of assembling a variety of specialized robots using a standard set of components. Numerous inter-reconfigurable robots have been developed for a variety of applications ranging from surveillance to space exploration.

PolyBot is an inter-reconfigurable robot developed at the Palo Alto Research Center by Yim et al. [21]. The component robot modules are cube-shaped and possess one rotational degree of freedom driven by a brushless motor, as well as force, torque, touch, and infrared sensors. Various combinations of such modules have demonstrated locomotion capabilities like walking, rolling, and climbing. The connection mechanism was realized through hermaphroditic connectors and actuated latches.

Programmable parts is a modular inter-reconfigurable robotic platform developed at the University of Washington [22]. The individual component robotic modules move around randomly, get connected, communicate and then decide whether to stay connected or detach. Neighbourhood interactions play a vital role in formulating global configuration and behaviours. The local interactions are governed by chemical master equations and non-equilibrium statistical dynamics.

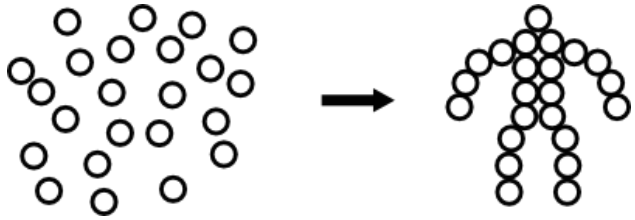


Figure 2.2 Conceptual Depiction of an Inter-Reconfigurable Robot

Another modular inter-reconfigurable robotic platform is Molecube developed by Zykov et al. at Cornell University [23]. Each component robotic cube possesses two degrees of freedom and connects by a link frame. Molecubes were capable of physical self-reproduction with three and four modules. The robot allowed straight-line traversal and 90 degrees convex and concave transitions for self-reconfiguration.

The SuperBot inter-reconfigurable system is developed by Shen et al. at the University of South California [24]. Each module possesses three degrees of freedom and supports both chain and lattice architectures. The dock connectors are available in each of their module sides which

connect to the other modules. These connectors allow power transfer and communication between modules.

Another inter-reconfigurable modular robotic platform is Miche developed by Rus et al. at MIT [25]. Each component robotic module is an autonomous robot capable of self-constructing, deconstructing, and communicating with its neighbouring units. Miche modules have switchable magnets to bind or detach from neighbouring units. Modules organize in an arbitrary form and once a specific form is chosen by a human user, the robot collectively plans its disassembly and the unnecessary units fall off from the main structure due to gravity as the connection gets demagnetized.

Roombots is an inter-reconfigurable concept proposed by Sproewitz et al. at EPFL [26]. This ongoing project aims to develop a modular robotic system for making adaptive furniture by self-reconfiguring its units. Each module is designed with three degrees of freedom and strain-free mechanical latches for self-assembly and disassembly. Roombots uses a centralized approach to reconfiguration planning using graph signature and a graph-edit distance approach.

Gross et al [27] discuss the capabilities of Swarm-bot, an inter-reconfigurable robot that consists of component autonomous mobile robots called s-bots. The individual component robots can either act independently or dock with other s-bots using their grippers to form complex morphologies. They also present successful experimental results involving 16 physical robots and demonstrate self-reconfiguration in response to terrain changes.

An inter-reconfigurable modular humanoid service robot is presented in Taira et al [28]. The work demonstrated a humanoid robot consisting of five functional component robots including as arm robot, a mobile robot and a head robot that are reconfigurable for expandability.

A shape-shifting robot with inter-reconfigurability at its core is demonstrated for urban search and rescue operation in Li et al [29]. A three-module reconfiguring robot consisting of line, triangle and row type symmetry configurations has been built and experimented for its mobility and flexibility.

Another related project in urban search and rescue domain is that of Zhang et al [30]. The inter-reconfigurable robot consists of three identical component mobile robots that assemble to

form new topologies using the active joints formed by the serial and parallel mechanisms of the component robots.

Wolfe et al [31] present a hybrid chain/lattice design inter-reconfigurable robot, M^3 Express. Each component robot module is equipped with a dual-purpose docking mechanism with optical and electrical interfacing is demonstrated that enables active mechanical dock as well act as driven wheels. Experimental results demonstrated the mechanical docking, connector strength and accuracy of dead reckoning locomotion.

An intra-reconfigurable modular robotic system is given in Harada et al [32] for deployment in endoluminal interventions in the gastrointestinal tract. The proposed miniature robotic component modules are ingested and assembled to a variety of configurations in the stomach cavity enabling complicated surgical tasks.

Another related work is Castano et al [33], where CONRO, a metamorphic inter-reconfigurable robot capable of separation of its locomotion and reconfiguration stages is presented. Each component modular robot has two degrees of freedom and uses SMA locking mechanism for inter-robot docking. Table 2.2 summarizes the exemplary approaches to inter-reconfigurability from literature across mobility, sensing and computing functional types.

Table 2.2 Exemplary Approaches to Inter-Reconfigurability

Intra-r types	Exemplary approaches
Architectures	Chain, Lattice, Deterministic, Mobile, Stochastic,
Docking	Magnetic, Mechanical, Electro-mechanical, Chemical
Control	Centralized, De-centralized, Hierarchical

2.3 Nested-Reconfigurable Robots

Reconfigurable robots clearly fall into either inter- or intra- reconfigurable topologies. For many real-world applications however, a nested reconfigurable approach where these are combined would be appropriate. Nested reconfigurable robots offer a new perspective wherein a set of modular robots are capable of reconfiguring their morphologies individually (intra-) as well as combine with a group of other modular robots (inter-) to form complex morphologies suitable to accomplish a given task.

While the distinction is made elsewhere between intra- and inter-reconfigurability (Khoshnevis et al 2001), it has been framed as assembly / disassembly of a set of robots at macro- and micro- scale wherein the individual robotic module maintains its morphology as constant when assembled in an aggregate structure. Our approach to intra-reconfigurability refers to the ability of the component robot at the atomic level to transform its morphology without splitting. The concept is illustrated in Figure 2.3

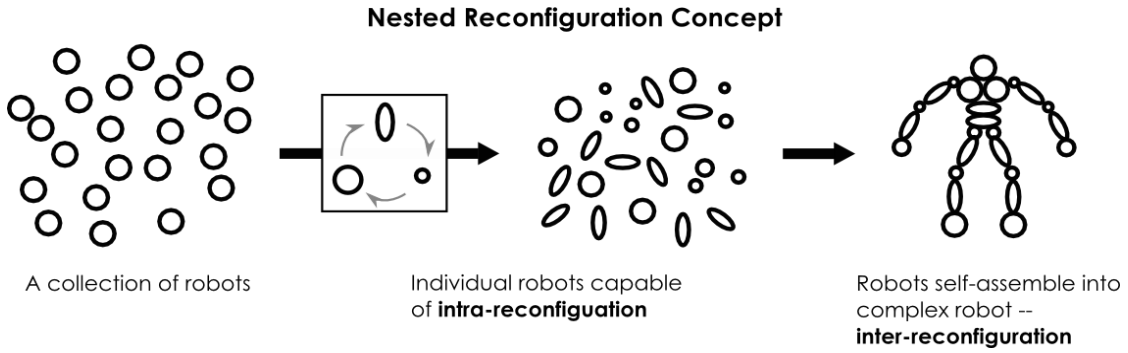


Figure 2.3 Nested Reconfigurability Concept

For example, in a sample return mission, a large nested reconfigurable robot may be deployed to navigate a longer target distance in a short time. While navigating, this robot would be capable of reconfiguring its morphology both in its larger form as well as internally within the individual robots that make up the larger robot to overcome obstacles and terrain variations along the way. Once the target location is reached, this robot may disassemble into smaller robots that can move about collecting samples in the local areas while capable of reconfiguring their internal morphologies individually. Tan et al. described a nested reconfigurable robot named Hinged-Tetro [34] that can transform between different morphologies as a single unit. It can also assemble and disassemble with external robots to create global morphologies. In our earlier work, we suggested and validated a Tetris-inspired intra-reconfigurable floor-cleaning robot, hTetro that can adjust its morphology to any of the seven one-sided tetromino forms as means of maximizing the floor coverage area [35]. We also compared the area coverage performance of this hTetro robot with that of a commercially available fixed morphology robot. The experimental design involved a human user reactively switching the morphology of the hTetro robot in relation to the perceived set of obstacles in the environment as a means of maximizing coverage performance without the involvement of global motion planning.

2.4 Fault Diagnosis in Reconfigurable Robots – Needs analysis

Well-ordered adaptation to fault tolerance has become one of the key tests for the future reconfigurable robotic systems. As mentioned in [36], frameworks and control science must be capable of proposing efficient control techniques to optimize the system's behaviour to accomplish proposed functions and achieve required performance and reduce negative effects such as resource utilization and security conduct. Nevertheless, the expanding multifaceted nature of reconfigurable robotic systems advancing in open, intricate, unstructured and dynamic condition and interfacing with people infers that the acknowledgment of modern missions dependably appears to presently be a challenge. In future, fault tolerance integration will play a vital role in the development of advanced robotic systems. The software development related to the control architecture will be an imperative constituent of the adaptation to the fault tolerance applications.

2.5 Fault Diagnosis in Machine Learning

Since the maintenance has huge effects in industry, it has gotten a profound consideration from the experts and useful support. As indicated in literature, maintanance costs are a noteworthy piece of the aggregate working expenses of all production and manufacturing units, which can represent the deciding moment of a business. Based on the industry, it is predicted that the maintanance costs can be from 15% to 40% of a manufactured product [37]. Nevertheless, it is expected that these costs can increase in the future due to the development and new technologies.

These days, support from computer technology is very important in the development of maintance strategy. Recently, artificial intelligence (AI) method is used as tool for routine manitanance which leads the conventional maintanace program to an intelligent maintance system. These systems comprises of hardware and software parts which makes the maintance routine as performed by human worker. Application of Expert Systems (ES) is one such solution to the maintance routine.

In ES, the whole task-specific knowledge is transferred from the human to computer systems. This expertise is then stored in the computer which ca be used based on the specific needs. As the computers can make accurate and specific conclusions faster, it can provide guidance and if needed they can explain the logic bahind the guidance provided [38].

2.6 Support Vector Machines

A new computational learning technique, Support Vector Machine (SVM) is based on the statistical learning theory, which can serve as ES. Some fundamental papers acquainted underneath with demonstrate the improvement of SVM that initially originated from factual learning hypothesis (SLT) created by Vapnik [39]. SVM, a Vapnik– Chervonenkis theory (VC-hypothesis) based method has emerged as a mathematical structured framework for evaluating (learning) conditions from finite samples. This approach consolidates major ideas and standards connected to learning, formulation, and self-consistent mathematical theory. In addition, calculated structure of VC-hypothesis can be utilized for enhanced understanding different learning strategy created in insights, neural systems, fluffy frameworks, flag preparing, and so forth. A noteworthy applied commitment of VC-hypothesis is returning to the issue proclamation suitable for current learning technique that makes a reasonable refinement between the issue definition and arrangement approach used to take care of the issue.

In light of VC measurement and leave-one-out strategy, the limits on the speculation execution are optimized using training algorithm as proposed in [40] which maximizes the difference between the preparation calculation that consequently augments the edge between the training patterns and the decision boundary. This calculation builds, and after that pursuit the isolating hyper planes with most extreme edge by changing the issue depiction into double space by means of Lagrangian. SVM is accounted for effectively connected in optical character acknowledgment issues with great speculation capacity contrasted and two-layer back-spread neural system. Alternate achievement is likewise revealed in Ref. [41] that SVM is superior to direct classifier, 3-closest neighbor and two-layer neural system in optical character recognition.

Besides, the advancement of SVM solver to solve quadratic issue in SVM demonstrates the seminal in machine learning technique. Sequential minimal optimization (SMO) was developed by Platt [42] that reported more compelling than traditional QP solver. The other created SVM solver will be examined later.

As compared to other traditional methods such as neural network, SVM gained much popularity due to its generalization ability excellence. With this, the applications of SVM became

manifold ranging from face detection and recognition, object detection and recognition, handwritten character and digit recognition, text detection and categorization, speech recognition, information and image retrival. But premitive literature notes is identified in the applications of SVM in machine condition monitoring and diagnosis.

SVM is used for recognizing speech patterns in machine condition checking and fault tolerance tasks which are then categorized based on the fault accurances in the machine. When the signal acquisition is completed, feature representation method can be used to define the features such as statistical features of signal for classification needs. These features can be considered as petterns which are recognized using the SVM classification method.

In feature representation, large amount of features are obtained. As not all these features contain needed information about the machine condition, unwanted features should be excluded in the process to improve the classification's accuracy. With this, feature extraction and selection become necessary to obtain better features for classification routine [43].

Traditional pattern recognition method and artificial neural networks (ANN) are reported that enough samples are available which may not be always possible in real-time. [44]. SVM in view of measurable learning hypothesis that is of specialties for fewer examples has preferable speculation over ANN and assurance the nearby and worldwide ideal arrangement are precisely the same [39]. Interim, SVM can take care of the learning issue with few examples. Because of the way that it is difficult to acquire adequate blame examples by and by, SVM is brought into machines fault diagnosis because of its high accuracy and great speculation for fewer examples.

To minimize the error on training dataset, a set of learning apporaches are designed, called emperical risk minimization (ERM) which follow the ERM principle. Neural networks are one of the ERM based method. In contrary, the SVM method is based on the structural risk minimization (SRM) priniciple, a branch of statistical learning theory. The SRM priniciple prvides better generalization which are achieved through minimization of the upper bound of the generalization error [39–41].

As the training set of SVM is carried out in such a way that the classified vectors do not have distinct influence on the SVM's performance but has the performance of conventional classifier. For this reason, it is identified to be efficient in large classification tasks which will also

benefit in fault classification as the number of features is based on the fault diagnosis that need not to be limited. SVM method is reported to have better generalization properties as compared to the conventional methods. This is because, in SVM training classifier the structural misclassification risk is minimized, whereas in traditional classifiers are trained to minimize the empirical risk. Chritiani and Shawe-Taylor reviewed the performance of SVM in various classification tasks [45].

Given information input x_i (i=1, 2, … , M), M is the quantity of samples. It is assumed that the samples have two classes (positive and negative). Every one of these classes connect with labels be y_i=1 for positive class and y_i=−1 for negative class, respectively. On account of linearly data, it is conceivable to decide the hyperplane $f(x)$=0 that separates the given information

$$f(x)=\mathrm{w}^{\mathrm{T}}{}_{\mathrm{x+b}}=\sum_{\mathrm{j}}{}^{1}=Mw_jx_{j+b}=0, \tag{2.1}$$

where **w** is M-dimensional vector and b is a scalar.

To define the position of separating hyperlane, the vector w and scalar b are used. The choice function is made through sign $f(x)$ to make separating hyperplane that group input data in either positive class or negative class. A particular isolating hyperplane ought to be fulfill the requirements

$$f(x_i)=1_i f y_i=1, f(x_i)=-1_i f y_i=-1 \tag{2.2}$$

Eq. 2 can also be written as,

$$y_i f(x_i)=y_i(w^T x_{i+b})\geqslant 1 \text{for} i=1,2,\ldots,M. \tag{2.3}$$

Optimal separating hyperplane is the separating hyperplane that makes the most extreme separation between the plane and the closest information, i.e., the greatest edge, is known as the ideal isolating hyperplane. A case of the optimal hyperplane of two data sets is illustrated in Figure 2.4

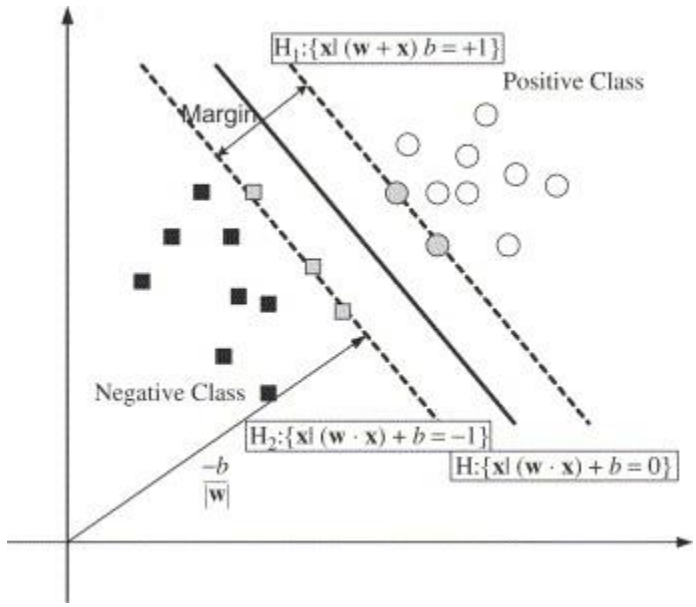


Figure 2.4 Classification of two classes using SVM

In Figure 2.4, arrangement data points for two distinct classes of data are presented, black squares for negative class and white circles for positive class. The SVM tries to put a straight limit between the two unique classes, and orientate it in such way that the edge spoke to by the specked line is amplified. Besides, SVM endeavors to orientate the limit to guarantee that the separation between the limit and the closest information point in each class is maximal. At that point, the limit is put amidst this edge between two focuses. The closest information indicates that utilized characterize the edge are called bolster vectors, spoke to by the dim circles and squares. At the point when the help vectors have been chosen, whatever is left of the list of capabilities isn't required, as the help vectors can contain all the data based need to characterize the classifier. From the geometry the geometrical edge is observed to be $||w||^{-2}$.

Considering the noise with slack variables ξ_i and the error penalty C, the ideal hyperplane separating the data can be acquired as an answer for the accompanying optimization issue:

By minimizing $\frac{1}{2}\left|\left|w\right|\right|^2 + \mathrm{C}\sum_{\mathrm{I}}{}^{1} = M\xi_i$ (2.4)

Subject to $\{y_i(w^T x_{i+b}) \geqslant 1\text{-}\xi_{i,i}=1,\ldots,\mathrm{M},\xi_i \geqslant 0, i=1,\ldots,M,$ (2.5)

where ξ_i is estimating the separation between the margin and the illustrations x_i that lying on the wrong side of the boundary.

By converting the problem with Kuhn–Tucker condition into the equivalent Lagrangian dual problem, the calculation can be further simplified as;

$$\text{minimize } L(w,b,\alpha)=\frac{1}{2}||w||^2-\sum_i^1 =M\alpha_i y_i(w\cdot x_{i+b})+\sum_i^1=M\alpha_i. \quad (2.6)$$

The objective is simplifying Eq. (2.6) regarding w and b, while requiring the subordinates of L to α to vanish. At ideal point, we have the accompanying saddle-point equations:

$$\partial_L\partial_w=0,\partial_L\partial_b=0, \quad (2.7)$$

Eq. (2.7) can be replaced to form

$$w=\sum_i^1=M\alpha_i y_i x_i,\sum_i^1=M\alpha_i y_i=0. \quad (2.8)$$

Through Eq. (2.8), it can be seen that **w** is contained in the subspace spanned by the $\mathbf{x}_i$. Using Eq. (2.8) in Eq. (2.6), dual quadratic optimization problem can be obtained

$$\text{maximize } L(\alpha)=\sum_i^1=M\alpha_i-\frac{1}{2}\sum_{i,j}^0=M\alpha_i\alpha_j y_i y_j x_i\cdot x_j \quad (2.9)$$

$$\text{subject to } \alpha_i\geqslant 0, i=1,\ldots,M,\sum_i^1=M\alpha_i y_i=0. \quad (2.10)$$

Hence, while solving the dual optimization problem, the coefficients α_i can be obtained which is needed to express the **w** to solve Eq. (2.4) that leads to non-linear decision function.

$$f(x)=\text{sign}(\sum_{i,j}^1=M\alpha_i y_i(x_i x_j)+b). \quad (2.11)$$

SVM can likewise be utilized as a part of non-linear order errands with use of kernel functions. The data to be grouped is mapped onto a high-dimensional feature space, where the linear arrangement is conceivable. Through the non-direct vector work $\Phi(x)=(\varphi 1(x),\ldots,\varphi l(x))$ to map the n-dimensional input vector x onto l dimensional element space, the linear decision capacity in dual form is given by

$$f(x)=\text{sign}(\sum_{i,j}^1=M\alpha_i y_i(\Phi^T(x_i)\cdot\Phi(x_j))+b). \quad (2.12)$$

Even though high-dimensional element space working empowers the outflow of complex functions, it also creates the issue. Computational issue happen because of the expansive vectors and the over fitting likewise exists because of the high-dimensionality. The latter issue can be comprehended by utilizing the kernel function. Kernel is a function that profits a dot product of the element space mappings of the first information focuses, expressed as $K(x_i,x_j)=(\Phi^T(x_i)\bullet\Phi(x_j))$. While

applying a part work, the learning in the element space does not require express assessment of Φ and the choice capacity will be

$$f(x)=\text{sign}(\sum^{1}_{i,j}=1M\alpha_i y_i K(x_i,x_j)+b). \quad (2.13)$$

To compute a dot product in feature space, function that fulfills Mercer's hypothesis [39,45] can be used. There are diverse Kernel functions used in SVMs, namely linear, polynomial and Gaussian RBF. The choice of the proper piece work is vital, since the bit characterizes the element space in which the preparation set illustrations will be arranged. The meaning of honest to goodness portion work is given by Mercer's theorem. The function must be constant and positive unequivocal. In this work, linear, polynomial and Gaussian RBF capacities were assessed and illustrated in Table 2. 3.

Table 2.3 kernel functions Formulation

Kernel	$K(\mathbf{x}, \mathbf{x}_j)$
Linear	$\mathbf{x}^T \cdot \mathbf{x}_j$
Polynomial	$(\gamma \mathbf{x}^T \mathbf{x}_j + r)^d$, $\gamma > 0$
Gaussian RBF	$\exp(-\|\mathbf{x}-\mathbf{x}_j\|^2/2\gamma^2)$

2.7 SVM - Multi-class

The above discussions are connected with binary classification in which the class labels can take two values, 1 and −1. In reality issue, in any case, we discover in excess of two classes. For instance, in blame analysis of turning hardware there are a few blame classes, mechanical unbalance, misalignment, bearing deficiencies, and so forth. Consequently, in this segment the multi-class grouping procedure are discussed.

The soonest utilized usage for SVM multi-class grouping is OAA techniques. It develops k SVM models where k is the quantity of classes. The ith SVM is prepared with all of cases in the ith class with positive marks, and the various cases with negative names. Consequently given l preparing information $(x_1, y_1), \ldots, (x_l, y_l)$, where $x_i \in R^n$, $i=1, \ldots, l$ and $y_i \in \{1, \ldots, k\}$ is the class of x_i, the ith SVM take care of the accompanying issue:

minimize: $\frac{1}{2}||w_i||^2+C\sum_i^{11}=\xi_{ji}(w_i)^T$ (2.14)

subject to:$(w_i)^T\varphi(x_j)+b_i\geqslant 1-\xi_{jii} fy=i,$ (2.15)

$(w_i)^T\varphi(x_j)+b_i\leqslant -1+\xi_{jii} fy\neq i,$ (2.16)

$\xi_{ji}\geqslant 0, j=1,\ldots,l,$ (2.17)

where the preparation information x_i is mapped to a higher-dimensional space by work φ and C, is the punishment parameter.

Through Eq. (2.14) $\frac{2}{||w_i||}$ can be maximized, the edge between two gatherings of information. At the point when information isn't distinguishable, there is a penalty term $C\sum_i^{11}=\xi_{i,i}$, which can lessen the quantity of training errors.

OAO method is another major and important method. This approach constructs $\frac{k(k-1)}{2}$ classifiers where each one is trained on the data from two different classes. For training data from the ith and the jth classes, the following binary classification problem is solved:

minimize: $\frac{1}{2}||w_{ij}||^2+C\sum_t\xi_{tij}(w_{ij})^T$ (2.18)

subject to:$(w_{ij})^T\varphi(x_t)+b_{ij}\geqslant 1-\xi_{tiji} fy_t=i,$ (2.19)

$(w_{ij})^T\varphi(x_t)+b_{ij}\leqslant -1+\xi_{tiji} fy_t=j,$ (2.20)

$\xi_{tij}\geqslant 0, j=1,\ldots,l.$ (2.21)

Several methods are practiced for the future testing after all $\frac{k(k-1)}{2}$classifiers are built. After a few tests, the choice is made utilizing the accompanying methodology: if sign $((w_{ij})^T\varphi(x)+b_{ij})$ says x is in the ith class, at that point the vote in favor of the ith class is included by one. Something else, the jth is expanded by one. At that point x is anticipated in the class utilizing the biggest vote. The voting approach depicted above is likewise called as Max Win technique.

In this technique, the training procedure is like OAO system by solving $\frac{k(k-1)}{2}$binary SVM. In any case, in the testing procedure, it utilizes an established paired DAG which has $\frac{k(k-1)}{2}$internal nodes and k takes off. Every node is binary SVM of ith and jth classes. Given test samples x, beginning at the root node, the binary choice function is assessed. At that point it moves to either

left or right contingent upon the yield esteem. The detail clarification of this strategy is proposed to see Refs. [42,46].

SVM performs machine condition observing and finding utilizing its superb capacity in arrangement process. In light of the information vectors that comprise of portrayal of blame in machine, SVM will perceive these examples. Typically, each blame produces unique highlights that can be considered as examples. So the fundamental assignment of SVM here is to perceive and characterize these examples as exact as conceivable identified with blame. SVM is spurred to speak to designs in a high measurement regularly considerably higher than the first element space. With a fitting nonlinear mapping utilizing piece capacity to an adequately high measurement, information from at least two classes can simply be isolated by a hyperplane [47].

Data preprocessing is vital step for a good classification,. A decent data preprocessing will lessen the noise in the data and holds as much data as possible [48]. From spotless information, the highlights (condition pointers) can be figured and considered as examples for blame determination reason. In the arrangement procedure for blame analysis, when the quantity of articles in the preparation set is too little for the quantity of highlight utilized, the greater part of characterization methods can't discover great order limits. This is called revile of dimensionality [47]. Also, by a decent preprocessing, the quantity of highlights per question can be diminished with the end goal that the order issue can be understood well. Blame determination schedule that utilization information spoke to as highlight is called include based symptomatic. As of late, highlight based demonstrative system has been utilized for blame finding of machine. Specialists who utilized highlights technique for blame conclusion are accounted for as takes after: Expense et al. [49] utilized component based system from control range, envelope range, autoregressive displaying, music range and traditional range for disappointment discovery of a little submersible pump. They attempted to locate the best portrayal of information highlights to such an extent that the objective class can best be recognized from the exception class. The help vector information depiction (SVDD) was proposed to achieve their work for finding the littlest circle containing all objective information.

Few other researchers used statistical features based on moments, cumulants and other statistical features of the time data series and spectral of vibration data for fault detection as reported in the literature [50– 56]. Statistical features of time domain and frequency domain for fault

detection in rotating machinery and cavitation of butterfly valve was used by Yang et al. [57– 65]. On account of induction motor, the autors obtained data of vibration and stator current flag. fault diagnosis of turbo-pump rotor using data features was performed by Yuan and Chu [66,67] that are gained from ferquency bands of secondary vibration signal. The frequency of secondary signal separated into 9 groups then the recurrence amplitudes on each band and their normal esteem are ascertained as highlights. Sun et al. [68,69]employed factual highlights, which originate from acoustic emanation motion for wear identification in machine device. They likewise utilized cutting parameter, for example, cutting rate, profundity of cut and sustain rate as extra highlights. Cho et al. [70]carried out device break identification utilizing highlights from cutting powers and power utilization in end processing machine. The other application was accounted for that Han et al. [71] directed problem area recognition in control plant utilizing highlights from information temperature, which are obtained, by thermocouple and infrared sensors. In addition, Ramesh et al. [72] led a forecast of warm blunder in machine devices utilizing highlights from temperature sensors.

Huge dimensionality problem of features is possibly occurred in feature-based diagnosis process, after defining the features. As not all features are useful and optimal for classification process this limitation may not be avoided. The presence of irrelative highlights has a tendency to debase the execution of classifier. One of answers for take care of this issue is performing highlight extraction, which can remove the ideal highlights and at the same time diminish the dimensionality of highlights. Essentially, highlight extraction implies mapping procedure of information from higher measurement into low measurement space. Numerous strategies have been proposed to perform dimensionality lessening utilizing direct and nonlinear procedures as revealed in Ref. [73]. In machine condition observing and blame finding research territory, include extraction utilizing part examination was accounted for as takes after: utilizing straight strategy, main segment investigation (PCA) was directed by Widodo et al. [63,64] and Yuan and Chu [67], utilizing autonomous part investigation (ICA) [63,64]. Besides, nonlinear component extraction utilizing bit PCA and portion ICA was likewise performed by Widodo and Yang [65]. Alternate systems called harsh set hypothesis (RST) was led for separating ideal highlights and lessen measurement of highlights by Xu and Tooth [53,74]. In their examination, RST was utilized to preprocess the information for taking out excess data and diminishing the sample dimension.

Furthermore, experts in the field recommeded using feature selection after the feature set is defined from the original data. Genetic algorithm (GA) and distance evaluation technique (DET) are the methods that addressed the feature seletion process. Jack and Nandi [50], Samanta et al. [51,52], and Li et al. [75] used GA approach in machine fault diagnosis. Success in feature selection using DET was reported by Hu et al. [56] and Yang et al. [57,59].

From previous discussions, it can be noted that SVM utilizing feature-based determination has been broadly utilized as a part of numerous uses of machine condition checking and blame conclusion. A large portion of aftereffects of highlight based strategy were moderately fulfilled by past papers. It implies that component based technique is prescribed strategy when acknowledgment and grouping procedure will be performed.

2.8 Machine condition monitoring and fault diagnosis Ontology

As indicated by Williams et al. [76], through adfaptation from British Standard (BS 3811:1984), condition monitoring is characterized as the persistent or occasional estimation and elucidation of information to demonstrate the state of a thing to decide the requirement for support. Condition monitoring is required for ensure the survival of machine with the goal that early blame can be recognized and analyzed as ahead of schedule as could be expected under the circumstances. The likelihood of failure can't be neglected in the machine, yet early analysis of beginning of failure is helpful to dodge the machine breakdown. At the point when fault occurance exists in machines, it will give a few side effects like unreasonable vibration and clamor, greatly expanded temperature, oil trash, and so on. In this area, the survey of condition observing and blame determination utilizing SVM will be routed to machines, which have side effects prompt disappointment.

2.8.1 Rolling element bearings diagnosis

To measure the machinary vibrations, bearings are the suitable locations. This is because the dynamic loads and forces of machines are applied in bearings which are one of the critical componenent of any machine. Hence, the condition monitoring and fault analysis can indicated the condition of machines. In this section, we review the works of authors who contributed by studying the fault diagnosis of bearings.

Fault detection of roller bearing was directed by Jack and Nandi by utilizing SVM and ANN [50]. The creators have utilized vibration information got from little test fix and reenacted the bearing condition, which has four blames to be specific inward race blame, external race blame, moving component blame and enclosure blame. They characterized and ascertained factual highlights in light of minutes and cumulants and chose the ideal highlights utilizing GA. In the arrangement procedure, they utilized SVM using RBF kernel with steady kernel parameter. In another study, SVM for fault detection of roller bearing was used through vibration motion with clamor [77]. Surprisingly , the authors have not discussed regarding their exceptional strategy during their exploration with the exception of SVM classification routine. But, it is reported SVM has promising application in fault analysis. Further, Samanta et al. [51,52] have enhanced the past strategies in fault detection of bearing byconnecting GA for include choice and seeking legitimate RBF kernel parameters. A few impact conditions, for example, sensor area, flag preprocessing, number of highlights were exhibited to demonstrate the execution of SVM contrasted and ANN. Rojas and Nandi [78,79] have enhanced their past research on bearing flaw determination. They proposed a reasonable plan for quick identification and arrangement of moving component bearing. SMO was actualized for tackling SVM streamlining issue. Zhang et al. [80] proposed probabilistic SVM (ProSVM) for blame analysis of bearing. It was expected to adequately decrease the quantity of tests on the state of keeping the grouping exactness. Sugumuran et al. [55] utilized blame finding of roller bearing utilizing choice tree (DT) and proximal SVM (PSVM). DT was intended to distinguish the best highlights from a given arrangement of tests for order reason. They guaranteed that PSVM has capacity to proficiently order the deficiencies utilizing measurable highlights. As of late, Hu et al. [56] proposed a strategy, which utilized enhanced wavelet bundle change (IWPT) and SVM troupe for blame analysis of moving component bearing. They additionally utilized element determination utilizing DET because of its unwavering quality and straight forwardness.

In this section, we described the development and technical improvement of using SVM in bearing fault diagnosis process. It is predicted that further enhanced and advanced study to use SVM in bearing fault diagnosis process would exist.

2.8.2 Induction motors Diagnosis

Induction engine is an important segment in numerous mechanical procedures, which manages moving and lifting items. Unique consideration is critically required in condition observing of enlistment engines so as to ensure its steady and superior. By applying early blame determination of working acceptance engines, which give early blame condition, little push to beat such blame, can maintain a strategic distance from more genuine conditions.

A method called coupling pairwise SVM for fault classification induction motor was proposed by Pöyhönen et al. [81,82], in which the power gauge thickness (using Welch's strategy) was calculated from coursing streams in parallel branches of engine. SVM, were then prepared to recognize a sound range from broken spectra. In his examination, enlistment engine was contained blames as takes after: broken rotor bars, broken end-ring in rotor confine, shorted loop and shorted hand over stator winding. Fault detection of induction motor by Zhitong et al. using SVM system for identifying broken rotor bars [83] . In their investigation, induction motor was tried different settings, such as with no fault, one broken bar, two broken bars and three broken bars. They utilized stator current to acquire the flag and ascertained the recurrence range for doing issue recognition. Tooth [74] directed a shortcomings analysis framework in view of combination of RST and SVM. He utilized stator current range as information sources. RST can perform highlight extraction and lessening for evacuating excess properties. The states of enlistment engine were wellbeing, broken bar, dynamic erraticism and static unconventionality. The outcome demonstrated that the proposed strategy has great execution in conclusion precision and needs brief time in training.

Widodo and Yang [63– 65] applied fault diagnosis using SVM consolidated by feature extraction through segment examination (PCA, ICA, KPCA and KICA). The factual element in time area and recurrence space from current and vibration flag were figured as highlights portrayal. The proposed technique was intended to identify blame in acceptance engine, for example, broken rotor bars, bowed rotor, bearing shortcoming, rotor unbalance, erraticism and stage unbalance. As of late, they led blame finding of enlistment engine in view of start-up transient current flag. Transient current flag has trademark (likeness) that was hard to recognize among shortcomings. In this manner, they proposed wavelet SVM (W-SVM) for acquire a novel strategy in characterization process. The fundamental thought of W-SVM was developing a part work utilizing wavelet capacity and after that instigating into SVM theory [84,85].

2.8.3 machine tools Diagnosis

In fault detection of machine tool, AI techniques have been recently used extensively. Furthermore, AI is capable of predicting the expected lifetime of machine tools. In this section, SVM for condition monitoring and diagnosis of machine tools is discussed.

A hybrid SVM-Bayesian System (BN) for anticipating the thermal error in machine apparatus as per particular condition was introduced by Ramesh et al. [72] . In this examination, SVM-BN was produced first all to arrange the mistake into bunches relying upon the working condition and after that complete a mapping of the temperature profile with the deliberate blunder. This idea prompt a more summed up expectation display then the regular strategy for straightforwardly mapping blunder and temperature independent of condition. Such model is particularly helpful in a creation domain wherein the machine apparatuses are liable to an assortment of working conditions. The other research was completed by Sun et al. [68,69]who ordered apparatus wear utilizing SVM in light of assembling thought. This exploration was meant to propose another execution assessment work for apparatus condition checking (TCM). To start with, they examined two kinds of assembling misfortune because of misclassification (misfortune caused under expectation and over forecast) at that point both are used to figure comparing weights of the proposed execution assessment work. At that point the normal loss of future misclassification is acquainted with assess the acknowledgment execution of TCM. At long last, an updated SVM approach is executed to do the multi characterization of hardware states. With this approach, an apparatus is supplanted or proceeded in light of the instrument condition alone, as well as the hazard in cost brought about due to underutilized or abused device. In late production, Cho [70] was directed TCM for apparatus breakage recognition utilizing SVM in processing process. SVM was routed to perceive process variations from the norm and start remedial activity amid an assembling procedure. They connected help vector relapse (SVR) for apparatus breakage assurance and guaranteed superior to conventional various variable relapse (MVR) approach .

The literature in implementation of SVM in TCM was presented in this section. Nevertheless, premitive focus was noted that discuss about TCM between the years 1999 and 2006.

2.8.4 Pumps, compressors, valves and turbines

Tax et al. conducted a study on the detection of pump failure using SVDD [49]. The necessity of preprocessing data such as feature extraction and selection was mentioned in this work. Also, the authors assessed numerous feature extraction procedures in an uncommon kind of anomaly discovery issue. The utilization of SVDD was intended to get sign of the multifaceted nature of the ordinary class in informational collections and how well it is required to be recognizable from the anomalous information. Gao et al. [86] connected SVM for blame finding of valve in responding pumps. As preprocessing, the wavelet bundle change (WPT) was utilized to separate element vectors from vibration flag. They mimic 10 states of valve, which must be ordered utilizing SVM. SVM was effective connected for blame analysis of turbo-direct rotor by Yuan and Chu [66]. The first information originated from vibration flag then the element extraction was performed by applying PCA to separate the ideal highlights and decrease the measurement of highlights. Likewise, in view of same information Yuan [67] was additionally completed blame analysis of turbo-pump utilizing SVM with parameter enhancement. In this examination, counterfeit inoculation calculation (AIA) was utilized to advance parameters in SVM.

In another study, condition classification of small reciprocating compressor for refrigerator using SVM was performed by Yang et al. [61]. In this study, wavelet change, and factual technique were utilized to separate remarkable highlights from push clamor and vibration flag. Additionally, cycle strategy was utilized to choose the best possible RBF portion parameters in SVM. In the same way, Yang et al. [62] completed cavitation recognition of butterfly valve by using SVM. The other research using SVM for fault diagnosis of responding compressor was conducted by Ren et al. [54]. This exploration was intended to identify valve blame utilizing vibration flag. Vibration flag was disintegrated utilizing neighborhood wave strategy and information was procured from valve surface.

In turbine detection, Li et al. [75] using SVM for fault diagnosis of steam turbine. Outfit learning in light of GA, in particular direct hereditary troupe (DGE) was performed to accomplish great execution in order. The proposed framework effectively recognized rotor unbalance in steam turbine. Zhang et al. [87] effectively connected fluffy help vector machine (FSVM) for condition observing of pipe gas turbine set in light of vibration flag. FSVM changed isolating hyperplane by

adding fluffy coefficients to each preparation information test so as to demonstrate misfortune contrast of misclassifying preparing information test of various kinds.

2.8.5 HVAC machines

Batur and Chan reported the fault detection of heat exchanger using SVM consolidated by minimum squares parameter distinguishing proof strategy to empower constant observing [88]. In this framework, SVM was routed to identify irregular state of warmth exchanger, for example, air in steam line, discouraged tubes, high condensate stream and low condensate stream. The other research was directed by Han et al. [71] for problem area recognition in control plant evaporator air preheater in light of minimum squares bolster vector machine (LS-SVM). In this framework, separate models of 3 sets of flame status have been fabricated in view of LS-SVM utilizing RBF kernel and polynomial kernel. The hyperparameters of classifiers were blocked by forget one cross approval. Beneficiary working trademark (ROC) bend demonstrated that LS-SVM has great characterization and speculation capacity. Choi et al. [89] did blame determination of chillers machine utilizing SVM in light of factual test, for example, summed up probability proportion (GLR). The framework was subjected to five kinds of issues, incorporating lessening in water stream rates in condenser, evaporator, fouling in condenser and evaporator and refrigerant undercharge.

2.8.6 Other machines

Several other deployment of SVM for machine condition monitoring and fault diagnosis are reported in the literature. For instance, SVM is used for engine knock detection by Rychetsky et al. [90]. In this study, support vector was consolidated by kernel adatron method to give non-linearity, a predisposition and delicate edge. This announced part adatron calculation could be union quick and legitimate for mix with SVM. SVM classifier was routed to order the present thumping condition (3 classes): no-thump, marginal thump and hard thumping. Xu and Wang [53] utilized RST joined with SVM (RST-SVM) for blame identification of diesel motor. Blame determination of diesel motors is a troublesome issue because of the mind boggling structure of the motor and the nearness of multi-energize sources. In this study, finding method was routed to analyze blame conditions, for example, consumption leeway is too little, admission valve freedom is too extensive

and deplete valve freedom is too vast. In addition, incorporating the upsides of RST in successfully manages the vulnerability data and SVM delivered more prominent speculation execution. The analysis of the diesel exhibited that the arrangement can decrease the cost and raises the productivity of conclusion and checked the practicality of building application. Hu et al. [91] created strategy called combination multi-class SVM for blame conclusion of diesel motor. The principle thought of this strategy is joining the data of a few sources at that point builds it as an info space. At that point SVM classifier acknowledged grouping by finding the ideal hyperplanes with a maximal edge. The framework utilized vibration motion from three accelerometers, which are joined on first chamber head, second barrel head and the focal point of the cylinder stroke. Four conditions were reproduced for diagnosing process those are admission valve freedom is too little; consumption valve leeway is too extensive; deplete valve freedom is too substantial and ordinary condition. The result indicated that the proposed approach can notably improve the accuracy of fault diagnosis.

Furthermore, SVM techniques were also used in the fault diagnosis in sheet metal stamping process. In the research performed by Ge et al. [92], strain signal of stamping process was used by the authors that are highly nonlinear, and non-stationary which are typical signal in metal forming process. Two kinds of operation of metal stamping were tested in this experiment namely, single step blanking and multi-step progressive stamping. The conditions for recreating the procedure were no blame, misfeed (work piece isn't lined up with the passes on), slug (the remaining of the position gap is left on the surface of the up and coming work piece), work piece is too thick, work piece is too thin and work piece material is absent.

Gear fault detection was carried out by Samanta in which she used SVM combined with GA [52]. In this study, for feature extraction process, the time-domain vibration signal of a rotating machine with normal and defective gears were processed. The extracted features from original signal were used as inputs to SVM classifier. In this study, GA was conducted in feature selection process and RBF kernel parameters optimization.

The above presented literatures details the deployment of machine learning approaches in fault monitoring and diagnosis. But, no literature reported the application of machine learning for fault diagnosis in reconfigurable robotics that opens several research opportunities which will be studied in this doctoral thesis.

3 Constrained VPH+: a local path planning algorithm for a bio-inspired crawling robot with customized ultrasonic scanning sensor

Robots that has the ability to reconfigure their morphologies with respect to the application scenario go a long way in improving the versatility, fault tolerance, and efficiency for a variety of harsh missions in real world. Some related to reconfigurable robots are [25,95–98].

Many reconfigurable robotic platforms are based on naturally evolving mechanisms like bio-inspired or biomimetic designs. An improvement in the research of reconfigurable robots has been the development of a bio-inspired platform which displays both rolling and crawling abilities, leading to the robotic platform called as BiLBIQ [98]. While, most of these efforts are focused more on mechanical design, almost none of them are focuses on perception or autonomous functionality.

While remote-controlled robot are sufficient for most applications discussed so far, emerging applications in surveillance and security field require a dire need of development in robots possessing a measure of autonomy and intelligence, which also includes basic functionality that relates to mapping and local path planning. Unfortunately, integration of complex reconfigurable design mechanisms in addition to perception introduces a multitude of new research challenges.

Recently, a family of reconfigurable robotic platforms (i.e., Scorpio) with the capability to crawl and roll has been developed. These robots mimic the morphology of a huntsman spider that can alternate its shape between crawling and rolling by reconfiguring their legs.

The development of scorpio focuses more towards the applications related to surveillance and security; thus, the main focus has been the size reduction of the platform, as well as power efficiency improvement of the overall system. Moreover, the changing requirements of the platform have resulted in a fast evolution of five stages of the robotic platform in 2 years. In order to stick to both these constraints, the Arduino Mini Pro 328 has been chosen as the onboard processing unit which allows for rapid prototyping of applications.

Considering the relative computational inexpensiveness in processing the data, as well as to reduce power consumption, a customized ultrasonic sensor is designed for the system. This allows greater control of the sensor to restrict environmental observations to data pertaining to the successful completion of the mission.

Previous efforts in Scorpio focus on the efficient rolling controller development [100, 101] and the formulation and implementation of an intelligent vision-based terrain perception module [102]. This study strives to add the suite of autonomous functions for the Scorpio by development of an efficient obstacle avoidance algorithm in crawling mode.

One of the fancied applications for the Scorpio platform has been the fast movement over unknown, unstructured terrain within a fixed period of time. Using the hardware described above, A wall-following algorithm [103] was first implemented as a proof-of-concept obstacle avoidance algorithm. Despite its ability to move within a room and find an exit, it was not an efficient algorithm, which leads to oscillations in certain scenarios.

The primary assumption of a conventional local path planning algorithms [104–105] is the use of a robotic platform with Ackermann steering model, as well as highly accurate, rapidly scanning range sensors giving out highly dense data. Due to the inability of the Scorpio robot to satisfy the assumptions for both the proprioceptive and the exteroceptive models, design of a local path planning algorithm which is a better fit for the robot is required.

In this study, we present the design and development of Enhanced Vector Field Histogram (VPH+) based a local path planning algorithm and inspired by the inherent constraint due to the choice of sensor and platform. The remainder of the paper is organized as follows: The “Design of the robot” section describes the design specification of the robotic platform. Thereafter, in “Methods” section, we explain the design of the robot, sensor, and algorithm. The “Formulation” section derives the Constrained VPH+ algorithm from the VPH+ algorithm, while “Implementation” section provides a pseudocode for the algorithm, along with a discussion of some issues faced in implementing the algorithm on the Scorpio. The results are described in “Results” section. Finally, “Conclusions and future work” section concludes the paper and suggests avenues of future research.

3.1 Huntsman spider

The inspiration of the Scorpio robot is drawn from Cebrennus rechenbergi, a species of huntsman spider. This species has the ability to both crawl and roll motion, as shown in Figure **Error! Reference source not found.**. The rolling locomotion of the huntsman spider was d

iscovered by Ingo Rechenberg from TU Berlin [101]. The habitat of Cebrennus rechenbergi is the sand dunes of the Erg Chebbi desert in Southern Morocco on the boundary of the Sahara Desert.

While under normal condition, the spider crawls on eight legs similar to other species of spiders, when provoked or threatened by an external stimulus, it can escape by increasing its normal crawling speed using forward or backward flips by using its eight legs simultaneously, which is similar to acrobatic flic-flac movements used by gymnasts.

Most remarkably, the spider turns somersaults to move independent of surrounding conditions. Therefore, a slope is not required to initiate the rolling process using the gravitational force. It also does not require a run-up or a start-up gesture to trigger the rolling locomotion.

The primary reason for a deploying rolling mechanism in huntsman spiders as observed so far seems to increase the displacement of the spider to escape from threatening situations, or when meeting a conspecific. So far, the spider has not been observed to use the rolling mode for tasks such as changing positions and hunting for prey.

3.2 Local path planning

The occupancy grid map which is an environment representation technique arrived in [105] is still used as a map representation technique in state-of-the art algorithms. Elfes [106] first used the occupancy grids for navigation and mobile robot perception.

A local path planning algorithm known as Virtual Force Field (VFF) was developed by Borenstein and Koren [107] using occupancy grids for obstacle representation and potential fields [116] for navigation. The VFF algorithm employed the use of repelling force fields around obstacles and an attracting force in the direction of the target. The simplicity of the formulation made it an attractive path planning algorithm in the robotics research community.

To respond to this problem, the article [108] first introduced the Vector Field Histogram (VFH) and further expanded and analyzed in [109, 110]. A two-dimensional Cartesian grid is used in VFH as a representative world model, while local environmental data are represented as a one-dimensional polar histogram around the robot. The obstacle density in that direction represents each section of the polar histogram. A direction which contains the best trade-off between the obstacle

density and the goal direction is chosen by the algorithm. A subsequent step provided appropriate steering commands to generate motion in the desired direction.

Local path planning and obstacle avoidance have been the focus of a maximum portion of research in the robotics research community [104–105]. Initial papers focused on development of techniques for perception and mapping using noisy sonar sensors [104–105].

Mathematical and practical drawbacks of the VFF were discovered and analyzed in [108]. Although the inexpensive way to generate an approximate representation of the map is the occupancy grid method, it was unable to compensate for the contradiction between the complexity and roughness of the grids, thereby making it unsuitable for use with a low precision sensor like an ultrasonic range sensor. A character of the algorithm to get stuck in local minima was also discovered in scenarios where the goal was behind an obstacle.

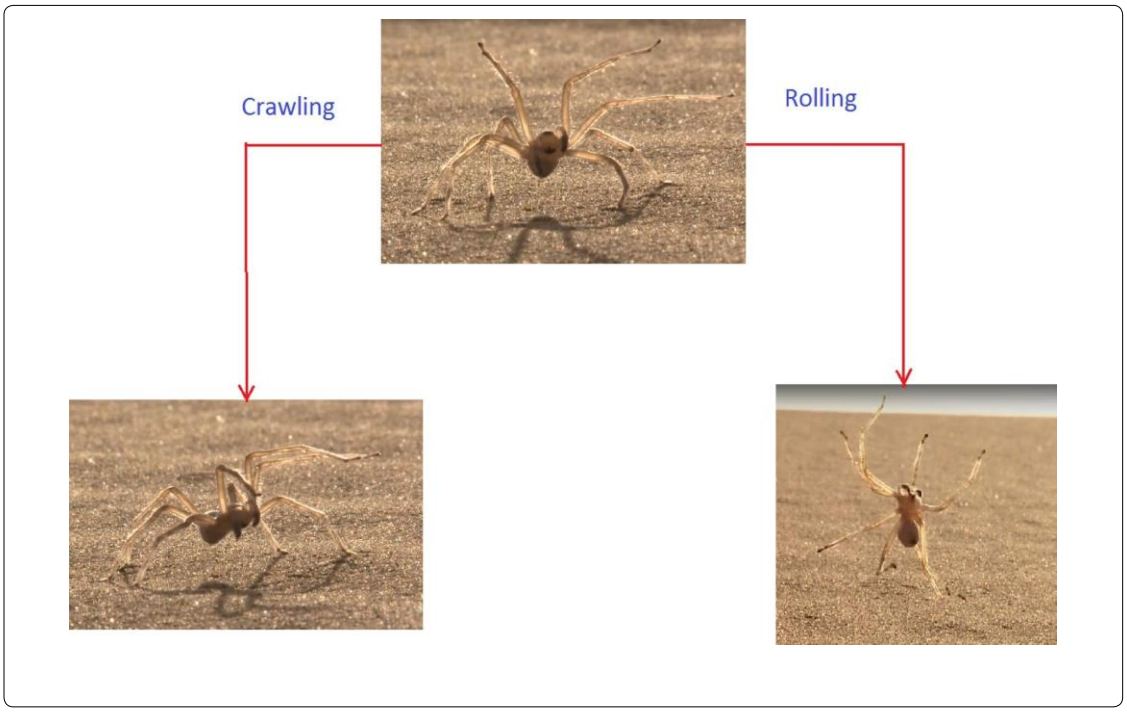


Figure 3.1 The huntsman spider. The huntsman spider of Southern Morocco (top) performing crawling motion (bottom left) and rolling motion (bottom right)

Extensive testing and implementation of the VFH leads to the discovery that the VFH algorithm does not take into account the vehicle kinematics, resulting in unrealistic and impractical inputs provided to the robot. In response, A algorithm specifically designed to account for the

limited velocities and accelerations inherent in wheeled mobile robots called as Dynamic Window Planning approach was postulated in [111].

An improvement in VFH algorithm leads to VFH+ [112]. The VFH+ algorithm includes two additional stages to the VFH algorithm. The tendency of creating oscillations in environments with multiple narrow openings due to the sharp thresholding near the entrances was displayed by the VFH algorithm. The initial stage in the VFH+ algorithm was used to provide a hysteresis function between two threshold values, to reduce the oscillations and which indicates the obstacle density to consider only directions possible for the robot to travel in. The robot was assumed to be a wheeled robot with Ackermann steering mechanism, and hence, the robot trajectory was assumed to be based on series of circular arcs.

The A* search algorithm is used in VFH* algorithm [113] to add look-ahead capability to the VFH+ algorithm to reduce problems arising from purely local obstacle avoidance.

An and Wang [114] developed the Vector Polar Histogram (VPH) algorithm by amalgamating the VFH+ algorithm and the Potential Field Method. The VPH algorithm relies more on the newly available laser range scanners with comparably higher accuracy to accurately represent the local obstacle map.

The grouping individual obstacles to obstacle blocks and determining concave blocks ahead of time extends the VPH algorithm to VPH+ algorithm, in order to increase the efficiency of the robot in traversing the environment. The VPH+ algorithm also extended the VPH cost function to take the robot heading and speed into account.

3.3 Design of the robot

The section presents the mechanical design and system architecture of the Scorpio robot.

Mechanical design

A huntsman spider is used for the design of the introduced in the previous section, which enables it to perform crawling and rolling locomotion's. While there are eight legs in huntsman spider, the Scorpio robot is designed with four legs which are enough to perform crawling and rolling locomotion's.

Figure 3.2. It is observed that the Scorpio robot consists of four legs (tibia), four servo covers and joints (femur), four main joints (coxa), and a body. The processor, controller, and sensors are present inside the body made up of PLA plastic. The locomotion is achieved using twelve servo motors. Each leg (model shown in Figure.

Figure 3.3 Scorpio leg. The model of an individual Scorpio leg is mounted with three servos, thereby having 3 degrees of freedom. These legs are able to rotate and transform from crawling to rolling gaits. The specifications of the Scorpio robot are listed in Table 3.1Table 3.1Specifications of the Scorpio robot [117]

For crawling motion, the Scorpio robot opens up its four legs as shown in Figure 3.4a. The crawling involves 2 degrees of freedom. Transforming from crawling pose to cylindrical exoskeleton for rolling requires a 3 degrees of freedom motion. The center of gravity in a Scorpio robot is shifted using its legs to push from the ground to achieve the rolling motion with 1 degree of freedom. The rolling speed of the Scorpio robot doubles the rate of crawling speed.

Sensor design

The team aimed to create a customized range sensor with a greater degree of control. To this end, an ultrasonic sensor (model number SRF01) was mounted on a HS35HD Ultra Nano Servo Motor. The ultrasonic sensor has a beam width of 12°. Further details of the sensor are tabulated in

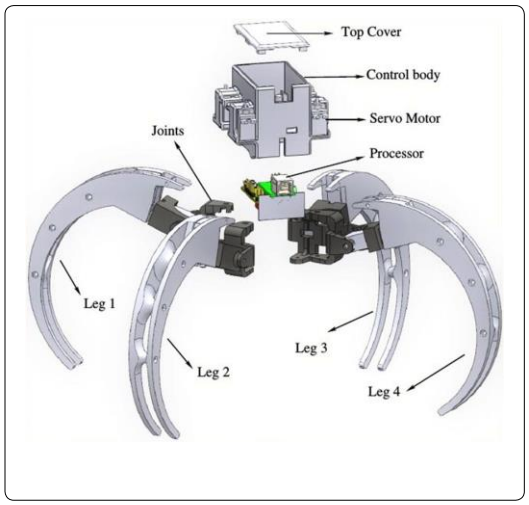


Figure 3.2 Exploded view of the Scorpio. An exploded view of the Scorpio model displaying all the components of the Scorpio

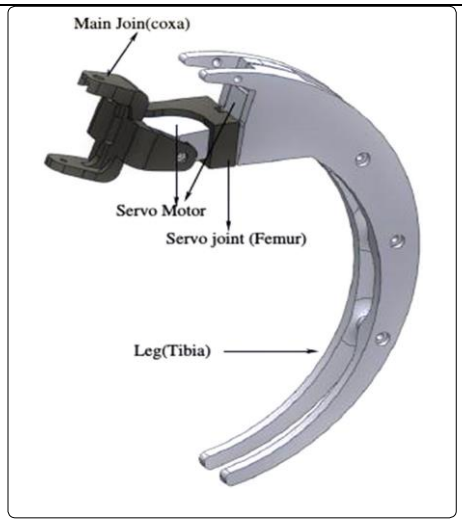


Figure 3.3 Scorpio leg. The model of an individual Scorpio leg

Table 3.2. A picture of the sensor mounted on the Scorpio can be found in Figure 3.5.

The ultrasonic sensor is triggered after the servo motor is actuated to the desired angle. The time taken for the sensor to rotate to the desired angle, transmit, and receive the data is roughly 0.15 s per point. Due to the sensing speed, Scanning of the entire environment at each instance is not recommended. Figure Figure 3.6 displays a sample scan taken with the ultrasonic sensor.

Mode of locomotion

The Scorpio platform demonstrates different modes of locomotion in the two morphologies, each with their respective strengths and weaknesses.

Table 3.1Specifications of the Scorpio robot [117]

Controller	Arduino Mini Pro 328
Servo motor	JR ES 376
Servo controller	Pololu Micro Maestro 18 channel USB servo controller
Battery	LiPo 1200 mAh 7.4 v
ZigBee	XBee Pro S1, Digi International
Full body material	Polylactic acid (PLA)
LXWXH (walking) in mm	230 mm × 230 mm × 215 mm
Weight (full weight) in g	430 g

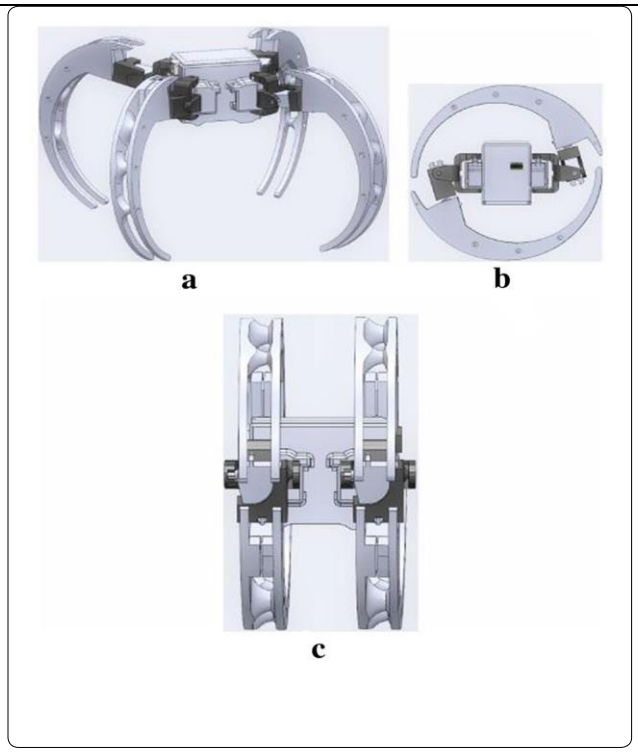


Figure 3.4 Scorpio robot model in different configurations. a Crawling configuration, b rolling configuration side view, c rolling front view

- The rolling mode allows for greater speed of locomotion with reduced environmental perception.
- The crawling mode conversely trades better environmental perception for reduced locomotion.

This study uses only the crawling mode of locomotion, to improve perception of environmental obstacles. While the limited field of view in rolling mode limits the potential for integration into the robot motion planner, future improvements in terrain traversal efficiency requires the use of rolling mode to be taken into account.

Table 3.2 Specifications of the custom ultrasonic sensor

Sensor	SRF01
Beam width	10°
Minimum range Maximum range	 6 m
Time period per data point	0.15 s
Motor	HS-35HD Ultra Nano Servo

Size in mm	18.6 × 7.6 × 15.5
Weight (g)	4.5
Torque (kg/cm)	0.8
Speed	0.10

Algorithm design

The local path planning algorithm design is informed by the following constraints inherent in the Scorpio robotic platform:

1. Kinematic constraints: Since crawling is the mode of locomotion in the Scorpio, a standard wheeled robot kinematics are not valid. Accordingly, all recent standard local path planning algorithms with assumption of Ackermann steering model cannot be used [107–115].
2. Sensor inaccuracy constraints: The customized ultrasonic sensor has a beam width of 15°, making it more inaccurate as the range increases. Thus, same reading is obtained for any obstacle in the sonar beam at the same range (as seen in Figure 3.7). Hence, algorithms which rely on the accuracy of environmental information provided by the exteroceptive sensor [114, 115] will perform poorly on the Scorpio.
3. Sensor speed constraints: Slow scanning speed of the sensor implies performing scans over the full angular range frequently will negatively impact the efficiency of the algorithm. As a result, all recent local path planning algorithms which uses the full sensor range [108–115] cannot be used.
4. Complex kinematic model: The Scorpio platform consists of twelve joint motors, resulting in a complex kinematic motion model. The speed and efficiency of the robot is greatly decreased due to Computation of the entire kinematic model for locomotion on the Arduino processor, making it invalid all recent algorithms requiring computation of exact vehicle kinematics [111–115].
5. Processor constraints: The Arduino Mini Pro processor is single-threaded; thus, each separate action has to be performed serially. As a result, the platform has an inability to scan the environment and move simultaneously leading to the invalidation of all algorithms in which robot velocity is an important determinant of direction of motion [110–115].

Thus, to summarize, the inferences on the local obstacle distribution using sparse, inaccurate data obtained at a low frequency can be obtained from optimal local path planning algorithm. Moreover, due to inefficiency of the mechanism while turning, it is required to ensure the robot

motion is as straight as possible, with minimum turning. The aim of the algorithm is to make sure the robot travels the maximum distance from the starting point in a fixed period of time.

The most recent state-of-the-art local path planning algorithms in the robotics research community are the Enhanced Vector Polar Histogram (VPH+) [115]. The VPH+ algorithm improves the VPH algorithm formulation with obstacle grouping and classification, along with inclusion of the robot velocity and heading into the cost function. Because the sensors used by the Scorpio are too inaccurate to group obstacles accurately, the single threaded onboard processor guarantees the platform will not be in motion when the algorithm is being computed, the VPH+ algorithm will be modified with suitable constraints to better fit the Scorpio platform.

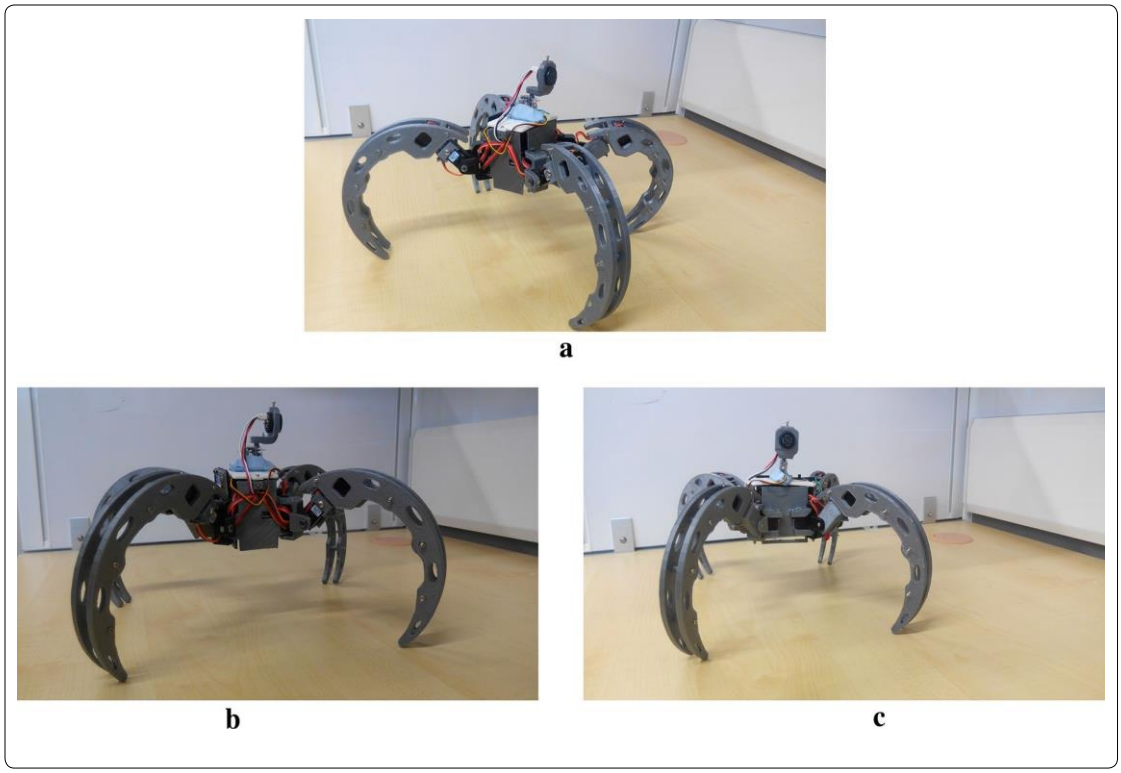


Figure 3.5 Scorpio with customized ultrasonic sensor. a Perspective view, b side view, c front view

Figure 3.6 Sample ultrasonic scan. Sample scan with ultrasonic sensor

The approximation of the Scorpio kinematic model is done using model primitives, to ensure acceptable performance on the Arduino Mini Pro processor.

3.4 Formulation

The VPH+ algorithm represents the local environment as a polar histogram. A cost function is generated for each sensor angle using inputs from both the range sensor and the robot kinematic model. It leverages on the accuracy and the range of the laser range sensor to determine the boundaries of obstacle blocks, as well as to classify the obstacle as concave or convex. Concave obstacles are avoided outright, with the histogram value at the corresponding angles being set to zero.

VPH+ algorithm

Figure 3.7 depicts the diagram used to determine the reachable distance in each direction. The VPH+ algorithm modifies each range reading with the radius of the robot. Thus, in the diagram, the maximum traversable distance by the robot in direction O_i due to obstacle O_j is d'_{ij}, the length of P_rM given by:

$$d'_{ij} = \begin{cases} d_i & s_{ij} > R \\ d_j \cos y_{ij} & \text{otherwise} \end{cases} \tag{3.1}$$

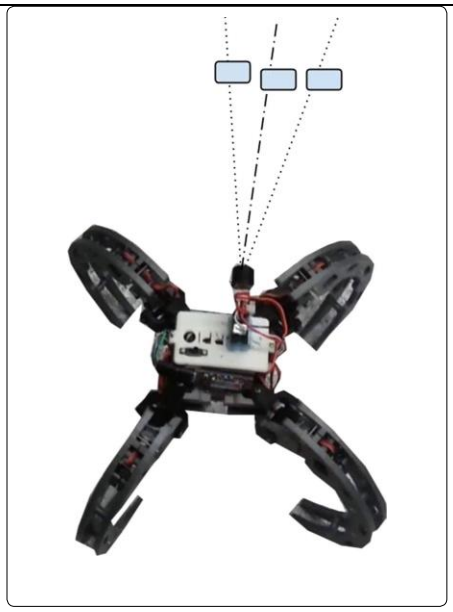


Figure 3.7 Inaccuracy due to beam width. All three obstacles at the same range will produce the same reading due to the large beam width

where y_{ij} is the angle between points i and j, $s_{ij} = d_j \sin y_{ij}$, R is the robot radius.

The overall maximum traversable distance in direction O_i is given by

$$D_i = \min(d'_{ij}) - R; \quad (j = 0, 1, ..., n - 1) \tag{3.2}$$

where n is the total number of beams per scan given by π/α_{BW} for a sensor with beam width α_{BW}.

If the distance between them is lesser than a user-defined safe threshold dthr, which is larger than the robot radius, plus a buffer distance, the observed points are then grouped as obstacles. The distance between two adjacent points is given by:

$$d_{i,i+1} = \sqrt{d_i^2 + d_{i+1}^2 - 2d_i d_{i+1} \cos\alpha_{BW}} \tag{3.3}$$

The VPH+ classifies obstacles as concave or convex, depending on which the following symbol function is constructed:

$$B(i) = \begin{cases} 0 & i \in \text{concave obstacle block} \\ 1 & \text{othererwise} \end{cases} \tag{3.4}$$

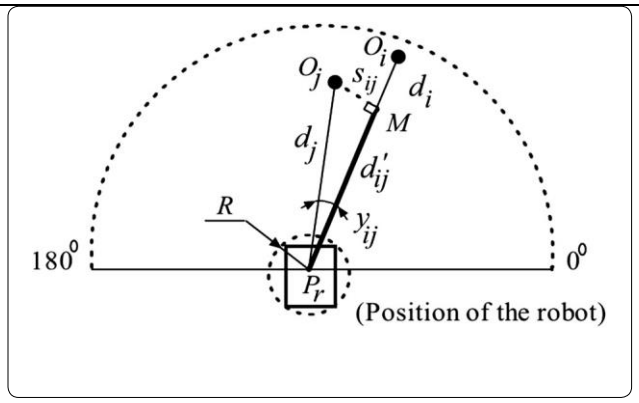


Figure 3.8 Grouping of obstacle blocks in VPH+. Adjacent points closer than a pre-defined threshold are classified as a single obstacle

The robot kinematics are then considered with the current robot velocity, minimum turning radius, and maximum robot velocity being used to generate a value for the safe distance d_{safe}. A threshold function *H(i)* similar to *B(i)* is then created such that

$$H(i) = \begin{cases} 1 & D_i \geq d_{safe} \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

An angular cost function is constructed considering the angles shown in Figure 3.9 given by:

$$S(i) = k_1 h_g + k_2 h_0 + k_3 \tag{3.6}$$

where h_g is the angle between the goal and the current sensor angle, h_0 is the angle between the current direction of the robot and the current sensor angle, k_1, k_2, and k_3 are user-defined constants, $k_1 h_g$ is the cost associated with deviating from the direction to the goal, $k_2 h_0$ is the cost associated with deviating from the current direction of motion, and k_3 is a nonzero constant used to ensure the denominator is nonzero. Thus, a robot with high k_1 and low k_2 will prioritize goal following at the cost of maintaining a smooth trajectory.

The final cost function is calculated with the individual cost functions with the expression:

$$C(i) = \frac{B(i)H(i)Di}{S(i)} \tag{3.7}$$

The desired direction is the direction with the maximum cost function.

$$\theta_{final} = \max(C(i)) \tag{3.8}$$

Effect of sensor inaccuracy

The design of the Constrained Enhanced Vector Polar Histogram (CVPH+) is informed by the constraints from the sonar sensor inaccuracy, the small size of the platform, as well as the low processing power of the

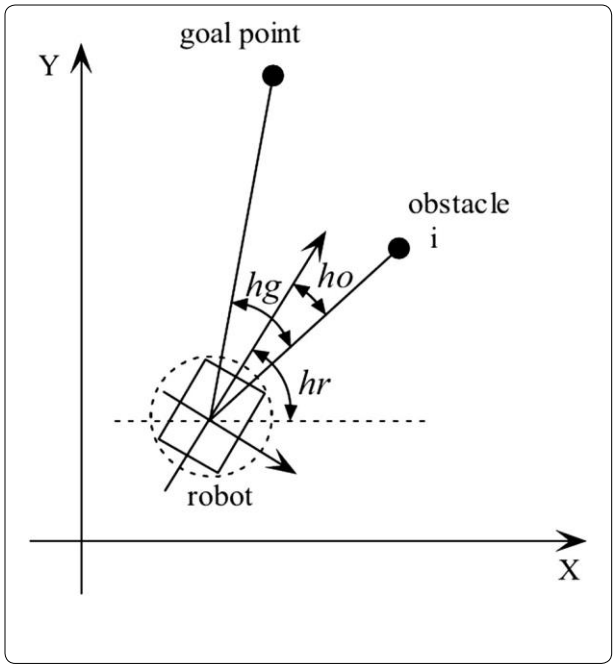


Figure 3.9 Calculation of cost function in VPH+. Angles used for the calculation of each cost function

onboard processing unit. The field of view of the ultrasonic sensor can be divided into *n* sectors, as shown in Figure 3.10, where *n* is given by:

$$n = \frac{\alpha_{FOV}}{\alpha_{BW}} \tag{3.9}$$

where α_{FOV} is the total field of view of the sensor and α_{BW} is the beam width of the sensor, as defined earlier.

Inaccuracies arising due to the beam width of the sonar are of the order of the robot size. For instance, considering the scenario in Figure 3.11, the lateral beam width becomes equal to the diameter of the robot (d_R, say) at the acceptable distance d_{acc}. The value for dacc can be found from the expression given by:

$$d_{acc} = \frac{d_R}{\alpha_{BW}} \tag{3.10}$$

Consequently, obstacle pings with ranges greater than dacc cannot be used directly in VPH+ for obstacle grouping and to obtain obstacle boundaries. Further accuracy regarding obstacle spacing becomes unnecessary when the range reading from obstacles is less than d_{acc} (implying that the distance between two obstacles d_{GAP} is less than the size of the robot d_R); hence, the ranges can be used directly to obtain further inferences regarding the obstacle field.

In the case of [115], the LMS200 range sensor with an angular beam width of 4.4 millirad [118] and a maximum range of 80 m. Thus, the diameter of the robot must be greater than 0.352 m, according

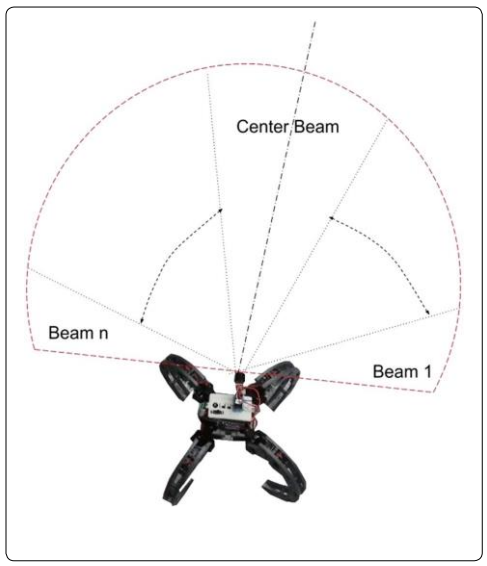


Figure 3.10 Sensor field of view. Field of view of the custom-made ultrasonic sensor used

to Eq. 3.10. The Pioneer P3AT robot used by them has a width of 0.381 m, which fulfills this criterion.

Cost function reformulation

In the cost function for the VPH+ algorithm, the B(i) and H(i) terms are indicator terms having values of only 1 or 0. Consider the modified cost function

$$C'(i) = \frac{k_1 h_g + k_2 h_0}{D_i} \tag{3.11}$$

Assuming the same values for B(i) and H(i) in the VPH+ formulation, in this case, the desired direction at the end of the inference cycle will be given by:

$$\theta_{\text{final}} = \min(C'(i)) \tag{3.12}$$

Let the absolute angle for the goal as seen in Figure 3.9 be α_G and the absolute angle for the point i be α_i. Then, from Figure 3.9, the following relations are true:

$$h_g = \alpha_G - \alpha_i \tag{3.13}$$

$$h_0 = h_r - \alpha_i \tag{3.14}$$

The cost function then becomes:

$$C'(i) = \frac{k_1(\alpha_G - \alpha_i) + k_2(h_r - \alpha_i)}{D_i} = \frac{K_\alpha - K_k \alpha_i}{D_i} \tag{3.15}$$

where Kα is a constant for the current inference cycle given by:

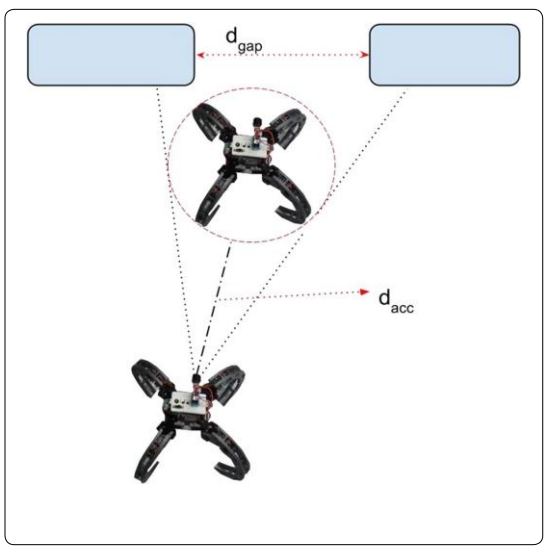


Figure 3.11 Sensor inaccuracy. Constraints to the algorithm due to inaccuracy of the sensor

$$K_\alpha = k_1 \alpha_G + k_2 h_r \tag{3.16}$$

and Kk is a constant for the algorithm obtained from the following expression:

$$K_k = k_1 + k_2 \tag{3.17}$$

If cost function $C'(j)$ will be lesser than $C'(i)$, another angle, α_j, will be chosen. However, the maximum range of the sensor r_{max} tells us that an upper limit to the angle can be determined and is given by the expression:

$$\mathrm{A}_{j,max} \leq \frac{K_k}{K_\alpha} - \frac{r_{\max}}{D_i K_\alpha}(K_\alpha - K_k \alpha_i) \tag{3.18}$$

From observation, We can see that the numerator of the cost function is minimum for scan angles that directly point toward the goal or toward the robot heading, depending on the values of k_1 and k_2.

Hence, the maximum scanning angle for the current scan needs to take both into account, along with the maximum possible angle reachable by the sensor $\alpha_{\text{sensor,max}}$.

$$\alpha_{\max} = \max(\alpha_{G,\max}, \alpha_{0,\max}, \alpha_{\text{sensor,max}}) \tag{3.19}$$

The resulting number of beams required is then obtained by:

$$n_s = \frac{\alpha_{\max}}{\alpha_{\mathrm{BW}}} \tag{3.20}$$

A simple example would be when the robot is pointed toward the goal (i.e., $\alpha_G = h_r$, and there is no obstacle in front). In this case, it is expected that the cost function will beat it's least in the direction of the goal; therefore, the goal direction will have the least cost function. Sure enough, on inputting the values in the above equation, we receive

$$\alpha_{j,\max} \leq \alpha_G \tag{3.21}$$

Since α_G is the first angle sector, the relation simplifies to $\alpha_{j,\max}$ being equated to α_G, the goal angle.

Figure 3.12 shows the numbering convention used for the sonar data. The sonar scan is portioned into an odd number of scans n. The center beam is numbered 0, the beams on the left are numbered negative, while the beams on the right are numbered positive.

3.5 Implementation

Gait primitives for the Scorpio robot were initially obtained. The motion of the Scorpio robot characterized into three different types: left, right, and forward. Histograms of the gaits were obtained, as shown in Figure 3.13, Figure 3.14and Figure 3.15.

The inexact nature of the 3D printing process introduces the structural inaccuracies resulted in different values for left and right gait primitives. The forward motion gait was also discovered to display a drift, as shown in Figure 3.16, which was compensated for in the implementation of the algorithm.

The values of k_1 and k_2 were obtained from by optimizing the implementation of the VPH+ algorithm for the best result, as tabulated in Table3.3.

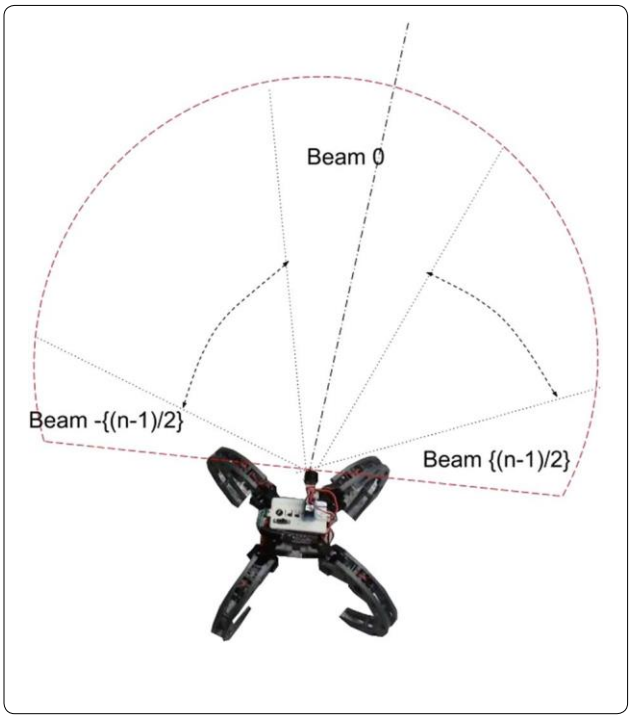


Figure 3.12 Sonar sector numbering. Numbering convention used for sonar data

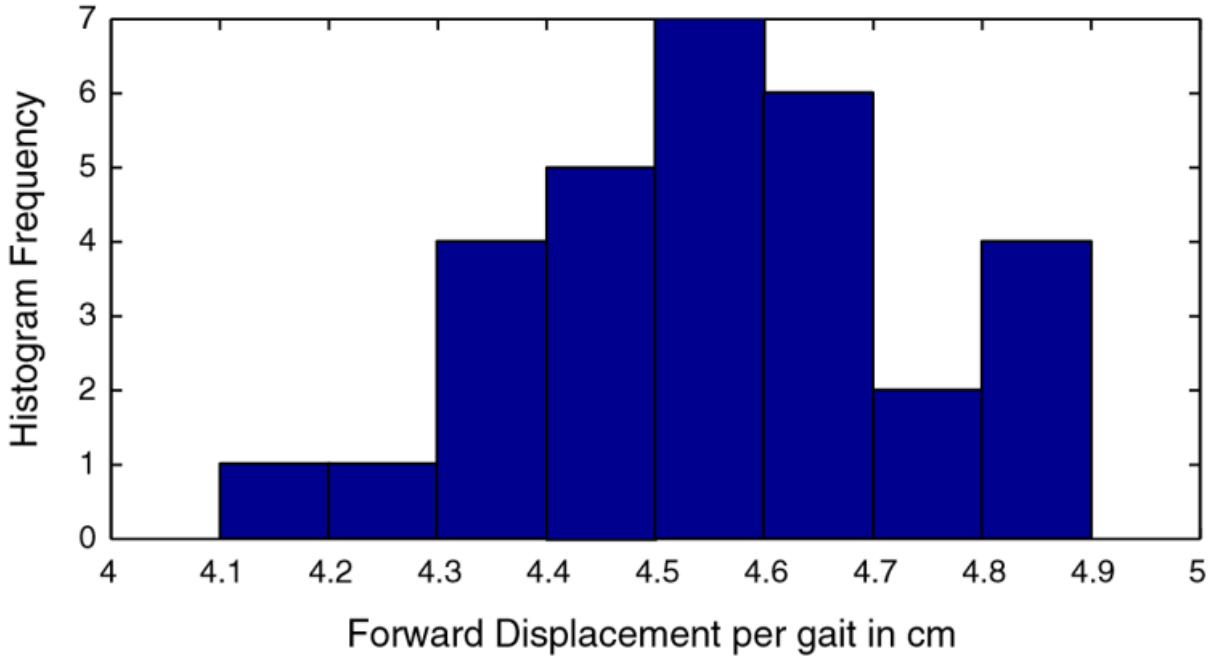


Figure 3.13 Forward motion gait primitive. Histogram for forward motion

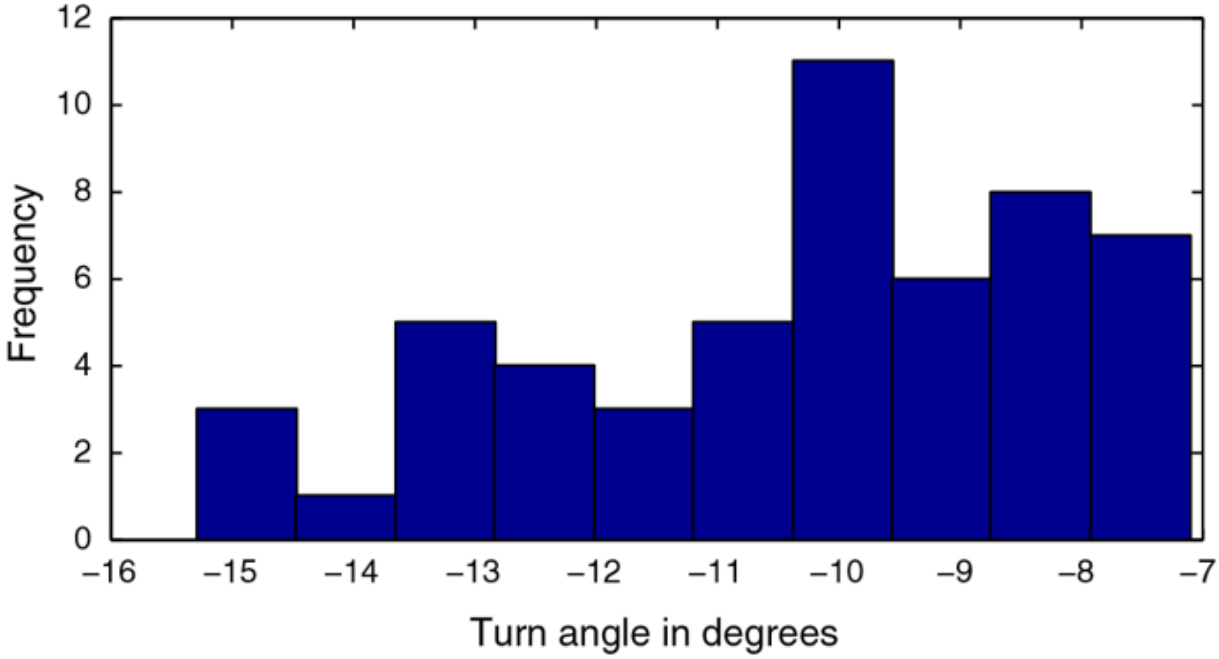


Figure 3.14 Leftward motion gait primitive. Histogram for leftward motion gain primitive

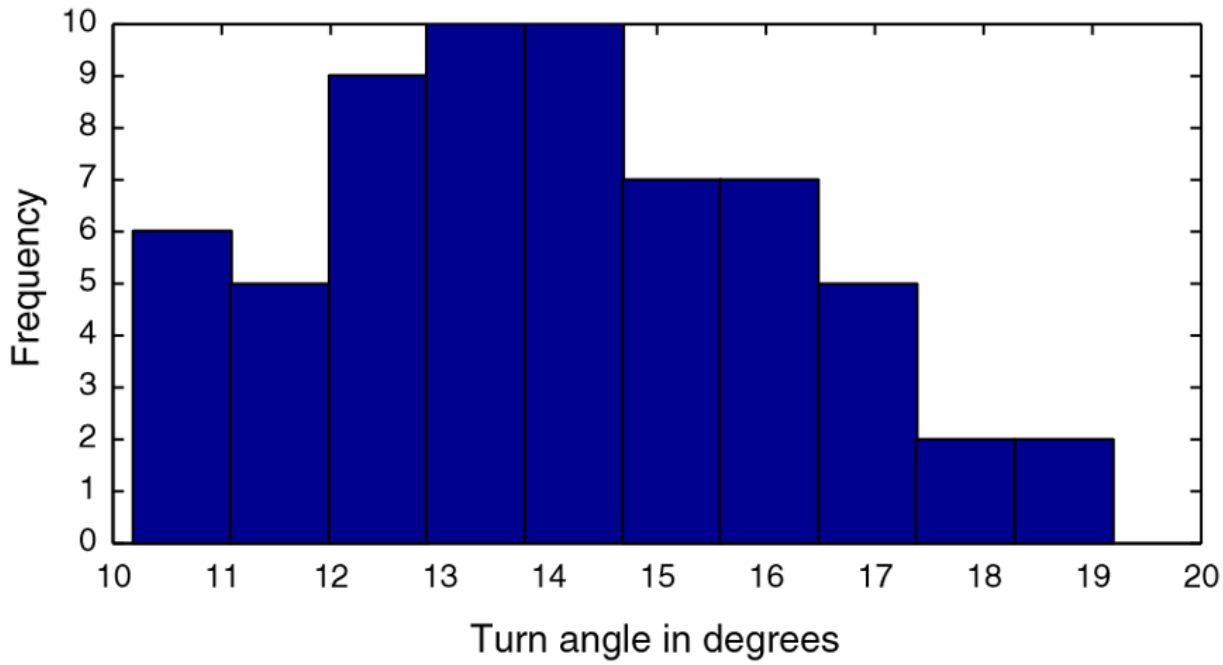


Figure 3.15 Rightward motion gait primitive. Histogram for rightward motion gain primitive

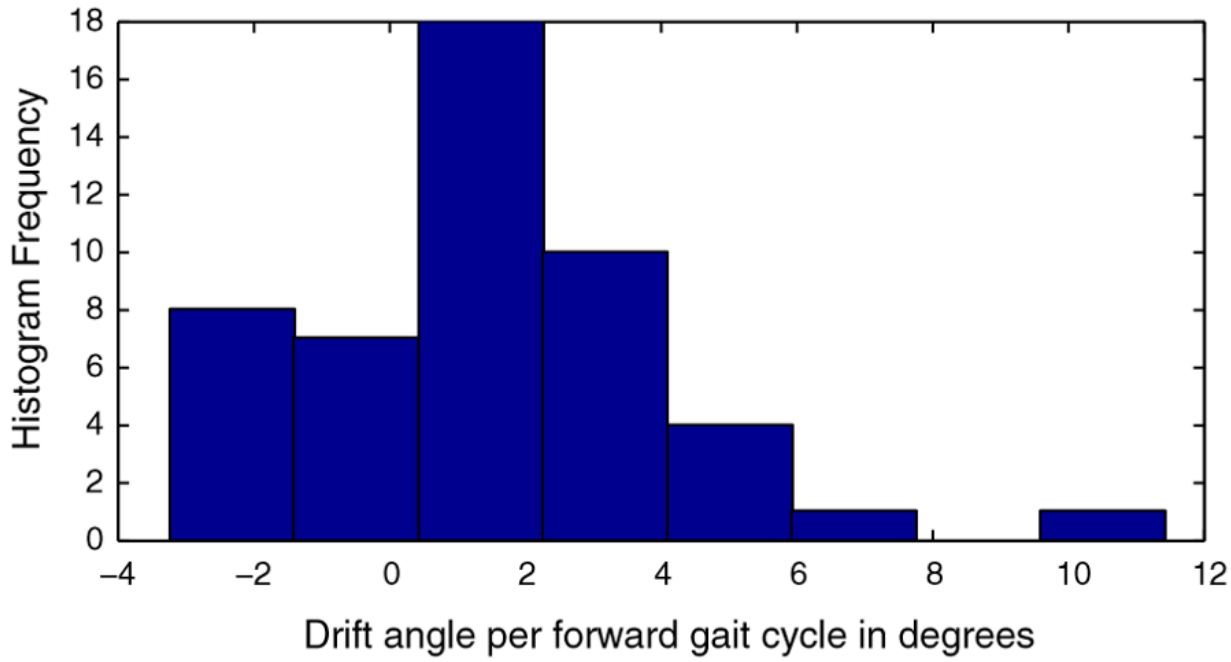


Figure 3.16 Forward motion drift. Histogram for forward motion drift

The pseudocode of the Constrained VPH+ algorithm is displayed below.

Pseudocode

loop

While robot path is not completed Scan and obtain ranges d_g and d_0 in direction towards goal, and towards robot heading. Calculate number of beams n_s

Calculate constants K_k and K_α

for i = 1 **to** n_s **do**
Obtain range scan di
if $d_i > d_{acc}$ **then**
Final distance $D_i = d_i - d_R$
else
Append d_i to unprocessed ranges array R
end if

end for

Calculate maximum reachable distance in each direction for n_R unprocessed ranges in array R

for i = 1 **to** n_R **do**

for j = 1 **to** n_R **do**
Perpendicular dist $s_{i,j} = d_j sin(\alpha_{BW}|i - j|)$
if ($s_{i,j} > d_R$) **or**(($s_{i,j} <= d_R$) and ($d_j \cos(\alpha_{BW}|i - j|) > d_i$))
then $d'_{i,j} = d_i$
else
$d'_{i,j} = d_i cos(\alpha_{BW}|i - j|)$
end if
end for
$D_i = min(d'_{i,j}) - d_R$
end for
Group Obstacles to Block
for i = 1 **to** ($n_R - 1$) **do**

$d_{i,i+1} = \sqrt{d_i^2 + d_{i+1}^2 - 2d_i d_{i+1} cos(\alpha_{BW})}$
if $d_{i,i+1} >= d_{acc}$ **then**
$n_{block} = n_{block} + 1$
$block_i = n_{block}$
end if
end for
Construct Symbol Function
for i = 2 **to** ($n_R - 1$) **do**
Let $n_{Start}(block_i)$ be starting point of $block_i$

Let $n_{End}(block_i)$ be end point of $block_i$
if $d_{nStart}(block_i) < d_{nEnd}(block_{i-1})$ and $d_{nStart}(block_{i+1}) > d_{nEnd}(block_i)$ **then**
  $block_i$ is Concave ⇒ $B_i = 0$
else
  $block_i$ is not Concave ⇒ $B_i = 1$
end if
end for
For all other points which are not part of the unprocessed ranges array (ie, scan range $> d_{acc}$), $B_i = 1$
Construction of Cost Function
for $i = 1$ **to** n_s **do**
  Cost $C'_i = B_i \frac{k_\alpha - k_k \alpha_i}{D_i}$
end for
Direction of next motion $\theta_{final} = \alpha_{BW} * \min_i(C'(i))$

Distance of next motion $r_{final} = D_{\min_i}(C(i))$
end loop

Table3.3 Parameters used in experiment

α_{BW}	12°
r_{max}	8 m
r_{min}	0.18 m
α_{FOV}	165°
n_{BEAM}	15
d_R	0.15 m
d_{acc}	1.15 m
k_1	0.6
k_2	0.4

The algorithm and its performance were compared against the VPH+ algorithm on two obstacle courses shown in Figure 3.17 and Figure 3.18. First, a simple obstacle course was created consisting of a single obstacle, on which the algorithms were evaluated. Secondly, a higher obstacle density was used to create a more complicated obstacle course.

The height of the ultrasonic sensor is around 17 cm from the ground; therefore, it was attempted to ensure obstacles of a similar minimum height. The effectiveness of the algorithm in realistic settings is ensured in a disarranged office environment.

To ensure minimal errors due to pose inconsistency, the goal was set to be simply 1.20 m from the start point.

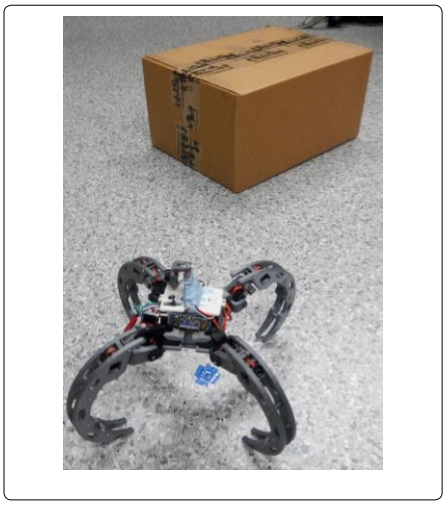


Figure 3.17 Simple obstacle course. Simple obstacle course for the robot

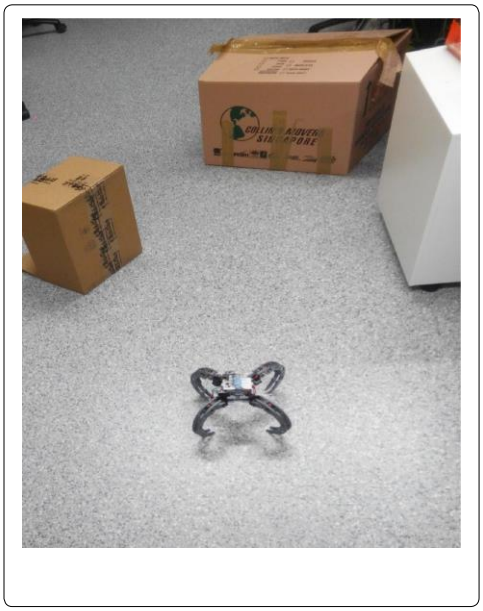


Figure 3.18 Simple obstacle course path comparison. Path comparison in simple obstacle course

The trials were conducted 6 times for each algorithm at each obstacle course.

3.6 Experimental Results

Figure 3.19 shows the comparison of the paths taken by the robot for the simple obstacle course, consisting of a single obstacle, shown in Figure 3.17As seen in the figure, the robot's path obtained from both the algorithms looks pretty similar. However, the VPH+ algorithm makes the robot to double back due to incorrect grouping of obstacle caused by the sonar's wide beam width.

The VPH+ algorithm took an average of 249.7254 s, while the CVPH+ algorithm clocked an average time of 224.3952 s.

Figure Figure 3.20 shows the comparison between the paths taken in the more complicated obstacle course shown in Figure 3.18. The fact that the sonar beam width indicates a high level of closeness of the two obstacles causes the VPH+ algorithm to erroneously group them together, resulting in the robot doubling back halfway to look for another path. The contours of two obstacles is followed by CVPH+ algorithm, resulting in a longer, more conservative path. The cause of this path is the choice of constants k1 and k2, which were optimized for the first obstacle course. Further optimization of the constants can lead to more performance gains for the CVPH+ algorithm.

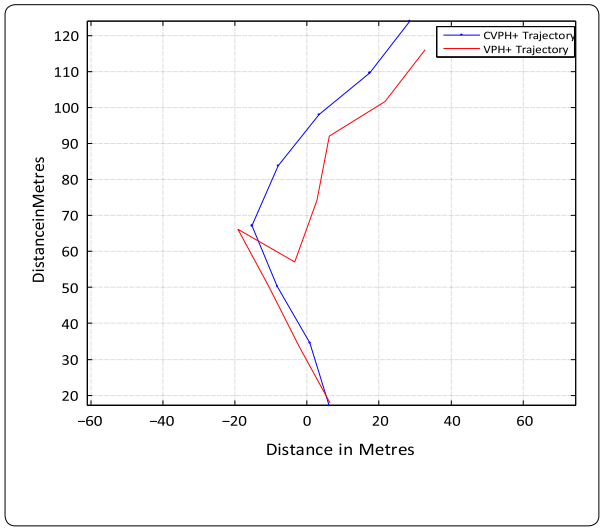


Figure 3.19 Crowded obstacle course. Crowded obstacle course for the robot

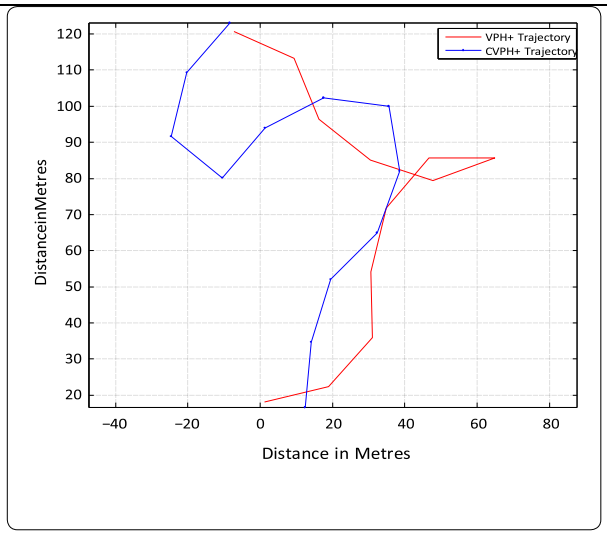


Figure 3.20 Crowded obstacle course path comparison. Path comparison in crowded obstacle course

The VPH+ algorithm took an average of 336.5960 s, while the CVPH+ algorithm took 329.4032 s.

3.7 Conclusions

The CVPH+ algorithm is found to perform better than the VPH+ algorithm in both empty and clutter-filled environments. By formalizing implicit assumptions made in the VPH+ algorithm regarding the sensor and the robot model, both of which were invalid for the Scorpio robot, the algorithm is improved.

Further insight was obtained from examining the original VPH+ cost function further and reformulating it to improve the performance of the CVPH+ algorithm.

The performance of both the VPH+ and the CVPH+ algorithms is reduced because of Pose inaccuracy due to the noisy gait primitive estimates, as well as the inherent inaccuracies due to the 3D printing process. A more comprehensive formulation taking into account pose and map estimates will result in better performance of the algorithms.

A further avenue of future research includes the use of the rolling configuration of the robot to improve its speed in traversing the environment. An interesting aspect to be explored is the challenges caused by the displacement of the ultrasonic sensors in the rolling mode of the robot.

4 Scorpio: A biomimetic reconfigurable rolling–crawling robot

Mobile robots possess entirely different abilities in their operations unlike the industrial robots which operate at one fixed location. Mobile robots face various challenges because they must move around an environment without any guidance. [119] Due to their difference in the application of Mobile robots, they are endowed to venture various degrees of freedom DOF for their smooth operations. For navigation purposes these mobile robots tend to use various ways for traversing through various terrains based on their field of application like most of the robots use wheels, legs or tracks. Based on their required motion for climbing, swimming, rolling or jumping the locomotion gaits are tuned and achieved. While some movements require different types of tuning because it differs for each motion of a robot. Because some may require lesser energy for jumping and the same may require more energy while climbing. [120] The robot may also operate inside water and swim in the aquatic areas. [121] While high speeds and energy efficiency can be attained while they roll on certain platforms. [122]

Biological species have always been a great inspiration for designing mobile robots. These robots make a concoction between nature and technical innovations. [123] Like humans most of the animals tend to move and traverse through their legs for various terrains. Because they tend to adapt and achieve locomotion easily over various platform surfaces. Nature has complemented the design of mobile robots in a variety of ways. [124–126] Li et al.[127] mentioned the subdue issue for gait trotting a quadruped robot which uses bionic springy legs. Ho et al.[128] particularizes the archetype of a robot whose leg movements are actualized using piezocomposite actuators where the design was driven from the behaviors of insects. A robot with four legs and a spherical shaped body which possess two degrees of freedom was designed from the features of a turtle.[129] In Chen et al.,[34] Most of the robots are used for various purposes like inspection, surveillance, search and rescue operations so for most of these tasks the robots might have to pass through a variety platforms and landscapes so it was realized that the use of leg mechanisms would be relevant and convenient for its locomotion. Also the behaviors of certain species of insects have inspired in designing a small robot with sex legs which possess flexible operating mechanisms.

Another new advancement for mobile robots is the ability to reconfigure itself. [130] Peculiarly designed robots which can reconfigure itself when they come across various platforms

have already been on research and efforts are done. The caterpillar robot which uses a trapezoidal wave motion for its movement was one noticeable robot with the reconfigurable capabilities. [131] Nansai et al. contemplated and planned legged robot with reconfigurable capabilities that was built from the idea of Theo Jansen mechanism.[66] Altering the distance of the links with the motors, just with one driving motor the leg could achieve various gaits. Nemoto et al.[132] have designed the rolling ability using mathematical modelling for a wheel spider one of the species found in the South African Desert Namib, [133]where the robotic equivalent for a hexapod robot was achieved through an energy consumption method that was well supported to control its motion over horizontal platforms. Chen et al.[134] has intended for a mobile that uses leg-wheel transformation. Robots that could roll and hop with its legs was proposed in Armour et al.[120] . Initially efforts to handle the issues raised by reconfigurable legged robots to traverse over various platforms was undertaken by Sinha et al.[135] Chadil et al.[136] shown and explained a three-legged spherical shaped robot with omni wheels that is reconfigurable into two hemispheres. A four-legged robot that could move using both legs and wheels was designed,[137] and formulated by using electrical paddle modules.

A mighty superiority of legged robots requires a variety of mechanisms for achieving different gaits for its various types of movements. When this kind of methodology is used it would end up for the robot requiring increased computational complicacy, increase in the size and weight of the robot as well many controllability issues would raise. To overcome such hurdles the use of a single mechanism, developing a reconfiguration would be more convenient for application.

In regards with new experimental ideas for improvements and devising of our earlier work,[138] we developed a robot named Scorpio which uses its legs for its locomotion and movement through all kinds of uneven grounds and terrains. Its legs can roll, crawl and climb efficiently on even terrains. The robot was developed with an inspiration from the huntsman spider species named Cebrennus rechenbergi. The contents of the paper are sorted out as mentioned “sec.con.” that explains how was the research encouraged and commenced. The mechanical design of the robot has been explained in ‘‘sec.str’’. The ‘‘sec.con.’’ explains the software and the hardware of the entire control system , all the reports on automated recovery, perception of reconfiguration ability m transformation and results on gait generation are narrated in ‘‘sec.exp’’ with the conclusion of the paper in ‘‘sec.concl’’.

4.1 Bio-inspiration

In the sand dunes of a desert named Erg Chebbi located at the Southern Morroccon region-boundary to the Saharan Desert, a news species of spider named Cebrennus rechenbergi rechenbergi (shown in Figure 4.1, also called the Moroccan flic-flac spider) was discovered which could even roll using the 8 legs whereas most of the spiders only walk using legs.[139] Ingo Rechenberg from TU Berlin was the discoverer of the rolling ability of these spiders.

Figure 4.1 Cebrennus rechenbergi, also known as the Moroccan flicflac spider, is a species of huntsman spider indigenous to Morocco.[139] Credit: Peter Ja¨ger / Ingo Rechenberg.

Our paper focus on the robot whose design was motivated by the specialty of the morphemics of a huntsman spider.[140]

The spider uses its eight legs to crawl. But when the spider is externally stimulated and provoked or threatened the speed of the spider for crawling is doubled by using forward and backward flips concurrently. These movement are kind of like flic-flac movements used by gymnasts. [141] The spider doesn’t use its walking gesture or doesn’t require a slop to star rolling. It uses the gravitation force and somersaults itself and rolls easily. In Figure 4.2the sequence on how the rolling locomotion happens is illustrated. While landing the spider uses its leg where the next jump is triggered.

The unusual somersault rolling motion of the spider is noticed only for certain scenarios of external stimulation. For instance, appearance of conspecifics, sand cats, or predators. There is no more clear research or observance has been done on how the spider uses its rolling ability while it

seeks down its prey or behaviors in tunnels and no specific details on whether the sex of the species play a role or not.

A new field in robotics was inspired form this. Ralf Simon King[140] who try to study and examine in detail about these spiders by using video footages and also developed a prototype of it using ROBOTIS BIOLOID named BiLBIQ – a bi-loco motional biomimetically inspired quadruped robot. There were a couple of issues that had to be resolved:

- Capabilities for autonomous and perception;
- Better design focusing on the robot's miniaturization;
- Aanalysis and examination for developing a more user-friendly design for the crawling robot to roll and shift shapes efficiently.

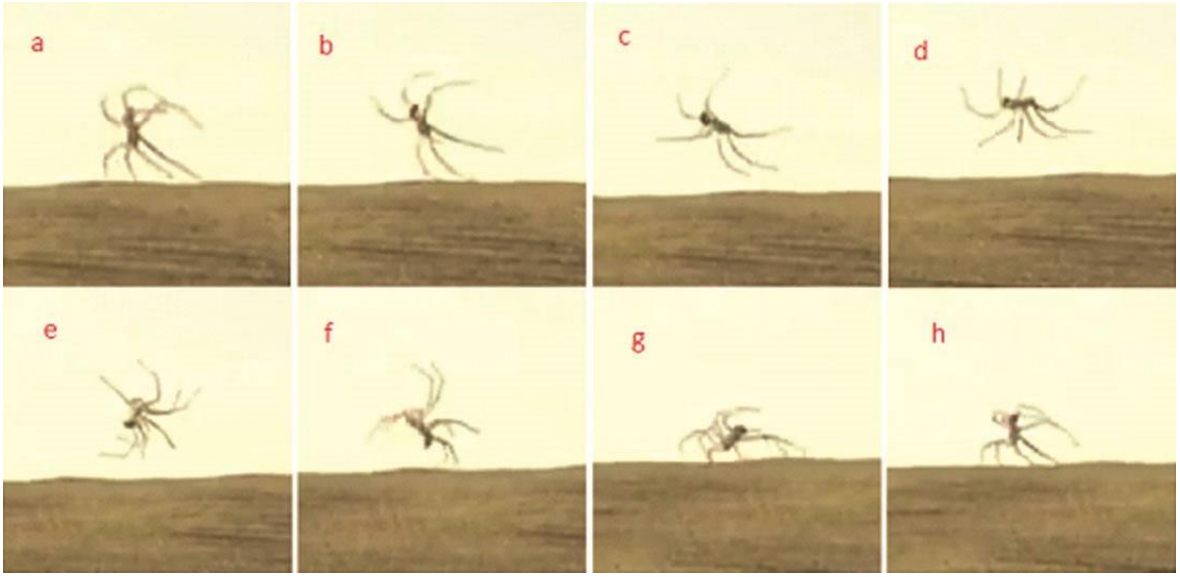


Figure 4.2 Subfigures (a) to (h) taken from the somersault sequence. Huntsman spider (Cebrennus rechenbergi) is in rolling mode, lands on its legs, and triggers a new jump.23 Credit: Peter Ja¨ger/Ingo Rechenberg.

Our work presented in the following sections focuses on tackling these issues.

4.2 Structural Design

The main aim for the project is to develop a bio-inspired robot from Centrennus rechenbergi that can comprehensively crawl and roll. During our concept generation process there were a few constraints faced while designing:

- rolling Form Diameter <180 mm;
- weight <500 g;
- power <5 kW/350 V;
- operate between 10°C and 45°C.

Following lots of product development discussions and new idea generation sessions, we came up with a new concept of a quadruped robot design that focused on achieving a perfect circumference with its legs so that can help with the robots rolling motion. To achieve perfect rolling, the curvature of the legs was designed and enhanced efficiently to make a perfect circle. Based on size and the extent of design constructs the other components for the robot like the motors, electronic boards, etc. were chosen appropriately.

After a clear observance of the behaviors of the behaviors of the Cenrennus rechenbergi spider, it was seen that it uses only 4 out of its eight legs for the rolling motion. So, for our design we decided to design a quadruped robot that could be efficiently designed for crawling and rolling. All the legs were designed to be hollow to lessen the weight and they were optimized using three-dimensional 3D printed materials.

Table 4.1 Specifications of the mechanical property of Scorpio.

Full body material	PLA (Poly lactic acid or Polyclactide)
Diameter (while rolling) in mm	168 mm
L x W x H (while walking) in mm	230 mm x 230 mm x 175 mm
Weight (Full weight) in gram	430 grams

The body was designed with discrete sections so handling or replacing specific components could be made easier. To illustrate, the battery of the robot is placed in the bottom area which is parted from the other sections of the body where other electronic components are places. Table 4.1List the Specifications of the Scorpio Robot.

Figure 4.3 illustrates the assembly of the parts of the Scorpio and the local views on where the battery is held and all the joints are placed. Figure 4.4 shows us the CAD models of the transformation from crawling to rolling. The robots four legs are opened when it crawls and for crawling the robot comprises two degrees of freedom. For the rolling motion the robot need to transform itself from crawling to a cylindrical exoskeleton form that require three degrees of freedom. By pushing its legs away from the ground and with a shift in the center of gravity the Scorpio robot attains the rolling motion with one degree of freedom. The crawling speed of the robot is doubled by the rolling speed. Figure 4.5 shows the prototype of the Scorpio robot, the poly lactic acid(PLA) plastic is used to make all the mechanical parts of the robot using a 3D printer.

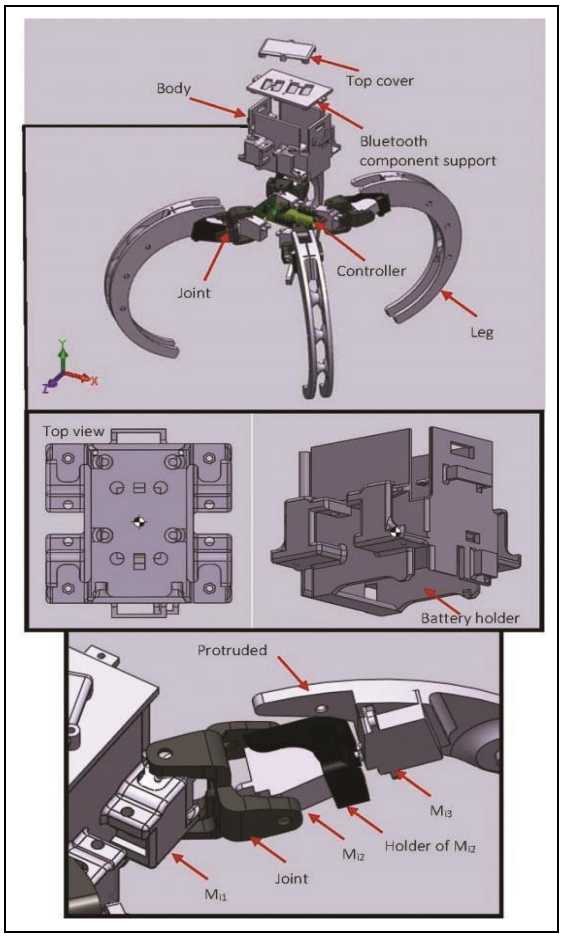


Figure 4.3 CAD models of Scorpio in exploded and local views.

4.3 Kinematic modeling

The Product of Exponentials (POE) formula is used for the forward kinematic modeling of the robot. [142] But the POE formula was used only for modelling the serial and parallel manipulators. Earlier the Denavit-Hartenberg (D-H) convention was used for the kinematic modelling of the many quadruped and hexapod robots. But the D-H convention would be outperformed as the POE formula changes the kinematic parameters, for the variation of slight changes in the joint axes. Which shows us that the error models that are singularity free. [143] So the error models become complete now. So, we use the POE formula for formalizing the forward kinematics for our Scorpio Robot. The parameters of the leg and the frame assignments are shown in Figure 4.6. frame R is assigned as the reference frame which is located at the mid of the robot's body. The point where there contact with the ground originates is set as Frame T.

For the *i*th leg, given the joint variables θ_{ij} (j=1, 2, 3), the corresponding exponential of a twist is

$$e^{\xi_{ij}\theta_{ij}} = \begin{bmatrix} e^{\omega_{ij}\theta_{ij}} & \left(I - e^{\omega_{ij}\theta_{ij}}\right)q_{ij} \\ 0 & 1 \end{bmatrix} \tag{4.1}$$

The forward kinematics map of a leg is

$$\mathrm{g}_i\left(\theta_i\right) = e^{\xi_{i1}\theta_{i1}} e^{\xi_{i2}\theta_{i2}} e^{\xi_{i3}\theta_{i3}} \mathrm{g}_i\left(0\right) \tag{4.2}$$

where $s_{ij}=\sin\theta_{ij}$, $c_{ij}=\cos\theta_{ij}$,

$$e^{\xi_{i1}\theta_{i1}} = \begin{bmatrix} c_{i1} & -s_{i1} & 0 & 0 \\ s_{i1} & c_{i1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$e^{\xi_{i2}\theta_{i2}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{i2} & s_{i2} & \left(1-c_{i2}\right)L_{i1} \\ 0 & -s_{i2} & c_{i2} & L_{i1}s_{i2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$e^{\xi_{i3}\theta_{i3}} = \begin{bmatrix} c_{i3} & 0 & s_{i3} & 0 \\ 0 & 1 & 0 & 0 \\ -s_{i3} & 0 & c_{i3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathrm{g}_i\left(0\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a_{iy} \\ 0 & 0 & 1 & a_{iz} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

which is the initial configuration when θ_i =0.

Expanding the terms in the product of exponentials formula (2) yields

$$\mathrm{g}_i\left(\theta_i\right) = \begin{bmatrix} R\left(\theta_i\right) & P\left(\theta_i\right) \\ 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} c_{i1}c_{i3}+s_{i1}s_{i2}s_{i3} & -s_{i1}c_{i2} & c_{i1}s_{i3}-s_{i1}s_{i2}c_{i3} & -a_{iy}s_{i1}c_{i2}+a_{iz}\left(c_{i1}s_{i3}-s_{i1}s_{i2}c_{i3}\right)-L_{i1}s_{i1}\left(1-c_{i2}\right) \\ s_{i1}c_{i3}-c_{i1}s_{i2}s_{i3} & c_{i1}c_{i2} & s_{i1}s_{i3}+c_{i1}s_{i2}c_{i3} & a_{iy}c_{i1}c_{i2}+a_{iz}\left(s_{i1}s_{i3}+c_{i1}s_{i2}c_{i3}\right)+L_{i1}c_{i1}\left(1-c_{i2}\right) \\ -c_{i2}s_{i3} & -s_{i2} & c_{i2}c_{i3} & -a_{iy}s_{i2}+a_{iz}c_{i2}c_{i3}+L_{i1}s_{i2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

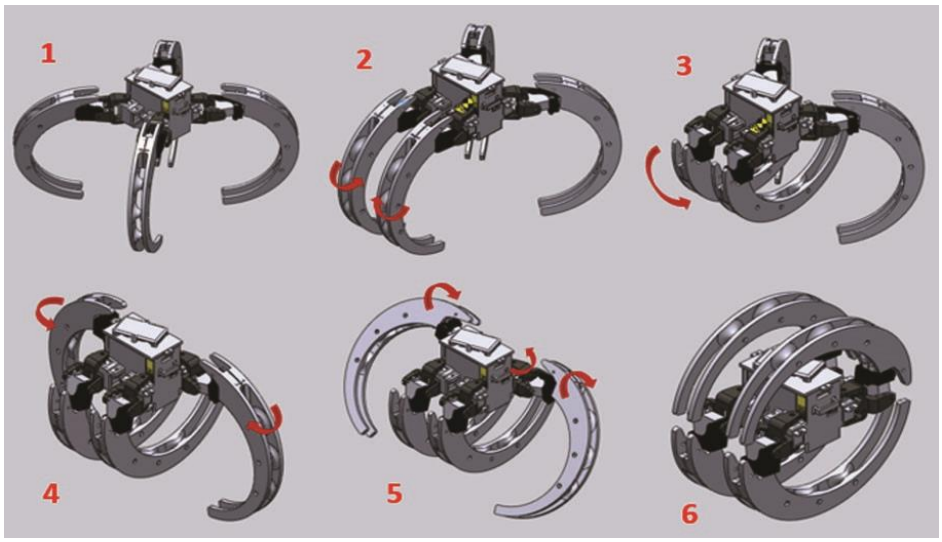


Figure 4.4 Transformation from crawling to rolling in CAD models.

Figure 4.5 Prototype of Scorpio robot.

where $R(\theta_i)\mathrm{R}(\theta_i)$ is the orientation coordinate and $P(\theta_i)\mathrm{R}(\theta_i)$ is the position coordinate.

Matrix g_{RSi} transforms the coordinates from Frame *S* into Frame *R*, which is expressed as $g_{RSi} = \begin{bmatrix} R_{i0} & P_{i0} \\ 0 & 1 \end{bmatrix}$.

The forward kinematics of the robot is represented with a kinematic set of four legs. $S = \{g_i(\theta_i) g_{RSi}, i = 1, \ldots, 4\}$. Ideally the parameters $L_{i1}R(\theta_i)$, L_{i2}, α_{iy}, α_{iz}, R_{i0}, and P_{i0} are the same for four legs in terms of mechanical design. Real values may differ lightly due to errors in fabricating. There is a need to perform kinematic calibration to detect the real values. We can also see that there is no L_{i2} L_{i2} in (2). This is because the information of L_{i2}, has been included in

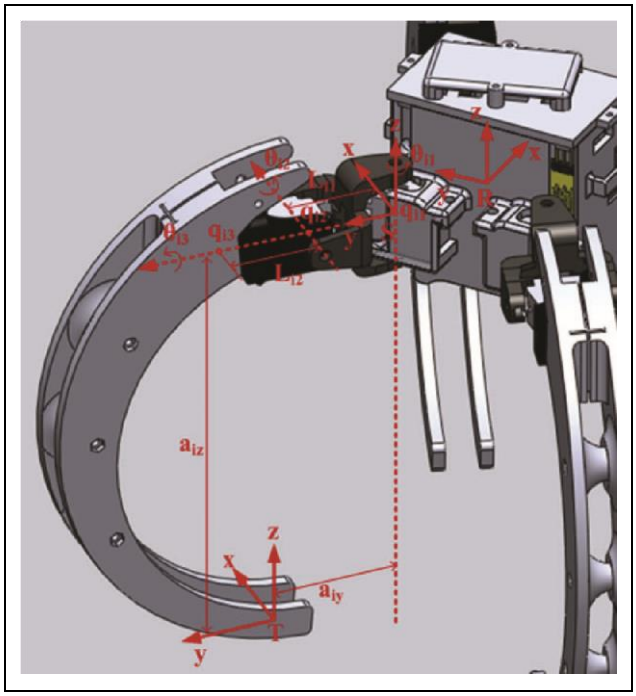


Figure 4.6 Frames assignment of a single leg following the product of exponentials formula.

α_{iy}. In this sense, L_{i2} is the redundant parameter which could be neglected.

4.4 Control system

The Scorpio robot needs to be equipped with software and hardware components than can be efficient enough to control the system so that it can deployed efficiently and could also make autonomous exploration.

Table 4.2 Specifications of the hardware used for Scorpio.

Controller	Arduino Mini pro 328
Servo Motor	JR ES 376
Servo Controller	Pololu-Micro Maestro 18-channel USB servo controller
IMU (Inertial measurement unit)	MinIMU-9 v2
Camera	WiFi Ai-Ball Cam
Battery	Li-Po 1200mah 7.4v
Bluetooth controller	EZ-Link Bluetooth serial link & Arduino Programmer -v1.3
Power regulator	Dimension Engineering De-SW033

that it can deployed efficiently and could also make autonomous exploration.

Hardware components

Hardware components of the robot both with and outside comprises of a microcontroller, component for wireless communications, mobility unit, power unit, exteroceptive and interoceptive sensors.

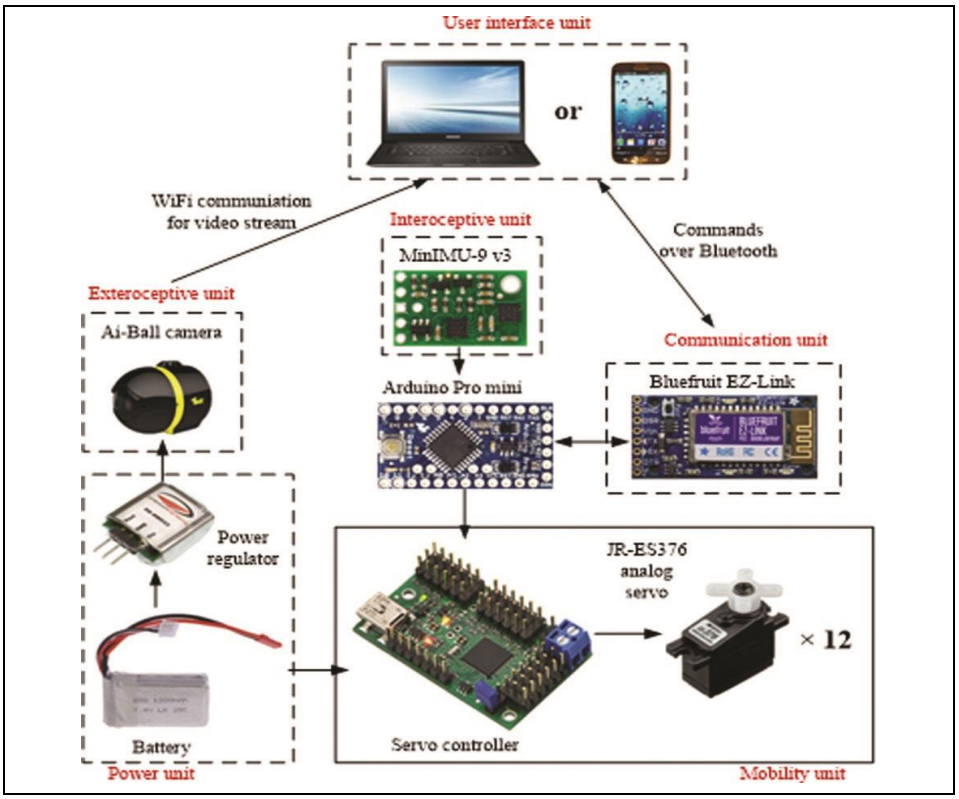


Figure 4.7 Hardware connection.

Table 4.2 lists the hardware components and its specifications

Microcontroller. The microcontroller we have chosen is Arduino Pro Mini. The Hardware connections are shown in Figure 4.7

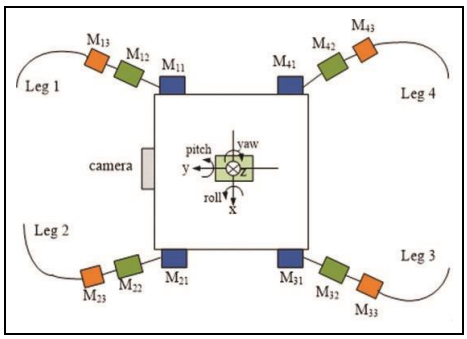


Figure 4.8 Schematic diagram of locations of servo motors and the reference frame of the IMU with roll, pitch, and yaw indicated.

the interoceptive, communication and mobility units are directly connected to the Arduino Pro Mini which is located at the center.

Communication. The spider robot control mode could be either automated or teleoperated. There is Bluetooth device (Adafruit Bluefruit EZLink) that in installed in the robot that helps for communication with the handset and the host computer. This Bluetooth device makes it easier for communication purpose and has transparent Arduino re-programming ability. So, in regard to communication for this device the commands and compiling could be made wirelessly.

Joints/motors. Figure 4.8 shows where the 12 servo motors are located in the robot. For the purpose of motion and transformation 12 ES376 servo motors from JR Pro are used inside the robot. These servo motors weigh 11.5 g, uses 4.8V and are ultra-compact (dimension 21.5 11.5 21.5 mm). To drive these motors and the interface of the microcontroller the robot is also equipped with Pololu Maestro servo controller.

Sensors. For the making the robot autonomous a variety of sensors have been implemented in it. The IMU Sensor Inertial Measurement Unit and the Camera are used for the interoception and exteroception stimulations. One Ai-Ball camera is used in the vision unit. The camera could be operated wirelessly through Wi-Fi with a 60-view angle. The camera weighs very minimum but to adapt for the minimization of the robot, the shell and the battery of the camera is removed and fitted over the central part of the controller. The Pololu MinIMU-9 v3 IMU sensor contains a L3GD20H three-axis gyro, an LSM303D three-axis accelerometer, and a three-axis magnetometer is mounted on the board. For calculating the absolute orientation of the sensor, it uses an I^2C to read the independent rotation, acceleration and magnetic measurements.

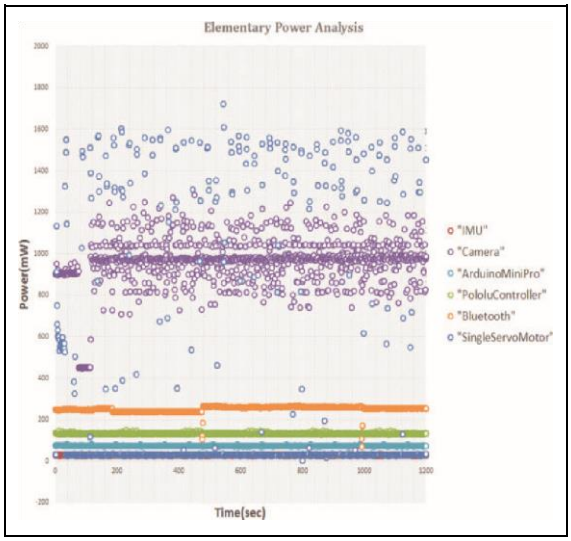


Figure 4.9 Power consumption of six elementary components in the robotic system.

Along with this a voltage regulator and a level-shifting circuit is present on the MinIMU-9 v3 which allows it to operate within 2.5V to 5.5V. Using exponential formula, a representation of the roll, pitch and yaw movement for a single leg are represented in the Figure 4.6 by a Schematic diagram.

Power supply. A Li-Po battery (1200mah 7.4) is used for power supply for the entire robot system. The servo controller is connected to the battery. For the camera a power regulator (Dimension Engineering DeSW033) is used to have consistent voltage and they are connected to the battery. Figure 4.9 shows us the power consumption test for the six components in no load state. The graphs show us that most of the energy are used by servo motors and the camera, the power consumption of Bluetooth controllers is the maximum, Pololu controllers and Arduino mini consume medium energy and the IMU sensor uses very less energy.

4.5 Software system

Two user interfaces one is Graphical User Interface (GUI) for the computer and one Android Application for handset are created with the current prototype.

For the purpose of controlling and monitoring the robot an GUI (Figure 4.10) written in C# was integrated so that the operator may use the control software to have interaction and control over the robot. This interface would be able to display the original video as well the one that is being processed. There are a serious of buttons placed below the display for that the operator can choose the required operation to be done.

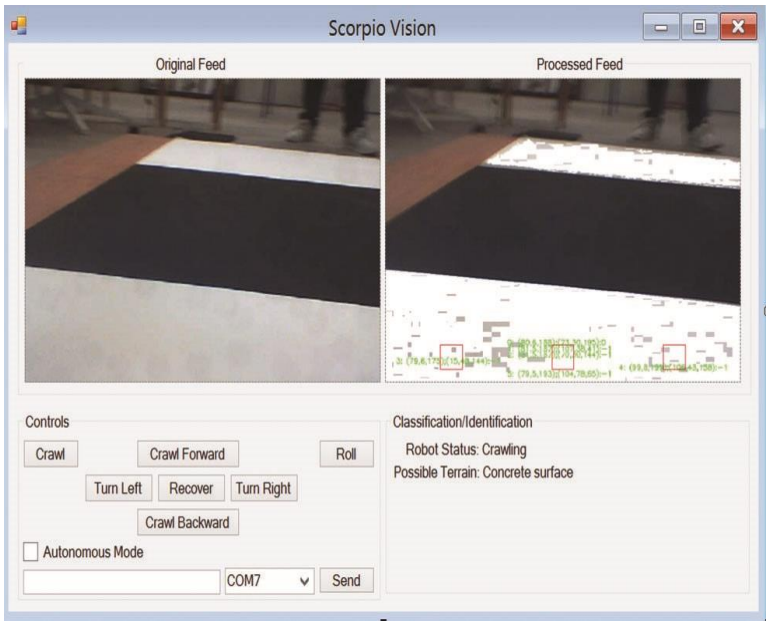


Figure 4.10 Software interface running in the host computer.

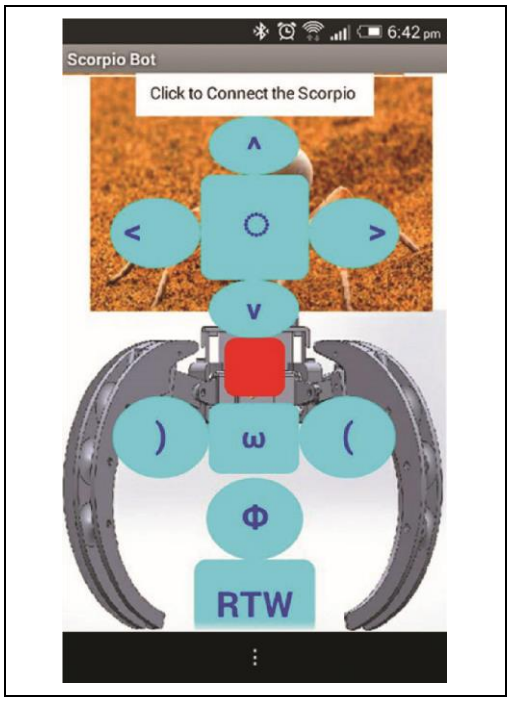


Figure 4.11 Android App interface running in the handset.

Table 4.3 Functions of the buttons on the Android App interface.

Button	Function	Button	Function
Click to Connect the Scorpio	Connecting the Bluetooth	ω	Upside down recovery
o	Rolling	Φ	Sideway recovery
^	Forward	)	Walking left
v	Backward	(	Walking right
<	Turning left	RTW	Rough terrain walking
>	Turning right	Red button	Stop
		Φ	Sideway recovery

The check box on the interface allows the operator to choose modes. He can check it the robot needs to be shifted to autonomous mode. Using this interface, the operator can send commands to enable and control for rolling, crawling, recovering mode and through the communication port COM7. With the use of the check box the robot would be able to turn to autonomous mode without the need for any human intervention. The desktop which is located remotely receives signals and videos from the camera and the robot. The other display shows us the status of movement of the robot and the terrains the robot passes by.

Figure 4.11 show us the Android application that runs in a handset. The app was created using a MIT App Inventor 2. [144] The meanings and functions of each symbols for operating the robot is mentioned in the Table 4.4.

Table 4.4 Joints (servo motors) actuated in different gaits (the actuated ones are marked with *).

Gait	Joints											
	Leg 1			Leg 2			Leg 3			Leg 4		
	M_{11}	M_{12}	M_{13}	M_{21}	M_{22}	M_{23}	M_{31}	M_{32}	M_{33}	M_{41}	M_{42}	M_{43}
Crawling	*	*		*	*		*	*		*	*	
Crawling to rolling	*	*		*	*		*	*	*	*	*	*
Rolling		*			*			*			*	
Sideway recovery												

	*	*	*	*	*	*	*	*	*	*	*	*
Upside down recovery	*	*	*	*	*	*	*	*	*	*	*	*

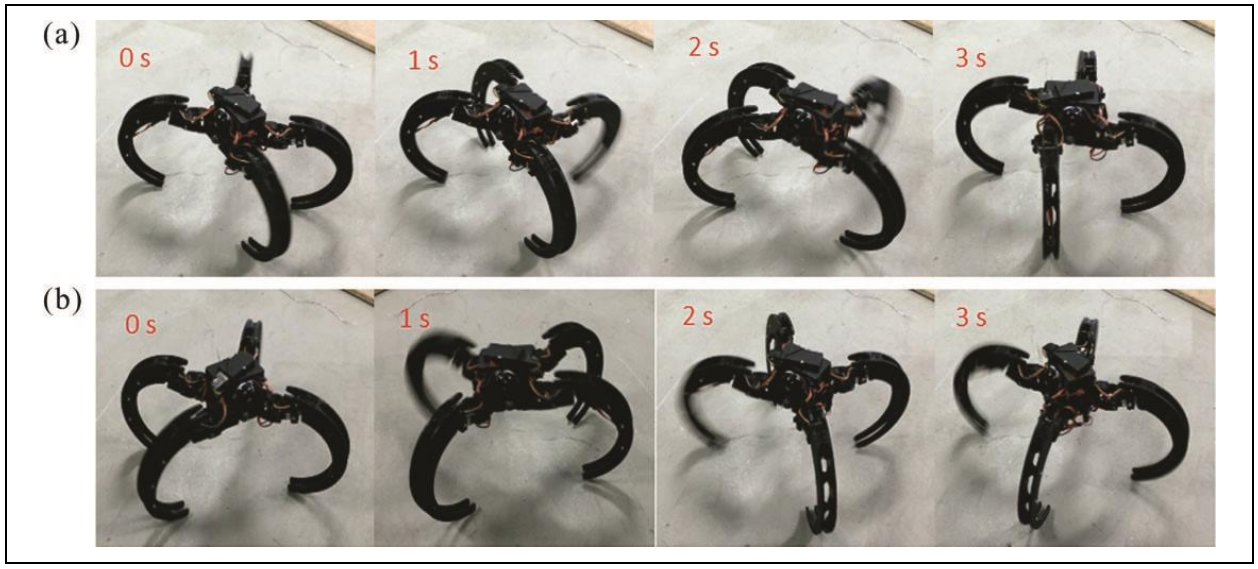


Figure 4.12 Turning left (a) and right (b) in crawling mode.

4.6 Experimental Result

Gaits generation and transformation

The Scorpio has enough degrees of freedom to perform most of the motions required. The mechanical design and architecture is designed well so that all the needed DOF are achieved and the robot could crawl and roll using its legs by transforming itself. The servo motor rotations control this transformation. Where these joints of the servo motors are actuated in various gaits. Table 4.4 shows us the how each motor is actuated with the various gaits. “the actuated gaits are marked * “ The M_{i1} and M_{i2} for four legs have been actuated to perform the following operations: crawl forward, backward, leftward and rightward. The turning left and right crawling motion is shown in the Figure 4.12 (a)left and (b)right.

The robot to start rolling from crawling, the legs 3 and 4 would form a circle shape with the legs 1 and 2, by changing their angles of rotation. The full phases of transformation from crawling to rolling is shown in the Figure 4.13. The Scorpio changes its gaits when it shifts from crawling to

rolling and again when its returning from rolling to crawling. The rolling locomotion gaits for the robot is shown in the flowchart shown in Figure 4.14. The angle for rolling and the angular velocity of the rolling are measured by the IMU sensor while the robot is rolling is indicated in the Figure 4.15. This angular velocity indicated helps us determine whether the legs must actively always support the robot while it performs rolling and needs to be actuated or not. For example, if the angular velocity is 200/s the legs don't get actuated else the legs have to push against the ground and co-ordinate together to induce the gait required for rolling. Figure 4.12 shows us a series of angular velocities ranging from rolling, pitching, yaw movement, crawling. Around 16 angles and angular velocities are measured using the IMU the robot shifts from crawling to rolling.

Recovery

To traverse over smooth terrains the rolling motion is most apt. But the rolling motion also faces many problems like when they face obstacles the robot might lose its control and tend to fall or when some force from the side arrives it might lose control and fall over its sides. For practical application the recovery is not an easy task. So, to overcome these issues the IMU sensor collects the roll and pitch details and also the details regarding terrain perception by vision are done, by using these details the robot makes the recovery when it falls. The entre control system is indicated by the flowchart in Figure 4.17 . The typical sideways recovery motion of the Scorpio are indicated in the Figure 4.18.

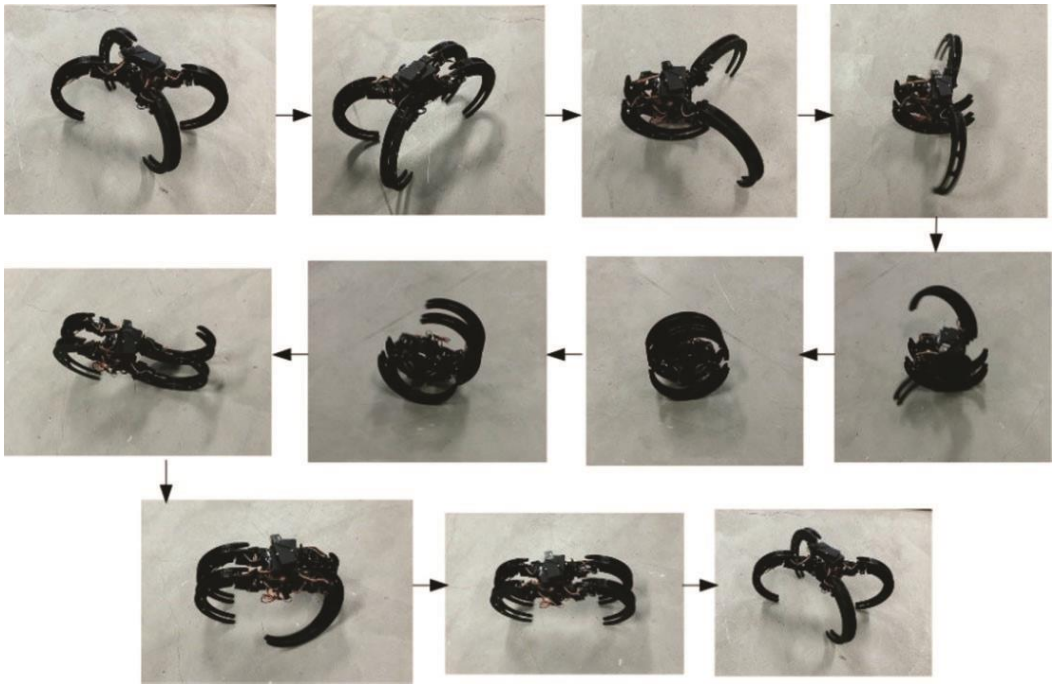


Figure 4.13 Transformation phases between crawling and rolling (crawling→rolling→crawling).

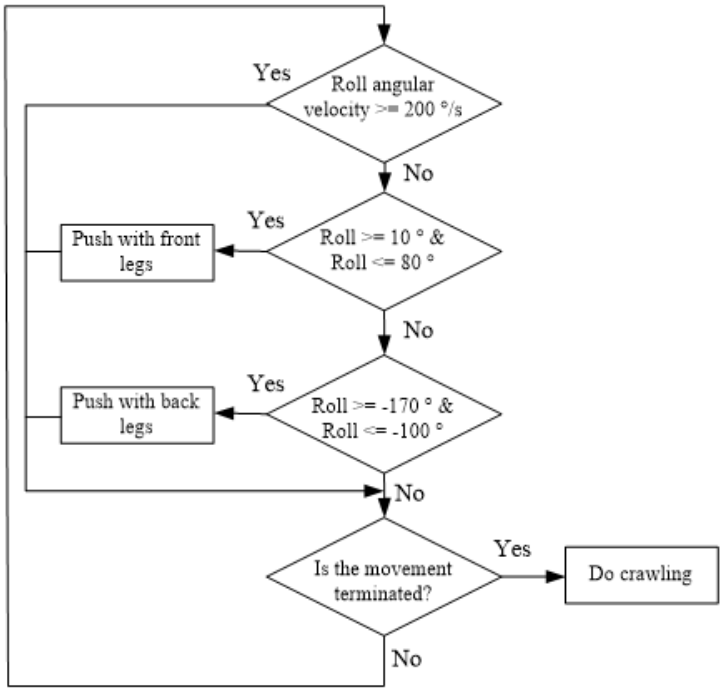


Figure 4.14 Flow chart of rolling gait generation of Scorpio robot.

For recovery the robot actuates the two legs of a half circle, then stretches all the four legs. The two legs on the floor push the robot back to the standing stage. At this point all the motors of the robot are actuated.

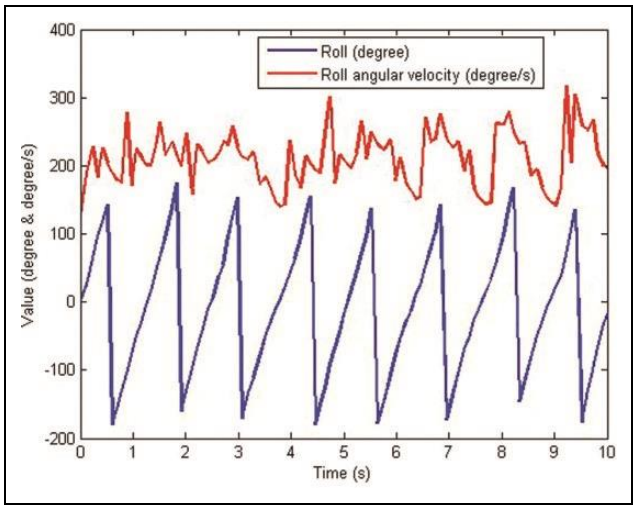


Figure 4.15 Roll and roll angular velocity measured by IMU during rolling motion.

In case the robot falls upside down the robot might face issues for the crawling gait. For such situations the reconfigurability mode is activated so that all the joints of the robot would help to recover itself from the upside-down position to the standing position. In Figure 4.19 the recovery modes of the Scorpio robot are shown. The robot rotates one pair of legs first to the opposite direction and then the next pair of legs. Eventually it pushes against the ground and regains to its default position.

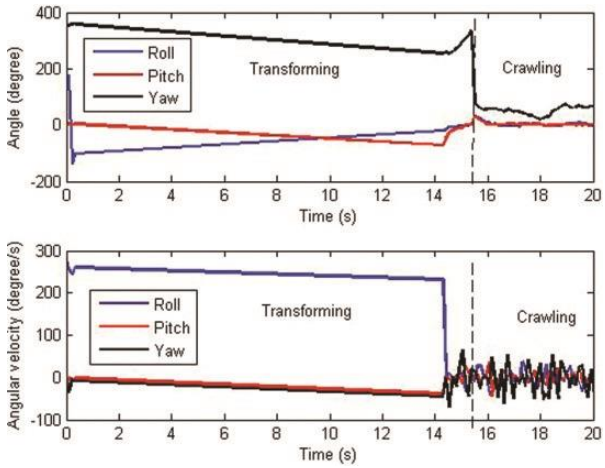


Figure 4.16 Angles and angular velocities measured by IMU when rolling changes to crawling.

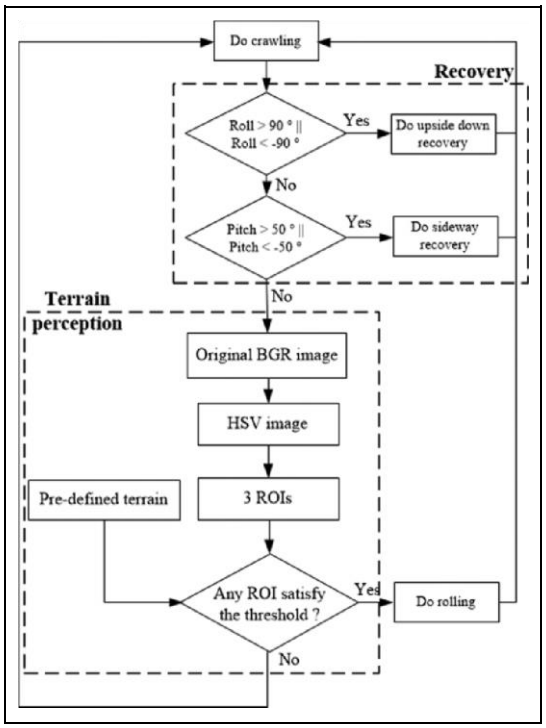


Figure 4.17 Flow-chart of the whole control system including recovery by IMU and terrain-perception based reconfiguration algorithm by vision.

So, the joints of the four legs helps the Scorpio for its ability to recover from any position. The details of pitching and roll angles that are collected by the IMU while the robot transforms from sideways and upside-down position are indicated in Figure 4.20. There has a difference in the values of before the robot has fallen and after it has recovered safely.

This restoration capableness enhances the robustness and versatileness of the robot in the extremity of navigation. The Scorpio escalating through stairs, by reconfiguring is shown in the Figure 4.21. Using manual mode, the Scorpio rolling through stair has been achieved. It uses the rolling gait to over stairs and then trips after escalating though some stairs. It uses the four legs stretches them and pushes against the stair to move down.

Terrain perception

IMU-based perception. Apart from its ability to recover itself the Scorpio also has the ability to traverse and adapt itself to various terrains. This increases the robustness and versatility of the Scorpio robot. This reconfigurability tendency makes the Scorpio robot more comfortable and efficiently perform various type of locomotion. The rolling gait of the robot while it reconfigures itself from going down a slope from a horizontal surface is shown in Figure 4.22. When the robot overcomes certain angle while rolling and when there is a deviation of the rolling angle beyond a certain threshold the IMU sensor detects and makes safe transformation. The Scorpio robot uses the gravitational force and moves downwards with the inertia. The rolling motion in this situation is much more competent

than the crawling ones in condition where the moving speed and consuming energy is efficient. So the IMU sensor is used for detecting the slopes and decides whether the robots requires to change its gait.

Vision-based perception. The interoceptive capability for the Scorpio robot is achieved through the IMO sensor. Similarly, the exteroceptive capabilities for perception of terrains is done using the vision. The various types of terrains are recognized and analyzed by the vision system. A multi-tier architecture for the vision has been implemented in this system, to ensure there is no disturbance in the operation and the stability and scalability of the system is well maintained through the entire operation. As well as the the real-time videos sent to the remote computer for processing and feedbacks reduces the on-board processing time.

We have used a color-based method for our image processing algorithm that shows us how the perception of terrains is calculated using vision. The terrains are detected and segmented and processed based on the image and the color difference is taken into consideration. Using the robot's vision there are multiple regions of interests applied to analyze what type if terrain it is. Figure 4.23

shows us the real-time image of the vision. The three squares indicated shows that field of the camera in the robot.

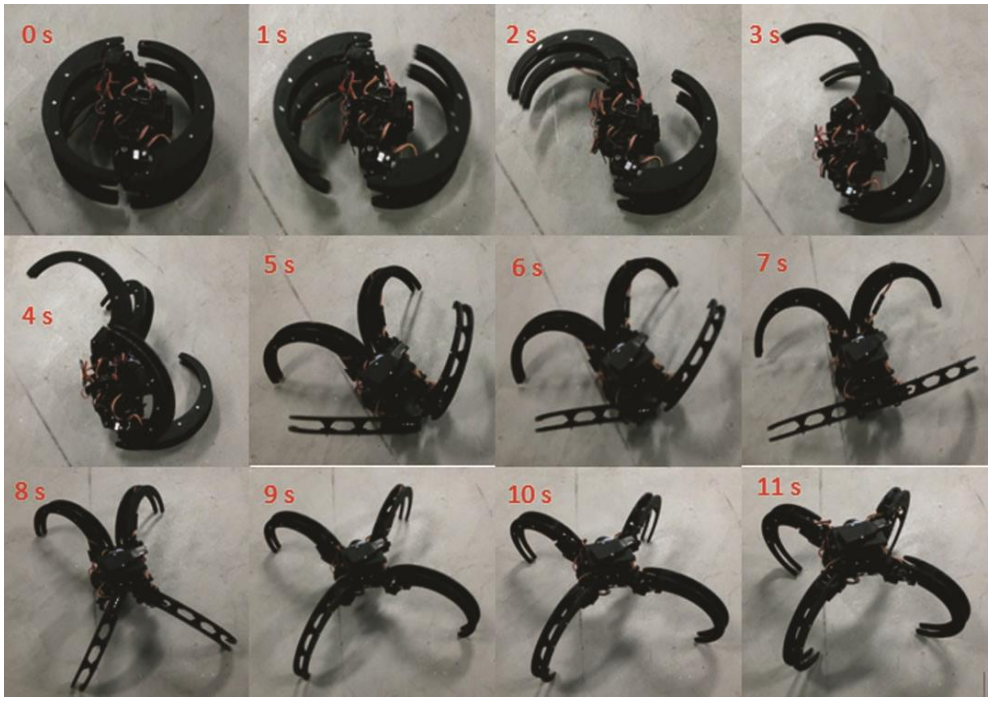


Figure 4.18 Sideways recovery of Scorpio.

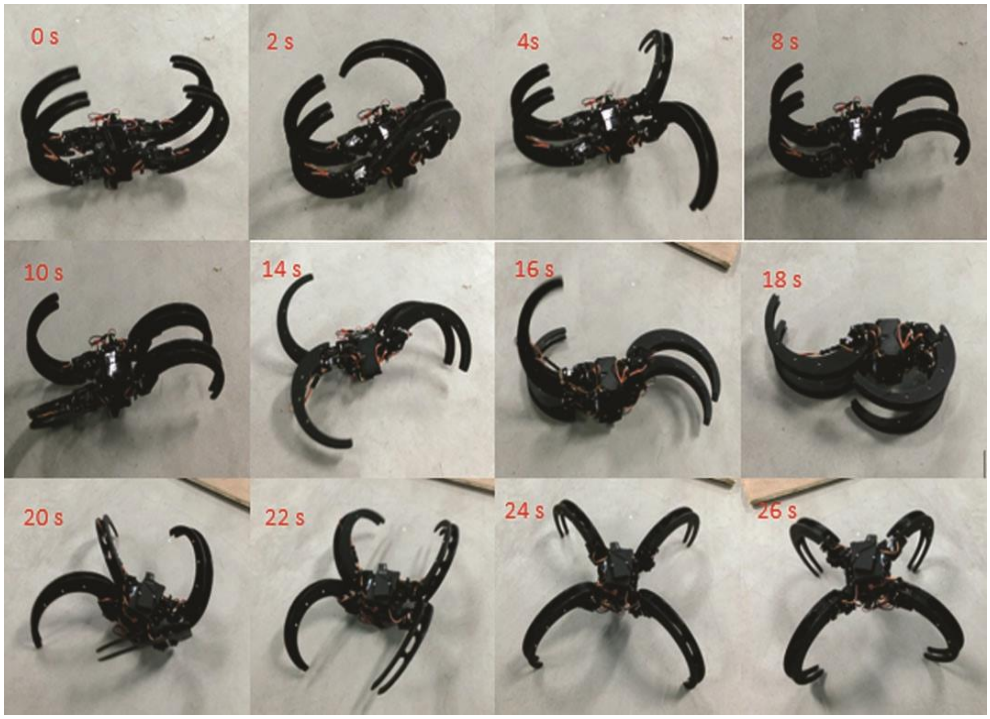


Figure 4.19 Upside down recovery of Scorpio.

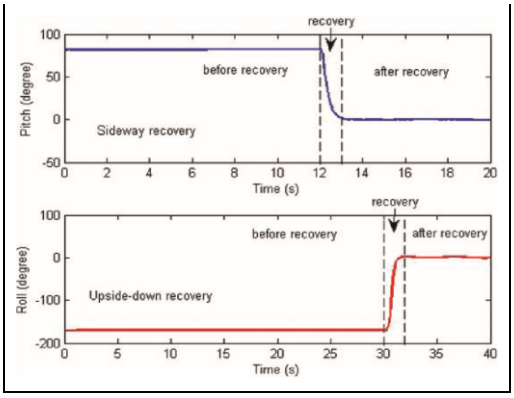


Figure 4.20 Pitch and roll angles measured by IMU during side- ways and upside-down recoveries respectively.

Terrain classifications are done by comparing the average color space over the areas it comes across along with the classifications that were determined earlier. The general concept is that similar terrains possess same color, if there is a change in terrains possibilities for change in color is possible. The flowchart for the entire control system based on both IMU perception and the vison based perception is shown in Figure 4.17. Based on the HSV values (Hue, Saturation and value) the pixel values for the image processing is done. For every ROI the pixel values are collected and the average is also calculated. Using predefined terrains and a real time streaming of the video and their parameters are shown in Figure 4.24. where (a) are few samples of predefined terrains; (b) real-time detected terrains. So now the parameter values are compared with those of the predefined values and checked. If there are variations in the HSV values of the ROI above the mentioned threshold then it is a new type

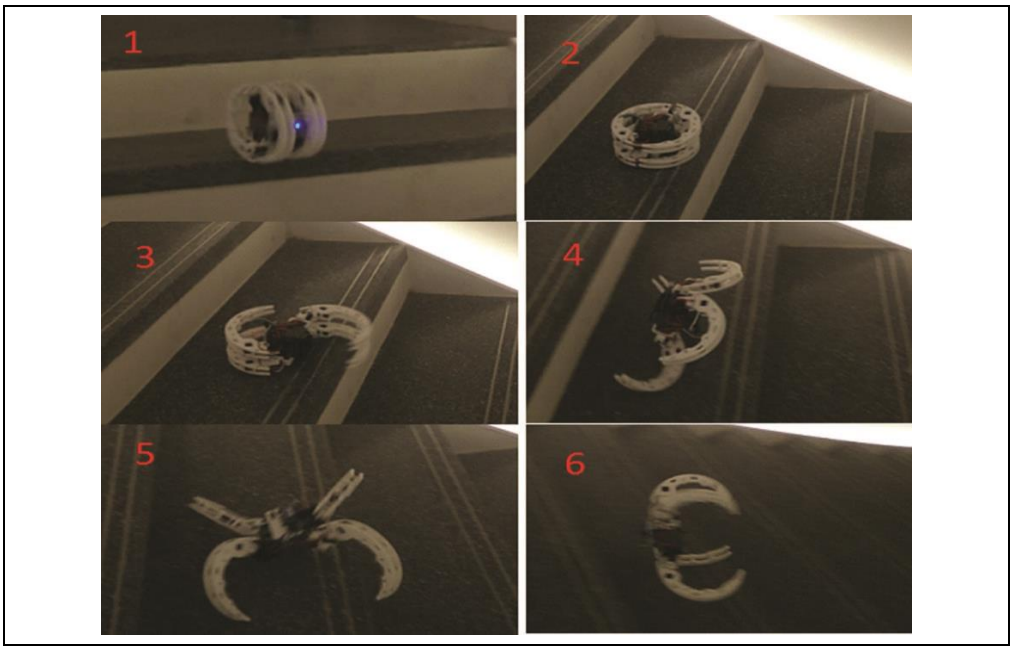


Figure 4.21 Scorpio going down stairs by reconfiguration.

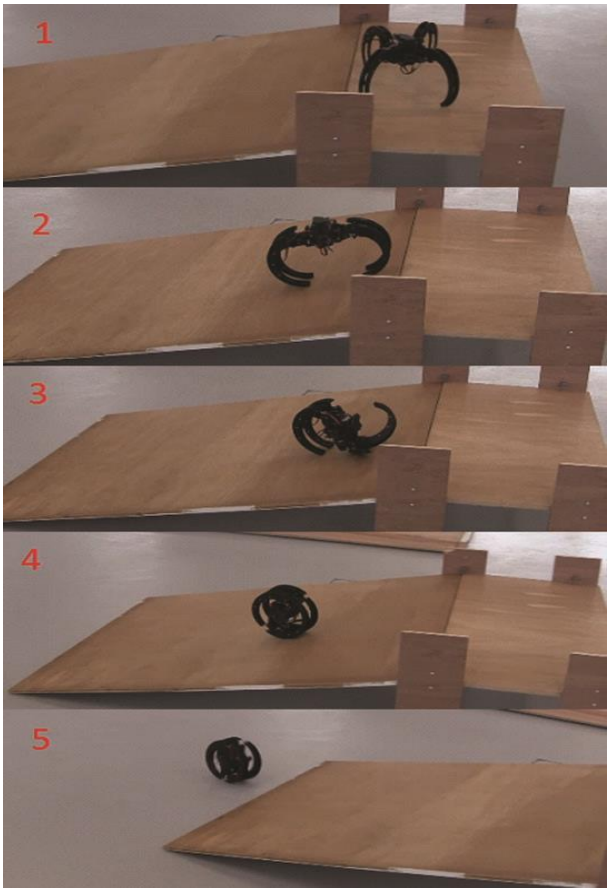


Figure 4.22 Scorpio going down the slope by reconfiguration.

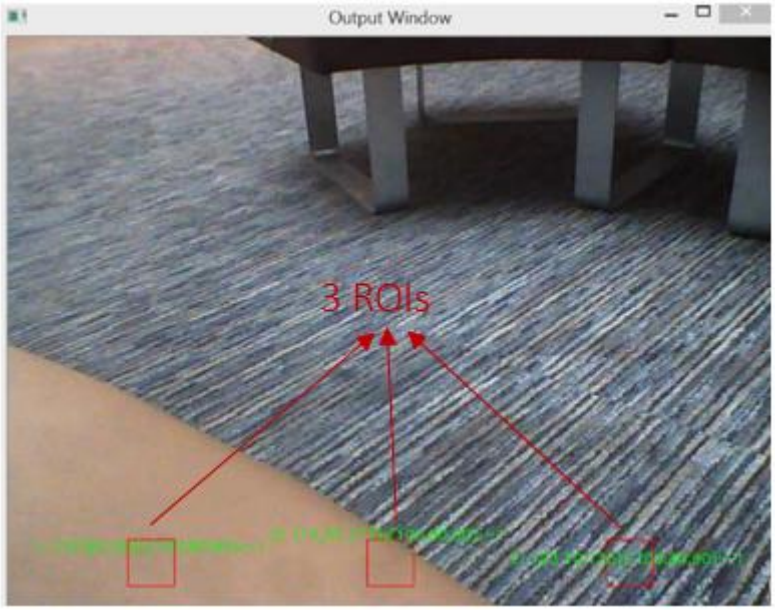


Figure 4.23 Example of real-time image from the vision.

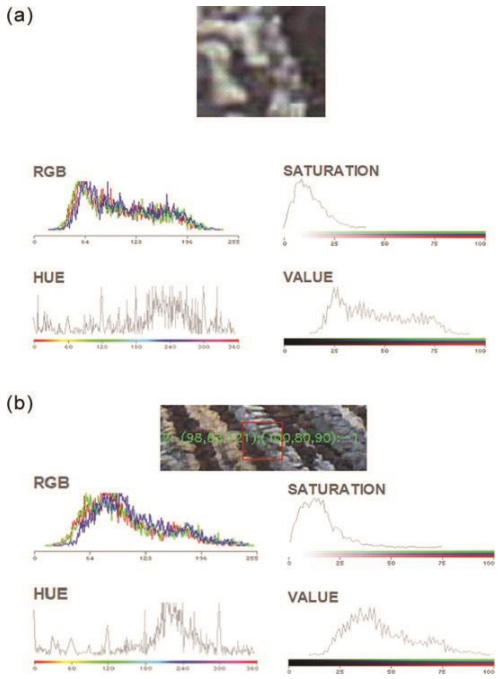


Figure 4.24 Pre-defined terrain type and real-time video stream from robot's vision and their parameters: (a) example of predefined terrain; (b) real-time detected terrain.

of terrain if it falls within those threshold values it is one of the predefined terrains. By a series of trial and error methods the threshold values are set. For our case the values are set between ±30 or ±40.

Algorithm 1 Color-based terrain perception.

Data: Original Blue, Green, and Red (BGR) image I of the terrain, pre-defined image Ip, thresholds *h, s, v*

Result: Corresponding gait adapted to the terrain

Initialization

HSV image $J \leftarrow$ convertColor(I_p)

Pre-defined HSV image $J_p \leftarrow$ convertColor(I_p)

Three *ROI* s $\leftarrow$ select(J)

Three ROI_p s $\leftarrow$ select(J_p)

$H_p, S_p, V_p \leftarrow$ mean(ROI_p)

for *m=1,2,3* **do**

$H_m, S_m, V_m \leftarrow$ mean(ROI_m)

Errors $herr_m$, $serr_m$, $verr_m$ $\| H_m - H_p \|$,

$\| S_m - S_p \|$, $\| V_m - V_p \|$

end

if $herr_m < h$ && $serr_m < s$ && $verr_m < v$ then

doRolling;

else

doCrawling;

end

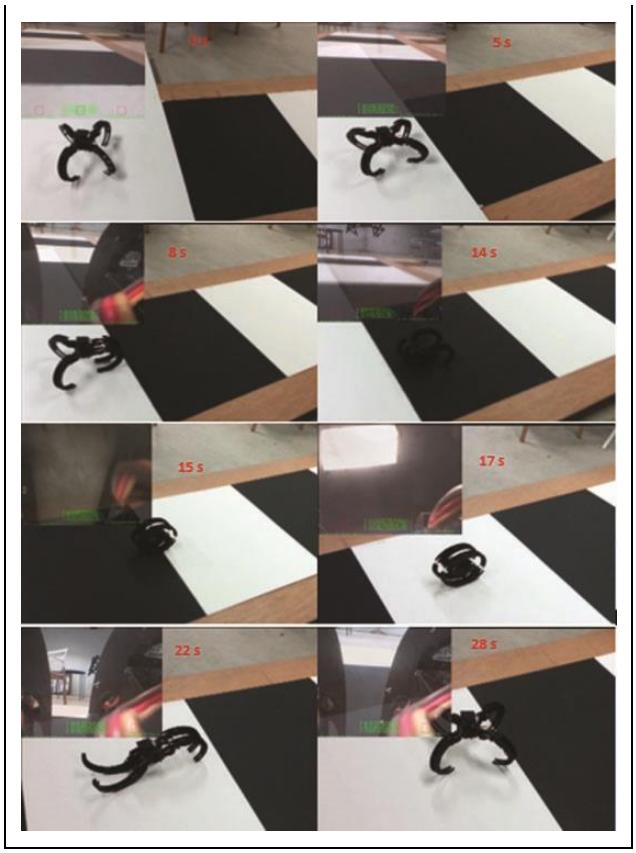


Figure 4.25 Black and white terrain perception by on-board camera and reconfiguration gait generation based on the visual information.

For illustrating and demonstrating the reconfiguration capability of the Scorpio, we adopted a method of using black and white cardboard over the floor. We see whether the terrain perception generates the required gaits based on the visual data collected. In Figure 4.25 we can see the perception using vision on black and white surfaces. The top-left pane shows the view from the camera on the board. The robot would decide to crawl if white color is detected, if not the robots detects black color it would reconfigure itself to rolling mode. Once it rolls the robot transforms to crawling position and does the terrain perception to analyze whether to crawl further or roll. In the subfigures the robot at 14t, 15, and 17s after rolling transforms itself to crawling at 22 and 28s. So based on these perceptions the robot decides what to do.

4.7 Conclusions

In this chapter, we described and recited the mechanical design and performance testing of a uniquely designed legged robot Scorpio inspired a Cenrennus rechenbergi spider which

can crawl and roll. The Scorpio robot has 4 legs that are reconfigurable and has the capability to change its morphology for transforming itself from crawling to rolling. The uniqueness of not using any extra mechanism and by using the same set of actuators the robot can efficiently crawl and roll. The mechanical design of the robot, its control design, sensors and software's are explained in detail.

The robots' performance based on each type of perception is experimented and showcased. As confirmed the robot is capable to crawl rolls and transform between these two states. From the interoceptive perspective the robot can transform itself while it falls sideways and turns upside-down. From the exteroceptive side the vison system has helped the robot in perceiving the terrains and able to actuate the required gaits.

For further research the autonomous capabilities will be explored more widely. Advanced algorithms for perception and identification of various terrains and floorings would be studied. Also the system would be well developed to autonomously faces certain bumps and obstacles and escalate down the stairs using advanced vision calculation An more energy efficient system would be developed for energy management and highly capable to work competently within the power limit.

5 Fault Diagnosis of a Reconfigurable Crawling–Rolling Robot Based on Support Vector

Machines in the last few decades, robotic applications have drastically increased, and continue to increase, as we advance towards more sophisticated and fast-paced development environments. Modern robots are highly complex mechatronic systems with hardware and software modules that have a diverging set of features. Due to the highly complex nature of next-generation robotic systems, and the uncertain environments they occupy, modern robots are highly likely to encounter faults during runtime. Even well-designed robotic hardware will encounter a fault in its lifetime. Machine learning techniques have been used for automated fault diagnosis in many industrial applications. In [145], the authors proposed a fault diagnosis method for a spur bevel gear box. The statistical features used in this procedure are determined via wavelet coefficients of the vibration signals. This work uses Artificial Neural Networks (ANN), which provided an accuracy of 97.5%, and Proximal Support Vector Machines (SVM), which provided an accuracy of 97% for classification. In [146], Sheng-Fa Yuan proposed a fault detection technique for turbo-pump motors by using a Multi-class Support Vector Machine for classification and Principal Component Analysis (PCA) for dimensionality reduction.

This model exhibits a high level of precision, in the range of 95% to 100%, when tested with different kinds of Support Vector Machine methods. In [147], a fault diagnosis technique, which uses multiple components, is implemented for rotational mechanical systems using Support Vector Machine and decision tree methods. This work uses the vibration signals of machinery to extract the statistical features needed for classification. A neural network-based fault diagnosis technique for rotational machine parts is presented in [148]. The two neural networks used in this study were either non-linear radial basis function-based or time delay-based neural networks. It was tested in the automatic gear system of automobiles using their acoustic data. Another similar machine learning-based approach is proposed in [149] for figuring out the faults in ball bearings. A high-speed rotor supported on a ball bearing is used to collect the vibration-signal data; the features are extracted from this data set.

The classification is done using Support Vector Machines and Artificial Neural Networks. Aside from SVM and ANN, various other techniques have been explored in the context of fault

diagnosis in industrial settings. In [150], a Non-linear Unknown Input Observer was used to diagnose a malfunctioning part in an autonomous spacecraft used in the rendezvous phase of the Mars Sample Return (MSR) mission. This approach does not involve machine learning. Instead, it uses observers and position models to detect faults. The approach in [98] uses an analytical redundancy method to diagnose faults in a vehicle, roll rate sensor. Among the models found in analytical redundancy, an Eigen structure assignment is utilized in this work because it provides a higher level of robustness than other models. Another strategy mentioned in [151] for fault detection in ball bearings uses the three-phase induction motor. This approach uses PCA and Fisher's Linear Discriminant Analysis (LDA) for dimensionality reduction. Furthermore, it uses random tree algorithms and a C4.5 decision tree algorithm to classify faulty and non-faulty data. In [95], a genetic algorithm approach is used for detecting faults in analog circuits. The parameters used for this process are obtained from the transfer function of the circuit under test and could detect faults efficiently with a high degree of accuracy.

In the robotics field, machine learning techniques are becoming popular towards handling fault diagnostic issues. In [152], the authors proposed an SVM method for fault diagnosis in Autonomous Underwater Vehicles (AUV). The algorithm was specifically designed to detect any actuator faults in the robot, where the interpolation of the unknown actuator fault is accomplished via a Radial Basis Function (RBF). The case study indicates that this approach can detect faults accurately. A fault diagnosis approach for robotic manipulation using a Linear SVM is proposed in [100].

The robotic manipulator used an external thread fastening application, which provided two feature parameters; namely, estimated insertion length and maximum reaction force. The result indicates that an accuracy of above 90% was achieved when evaluating the performance of the classifier. In [143], SVM and Cuckoo Search (CS) algorithms are used for fault diagnosis in a hydraulic system equipped with quadruped robots. The Rough set is used in the dimensionality reduction of the features. The proposed method, which uses SVM along with a CS algorithm and a rough set, provides a higher accuracy than the traditional SVM model. In [101], SVM is used for fault diagnosis in and analysis of the control software of mobile robots. The operation parameter of the robot under various scenarios is used to train and test the SVM classifier. Even though numerous

studies apply machine learning techniques for fault diagnosis in robotic systems, these works are often limited to fixed-morphology robotic systems.

This article reports an application of a machine learning approach for fault diagnosis of a reconfigurable robot that assumes multiple morphologies. The main departure from the state-of-the-art in this area is that a large set of fault types are generated and handled due to shapeshifting characteristics and

extended locomotion gaits in the case of reconfigurable robots. Our aim is to build reconfigurable robots that can autonomously detect and overcome faults by changing their morphologies and any associated locomotion modes. There have been many advances in the field of reconfigurable robotics over the past decade. The reconfigurable robot presented in [153] transforms itself from a four-wheeled driving mechanism to a two-legged walking mechanism having two degrees of freedom, with each leg controlled using the same set of actuators. The legged mode in this reconfigurable robot would provide it with the capability to get past any obstacles, while the wheeled mode would often be used to travel faster and provide better efficiency. Another research study, detailed in [154], discusses system modeling issues for an interconnected underwater reconfigurable robot. Hydrostatic and hydrodynamic tests were used to develop a dynamic model for the robot of interest. A simulation environment was proposed for a modular and reconfigurable robotic manipulator in [148]. This work validated the use of the proposed simulator for the testing of newly designed robots using the automatically generated codes for kinematics and a controller. The robot proposed in [155] can crawl, jump, travel through rough terrain, and get past obstacles in its path. Experimental results show that it moves at a rate of 260 mm per 10 s, can climb a sliding surface with a slope of 20 degrees, and can jump 80 mm. Even though a vast amount of literature on reconfigurable robotics is available, they are often limited to mechanism designs with primitive pointers of fault diagnosis. Additionally, none of the work on reconfigurable robotics deals with the application of machine learning approaches to the fault diagnosis problem, presenting numerous opportunities for research and development. Fault diagnosis is a critical feature required for a reconfigurable robot, as it provides the knowledge of optimal morphology to compensate for a fault.

In this chapter, we present an SVM-based fault diagnostics system for our bioinspired reconfigurable robot, Scorpio. The diagnostic system detects and classifies faults for crawling and

rolling locomotion modes. Specifically, we classify between faulty and non-faulty conditions across nine different locomotion gaits, each with rolling and crawling modes, over three different speeds. We extract statistical features, including the mean, standard deviation, variance, root mean square, peak-to-peak value, root sum of squares level, maximum value, and diagonal sum, from the Inertial Measurement Unit (IMU) data onboard the robot platform. The selected features were used as input to the SVM, for classification purposes, between faulty and non-faulty conditions. Moreover, we tested the algorithm using two different types of SVM. Each classifier's performance was compared in the context of fault detection for our Scorpio robot.

5.1 Scorpio robot: System Overview

Our experiments presented in this thesis involving fault diagnostics using SVM were performed on Scorpio, a rolling-crawling reconfigurable urban reconnaissance robot. With the increased applications of the robotics in urban reconnaissance missions, a robot that can reconfigure itself to multiple morphologies and realize an extended set of locomotion modes are highly desired. To this end, we looked to nature to extract biological principles that enable a robot to assume multiple morphologies and significantly extend its locomotion capabilities beyond conventional means. In this work, we are developing a novel class of bio-inspired self-reconfigurable robot deriving inspiration from Cebrennus rechenbergi, a species of huntsman spider inspired design approaches to achieve rolling-crawling-climbing locomotion [98,156-158]. This spider is found in sand desert Erg Chebbi in southeastern Morocco. This spider can crawl in any directions like any other spider, but in addition to this, when threatened, it can also roll in different directions. This is one of the few species that can roll by creating rapid flicflac movements using its legs. An interesting characteristic of this spider is that it does not require a slippery or sloped surface to begin the rolling motion. Instead, it can transform itself into a rolling position on the level ground and start rolling using its flicflac movements. An interesting characteristic of this spider is that it does not require a slippery or sloped surface to begin the rolling motion. Instead, it can transform itself into a rolling position on level ground and start rolling using its flip-flop movements. Discovery of this spider has also inspired a group of other researchers to develop robotic platforms based on the Cebrennus rechenbergi spider [159].

An extended set of locomotion and behavioral gaits have been synthesized from an exhaustive study of gaits in Cebrennus Rechenbergi spiders, modeling, simulation and synthesis

with real hardware. Our fault diagnostics experiments included nine distinct locomotion gaits covering both crawling and rolling modes. The outer body of the robot is made using PLA plastic by 3D printing. The robot has four legs with each leg containing three servo motors providing it with three degrees of freedom to realize crawling and rolling motions. To reduce the weight and size of the robot, a motor with a dimension of 21.5 x 11.5 x 21.5mm and weight of 11.5 gram is chosen. In total, the robot consists of 12 JR ES376 servo motors with each requiring a voltage of 4.8 v. The Pololu Maestro servo controller is used to interface the 12 servo motors with a microcontroller and thereby providing the user with an option to control the motors. Figure 5.1 presents the CAD model of our Scorpio robot with actuator details.

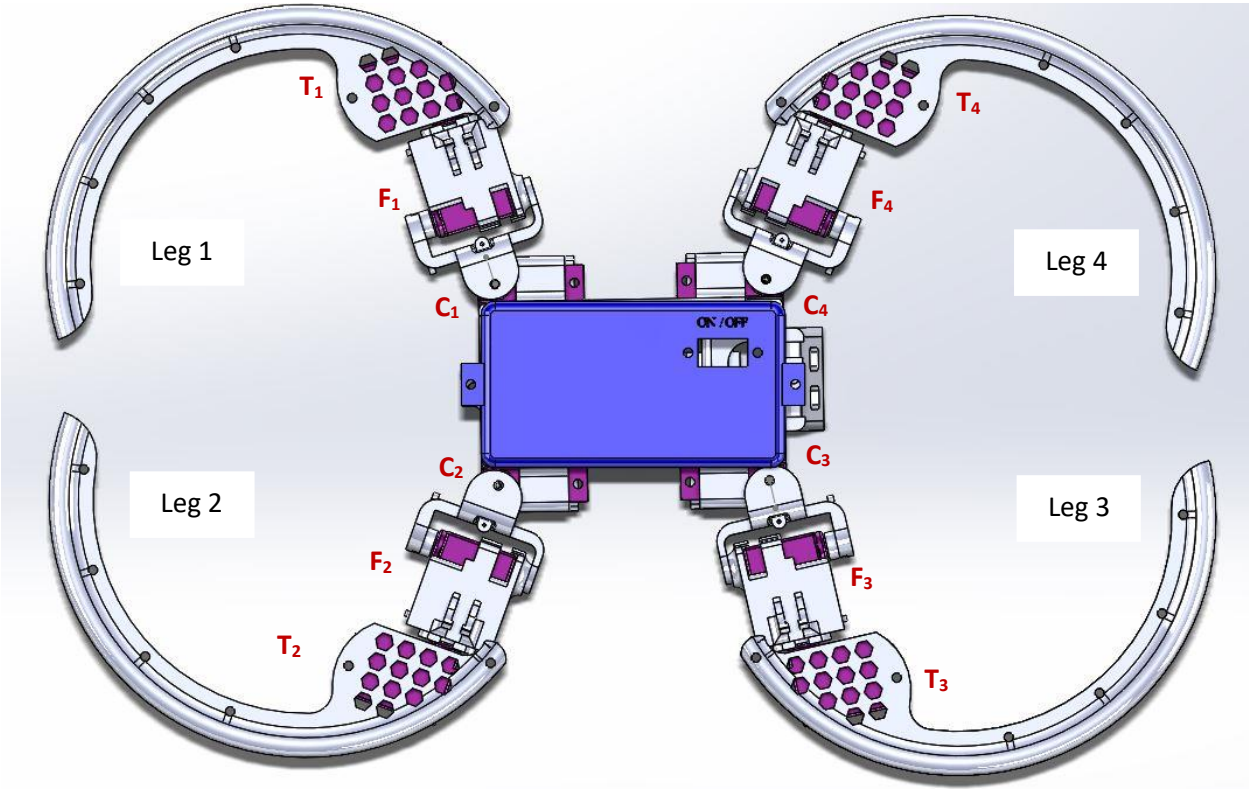


Figure 5.1 CAD Model of Scorpio Robot with Actuator Details

In addition to the actuators that enables robot locomotion, the platform is also equipped with an IMU sensor and a central microcontroller. The IMU sensor used is Pololu MinIMU-9 v3 which contains an L3GD20H 3-axis gyro, LSM303D 3-axis accelerometer, and 3-axis magnetometer. IMU sensor provides the rotational and translational data of the robot which enables the user to detect any anomalies in robot locomotion gaits. Arduino pro mini based on ATmega328 which

requires about 5 to 12 V is used as the central microcontroller. The hardware specification of our Scorpio robot is detailed in

Table 5.1 Hardware Specification of Scorpio Robot.

Table 5.1 Hardware Specification of Scorpio Robot

Hardware components	Specifications
Central Controller	Arduino Mini Pro with ATmega 328
Servo Motor	JR ES376
Servo Controller	Pololu maestro Servo controller
IMU	Pololu MinIMU-9 v3
Outer body	Polylactic acid or polylactide (PLA)
Power supply	Li-Po 1200mah
Communication Unit	Bluefruit EZ-Link

The IMU sensor constantly communicates the accelerometer and the gyroscope readings to Arduino pro mini microcontroller. The Arduino controller delivers the output control commands to the servo controller through a serial communication protocol. The Pololu Maestro servo controller is capable of driving up to 18 servo motors. The robot can be operated by a wireless device enabled by an onboard Bluefruit EZ-Link transceiver. The overall system architecture of the Scorpio robot is shown in Figure 5.2.

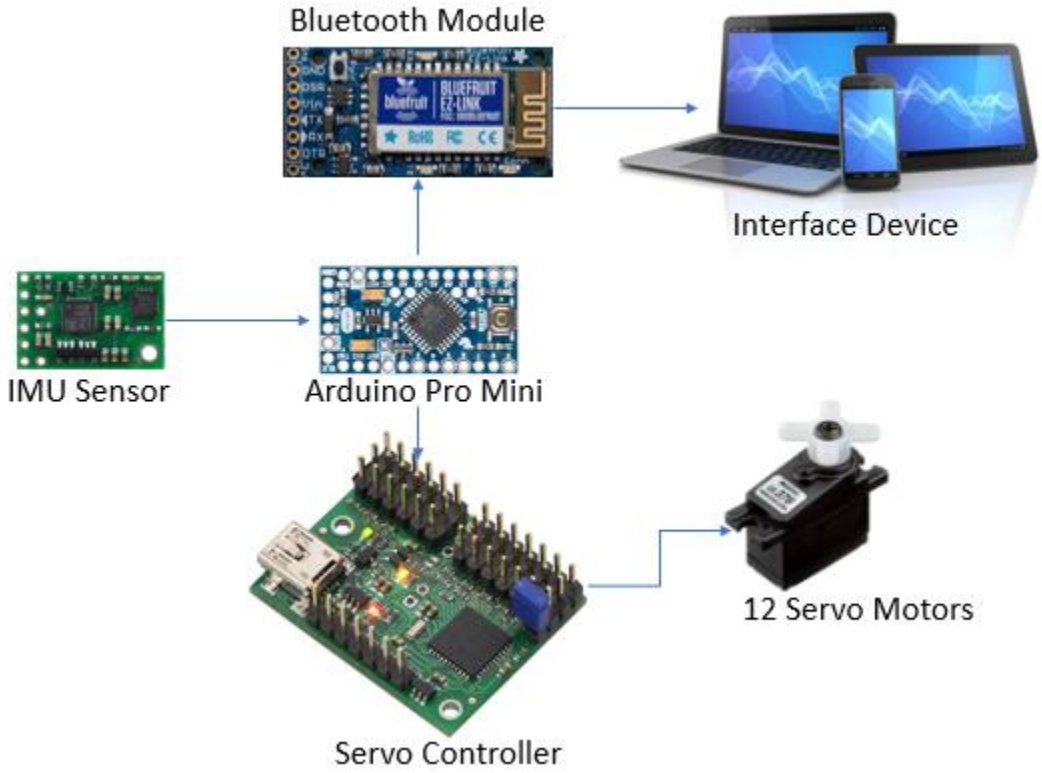


Figure 5.2 System architecture of the Scorpio robot

5.1.1 Crawling Locomotion

The crawling locomotion gait is achieved by simultaneous rotation of two motors about roll and yaw axis. For the forward or backwards crawling, the fem motor is used to lift the leg and provide the roll movement, and the Cox motor is used to push the leg forward or backwards and provide a yaw movement. To crawl right, the fem motor is used to give the roll movement, and the Cox motor will be moving the leg in the clockwise direction whereas, in the case of crawl left, the Cox motor will be moving the leg in the anti-clockwise direction. Similarly, a set of eight crawling gaits has been developed for our Scorpio robot. Only the Cox and fem motors are used in all the crawling locomotion gaits.

5.1.2 Rolling locomotion

Crawling is used as a default locomotion gait in our Scorpio robot. For the robot to roll, it initially has to transform from crawling position to rolling position. Rolling locomotion offers superior locomotion performance while traversing on completely flat terrain, over slopes and staircases. The transformation is done by shifting the leg 3 and leg 4 upside down and forming a

circle along with leg 1 and leg 2. Along with fem and Cox motors, T_3 and T_4 are also used to turn the leg 3 and leg 4 for attaining the rolling position. In the rolling position, two legs will be in contact with the ground at any particular time. The motor on these two legs will be actuated to attain the rolling forward or backward locomotion. Once the robot has attained one-half of the rolling, the other two legs will be in contact with the ground which will be actuated to achieve the next half of the rolling. Only fem motors are used to achieve the rolling locomotion. Fig Figure 5.3 Crawling and Rolling morphologies of our Scorpio robot

presents the crawling and rolling morphologies of our Scorpio robot.

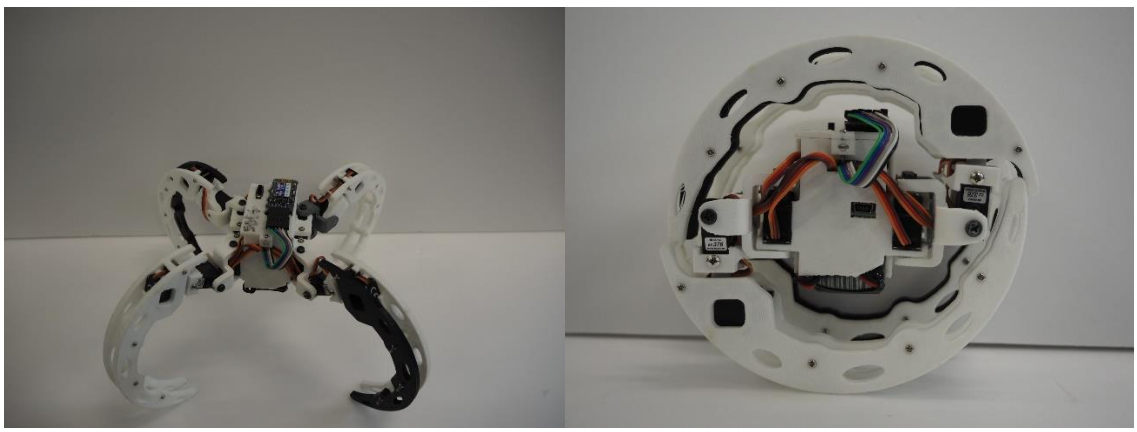


Figure 5.3 Crawling and Rolling morphologies of our Scorpio robot

5.2 Feature extraction

To perform the classification, we require the statistical features that explain the characteristics of the robot in case of faulty and non-faulty conditions. In this study, we take the gyroscope and accelerometer readings from the IMU chip. From the obtained readings, we extract the statistical features like mean, standard deviation, variance, root-mean-square level, peak to peak, root sum of squares level and maximum.

(a) **Mean:** Mean is the measure of the average of all the IMU data given at a particular instance. The mean of all the instance of IMU data is calculated and used as one of the statistical features.

$$\text{Mean} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{5.1}$$

Where,

x represents the IMU data for an instance

N accounts for the number of IMU data in that instance.

In our case, N = 7.

Standard deviation: Standard deviation is the measure of how spread out each of the IMU data is from each other. It will also be able to gauge the difference of each of the IMU data from its mean. It calculates the standard deviation of IMU data obtained from each of the instances.

$$\text{Standard deviation} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{5.2}$$

Where,

μ is the mean of the IMU data for an instance

x represents the IMU data for an instance and N = 7.

(b) **Variance:** This measure represents the variance in the obtained IMU data at an instance. It is the squared standard deviation of the IMU data from its mean.

$$\text{Variance} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{5.3}$$

Where,

μ is the mean of the IMU data for an instance

x represents the IMU data for an instance and N = 7.

(c) **Root Mean Square:** Root Mean Square also referred as quadratic mean is the square root of the arithmetic mean of the squared sample IMU data. It represents the magnitude or the typical size of the obtained IMU data for each instance.

$$\text{Root Mean Square} = \sqrt{\frac{\sum_{i=1}^{N}(x_i)^2}{N}} \tag{5.4}$$

Where,

x represents the IMU data for an instance and N = 7.

(d) **Peak to peak value:** Peak to peak value calculates the difference between the maximum IMU value and the minimum IMU value in an instance. The maximum IMU value at an instance is named as crest, and the minimum IMU value at an instance is named as trough

Peak to peak value = crest – trough (5.5)

(e) **Root sum of squares level:** The root sum of squares is calculated by taking a square root of the sum of the squared IMU data obtained at an instance.

Root sum of squares level = $\sqrt{\sum_{\mathbf{i=1}}^{\mathbf{N}} |\mathbf{x_i}|^{\mathbf{2}}}$ (5.6)

Where,

x represents the IMU data for an instance and N = 7.

(f) **Maximum Value:** This feature represents the highest value of the data obtained from IMU at a particular instance.

(g) **Sum:** This features adds up all the IMU data obtained from the IMU data at a particular instance.

Sum = $\sum_{\mathbf{i=1}}^{\mathbf{N}} \boldsymbol{x_i}$ (5.7)

Where,

x represents the IMU data for an instance and N = 7.

5.3 Support Vector Machine (SVM)

Once the statistical features are extracted from the IMU data, the next step would be to use the features to generate a classification strategy for the data. The Support Vector Machine is viewed as one of the most accurate classification algorithm used in supervised learning method of machine learning. In this study, we will be using support vector machine to implement the fault diagnosis algorithm. We will be using the MATLAB platform for writing the algorithm and will also be using LIBSVM (Library of SVM) package to implement SVM in MATLAB.

The LIBSVM requires the user to provide the data in a particular format. The user must provide the training data and training labels before the classifier is used to classify the test data. To

obtain the training and testing data, the set of features obtained from the last few cycles is used as the testing data, and the remaining data is used as the training data. In addition to that, we also created a training label indicating the true class of the training data in the same sequence. The flowchart indicating the fault diagnosis procedure is shown in Figure 5.4.

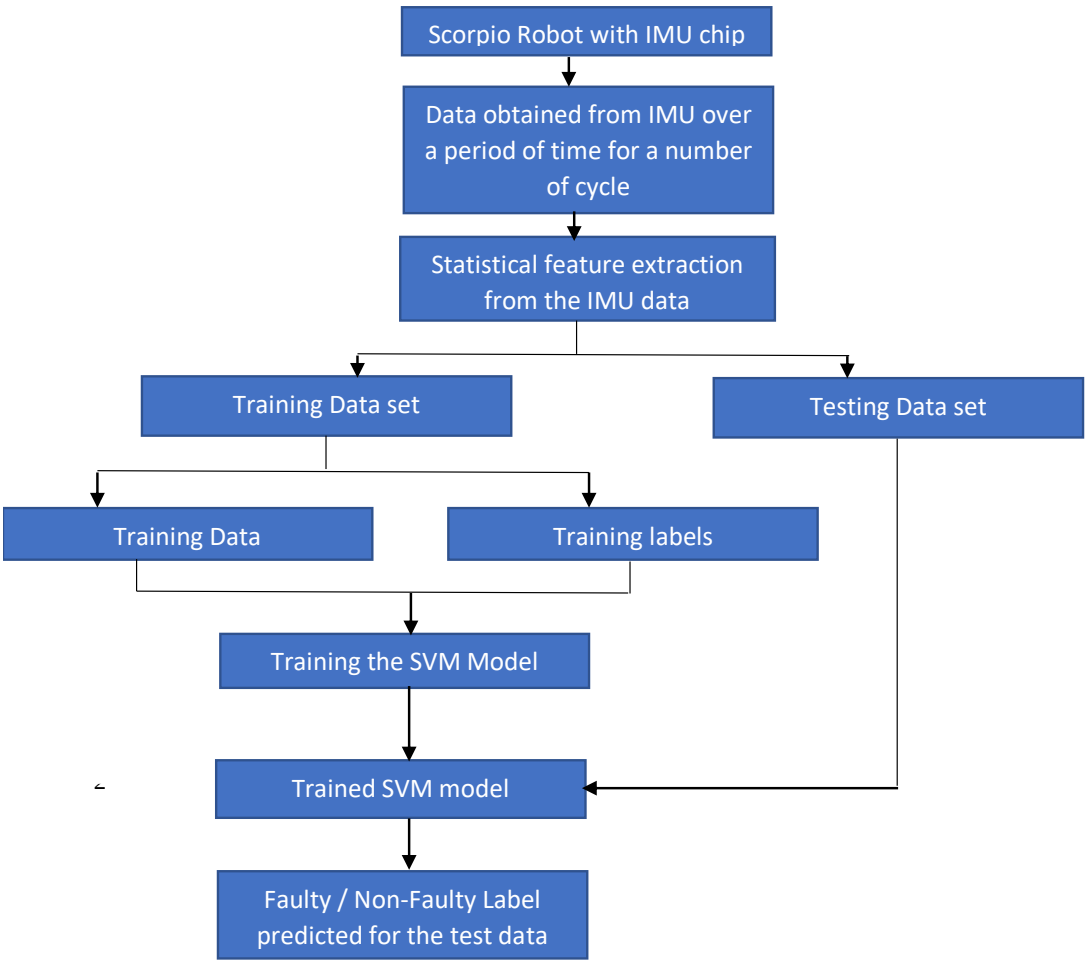


Figure 5.4 Flowchart representing the fault diagnostic procedure. Measurement Unit (IMU): Inertial Measurement Unit; Support Vector Machine (SVM): Support Vector Machine.

The goal of support vector machine is to figure out a hyperplane that separates training data belonging to two different classes. There can be some correct solution to a single classification problem, but an optimal solution for a classification problem using SVM is when a hyperplane separates the data from two different classes without any error and also possess the maximum separation margin from the closest training data in any classes. Such a hyperplane is called as a

maximum-margin hyperplane. This maximum separation margin between the two classes will make sure that the generalisation error, when tested with a testing data, is minimal which in turn provides the classifier with a better accuracy. The obtained training data may be linearly separable, or it could also be a non-linear data. To make the classification easier and avoid error, the training data is mapped into a higher dimensional space, and then the separating hyperplane is constructed in the higher dimensional space. To build an optimal separating hyperplane, two parallel hyperplanes are constructed which will separate the training data of the two classes. These two parallel hyperplanes are called as bounding planes and the area between the two hyperplanes is called as margin. The final separating hyperplane should lie in the margin area i.e. The area between the two parallel hyperplanes. The two-parallel hyperplane will contain some training data points lying on them, and these training data points which are called as support vectors will be useful in constructing the final separating hyperplane.

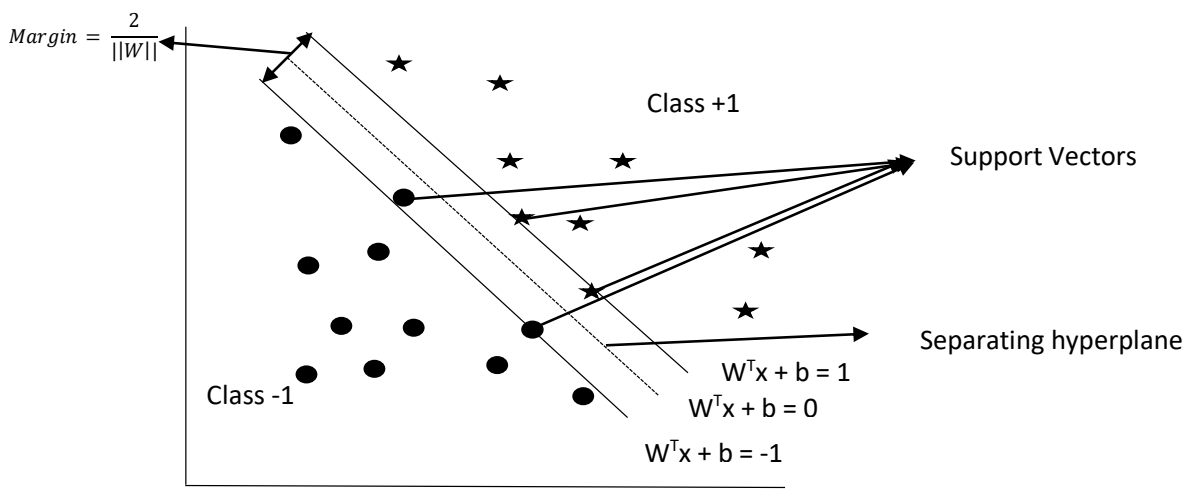


Figure 5.5 Standard Support Vector Machines Classifier.

Let us consider a two-class classification problem. We have a set of training data in the following format.

$$(x_i, y_i)$$

where i = 1,2,….n. n is the number of samples.

x_i is the training data set, and y_i is the label for the ith training data.

$x_i \in R^n$ and $y_i \in \{+1, -1\}$

The space between the two bounding planes is called as margin, and it is estimated to be $\frac{2}{||W||}$

Now, the SVM construct a hyperplane that separates the two class +1, -1. The equation of the hyperplane is given by

$$W^T x + b = 0 \tag{5.8}$$

The classification is done using the following strategy.

X_i belongs to class +1 if $W^T x + b > 0$

X_i belongs to class -1 if $W^T x + b < 0$

In addition to this, we can also check if the classification is proper by using the following equation. A proper classification must satisfy the following condition.

$$y(w^T x + b) \geq 1 \tag{5.9}$$

In this study, we have used two types of SVM namely C-Support Vector Classification and Nu or V- Support vector classification.

c – Support Vector Classification:

In c-SVC, C is the regularisation parameter, and the value of c ranges from zero to infinity.

Given a two-class classification problem (x_i, y_i) [160], Where x_i is the training data and $x_i \in R^n$, i = 1,2,…l.

y_i is the training label and $y_i \in \{+1, -1\}$, $y \in R^l$.

c-SVC solves the following primal optimisation problem

$$\min \quad \frac{1}{2} w^T w + C\sum_{i=1}^{l} \xi_i \tag{5.10}$$

subject to $y_i(w^T\varphi(x_i) + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$

The training data is x_i is mapped into a higher dimensional space by $\varphi(x_i)$. This is done to provide a good classification in case of nonlinear training data.

We also need to solve the following dual problem.

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \tag{5.11}$$

subject to $y^T \alpha = 0$ and $0 \leq \alpha_i \leq C$.

Where e = [1, ... , 1]

Q is a positive semidefinite matrix and $Q_{ij} = y_i y_j K(x_i, x_j)$

The decision function obtained is

$$\text{sgn}(w^T \varphi(x) + b) = \text{sgn}(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b) \tag{5.12}$$

v – Support Vector Classification or nu-Support Vector Classification:

In this Support vector classification, the variable v or nu is taken as the regularisation parameter, and it ranges from 0 to 1.

Given a two-class classification problem (x_i, y_i)

Where x_i is the training data and $x_i \in R^n$, i = 1,2,…l.

y_i is the training label and $y_i \in \{+1, -1\}$, $y \in R^l$.

v-SVC solves the following primal optimisation problem

$$\min_{w, b, \xi, \rho} \frac{1}{2} w^T w - v\rho + \frac{1}{l} \sum_{i=1}^{l} \xi_i \tag{5.13}$$

subject to $y_i(w^T \varphi(x_i) + b) \geq \rho - \xi_i$ and $\xi_i \geq 0$, $\rho \geq 0$.

The dual problem in v-SVC is

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha \tag{5.14}$$

subject to $y^T \alpha = 0$, $0 \leq \alpha_i \leq 1/l$ and $e^T \alpha \geq v$.

Where $Q_{ij} = y_i y_j K(x_i, x_j)$

The final decision function is

$$\text{sgn}(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b) \tag{5.15}$$

In addition to using two different types of SVM, we have used two different kernels in each type of SVM. The two kernels used are linear and radial basis function.

Linear Kernel:

The linear kernel is used mostly when the training and testing data is linearly separable. The data is linearly separable if there exists a single decision boundary line that could provide the necessary classification for the data.

The linear kernel is as follows.

$$K(x_i, x_j) = x_i^T x_j \tag{5.16}$$

Radial Basis Function Kernel:

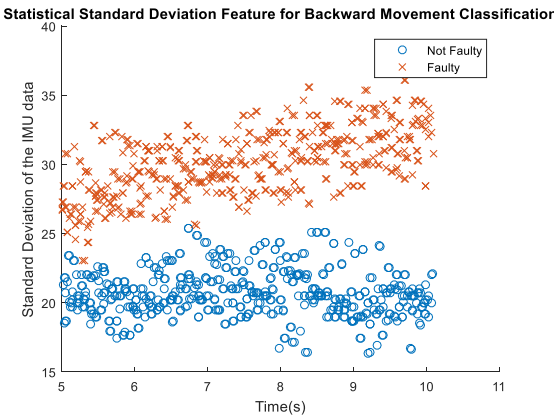
If the data is nonlinear, then the radial basis function kernel is a better option. This kernel maps the data into a higher dimensional space, and therefore it could provide a better classification for the nonlinear data than the linear kernel.

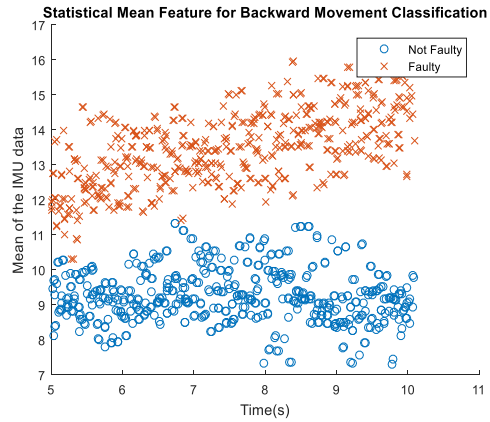
The radial basis function kernel is as follows.

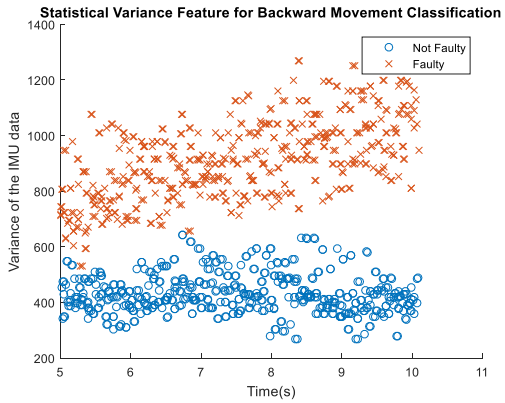
$$K(x_i, x_j) = \exp(-\Upsilon \| x_i - x_j \|^2), \Upsilon > 0 \tag{5.17}$$

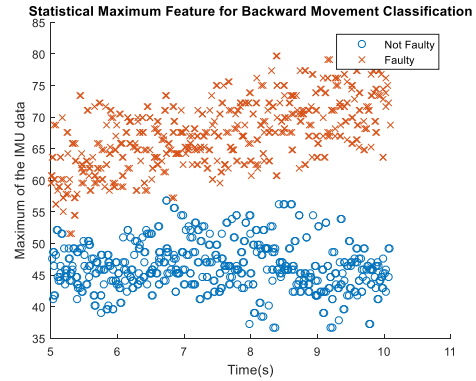
5.4 Experimental setup:

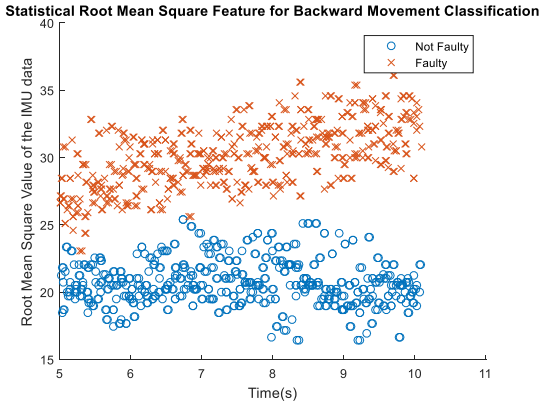
The experiment conducted in this study required the features that provide information related to the faulty and non-faulty working of the Scorpio robot for nine different gaits in three different speeds. For this purpose, the statistical features used for the SVM classification are extracted from the IMU data. The IMU data is obtained for all the nine gaits with three different speed. The speed is varied by the varying the degree of rotation per period in the servo motor. There are eight statistical features used to perform the classification. A scatter plot indicating the statistical features obtained from the IMU data of Backwards gait at 3 degrees per period speed of the servo motor is shown in Figure 5.6.

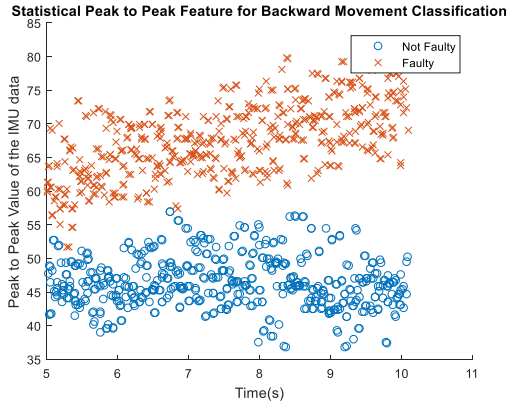

Statistical Standard Deviation Feature for Backward Movement Classificatior
Standard Deviation of the IMU data
Time(s)
Not Faulty
Faulty


Statistical Mean Feature for Backward Movement Classification
Mean of the IMU data
Time(s)
Not Faulty
Faulty


Statistical Variance Feature for Backward Movement Classification
Variance of the IMU data
Time(s)
Not Faulty
Faulty


Statistical Maximum Feature for Backward Movement Classification
Maximum of the IMU data
Time(s)
Not Faulty
Faulty


Statistical Root Mean Square Feature for Backward Movement Classification
Root Mean Square Value of the IMU data
Time(s)
Not Faulty
Faulty


Statistical Peak to Peak Feature for Backward Movement Classification
Peak to Peak Value of the IMU data
Time(s)
Not Faulty
Faulty

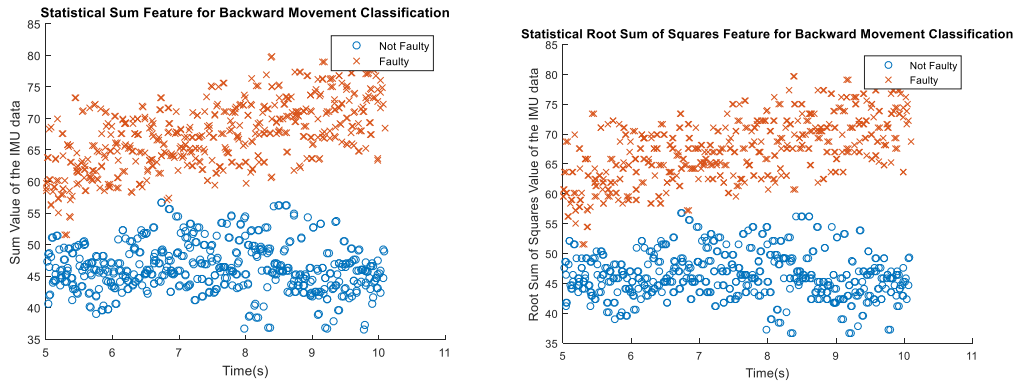
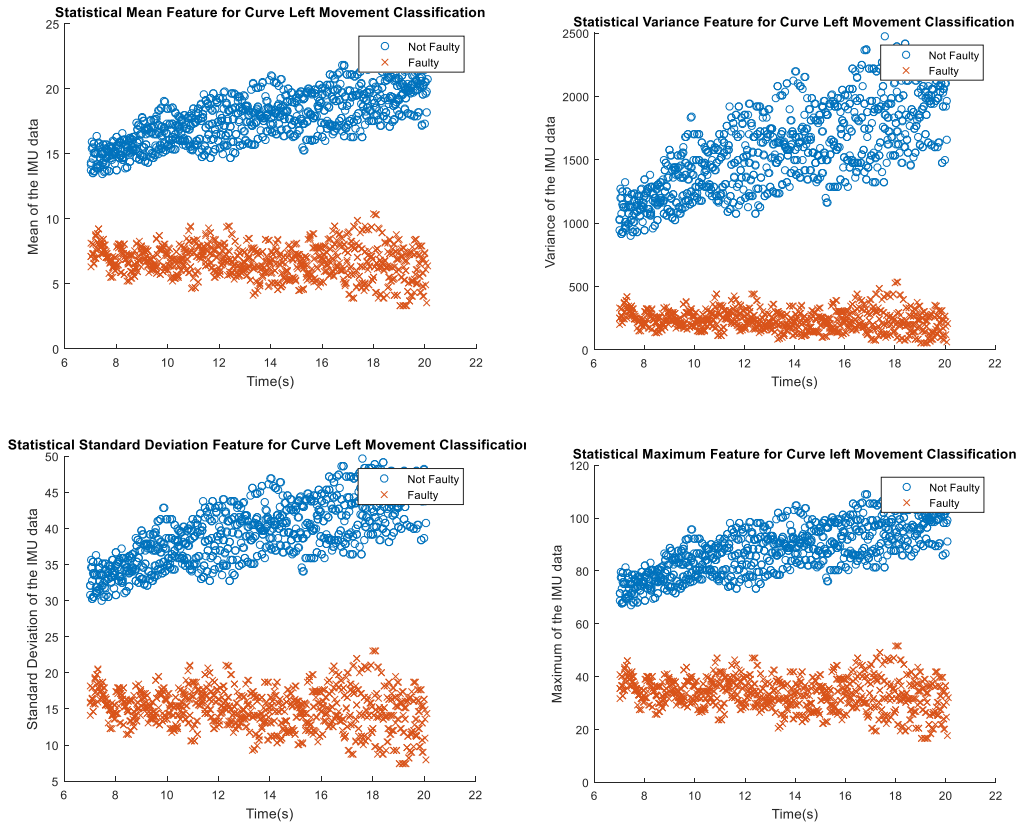


Figure 5.6 Statistical features used in the fault detection of Backwards gait at the rate of 3 degrees per period.

Another scatter plot indicating the statistical features obtained from the IMU data of Curve Left gait at 6 degrees per period speed of the servo motor is shown in Figure 5.7.



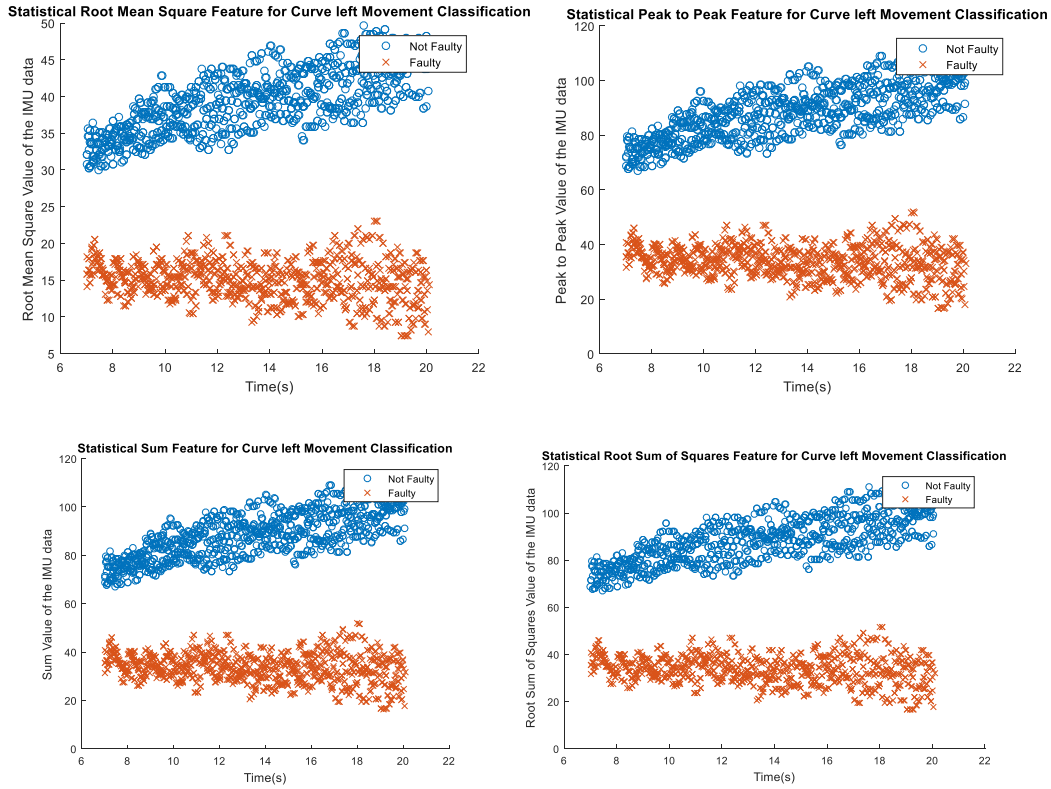


Figure 5.7 Statistical features used in the fault detection of Curve left gait at the speed of 6 degrees per period.

The plots in Figure 5.6 and Figure 5.7denotes the training set used for the classification. The plot denotes a clear separation between the faulty and non-faulty classes. The SVM classifier, when trained with the training data obtained from a particular gait, will be able to provide a useful classification for the corresponding gait. Depending on the gait, the period in which the IMU data is considered for obtaining the statistical feature varies. For example: In Figure 5.6, for the backwards movement gait, the IMU data is considered in the period between 5 to 10 seconds whereas In Figure 5.7, for curve left the movement, the IMU data is considered in the period between 7 to 20 seconds. A period in which there occurs an unexpected change in the movement of the Scorpio robot due to the fault is chosen for consideration and the IMU data from that period is used to extract the features. So, typically, this algorithm will be able to detect the fault in the robot by observing the motions of the Scorpio robot. Since the algorithm can detect the fault within few seconds of the sudden movement, it would be faster than any manual method.

In all gaits except rolling, the fault readings for the training and testing data is obtained by deactivating the following motors namely M21 (Back-Left Cox), M31(Back-Right Cox), M41(Front-Right Cox). In case of rolling, the fault data is obtained by providing disturbance in two of the fem motors namely M12(Front-Left Fem), M42(Front-Right Fem) such that it cannot provide enough push to make the robot roll and thereby trying to achieve the rolling motion by the movement of just two fem motors which are not powerful enough to provide a full 180-degree rotation for the robot. The non-fault readings are obtained when all the motors are working properly.

5.5 Experimental Results:

The classification is carried using the MATLAB software and Library of SVM support package for the nine gaits. The number of training and testing samples varies from one gait to another and also from one speed to another. The accuracy obtained from the SVM classification in nine gaits for three different speed is given in Table 5.2.

Table 5.2 Performance of the SVM classifier in nine gaits of the Scorpio robot

Type of Gaits		c-SVC		v-SVC or nu-SVC	
		Linear Kernel	Radial Basis Function Kernel	Linear Kernel	Radial Basis Function Kernel
Forward movement	3 degree/movement	98.2558%	97.093%	97.093%	95.9302%
	6 degree/movement	100%	100%	100%	100%
	12 degree/movement	100%	100%	100%	100%
Backward movement	3 degree/movement	100%	100%	99.4186%	100%
	6 degree/movement	100%	97.6285%	100%	100%
	12 degree/movement	100%	100%	100%	100%
Right Turn movement	3 degree/movement	98.3051%	98.7288%	98.3051%	98.3051%
	6 degree/movement	100%	97.0464%	100%	94.0928%
	12 degree/movement	91.6667%	92.2619%	95.2381%	91.6667%
Left Turn movement	3 degree/movement	100%	100%	100%	100%
	6 degree/movement	100%	100%	100%	100%

	12 degree/movement	95.9016%	96.7213%	95.9016%	95.082%
Curve Right movement	3 degree/movement	100%	98.8889%	98.5185%	98.1481%
	6 degree/movement	99.4048%	95.8333%	97.619%	100%
	12 degree/movement	100%	100%	100%	95.2381%
Curve Left movement	3 degree/movement	100%	100%	100%	99.7758%
	6 degree/movement	100%	100%	100%	95.9091%
	12 degree/movement	100%	98.3264%	100%	98.3264%
Shift Right movement	3 degree/movement	100%	100%	100%	100%
	6 degree/movement	100%	100%	98.8372%	98.8372%
	12 degree/movement	91.5663%	92.7711%	93.9759%	95.7831%
Shift Left movement	3 degree/movement	100%	100%	99.5025%	99.5025%
	6 degree/movement	98.8848%	98.1413%	92.9368%	93.6803%
	12 degree/movement	100%	100%	92.5743%	100%
Rolling movement	3 degree/movement	100%	100%	100%	100%
	6 degree/movement	92.4419%	95.9302%	100%	97.6744%
	12 degree/movement	95.2941%	94.1176%	96.4706%	92.9412%

The classification is done in both c-SVC and v-SVC classifier using linear and RBF kernel. The overall accuracy provided by these different classifiers is given in Table 5.3.

Table 5.3 Overall accuracy of the four classifiers.

Classifier Type		Overall Accuracy (%)
c-SVC	Linear Kernel	98.5823%
	Radial Basis Function (RBF) Kernel	98.2774%
v-SVC	Linear Kernel	98.3849%
	Radial Basis Function (RBF) Kernel	97.8109%

The linear kernel in c-SVC classifier gives the highest overall accuracy of 98.5823%. The reason behind this is that most of the statistical features used are linearly separable i.e. linear in nature. We can observe that the linear kernel in both c-SVC and v-SVC classifier provides a better accuracy than the corresponding RBF kernel.

The confusion matrix for all the classification is given in Table 5.4.

Table 5.4 Confusion matrix obtained for the classification

Type of Gaits		c-SVC		v-SVC or nu-SVC	
		Linear Kernel	Radial Basis Function Kernel	Linear Kernel	Radial Basis Function Kernel
Forward movement	3 degree/movement	$\begin{bmatrix} 83 & 3 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 83 & 3 \\ 2 & 84 \end{bmatrix}$	$\begin{bmatrix} 81 & 5 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 83 & 3 \\ 4 & 82 \end{bmatrix}$
	6 degree/movement	$\begin{bmatrix} 86 & 0 \\ 0 & 85 \end{bmatrix}$	$\begin{bmatrix} 86 & 0 \\ 0 & 85 \end{bmatrix}$	$\begin{bmatrix} 86 & 0 \\ 0 & 85 \end{bmatrix}$	$\begin{bmatrix} 86 & 0 \\ 0 & 85 \end{bmatrix}$
	12 degree/movement	$\begin{bmatrix} 69 & 0 \\ 0 & 68 \end{bmatrix}$	$\begin{bmatrix} 69 & 0 \\ 0 & 68 \end{bmatrix}$	$\begin{bmatrix} 69 & 0 \\ 0 & 68 \end{bmatrix}$	$\begin{bmatrix} 69 & 0 \\ 0 & 68 \end{bmatrix}$
Backward movement	3 degree/movement	$\begin{bmatrix} 86 & 0 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 86 & 0 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 85 & 1 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 86 & 0 \\ 0 & 86 \end{bmatrix}$
	6 degree/movement	$\begin{bmatrix} 126 & 0 \\ 0 & 127 \end{bmatrix}$	$\begin{bmatrix} 120 & 6 \\ 0 & 127 \end{bmatrix}$	$\begin{bmatrix} 126 & 0 \\ 0 & 127 \end{bmatrix}$	$\begin{bmatrix} 126 & 0 \\ 0 & 127 \end{bmatrix}$
	12 degree/movement	$\begin{bmatrix} 85 & 0 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 85 & 0 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 85 & 0 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 85 & 0 \\ 0 & 86 \end{bmatrix}$
Right Turn movement	3 degree/movement	$\begin{bmatrix} 115 & 4 \\ 0 & 117 \end{bmatrix}$	$\begin{bmatrix} 116 & 3 \\ 0 & 117 \end{bmatrix}$	$\begin{bmatrix} 115 & 4 \\ 0 & 117 \end{bmatrix}$	$\begin{bmatrix} 115 & 4 \\ 0 & 117 \end{bmatrix}$
	6 degree/movement	$\begin{bmatrix} 118 & 0 \\ 0 & 119 \end{bmatrix}$	$\begin{bmatrix} 118 & 0 \\ 7 & 112 \end{bmatrix}$	$\begin{bmatrix} 118 & 0 \\ 0 & 119 \end{bmatrix}$	$\begin{bmatrix} 118 & 0 \\ 14 & 105 \end{bmatrix}$
	12 degree/movement	$\begin{bmatrix} 84 & 0 \\ 14 & 70 \end{bmatrix}$	$\begin{bmatrix} 84 & 0 \\ 13 & 71 \end{bmatrix}$	$\begin{bmatrix} 84 & 0 \\ 8 & 76 \end{bmatrix}$	$\begin{bmatrix} 84 & 0 \\ 14 & 70 \end{bmatrix}$
Left Turn movement	3 degree/movement	$\begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$	$\begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$	$\begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$	$\begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$

	6 degree/movement	$\begin{bmatrix} 85 & 0 \\ 0 & 87 \end{bmatrix}$	$\begin{bmatrix} 85 & 0 \\ 0 & 87 \end{bmatrix}$	$\begin{bmatrix} 85 & 0 \\ 0 & 87 \end{bmatrix}$	$\begin{bmatrix} 85 & 0 \\ 0 & 87 \end{bmatrix}$
	12 degree/movement	$\begin{bmatrix} 55 & 5 \\ 0 & 62 \end{bmatrix}$	$\begin{bmatrix} 56 & 4 \\ 0 & 62 \end{bmatrix}$	$\begin{bmatrix} 55 & 5 \\ 0 & 62 \end{bmatrix}$	$\begin{bmatrix} 54 & 6 \\ 0 & 62 \end{bmatrix}$
Curve Right movement	3 degree/movement	$\begin{bmatrix} 136 & 0 \\ 0 & 134 \end{bmatrix}$	$\begin{bmatrix} 136 & 0 \\ 3 & 131 \end{bmatrix}$	$\begin{bmatrix} 132 & 4 \\ 0 & 134 \end{bmatrix}$	$\begin{bmatrix} 136 & 0 \\ 5 & 129 \end{bmatrix}$
	6 degree/movement	$\begin{bmatrix} 84 & 0 \\ 1 & 83 \end{bmatrix}$	$\begin{bmatrix} 77 & 7 \\ 0 & 84 \end{bmatrix}$	$\begin{bmatrix} 84 & 0 \\ 4 & 80 \end{bmatrix}$	$\begin{bmatrix} 84 & 0 \\ 0 & 84 \end{bmatrix}$
	12 degree/movement	$\begin{bmatrix} 84 & 0 \\ 0 & 84 \end{bmatrix}$	$\begin{bmatrix} 84 & 0 \\ 0 & 84 \end{bmatrix}$	$\begin{bmatrix} 84 & 0 \\ 0 & 84 \end{bmatrix}$	$\begin{bmatrix} 76 & 8 \\ 0 & 84 \end{bmatrix}$
Curve Left movement	3 degree/movement	$\begin{bmatrix} 223 & 0 \\ 0 & 223 \end{bmatrix}$	$\begin{bmatrix} 223 & 0 \\ 0 & 223 \end{bmatrix}$	$\begin{bmatrix} 223 & 0 \\ 0 & 223 \end{bmatrix}$	$\begin{bmatrix} 222 & 1 \\ 0 & 223 \end{bmatrix}$
	6 degree/movement	$\begin{bmatrix} 220 & 0 \\ 0 & 220 \end{bmatrix}$	$\begin{bmatrix} 220 & 0 \\ 0 & 220 \end{bmatrix}$	$\begin{bmatrix} 220 & 0 \\ 0 & 220 \end{bmatrix}$	$\begin{bmatrix} 202 & 18 \\ 0 & 220 \end{bmatrix}$
	12 degree/movement	$\begin{bmatrix} 119 & 0 \\ 0 & 120 \end{bmatrix}$	$\begin{bmatrix} 119 & 0 \\ 4 & 116 \end{bmatrix}$	$\begin{bmatrix} 119 & 0 \\ 0 & 120 \end{bmatrix}$	$\begin{bmatrix} 119 & 0 \\ 4 & 116 \end{bmatrix}$
Shift Right movement	3 degree/movement	$\begin{bmatrix} 151 & 0 \\ 0 & 150 \end{bmatrix}$	$\begin{bmatrix} 151 & 0 \\ 0 & 150 \end{bmatrix}$	$\begin{bmatrix} 151 & 0 \\ 0 & 150 \end{bmatrix}$	$\begin{bmatrix} 151 & 0 \\ 0 & 150 \end{bmatrix}$
	6 degree/movement	$\begin{bmatrix} 86 & 0 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 86 & 0 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 84 & 2 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 84 & 2 \\ 0 & 86 \end{bmatrix}$
	12 degree/movement	$\begin{bmatrix} 84 & 0 \\ 14 & 68 \end{bmatrix}$	$\begin{bmatrix} 84 & 0 \\ 12 & 70 \end{bmatrix}$	$\begin{bmatrix} 84 & 0 \\ 10 & 72 \end{bmatrix}$	$\begin{bmatrix} 84 & 0 \\ 7 & 75 \end{bmatrix}$
Shift Left movement	3 degree/movement	$\begin{bmatrix} 200 & 0 \\ 0 & 202 \end{bmatrix}$	$\begin{bmatrix} 200 & 0 \\ 0 & 202 \end{bmatrix}$	$\begin{bmatrix} 198 & 2 \\ 0 & 202 \end{bmatrix}$	$\begin{bmatrix} 200 & 0 \\ 2 & 200 \end{bmatrix}$
	6 degree/movement	$\begin{bmatrix} 131 & 3 \\ 0 & 135 \end{bmatrix}$	$\begin{bmatrix} 131 & 3 \\ 2 & 133 \end{bmatrix}$	$\begin{bmatrix} 115 & 19 \\ 0 & 135 \end{bmatrix}$	$\begin{bmatrix} 132 & 2 \\ 15 & 120 \end{bmatrix}$
	12 degree/movement	$\begin{bmatrix} 100 & 0 \\ 0 & 102 \end{bmatrix}$	$\begin{bmatrix} 100 & 0 \\ 0 & 102 \end{bmatrix}$	$\begin{bmatrix} 85 & 15 \\ 0 & 102 \end{bmatrix}$	$\begin{bmatrix} 100 & 0 \\ 0 & 102 \end{bmatrix}$
Rolling movement	3 degree/movement	$\begin{bmatrix} 50 & 0 \\ 0 & 53 \end{bmatrix}$	$\begin{bmatrix} 50 & 0 \\ 0 & 53 \end{bmatrix}$	$\begin{bmatrix} 50 & 0 \\ 0 & 53 \end{bmatrix}$	$\begin{bmatrix} 50 & 0 \\ 0 & 53 \end{bmatrix}$

	6 degree/movement	$\begin{bmatrix} 85 & 0 \\ 13 & 74 \end{bmatrix}$	$\begin{bmatrix} 78 & 7 \\ 0 & 87 \end{bmatrix}$	$\begin{bmatrix} 85 & 0 \\ 0 & 87 \end{bmatrix}$	$\begin{bmatrix} 81 & 4 \\ 0 & 87 \end{bmatrix}$
	12 degree/movement	$\begin{bmatrix} 84 & 0 \\ 8 & 78 \end{bmatrix}$	$\begin{bmatrix} 78 & 6 \\ 4 & 82 \end{bmatrix}$	$\begin{bmatrix} 84 & 0 \\ 6 & 80 \end{bmatrix}$	$\begin{bmatrix} 78 & 6 \\ 6 & 80 \end{bmatrix}$

Interpretation of Confusion matrix:

- In Machine learning, the performance of the classifier is evaluated using the confusion matrix which provides the information such as true positives, true negatives, false positives and false negatives.
- We get a 2x2 confusion matrix indicating the number of correct classification and misclassification of the classifier. The column indicates the instance in the predicted class while the row indicates the instance in the original class.
- The sum of diagonal elements indicates the number of correct classification and the sum of anti-diagonal elements indicate the number of misclassification.

5.6 Conclusion:

In this chapter, we have presented a novel approach towards fault detection in the crawling-rolling self-configurable Scorpio robot using Support Vector Machine. The fault detection is an important aspect for this robot so that it can reconfigure itself to compensate for any current fault. The fault detection was carried for a total of nine gaits at three different speeds. The eight statistical features are extracted from the IMU data which provides information regarding the movement of the Scorpio robot. The classification was carried out using two different SVM kernel function of two SVM model in Support Vector Machine. The SVM was able to provide an accuracy of above 90% for the test cases of all the nine gaits using two different SVM classification strategy and two different kernels. The overall accuracy of all the four classifiers is above 97% which is practically good. The linear kernel of c-SVC and nu-SVC classifier appears to provide a better overall accuracy when compared to the RBF kernel of c-SVC and nu-SVC classifier.

6 Morphological Fault Diagnosis in a Tetris-inspired Support Vector Machine-Based Reconfigurable Cleaning Robot

Lifestyle has nowadays become easier and smarter as the types and number of robots increase in number. Household floor cleaning and maintenance is a repetitive, indispensable and monotonous exercise in our day to day life. Researchers have in the past three decades significantly focused on the development and use of self-sufficient floor cleaning robot and robotic appliances of such nature are very critical in our day-to-day life. However, robotic appliance and application demands are projected to rise due to its potentialities in enhancing the quality and productivity in floor cleaning. Organizations such as Samsung, iRobot, Dyson and Xiaomi are heavily investing on the development of cleaning robots and availing them to the market as products. These robots' shape can generally be distinguished by 'D'- shaped and circular morphologies. They also work with such onboard sensors as RGB-D and LIDAR camera which facilitates autonomous navigation of cleaning robots when necessary. It is estimated that the rising cleaning robot demand will increase its market value to USD 4.34 Billion by the year 2023 [161]. Several studies are still underway with the aim of significantly improving the mechanism design, human-robot interaction, autonomy, robot teaming and benchmarking aspects of cleaning robots. By deeply assessing robots mechanical cleaning aspects, Ulrich [161] proposes that for efficient and smooth locomotion, robots should have tracked wheels. He established that square-shaped robots can efficiently access walls and corners. According to Takahisa [162], cleaning robots should be fitted with L-shaped legs for stair climbing and translational movements. On the other hand, Palacin in [163] validated the fitting of a cleaning robot with a roller brush instead of a vacuum for improved effectiveness. From the perspective of autonomy, a novel approach was introduced by [164] Tae. The approach implemented SLAM for floor-cleaning robots with limited sensor. The hierarchical proposal of RBPF SLAM for floor-cleaning robots is to compute and fix robots with two of each ultrasonic and infrared sensor. An autonomy approach was proposed by [165] Jeong based on CV-SLAM where one camera that faced upwards for natural landmarks' visual correspondence thus enhancing accurate and fast computation for navigation was used. A SLAM approach was also presented by [166] Ilari Vallivaara concerning the magnetic field. This approach uses the variations in the indoor magnetic field and quantifies the quality of the acquired map focusing on localization problems facing mobile robot floor-cleaning situation. This work also argues that there exist many methods

in implementing robot autonomy but reducing autonomy-related complexity through a suitable human interface is more important. Moreover, Zhao [167] proposed a mobile phone app which controls cleaning robot using WLAN based on human-robot interaction. This enables users to carry out basic operations such as setting time, screen printing and controlling speed of the robot. According to Haydar Sahin [168], increasing the level of autonomy in a mobile robot raises its complexity as well as the rate of fault thus making it hard to operate and maintain. He therefore presented graphical indicators showing the work of the robot that enables the user to easily control its actions.

The multitude of design dimensions in floor cleaning robot makes it critical to examine all possible aspects of the robot's performance for reproducibility and repeatability. An experimental protocol to benchmark the robot's navigation system has been defined by Sprunk [169] based on localization, mapping and planning. On the other hand, Rhim [170] identified the floor-cleaning robot's exclusive features and evaluated two cleaning robots that are commercially available. Similarly, Rainer [34] recommended a framework which aids in analyzing the SLAM approach results based on the mechanism for measuring the corrected trajectory error.

Although research done in this domain potentially highlights the benefits of floor-cleaning robots, they still suffer coverage performance issues due to fixed morphologies. Therefore, maximizing the coverage area requires the development of a reconfigurable mechanism as a viable solution to make the morphology of the robot change with respect to the environment.

A special interest in reconfigurable robotics domain over the last three decades has resulted in a vast amount of literature. Reconfigurable robots have been classified into three types [148]; Nested reconfigurable, inter-reconfigurable and intra-reconfigurable robots. Intra-reconfigurable robots are robots that by itself can change its morphology. An intra-reconfigurable Scorpio [24] robot as presented by Tan et al. can switch between wall climbing, crawling, and rolling morphologies. Reconfigurable Superbot [171] robots can change their size and shape autonomously to specifically meet operational demands. On the other hand, inter reconfigurable robots can be able to resemble and disassemble forming a global morphology. Under this classification lies Polybot [172] where each module can be able to connect to one or multiple modules. [173] have also introduced conro as the basic block reconfigurable robot. The building block in deployable reconfigurable robot shown in the module has a very huge potential. Nested type of reconfigurable

robots combines the concepts of both inter and intra reconfigurable robots to realize a wide range of morphologies. On the other hand, Hinged-Tetro [174] nested reconfigurable robot as presented by Tan et al. is capable of transforming between single unit morphologies, and with other homogeneous/heterogeneous robots can combine to generate global morphologies. We validated and put forward hTetro robot in our earlier work, a Tetris-inspired intra-reconfigurable robot that clean the floor may change its morphology to one of seven different shapes that assist to increase the coverage area. When commercially available cleaning robot that has fixed morphology was compared to hTetro, the area of coverage problem was found to be outperformed by the hTetro due to reconfiguring ability[175]. However, reconfigurable robots such as hTetro is to a great extent possesses complicated mechatronic systems where the hardware and software modules have a set of features which are largely diverging in nature. In uncertain environments during runtime, faults can be encountered in hTetro due to such complex settings. Morphological faults which involve partial or incomplete transformation has the potential to reduce performances and area coverage. Even a hTetro robot that is well designed in its lifetime could encounter morphological fault. A lot of literature has been written on reconfigurable robotics however research in the area of fault diagnosis is highly limited. For optimal deployment, hTetro robot should incorporate diagnostic features in order to compensate for morphological faults. To automate fault diagnosis, several strategies and approaches have been discussed. A discrete time approach to fault diagnosis is presented in [176] based on SVMs has been developed specifically for autonomous vehicles underwater. Algorithm has been developed in order to detect faults in robot through adoption of RBF. Diagnostic observer based on AUV's dynamic model is developed. Unknown dynamics, disturbances and unknown uncertainties with SVM adoption have been shown to be compensated. [177] presents a novel framework for hybrid fault diagnosis that uses a combination of Dempster-Shafer (D-S) and ML kernel (Mittag-Leffler kernel) SVM (support vector machine) to enhance consistency and safety in the performance of robots wheeled. The approach proposed utilizes sensor data that run under diverse conditions for sampling as well as constructs PCA (principal component analysis) model in order to extract fault features. The probabilistic SVM classifiers are trained by these fault feature vectors. To get final diagnosis results, BPAs are constructed based on probabilistic outcomes and are fused by Dempster Shafer fusion algorithm. [178] Presented a novel multi-fault categorization from rolling bearings element. Vibration signals here from rolling bearing is however measured and using EMD (empirical model decomposition) pre-processed.

Fault diagnosis in a rolling element bearing based on particle swarm optimization algorithm and WSVM is also studied. Likewise rolling element bearing fault diagnosis using vibration signals based on artificial neural networks is discussed in [179]. The algorithm consumes fewer amount of input features which set up a faster training with 100% success rate result in various bearing conditions. Fault diagnosis has been discussed in [180] concerning motor bearing rolling based on artificial neural networks. It has been established that neural networks may be used effectively via proper interpretation and measurement of motor bearing vibration.

Besides ANNs and SVMs, there are various alternative techniques that been studied for diagnosis of faults. Most of the industrial robot machines, these learning methods are used to completely automate fault diagnosis. In [181], Bernd developed an algorithm to diagnose faults in an industrial robot. This algorithm was based on a mathematical concept which is expressed as a non-linear differential equation. Fault diagnostic is identified using the anomalies experienced within the robots' inner physical coefficients such as damping, friction coefficients, gravitational torque, and inertia moments. In [182], Zhang applied genetic programming technique for fault diagnosis in a rotating machinery and compared the results with those of SVMs and ANNs, it was reported that genetic programming demonstrates execution which is better than or equivalents to the other techniques by taking significantly shorter training periods.

Various studies apply machine learning techniques for fault detection in robotic systems. Nevertheless, such works are only limited to the robots with fixed morphology. In this study, we present an application that utilizes machine learning methods for fault detection in robots that can re-configure to assume multiple morphologies. One of the ways in which this research differs from the state of the art is that it accounts for the fault types associated with multiple morphologies in a reconfigurable robot and associated locomotion gaits.

In this study, a SVM-based fault detection technique for a novel Tetris-inspired re-configurable floor vacuuming robot will be presented. The system for diagnosis is required to diagnose and categorize the morphological faults within the seven-real forms of hTetro platform. Morphological faults with partial or incomplete transformation in such robots are likely to significantly reduce the performance coverage area. Various statistical characteristics were extracted, such as the variance, mean, standard deviation, peak-to-peak values, root mean square, maximum value, diagonal sum, and root sum of square levels, from the attached motor data

incorporated within the robotic system. SVM applies the selected characteristics as input to differentiate non-faulty and faulty morphologies after every configuration. Two types of SVM were used to subsequently test the algorithm. Each classifier's fault diagnosis performance was validated using the hTetro robot, and each classifier's results were subsequently compared.

6.1 hTetro: Robot Architecture

The proposed fault detection technique was authenticated on the hTetro, which is a Tetris-inspired re-configurable floor-vacuuming robotic platform. This platform was developed based on the "polyominoes hinged dissection" theory. A hinged dissection refers to a geometric analysis approach whereby a planer structure undergoes dissection to form finite pieces which are connected using "hinged" points so that the structure transforms from one state to another by swinging continuously on the hinged points with the chain remaining intact [183]. A number of studies describe the hinged dissections. There are various notable efforts including changing the equilateral triangle's structure into a polygon [184], applying a 3-dimensional hinged dissection to generate a 3-dimensional polyhedron by unifying multiple rigid copies of a similar polyhedron [185], as well as developing different patterns of design by rearranging shapes [186]. In the robotics literature, the use of hinged dissection concept to develop a re-configurable robot known as a hinged-Tetro was explored in [139], and the results demonstrated that LLL and LLR hinged dissections can be applied to generate all the seven single-sided tetrominoes shapes. LLR hinged dissection was used to enable hTetro robot have the transforming ability, as demonstrated in Figure 6.1.

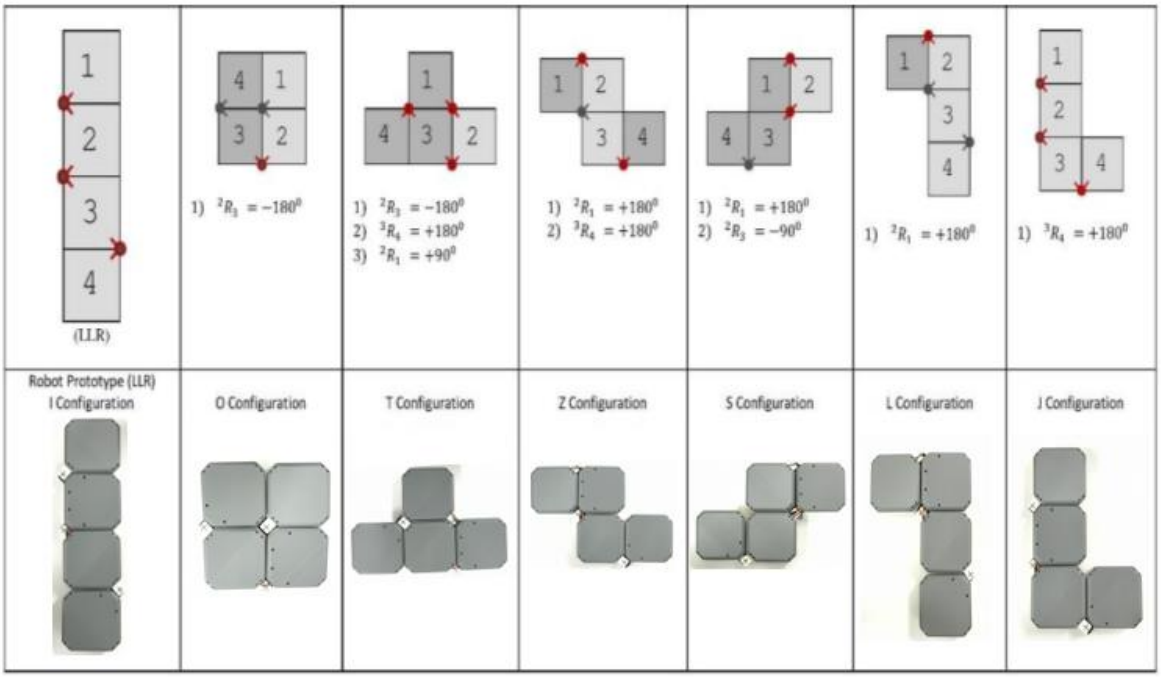


Figure 6.1 LLR hinged points of the hTetro robot.

6.1.1 Mechanism Design

An hTetro robot is made of four square blocks and three (LLR) hinged points of actuation for transformation. A comprehensive component list of the hTetro robot has been presented in Figure 6.2. Block 1 consists of the components that are required to conduct mobility functions, electronic peripherals are outfitted on block 2, and both block 3 and 4 comprise of accessories that work together with other electronic components to accomplish cleaning. The hTetro block has dimensions of 140 x 140 x 75 millimeter and a 4 mm wall thickness. These blocks are fabricated to have a honeycomb structure by using a PLA material to reduce its tensile strength. The hTetro robot consists of six DC motors, of which four are fitted on block 1 to perform the basic mobility functions. Also, the other two DC motors are mounted on Block 3. They are designed to improve the robot's stability during locomotion. In order to attain a seamless access within a specific area, the hTetro has Omni-directional potentials. Moreover, Blocks 4 and 2 are equipped with caster wheels to maintain the two block's position on the same plane. A customized vacuum module is incorporated into Block 3 and Block 4. The vacuum collects the dirt particles and carries the cleaning function of the robot. The duct and suction chamber were fabricated and designed in a way that eliminates dust or vacuum leakage during the process of cleaning. To achieve seven single

sided Tetris-transformations in hTetro, the hinges were equipped with three smart servos. Two of them are integrated on Block 2, and they serve as an anchor on which Blocks 1 and 3 are driven, while Block 4 is driven by the remaining servo. All the servo motors lock their positions by using their stall torques in order to maintain morphology of the hTetro robot in the entire navigation process. Block 4 is fitted with Raspberry PI. Raspberry PI manages the user interface communications and control actions of higher-levels. Block 2 uses Arduino Mega micro-controller (16-bit) as its low-level controller.

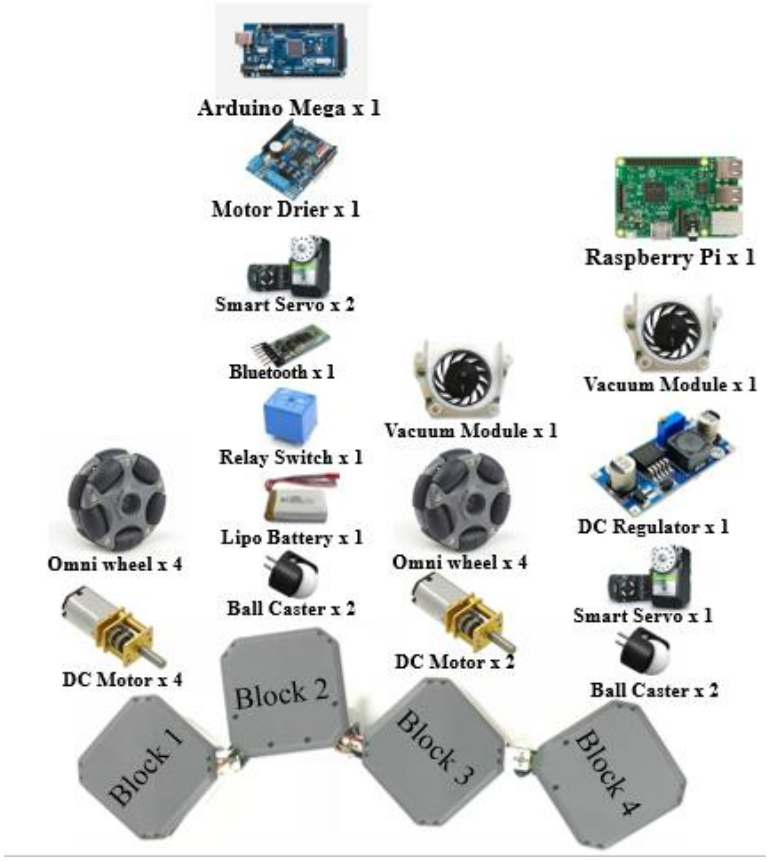


Figure 6.2 Components of the hTetro robot

The Arduino Mega 16-bit microcontroller handles the hTetro robot's transformation gait functions and locomotion. Raspberry PI sends action command signals to the microcontroller, which then executes the function by transmitting PWM motor primitives to reach the driver unit. A toggle switch connected to a 7.4voltage LiPo battery is used to power off and on the onboard electronics. Block 4 has a direct current step-down regulator which is used to control the input

voltage at 5v as required by the onboard Raspberry PI to operate. The vacuum module is regulated by an activated relay switch. The Raspberry PI receives user commands to regulate the device via a wireless Bluetooth transmission interface. A customized Android application that is user-friendly was developed, as demonstrated in Figure 6.3, in order to communicate effectively with an hTetro robot. Users have a number of options available to choose from in order to achieve transformation, locomotion, as well as cleaning functions. A control screen has arrow keys with which a user can convey navigational commands, ON/OFF buttons by which the user initiates /ceases the vacuuming functions, play and pause buttons to start or pause the current operations, and seven Tetris-shaped buttons to initiate shape-shifting actions.

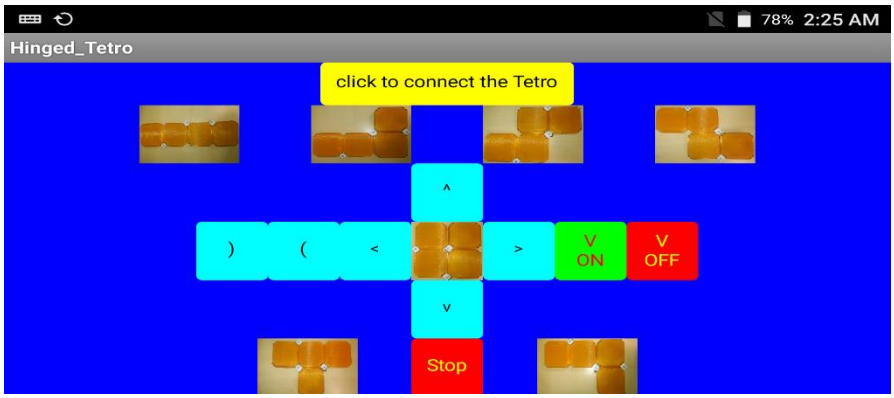


Figure 6.3 hTetro Interface.

6.2 Feature Extraction

To distinguish and further categorize the robot's non-faulty and faulty condition, statistical features were extracted out from the experimental data by means of the robot's IMU readings. The statistical aspects taken out from the acquired motor data comprised standard deviation, mean, variance, peak-to-peak, root-mean-square level, maximum value and root sum of squares level. The IMU data used for this analysis comprised seven values: three axes for the gyroscope, three axes for the accelerometer, and one axis for the magnetometer.

(a) Mean: Mean is the total of samples of IMU data divided by the samples amount. Every IMU data set mean was computed by means of Eq. (6.1).

$$\text{Mean} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{6.1}$$

where,

x represents the IMU data for an instance

N accounts for the number of IMU data in that instance.

In our case, N = 3.

(b) Standard deviation: Standard deviation is the gauge of dispersion degree present within the specified set of IMU information. In other terms, the standard deviation determines how far away every IMU data position was from its mean value. The formula employed to compute the standard deviation is specified in Eq. (6.2).

$$\text{Standard deviation} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{6.2}$$

where,

μ is the mean of the IMU data for an instance

x represents the IMU data for an instance, and N = 3.

(c) Variance: This is the squared value of the standard deviation of the IMU information which was computed by means of Eq. (6.3)

$$\text{Variance} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{6.3}$$

where,

μ is the mean of the IMU data for an instance

x represents the IMU data for an instance, and N = 3.

Root Mean Square: Root mean square is as well denoted as quadratic mean is the square root of the mean squared IMU data models. It symbolizes the level or the basic IMU data size for every

(d) data sample gathered. The technique for computing the root mean square is specified in Eq. (6.4).

$$\text{Root Mean Square} = \sqrt{\frac{\sum_{i=1}^{N}(x_i)^2}{N}} \tag{6.4}$$

where x represents the IMU data for an instance and N = 3.

(e) Peak-to-peak value: The peak-to-peak value computes the disparity between the minimum IMU value and the maximum IMU value at a specified moment. The utmost IMU value at a given time is the crest, and the least IMU value at a given moment is the trough. The peak-to-peak value was estimated by means of Eq. (6.5).

$$\text{Peak to peak value} = \text{crest} - \text{trough} \tag{6.5}$$

(f) Root sum of squares level: The root total of squares is computed by taking the square root of the total of the squared IMU data acquired for a specified case. The root total of squares level was computed by means of Eq. (6.6).

$$\text{Root sum of squares level} = \sqrt{\sum_{i=1}^{N} |x_i|^2} \tag{6.6}$$

Where x represents the IMU data for an instance and N = 3.

(g) Maximum Value: This feature symbolizes the greatest value of the IMU data acquired from the robot at a single incidence.

(h) Sum: This aspect is the summed-up value of the entire IMU data at a specified moment. The sum of the IMU data was calculated using Eq. (6.7).

$$\text{Sum} = \sum_{i=1}^{N} x_I \tag{6.7}$$

where x represents the IMU data for an instance, and N = 3.

6.3 Support Vector Machine (SVM)

After the features had been mined, a method of classification was created that could assist to distinguish between the non-faulty and faulty readings. The study employed the Library of SVM (LIBSVM) package together with MATLAB to build the machine learning model. Originally, LIBSVM is used to train the classifier by means of the training sample and its parallel training labels prior to the model is experimented through the testing samples. To this end, the acquired sample was separated into two sets. The initial set comprised most of the sample and was employed as the training model to prepare the classifier whereas the second set comprised the experiment

vectors that were used to authenticate the model accuracy. Figure 6.4 illustrates a flowchart of the process of fault classification.

The purpose of using a SVM is to establish the hyperplane that might separate the information belonging to diverse labels. Figure 6.5 shows the standard SVM classifier. A data set might have diverse likely classifications, and each of these categorizations could offer diverse solutions which consecutively present a varying accuracy level to the classifier. The aim of the study was to recognize the optimal solution that might present the utmost precision by employing a separating hyperplane. The separating hyperplane was the main categorization instrument applied within the SVM. The SVM produces the separating hyperplane that categorized data on diverse sides of the hyperplane such that it comprised diverse classes. This ideal hyperplane is identified as a maximum-margin hyperplane. The utmost margin of separation is the distance between the nearest data points belonging to two categories and the hyperplane. The greater the margin the lesser the likelihood of error while performing an assessment by means of the testing information. In turn, this improves the precision of the classifier. The training data could be linearly divisible or nonlinear. To enhance the precision of the classifier, the data points were plotted within the greater dimensional space. This made sure that the SVM might offer a good outcome regardless of whether the data was linearly divisible or not. Accordingly, the matching maximum-margin hyperplane was as well developed for the data points. Besides developing the maximum-margin hyperplane, two vaulting hyperplanes were as well developed. These two bounding hyperplanes were analogous to each other and to the utmost-margin hyperplane. The maximum-margin hyperplane ought to rest in the part between the two bounding hyperplanes. Some training data rest on the bounding planes and this teaching data is recognized as the support vector. The support vector served a decisive responsibility in constructing the concluding maximum-margin hyperplane.

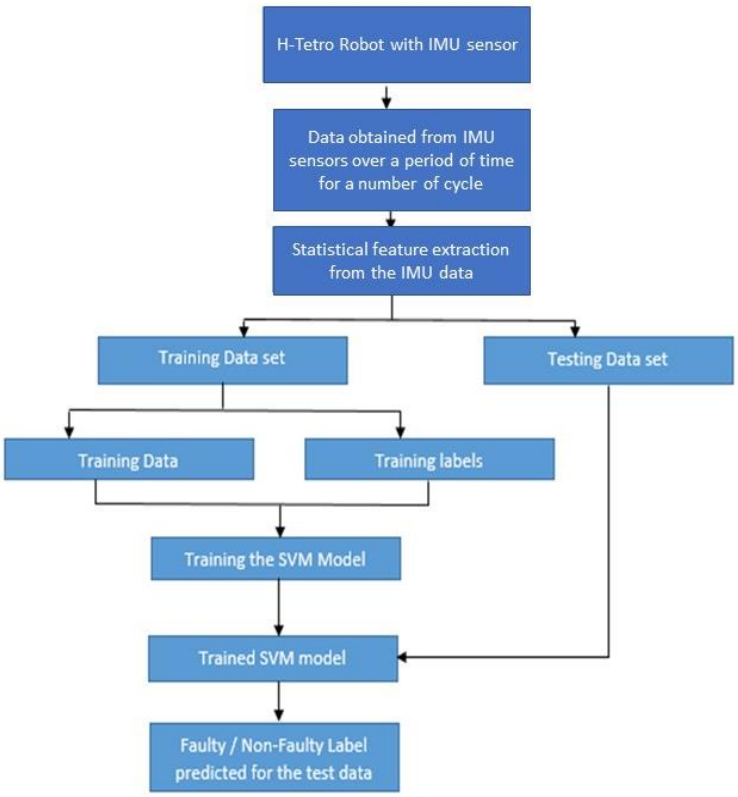


Figure 6.4 Flowchart of fault diagnosis procedure using SVM

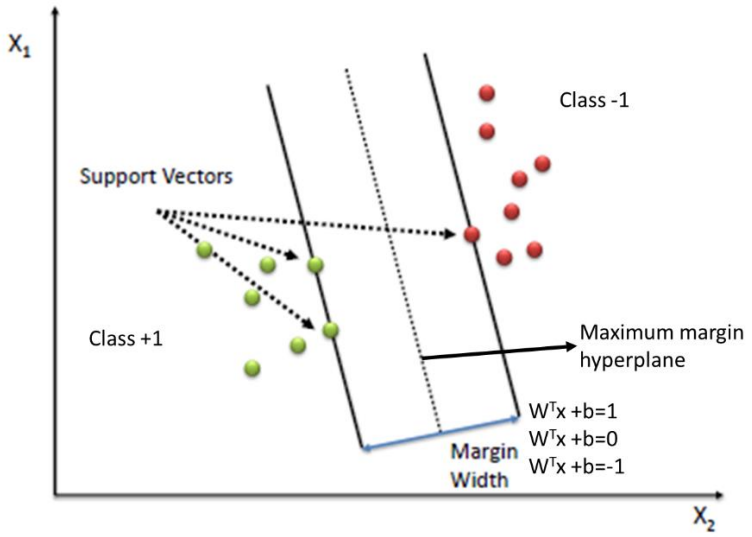


Figure 6.5 Support vector machine classifier

Considering a two-class classification strategy, this analysis presumed a set of training vectors within the subsequent (x_i, y_i)

where i = 1,2,….n. n is the number of samples.

x_i is the training vectors, and y_i is the label for the ith training data.

$x_i \in R^n$ and $y_i \in \{+1, -1\}$

The space between the two bounding planes was estimated to be $\frac{2}{||W||}$.

The objective was to employ the SVM to build a hyperplane that separated class +1 and class -1. The equation applied to calculate the hyperplane was specified by:

$$W^T x + b = 0. \tag{6.8}$$

The classification was conducted by means of the following tactics.

X_i belongs to class +1 if $W^T x + b > 0$.

X_i belongs to class −1 if $W^T x + b < 0$.

Additionally, the analysis confirmed whether the categorization was valid using the subsequent constraints. An authentic classification should satisfy the following specified constraints:

$$y(w^T x + b) \geq 1. \tag{6.9}$$

In this work, two types of SVM were applied: C-support vector categorization (c-SVC) and v-support vector categorization (v-SVC).

c-Support Vector Classification (c-SVC): In c-SVC, c is the parameter of regularization. The value, c, can range from zero to infinity. Taking into account a two-class categorization problem (x_i, y_i) [194], where x_i is the training vector, $x_i \in R^n$ for i = 1,2,…n, y_i, is the training label, and $y_i \in \{+1, -1\}$ for all $y \in Rl$, c-SVC solves the next primal optimization dilemma as follows:

$$\min_{w, b, \xi} \frac{1}{2} w^T w + C\sum_{i=1}^{l} \xi_I \tag{6.10}$$

subject to $y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$.

The preparation vector, xi, is mapped into a greater dimensional space by φ(xi). The intention of this is to produce a good categorization in the situation of nonlinear training data. Moreover, we need to resolve the subsequent dual problem:

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \tag{6.11}$$

subject to $y^T \alpha = 0$ and $0 \leq \alpha_i \leq C$.

$$\mathrm{sgn}(w^T\varphi(x) + b) = \mathrm{sgn}(\sum_{i=1}^{l} y_i\alpha_i K(x_I, x) + b) \tag{6.12}$$

Where e = [1, ... , 1]

Q is a positive semidefinite matrix, and Qij = yiyjK(xi, xj).

The decision function acquired is characterized as follows:

v-Support Vector Classification (v-SVC): In this classification strategy, the variable v ranges from 0-1 and is treated as the regularization factor. Taking into account a two-class categorization problem (xi, yi), where xi is the preparation data and xi ϵ Rn, i = 1,2,…l. yi is the preparation label, and yi ϵ {+1, −1}, y ϵ R^l, v-SVC solves the next primal optimization dilemma as follows:

$$\min_{w, b, \xi, \rho} \frac{1}{2} w^T w - v\rho + \frac{1}{l}\sum_{i=1}^{l} \xi_I \tag{6.13}$$

subject to $y_i(w^T\varphi(x_i) + b) \geq \rho - \xi_i$ and $\xi_i \geq 0$, $\rho \geq 0$.

The dual problem in v-SVC is as follows:

$$\mathrm{Min}_{\alpha} \frac{1}{2}\alpha^T Q\alpha \tag{6.14}$$

subject to yT α = 0, 0 ≤ αi ≤ 1/l and eT α ≥ v.

where $Q_{ij} = y_i y_j K(x_i, x_j)$.

The final decision function is as follows:

$$\mathrm{sgn}(\sum_{i=1}^{l} y_i\alpha_i K(x_I, x) + b) \tag{6.15}$$

Besides employing two diverse kinds of SVM, we used three diverse kernels for every kind of SVM. The two kernels are the polynomial and the linear, radial basis function. The linear kernel is most favourable in cases wherein the preparation and testing information is linearly divisible.

The data are linearly divisible if a distinct linear or a straight choice boundary line offers the essential separation for categorizing the data.

The linear kernel is characterized by:

$$K(x_i, x_j) = x_i^T x_j. \tag{6.16}$$

In cases where the data is nonlinear, the radial basis function kernel is a superior choice. This kernel maps the information into a higher dimensional space and thus offers a better categorization for nonlinear data compared to the linear kernel. The radial basis function kernel is differentiated as follows:

$$K(x_i, x_j) = \exp(-\Upsilon \parallel x_i - x_j \parallel^2),\ \Upsilon > 0. \tag{6.17}$$

The polynomial kernel employs three diverse kernel parameters, known as r, d, ϒ, for assessment. The polynomial kernel is specified by:

$$K(x_i, x_j) = (\Upsilon\, x_i^T x_j + r)^d,\ \Upsilon > 0. \tag{6.18}$$

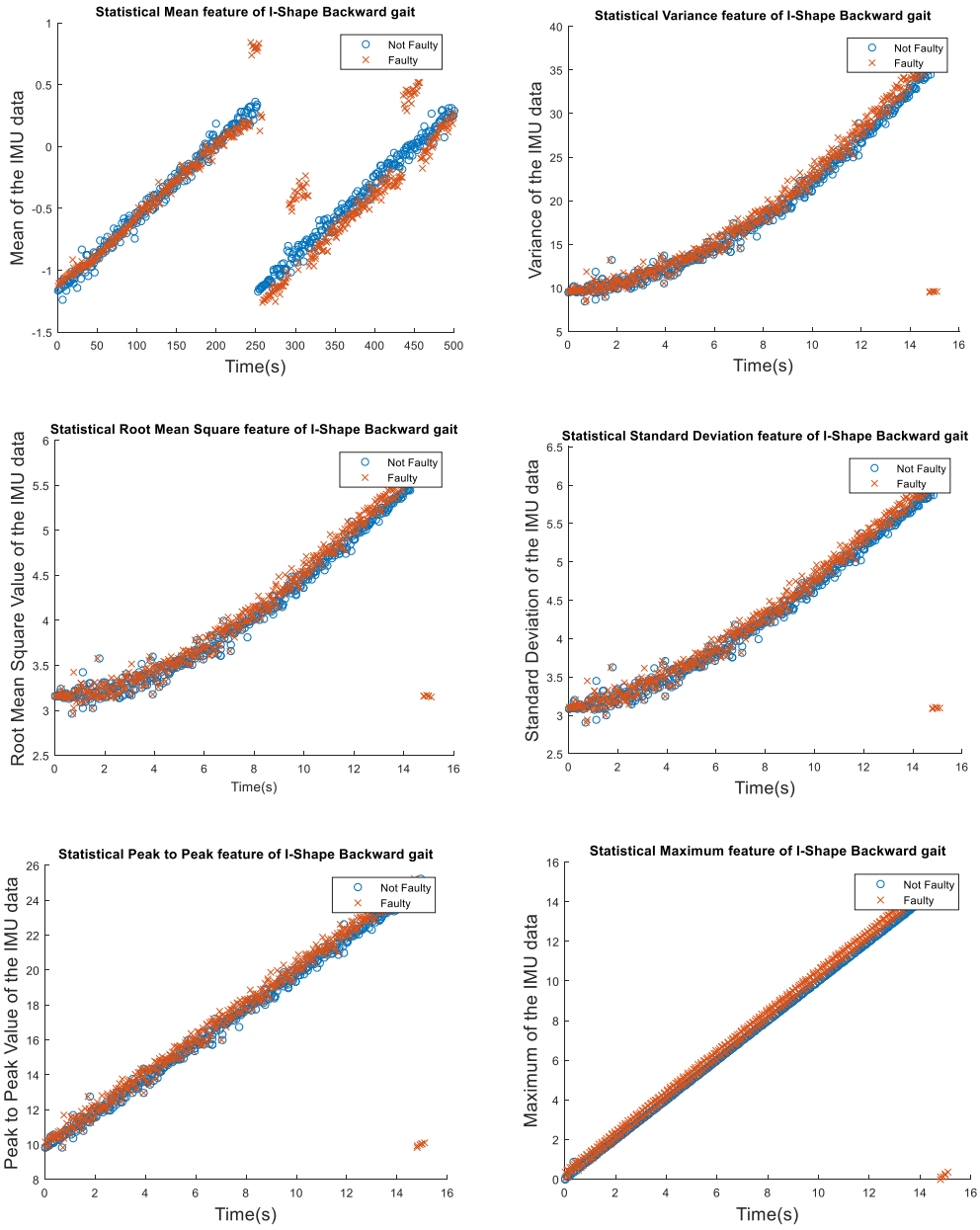
6.4 Experimental Setup

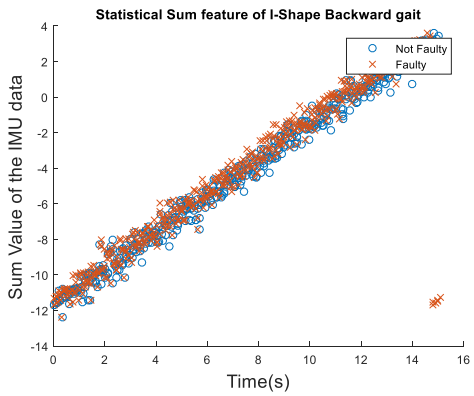
The experiments carried out in this thesis used statistical features that offered the decisive information needed to differentiate between non-faulty and faulty locomotion gaits in our hTetro robot. The IMU information was acquired for all seven morphologies of hTetro robot with two gaits considered for each form. Once the hTetro robot is initiated, it gets some readings to institute a baseline orientation wherein it anticipates both the pitch and the roll of the sensors to be zero. Following the startup routine, the program continually gets readings from the gyro, magnetometer, and accelerometer. The study employed a total of eight statistical aspects listed in Section 3, to carry out the fault categorization. Scatter plots showing the statistical features acquired from the IMU sensor for the I-locomotion gait are demonstrated in Figure Figure 6.6 Scatter plots indicate the statistical features used in fault detection of two shapes: (a) I-shape Backward locomotion gait; (b) S-shape Forward locomotion gait.

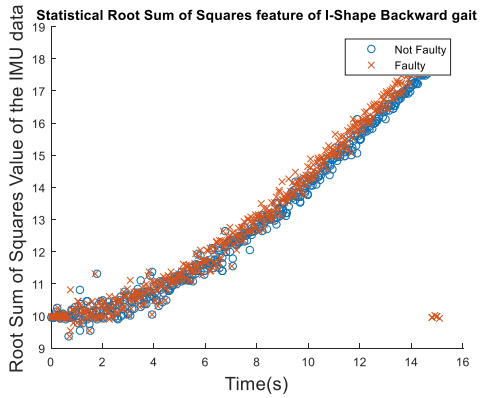
a, and the S-locomotion gait is shown in Figure Figure 6.6 Scatter plots indicate the statistical features used in fault detection of two shapes: (a) I-shape Backward locomotion gait; (b) S-shape Forward locomotion gait.

b.

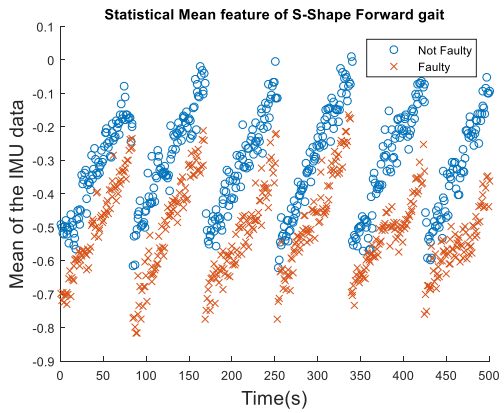
(a)

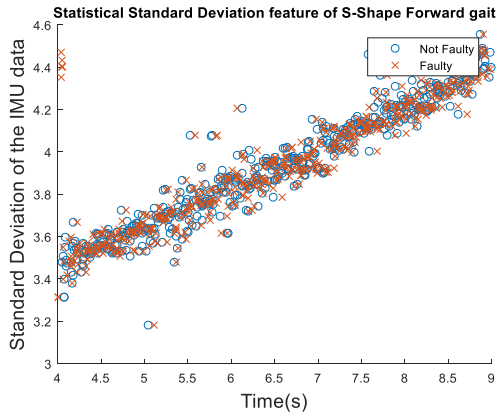

Statistical Mean feature of I-Shape Backward gait
Mean of the IMU data
Time(s)
Statistical Variance feature of I-Shape Backward gait
Variance of the IMU data
Time(s)
Statistical Root Mean Square feature of I-Shape Backward gait
Root Mean Square Value of the IMU data
Time(s)
Statistical Standard Deviation feature of I-Shape Backward gait
Standard Deviation of the IMU data
Time(s)
Statistical Peak to Peak feature of I-Shape Backward gait
Peak to Peak Value of the IMU data
Time(s)
Statistical Maximum feature of I-Shape Backward gait
Maximum of the IMU data
Time(s)
Not Faulty
Faulty

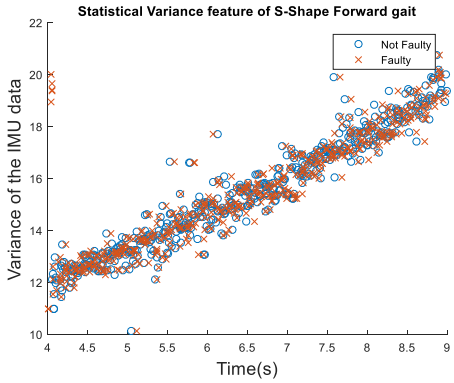

Statistical Sum feature of I-Shape Backward gait
Sum Value of the IMU data
Time(s)
Not Faulty
Faulty

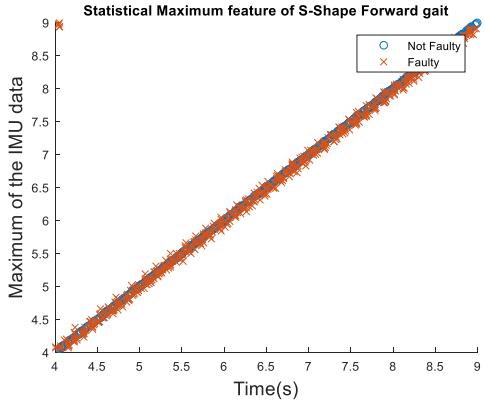

Statistical Root Sum of Squares feature of I-Shape Backward gait
Root Sum of Squares Value of the IMU data
Time(s)
Not Faulty
Faulty

(b)


Statistical Mean feature of S-Shape Forward gait
Mean of the IMU data
Time(s)
Not Faulty
Faulty


Statistical Standard Deviation feature of S-Shape Forward gait
Standard Deviation of the IMU data
Time(s)
Not Faulty
Faulty


Statistical Variance feature of S-Shape Forward gait
Variance of the IMU data
Time(s)
Not Faulty
Faulty


Statistical Maximum feature of S-Shape Forward gait
Maximum of the IMU data
Time(s)
Not Faulty
Faulty

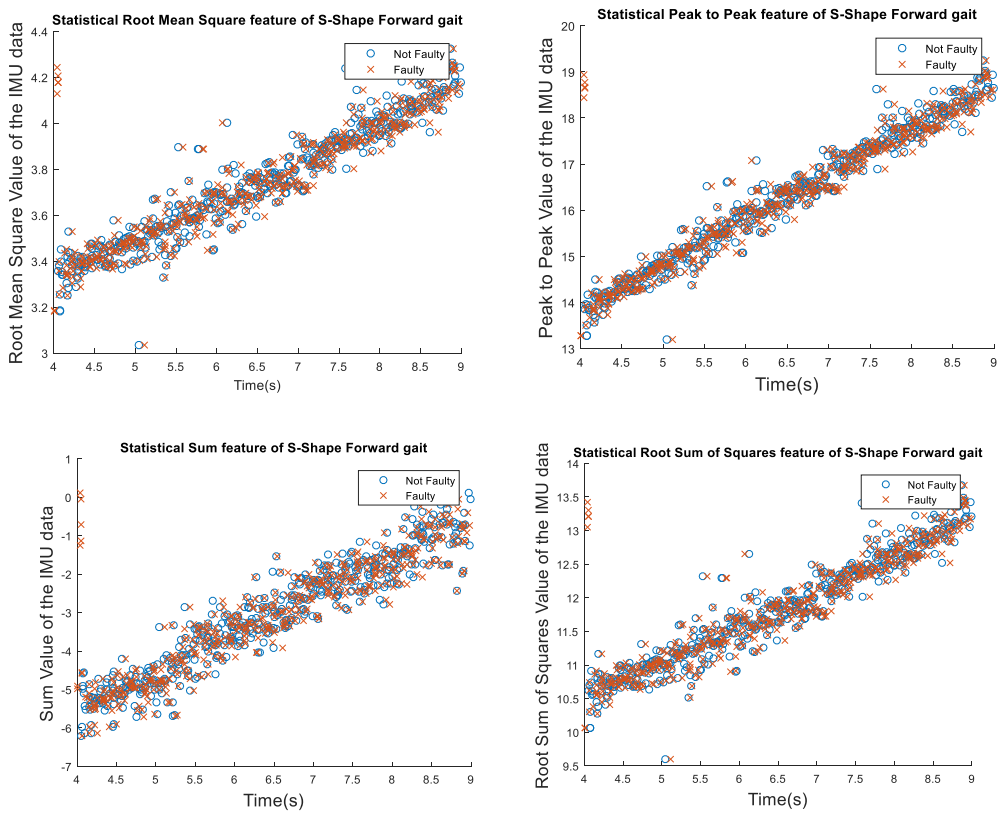


Figure 6.6 Scatter plots indicate the statistical features used in fault detection of two shapes: (a) I-shape Backward locomotion gait; (b) S-shape Forward locomotion gait.

The plot in Figure 6.1shows the sample preparation sets employed to categorize the faults. The plot refers to an apparent separation between the non-faulty and faulty categories. For the purposes of calibration, and because of the varied nature of the shapes deemed within this research, the feature modeling of the IMU information was carried out independently for every shape and locomotion gait. For instance, for the I-shape backward locomotion gait shown in Figure Figure 6.6 Scatter plots indicate the statistical features used in fault detection of two shapes: (a) I-shape Backward locomotion gait; (b) S-shape Forward locomotion gait.

a, the IMU data were modeled during a time between 1s and 15 s, while for the S-shape forward gait, the IMU data were modeled throughout a time between 4s and 9s. The study made certain that the IMU data that was modeled for both the non-faulty and faulty classes, were gathered throughout the similar period that justifies the calibration. The faulty data for seven morphologies were acquired by attaching a payload to block 1 of the robot and causing it to drag while in motion. These faults were of specific significance for this study since they were familiar within the application cases deemed. Future research is set to enlarge the range of faults this research evaluates. The non-faulty IMU data were acquired under normal operating conditions.

6.5 Experimental Result

The fault classification was carried out by means of MATLAB and the Library of SVM support package for every locomotion gait. The amount of testing and training samples varied in accordance with gaits and morphology. The precision acquired for the SVM fault classification over the entire seven morphologies and the two different gaits is presented in **Error! Reference source not found.**.

Table 6.1 Performance of the SVM classifier across the seven shapes of the hTetro robot.

Different Shapes		c-SVC 1		v-SVC 2	
		Linear Kernel	Radial Basis Function Kernel	Linear Kernel	Radial Basis Function Kernel
I shape	Forward	100%	100%	86.9198%	100%
	Backward	100%	100%	95.8498%	100%
J shape	Forward	99.8028%	99.8028%	99.6055%	96.0552%
	Backward	100%	100%	100%	100%
L shape	Forward	100%	100%	100%	100%
	Backward	100%	92.4242%	100%	90.9091%
O shape	Forward	100%	100%	100%	95.6012%
	Backward	100%	100%	100%	100%

S shape	Forward	100%	100%	100%	100%
	Backward	100%	100%	89.6825%	100%
T shape	Forward	93.5644%	94.0594%	88.6139%	93.0693%
	Backward	100%	100%	97.7778%	100%
Z shape	Forward	93.5547%	96.6797%	93.5547%	86.1328%
	Backward	93.2203%	97.4576%	93.2203%	96.1864%

1 c-SVC: C-Support Vector Classification; 2 v-SVC v-Support Vector Classification.

The classification was carried out for both the v-SVC and c-SVC classifiers by means of RBF kernel and linear. The general precision of every classifier is illustrated in Table 6.2.

Table 6.2 Overall accuracy of the four classifiers.

Classifier Type		Overall Accuracy (%)
c-SVC	Linear Kernel	98.581%
	Radial Basis Function Kernel	98.601%
v-SVC	Linear Kernel	98.6087%
	Radial Basis Function Kernel	96.996%

The linear kernel within the v-SVC classifier generated the maximum overall precision of 98.6087%. This high precision was attributed to the fact that most of the statistical features used were linearly divisible; i.e., linear in nature. The linear kernel, within both the v-SVC and c-SVC classifiers, generated a greater level of precision compared to the matching RBF kernel. The performance of the machine learning algorithm is analyzed using the confusion matrix that offers

every likely outcome in tabular form: true negatives, true positives, false negatives, and false positives. In the present study, we built a 2×2 confusion matrix that gives the precise classification and misclassification of the concerned approach. The diagonal components is the amount of correct classification whereas the anti-diagonal components is the amount of misclassifications. The confusion matrix for the experimental case is illustrated in Table 6.3.

Table 6.3 Confusion matrix for the fault classification experiments

Type of Locomotion Gait		c-SVC		v-SVC	
		Linear Kernel	Radial Basis Function Kernel	Linear Kernel	Radial Basis Function Kernel
I Shape	Forward	$\begin{bmatrix} 118 & 0 \\ 0 & 119 \end{bmatrix}$	$\begin{bmatrix} 118 & 0 \\ 0 & 119 \end{bmatrix}$	$\begin{bmatrix} 87 & 31 \\ 0 & 119 \end{bmatrix}$	$\begin{bmatrix} 118 & 0 \\ 0 & 119 \end{bmatrix}$
	Backward	$\begin{bmatrix} 255 & 0 \\ 0 & 251 \end{bmatrix}$	$\begin{bmatrix} 255 & 0 \\ 0 & 251 \end{bmatrix}$	$\begin{bmatrix} 234 & 21 \\ 0 & 251 \end{bmatrix}$	$\begin{bmatrix} 255 & 0 \\ 0 & 251 \end{bmatrix}$
J Shape	Forward	$\begin{bmatrix} 252 & 1 \\ 0 & 254 \end{bmatrix}$	$\begin{bmatrix} 252 & 1 \\ 0 & 254 \end{bmatrix}$	$\begin{bmatrix} 251 & 2 \\ 0 & 254 \end{bmatrix}$	$\begin{bmatrix} 253 & 0 \\ 20 & 234 \end{bmatrix}$
	Backward	$\begin{bmatrix} 118 & 0 \\ 0 & 117 \end{bmatrix}$	$\begin{bmatrix} 118 & 0 \\ 0 & 117 \end{bmatrix}$	$\begin{bmatrix} 118 & 0 \\ 0 & 117 \end{bmatrix}$	$\begin{bmatrix} 118 & 0 \\ 0 & 117 \end{bmatrix}$
L Shape	Forward	$\begin{bmatrix} 252 & 0 \\ 0 & 253 \end{bmatrix}$	$\begin{bmatrix} 252 & 0 \\ 0 & 253 \end{bmatrix}$	$\begin{bmatrix} 252 & 0 \\ 0 & 253 \end{bmatrix}$	$\begin{bmatrix} 252 & 0 \\ 0 & 253 \end{bmatrix}$
	Backward	$\begin{bmatrix} 33 & 0 \\ 0 & 33 \end{bmatrix}$	$\begin{bmatrix} 33 & 0 \\ 5 & 28 \end{bmatrix}$	$\begin{bmatrix} 33 & 0 \\ 0 & 33 \end{bmatrix}$	$\begin{bmatrix} 33 & 0 \\ 6 & 27 \end{bmatrix}$
O-Shape	Forward	$\begin{bmatrix} 168 & 0 \\ 0 & 173 \end{bmatrix}$	$\begin{bmatrix} 168 & 0 \\ 0 & 173 \end{bmatrix}$	$\begin{bmatrix} 168 & 0 \\ 0 & 173 \end{bmatrix}$	$\begin{bmatrix} 168 & 0 \\ 15 & 158 \end{bmatrix}$

	Backward	$\begin{bmatrix} 101 & 0 \\ 0 & 84 \end{bmatrix}$	$\begin{bmatrix} 101 & 0 \\ 0 & 84 \end{bmatrix}$	$\begin{bmatrix} 101 & 0 \\ 0 & 84 \end{bmatrix}$	$\begin{bmatrix} 101 & 0 \\ 0 & 84 \end{bmatrix}$
S Shape	Forward	$\begin{bmatrix} 86 & 0 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 86 & 0 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 86 & 0 \\ 0 & 86 \end{bmatrix}$	$\begin{bmatrix} 86 & 0 \\ 0 & 86 \end{bmatrix}$
	Backward	$\begin{bmatrix} 251 & 0 \\ 0 & 253 \end{bmatrix}$	$\begin{bmatrix} 251 & 0 \\ 0 & 253 \end{bmatrix}$	$\begin{bmatrix} 199 & 52 \\ 0 & 253 \end{bmatrix}$	$\begin{bmatrix} 251 & 0 \\ 0 & 253 \end{bmatrix}$
T Shape	Forward	$\begin{bmatrix} 89 & 13 \\ 0 & 100 \end{bmatrix}$	$\begin{bmatrix} 90 & 12 \\ 0 & 100 \end{bmatrix}$	$\begin{bmatrix} 79 & 23 \\ 0 & 100 \end{bmatrix}$	$\begin{bmatrix} 90 & 12 \\ 2 & 98 \end{bmatrix}$
	Backward	$\begin{bmatrix} 67 & 0 \\ 0 & 68 \end{bmatrix}$	$\begin{bmatrix} 67 & 0 \\ 0 & 68 \end{bmatrix}$	$\begin{bmatrix} 64 & 3 \\ 0 & 68 \end{bmatrix}$	$\begin{bmatrix} 67 & 0 \\ 0 & 68 \end{bmatrix}$
Z-Shape	Forward	$\begin{bmatrix} 257 & 0 \\ 33 & 222 \end{bmatrix}$	$\begin{bmatrix} 257 & 0 \\ 17 & 238 \end{bmatrix}$	$\begin{bmatrix} 257 & 0 \\ 33 & 222 \end{bmatrix}$	$\begin{bmatrix} 257 & 0 \\ 71 & 184 \end{bmatrix}$
	Backward	$\begin{bmatrix} 103 & 16 \\ 0 & 117 \end{bmatrix}$	$\begin{bmatrix} 119 & 0 \\ 6 & 111 \end{bmatrix}$	$\begin{bmatrix} 103 & 16 \\ 0 & 117 \end{bmatrix}$	$\begin{bmatrix} 119 & 0 \\ 9 & 108 \end{bmatrix}$

6.6 Conclusion

To operate efficiently, autonomous robots have to be capable of executing precise fault diagnosis. This chapter presents a new approach to fault diagnosis that using SVM. The technique was experimented on a reconfigurable cleaning robot hTetro. In particular, the study differentiated between non-faulty and faulty circumstances across seven diverse morphologies across two different gaits. The statistical aspects, as well as mean, variance, standard deviation, root mean square, root sum of squares level, peak-to-peak value, maximum value, and sum, were taken out from the IMU data and were consequently used as input into an SVM that classified between non-faulty and faulty states. The study tested the algorithm by means of two diverse forms of SVM:

v-SVC and c-SVC, and consequently contrasted their performance while diagnosing flaws within the hTetro performance. Besides employing two diverse kinds of SVM, the study tested two diverse kernels: radial basis and linear function. The findings showed that the entire four SVM-based methods attained a classification precision of more than 90% over the seven different shapes at forward and backward locomotion. The general precision of the entire four classifiers was higher than 97%. The c-SVC's RBF kernel and linear kernel of the v-SVC classifiers attained a greater overall precision compared to the other classifiers. Future research will entail integrating extra sensors into the robot to broaden the range of fault classification and categorize extra fault types. A methodical dynamic model is being created to calculate the consequences of a recognized set of familiar faults on locomotion gaits.

7 Towards Artificial Neural Network Based Fault Diagnosis in a Bio-inspired Reconfigurable Robot

The employment of robots to undertake daily tasks has escalated exponentially in the past few years, and human dependence on robotic technologies keeps increasing as technologies develop. Modern robots are incredibly complex mechatronic systems, which use various hardware and software modules, which contain numerous features. The complex nature of modern robot systems coupled with the ambiguous settings wherein they operate implies that they experience faults regularly, regardless of their ingenious designs. In view of this, machine learning methods have increasingly become indispensable and are often utilized for detecting and diagnosing faults within different industrial systems. Researchers in [187] suggested the creation of a mechanism of fault diagnosis for the power transformer filled with oil. With the assistance of dual-step Artificial Neural Network (ANN) classification, they locate the absence or presence of fault for the involved cellulose. The authors in [132] proposed a mechanism for diagnosing faults, which employed an SVM to track the maintenance system of the machine to decrease the maintenance costs, enhance machine availability, and escalate productivity. The SVM model attained a classification precision ranging between 95 and 100% when contrasted against other SVM models. A decision tree and SVM technique was utilized for diagnosing faults, which emerged within oscillating mechanical systems in a study by [51]. The study employed the vibration signals generated in the system operation to obtain the statistical properties needed for the nu-SVC and c-SVC SVM classification models. In [133], authors suggested multi scale-entropy and adaptive neuro-fuzzy-oriented fault diagnosis method, which can be utilized for detection of bearing faults [189]. An examination of two diverse neural networks was undertaken: non-linear radial basis function-based neural networks and time delay-based neural networks. The weights were updated through a general parameter learning method. In [135], a comparative study of genetic algorithm alongside artificial neural network and support vector machine is presented in the context of identification of bearing faults. The system functioned by gathering vibration-signal data of rotor machine containing non-fault and fault bearing for extraction of features. The classification was subsequently undertaken with SVMs and ANNs.

Beside SVMs and ANNs, investigators have studied various methods for diagnosing faults in industrial applications. [179] discusses a fault isolation and detection technique, which used a non-linear unknown input observer in the analysis of malfunctioning features of the independent spacecraft, which was utilized in the rendezvous stage of the mission for Mars Sample Return (MSR). Unlike the use of a machine learning method, the fault detection approach used observers and position models. The system that [137] proposed for detection of a motor roller bearing fault, was a fault diagnosis approach based on neural-network and they employed practical and theoretical results to infer that the neural network method is efficient. [188] investigated a method for locating ball-bearing faults within a 3-phase induction motor. The suggested system combined Fisher's PCA and LDA to decrease dimensionality. The investigators utilized a C4.5 decision tree algorithm alongside random tree algorithms to classify incorrect and accurate data. [190] presented a description on the employment of a complementary vector data explanation for fault diagnosis within ancient circuits. The suggested process used fractional wavelet transform, which were derived from the fuzzy classifier on the basis of complementary vector data explanation in effectively testing the circuits and detecting faults.

In the robotics field, machine learning techniques are increasingly becoming prevalent as the means for overcoming issues associated with diagnosis of faults. [140] investigated a neural network technique for diagnosing faults within the actuator of autonomous underwater vehicles (AUV). The key objective of the algorithm involved detecting actuator faults, which affected the propulsion and control surface system. A simulated case study was created for testing the suggested technique with the results showing that the detection of faults within the NPS AUV II to the highest accuracy level could be undertaken through the algorithm. The technique that [141] proposed, featured a method of fault diagnosis, which used linear SVM. The testing of the system was upon a robotic manipulator, which functioned as an outside application of thread fastening. It offered two characteristic parameters, approximating the optimal force of reaction and length of insertion, to differentiate between four fastening events. The results for the classifier performance evaluation showed that a 90% accuracy level was attained by the system. An investigation by [142] investigated the use of a fault diagnosis method to a quadruped robot equipped hydraulic system. The system combined Cuckoo Search (CS) and the SVM algorithms to categorize faults alongside a rough dimensionality to decrease the properties. The results showed that the CS and SVM

alongside a rough dimensionality produced a greater level of precision than using the conventional SVM technique alone. The technique that [191] proposed was intended to locate faults, which arose within the mobile robots' control software. The robot's operation parameters in different situations were used for SVM classifier testing and refining. While many investigators have utilized machine learning methods in the diagnosis of faults, which arise within robotic systems, the investigations have often been restricted to robotic systems with fixed-morphology. Various studies apply machine learning techniques for fault detection in robotic systems. Nevertheless, such works are only limited to the robots with fixed morphology and very little attention has been given to the emerging class of reconfigurable robot involving multiple morphologies [192]. In this study, we present an application that utilizes machine learning methods for fault detection in robots that can re-configure to assume multiple morphologies. One of the ways in which this research differs from the state of the art is that it accounts for the fault types associated with multiple morphologies in a reconfigurable robot and associated locomotion gaits.

Our current research efforts aim to develop reconfigurable robots, which can independently manage and detect faults, while being changing of changing their shapes as well as adjusting the modes of locomotion. Last 10 years have witnessed tremendous advancement in the area of reconfigurable robotics. [192] unveiled a reconfigurable robot, which could transform from a two-wheeled driving system to a tri-legged walking system. Using the legged modes, the robot could navigate different obstacles in its path, while the wheeled mode enabled it to move effectively from one point to another. Another study by [96] described an unmanned underwater reconfigurable vehicle. They used a three-level guidance for accelerating motion, turning and controlling the forward. [193] put forward and validate a self-reconfigurable modular robotic block capable of connecting with other blocks to create various morphologies for performing different operations. A robot presented in [194] showed the capacity of crawling and swimming across several terrains as well as avoid obstacles, which block its path. The results of the experiment showed that the robot could monitor the light at a distance of 50 cm the robot's swimming speed stood at 0.37 m/s whereas the minimum radius of turning was 0.12m. Despite a considerable literature, which explores the subject of reconfigurable robots, the investigation of issues linked fault diagnosis within shape-shifting robots can be considered scarce [147]; therefore, a research gap exists within this area that should be filled. It is important for reconfigurable robots to have

the capacity of diagnosing and detecting faults.

This chapter presents a description on the use of a fault diagnostic system that is based on a neural network capable of classifying defects linked to crawling and rolling locomotion modes. The proposed system was validated across faulty and non-faulty conditions involving 9 locomotion gaits, across rolling or crawling modes and three distinct speeds. Key statistical features such as diagonal sum, maximum value, root sum of square level, peak-to-peak value, root mean square, variance, standard deviation, and mean were obtained from the IMU sensor placed onboard the Scorpio robot. Such statistical features were used as input to the neural network and for classifying faulty conditions from non-faulty ones.

7.1 A System Overview of the Scorpio Robot

This study presents our experiments involving a crawling–rolling reconfigurable robot called Scorpio, which employed neural networks towards diagnosis of faults. With respect to the rising number of robots, which are utilized in urban reconnaissance activities, there is considerable necessity for such systems to assume diverse morphologies as a means of navigating complex terrains. In this research work, the robot that was tested featured a self-reconfigurable, bio-inspired system based on Cebrennus rechenbergi, a huntsman spider species, which originates from Morocco's Erg Chebbi desert. Like many spider species, the spider mainly moves through crawling. However, in life-threatening scenarios, it escapes by rolling within diverse directions using flip-flop motions with its legs. It is the only species on earth that can move in that manner. Another interesting feature of the spider pertains to the fact that it does not require sloped or slippery surface to initiate the rolling movement. Rather, it shifts into the rolling posture on flat ground and uses flip-flop motions to roll. Similar to the aforementioned spider, the robot, which formed the basis of this study was meant to effectively synthesize numerous modes of locomotion including climbing, crawling, and rolling [149] [150].

Through a thorough analysis of the gaits that Cebrennus rechenbergi spiders use, a broad set of locomotion and behavioural gaits were synthesized, simulated, modeled, and identified via real hardware. Several experiments of fault diagnosis were undertaken including various locomotion gaits such as right and left shifts, right and left curves, right and left turns, backward, and forward rolling. The robot's external body was fabricated from PLA plastic with a three-dimensional (3D) printer. The robot had four legs (quadruped), each leg of 145 that contained 3

servo motors. The motors provided 30 of freedom, which were necessary for the machine to conduct crawling and rolling movements. To reduce the robot's weight and size, motors measuring 11.5x21.5x21.5 mm with 11.5gms in terms of weight were chosen. Twelve JRES376 servomotors were added to the robot, each containing a voltage of 4.8 v150. The twelve servomotors were micro-controlled using a Pololu Maestro servo controller. Figure 7.1 illustrates the Scorpio robot's CAD model with the details of the actuator.

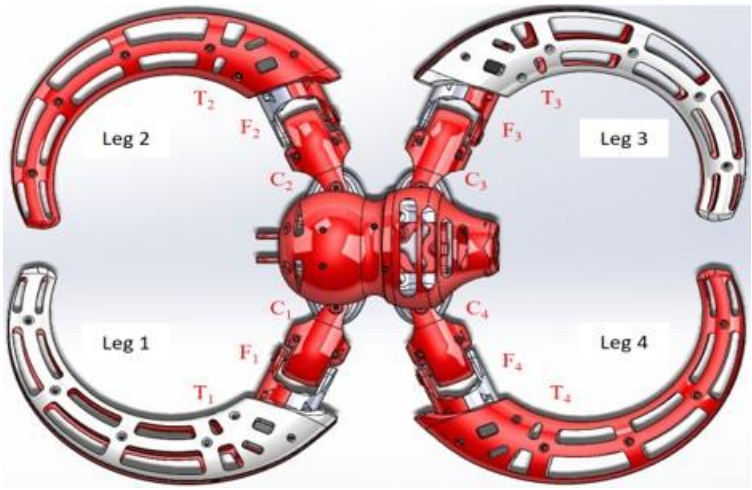


Figure 7.1 CAD model of Scorpio robot with actuator details

Apart from the actuators, which facilitated locomotion of the robot alongside the central microcontroller, the system had a 3-axis magnetometer, an LSM303D 3-axis accelerometer, and a Pololu MinIMU-9 v3 IMU sensor (containing an L3GD20H 3-axis gyro). The role of the IMU detector was to compute translational and rotational data, which the users could use for detection locomotion gait anomalies. The central microcontroller was an Arduino pro-mini premised on ATmega328 that requires 5 -12 volts. The Scorpio robot's hardware specifications are summarized in Table 7.1.

Table 7.1 Hardware Specification of Scorpio Robot.

Hardware Components	Specifications
Central Controller	Arduino Mini Pro with Atmega 328
Servo Motor	JR ES376

Servo Controller	Pololu maestro Servo controller
IMU	Pololu MinIMU-9 v3
Outer body	Polylactic acid or polylactide (PLA)

The IMU sensor frequently interconnected with the accelerometer and gyroscope through short messages to the Arduino pro mini microcontroller. The servo controller received the output control commands from the Arduino controller through the serial communication procedure. The Pololu Maestro servo controller is capable of controlling as much as 18 servomotors. The robot can be operated using a wireless gadget furnished with a committed Blue fruit EZ-Link transceiver. Figure 7.2 illustrates the complete system design of the Scorpio robot.

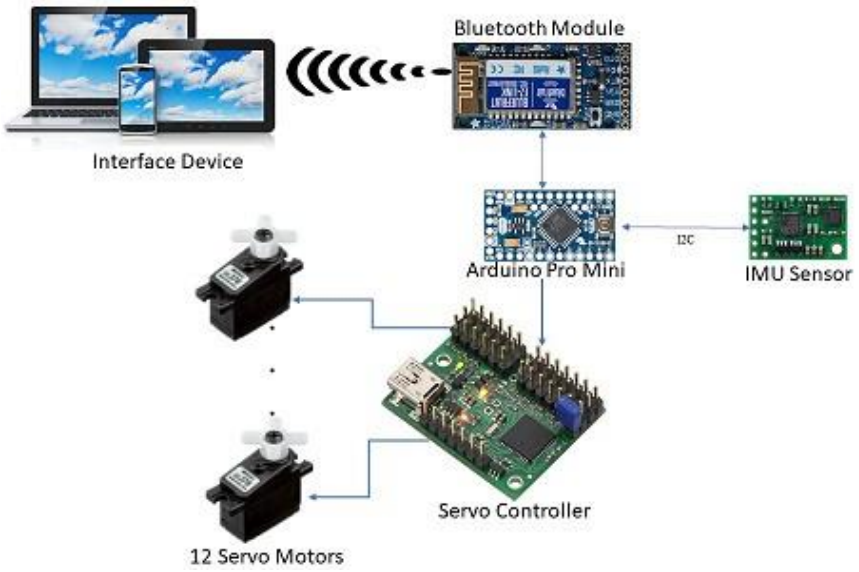


Figure 7.2 System architecture of the Scorpio robot

7.1.1 Crawling Locomotion

Both motors instantaneously rotate around the roll and yaw axis to accomplish the crawling locomotion. The leg is raised by means of the fem motor, while the roll motion is consequently generated. Once the robot takes a forward and backward crawling motion. The yaw motion is then produced by the Cox motor pushing the leg forward or backward. The Cox motor pushes the leg in the clockwise direction and the fem motor is employed in instigating the roll motion to allow the

robot to crawl right. On the other hand, the Cox motor pushes the leg in the anti-clockwise direction to allow the robot to crawl left. At the same time, 8 crawling gaits were created for Scorpio. The crawling locomotion gaits could be managed using the Cox and fem motors only.

7.1.2 Rolling Locomotion

Crawling was the default gait of locomotion for Scorpio. The rolling locomotion, however, is more sophisticated as it enables the robot to traverse flat terrains, slopes and stairways. To trigger rolling locomotion, the robot transforms into a default rolling gait state. This is realized by legs three and four going into an inverted gait to form a ring with legs one and two. The loci of legs three and four are moved over the Cox and fem motors as well as T3 and T4. Just two out of the four legs of the scorpion touch the ground at any given time whenever the Scorpion is in the rolling position. The motors using the legs are stimulated to perform the forward or backward rolling locomotion. As soon as the robot has performed 50% of the rolling motion, the next 2 legs touch the ground. The legs are then reset to accomplish the subsequent phase of the rolling course. The process through which the rolling movement is reached involved fem motors only. Figure 7.3 demonstrates the crawling and rolling morphologies of Scorpio.

Figure 7.3 Crawling and Rolling morphologies of our Scorpio robot

7.2 Extraction of Features

We implemented statistical features to describe the robot in both faulty and non-faulty conditions so as to detect and classify faults. In this study, we use gyroscope and accelerometer readings from an IMU sensor. From the obtained results, we extract the mean, standard deviation, variance, root-mean-square level, peak-to-peak, root sum of squares level, and maximum value. All statistical features are described comprehensively below. The dimensionality of the initial IMU

sensor data features were significantly reduced by the use of statistical approaches for feature extraction.

1. Mean: This is the average of an IMU data set at any particular instant in time. Eq. (7.1) is used in the calculation of the mean of every IMU data set.

$$Mean = \frac{\sum_{i=1}^{N} x_i}{\mathrm{N}} \tag{7.1}$$

where,

x is the IMU data for instance, and N is the number of IMU data in that instance

In our case, N=7.

2. Standard deviation: It determines the point to which every IMU data set is distributed from one another. This data can be used in assessing the distinction between each IMU data point and the mean. Eq. (7.2) is used in the calculation of the standard deviation of IMU data attained from every situation of data collection.

$$Standard\ deviation = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{7.2}$$

where,

μ is the mean of the IMU data for instance, and x represents the IMU data for an instance and N =7.

3. Variance: It is the squared standard deviation of the average IMU values. This quantity characterizes the discrepancy of the IMU data at any point in time and can be computed by Eq. (7.3), as shown:

$$Variance = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{7.3}$$

where,

μ is the mean of the IMU data for instance, and x represents the IMU data for an instance and N =7.

4. Root Mean Square: Root Mean Square is also referred to as quadratic mean and is the square root of the arithmetic average of the squared specimen of IMU values. Root Mean Square

symbolizes the degree or the characteristic extent of IMU data per data collection area and is calculated by Eq. (7.4):

$$Root\ Mean\ Square = \sqrt{\sum_{i=1}^{N} \frac{(x_i)^2}{N}} \tag{7.4}$$

Where, x represents the IMU data for an instance and N=7.

5. Peak-to-peak value: This is the variation between the minimum and maximum IMU data at any point in time. the minimum IMU data in a point is the trough, while the maximum IMU figure in a popint is the crest.

$$Peak\ to\ peak\ value = crest - Trough \tag{7.5}$$

6. Root sum of squares level: This is computed by finding the square root of the summation of the squared IMU value acquired for a particular instance

$$Root\ Sum\ of\ Square\ level = \sqrt{\sum_{i=1}^{N} |x_i|^2} \tag{7.6}$$

Where, x represents the IMU data for an instance and N=7.

7. Maximum Value: This is the greatest value of the information attained from the IMU in a particular event.
8. Sum: This is a feature that adds up all the IMU values attained for a particular event.

$$Mean = \sum_{i=1}^{N} x_i \tag{7.7}$$

Where, x represents the IMU data for instance and N = 7.

7.3 Artificial Neural Network

The statistical features were used in developing a system in which the information could be categorized as soon as they were extracted from the IMU data. Artificial neural networks stands for a progressive machine learning technique through which a robot can obtain, keep, and recall complex material. They are capable of replicating the innermost operations of biological neurons and are structured in layers. A link for activation is applied in connecting the neurons in every layer with the neurons in the adjacent layer. The information signals are successively conveyed through this link. The neurons are capable of learning from their precursors when

signals are transmitted in this manner. Furthermore, it is conceivable to influence the learnings of particular neurons since the strength of the information signals controls the precision of the neural networks by altering the strengths issued to the information signals. By itself, the responsibility of calculating the weights requiring be ascribed to every information signals released from the neurons lies in our hands. All neural networks display exceptional design that differs from one application to another. The utilized neural networks with Scorpio had one output, one hidden and one input layer. whereas the hidden layer had 4 neurons, not inclusive of the bias neurons, the input layer contained eight neurons, not including the bias neurons. The output layer contained only one neuron since the fault diagnosis used in this study assumed the form of a double classification. Figure 7.4 illustrates the design summary of the neural networks used with

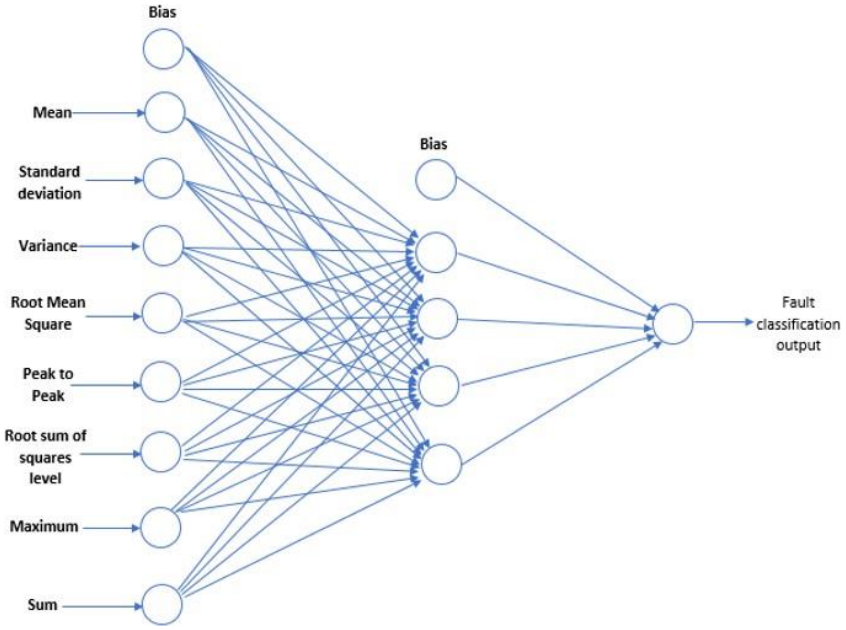


Figure 7.4 ANN architecture

Scorpio. To compute the hypothesis, the neural network employed the forward propagation algorithm. This hypothesis applied the original weights to give the output of the neural network. The calculation of the cost function was done. This stands for the divergence between the calculated and expected output. it also has the form of a base value that varies in terms of the use of the neural network. The back-propagation algorithm and gradient descent method were applied in updating the weights. The gradients of the weights are calculated using the back-propagation algorithm calculates. These are then used in the gradient descent process to determine the conventional lowest

point for the weights used in the neural network. The three levels were used in the neural network applied in Scorpio include backward propagation, cost function determination, and forward propagation. Each of these is discussed in detail below.

7.3.1 Forward propagation algorithm

Forward propagation encompasses the application of the activation function to estimate the output proposition. Let x be the Neurons in the input layer, f be activation function at the outset, a2 be Neurons in the hidden layer, y be initial output for the exercise data, a3 be Neurons in the output layer (Output of the neural networks), w2 be Weights for the contacts between the hidden and the output layer, and w1 be Weights for the links between the input and the hidden layer. We only have the values of the input layer neurons. We are capable of calculating a2 and a3 via the use of forwarding propagation. It is probably to define the value of a2 by computing the activation function for the product of the weights w1and x using Eq. (7.8).

$$a_2 = f(w_1^T * x) \tag{7.8}$$

By defining the activation function for the product of weights w2 and a2, the value of a3 can then be worked out using Eq. (7.9).

$$a_3 = f(w_2^T * a_2) \tag{7.9}$$

In the scenario of Scorpio, a sigmoid function was applied as the activation function.

This is as per Eq.(7.10).

$$f(z) = f\left(\frac{1}{1+e^{-z}}\right) \tag{7.10}$$

The magnitude of a3 gives the output for the classification.

7.3.2 Cost function algorithm

The cost function makes it possible to determine the degree at which the neural network output is dissimilar from the expected output. A high-cost function is symptomatic of a failed precision. The cost function is recomputed whenever the weights are updated. The process is stopped the moment the cost function drops below a base value, as this indicates that the essential level of accuracy has been attained.

The cost function is denoted by Eq. (7.11)

$$Cost\ Function = \sum_{i=1}^{n}(y^i * \log(a_3^i)) + ((1 - y^i) * \log(1 - a_3^i) + \frac{\lambda}{2n}\sum_{i=2}^{h}(w_2^i)^2 \quad (6.11)$$

Where, n is the total number of training data.

7.3.3 Backpropagation algorithm

This algorithm is used in the calculation of the gradients for the weights, which are later applied in updating the weights. The process, as indicated by the name of the algorithm, entails starting from the output layer and continuing to the input layer.

Using Eqs 12 and 13, the gradients for w1 and w2 were calculating accordingly. i.e., let g2 be the gradient of w2 and g1 be gradient of w1.

$$g_1 = \frac{1}{n}(\sigma_1^1 + \lambda w_1^1) \text{ for i} \neq 1 \text{ for } g_1 = \frac{1}{N}\sigma_1^1 \text{ for i= 1} \quad (7.12)$$

$$g_1 = \frac{1}{n}(\sigma_2^1 + \lambda w_2^1) \text{ for i} \neq 1 \text{ for } g_1 = \frac{1}{N}\sigma_2^1 \text{ for i= 1} \quad (7.13)$$

We need to determine σ1 and σ2 using Eqs. 7.14 and 7.15 respectively:

$$\sigma_1 = \sigma_1 + (x^T * \delta_1) \quad (7.14)$$

$$\sigma_2 = \sigma_2 + (a_2^T * \delta_2) \quad (7.15)$$

Where,

$$\delta_2 = a_3 - y \quad (7.16)$$

$$\delta_1 = (\delta_2 * w_2).* a_2.* (1 - a_2) \quad (7.17)$$

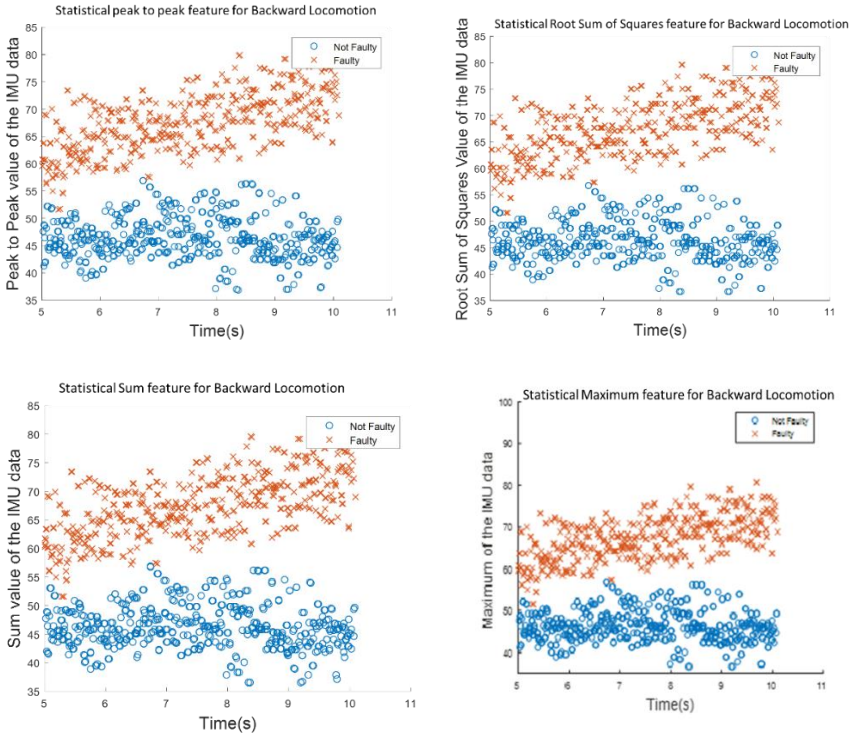
The w2 and a2 used to determine δ1 must not include the bias term.

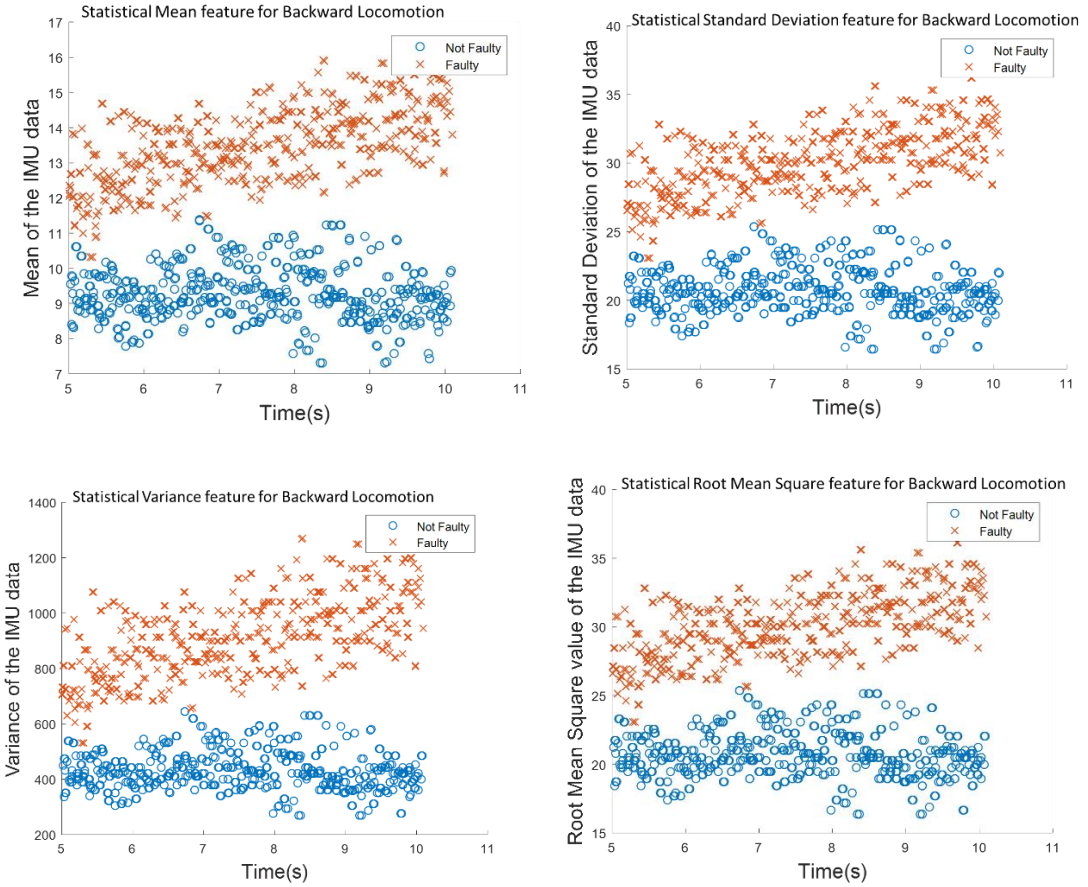
7.4 Experimental setup

This study presents our experiments that used the statistical features in providing the critical information needed by the Scorpio robot in classifying between the locomotion modes which are non-faulty and those that are faulty. We collected 330 data associated with nine locomotion gaits that spanned the rolling and crawling modes at three speeds. Figure 7.5a illustrates the scatter plots of the attained statistical features for the backward locomotion pace at a speed of three degrees for

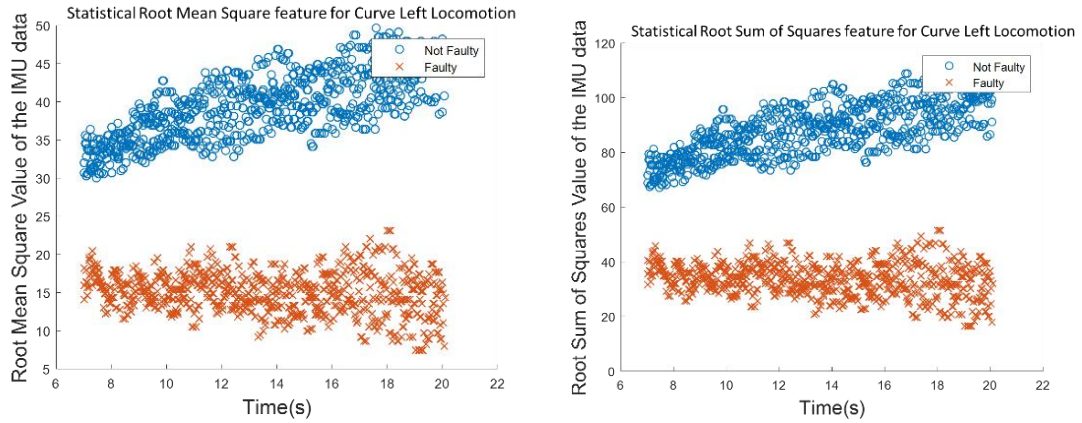
every period from the IMU. Figure 7.5b illustrates the scatter plots of the statistical features attained for the curve left locomotion at a speed of six degrees for every period from the IMU.

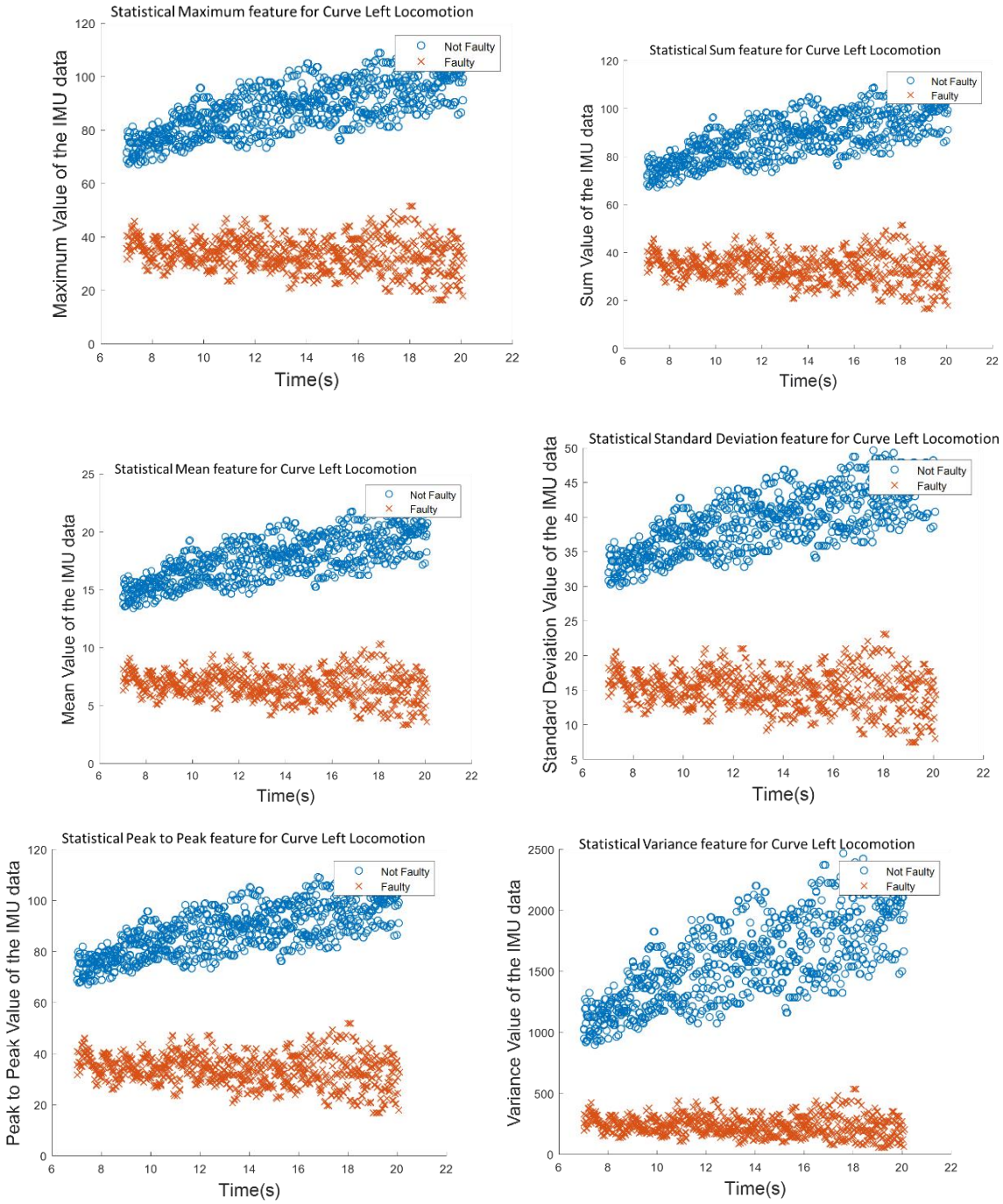
In Figure 7.5, the plot highlights the sample training sets that were used in classifying the faults. There is a clear illustration of separation between faulty and non-faulty classes. Owing to the distinct nature of the gaits assessed in the recent study and for the intention of calibration, the IMU data's statistical features were evaluated at different points in time for every locomotion gait. A good example is the backward locomotion gait shown in Figure 7.5a, the IMU data was collected at a point in time between 10 and 5 seconds, whereas, for the curve left locomotion gait exhibited in Figure 7.5b, the IMU data was collected at a point in time between 20 and 7 seconds. We made sure that the IMU data for both the faulty and non-faulty classes were sampled at the same moments in time, which justify the calibration. With the exception of the rolling gait, the faulty IMU data that was employed to train and test classes was attained through the deactivation of the following motors: C4 (Front-Right Cox), C3 (Back-Right Cox), C2 (Back-Left Cox). When it comes to the rolling motion, the fault data was attained through the simulation of a disturbance in two of the fem motors, F4 (Front-Right Fem) and F1 (Front-Left Fem).




Statistical Mean feature for Backward Locomotion
Mean of the IMU data
Time(s)
Statistical Standard Deviation feature for Backward Locomotion
Standard Deviation of the IMU data
Time(s)
Statistical Variance feature for Backward Locomotion
Variance of the IMU data
Time(s)
Statistical Root Mean Square feature for Backward Locomotion
Root Mean Square value of the IMU data
Time(s)
Not Faulty
Faulty

(a)


Statistical Root Mean Square feature for Curve Left Locomotion
Root Mean Square Value of the IMU data
Time(s)
Statistical Root Sum of Squares feature for Curve Left Locomotion
Root Sum of Squares Value of the IMU data
Time(s)
Not Faulty
Faulty



(b)

Figure 7.5 Scatter plot indicates statistical features used in fault detection of two locomotion: (a)Backward locomotion at a speed of three degrees per period; (b)Curve left locomotion at a speed of six degrees per period

7.5 Experimental Results

For every locomotion gait, the classification was carried out with the use of the artificial neural network. The number of testing and training samples differed in accordance with the speed and gait. It is clearly illustrated in Table 7.2, that the precision attained from the classification process using neural network across all three speeds as well as all nine locomotion gaits. The outcomes clearly indicated that the number of iterations needed by the neural network in accomplishing the most favorable weights differed from one locomotion gait to another. The rate of learning also differed from one locomotion gait to another. In particular, the rate of learning and the number of iterations differed according to the degree to which the fault influenced the robot locomotion of. In machine learning applications, a confusion matrix is frequently used in evaluating the classifier's performance. This matrix stands for every possible result in table form: false and positives and false and true negatives. With the purposes of testing Scorpio, we came up with a 2x2 confusion matrix that really specified the accurate misclassification and classifications of the interest classifier. The column pointed out the instance in the predicted class, whereas the row pointed out the instance in the original class. The amount of the diagonal components indicated the number of accurate classifications, whereas the amount of the anti-diagonal components indicated the number of misclassifications. Table 7.3, clearly illustrates the confusion matrix that was used in the Scorpio experimental cases.

Table 7.2 Performance of the neural network classifier across nine locomotion gaits of the Scorpio robot at all three speeds

Type of Locomotion Gaits		Overall Accuracy (%)
Rolling	3 degree/period	100%
	6 degree/period	98.26%
	12 degree/period	95.29%

Forward	3 degree/period 6 degree/period 12degree/period	98.26% 100% 100%
Backward	3 degree/period 6 degree/period 12degree/period	100% 91.70% 100%
Left Turn	3 degree/period 6 degree/period 12 degree/period	100% 97.67% 97.54%
Right Turn	3 degree/period 6 degree/period 12 degree/period	98.31% 95.78% 100%
Curve Right	3 degree/period 6 degree/period 12 degree/period	98.89% 100% 99.4%
Curve Left	3 degree/period 6 degree/period 12 degree/period	100% 100% 100%

Shift Right	3 degree/period 6 degree/period 12 degree/period	100% 100% 100%
Shift Left	3 degree/period 6 degree/period 12 degree/period	100% 98.14% 100%

Table 7.3 Confusion matrix obtained for the Scorpio fault classification experiments

Type of Locomotion Gaits		Confusion Matrix
Rolling	3 degree/period	$\begin{bmatrix} 50 & 0 \\ 3 & 53 \end{bmatrix}$
	6 degree/period	$\begin{bmatrix} 82 & 0 \\ 3 & 87 \end{bmatrix}$
	12 degree/period	$\begin{bmatrix} 84 & 8 \\ 0 & 78 \end{bmatrix}$
Forward	3 degree/period	$\begin{bmatrix} 83 & 0 \\ 3 & 86 \end{bmatrix}$
	6 degree/period	$\begin{bmatrix} 86 & 0 \\ 0 & 85 \end{bmatrix}$
	12degree/period	$\begin{bmatrix} 69 & 0 \\ 0 & 68 \end{bmatrix}$
Backward	3 degree/period	$\begin{bmatrix} 86 & 0 \\ 3 & 86 \end{bmatrix}$
	6 degree/period	$\begin{bmatrix} 126 & 21 \\ 0 & 106 \end{bmatrix}$
	12degree/period	$\begin{bmatrix} 85 & 0 \\ 0 & 86 \end{bmatrix}$
Left Turn	3 degree/period	$\begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$
	6 degree/period	$\begin{bmatrix} 85 & 4 \\ 0 & 83 \end{bmatrix}$
	12 degree/period	$\begin{bmatrix} 60 & 3 \\ 0 & 86 \end{bmatrix}$

Curve Right	3 degree/period	$\begin{bmatrix} 133 & 0 \\ 3 & 134 \end{bmatrix}$
	6 degree/period	$\begin{bmatrix} 84 & 0 \\ 0 & 84 \end{bmatrix}$
	12 degree/period	$\begin{bmatrix} 84 & 1 \\ 0 & 83 \end{bmatrix}$
Curve Left	3 degree/period	$\begin{bmatrix} 223 & 0 \\ 0 & 223 \end{bmatrix}$
	6 degree/period	$\begin{bmatrix} 220 & 0 \\ 0 & 220 \end{bmatrix}$
	12 degree/period	$\begin{bmatrix} 119 & 0 \\ 0 & 120 \end{bmatrix}$
Shift Right	3 degree/period	$\begin{bmatrix} 151 & 0 \\ 0 & 150 \end{bmatrix}$
	6 degree/period	$\begin{bmatrix} 86 & 0 \\ 0 & 86 \end{bmatrix}$
	12 degree/period	$\begin{bmatrix} 84 & 0 \\ 0 & 82 \end{bmatrix}$
Shift Left	3 degree/period	$\begin{bmatrix} 200 & 0 \\ 0 & 202 \end{bmatrix}$
	6 degree/period	$\begin{bmatrix} 131 & 2 \\ 3 & 133 \end{bmatrix}$
	12 degree/period	$\begin{bmatrix} 100 & 0 \\ 0 & 102 \end{bmatrix}$

7.6 Conclusion

While robots are increasingly becoming autonomous in carrying out functions devoid of any form of human involvement, they should also autonomously overcome and detect any faults and errors. This chapter presents a neural network method towards classifying and detecting faults that was tested on a bio-inspired reconfigurable robot, the Scorpio. The system was meant to classify

and identify any faults that occurred when Scorpio was engaged in crawling and rolling activities. Specifically, we classify between faulty and non-faulty conditions across nine different locomotion gaits involving rolling as well as crawling modes over three different speeds. Statistical features extracted from the sensor data comprising of the sum, mean, maximum value, standard deviation, root sum of squares level, variance, peak-to-peak value and root mean square, were employed as input to the neural networks to assist the system in distinguishing between non-faulty and faulty conditions. In computing the hypothesis from the output neuron, the forward propagation algorithm was applied. The sigmoid function was employed in activating the function applied in the forward propagation. Backward propagation was applied in identifying the gradient of weights, and these were later used in updating the neural network's weights. Consequently, the weights were updated up to the point where the cost function was less than threshold value. The results indicate a classification accuracy of over 91% for all of the ANN-based approaches tested over the nine locomotion gaits at three different speeds. Future research will entail integrating extra sensors into the robot to broaden the range of fault classification and categorize extra fault types. A methodical dynamic model is being created to calculate the consequences of a recognized set of familiar faults on locomotion gaits.

8 Conclusion

In this thesis, we focused on investigation and diagnosis of faults that affect task performance in reconfigurable robots using machine learning techniques, and we are using data-driven fault detection method to detect the faults from the reconfigurable robot , Here we performed on a theoretical level and experimental level. In a theoretical level, we used two different frameworks namely Support vector machine and Artificial neural network to detect the fault in the reconfigurable robot. And in this thesis, we did the alternative method for detecting faults, namely fault injection and learning. We collect the data from the robot and use that data to train the robot to detect the presence of the faults. The advantage of our method is that no need any additional sensors are needed, and we do not need to build a mathematical model of how the robot should behave. Also, in this thesis, we tried a new path planning approach based on VPH+. In an experimental level, we designed and built two novel reconfigurable robots as part of this doctoral thesis. The first robot was a bio-inspired reconfigurable crawling-rolling platform, Scorpio. The second robot was a Tetris inspired reconfigurable floor cleaning robot, hTetro. Exhaustive experiments were run to collect data while operating under normal conditions and after faults have been induced. Based on the extracted data, we used machine learning approaches to train and identify the presence of faults in robot operation.

8.1 Constrained VPH+: a local path planning algorithm for a bio-inspired crawling robot with customized ultrasonic scanning sensor

The CVPH+ algorithm is found to perform better than the VPH+ algorithm in both empty and clutter-filled environments. By formalizing implicit assumptions made in the VPH+ algorithm regarding the sensor and the robot model, both of which were invalid for the Scorpio robot, the algorithm is improved.

Further insight was obtained from examining the original VPH+ cost function further and reformulating it to improve the performance of the CVPH+ algorithm.

The performance of both the VPH+ and the CVPH+ algorithms is reduced because of Pose inaccuracy due to the noisy gait primitive estimates, as well as the inherent inaccuracies due to the

3D printing process. A more comprehensive formulation taking into account pose and map estimates will result in better performance of the algorithms.

A further avenue of future research includes the use of the rolling configuration of the robot to improve its speed in traversing the environment. An interesting aspect to be explored is the challenges caused by the displacement of the ultrasonic sensors in the rolling mode of the robot.

8.2 Scorpio: A biomimetic reconfigurable rolling–crawling robot

In this study, we described and recited the mechanical design and performance testing of a uniquely designed legged robot Scorpio inspired a Cenrennus rechenbergi spider which can crawl and roll. The Scorpio robot has 4 legs that are reconfigurable and has the capability to change its morphology for transforming itself from crawling to rolling. The uniqueness of not using any extra mechanism and by using the same set of actuators the robot can efficiently crawl and roll. The mechanical design of the robot, its control design, sensors and software's are explained in detail.

The robots' performance based on each type of perception is experimented and showcased. As confirmed the robot is capable to crawl rolls and transform between these two states. From the interoceptive perspective the robot can transform itself while it falls sideways and turns upside-down. From the exteroceptive side the vison system has helped the robot in perceiving the terrains and able to actuate the required gaits.

For further research the autonomous capabilities will be explored more widely. Advanced algorithms for perception and identification of various terrains and floorings would be studied. Also the system would be well developed to autonomously faces certain bumps and obstacles and escalate down the stairs using advanced vision calculation An more energy efficient system would be developed for energy management and highly capable to work competently within the power limit.

8.3 Fault Diagnosis of a Reconfigurable Crawling–Rolling Robot Based on Support Vector

In this study, we have presented a novel approach towards fault detection in the crawling-rolling self-configurable Scorpio robot using Support Vector Machine. The fault detection is an important aspect for this robot so that it can reconfigure itself to compensate for any current fault. The fault detection was carried for a total of nine gaits at three different speeds. The eight statistical features are extracted from the IMU data which provides information regarding the movement of the Scorpio robot. The classification was carried out using two different SVM kernel function of two SVM model in Support Vector Machine. The SVM was able to provide an accuracy of above 90% for the test cases of all the nine gaits using two different SVM classification strategy and two different kernels. The overall accuracy of all the four classifiers is above 97% which is practically good. The linear kernel of c-SVC and nu-SVC classifier appears to provide a better overall accuracy when compared to the RBF kernel of c-SVC and nu-SVC classifier.

8.4 Morphological Fault Diagnosis in a Tetris-inspired Support Vector Machine-Based Reconfigurable Cleaning Robot

To operate efficiently, autonomous robots have to be capable of executing precise fault diagnosis. This study presents a new approach to fault diagnosis that using SVM. The technique was experimented on a reconfigurable cleaning robot hTetro. In particular, the study differentiated between non-faulty and faulty circumstances across seven diverse morphologies across two different gaits. The statistical aspects, as well as mean, variance, standard deviation, root mean square, root sum of squares level, peak-to-peak value, maximum value, and sum, were taken out from the IMU data and were consequently used as input into an SVM that classified between non-faulty and faulty states. The study tested the algorithm by means of two diverse forms of SVM: v-SVC and c-SVC, and consequently contrasted their performance while diagnosing flaws within the hTetro performance. Besides employing two diverse kinds of SVM, the study tested two diverse kernels: radial basis and linear function. The findings showed that the entire four SVM-based methods attained a classification precision of more than 90% over the seven different shapes at forward and backward locomotion. The general precision of the entire four classifiers was higher than 97%. The c-SVC's RBF kernel and linear kernel of the v-SVC classifiers attained a greater overall precision compared to the other classifiers. Future research will entail integrating extra

sensors into the robot to broaden the range of fault classification and categorize extra fault types. A methodical dynamic model is being created to calculate the consequences of a recognized set of familiar faults on locomotion gaits.

8.5 Towards Artificial Neural Network Based Fault Diagnosis in a Bio-inspired Reconfigurable Robot

While robots are increasingly becoming autonomous in carrying out functions devoid of any form of human involvement, they should also autonomously overcome and detect any faults and errors. This study presents a neural network method towards classifying and detecting faults that was tested on a bio-inspired reconfigurable robot, the Scorpio. The system was meant to classify and identify any faults that occurred when Scorpio was engaged in crawling and rolling activities. Specifically, we classify between faulty and non-faulty conditions across nine different locomotion gaits involving rolling as well as crawling modes over three different speeds. Statistical features extracted from the sensor data comprising of the sum, mean, maximum value, standard deviation, root sum of squares level, variance, peak-to-peak value and root mean square, were employed as input to the neural networks to assist the system in distinguishing between non-faulty and faulty conditions. In computing the hypothesis from the output neuron, the forward propagation algorithm was applied. The sigmoid function was employed in activating the function applied in the forward propagation. Backward propagation was applied in identifying the gradient of weights, and these were later used in updating the neural network's weights. Consequently, the weights were updated up to the point where the cost function was less than threshold value. The results indicate a classification accuracy of over 91% for all of the ANN-based approaches tested over the nine locomotion gaits at three different speeds. Future research will entail integrating extra sensors into the robot to broaden the range of fault classification and categorize extra fault types. A methodical dynamic model is being created to calculate the consequences of a recognized set of familiar faults on locomotion gaits.

Reference

[1] Fujita, M., Kitano, H., & Kageyama, K. (1999). A reconfigurable robot platform.Robotics and Autonomous Systems, 29(2), 119-132.

[2] Michaud, F., Arsenault, M., Bergeron, Y., Cadrin, R., Gagnon, F., Legault, M. A., ... & Bisson, J. (2003, September). Co-design of AZIMUT, a multi-modal robotic platform. In proc. ASME 2003 Design Eng. Tech. Conf. and Computers and Info. In Eng. Conference (pp. 46-50).

[3] Sun, Y., & Ma, S. (2011, September). ePaddle mechanism: Towards the development of a versatile amphibious locomotion mechanism. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on (pp. 5035-5040). IEEE.

[4] Yoon, J., & Ryu, J. (2005, April). A novel reconfigurable ankle/foot rehabilitation robot. In Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on (pp. 2290-2295). IEEE.

[5] Low, K. H., & Yu, J. (2007, December). Development of modular and reconfigurable biomimetic robotic fish with undulating fin. In Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on (pp. 274-279). IEEE.

[6] SIDDIQI, A., DE WECK, O. L., & IAGNEMMA, K. (2006). Reconfigurability in planetary surface vehicles: Modeling approaches and case study. matrix, 50, 3.

[7] GUOWU, W., JIAN, S., Wang, S., & Luo, H. (2011). Kinematic analysis and prototype of a metamorphic anthropomorphic hand with a reconfigurable palm.International Journal of Humanoid Robotics, 8(03), 459-479.

[8] Aghili, F., & Parsa, K. (2009). A reconfigurable robot with lockable cylindrical joints. Robotics, IEEE Transactions on, 25(4), 785-797.

[9] Chen, I. M. (2001). Rapid response manufacturing through a rapidly reconfigurable robotic workcell. Robotics and Computer-Integrated Manufacturing, 17(3), 199-213.

[10] Voyles, R. M., & Godzdanker, R. (2008, October). Side-slipping locomotion of a miniature, reconfigurable limb/tread hybrid robot. In Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on (pp. 58-64). IEEE.

[11] Xu, H., Tan, D., Zhang, Z., Ma, Y., & Peng, G. (2009, February). An innovative reconfigurable mobile robot with multi-maneuver modes. In Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on (pp. 1659-1664). IEEE.

[12] Parker, G. B., & Nathan, P. J. (2010, July). Concurrently Evolving Sensor Morphology and Control for a Hexapod Robot. In Evolutionary Computation (CEC), 2010 IEEE Congress on (pp. 1-6). IEEE.

[13] Balakrishnan, K., & Honavar, V. (1996, July). On sensor evolution in robotics. In Proceedings of the First International Conference on Genetic Programming (pp. 455-460).

[14] Bugajska, M. D., & Schultz, A. C. (2000). Co-evolution of form and function in the design of autonomous agents: Micro air vehicle project. NAVAL RESEARCH LAB WASHINGTON DC CENTER FOR APPLIED RESEARCH IN ARTIFICIAL INTELLIGENCE.

[15] Djath, K., Dufaut, M., & Wolf, D. (2000). Mobile robot multisensor reconfiguration. In Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE (pp. 110-115). IEEE.

[16] Lichtensteiger, L. (2000). Towards optimal sensor morphology for specific tasks: Evolution of an artificial compound eye for estimating time to contact.Sensor Fusion and Decentralized Control in Robotic Systems III, Proceedings of SPIE, 4196, 138-146.

[17] Alves, J. C., & Cruz, N. A. (2009, August). An FPGA-Based Embedded System for a Sailing Robot. In Digital System Design, Architectures, Methods and Tools, 2009. DSD'09. 12th Euromicro Conference on (pp. 830-837). IEEE.

[18] Spinka, O., Holub, O., & Hanzalek, Z. (2011). Low-cost reconfigurable control system for small UAVs. Industrial Electronics, IEEE Transactions on, 58(3), 880-889.

[19] Rawashdeh, O. A., Chandler, G. D., & Lumpp, J. E. (2005, May). A UAU test and development environment based on dynamic system reconfiguration. InACM SIGSOFT Software Engineering Notes (Vol. 30, No. 4, pp. 1-7). ACM.

[20] Kim, D., & Park, S. (2006). Designing dynamic software architecture for home service robot software. Embedded and Ubiquitous Computing, 437-448.

[21] Yim M., Duff D. G., Roufas K. D. , "PolyBot: a modular reconfigurable robot" , Proceedings IEEE International Conference on Robotics and Automation , vol.1, no., pp.514-520, 2000

[22] Bishop J., Burden S., Klavins E., Kreisberg R., Malone W., Napp N., Nguyen T., "Programmable parts: a demonstration of the grammatical approach to self-organization" IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3684- 3691, 2005

[23] Victor Z., Andrew C., Hod L., "Molecubes: An Open-Source Modular Robotics Kit", Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Self-Reconfigurable Robotics, 2007.

[24] Salemi, B., Moll M., Wei-Min Shen, "SUPERBOT: A Deployable, Multi-Functional, and Modular Self-Reconfigurable Robotic System," Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.3636-3641, 2006

[25] Kyle Gilpin., Keith Kotay., Daniela Rus., Iuliu Vasilescu., "Miche: Modular Shape Formation by Self-Disassembly," The International Journal of Robotics Research, Vol. 27, pp.345-372, March 2008

[26] Sprowitz A., Pouya S., Bonardi S., Van den Kieboom J., Mockel R., Billard A., Dillenbourg P., Ijspeert A.J., "Roombots: Reconfigurable Robots for Adaptive Furniture", IEEE Computational Intelligence Magazine, vol.5, no.3, pp.20-32, Aug. 2010

[27] Groß, R., Bonani, M., Mondada, F., & Dorigo, M. (2006). Autonomous self-assembly in swarm-bots. Robotics, IEEE Transactions on, 22(6), 1115-1130.

[28] Taira, T., Kamata, N., & Yamasaki, N. (2005, August). Design and implementation of reconfigurable modular humanoid robot architecture. InIntelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on (pp. 3566-3571). IEEE.

[29] Li, B., Ma, S., Liu, J., & Wang, Y. (2005, June). Development of a shape shifting robot for search and rescue. In Safety, Security and Rescue Robotics, Workshop, 2005 IEEE International (pp. 31-35). IEEE.

[30] Zhang, H., Wang, W., Deng, Z., Zong, G., & Zhang, J. (2006). A novel reconfigurable robot for urban search and rescue. International Journal of Advanced Robotic Systems, 3(4), 359-366.

[31] Wolfe, K. C., Moses, M. S., Kutzer, M. D., & Chirikjian, G. S. (2012, May). M3 Express: A low-cost independently-mobile reconfigurable modular robot. In Robotics and Automation (ICRA), 2012 IEEE International Conference on (pp. 2704-2710). IEEE.

[32] Harada, K., Susilo, E., Menciassi, A., & Dario, P. (2009, May). Wireless reconfigurable modules for robotic endoluminal surgery. In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on (pp. 2699-2704). IEEE.

[33] Castano, A., Behar, A., & Will, P. M. (2002). The Conro modules for reconfigurable robots. Mechatronics, IEEE/ASME Transactions on, 7(4), 403-409.

[34] N. Tan, N. Rojas, R. Elara Mohan, V. Kee, and R. Sosa, "Nested Reconfigurable Robots: Theory, Design, and Realization," Int. J. Adv. Robot. Syst., vol. 12, no. 7, p. 110, Jul. 2015.

[35] V. Prabakaran, R. E. Mohan, V. Sivanantham, T. Pathmakumar, and S. S. Kumar, "Tackling Area Coverage Problems in a Reconfigurable Floor Cleaning Robot Based on Polyomino Tiling Theory," Appl. Sci., vol. 8, no. 3, p. 342, 2018.

[36] G. W. Housner et al., "Structural control: past, present, and future," J. Eng. Mech., vol. 123, no. 9, pp. 897–971, 1997.

[37] R. K. Mobley, An introduction to predictive maintenance. Butterworth-Heinemann, 2002.

[38] J. Amnson, T. Liang, and E. Turban, "Decision Support Systems and Intelligent Systems." New Jersey: Prentice Hall, 2001.

[39] V. Vapnik, "The nature of statistical learning theory Springer New York Google Scholar," 1995.

[40] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in Proceedings of the fifth annual workshop on Computational learning theory, 1992, pp. 144–152.

[41] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.

[42] J. Platt, "Fast training of support vector machines using sequential minimal optimization. Advances in Kernel Methods Support Vector Learning (pp. 185--208)," AJ, MIT Press. Cambridge, MA, 1999.

[43] O. Barzilay and V. L. Brailovsky, "On domain knowledge and feature selection using a support vector machine," Pattern Recognit. Lett., vol. 20, no. 5, pp. 475–484, 1999.

[44] M. Zacksenhouse, S. Braun, M. Feldman, and M. Sidahmed, "Toward helicopter gearbox diagnostics from a small number of examples," Mech. Syst. Signal Process., vol. 14, no. 4, pp. 523–543, 2000.

[45] N. Cristiani and S. J. Taylor, "An introduction to support vector machines," 2000.

[46] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," IEEE Trans. Neural Networks, vol. 13, no. 2, pp. 415–425, 2002.

[47] R. O. Duda, "Hart PE, Stork DG," Pattern Classification. New York Wiley, 2001.

[48] C. M. Bishop, "Neural networks for pattern recognition: Oxford Univ," Press, N Y, 1995.

[49] T. AYDMJ and R. P. W. Duin, "Pump failure determination using support vector data description," Lect. Notes Comput. Sci., pp. 415–425, 1999.

[50] L. B. Jack and A. K. Nandi, "Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms," Mech. Syst. Signal Process., vol. 16, no. 2–3, pp. 373–390, 2002.

[51] B. Samanta, K. R. Al-Balushi, and S. A. Al-Araimi, "Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection," Eng. Appl. Artif. Intell., vol. 16, no. 7–8, pp. 657–665, 2003.

[52] B. Samanta, "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms," Mech. Syst. Signal Process., vol. 18, no. 3, pp. 625–644, 2004.

[53] Y. Xu and L. Wang, "Fault diagnosis system based on rough set theory and support vector machine," in International Conference on Fuzzy Systems and Knowledge Discovery, 2005, pp. 980–988.

[54] Q. Ren, X. Ma, and G. Miao, "Application of support vector machines in reciprocating compressor valve fault diagnosis," in International Conference on Natural Computation, 2005, pp. 81–84.

[55] V. Sugumaran, V. Muralidharan, and K. I. Ramachandran, "Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing," Mech. Syst. Signal Process., vol. 21, no. 2, pp. 930–942, 2007.

[56] Q. Hu, Z. He, Z. Zhang, and Y. Zi, "Fault diagnosis of rotating machinery based on improved wavelet package transform and SVMs ensemble," Mech. Syst. Signal Process., vol. 21, no. 2, pp. 688–705, 2007.

[57] B.-S. Yang, T. Han, and W.-W. Hwang, "Fault diagnosis of rotating machinery based on multi-class support vector machines," J. Mech. Sci. Technol., vol. 19, no. 3, pp. 846–859, 2005.

[58] B.-S. Yang, D.-S. Lim, and J.-L. An, "Vibration diagnostic system of rotating machinery using artificial neural network and wavelet transform," in COMADEM 2000: 13 th International Congress on Condition Monitoring and Diagnostic Engineering Management, 2000, pp. 923–932.

[59] B.-S. Yang, T. Han, and J. L. An, "ART--KOHONEN neural network for fault diagnosis of rotating machinery," Mech. Syst. Signal Process., vol. 18, no. 3, pp. 645–657, 2004.

[60] B.-S. Yang, S. K. Jeong, Y.-M. Oh, and A. C. C. Tan, "Case-based reasoning system with Petri nets for induction motor fault diagnosis," Expert Syst. Appl., vol. 27, no. 2, pp. 301–311, 2004.

[61] B.-S. Yang, W.-W. Hwang, D.-J. Kim, and A. C. Tan, "Condition classification of small reciprocating compressor for refrigerators using artificial neural networks and support vector machines," Mech. Syst. Signal Process., vol. 19, no. 2, pp. 371–390, 2005

[62] B.-S. Yang, W.-W. Hwang, M.-H. Ko, and S.-J. Lee, "Cavitation detection of butterfly valve using support vector machines," J. Sound Vib., vol. 287, no. 1–2, pp. 25–43, 2005.

[63] A. Widodo, B.-S. Yang, and T. Han, "Combination of independent component analysis and support vector machines for intelligent faults diagnosis of induction motors," Expert Syst. Appl., vol. 32, no. 2, pp. 299–312, 2007.

[64] A. Widodo, B. S. Yang, T. Han, and D. J. Kim, "Fault diagnosis of induction motor using independent component analysis and multi-class support vector machine," in Proceedings of the 11th Asia-Pacific Vibration Conference, 2005, pp. 144–149.

[65] A. Widodo and B.-S. Yang, "Application of nonlinear feature extraction and support vector machines for fault diagnosis of induction motors," Expert Syst. Appl., vol. 33, no. 1, pp. 241–250, 2007.

[66] S.-F. Yuan and F.-L. Chu, "Support vector machines-based fault diagnosis for turbo-pump rotor," Mech. Syst. Signal Process., vol. 20, no. 4, pp. 939–952, 2006.

[67] S. Yuan and F. Chu, "Fault diagnosis based on support vector machines with parameter optimisation by artificial immunisation algorithm," Mech. Syst. Signal Process., vol. 21, no. 3, pp. 1318–1330, 2007.

[68] J. Sun, M. Rahman, Y. S. Wong, and G. S. Hong, "Multiclassification of tool wear with support vector machine by manufacturing loss consideration," Int. J. Mach. Tools Manuf., vol. 44, no. 11, pp. 1179–1187, 2004.

[69] J. Sun, G. S. Hong, M. Rahman, and Y. S. Wong, "The application of nonstandard support vector machine in tool condition monitoring system," in Electronic Design, Test and Applications, Proceedings. DELTA 2004. Second IEEE International Workshop on, 2004, pp. 295–300.

[70] S. Cho, S. Asfour, A. Onar, and N. Kaundinya, "Tool breakage detection using support vector machine learning in a milling process," Int. J. Mach. Tools Manuf., vol. 45, no. 3, pp. 241–249, 2005.

[71] L. Han, L. Ding, J. Yu, Q. Li, and Y. Liang, "Power plant boiler air preheater hot spots detection system based on least square support vector machines," in International Symposium on Neural Networks, 2004, pp. 598–604.

[72] R. Ramesh, M. A. Mannan, A. N. Poo, and S. S. Keerthi, "Thermal error measurement and modelling in machine tools. Part II. Hybrid Bayesian Network support vector machine model," Int. J. Mach. Tools Manuf., vol. 43, no. 4, pp. 405–419, 2003.

[73] I. K. Fodor, "UCRL-ID-148494 A Survey of Dimension Reduction Techniques A survey of dimension reduction techniques," 2002.

[74] R. Fang, "Induction machine rotor diagnosis using support vector machines and rough set," in International Conference on Intelligent Computing, 2006, pp. 631–636.

[75] Y. Li, Y.-Z. Cal, R.-P. Yin, and X.-M. Xu, "Fault diagnosis based on support vector machine ensemble," in Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on, 2005, vol. 6, pp. 3309–3314.

[76] J. H. Williams, A. Davies, and P. R. Drake, Condition-based maintenance and machine diagnostics. Springer Science & Business Media, 1994.

[77] W. Yan and H. Shao, "Application of support vector machine nonlinear classifier to fault diagnoses," in Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on, 2002, vol. 4, pp. 2697–2700.

[78] A. Rojas and A. K. Nandi, "Practical scheme for fast detection and classification of rolling-element bearing faults using support vector machines," Mech. Syst. Signal Process., vol. 20, no. 7, pp. 1523–1536, 2006.

[79] A. Rojas and A. K. Nandi, "Detection and classification of rolling-element bearing faults using support vector machines," in Machine Learning for Signal Processing, 2005 IEEE Workshop on, 2005, pp. 153–158.

[80] Z. Zhang, W. Lv, and M. Shen, "Active learning of support vector machine for fault diagnosis of bearings," in International Symposium on Neural Networks, 2006, pp. 390–395.

[81] S. Poyhonen, M. Negrea, A. Arkkio, H. Hyotyniemi, and H. Koivo, "Fault diagnostics of an electrical machine with multiple support vector classifiers," in Intelligent Control, 2002. Proceedings of the 2002 IEEE International Symposium on, 2002, pp. 373–378.

[82] S. Pöyhönen, A. Arkkio, P. Jover, and H. Hyötyniemi, "Coupling pairwise support vector machines for fault classification," Control Eng. Pract., vol. 13, no. 6, pp. 759–769, 2005.

[83] C. Zhitong, F. Jiazhong, C. Hongpingn, H. Guoguang, and E. Ritchie, "Support vector machine used to diagnose the fault of rotor broken bars of induction motors," in Electrical Machines and Systems, 2003. ICEMS 2003. Sixth International Conference on, 2003, vol. 2, pp. 891–894.

[84] A. Widodo, B.-S. Yang, and S.-M. Lee, "Faults detection and classification of induction motor using wavelet support vector machine," in Korean Society of Power System Engineering (KSPSE) Conference, 2006, pp. 79–84.

[85] A. Widodo, B.-S. Yang, and S.-M. Lee, "Wavelet Support Vector Machine for Machine Faults Classification," in Lecture Series on Computer and Computational Sciences, 8 (2007), 2007, vol. 8, pp. 1–5.

[86] J. Gao, W. Shi, J. Tan, and F. Zhong, "Support vector machines based approach for fault diagnosis of valves in reciprocating pumps," in Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on, 2002, vol. 3, pp. 1622–1627.

[87] Z. Zhang, Q. Hu, and Z. He, "Fuzzy support vector machine and its application to mechanical condition monitoring," in International Symposium on Neural Networks, 2005, pp. 937–942.

[88] C. Batur, L. Zhou, and C.-C. Chan, "Support vector machines for fault detection," in Decision and Control, 2002, Proceedings of the 41st IEEE Conference on, 2002, vol. 2, pp. 1355–1356.

[89] K. Choi, S. M. Namburu, M. S. Azam, J. Luo, K. R. Pattipati, and A. Patterson-Hine, "Fault diagnosis in HVAC chillers," IEEE Instrum. Meas. Mag., vol. 8, no. 3, pp. 24–32, 2005.

[90] M. Rychetsky, S. Ortmann, and M. Glesner, "Support vector approaches for engine knock detection," in Neural Networks, 1999. IJCNN'99. International Joint Conference on, 1999, vol. 2, pp. 969–974.

[91] Z. Hu, Y. Cai, X. He, and X. Xu, "Fusion of multi-class support vector machines for fault diagnosis," in American Control Conference, 2005. Proceedings of the 2005, 2005, pp. 1941–1945.

[92] M. Ge, R. Du, G. Zhang, and Y. Xu, "Fault diagnosis using support vector machine with an application in sheet metal stamping operations," Mech. Syst. Signal Process., vol. 18, no. 1, pp. 143–159, 2004.

[93] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "A fast iterative nearest point algorithm for support vector machine classifier design," IEEE Trans. neural networks, vol. 11, no. 1, pp. 124–136, 2000.

[94] V. Franc and V. Hlaváč, "An iterative algorithm learning the maximal margin classifier," Pattern Recognit., vol. 36, no. 9, pp. 1985–1996, 2003.

[95] Nansai S, Rojas N, Elara MR, Sosa R. Exploration of adaptive gait patterns with a reconfigurable linkage mechanism. In: 2013 IEEE-RSJ international conference on intelligent robots and systems (IROS). IEEE, Japan. 2013; p. 4661–8.

[96] Yim M, Shen W-M, Salemi B, Rus D, Moll M, Lipson H, Klavins E, Chirikjian GS. Modular self-reconfigurable robot systems [grand challenges of robotics]. IEEE Robot Autom Mag. 2007;14(1):43–52. 3. Murata S, Kurokawa H. Self-reconfigurable robots. IEEE Robot Autom Mag. 2007;14(1):71–8.

[97] Moubarak P, Ben-Tzvi P. Modular and reconfigurable mobile robotics. Robot Auton Syst. 2012;60(12):1648–63.

[98] King RS. BiLBIQ: a biologically inspired robot with walking and rolling locomotion, biosystems and biorobotics, vol. 2. Springer. 2013. http://www.springer.com/engineering/robotics/ book/978-3-642-34681-1.

[99] Nemoto T, Mohan RE, Iwase M. Realization of rolling locomotion by a wheel-spider-inspired hexapod robot. Robot Biomim. 2015;2(1):1–16..

[100] Nemoto T, Mohan RE, Iwase M. Energy-based control for a biologically inspired hexapod robot with rolling locomotion. Digit Commun Netw. 2015;1(2):125–33.

[101] Sinha A, Tan N, Mohan RE. Terrain perception for a reconfigurable biomimetic robot using monocular vision. Robot Biomim. 2014;1(1):1–11.

[102] Dwyer B. Wall following robot. Diss. Worcester Polytechnic Institute; 2013.

[103] Elfes A. A sonar-based mapping and navigation system. In: Proceedings 1986 IEEE international conference on robotics and automation, vol. 3. 1986; p. 1151.

[104] Elfes A. Occupancy grids: a probabilistic framework for mobile robot perception and navigation. 1989.

[105] Elfes A. Using occupancy grids for mobile robot perception and navigation. Computer. 1989;22(6):46–57.

[106] Borenstein J, Koren Y. High-speed obstacle avoidance for mobile robots. In: Proceedings of the 1988 IEEE international symposium on intelligent control. 24–26 Aug 1988; p. 382, 384.

[107] Koren Y, Borenstein J, Potential field methods and their inherent limitations for mobile robot navigation. In: Proceedings of the 1991 IEEE international conference on robotics and automation, vol. 2. 9–11 Apr 1991; p. 1398, 1404.

[108] Borenstein J, Koren Y. Real-time obstacle avoidance for fast mobile robots in cluttered environments. In: Proceedings of the 1990 IEEE international conference on robotics and automation, vol. 1. 13–18 May 1990; p. 572, 577.

[109] Borenstein J, Koren Y. The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Trans Robot Autom. 1991;7(3):278–88.

[110] Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance. IEEE Robot Autom Mag. 1997;4(1):23,33.

[111] Ulrich I, Borenstein J. VFH+: reliable obstacle avoidance for fast mobile robots. In: Proceedings of the 1998 IEEE international conference on robotics and automation, vol. 2. 16–20 May 1998; p. 1572, 1577.

[112] Ulrich I, Borenstein J. VFH*: local obstacle avoidance with look-ahead verification. In: 2000 Proceedings of the ICRA ’00 IEEE international conference on robotics and automation, vol. 3. 2000; p. 2505, 2511.

[113] An D, Wang H. VPH: a new laser radar based obstacle avoidance method for intelligent mobile robots. IEEE Int Conf Intell Control Autom. 2004;5:4681–5.

[114] Gong J, Duan Y, Man Y, Xiong G. VPH+: an enhanced vector polar histogram method for mobile robot obstacle avoidance. In: ICMA 2007. International conference on mechatronics and automation, 2007. 5–8 Aug 2007; p. 2784, 2788.

[115] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings of the 1985 IEEE international conference on robotics and automation, vol. 2. Mar 1985; p. 500, 505.

[116] SICK AG. Telegrams for operating/configuring the LMS 2XX laser measurement systems. Firmware version V.2.10/X1.14. Sick AG: Waldkirch, Germany. 2003.

[117] Lanz-Cortiella R, Llorens-Calveras J, Rosell-Polo JR, Gregorio-Lopez E, Palacin-Roca J. Characterisation of the LMS200 laser beam under the influence of blockage surfaces. Influence on 3D scanning of tree orchards. Sensors. 2011;11(3):2751–72.

[118] Tan N, Mohan RE and Watanabe A. Towards a framework for robot-inclusive environments. Autom Constr 2016; 69: 68–78.

[119] Armour R, Paskins K, Bowyer A, et al. Jumping robots: a biomimetic solution to locomotion across rough terrain. Bioinspir Biomim 2007; 2: S65–S82.

[120] Crespi A, Karakasiliotis K, Guignard A, et al. Salamandra Robotica II: An amphibious robot to study salamander-like swimming and walking gaits. IEEE Trans Rob 2013; 29: 308–320.

[121] Phipps CC, Shores BE and Minor MA. Design and quasistatic locomotion analysis of the rolling disk biped hybrid robot. IEEE Trans Rob 2008; 24: 1302–1314.

[122] Webb B and Consilvio T. Biorobotics. MIT Press, 2001.

[123] Wang C, Zhang T, Wei X, et al. Dynamic imbalance analysis and stability control of galloping gait for a passive quadruped robot. Appl Bionics Biomech 2015: 1–17.

[124] Yang T, Zhou C and Mohan RE. A fast vision system for soccer robot. Appl Bionics Biomech 2012; 9: 399–407.

[125] Cui B, Chen L, Wang Z, et al. Design and dynamic analysis of a novel biomimetic robotics hip joint. Appl Bionics Biomech 2015; 2015: 1–8.

[126] Li M, Jiang Z, Wang P, et al. Control of a quadruped robot with bionic springy legs in trotting gait. J Bionic Eng 2014; 11: 188–198.

[127] Ho T, Choi S and Lee S. Development of a biomimetic quadruped robot. J Bionic Eng 2007; 4: 193–199.

[128] Shi L, Guo S, Mao S, et al. Development of an amphibious turtle-inspired spherical mother robot. J Bionic Eng 2013; 10: 446–455.

[129] Chen J, Liu Y, Zhao J, et al. Biomimetic design and optimal swing of a hexapod robot leg. J Bionic Eng 2014; 11: 26–35.

[130] Wei H, Cui Y, Li H, et al. Kinematics and the implementation of a modular caterpillar robot in trapezoidal wave locomotion. Int J Adv Rob Syst 2013; 10(304): 1–11.

[131] Saravanan, N.; Siddabattuni, V.N.S.K.; Ramachandran, K.I. Fault diagnosis of spur bevel gear box using artificial neural network (ANN) and proximal support vector machine (PSVM). Appl. Soft Comput. 2009, 10.

[132] Saimurugan, M.; Ramachandran, K.I.; Sugumaran, V.; Sakthivel, N.R. Multi component fault diagnosis of rotational mechanical system based on decision tree and support vector machine. Expert Syst. Appl. 2011, 38.

[133] Satoh, S.; Shaikh, M.S.; Dote, Y. Fast Fuzzy Neural Network for Fault Diagnosis of Rotational Machine Parts using General Parameter Learning and Adaptation. In Proceedings of the IEEE Mountain Workshop on Soft Computing in Industrial Applications, Blacksburg, VA, USA, 25–27 June 2001.

[134] Kandar, P.K.; Sharma, S.C.; Harsha, S.P. Fault diagnosis of ball bearings using machine learning methods.Expert Syst. Appl. 2011, 38.

[135] Fonod, R.; Henry, D.; Charbonnel, C.; Bornschleg, E. A Class of Nonlinear Unknown Input Observer for Fault Diagnosis: Application to Fault Tolerant Control of an Autonomous Spacecraft. In Proceedings of the 10th International Conference on Control, Loughborough, UK, 9–11 July 2014. Appl. Sci. 2017, 7, 1025 17 of 17

[136] Xu, L.; Tseng, H.E. Robust Model-Based Fault Detection for a Roll Stability Control System. IEEE Trans. Control Syst. Technol. 2007, 15, 3.

[137] Peng, H.; Chiang, P. Control of Mechatronics Systems Ball Bearing Fault Diagnosis Using Machine Learning Techniques. In Proceedings of the 2011 8th Asian Control Conference (ASCC), Kaohsiung, Taiwan, 15–18 May 2011.

[138] Karthi, S.P.; Shanthi, M.; Bhuvaneswari, M.C. Parametric Fault Diagnosis in Analog Circuit Using Genetic Algorithm. In Proceedings of the IEEE International Conference on Green Computing, Communication and Electrical Engineering (ICGCCEE), Coimbatore, India, 6–8 March 2014.

[139] Antonelli, G.; Caccavale, F.; Sansone, C.; Villani, L. Fault Diagnosis for AUVs using Support Vector Machines. In Proceedings of the 2004 IEEE International Conference on Robotlcs 8 Automation, New Orleans, LA, USA, 26 April–1 May 2004.

[140] Matsuno, T.; Huang, J.; Fukuda, T. Fault Detection Algorithm for External Thread Fastening by Robotic Manipulator Using Linear Support Vector Machine Classifier. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013.

[141] Ma, L.; Zhao, J.; Wang, J.; Wang, S. Fault Diagnosis of Hydraulic System of Quadruped Robot by SVM Based on Rough Set and CS Algorithm. In Proceedings of the 34th Chinese Control Conference, Hangzhou, China, 28–30 July 2015.

[142] Lin, J.; Jiang, J. Fault Detection and Analysis of Control Software for a Mobile Robot. In Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA), Jinan, China, 16–18 October 2006.

[143] Chen, S.; Huang, K.; Chen, W.; Shen, S.; Li, C.; Lin, P. Quattroped: A Leg–Wheel Transformable Robot. IEEE ASME Trans. Mech. 2014, 19, 2.

[144] Nielsen, M.C.; Eidsvik, O.A.; Blanke, M.; Schjølberg, I. Validation of Multi-Body Modelling Methodology for Reconfigurable Underwater Robots. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 18–22 September 2016.

[145] Do, H.M.; Kim, G.H.; Choi, T.; Kim, D.H.; Son, Y. Development of Simulation Model for Modular and Reconfigurable Robots. In Proceedings of the 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAl), Xi'an, China, 19–22 August 2016.

[146] Sugiyama, Y.; Hirai, S. Crawling and Jumping by a Deformable Robot. Int. J. Robot. Res. 2006, 25, 603–620.

[147] Yanagida, T.; Mohan, R.E.; Pathmakumar, T.; Elangovan, K.; Iwase, M. Design and Implementation of a Shape Shifting Rolling-Crawling-Wall-Climbing Robot. Appl. Sci. 2017, 7, 342.

[148] Tan, N.; Mohan, R.E.; Elangovan, K. A Bio-inspired Reconfigurable Robot. In Advances in Reconfigurable Mechanisms and Robots II; Springer: Dordrecht, The Netherlands, 2016.

[149] Tan, N.; Mohan, R.E.; Elangovan, K. Scorpio: A biomimetic reconfigurable rolling–crawling robot. Int. J. Adv. Robot. Syst. 2016, 13.

[150] Masataka, F.; Mohan, R.E.; Tan, N.; Nakamura, A.; Pathmakumar, T. Terrain Perception in a Shape Shifting Rolling-Crawling Robot. Robotics 2016, 19.

[151] Chang, C.; Lin, C. A Library for Support Vector Machines. ACM Trans. Intell. Syst. Technol. 2011, 2, 27.

[152] Nemoto T, Mohan RE, Nansai S, et al. Wheel spider with rolling locomotion: Modeling and simulation. In: 6th international conference on automation, robotics and applications (ICARA), 17–19 February 2015, Queenstown, New Zealand, pp. 337–342. IEEE.

[153] Chadil N, Phadoongsidhi M, Suwannasit K, et al. A reconfigurable spherical robot. In: IEEE international conference on robotics and automation (ICRA), Shanghai, China, 9–13 May 2011, Shanghai, China, pp.2380–2385. IEEE.

[154] Sun Y and Ma S. Decoupled kinematic control of terrestrial locomotion for an ePaddle-based reconfigurable amphibious robot. In: IEEE international conference on robotics and automation (ICRA), Shanghai, China, pp.1223–1228.

[155] Prostak S. Cebrennus rechenbergi: Cartwheeling spider discovered in Morocco, http://www.sci-news.com/biology/sci ence-cebrennus-rechenbergi-spider-morocco-01903.html (2014, accessed 28 February 2016).

[156] Ja¨ger P. Cebrennus Simon, 1880 (Araneae: Sparassidae): A revisionary up-date with the description of four new species and an updated identification key for all species. Zootaxa 2014; 3790: 319–356.

[157] Murray RM, Li Z and Sastry SS. A Mathematic Introduction to Robotic Manipulation. Boca Raton: CRC Press, 1994.

[158] Chen G, Wang H and Lin Z. Determination of the identifiable parameters in robot calibration based on the POE formula. IEEE Trans Rob 2014; 30: 1066–1077.

[159] App inventor, http://ai2.appinventor.mit.edu (accessed 12 December 2014).

[160] http://www.marketsandmarkets.com/Market-Reports/cleaning-robot-market-22726569.html

[161] Ulrich, Iwan, Francesco Mondada, and J-D. Nicoud. "Autonomous vacuum cleaner." Robotics and autonomous systems 19, no. 3-4 (1997): 233-245.

[162] Kakudou, Takahisa, Keigo Watanabe, and Isaku Nagai. "Mobile mechanism of a climbing robot for cleaning and locomotion on stairs." In SICE Annual Conference (SICE), 2012 Proceedings of, pp. 145-148. IEEE, 2012.

[163] Palacin, Jordi, José Antonio Salse, Ignasi Valgañón, and Xavi Clua. "Building a mobile robot for a floor-cleaning operation in domestic environments." IEEE Transactions on instrumentation and measurement 53, no. 5 (2004): 1418-1424.

[164] Lee, Tae-kyeong, Seongsoo Lee, and Se-young Oh. "A hierarchical RBPF SLAM for mobile robot coverage in indoor environments." In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 841-846. IEEE, 2011.

[165] Jeong, WooYeon, and Kyoung Mu Lee. "CV-SLAM: A new ceiling vision-based SLAM technique." In Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on, pp. 3195-3200. IEEE, 2005.

[166] Vallivaara, Ilari, Janne Haverinen, Anssi Kemppainen, and Juha Röning. "Magnetic field-based SLAM method for solving the localization problem in mobile robot floor-cleaning task." In Advanced Robotics (ICAR), 2011 15th International Conference on, pp. 198-203. IEEE, 2011.

[167] Zhao, Zheng, Weihai Chen, Chen CY Peter, and Xingming Wu. "A novel navigation system for indoor cleaning robot." In Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on, pp. 2159-2164. IEEE, 2016.

[168] Sahin, Haydar, and Levent Guvenc. "Household robotics: autonomous devices for vacuuming and lawn mowing [Applications of control]." IEEE Control Systems 27, no. 2 (2007): 20-96.

[169] Sprunk, Christoph, Jörg Röwekämper, Gershon Parent, Luciano Spinello, Gian Diego Tipaldi, Wolfram Burgard, and Mihai Jalobeanu. "An experimental protocol for benchmarking robotic indoor navigation." In Experimental Robotics, pp. 487-504. Springer, Cham, 2016.

[170] Rhim, Sungsoo, Jae-Chang Ryu, Kwang-Ho Park, and Soon-Geul Lee. "Performance evaluation criteria for autonomous cleaning robots." In Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on, pp. 167-172. IEEE, 2007.

[171] Kümmerle, Rainer, Bastian Steder, Christian Dornhege, Michael Ruhnke, Giorgio Grisetti, Cyrill Stachniss, and Alexander Kleiner. "On measuring the accuracy of SLAM algorithms." Autonomous Robots 27, no. 4 (2009): 387.

[172] Yim, Mark, et al. "Self-reconfigurable robot systems: PolyBot." Journal of the Robotics Society of Japan 21.8 (2003): 851-854.

[173] Castano, Andres, Ramesh Chokkalingam, and Peter Will. "Autonomous and self-sufficient conro modules for reconfigurable robots." Distributed autonomous robotic systems 4. Springer, Tokyo, 2000. 155-164.

[174] Kee, Vincent, et al. "Hinged-Tetro: A self-reconfigurable module for nested reconfiguration." 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. IEEE, 2014.

[175] Prabakaran, Veerajagadheswar, et al. "hTetro: A Tetris-inspired shape shifting floor cleaning robot." Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 2017.

[176] Yuan, Xianfeng, et al. "A novel Mittag-Leffler kernel based hybrid fault diagnosis method for wheeled robot driving system." Computational intelligence and neuroscience 2015 (2015): 65.

[177] Liu, Zhiwen, et al. "Multi-fault classification based on wavelet SVM with PSO algorithm to analyze vibration signals from rolling element bearings." Neurocomputing 99 (2013): 399-410.

[178] Samanta, B., and K. R. Al-Balushi. "Artificial neural network based fault diagnostics of rolling element bearings using time-domain features." Mechanical systems and signal processing17.2 (2003): 317-328.

[179] Li, Bo, et al. "Neural-network-based motor rolling bearing fault diagnosis." IEEE transactions on industrial electronics 47.5 (2000): 1060-1069.

[180] Freyermuth, Bernd. "An approach to model based fault diagnosis of industrial robots." Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on. IEEE, 1991.

[181] Zhang, Liang, Lindsay B. Jack, and Asoke K. Nandi. "Fault detection using genetic programming." Mechanical Systems and Signal Processing 19.2 (2005): 271-289.

[182] G. Frederickson, Hinged Dissections: Swinging and Twisting. Cambridge University Press, 2002.

[183] E. Demaine, M. Demaine, D. Eppstein, G. Frederickson, and E. Friedman, “Hinged dissection of polyominoes and polyforms,” Computational Geometry, vol. 31, no. 3, pp. 237 – 262, 2005.

[184] Erik D. Demaine1, Martin L. Demaine1, Jeffrey F. Lindy, and Diane L. Souvaine3, “Hinged dissection of polypolyhedra,” Volume 3608 pp 205-217.

[185] Sarhangi, Reza. "Making patterns on the surfaces of swing-hinged dissections." Bridges Leeuwarden Proceedings (2008): 251-25.

[186] Y. Zhang, X. Ding, Y. Liu, P. J. Griffin. An artificial neural network approach to transformer fault diagnosis. IEEE Transactions on Power Delivery (Volume: 11, Issue: 4, Oct 1996)

[187] Achmad Widodo, Bo-Suk Yang. Support vector machine in machine condition monitoring and fault diagnosis. Mechanical Systems and Signal Processing 21 (2007) 2560–2574

[188] Long Zhang, Guoliang Xiong, Hesheng Liu, Huijun Zou, Weizhong Guo. Bearing fault diagnosis using multi-scale entropy and adaptive neuro-fuzzy inference. Expert Systems with Applications Volume 37, Issue 8, August 2010, Pages 6077-6085

[189] Hui Luo, Youren Wang, Jiang Cui. A SVDD approach of fuzzy classification for analog circuit fault diagnosis with FWT as preprocessor. Expert Systems with Applications 38 (2011) 10554–10561

[190] Gianluca Antonelli, Fabrizio Caccavale, Carlo Sansone and Luigi Villani, DIAGNOSIS OF ACTUATOR FAULTS IN AUVs BASED ON NEURAL NETWORKS.

[191] Yoo-Seok Kim, Gwang-Pil Jung, Haan Kim, Kyu-Jin Cho and Chong-Nam Chu. Wheel Transformer: A Wheel-Leg Hybrid Robot With Passive Transformable Wheels. IEEE TRANSACTIONS ON ROBOTICS, VOL. 30, NO. 6, DECEMBER 2014

[192] M. Caccia and G. Veruggio. Guidance and control of a recon"gurable unmanned underwater vehicle.Control Engineering Practice Volume 8, Issue 1, January 2000, Pages 21-37.

[193] Alessandro Crespi, Daisy Lachat, Ariane Pasquier and Auke Jan Ijspeert. Controlling swimming and crawling in a fish robot using a central pattern generator. Auton Robot (2008) 25: 3–13

[194] K. Elangovan, K. T. S. Yokhesh, R. E. Mohan, M. Iwase, T. Nemoto, K. Wood, Fault diagnosis of a reconfigurable crawlingrolling robot based on support vector machines, Applied Sciences.