

Prismatic Modular Robotics Enabled Through Active and Passive Elements

Weibing Li

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy

The University of Leeds
Institute of Design, Robotics and Optimisation
School of Mechanical Engineering

March, 2018

Intellectual Property Statement

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

The right of Weibing Li to be identified as Author of this work has been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

© 2018 The University of Leeds and Weibing Li.

Acknowledgements

I would like to thank all the people who helped me directly or indirectly throughout this project. First and foremost I would like to thank sincerely my supervisors Professor Robert C. Richardson and Doctor Jongrae Kim for their patience and effort spent on providing me required facilities and constructive suggestions during this study. Without their guidance, support and encouragement, this project could not be accomplished. I want to thank the thesis viva examiners Professor Abbas A. Dehghani-Sanij and Doctor Pejman Iravani for their inspirational comments on improving the thesis. I wholeheartedly thank the University of Leeds for offering me a full research scholarship which makes it possible for me to live and study in Leeds.

I would express my deepest gratitude here to my great parents and my sister who keep offering me unconditional support. In particular, I am thankful to Miss Shao, a kind-hearted and positive girl, who has been accompanying me to this destination. Undoubtedly, their love motivates me to keep pursuing a brighter academic future.

Moreover, I want to acknowledge technical staff from the robotics laboratory and the mechanical workshop for their help ordering mechatronics components and manufacturing mechanical parts for this project.

Furthermore, I am grateful that I can conduct research with a lot of amazing officemates with different backgrounds from different countries. Thanks for them giving me a lot of discussions on research and also on other interesting topics, which makes me have some leisure time to keep sane during the arduous journey.

Last but not least, I sincerely thank researchers and engineers who keep updating and sharing their open-source projects associated with software and hardware development on the Internet, which definitely boosts the efficiency of overcoming obstacles and difficulties throughout this research.

Abstract

Robotic involvement is envisaged for exploration of human-inaccessible areas such as planetary space, confined and unstructured environments, and radioactive places. An exploration mission usually includes multiple tasks that are difficult or even impossible to finish using a single robot. Modular robots aim to solve this problem by providing a robotic system wherein robotic modules can be reconfigured to accomplish diverse tasks.

In this work, research is undertaken on the design, manufacturing and control of a modular robotic system consisting of straight extending modules. Each robotic module of the modular robot can be actively controlled or can respond passively to external forces. The modular elements can be connected simply for ease of manual reconfiguration.

A new connectivity strategy for building modular robotic structures using rigid connector nodes, active and passive modular elements is investigated. Comparisons of the new connectivity and a conventional connectivity using compliant connector nodes are made with respect to kinematics, locomotion and deformation of some robotic structures. Modular units including a prismatic actuator, a rigid connector node and a passive revolute joint are then designed, manufactured and tested. More modular elements are further replicated for building modular robotic structures leading to a final prototype system with eight prismatic actuators, four rigid connector nodes and four passive revolute joints. Each prismatic actuator is equipped with a locking mechanism and possesses three different working states: it can either be actuated, locked or passive. The three-state prismatic actuator is self-contained with its own computation, communication, actuation and sensing capabilities.

A proportional-integral-derivative (PID) controller is implemented to control the position of the prismatic actuator. The actuation and locking forces of the prismatic actuator are experimentally evaluated. The prismatic actuator can vertically lift an external load of 29.4 N. The locking force of the mechanical locker is 78.6 N, enabling the actuator to be capable of vertically supporting a weight of about 2.5 kg in the locked state. The minimum force required to passively move the prismatic actuator is also measured as

8.34 N. The performance of the PID controller, three states and state transitions of the prismatic actuator are then validated by a series of physical experiments. Experimental results demonstrate that the maximum absolute value of the displacement error is to be 0.175 mm in the actuated state, and state transitions between actuated, locked and passive states are physically achievable. Moreover, state transitions of two and multiple prismatic actuators are also realized resorting to communications between the prismatic actuators.

As a high-level control strategy, a central pattern generator (CPG) neural network is first applied to modular robotic structures composed of the fabricated robotic modules. Physical experiments show that the modular robotic structures achieve a worm-like locomotion gait through the coordination of their actuators' movements, substantiating the feasibility and effectiveness of the mechanical design and control strategy. Modular robotic structures with greater number of elements are constructed in a physics-based robot simulator. A generalized CPG neural network and a role-based control method are developed for controlling these simulated modular robots. Computer simulations are then conducted to further demonstrate locomotion capability of modular robotic structures composed of three-state prismatic actuators. Simulation results show that the generalized CPG method is scalable to a broad range of robotic structures with different number of modules. The three-state prismatic actuator can be applied to releasing physical constraints of a robotic structure during task execution and achieving a walking pattern by using state transitions.

CONTENTS

1	Introduction	1
1.1	Background	2
1.2	Motivation for Research	4
1.3	Research Aim	6
1.4	Statement of Contributions	7
1.5	Thesis Outline	8
2	Literature Review	11
2.1	Introduction	12
2.2	Block-Type Modules	15
2.3	Strut-Type Modules	32
2.4	Control Strategies	40
2.5	Communication Methods	43
2.6	Passive Compliance and Multiple-State Actuators	45
2.7	Conclusions	47
3	Connectivity of Strut-Type Modular Robots	50
3.1	Connectivities	51
3.2	Kinematics Analysis	53
3.2.1	Degrees of Freedom	53
3.2.2	Kinematics	55
3.3	Simulation and Analysis	59
3.3.1	Workspace and Manipulability	60
3.3.2	Robot Locomotion and Deformation	65
3.4	Conclusions	71

4	Mechatronics Designs and Physical Prototypes	72
4.1	Design Analysis	73
4.1.1	Connector Node	73
4.1.2	Actuation Mechanism	74
4.1.3	Locking Mechanism	76
4.1.4	Design Requirements	77
4.2	Mechatronics Designs and Hardware Implementation	78
4.2.1	Three-State Prismatic Actuator	78
4.2.2	Rigid Connector Node and Passive Revolute Joint	85
4.2.3	Physical Prototypes	87
4.3	Mechatronics Implementation Considerations	94
4.3.1	Lead Screw	94
4.3.2	Shafts and Bearings	97
4.3.3	Microcontroller	97
4.3.4	Motors	100
4.3.5	Motor Drivers	108
4.3.6	Sensors	109
4.3.7	Batteries, Switches and Voltage Regulator	111
4.3.8	Communication Module	111
4.4	Conclusions	112
 5	 Experimental Validation of Prismatic Actuators	 113
5.1	Overview	114
5.2	Experimental Setups	114
5.2.1	Actuated State	115
5.2.2	Locked State	117
5.2.3	Passive State	119
5.2.4	Three States of an Individual Prismatic Actuator	120
5.2.5	Communication and State Transitions of Multiple Actuators	121
5.3	Experimental Results	124
5.3.1	Actuated State	125
5.3.2	Locked State	127
5.3.3	Passive State	130
5.3.4	Three States of an Individual Prismatic Actuator	131

5.3.5	Communication and State Transitions of Multiple Actuators . . .	133
5.4	Conclusions	138
6	Control and Locomotion of Modular Robots	139
6.1	Amorphous Robots	140
6.1.1	Single-Module Worm Robot	140
6.1.2	Two-Module Worm Robot	147
6.1.3	Four-Module Square Robot	148
6.1.4	Movements of a 4-PRP Parallel Mechanism	154
6.1.5	Generalized Computer Simulations	157
6.2	Cross-Shaped Robot	166
6.3	Simulated Table-Shaped Robot	169
6.4	Conclusions	174
7	Conclusions and Future Work	175
7.1	Assessment of Research Objectives	176
7.1.1	Literature Review on Related Work	176
7.1.2	Development of Connectivity Strategy	176
7.1.3	Design and Fabrication of Modular Elements	176
7.1.4	Investigation of Communication Strategies	177
7.1.5	Construction of Simulated Modular Robots	177
7.1.6	Implementation of Control Strategies for Locomotion	177
7.2	Conclusions	177
7.2.1	Design	177
7.2.2	Modelling	179
7.2.3	Control	179
7.3	Future Directions	180
7.3.1	Design	180
7.3.2	Modelling	181
7.3.3	Control	182
A	Paper Published	183
B	Comparison of Modular Robotic Modules	193

C Experimental Data	200
C.1 Stall Current and Stall Torque of the 775 DC Motor	201
C.2 Electromagnetic Brake Test Results	202
C.3 Calibration Results of the Force Sensor	202
C.4 Measuring Results of the Mechanical Locker Force	204
C.5 Passive State Test Results	204
C.6 Communication Test Results	204
D Software Implementation for Experiments	207
D.1 Codes for Measuring Locking Force	208
D.1.1 Arduino Codes	208
D.1.2 Matlab Codes	215
D.2 Class Diagrams for Physical Experiments	223
E Software Implementation for Simulations	225
E.1 Class Diagrams for Amorphous Robots	226
E.2 Class Diagrams for the Table-Shaped Robot	227
References	229

LIST OF FIGURES

1.1	Envisaged Scenarios of a Modular Tetrahedral Robot [Curtis <i>et al.</i> , 2007a].	3
1.2	A Conceptual Reconfigurable Robotic Arm with Lockable Joints [Merat <i>et al.</i> , 2013].	5
1.3	Research Flowchart of the Modular Robot Project.	9
2.1	Illustrative Examples of Block- and Strut-Type MRSs.	14
2.2	Prototype and Components of YaMoR Robotic Module [Maye, 2007].	16
2.3	Molecube Robotic Module [Zykov <i>et al.</i> , 2005, 2007b].	17
2.4	Molecubes Actuation Module [Zykov <i>et al.</i> , 2007a, 2008].	18
2.5	ATRON Robotic Module [Christensen <i>et al.</i> , 2010a, Østergaard <i>et al.</i> , 2006].	20
2.6	MTRAN III Robotic Module [Kurokawa <i>et al.</i> , 2006, 2008, Murata <i>et al.</i> , 2007, 2002].	21
2.7	Dtto Robotic Module [Hackaday, 2016].	23
2.8	SuperBot Robotic Module [Salemi <i>et al.</i> , 2006, Shen <i>et al.</i> , 2008].	23
2.9	Roombots Robotic Module [Spröwitz <i>et al.</i> , 2014, 2010].	25
2.10	Electronics Arrangement of Roombots [Spröwitz <i>et al.</i> , 2014, 2010].	26
2.11	CAD Models of ModRED I and ModRED II [Hossain, 2014].	27
2.12	Electronics, Elastic Series Actuator and RoGenSiD Docking Mechanism of ModRED II [Hossain, 2014].	28
2.13	Sambot Robotic Module [Tindell <i>et al.</i> , 1994, Wei <i>et al.</i> , 2011].	29
2.14	Prototypes and Actuation Mechanism of SMORES [Davey <i>et al.</i> , 2012, Jing <i>et al.</i> , 2016].	31

2.15 Two Connection Mechanisms of SMORSE [Davey <i>et al.</i> , 2012, Tosun <i>et al.</i> , 2016].	32
2.16 TETROBOT Modular Robot [Mândru & Teuțan, 2005, Teuțan <i>et al.</i> , 2009].	33
2.17 Prototypes of Odin Robot [Garcia <i>et al.</i> , 2009, Lyder <i>et al.</i> , 2008].	34
2.18 CCP Joint, Flexible Connector and Electronics Boards of Odin Robot [Garcia <i>et al.</i> , 2009, Lyder <i>et al.</i> , 2008].	36
2.19 Conceptual Model, Actuation Mechanism, CAD Models and Physical Prototypes of a TET Walker Robot [Curtis <i>et al.</i> , 2006, 2007a,b].	37
2.20 Amorphous Modular Robot [Yu, 2010, Yu <i>et al.</i> , 2010].	39
2.21 OctaWorm Robot [Zagal <i>et al.</i> , 2012].	40
2.22 Three Controller Architectures.	41
3.1 Different Connectivities of a Strut-Type MRS.	52
3.2 Nomenclature of Two Parallel Mechanisms.	54
3.3 Matlab GUIs Created for the Two Parallel Mechanisms.	60
3.4 Workspace and Manipulability of the 4-RPR Mechanism.	61
3.5 Workspace and Manipulability of the 4-PRP Mechanism.	61
3.6 Constant Orientation Workspaces of the 4-PRP Mechanism with $\phi_e = 25^\circ$	62
3.7 Constant Orientation Workspaces of the 4-PRP Mechanism with Different Values of ϕ_e	62
3.8 Circular Path Tracking using the 4-RPR Mechanism.	64
3.9 Circular Path Tracking using the 4-PRP Mechanism.	65
3.10 Diagram of the Employed Moving Principle for Moving Rightward.	66
3.11 Square Robots with Different Connectivities.	67
3.12 Locomotion of a Square Robot with Compliant Nodes.	68
3.13 Locomotion of a Square Robot with Rigid Nodes.	68
3.14 Locomotion of a Two-Square Robot with Compliant Nodes.	69
3.15 Locomotion of a Two-Square Robot with Rigid Nodes.	69
3.16 Deformation of a Fourteen-Square Robot with Compliant Nodes.	70
3.17 Deformation of a Fourteen-Square Robot with Rigid Nodes.	70
4.1 Different Solutions to Connector Nodes.	74
4.2 Different Prismatic Actuation Solutions.	75

LIST OF FIGURES

4.3	Section View of a Three-State Prismatic Actuator with a Dog Clutch. . .	80
4.4	Rendered CAD Models of a Three-State Prismatic Actuator with an Electromagnetic Brake.	82
4.5	CAD Models of the Mechanical Locker.	83
4.6	Rendered Three-State Prismatic Actuator with a Mechanical Locker. . .	84
4.7	Section View of a Rendered Three-State Gear Box.	85
4.8	Connector Nodes and Revolute Joint.	86
4.9	Prototype I of a Three-State Prismatic Actuator.	88
4.10	Aluminium Version of Prototype II.	89
4.11	Mechatronics Parts of the Aluminium Version of Prototype II.	90
4.12	PLA Version of Prototype II.	91
4.13	Mechatronics Parts of the PLA Version of Prototype II.	92
4.14	Electronics Schematic of the PLA Version Prismatic Actuator.	93
4.15	Connector Nodes and Revolute Joint.	94
4.16	Assembly of Eight PLA Version Prismatic Actuators.	95
4.17	Lead Screw Efficiency Profiles.	96
4.18	Different Shafts and Bearings Considered in the Implementation Process.	98
4.19	Different DC Motors Considered in the Implementation Process.	101
4.20	Experimental Test of the DC Motor 5.	103
4.21	Diagrams Associated with the Spiral Cam Mechanism of the Locker. . .	105
4.22	Different Motor Drivers Considered in Implementation.	109
5.1	Experimental Setup for Verifying the Performance of the Actuated State.	115
5.2	Schematic Diagram of a Classic PID Controller.	116
5.3	Experimental Setup for Testing the Electromagnetic Brake.	118
5.4	Experimental Setup for Measuring Force of the Mechanical Locker. . . .	118
5.5	Schematic Diagram of a Circuit for Measuring Forces of the Locker. . .	119
5.6	Experimental Setup for Testing the Passive State.	120
5.7	Experimental Setup for Testing Three States and State Transitions of Individual Prismatic Actuators.	121
5.8	Experimental Setup for Testing the Three States and State Transitions.	122
5.9	Parallel Structures with Multiple Prismatic Actuators.	123
5.10	Step Responses of the Prismatic Actuator Controlled by a PID Controller.	125
5.11	Sinusoidal-Like Trajectory Tracking using the Prismatic Actuator. . . .	126

5.12 Periodical Sinusoidal-Like Trajectory Tracking using the Prismatic Actuator.	127
5.13 Experimental Test Results of the Electromagnetic Brake.	128
5.14 Force Measurement of the Mechanical Locker.	129
5.15 Experimental Test of the Mechanical Locker.	130
5.16 Three States of Prototype II.	132
5.17 Communication of Two Separate Actuators.	134
5.18 State Transitions of Two Separate Actuators.	135
5.19 Motions of Two Parallel Actuators with Actuated and Passive States. . .	136
5.20 Motions of Three Parallel Actuators with Actuated and Passive States. .	136
5.21 Motions of Four Parallel Actuators with Actuated and Passive States. .	137
6.1 A Robotic Module Composed of Two Prismatic Actuators and A Schematic Diagram of the Employed CPG Controller.	141
6.2 Motion Profiles of Two Actuation Functions.	143
6.3 Communication Architecture of the Single-Module Worm Robot.	145
6.4 Phase Coordination and Locomotion of the Single-Module Worm Robot. .	146
6.5 Displacements and Phase Difference of the Two Prismatic Actuators. . .	146
6.6 Communication Architecture of the Two-Module Worm Robot.	147
6.7 Phase Coordination and Locomotion of the Two-Module Worm Robot. .	149
6.8 Actuator Displacements and Phase Difference of the Two-Module Robot. .	150
6.9 A Four-Module Square Robot.	151
6.10 Basic Locomotion of the Square Robot When the Robot Moves Along Direction 2.	152
6.11 Coordination and Locomotion of the Square Robot When the Robot Moves Along Direction 3.	153
6.12 Displacements Measured by Encoders and Phase Difference of the Square Robot's Actuators.	154
6.13 A Physical 4-PRP Parallel Mechanism.	155
6.14 Movements of the 4-PRP Parallel Mechanism.	156
6.15 Circular Path of the End-Effector Platform and Displacements of Prismatic Actuators.	157
6.16 Simulated Amorphous Robot.	158
6.17 Simulation of a Single-Module Worm Robot.	161

6.18 Simulation of a Square Robot.	162
6.19 Motion and Position Error Profiles Synthesized by the Square Robot. . .	162
6.20 Simulation of a Two-Square Robot.	163
6.21 Motion and Position Error Profiles Synthesized by the Four-Square Ro- bot without Using Passive State of Actuators.	164
6.22 Locomotion of a Simulated Four-Square Robot with Passive State of Actuators.	165
6.23 Motion and Position Error Profiles Synthesized by the Four-Square Ro- bot with Passive State of Actuators.	166
6.24 A Cross-Shaped Robot.	167
6.25 Locomotion of the Cross-Shaped Robot.	168
6.26 A Table-Shaped Robot Consisting of 22 Struts.	169
6.27 Flowchart of the Role-Based Control Strategy in the Initialization Stage.	171
6.28 Flowchart of the Role-Based Control Strategy in the Cyclic Motion Stage.	172
6.29 Initialization Stage of the Table-Shaped Robot.	172
6.30 Cyclical Motion Stage of the Table-Shaped Robot for Walking Rightward.	173
6.31 Actuator Motion Profiles of the Table-Shaped Robot.	173
D.1 Class Diagrams of Software for Physical Experiments.	224
E.1 Class Diagrams of Software for Simulated Amorphous Robots.	226
E.2 Class Diagrams of Software for the Simulated Table-Shaped Robot. . . .	227

LIST OF TABLES

4.1	Requirements and Specifications of the Strut-Type Modular Robotic System to Be Developed	79
4.2	Comparison of Arduino Uno, Arduino Nano, PSoC 5LP CY8CKIT-050 and Raspberry Pi 3 Model B	99
4.3	Comparison of Five Different DC Motors	102
4.4	Parameters Related to the Spiral Cam Locking Mechanism	107
4.5	Specifications of 775 and N20 DC Motors	108
4.6	Comparison of L9110S, L298N, MD10C and BTS7960	109
4.7	Specifications of the Employed Encoder	110
4.8	Comparison of nRF24L01+, ESP8266 and HC-05 Wireless Modules	112
5.1	Effects of k_p , k_i and k_d on Performance Indices of a PID Response	116
5.2	Forces Required to Move Different Lead Screws in the Passive State	131
6.1	Related Parameters of the Role-Based Control Strategy	170
B.1	Comparison of Block-Type Modules	194
B.2	Comparison of Strut-Type Modules	199
C.1	Stall Current of the Selected Motor Under Different Voltages	201
C.2	Stall Torque of the Selected Motor Under Different Voltages	201
C.3	Electromagnetic Brake Test Results Under Different Voltages	202
C.4	Calibration Data of the Force Sensor	203
C.5	Measured Locking Forces (N) of the Mechanical Locker	204
C.6	Forces Required to Move Different Lead Screws in the Passive State	205
C.7	Communication Test Results	206

Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
ABS	Acrylonitrile Butadiene Styrene
ACM	Active Connection Mechanism
AF	Activation Function
CAD	Computer Aided Design
CAN	Controller Area Network
CCP	Cubic Closed-Packed
CEBOT	Cell Structured Robot
CMS	Centric Multilink Spherical
CNC	Computer Numerical Control
CPG	Central Pattern Generator
CPR	Cycles Per Revolution
CS	Chip Select
DC	Direct Current
DH	Denavit-Hartenberg
DOF	Degree of Freedom
EP	Electro-Permanent
FPGA	Field Programmable Gate Array
GPS	Global Positioning System
GUI	Graphical User Interface
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IR	Infra-red
ISM	Industrial, Scientific and Medical
MISO	Master-in-Slave-out

MOSI	Master-out-Slave-in
MRS	Modular Robotic System
NASA	National Aeronautics and Space Administration
PCB	Printed Circuit Board
PID	Proportional-Integral-Derivative
PLA	Poly-Lactic Acid
POM	Polyoxymethylene
PSoC	Programmable System on Chip
PWM	Pulse Width Modulation
RC	Radio Control
RF	Radio Frequency
RPM	Revolution Per Minute
SCK	Clock
SMA	Shape Memory Alloy
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter

CHAPTER 1

Introduction

This chapter details the background, motivation, aims and objectives, contributions and the outline of this work.

1.1 Background

Increased robotic involvement is envisaged for exploration of human-inaccessible areas such as planetary space, confined and unstructured environments, and radioactive places. Traditionally, a single piece robot is tailored and designed to explore an antecedently known or unknown environment. For example, the Curiosity rover from National Aeronautics and Space Administration (NASA) is a car-like robot designed for exploring on Mars to determine the planet's habitability [Grotzinger, 2013]. In most cases, an exploration mission includes different tasks that are difficult or even impossible to finish using a single robot. For example, a robotic arm has limited workspace and cannot move around the environment. A mobile robot with large wheels can traverse irregular terrain, but cannot enter a confined space. For space exploration missions, sending multiple robots that each can tackle a task is expensive and challenging as the shipping mass and volume, and the budget are restricted. Modular robotics aims to overcome this problem by providing a robotic system capable of reconfiguring its morphologies (that is, physical shapes) to handle diverse tasks.

The history of modular robotics dates back to 1980s. Fukuda *et al.* proposed and implemented the first modular robot named 'cell structured robot' (CEBOT) in [Fukuda *et al.*, 1988]. A modular robot is built from mechatronics modules that are capable of connecting to or disconnecting from each other either manually or autonomously [Yim *et al.*, 2007]. As a modular robot, the CEBOT possesses three main properties: (1) a CEBOT consists of several cell modules; (2) each cell module is intelligent; and (3) the shape of the assembled CEBOT is dependent on given tasks. Over the last two decades, various modular robots have been designed and prototyped to explore a robust, versatile and low-cost solution [Støy *et al.*, 2010].

A conceptual modular tetrahedral robot composed of extensible struts has been put forward by NASA for space exploration [Curtis *et al.*, 2007a]. As presented in Figure 1.1, the conceptual tetrahedral robot is versatile and dexterous enough to work under extremely hostile and unstructured environments; it can crawl through and stride across a valley, crawl in confined space and climb over an obstacle. The robot is also simulated to achieve self-repair by replacing a broken strut with a functional one. For

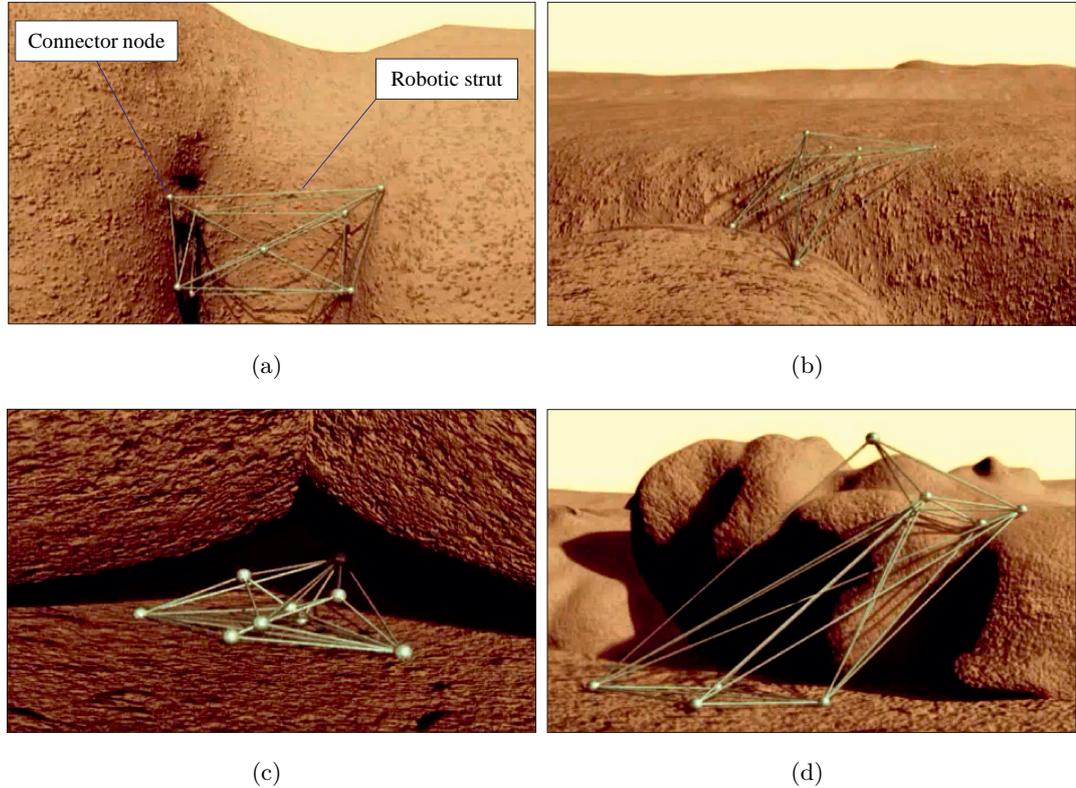


Figure 1.1: Envisaged Scenarios of a Modular Tetrahedral Robot [Curtis *et al.*, 2007a]. (a) Crawl Through a Valley. (b) Stride Across a Valley. (c) Crawl in Confined Space. (d) Climb over an Obstacle.

conventional mobile robots (e.g., the Curiosity rover), accomplishing such tasks could be impossible.

Actuators are responsible for applying forces to move a robot to interact with its working environment. Existing actuators can be categorized as prismatic and revolute actuators that are used to achieve linear straight and rotary motions, respectively. The choice of actuation, be it prismatic or revolute, is a critical factor that needs to be considered when designing a robot. The revolution of robotics has led to an increasing recognition that both the robot controller (i.e., brain) and the robot structure (i.e., body) can have intelligence to some extent [Caluwaerts *et al.*, 2014]. Not only can the brain affect the behaviour of the body, but the body itself also usually has some influence on its behaviour through environment interaction, especially in an unstructured

environment [Owaki *et al.*, 2011]. Following this recognition, various actuators with compliance capabilities have been designed and implemented. There mainly exist two types of robot compliance, i.e., active and passive compliance [Ham *et al.*, 2009, Wang *et al.*, 1998, 2016]. An active compliance system processes sensory data (e.g., force or torque information) received from the environment and then modifies the actuator movements following the software control commands. In general, the position and velocity of an active compliance actuator is controlled by a servo-control system designed to be robust to external forces, which makes it challenging to achieve compliance. An active compliance system has an inherent delayed response due to the time required to measure and respond to the sensory data. In contrast, a passive compliance mechanism is equipped with passive joints that are not motorized. Nevertheless, it can be moved freely and adjust to external forces exerted on it.

A conceptual reconfigurable robotic arm with lockable passive cylindrical joints was proposed for space applications in [Aghili & Parsa, 2009, Merat *et al.*, 2013]. Each cylindrical joint has two passive degrees-of-freedom (DOFs): a prismatic DOF and a revolute DOF. As shown in Figure 1.2(a), the robotic arm is initially folded to save transportation space. By forming a closed-loop kinematic chain shown in Figure 1.2(b), the positions of the cylindrical joints can be passively changed by driving other active actuators. Figure 1.2(c) shows a snapshot of the robotic arm during its reconfiguration state. Once a desired configuration is achieved, the two passive cylindrical joints are locked to serve as rigid links. Finally, the end-effector of the robotic arm can be released to obtain a serial robot as illustrated in Figure 1.2(d). In this way, the robotic arm is capable of changing its Denavit-Hartenberg (DH) parameters and adjusting itself to different tasks.

1.2 Motivation for Research

The simulated tetrahedral robot shown in Figure 1.1 is an idealized truss mechanism consisting of linear extensible struts jointed by connector nodes. Hereafter, this class of modular robots are termed strut-type modular robot. Ideally, this strut-type modular robot has two characteristics: (1) each strut is capable of rotating around its connected node and (2) all the struts linked by a same node share and intersect at a common centre of rotation. Unfortunately, these two requirements result in a fact that it is very difficult to mechanically implement an ideal point-like connector node [Lyder, 2010]. A

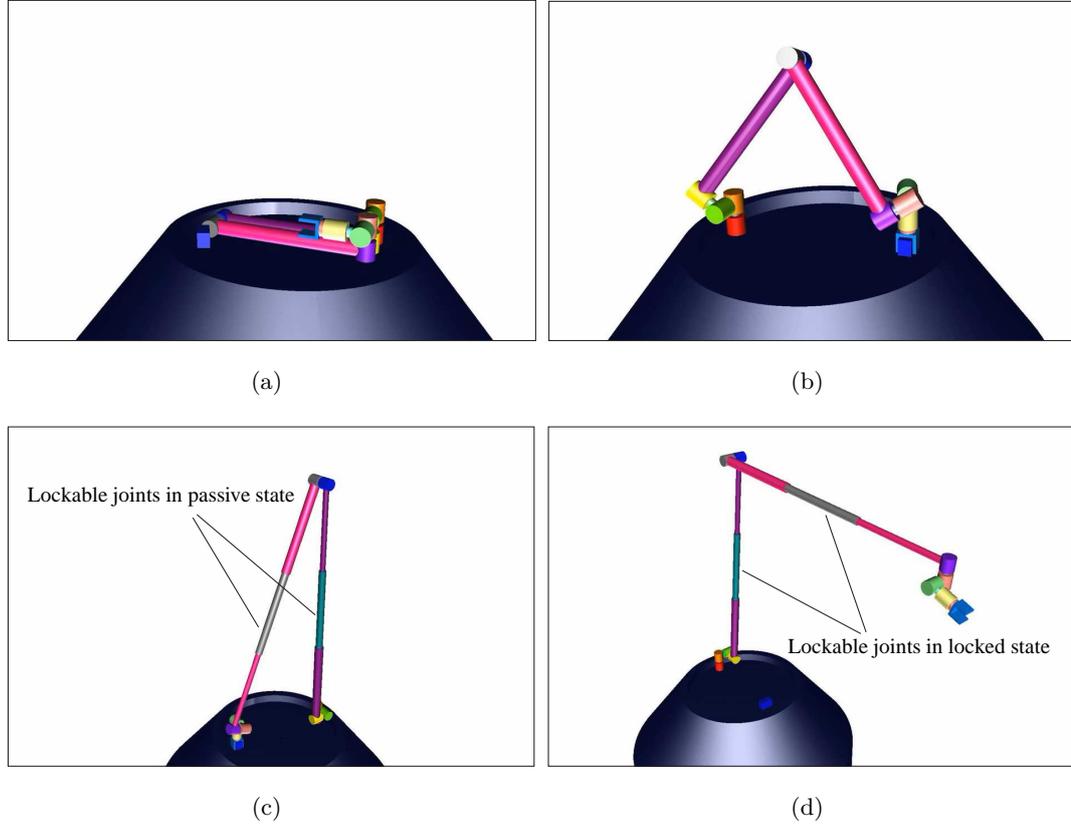


Figure 1.2: A Conceptual Reconfigurable Robotic Arm with Lockable Joints [Merat *et al.*, 2013]. (a) Initial Folded State. (b) Forming a Closed Kinematic Chain. (c) Reconfiguration State with Passive Joints. (d) Final State with Locked Joints.

common workaround connectivity solution is to let each strut rotate around its rotation centre on the node surface using passively movable joints such as ball-and-socket joints or universal joints [Curtis *et al.*, 2007a, Lyder, 2010, Yu, 2010]. This way relatively simplifies the physical implementation, however, it creates more complicated kinematic structures as there exists an offset between the node centre and the rotation centre of a strut. Miniaturizing the workaround connector nodes is difficult since these compliant nodes need space to be equipped with passively movable joints. Therefore, it is worth undertaking research on a new connectivity strategy that can eliminate the offsets and facilitate miniaturization in the future.

To obtain a resultant robotic arm shown in Figure 1.2(d) with desired DH paramet-

ers, multiple steps are involved as discussed before. The most important step shown in Figure 1.2(c) is to drive active actuators of the closed-loop robot to let the lockable passive joints passively achieve their desired displacements. This means the displacements of the active actuators have to be calculated to achieve the desired state of the lockable passive joints. Moreover, since the passive lockable joints have no sensors, their displacements are estimated using a complex control algorithm and the estimation error requires a large time to exponentially approach to zero [Merat *et al.*, 2013]. Evidently, this way of changing DH parameters is indirect, which is not efficient. Hence, it is meaningful to explore an easier and more direct way to achieve similar functions of the lockable passive joints.

1.3 Research Aim

This study aims to undertake research into the design, fabrication and control of a novel strut-type modular robot wherein each robotic strut possesses multiple working states, and the modular units are linked simply to allow manual reconfiguration. To achieve the project aim, the corresponding objectives are outlined as follows.

- 1) To formulate a literature review and analyse state-of-the-art modular robots.
- 2) To study the connectivity strategies of strut-type modular robots and develop novel concepts for simple module interconnection.
- 3) To design and create active and passive modular elements, allowing the robotic element to be capable of being actively controlled and responding passively to external forces, as well as allowing modular elements to be simply connected.
- 4) To replicate more modular elements, build physical modular robots and investigate communication strategies among robotic modules.
- 5) To establish a physics-based simulation environment that allows to analyse more complex modular robots.
- 6) To implement and apply control strategies to the simulated and physical modular robots to achieve communication and locomotion.

1.4 Statement of Contributions

The main contributions and originalities of this work can be categorized as knowledge contribution and technical contribution. Specifically speaking, the following facts contribute to knowledge in the robotics field:

- 1) A new connectivity approach to constructing strut-type modular robots has been put forward to facilitate miniaturization in the future and allow struts jointed by a same connector node to intersect at a same point.
- 2) Modular robotic elements (i.e., prismatic actuators) have been designed and fabricated to allow to be either actuated, locked and passive. Elements for interconnection have been also designed and manufactured to allow simple manual reconfiguration. This is the first time to show a physically achievable solution of a three-state prismatic actuator in the robotics community.

Technically, the work has the following contributions:

- 3) The actuated, locked and passive states of the prismatic actuator have been experimentally validated. State transitions between the three working states have been physically achieved.
- 4) Control strategies have been successfully implemented and applied to achieving communication and locomotion of simulated and physical modular robotic structures. The implementability and feasibility of the developed modular robotic system have been verified by the simulative and experimental results.

Papers Published (Appendix A)

- 1) Li, W., Richardson, R.C., & Kim, J. (2016). A novel strut-type modular robotic structure using rigid node. In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics*, 261–268.

Papers Pending Publication

- 2) Li, W., Richardson, R.C., & Kim, J. Design, implementation and validation of a prismatic actuator with actuation, locking and passive compliance capabilities. To be submitted.

- 3) Li, W., Richardson, R.C., & Kim, J. Design, implementation and control of a new modular robotic system with simulative and experimental verification. To be submitted.

1.5 Thesis Outline

This thesis consists of seven chapters. A chapter first introduces this research. Then, a literature review on related work is formulated. After that, the design, fabrication and control of the modular robotic system are detailed. Finally, conclusions and future directions are covered. For better understanding, Figure 1.3 shows the research flowchart of the whole modular robot project. As shown in the flowchart, the design, fabrication, controller development and experimentation processes are iterative to ensure the efficacy of the developed modular robotic system. A brief breakdown of each chapter is outlined below.

Chapter 1: Introduction

This chapter introduces the background, motivation, aims and objectives, contributions and originalities of this research.

Chapter 2: Literature Review

This chapter mainly conducts a review on the state-of-the-art modular robotic systems. Firstly, design and implementation of typical robotic modules are analysed. Secondly, numerous existing control strategies in modular robotics are studied. Thirdly, popular communication methods are discussed. Finally, robots with passive compliance and actuators with two or more working states are reviewed.

Chapter 3: Connectivity of Strut-Type Modular Robots

This chapter discusses and compares two different connectivity strategies of strut-type modular robots. The kinematics analysis of the two mechanisms formed by using the two connectivity strategies is first conducted. Locomotion and deformation of simulated modular robotic structures constructed using the two connectivity strategies are compared and discussed.

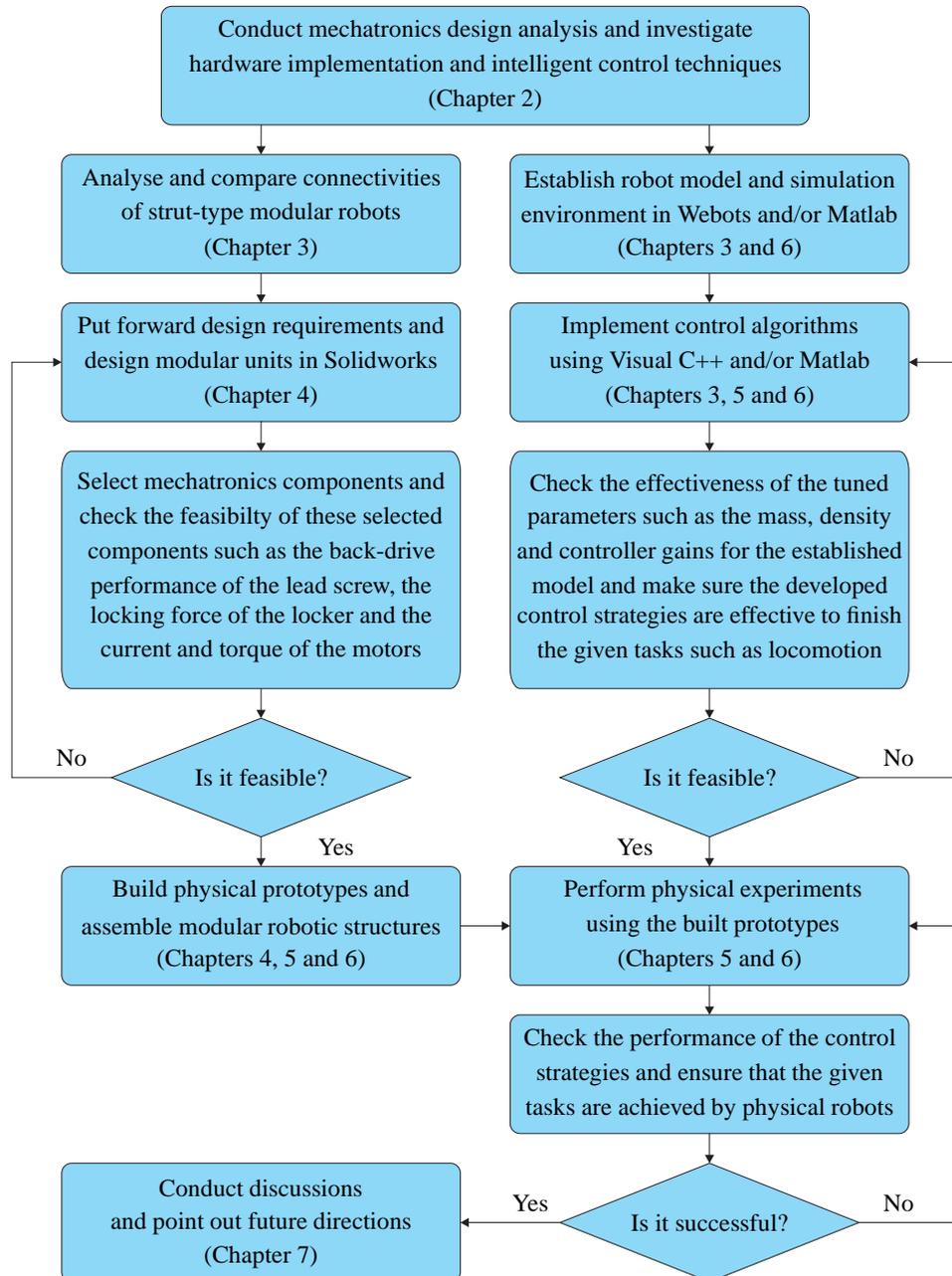


Figure 1.3: Research Flowchart of the Modular Robot Project.

Chapter 4: Mechatronics Designs and Physical Prototypes

This chapter details the mechatronics design and hardware implementation of the modular robotic system. Four design schemes are discussed with computer aided design

(CAD) models presented. Two different prototypes of prismatic actuators are built and compared with design analyses conducted. Meanwhile, a rigid connector node and a passive revolute joint are also designed and manufactured. Finally, more modular units are replicated for building modular robotic structures leading to a prototype robotic system with eight prismatic actuators, four rigid connector nodes and four passive revolute joints.

Chapter 5: Experimental Validation of Prismatic Actuators

This chapter verifies the performance of the fabricated three-state prismatic actuators. A proportional-integral-derivative (PID) controller is implemented and applied to the prismatic actuators. Actuation and locking forces are analysed and measured. Forces required to passively move the prismatic actuator are experimentally evaluated. Three states and state transitions of a single, two and multiple prismatic actuators are physically achieved and demonstrated.

Chapter 6: Control and Locomotion of Modular Robots

This chapter presents control strategies developed for modular robotic structures and conducts experiments to demonstrate the coordination and locomotion capabilities of the developed modular robot. Physics-based simulations in Webots [[Michel, 2004](#)] are performed to further show locomotion of more complex modular robots.

Chapter 7: Conclusions and Future Work

This chapter concludes research findings based on the simulations and experiments of this work and points out some potential improving directions for future research.

CHAPTER 2

Literature Review

This chapter explores state-of-the-art modular robots. Basic conceptions and knowledge are introduced first. Then, mechatronics designs of existing robotic modules are analysed. After that, typical control strategies and popular communication methods in modular robotics are reviewed. Finally, this chapter concentrates on related works on passive compliance and multiple-state actuators with compliance capabilities.

2.1 Introduction

A modular robotic system (MRS) consists of separate identical or different modules capable of attaching to or detaching from each other in a manual or self-adaptive manner [Yim *et al.*, 2009, 2007]. Each module of an MRS may have a simple physical embodiment with limited functionalities, nevertheless, the MRS as a whole robot can have powerful capabilities such as locomotion and manipulation. Ideally, an MRS is expected to exploit three properties: versatility, robustness and low cost [Støy *et al.*, 2010]. Versatility can be ensured by the self-adaptive or operator-guided shape-changing ability of the MRS. In other words, an MRS can have different robotic structures that are suitable for different tasks and different working environments. Robustness is attained by using the attachment and detachment abilities to replace a malfunctioning or broken robotic module with a functional one during the task execution process. In this way, the whole robotic system has high fault tolerance which leads to self-repair. Low cost is dependent on the development of hardware techniques and mass production technologies. Note that, for a certain task under a given working environment, a tailored robot could outperform an MRS in terms of the performance. In spite of this, an MRS is more preferred and feasible than a tailored counterpart when dealing with unknown or time-varying working environments.

There exist numerous criteria for classifying modular robots. In terms of the number of module types, an MRS could be homogeneous or heterogeneous [Murata & Kurokawa, 2007]. A homogeneous MRS has identical modules, which facilitates the design process. Only one robotic module needs to be designed and by means of mass production, a range of different robotic structures could be obtained by reconfiguring different assembly of identical modules. However, since each module may have to incorporate the actuation, computation, sensory and communication components, the homogeneous module tends to be complex and bulky, which limits the flexibility of the robot. A heterogeneous MRS has different types of modules with different capabilities.

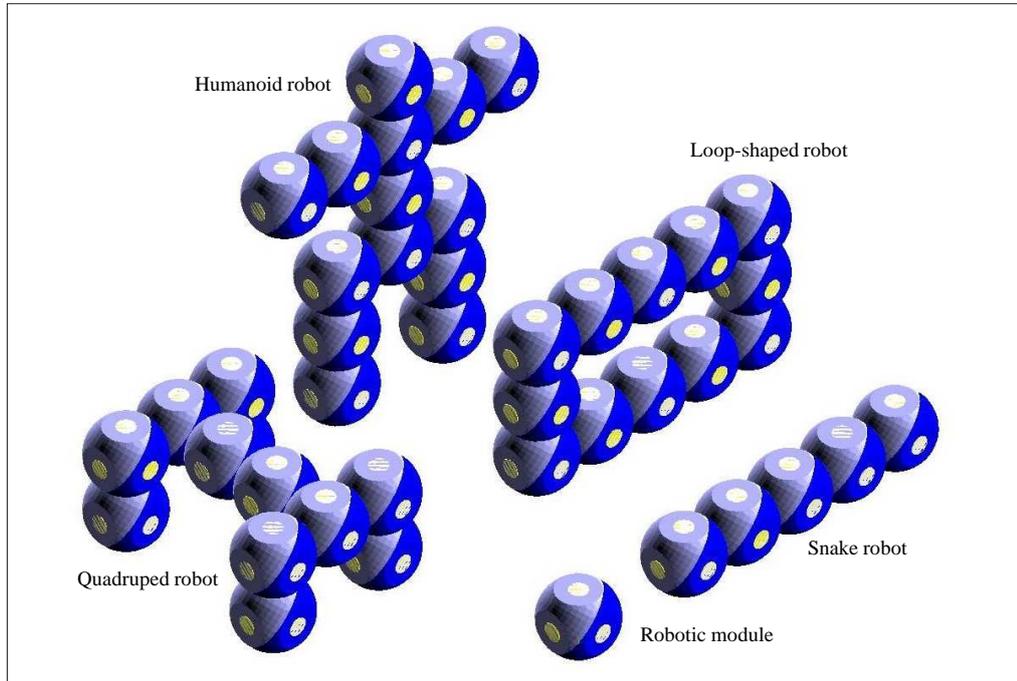
For example, a heterogeneous MRS may be composed of a series of different modules including an actuation module, a sensory module, a communication module and a manipulation module. The complexity, volume and weight of each heterogeneous module are less than those of a homogeneous module, while the variety and extensibility of robotic structures are increased at the cost of designing multiple modules.

Considering the hierarchy of robotic structures, modular robots can be categorized as chain-type, lattice-type and hybrid-type robots [Murata & Kurokawa, 2007, Støy *et al.*, 2010]. Chain-type MRSs have relatively simpler structures than the others as they usually exhibit a string-shaped, ring-shaped or tree-shaped structures. Locomotion generation and controller design are relatively easy for a chain-type MRS since less kinematic constraints are involved when designing a control framework. Lattice-type or mesh-type MRSs usually have a grid-based structure and adopt a discrete way for connecting and disconnecting modules. A lattice-type MRS would be more dexterous in terms of shape-changing than its chain-type counterpart. Hybrid-type MRSs can combine the characteristics and advantages of chain-type and lattice-type MRSs, thus they tend to be more versatile and flexible.

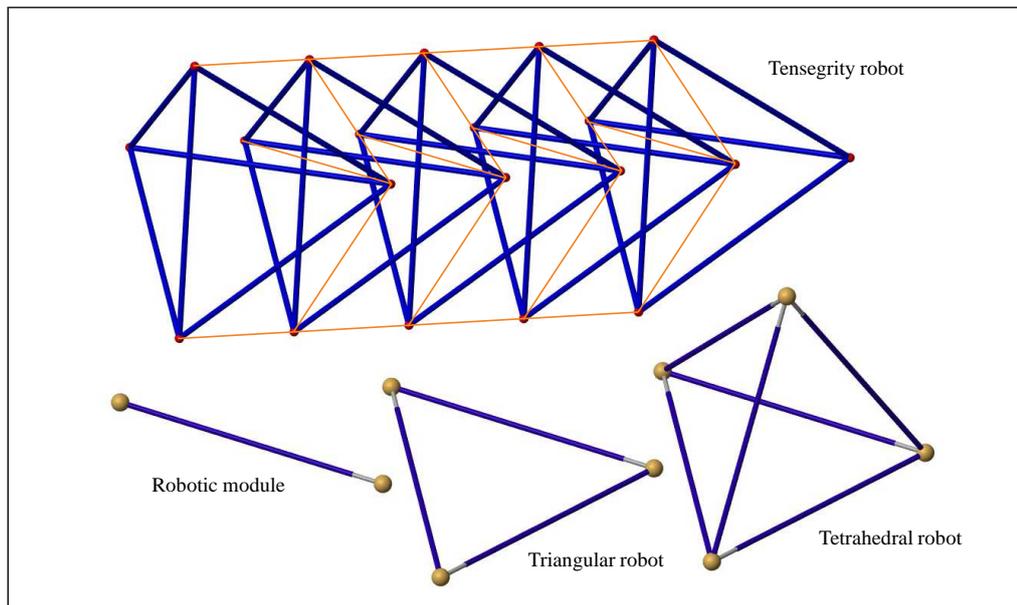
Existing designed robotic modules can be divided into block- and strut-type modules according to their appearance. A block-type module usually looks like a cubic box or a spherical ball while a strut-type module is similar to a straight bar. For better understanding, Figure 2.1 presents MRSs comprised of the two types of modules. Generally, there are two common shape-changing methods in modular robotics.

- Deformation: an MRS with a specific configuration can change its shape without changing the connectivity of robotic modules.
- Reconfiguration: an MRS can change its configuration (i.e., connectivity) by connecting and disconnecting robotic modules manually or self-adaptively.

Deformation can be used to adjust the MRS shape to internal and external forces exerted on the robotic structure with a specific configuration. In contrast, reconfiguration endows an MRS with a wide range of robotic configurations which can emulate conventional monolithic robots and are suitable for different tasks under different working environments. Block-type modular robots are suitable for reconfiguration and could achieve locomotion through reconfiguration [Christensen *et al.*, 2010a, Kurokawa *et al.*, 2008, Salemi *et al.*, 2006, Spröewitz *et al.*, 2008], while strut-type modular robots are



(a)



(b)

Figure 2.1: Illustrative Examples of Block- and Strut-Type MRSs. (a) Block-Type MRS Composed of Block-Type Modules. (b) Strut-Type MRS Composed of Strut-Type Modules.

adept at deformation and could achieve locomotion through deformation [Garcia, 2008, Lyder *et al.*, 2008, Zagal *et al.*, 2012]. The details of robotic modules of these modular robots will be explored in Sections 2.2 and 2.3.

2.2 Block-Type Modules

Up to date, researchers have designed and developed various block-type modular robotic modules. These modules generally consist of one, two or more revolute actuators. A block-type module with more revolute DOFs tend to have higher mobility at the cost of higher complexity. To obtain design and implementation techniques for our research from the existing modular robots, this section analyses mechatronics design and hardware implementation of some typical block-type modular robotic modules ranging from one-DOF modules such as YaMoR [Moeckel *et al.*, 2006], Molecubes [Zykov *et al.*, 2007a] and ATRON [Østergaard *et al.*, 2006] with limited mobility to wheeled modules like Sambot [Wei *et al.*, 2010] and SMORES [Davey *et al.*, 2012] that are mobile platforms. A comparison of these robotic modules is presented in Table B.1 of Appendix B.

YaMoR is a self-contained chain-type and homogeneous modular robot [Maye, 2007, Moeckel *et al.*, 2006, Spröewitz *et al.*, 2008]. Two generations of YaMoR modules have been built and this section only focuses on the latest version as the two generations share a lot of ideas regarding design and implementation. Each YaMoR module weighs 0.25 kg and has a dimension of $50 \times 45 \times 94$ mm as shown in Figure 2.2. Mechanically, the module has a U-shaped lever with a single revolute DOF and four printed circuit boards (PCBs) serving as a casing for housing electronics. To drive the DOF, a servomotor with a maximum torque of $73 \text{ N} \cdot \text{cm}$, a maximum rotation speed of $375^\circ/\text{s}$ and a working range of 180° was selected to be capable of lifting up to 3 other modules. A screw-and-pin based connection mechanism was designed to allow manual attachment and detachment of modules in multiple ways. Figure 2.2 shows the mechatronics components of a YaMoR module. Each module is powered by a lithium-ion battery and contains seven PCBs for computation, communication, actuation and sensing: (1) a microcontroller board containing a Philips LPC2138 chip based on an ARM7TDMITM architecture; (2) a sensor board with a three-dimensional (3D) accelerometer and an infra-red (IR) distance sensor; (3) a field-programmable gate array (FPGA) board for

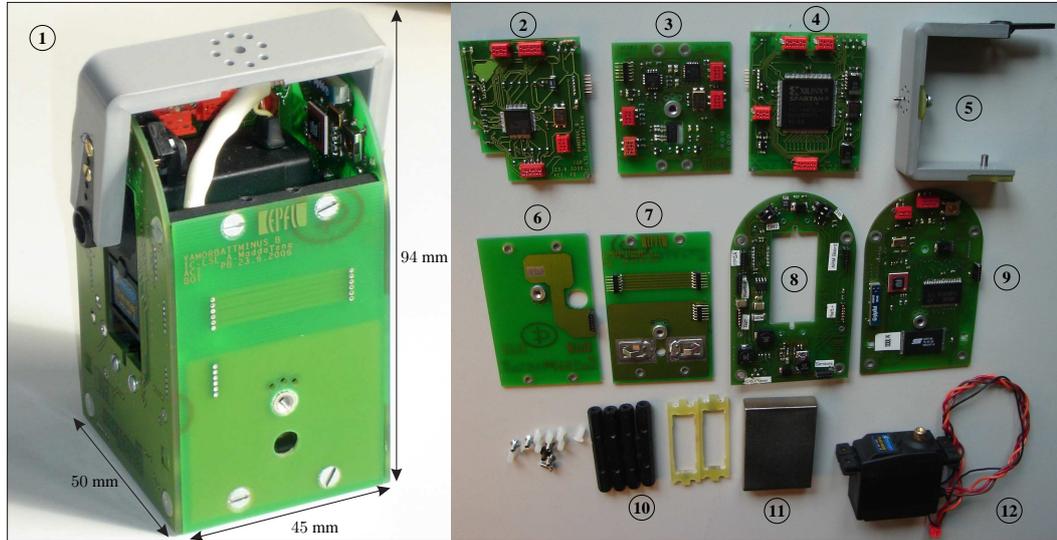
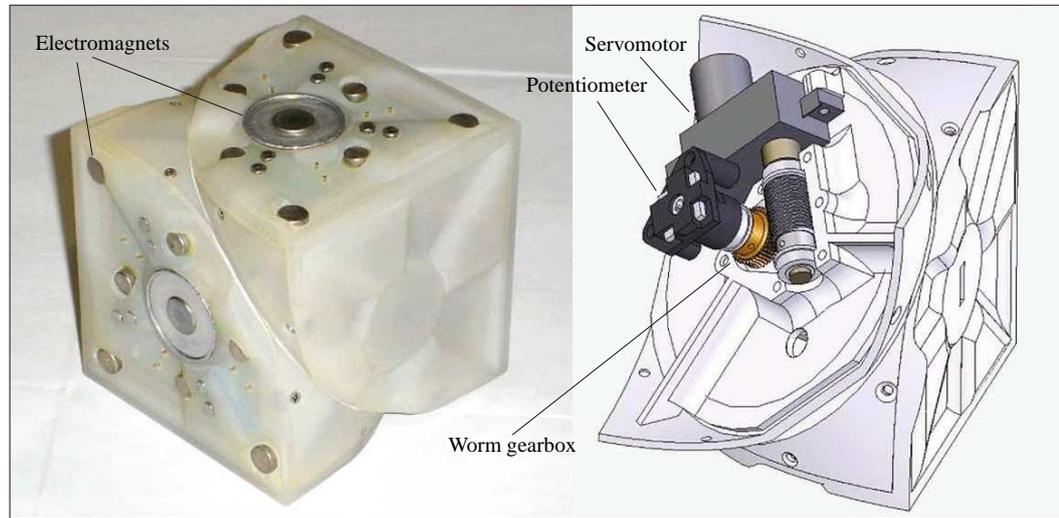


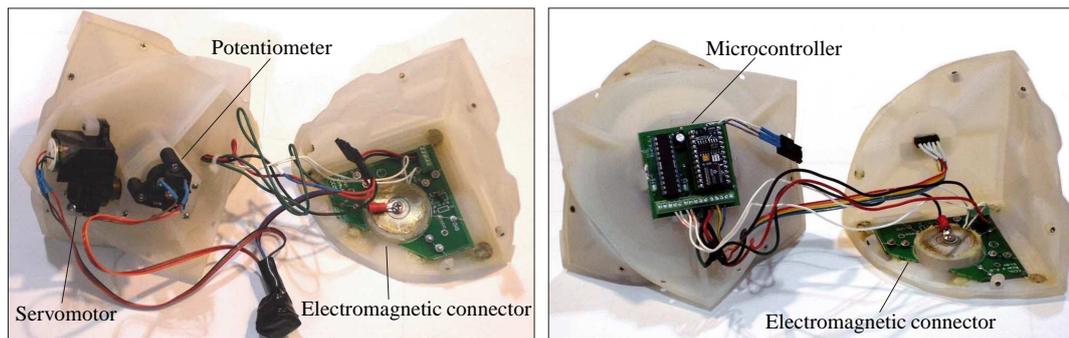
Figure 2.2: Prototype and Components of YaMoR Robotic Module [Maye, 2007]. (1) Prototype. (2) ARM Board. (3) Sensor Board. (4) FPGA Board. (5) U-Shaped Lever. (6) Battery Plus Board. (7) Battery Minus Board. (8) Power Board. (9) Bluetooth Board. (10) Fixations. (11) Battery. (12) Servomotor.

handling computationally expensive cases; (4) two boards for supporting the lithium-ion battery; (5) a board serving for power management; and (6) one board carrying a Zeevo chip ZV4002 for Bluetooth communication. The ARM and the FPGA boards can be configured to work separately or in parallel if necessary. One of the outstanding characteristics of YaMoR is that four of the mentioned PCBs (i.e., the two battery boards, the power board and the Bluetooth board) work as the outer casing, reducing manufacturing cost. In addition, the YaMoR module with one revolute DOF adopts a simple design, which facilitates the mass production of a large number of modules for investigating and verifying control strategies. The drawbacks of YaMoR modular robot include the limited mobility of each module and manual reconfiguration.

Molecube and Molecubes are modular robots of which each module has simple structure with only one revolute DOF [Zykov *et al.*, 2005, 2007a,b, 2008]. As presented in Figures 2.3 and 2.4, each module of Molecube and Molecubes is a cubical block that consists of two parts separated along a plane perpendicular to the long diagonal.



(a)



(b)

(c)

Figure 2.3: Molecube Robotic Module [Zykov *et al.*, 2005, 2007b]. (a) Prototype and Actuation Mechanism. (b) Servo Drive Half. (c) Microcontroller Half.

Each Molecube module weighs 0.625 kg and is $100 \times 100 \times 100$ mm in size. As shown in Figure 2.3(a) and (b), the module DOF is driven by a servo motor coupled with a worm set of which the gear ratio is 1 : 40, allowing a speed of $15^\circ/\text{s}$ and a torque of $1.41 \text{ N} \cdot \text{m}$. For feedback control, a potentiometer is attached to the worm wheel. All the six faces of a Molecube module can be equipped with a connector interface containing electromagnet and electrical terminals shown in Figure 2.3(b) and (c) which can connect modules mechanically and electronically. Each Molecube module has a Parallax BS II microprocessor, and power supply can be propagated from modules

2.2 Block-Type Modules

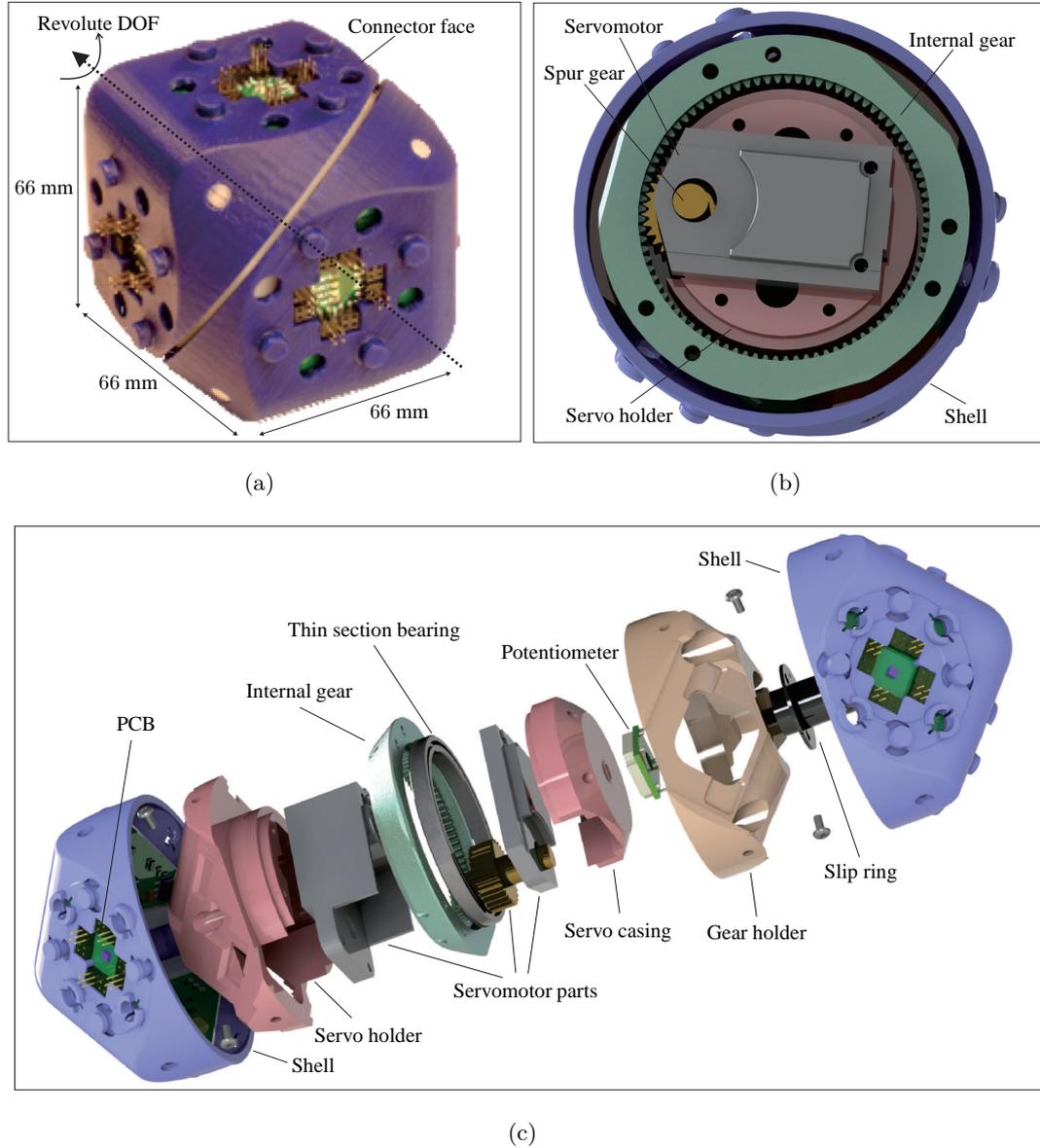


Figure 2.4: Molecubes Actuation Module [Zykov *et al.*, 2007a, 2008]. (a) Prototype. (b) Actuation Mechanism. (c) Exploded View of a Molecubes Module.

connected to the base plates to all consecutive modules. The Molecube module is designed for investigating and achieving self-replication using modular robots.

Molecubes is an open-source heterogeneous modular robotic system that aims to provide a low-cost platform for researchers, hobbyists and enthusiasts to accelerate the

development progress of modular robotics [Zykov *et al.*, 2007a, 2008]. An open-source Molecubes robot can consist of actuation modules, controller modules, power modules, wheel modules and gripper modules. As shown in Figure 2.4(a), the actuation module is $66 \times 66 \times 66$ mm in size and weighs 0.2 kg which is quite smaller and lighter than its predecessor (i.e., the Molecube module). A modified Dynamixel AX-12 servomotor paired with an internal gear is adopted to drive the revolute DOF continuously as can be seen in Figure 2.4(b). The gear ratio is around 1 : 762 with a maximum torque of $4.85 \text{ N} \cdot \text{m}$. Each half module has a PCB set carrying an ATmega324P processor as illustrated in Figure 2.4(c). The two processors of a module and the integrated processor of the servomotor are connected by an RS232 bus. The outward side of the PCB set has an electrical interface with 16 pairs of pins and sockets. Mechanically, modules can be connected manually through the acrylonitrile butadiene styrene (ABS) pins and sockets of the shells. A slip ring is employed to enable a continuous rotation and a potentiometer is attached to the slip ring to measure the joint position and achieve a closed-loop feedback control.

ATRON is a lattice-type homogeneous self-reconfigurable modular robot of which each module adopts a novel spherical shape [Christensen *et al.*, 2010a, Østergaard *et al.*, 2006]. A module weighs 0.85 kg and has a diameter of 110 mm. As seen from Figure 2.5(a), the module consists of two hemispheres held together by a thin section ball bearing providing a revolute DOF. Such a DOF is driven by a motor geared with a planetary gearbox as shown in Figure 2.5(b). A pancake-style slip ring was designed for power and serial communication between the two hemispheres to avoid wire twisting and obtain a continuous rotation. A hook-hole based mechanism was employed for connecting modules and each hemisphere has two active and two passive connectors shown in Figure 2.5(a) and (c). A passive connector is built from two stainless steel bars and an active connector has three retractable hooks. Such hooks are driven by a non-backdrivable mechanism composed of a direct current (DC) motor, a lead screw and a push bar, allowing ATRON modules to keep connected without any power consumption. The superior performance of the designed connectors has been validated through a horizontal chain composed of five modules connected by four connectors [Østergaard *et al.*, 2006]. Figure 2.5(d) shows an exploded view of the mechanical components of an ATRON module. As for electronics, each hemisphere has an ATmega128 and an ATmega8 processors. Sensors include an accelerometer for orientation determination

2.2 Block-Type Modules

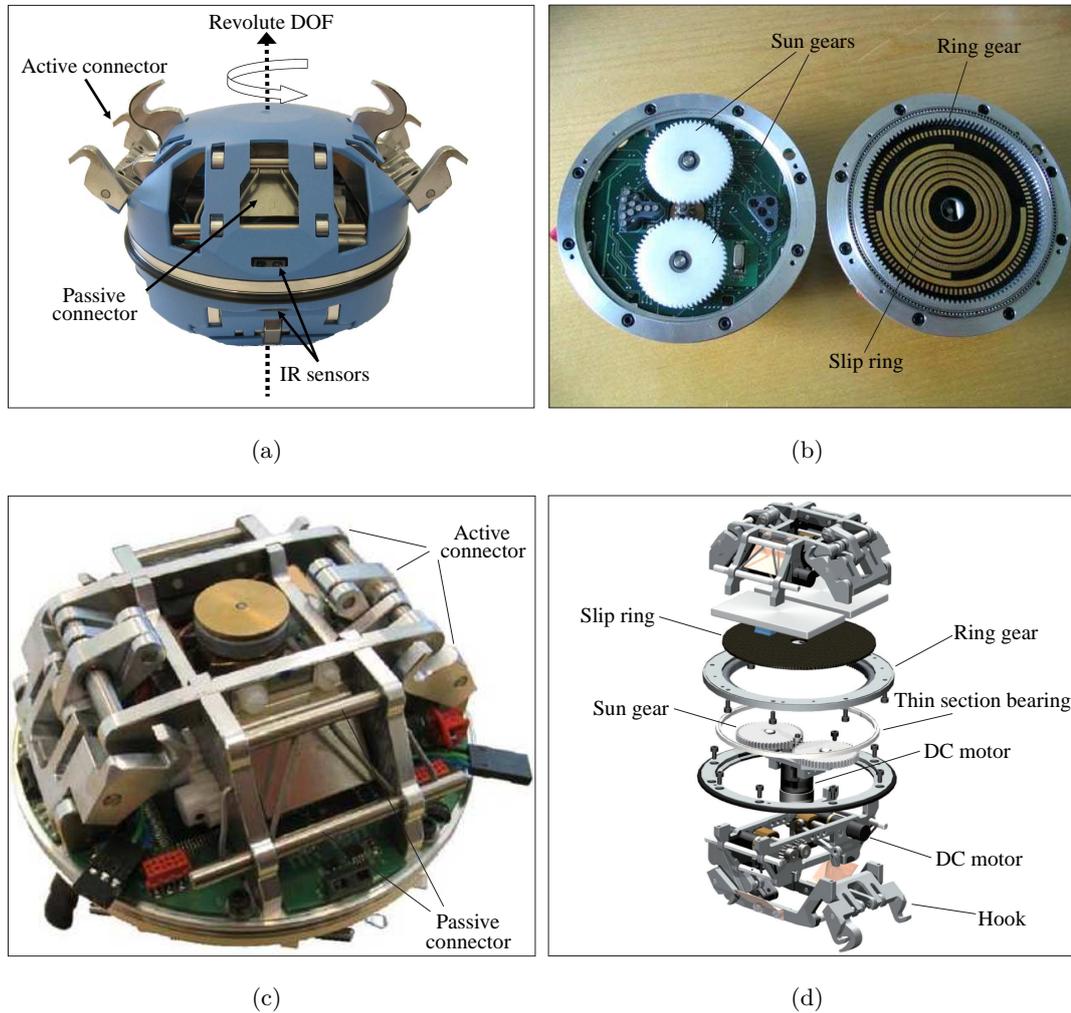


Figure 2.5: ATRON Robotic Module [Christensen *et al.*, 2010a, Østergaard *et al.*, 2006]. (a) Prototype. (b) Actuation Mechanism. (c) Module without Casing. (d) Exploded View of Mechanics.

and IR transmitters and receivers for neighbour-to-neighbour communication between modules as well as proximity detection.

MTRAN is one of the most maturely developed and investigated hybrid-type modular robots [Kurokawa *et al.*, 2006, 2008, Murata *et al.*, 2002]. A lot of time and effort have been continuously devoted to improving mechatronics design and hardware implementation of MTRAN module giving birth to three generations of prototypes (i.e.,

2.2 Block-Type Modules

MTRAN I, II and III). Figure 2.6 shows details about the MTRAN III module. Each MTRAN III module has a size of $6.5 \times 6.5 \times 13$ mm with a weight of 0.42 kg and consists of two cubic blocks connected by a beam. As shown in Figure 2.6(a), such a robotic module possesses two parallel revolute DOFs with certain mobility. The two revolute DOFs are driven by micro geared motors paired with spur gears and potentiometers. As shown in Figure 2.6(b), each module is integrated with one master and three slave microcontrollers and powered by a 7.4 V lithium-polymer battery. Figure 2.6(c) shows the mechanical components of an MTRAN module and the utilized connector mechanism. Most of the mechanical parts were built from polyacetal and ABS. A hook-hole

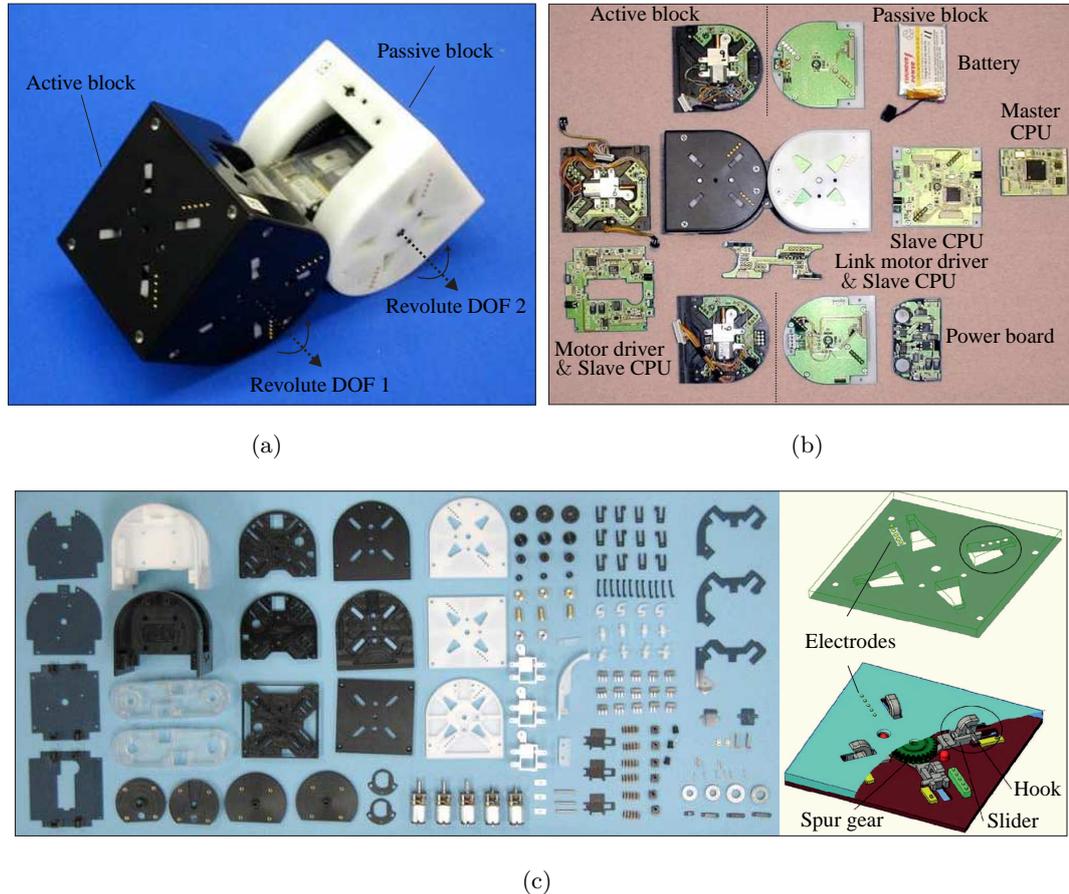


Figure 2.6: MTRAN III Robotic Module [Kurokawa *et al.*, 2006, 2008, Murata *et al.*, 2007, 2002]. (a) Prototype. (b) Circuit Boards. (c) Mechanical Components and Connector Mechanism.

mechanism driven by a micro DC motor was designed for connecting modules actively. Once the hooks are engaged, no further power is required to lock. This is an improvement of connection mechanisms of MTRAN I and II where permanent magnets and shape memory alloy (SMA) springs were adopted. With three active connectors and three passive connectors, MTRAN modules can be attached to form a broad range of robots with different configurations. The module has a set of sensors including IR diodes, accelerometers and IR proximity sensors enabling the module to locate other modules and to accomplish the docking process. Each module is also equipped with Bluetooth for communicating with other modules.

Dtto is a recent open-source 3D printable modular robot [Hackaday, 2016]. The Dtto module design follows some ideas from MTRAN. Since it is open-source like Molecubes, researchers, hobbyist and enthusiasts in modular robotics can learn a lot of knowledge from the Dtto project. This is significant for speeding up the development of this field. Like MTRAN, each Dtto module shown in Figure 2.7(a)–(c) has two parallel revolute DOFs driven by two servomotors. The two motors are attached to the central beam directly and simply using some screws. Figure 2.7(d) shows the employed connector mechanism. For coupling with other modules, each module has three active connectors of which the mechanism consists of two parts linked using a rubber band: a hook for attachment and a bar for detachment. Neodymium disc magnets are embedded in the connector surfaces for alignment during the docking process. Each module has a 3.7 V lithium-polymer battery and an LM317 voltage regulator of which the output voltage is used to power five servomotors of the module. It contains an Arduino Nano microcontroller for computation, an HC-05 Bluetooth and an nRF24L01+ modules for communication with other Dtto modules. This means each Dtto module is self-contained with computation and communication capabilities.

SuperBot Inspired by MTRAN, SuperBot was designed as a step towards real-world applications [Salemi *et al.*, 2006, Shen *et al.*, 2008]. As a substitute for the MTRAN beam, a middle revolute joint was introduced, leading to three DOFs of each SuperBot module. As shown in Figure 2.8(a), the three DOFs provide roll, pitch and yaw rotary motions, making a SuperBot module more flexible as compared with an MTRAN module. Each SuperBot module is made up of a kind of aluminium alloy having a dimension of $84 \times 84 \times 168$ mm and an estimated weight of 0.5 kg. As presented in Figure 2.8(b),

2.2 Block-Type Modules

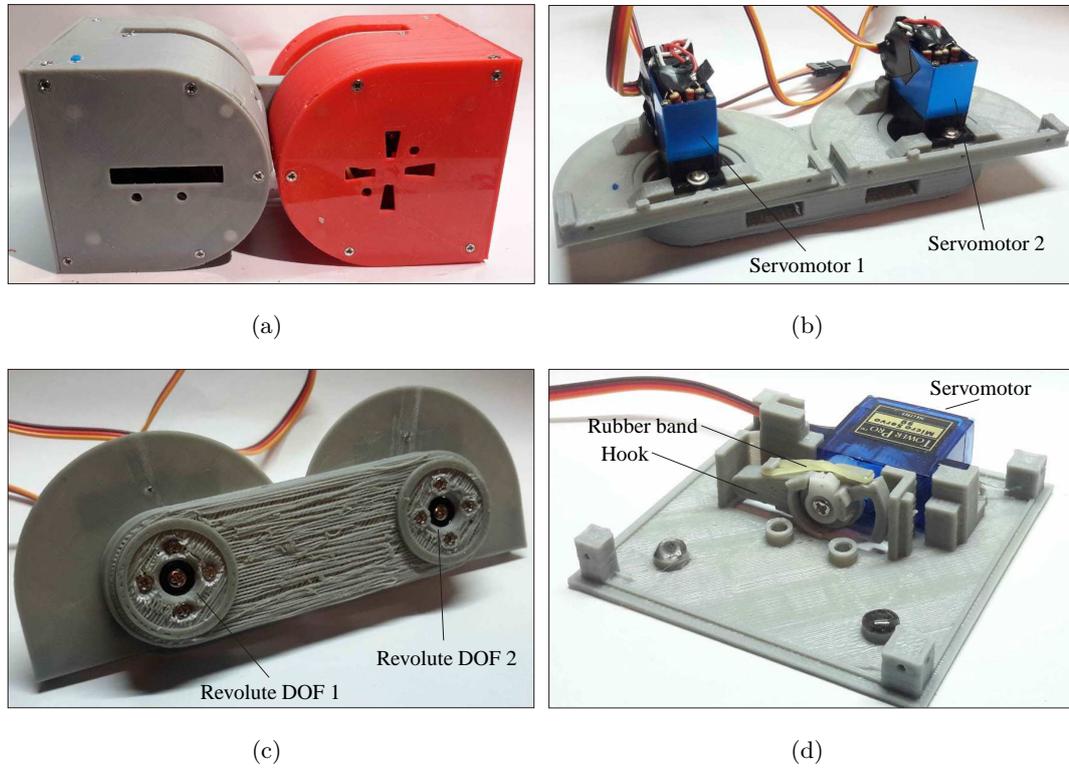


Figure 2.7: Dtto Robotic Module [Hackaday, 2016]. (a) Prototype. (b) Inside View of Actuation Mechanism. (c) Outside View of Actuation Mechanism. (d) Connector Mechanism.

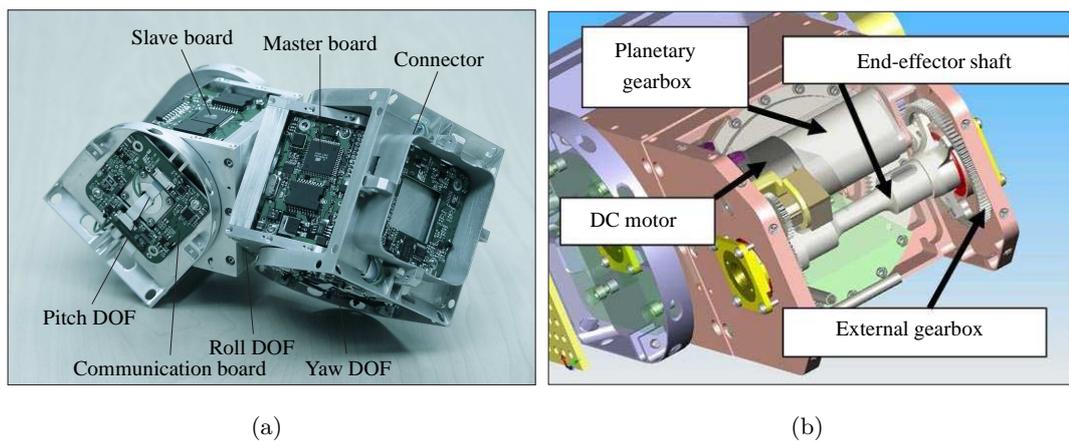
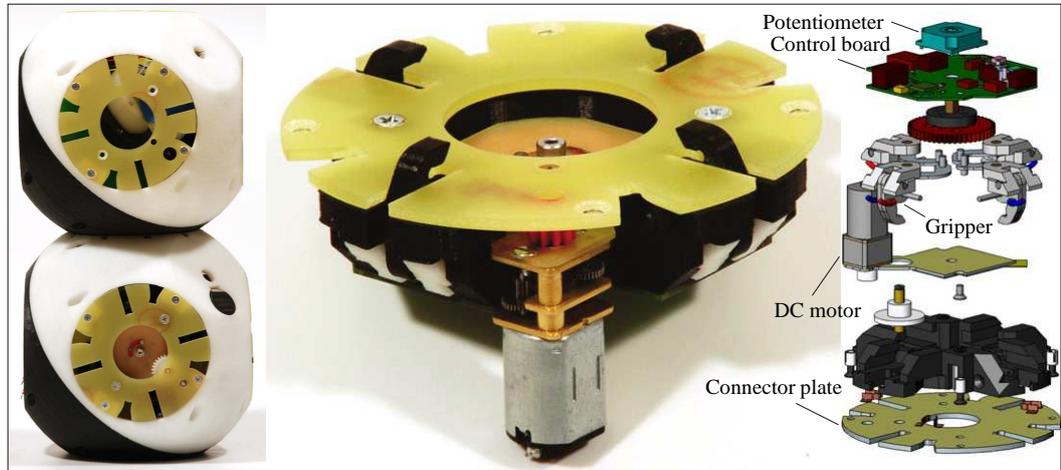


Figure 2.8: SuperBot Robotic Module [Salemi *et al.*, 2006, Shen *et al.*, 2008]. (a) Prototype. (b) Actuation Mechanism.

each DOF is driven by a DC motor coupled with a planetary gearbox and an external gearbox that provides a maximum of $6.38 \text{ N} \cdot \text{m}$ torque capable of lifting three other SuperBot modules. To achieve self-reconfiguration, a genderless connector named SINGO was designed and implemented [Shen *et al.*, 2009]. A SuperBot module has six SINGO connectors of which each one has four jaws and four slots and the jaws are driven by a circular gear with spiral grooves. Two attached modules can disconnect from both sides. With respect to electronics, the module has two ATmega128 microcontrollers of which one works as the master controller and the other one serves as the slave controller. Each controller takes charge of docking processes of three connectors and the two controllers communicate with each other through an inter-integrated circuit (I2C) bus. The master controller is used for radio frequency (RF) communication and managing one motor while the slave controller is responsible for the other two motors. Sensors of each module include a 3D accelerometer for monitoring orientation, an RF receiver for remote control and one potentiometer for each motor to sense the joint angular position. Each connector surface has four IR receivers and one IR transmitter for communication between modules during docking tasks.

Roombots Following the concept of adding a central joint in SuperBot module on the basis of MTRAN module, a modular robot called Roombots based on the Molecubes module design was designed and developed [Spröwitz, 2010, Spröwitz *et al.*, 2014, 2010]. The vision of the hybrid-type Roombots robot is to use modular robots for making self-adaptive intelligent furniture and providing assistance for elderly and disabled people. Each Roombots module has a dimension of $110 \times 110 \times 220 \text{ mm}$ and weighs 1.4 kg . As shown in Figure 2.9(a), the module has three revolute DOFs. Each DOF is driven by a designed gearbox presented in Figure 2.9(b). As can be seen from the figure, the actuation gearbox mainly consists of a planetary gearbox, a DC motor, a slip ring, a thin section bearing and other fixations. The speeds of the inner DOF and outer DOFs are $19.4 \text{ revolution per minute (RPM)}$ and 26.6 RPM , respectively. The nominal torques of the inner DOF and outer DOFs are separately $4.9 \text{ N} \cdot \text{m}$ and $3.6 \text{ N} \cdot \text{m}$. An active connection mechanism (ACM) based on a crank-slider and a slider-latch mechanisms was designed for attaching and detaching Roombots modules autonomously. Such a connection mechanism is shown in Figure 2.9(a) containing a DC motor, a control board, a position sensor, a connector board and four grippers. The stall torque of each crank-slider is estimated as $0.9 \text{ N} \cdot \text{m}$. Figure 2.10 illustrates the electronics

2.2 Block-Type Modules



(a)



(b)

Figure 2.9: Roombots Robotic Module [Spröwitz *et al.*, 2014, 2010]. (a) Prototype and Connection Mechanism. (b) Actuation Gearbox.

arrangement of a Roombots module. Each Roombots module possesses three motor controller boards for driving motors associated with the three DOFs, three sensor boards for absolute position sensing, several ACM control boards carrying a potentiometer for controlling position of latches, and a power management board. Each motor control board is equipped with a dsPIC microcontroller (dsPIC33FJ128MC802) capable of processing information and realizing the low-level position and speed control. Each revolute actuator of a Roombots module can be actively controlled to attain three

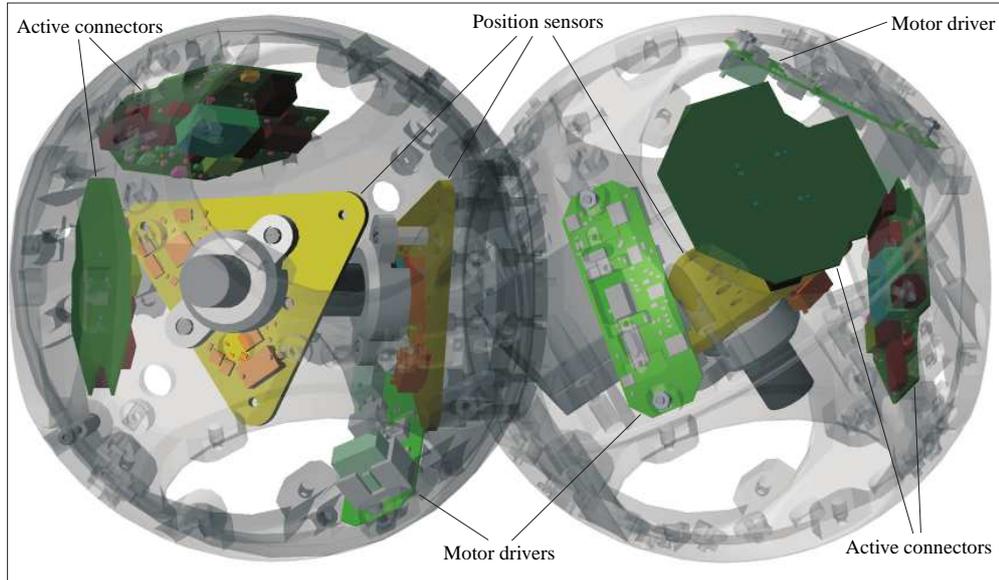


Figure 2.10: Electronics Arrangement of Roombots [Sprowitz *et al.*, 2014, 2010].

different actuation modes, i.e., oscillation, rotation and locked modes. Here the locked mode is actively achieved by using a constant setpoint signal.

ModRED is a homogeneous self-reconfigurable modular robot developed for planetary exploration and discovery [Baca *et al.*, 2014, Chu *et al.*, 2011, Hossain *et al.*, 2012, Hossain, 2014]. Up to date, as presented in Figure 2.11, two generations of ModRED prototypes have been designed and fabricated. Each ModRED module has four DOFs: three revolute DOFs and one prismatic DOF. One unique characteristic of ModRED is the prismatic DOF driven by a lead screw which is claimed to bring some advantages for docking and for achieving a larger workspace. A ModRED II module consists of five sections: two end brackets, one rotary segment, one linear segment and one central cubic segment housing motors, circuits and batteries. Figure 2.12 shows the electronics and connector mechanism of ModRED II. In fact, the main differences between ModRED I and ModRED II are the electronics deployment and docking mechanisms. Each ModRED I module has an Arduino Fio microcontroller board based on ATmega328P for achieving a distributed control strategy and possesses two docking interfaces driven by two latching solenoids. Robot systems composed of ModRED I modules are powered by an external 12 V power supply. To achieve some improvements, ModRED II was

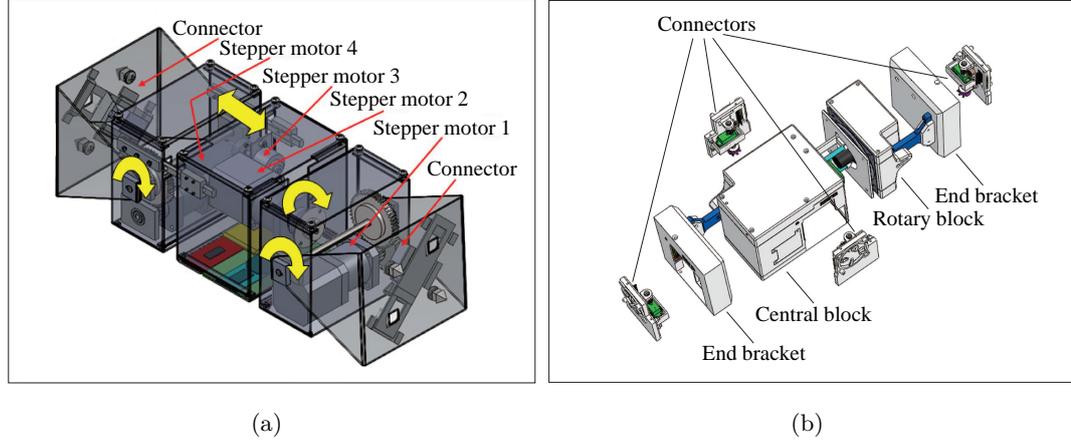
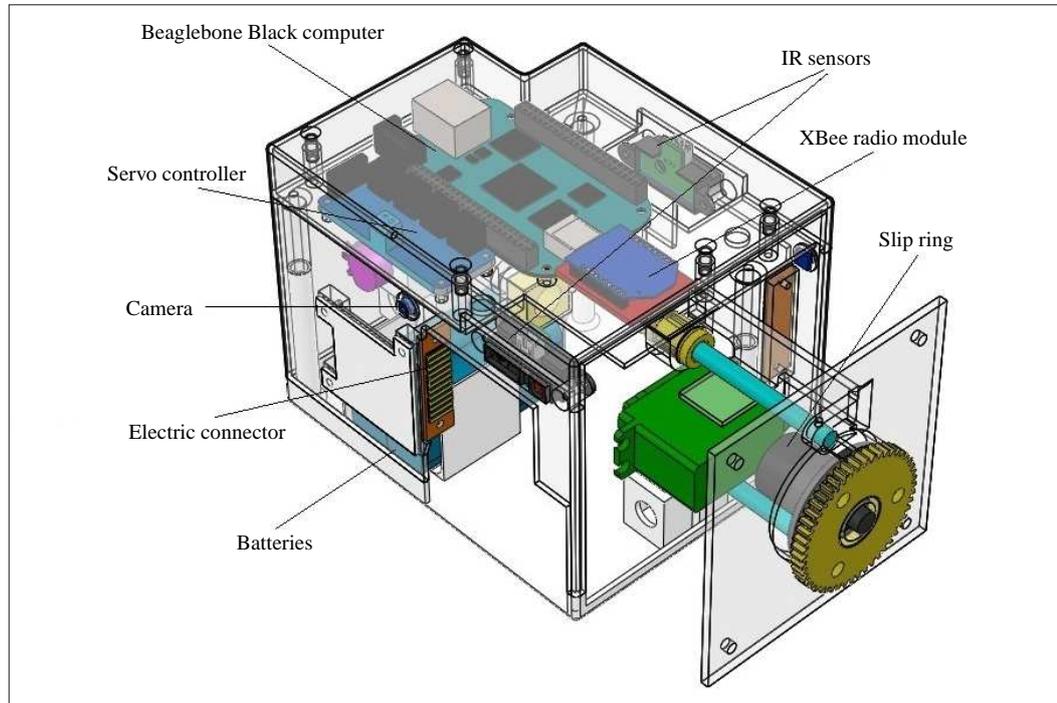


Figure 2.11: CAD Models of ModRED I and ModRED II [Hossain, 2014]. (a) ModRED I. (b) ModRED II.

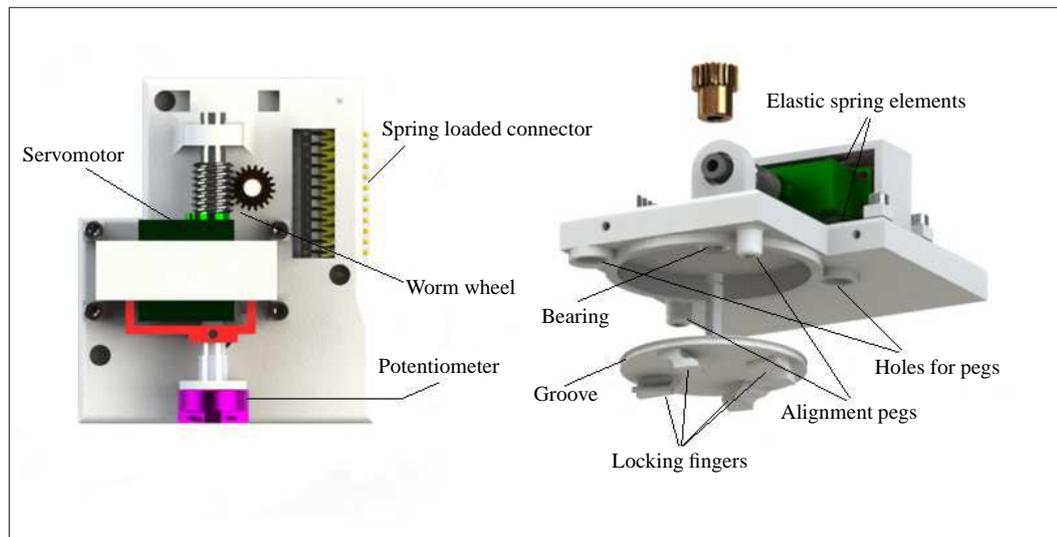
designed and developed by rearranging the electronics and adding two extra docking faces. Specifically, the processor is replaced by a Beaglebone Black ARM Linux single board computer as shown in Figure 2.12(a), which significantly improves the computation power of the robotic module. Elastic series motors presented in Figure 2.12(b) are designed to reduce the weight of the robotic module and four RoGenSiD docking mechanisms shown in Figure 2.12(b) are incorporated into ModRED II. A ModRED II module is powered by two lithium-ion batteries and contains eight servomotors for driving the four DOFs and four docking mechanisms. Sensors include a 9-DOF inertial measurement unit for long distance sensing, four IR sensors attached to each of the docking faces for shorter distance sensing and obstacle avoidance, four cameras for identifying docking faces and an XBee radio module for wireless communication. These features offer ModRED II more powerful capabilities in terms of computation, self-reconfiguration and locomotion as compared with its predecessor.

Sambot is a compact autonomous modular mobile robot having an overall size of $80 \times 80 \times 120$ mm as illustrated in Figure 2.13(a) [Wei *et al.*, 2010, 2011]. The main contribution of Sambot is its integration of self-mobility, self-assembly and self-reconfiguration capabilities. As seen from Figure 2.13(b), the robotic module has two wheels on its bottom driven by two micro DC motors, which endows the module with mobility. With regard to docking, the Sambot module has one active docking interface

2.2 Block-Type Modules



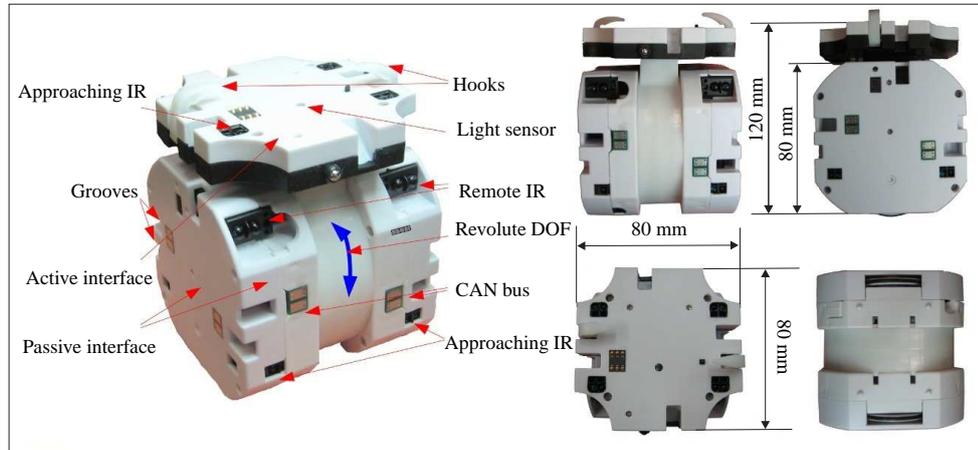
(a)



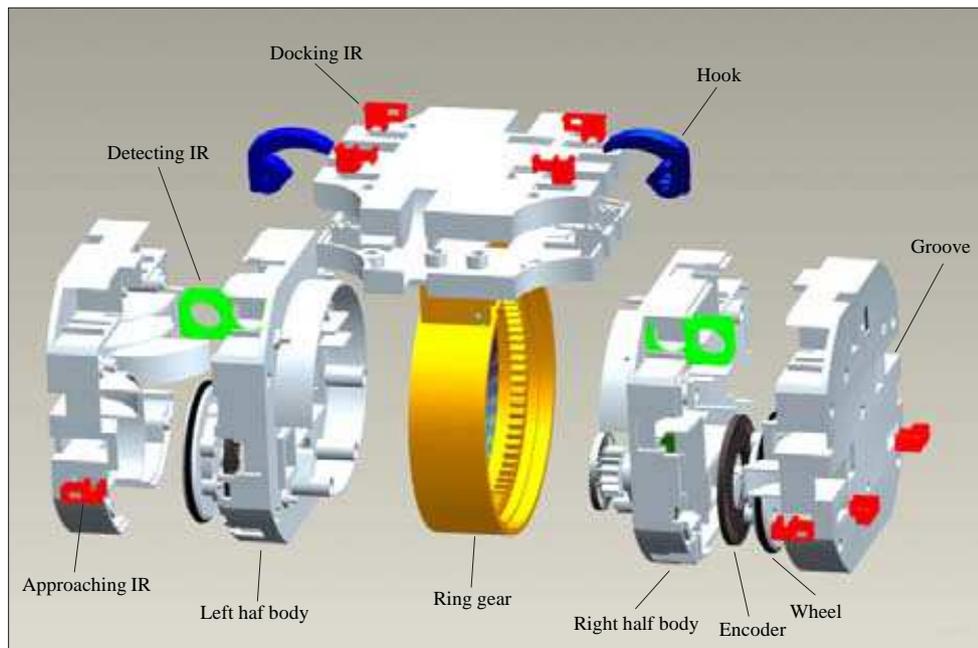
(b)

Figure 2.12: Electronics, Elastic Series Actuator and RoGenSiD Docking Mechanism of ModRED II [Hossain, 2014]. (a) Electronics Inside the Central Segment. (b) Elastic Series Actuator and Connection Mechanism.

2.2 Block-Type Modules



(a)



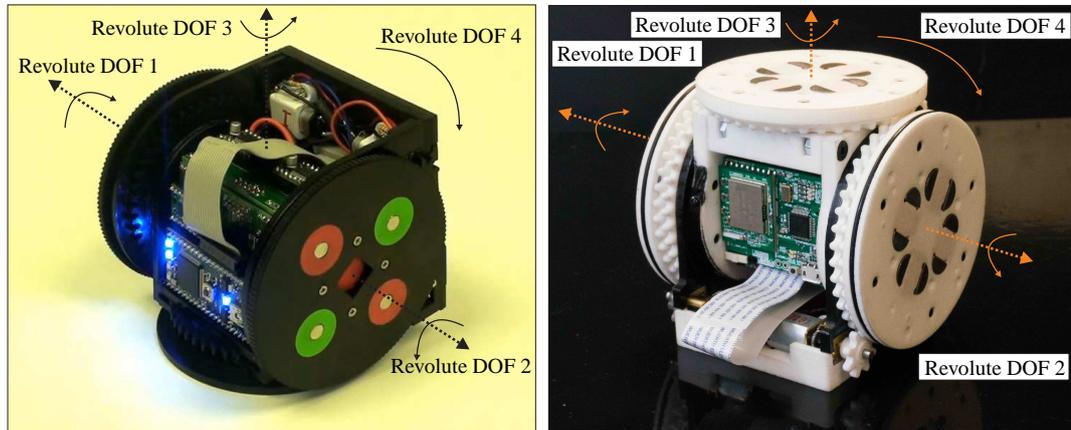
(b)

Figure 2.13: Sambot Robotic Module [Tindell *et al.*, 1994, Wei *et al.*, 2011]. (a) Prototype. (b) Actuation Mechanism.

that has a revolute DOF and four passive docking interfaces. The revolute DOF is driven by a micro DC motor coupled with an internal gear with a range of $\pm 150^\circ$, which also can work as a revolute joint in a robotic structure composed of Sambot

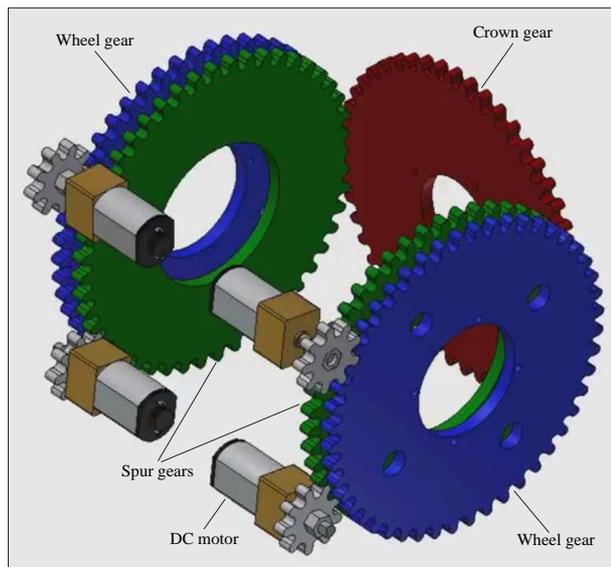
modules. A hook-groove based docking mechanism driven by a micro DC motor paired with worm gears was designed for attachment and detachment. In a Sambot module, an STM32F103CB processor serves as a master controller and each motor in a Sambot module is managed by a slave controller based on an ATmega8 processor. The master and slave controllers communicate mutually through an I2C bus. IR sensors were fully adopted for docking, obstacle avoidance and proximity detection. Each Sambot module can either work independently like an individual wheeled mobile robot or work as a unit in a robot structure formed by Sambot modules. A Sambot module with independent mobility in a swarm uses ZigBee for wireless communication while Sambot robots in a connected structure use controller area network (CAN) bus for wired communication.

SMORES shown in Figure 2.14 is a novel modular robot designed to emulate the movement abilities of other previously demonstrated modular robots [Davey *et al.*, 2012, Jing *et al.*, 2016, Tosun *et al.*, 2016]. In other words, SMORES promises to become a universal modular robotic system. The SMORES module design has two versions up to now: SMORES I and II that are kinematically equivalent to each other. Each SMORES module has four active revolute DOFs, i.e., the left, right, pan and tilt DOFs which are marked as revolute DOFs 1–4 in Figure 2.14(a) and (b), respectively. Only the tilt DOF has a motion range of 180° and the other DOFs are continuous. A geared drive transmission system presented in Figure 2.14(c) was designed to actuate the four DOFs. Specifically, for the left and right wheels, two motors and two spur gears were employed to endow the robotic module with self-mobility, which makes such a module can move like a wheeled platform. Two motors and three spur gears were used to actuate the pan and tilt DOFs. These two DOFs can be respectively actuated depending on whether the rotate directions of the two spur gears are the same or not. Such an awesome gear box can not only simplify the mechanical design but also bring versatility for the module. Each SMORES module has three active and one passive docking connectors. In SMORES I, each connector is embedded with four permanent magnets and the active connectors use a docking key drive mechanism shown in Figure 2.15(a) for active docking. The docking key is driven by only one motor and simultaneous docking is not allowed for active connectors within one module. As presented in Figure 2.15(b), SOMORES II employs electro-permanent (EP) magnets for coupling and uncoupling with other modules without using the docking key. Such an EP face connector design is economical and superior as EP magnets can be switched on or off and only consume



(a)

(b)



(c)

Figure 2.14: Prototypes and Actuation Mechanism of SMORES [Davey *et al.*, 2012, Jing *et al.*, 2016]. (a) SMORES I. (b) SMORSE II. (c) Actuation Mechanism.

power during state transitions. Attachment and detachment can be achieved by any one of two connected modules allowing simultaneous docking. As for electronics, each module has an MBED microcontroller for computation and an XBee radio module for wireless communication. The motors are paired with TB6612FNG motor drivers and continuous rotational potentiometers for controlling and sensing the angular positions.

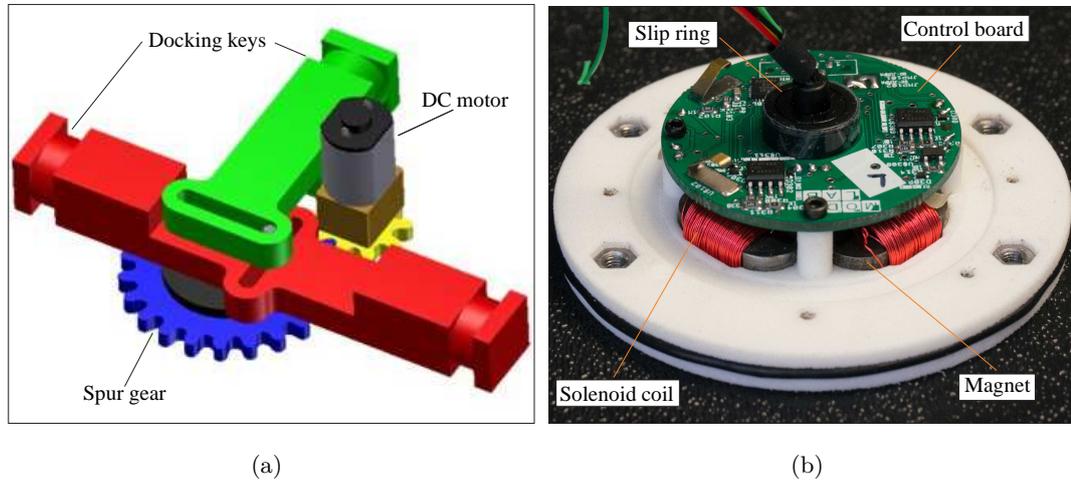


Figure 2.15: Two Connection Mechanisms of SMORSE [Davey *et al.*, 2012, Tosun *et al.*, 2016]. (a) Docking Key for SMORES I. (b) EP Face Connector for SMORSE II.

2.3 Strut-Type Modules

In general, strut-type modular robots are heterogeneous consisting of robotic struts and connector nodes. Robotic struts are linear extensible units that are jointed by connector nodes to form structures. Ideally, a connector node is capable of connecting multiple struts and all the struts jointed by a same node share a common centre of rotation. To attain design and implementation strategies, this section discusses the mechatronics design and hardware implementation of some typical strut-type modular robotic modules (i.e., robotic struts) together with connector nodes.

TETROBOT is a strut-type modular robotic system proposed and investigated in [Hamlin & Sanderson, 1998]. Such a robotic system has two basic hardware components: struts and nodes. The struts can be passive structural members or prismatic actuators. The employed prismatic actuators in [Hamlin & Sanderson, 1998] are electric lead screw actuators that can be driven by DC brushed motors and the coupled gear system. Each prismatic actuator is equipped with a potentiometer to track and record its displacement during the execution process. A centric multilink spherical (CMS) joint mechanism was designed to serve as a compliant node. The CMS node is a solution to the physical implementation of an ideal connector node allowing its

2.3 Strut-Type Modules

connected struts to share a common centre of rotation. A CMS node and a strut are linked through a pin-knuckle mechanism. Figure 2.16 shows details of the CMS node and an assembled TETROBOT prototype [Mândru & Teuțan, 2005, Teuțan *et al.*, 2009]. In a similar way, more complex and different parallel or hybrid serial-parallel

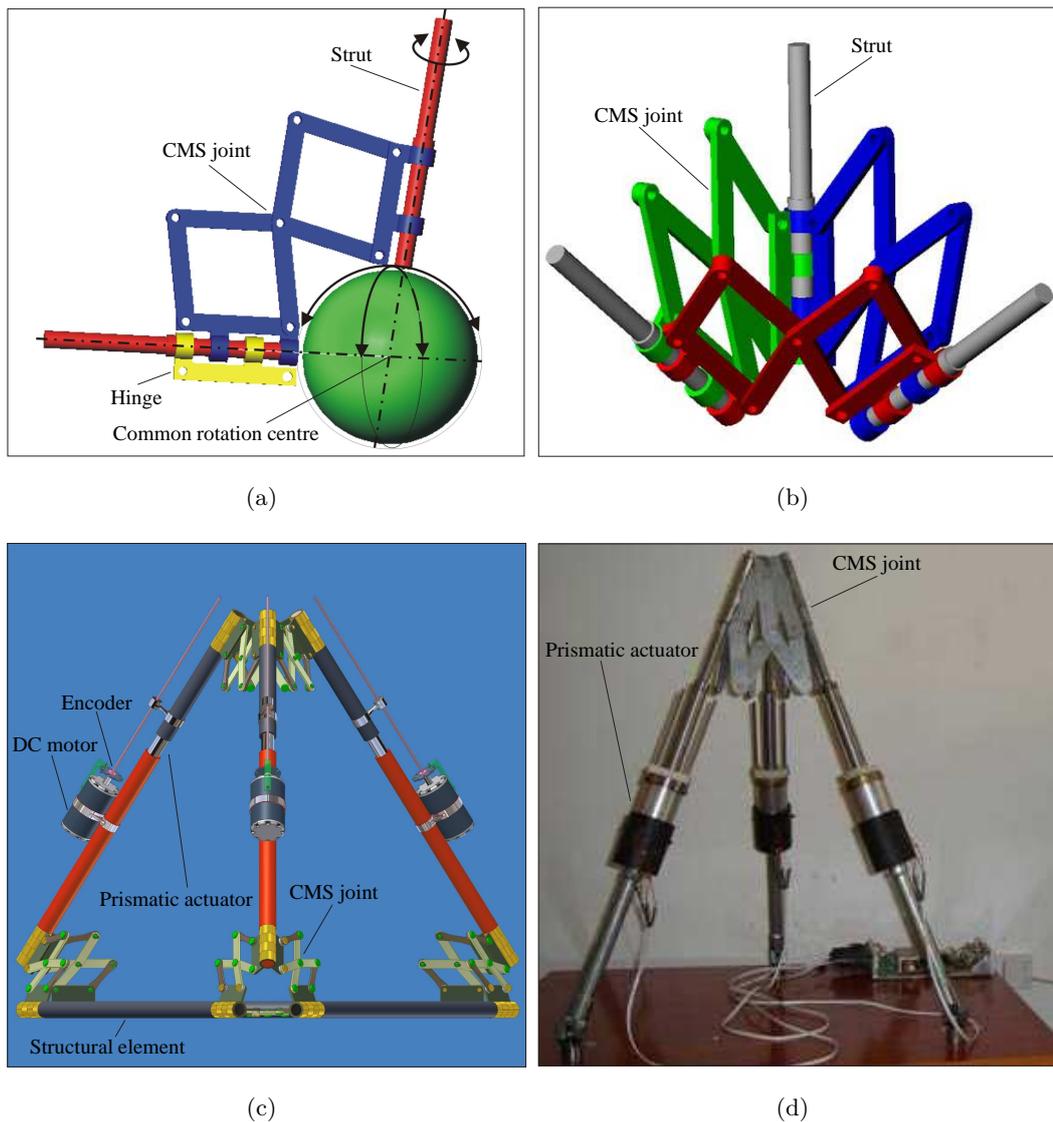


Figure 2.16: TETROBOT Modular Robot [Mândru & Teuțan, 2005, Teuțan *et al.*, 2009]. (a) Schematic of a CMS Node. (b) Example of Three Connected Struts. (c) CAD Model. (d) Physical Structure.

2.3 Strut-Type Modules

robotic structures can be constructed by using the fabricated struts and CMS nodes.

Odin is a heterogeneous modular robot that includes cubic closed-packed (CCP) joints, flexible connectors as well as actuation, power, structure and sensing strut modules [Garcia *et al.*, 2009, Lyder *et al.*, 2008, Støy *et al.*, 2007]. Figure 2.17(a) shows a CCP joint and an actuation module with two flexible connectors. An illustrative example of an Odin robot is presented in Figure 2.17(b). Such a robot has 8 actuation modules, 6 structure modules, and 7 CCP joint modules. The employed actuation

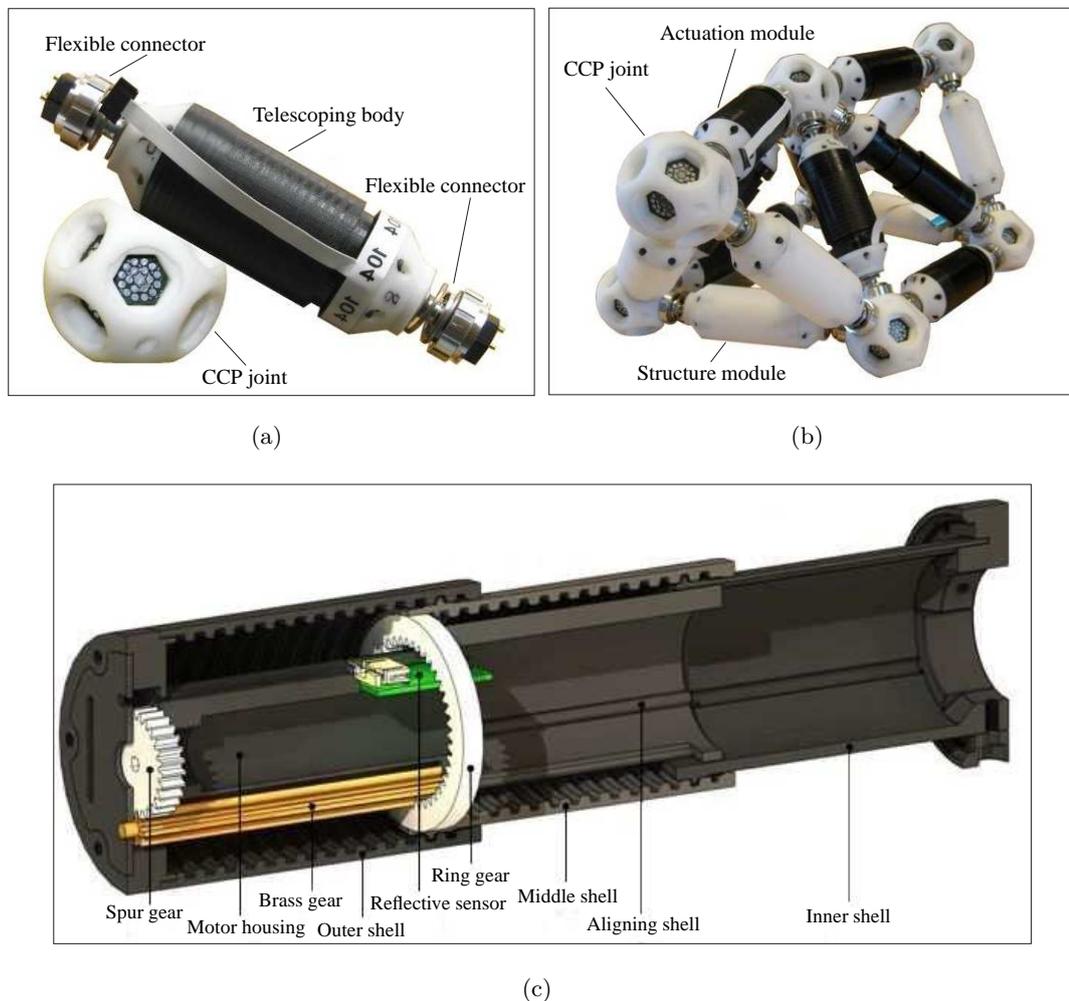
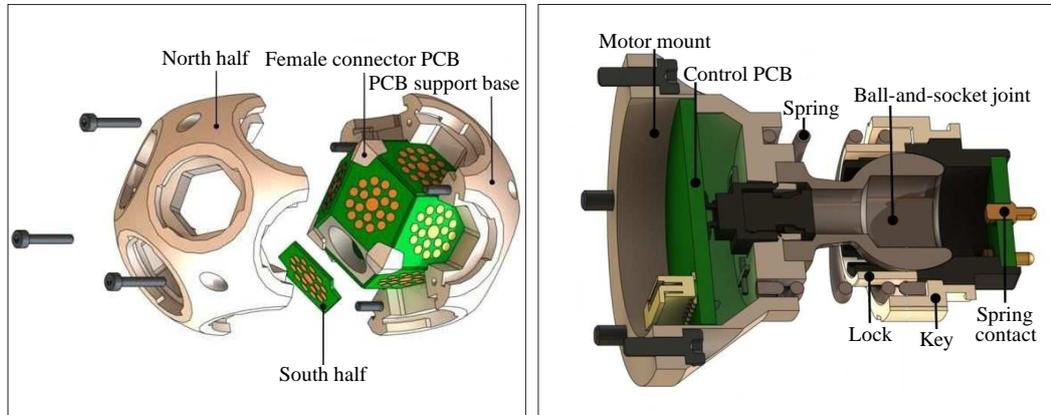


Figure 2.17: Prototypes of Odin Robot [Garcia *et al.*, 2009, Lyder *et al.*, 2008]. (a) Odin Modules. (b) Odin Robot. (c) Actuation Mechanism.

mechanism is shown in Figure 2.17(c). The telescoping body consists of four cylindrical shells, i.e., the outer, middle, inner, and aligning shells. The counter-clockwise internal thread of the outer shell is matched with a revolution of thread of the middle shell. The middle shell with internal clockwise threads can be driven by a DC motor coupled with a spur gear, a long gear and a ring gear. One revolution of thread of the inner shell fits with the internal threads of the middle shell. An aligning shell that can move linearly with the middle shell is used to align the outer and inner shells. A reflective sensor is employed to detect the length of the prismatic actuator. A CCP joint as shown in Figure 2.18(a) has 12 female connection slots of which each slot is equipped with a PCB for electrical connections including power sharing and a communication bus between connected strut modules. The flexible connector shown in Figure 2.18(b) has a spring and a ball-and-socket joint, which can provide passive compliance for a formed structure like the one presented in Figure 2.17(b). Besides, a PCB board and an electrical male connector are also incorporated into the flexible connector. In the Odin robot system, two different PCBs, i.e., a general board shown in Figure 2.18(c) and (d) and a specific board shown in Figure 2.18(e) and (f) are developed. The reusable general board contains an AT91SAM7S256 microcontroller for computation. For the actuation module, it also has a specific board carrying a motor driver.

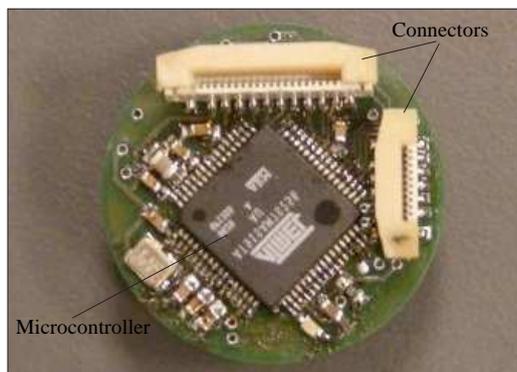
TET Walker Following the concept of tetrahedral robotics, NASA has designed and developed a series of tetrahedral walker robots [Curtis *et al.*, 2006, 2007a,b, Truskowski *et al.*, 2010], which results in a 12-TET walker robot shown in Figure 2.19(a). Such a 12-TET walker consists of 26 struts, 1 central payload node and 8 vertex connection nodes. Each strut consists of two prismatic actuators that point in opposite directions. The prismatic actuator exhibits nested screws with exoskeleton shown in Figure 2.19(b) to enjoy a high stroke length. The exoskeleton is keyed to prevent spinning, which keeps the nut from rotating and thus makes the system move linearly when the motor drives the inner screw. Figure 2.19(c)–(h) shows the CAD models and corresponding prototypes of a strut, a payload node and a vertex node. Each strut with embedded control electronics and battery is double sided and the two prismatic actuators are independently driven by two motors allowing flexibility to change the centre of gravity of the whole strut, which sometimes is desired for some locomotion gaits. The vertex nodes are based on a nine-faceted design involving passive joints that can not only eliminate play when struts are fixed but also allow full mobility when struts are moving.

2.3 Strut-Type Modules

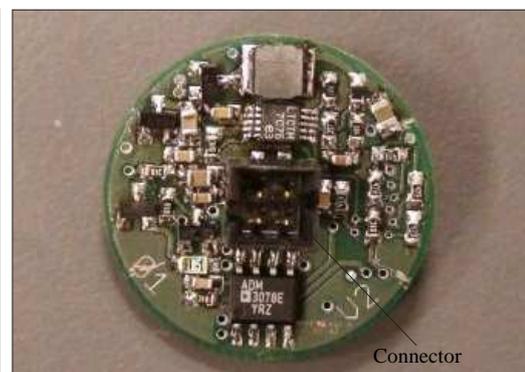


(a)

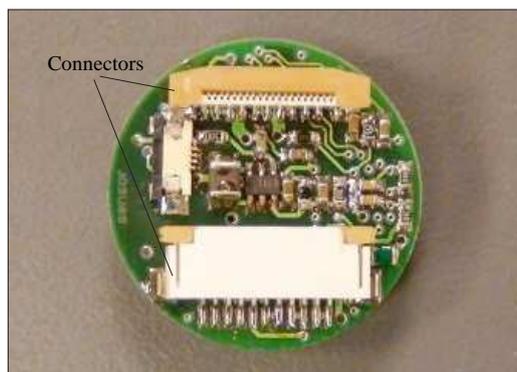
(b)



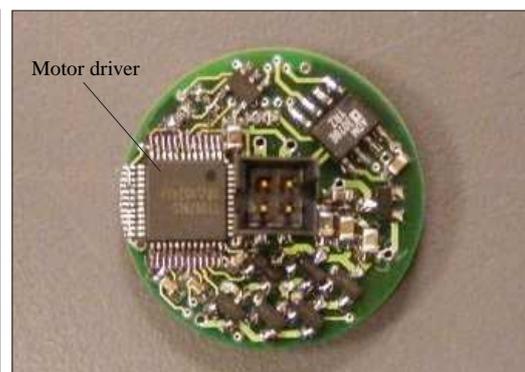
(c)



(d)



(e)



(f)

Figure 2.18: CCP Joint, Flexible Connector and Electronics Boards of Odin Robot [Garcia *et al.*, 2009, Lyder *et al.*, 2008]. (a) CCP Joint. (b) Flexible Connector. (c) Front of a General Board. (d) Back of a General Board. (e) Front of a Specific Board. (f) Back of a Specific Board.

2.3 Strut-Type Modules

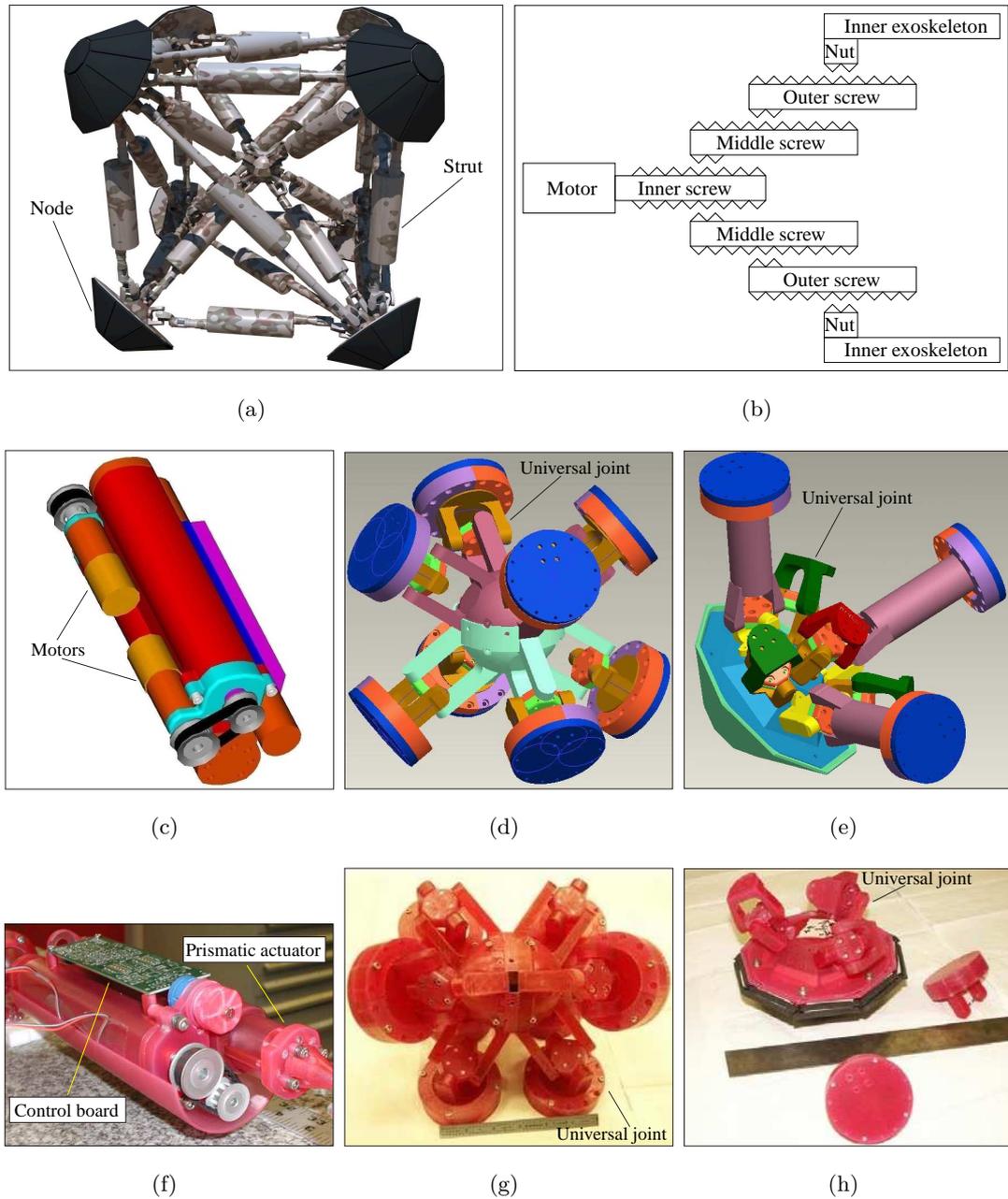


Figure 2.19: Conceptual Model, Actuation Mechanism, CAD Models and Physical Prototypes of a TET Walker Robot [Curtis *et al.*, 2006, 2007a,b]. (a) Conceptual Model. (b) Actuation Mechanism. (c) CAD of Strut. (d) CAD of Payload Node. (e) CAD of Vertex Node. (f) Physical Strut. (g) Physical Payload Node. (h) Physical Vertex Node.

The payload node was also endowed with same functionalities. The 12-TET walker is a mechanical and electrical realization of NASA’s conceptual tetrahedral robot, which has great significance in modular robotics. However, the 12-TET walker is too heavy to move and only simple motions of the robot are demonstrated. The robot is so complicated that an effective control method is still missing [Li *et al.*, 2014].

Amorphous Robot is a heterogeneous modular robot consisting of telescoping struts, spherical nodes and compliant connectors as shown in Figure 2.20(a) [Yu, 2010, Yu *et al.*, 2010]. Figure 2.20(b) shows the CAD model of a strut composed of two side-by-side stacked off-the-shelf prismatic actuators pointing in opposite directions. The compliant connector is presented in Figure 2.20(c) which is equipped with a ball-and-socket joint and a foam tube conferring some flexibilities on the connector. Such a compliant connector also enables the amorphous robot to possess deformation capabilities. Each strut has a light sensor to collect information from the environment. By considering a square robot shown in Figure 2.20(a) as a meta-module, different robotic structures can be formed by connecting such meta-modules. Amorphous robot adopts a simple design as the motivation of developing such a modular robotic system is to investigate a scalable multi-agent control algorithm [Yu, 2010].

OctaWorm shown in Figure 2.21 is a deformable modular robot designed to explore locomotion capabilities of deformable robots under confined environments such as cavities and pipes. Up to date, two prototypes of OctaWorm have been constructed [Zagal *et al.*, 2012]. The first one was based on hydraulic prismatic actuators while electric prismatic actuators were employed in the second prototype. This section only focuses on the electric prototype shown in Figure 2.21. An OctaWorm robot shown in Figure 2.21(a) consists of prismatic-actuator based struts and rubber ball anchoring nodes. Like Odin, struts connected with a node can rotate around the node surface. Figure 2.21(b)–(c) shows a prismatic-actuator based strut that follows the same principle of a power car antenna, which results in the extension ratio tending to around 6 : 1. The two prismatic actuators can be actuated by the equipped servo motor which is used to rotate a drum loaded with a plastic cable. Advantages of OctaWorm are simple design, light weight and low cost. However, the strut is susceptible to bend as it is too weak to form a massive structure. Besides, the demonstrated robot locomotion capability is limited, hence, using deformable robots to inspect blood vessels still rests on the

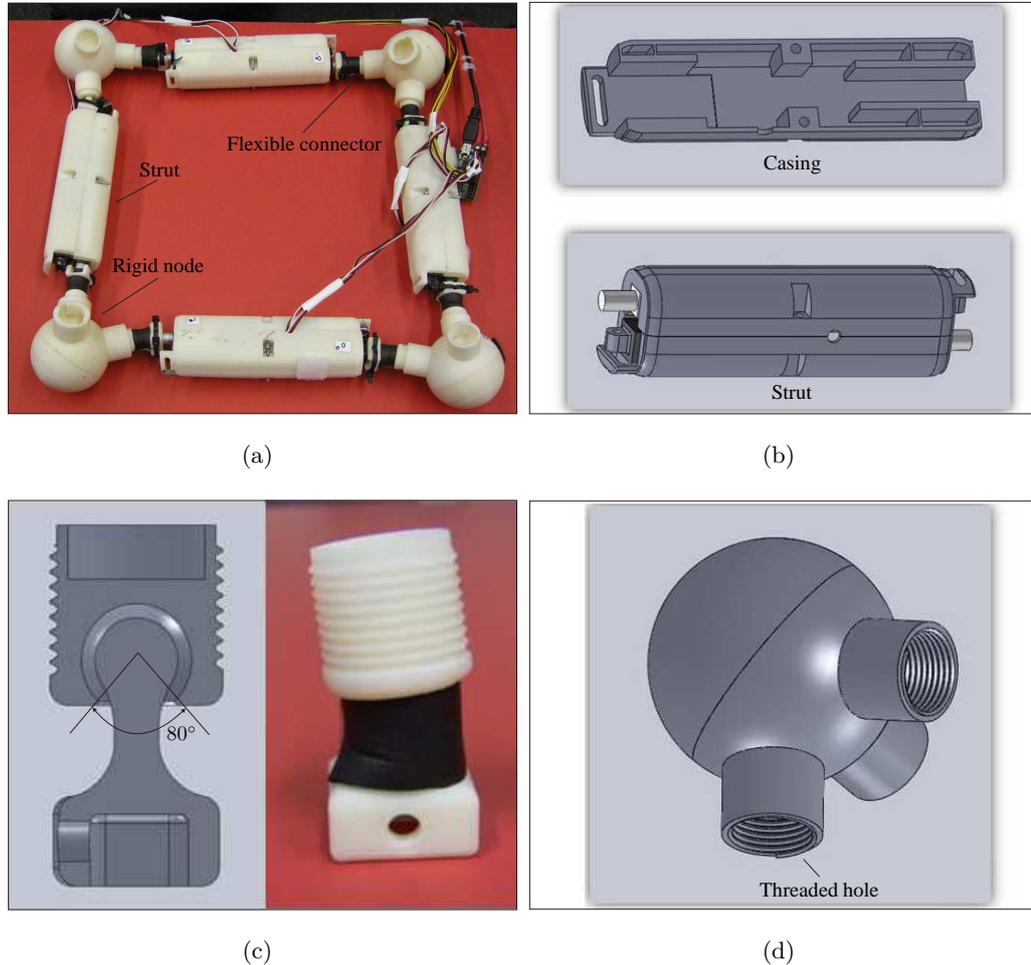
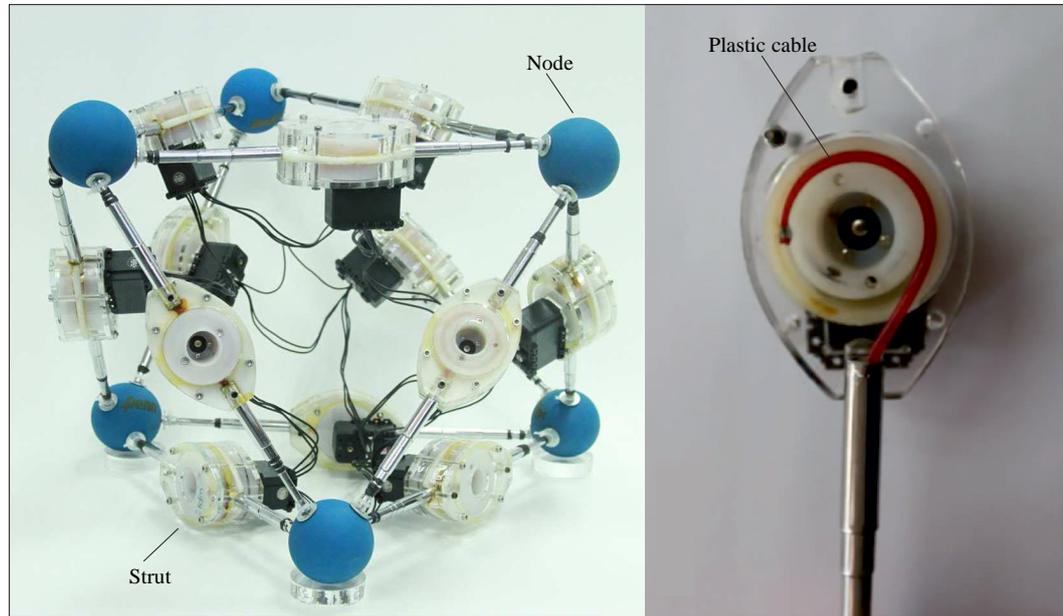


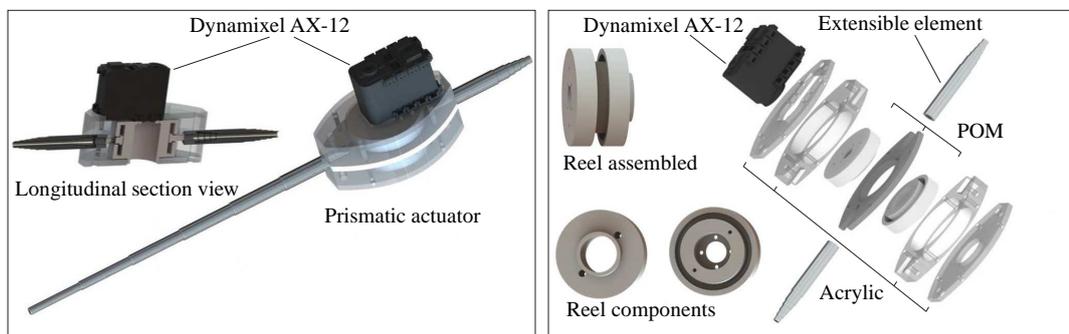
Figure 2.20: Amorphous Modular Robot [Yu, 2010, Yu *et al.*, 2010]. (a) Amorphous Robot. (b) CAD Model of the Strut. (c) Flexible Connector. (d) Node.

development of miniaturization technology and persevering effort of researchers.

As seen from the above strut-type modular robots, a TETROBOT structure formed by CMS joints is an ideal truss mechanism with all the struts jointed by a same node sharing a common centre of rotation. In Odin, TET Walker, Amorphous and Octa-Worm robotic structures, the rotation centre of each strut is located either on the surface of its connected node or other places. These robots with non-ideal nodes simplify the implementation process at the cost of introducing more complex kinematics. Features of the strut-type modules are compared in Table B.2 of Appendix B.



(a)



(b)

(c)

Figure 2.21: OctaWorm Robot [Zagal *et al.*, 2012]. (a) Robot Prototype. (b) CAD Model of the Strut. (c) Exploded View of the Strut.

2.4 Control Strategies

In robotics and system control, there are three types of controllers that are commonly investigated and employed as shown in Figure 2.22 [Frampton *et al.*, 2010].

- Centralized controller: a single controller processes the sensor measurements of all the subsystems and generates all the control signals at once.

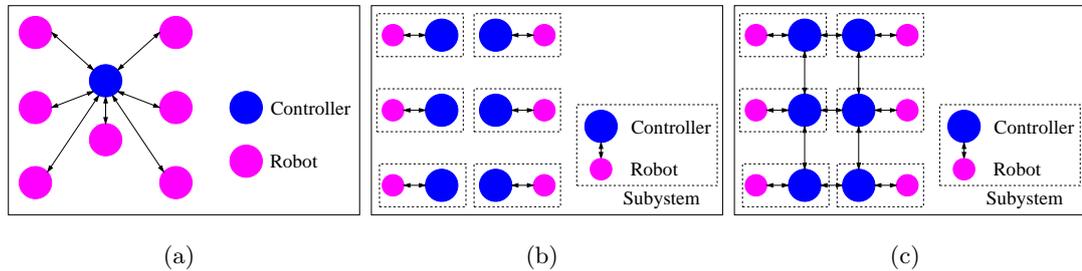


Figure 2.22: Three Controller Architectures. (a) Centralized Controller. (b) Decentralized Controller. (c) Distributed Controller.

- Decentralized controller: each subsystem has its own independent control loop that can make decisions based on its own measurements.
- Distributed controller: each localized controller is capable of communicating with its neighbouring controllers to share and exchange information.

A centralized controller is relatively simple to implement. However, it bears a high computational burden and is prone to be failed when applied to a large-scale robot system. Decentralized controller is scalable to robot size although the global performance lags behind a centralized controller. Distributed controller combines the advantages of centralized and decentralized controllers and it can be applied to accomplishing global objectives while retaining scalability.

Since the dawn of modular robotics, a lot of control strategies have been implemented for modular robots. Gait control tables are almost the simplest centralized method for controlling modular robots [Støy *et al.*, 2010]. Such a table contains sequential steps that each module within a modular robot should follow to achieve a locomotion gait of the robot as a whole. Gait control tables are extremely simple and are preferable for a simple purpose such as making a robot move. However, such a control method is dependent on the assigned identifications of modules and therefore it has no ability to deal with cases where modules are added, removed or interchanged. A gait control table is also limited for large-scale robotic structures as it would be a laborious process to manually design motions step by step for each module. Hormone-based method is a way to overcome the problems appeared in gait control tables [Fernández, 2014, Hamann *et al.*, 2010, Shen *et al.*, 2004]. Hormones of a module are messages that are utilized to synchronize motions and trigger actions of its neighbouring modules. Such

a method can be distributed and modules of a certain structure can be added, removed or interchanged. Unfortunately, it lacks robustness since it uses a hormone message that propagates through all the modules of a modular robot. If a module misses the hormone, the whole robot stops responding. Unlike the discrete way exploited in gait control tables and hormone-based control method, role-based control method uses a function of time to obtain smooth motions of modules and roles of modules are defined by the parameters associated with the actuation function [Støy *et al.*, 2002, 2010]. Each module takes cyclic actions for a specific time period and after completing the actions the module sends a message to its child module and receives a message from its parent module. If a message is received, the module resets its timer as zero. Modules in role-based control tend to be a little more independent compared with modules in hormone-based control. Since communication in role-based control is performed based on a parent-child architecture, robotic modules need to be connected carefully. Central pattern generator (CPG) neural network is a biologically inspired method that is one of the most popular control methods in modular robotics [Ijspeert *et al.*, 2007, Sato *et al.*, 2011, Watanabe *et al.*, 2009, Yu, 2010]. Mathematically, a CPG neural network is a system of coupled differential equations that can generate cyclic output signals. To achieve a certain task, such signals can be used to actuate robot joints and coordinate joint movements by adjusting coupling parameters within a CPG neural network. Evolutionary algorithms have been widely and maturely investigated to evolve values of coupling parameters self-adaptively to find the optimal values for accomplishing some tasks such as obtaining a locomotion gait that can traverse as far as possible within a certain time period [Caluwaerts *et al.*, 2014, Christensen *et al.*, 2010b, Spröewitz *et al.*, 2008]. A cellular automata was implemented for controlling locomotion through reconfiguration of thousands of simulated UBot modular robots [Zhu *et al.*, 2015]. The scalability of a phase-automata based distributed control method developed for chain-type PolyBot modular robots has been validated by using a physical snake robot with 55 modules [Zhang *et al.*, 2003]. A distributed control framework for modular robots covering a broad range of control strategies such as neural networks, genetic algorithms and CPG was proposed in [Christensen *et al.*, 2011]. Apart from controller evolution, some researchers have implemented strategies for evolving morphology and controller of modular robots simultaneously [Faña *et al.*, 2013, 2015, Guettas *et al.*, 2014, Marbach & Ijspeert, 2004]. These evolutionary methods are based on computer simulations,

which are applicable to chain- or tree-type modular robots. Unfortunately, similar frameworks are still missing for lattice-type modular robots.

2.5 Communication Methods

Communication is of paramount significance to modular robots especially robotic systems that rely on a message-passing network between individual and autonomous modules to achieve some tasks as a whole [Cabrera *et al.*, 2011, Fitch & Lal, 2009, Kuo & Fitch, 2014, Kuo, 2013]. There exist numerous wired and wireless communication strategies. On one hand, wired solutions tend to be more reliable and can transmit in a higher rate as compared with the wireless counterparts. On the other hand, wired solutions depend on physical connections which leads to some limitations in terms of maintenance and flexibility.

In general, communications between electronics components within a same module are implemented through wired strategies [Lyder *et al.*, 2008, Østergaard *et al.*, 2006, Salemi *et al.*, 2006, Spröwitz *et al.*, 2014]. Wired communication protocols including serial, pulse width modulation (PWM), serial peripheral interface (SPI), universal asynchronous receiver/transmitter (UART), I2C, and CAN have been widely applied in embedded systems [Heslinga *et al.*, 2012]. A serial bus is composed of two wires of which one is used for sending data and the other one for receiving data [Heslinga *et al.*, 2012]. Hence, serial devices possess two pins: a TX pin for transmitting and an RX pin for receiving. For a device, its TX and RX pins are depicted with respect to itself and therefore the TX pin of a device should be connected with the RX pin of another device and vice versa. PWM is a way of controlling analogue circuits digitally by changing the frequency and duty cycle of a digital square wave [Wu *et al.*, 1999]. This technique is commonly used for controlling motor speed [Ghosh *et al.*, 2016, Kundu *et al.*, 2017, Salemi *et al.*, 2006]. SPI is a synchronous bus containing four wires: clock (SCK), chip select (CS), master-in-slave-out (MISO), and master-out-slave-in (MOSI) [Leens, 2009]. When conducting communication, there would be a master device generating the clock signal for synchronization. The master uses the MOSI data line to transmit messages to its slave devices. The MISO data line is used for transmitting data back from slave devices to the master device. The CS is used to enable the chip to send or receive data. UART works as an intermediate interface between serial and parallel interfaces taking bytes of data in parallel and transmitting separate bits serially [Fang & Chen,

2011]. An I2C bus has two signals: a data signal SDA and a clock signal SCL [Leens, 2009]. I2C is advantageous over SPI as it needs less pins and supports multi-master communication systems. As a broadcast type of bus, a CAN bus consists of two data lines: CANH and CANL [Tindell *et al.*, 1994]. All nodes connected to the CAN bus can obtain the messages and each node can react to its interested message resorting to the local filter.

Wireless communication strategies have been widely applied in modular robotics for inter-module communication. Bluetooth [Salemi *et al.*, 2006, Spröwitz *et al.*, 2014], IR transmitters/receivers [Østergaard *et al.*, 2006, Wei *et al.*, 2011], Wi-Fi [Bruce *et al.*, 2014, Putra *et al.*, 2017], ZigBee [Fitch & Lal, 2009, Huasong *et al.*, 2010] and RF [Cabrera *et al.*, 2011, Kuo & Fitch, 2014, Kuo, 2013] are commonly used. Bluetooth uses radio waves in the unlicensed industrial, scientific and medical (ISM) band from 2.4 to 2.485 GHz to exchange data over short distances [Ferro & Potorti, 2005, Putra *et al.*, 2017]. A broad range of consumer electronics such as mobile phones and laptops are integrated with Bluetooth, this facilitates the realization of communication between a designed robotic system and other Bluetooth supported devices like a laptop that can act as a master controller. However, Bluetooth is not scalable as it is only applicable to small networks and requires a central node coordinator to be functional [Cabrera *et al.*, 2011]. IR transmitters/receivers are utilized for conveying data using IR radiation over short distances. IR devices are versatile and can be applied to obstacle avoidance and proximity detection. One drawback of IR is that it suffers from the limitation of line-of-light propagation [Kuo, 2013]. This means a visually unobstructed straight line should exist between a pair of transmitter and receiver to perform communication. Besides, it is difficult to implement a reliable IR communication network due to crosstalk [Østergaard *et al.*, 2006]. Like Bluetooth, a lot of consumer devices support Wi-Fi. Wi-Fi normally utilizes both the 2.4 GHz and 5 GHz bands and has a higher transmit rate than other solutions like Bluetooth [Ferro & Potorti, 2005]. Wi-Fi can transmit and receive data simultaneously and work over a relatively longer distance of 100 m but it needs to consume more power [Heslinga *et al.*, 2012]. ZigBee uses a band of 2.4 GHz to exchange information at a relatively low rate of 250 kbits/s. ZigBee is suitable to creating star or mesh networks and can work without a central node [Fitch & Lal, 2009]. Therefore, ZigBee is more scalable than Bluetooth and has been used for constructing large-scale wireless sensor networks. RF could be an alternative low-cost solution

2.6 Passive Compliance and Multiple-State Actuators

of ZigBee for establishing mesh communication networks [Hackaday, 2016, Ngajieh & Weiber, 2015].

Regarding communication network topologies, global and local communications have been maturely studied in modular robotics [Garcia *et al.*, 2009, 2007, Kuo, 2013]. A global bus allows modules in a robotic structure to exchange information with each other in a direct way. It is simple to achieve synchronization and coordination through a global bus but channel saturation would gradually become a problem with the increase of robot size. Local communication only allows a module to mutually communicate with its neighbouring modules. Global communication can be retained through local communication, where messages are propagated in a hop-by-hop mode. Unlike global communication, local communication can also be utilized for topology identification, i.e., determination of an unknown configuration of a modular robotic structure. In [Garcia *et al.*, 2009], the efficiencies of global and local communications are compared based on different modular robots and different robot configurations. As claimed, global and local communications are suitable for centralized and distributed controllers, respectively. Besides, global communication is generally applicable to robot configurations with low connectivities such as chains and trees. Local communication can be faster than global communication when dealing with information exchanged between distant robotic modules that are not too far apart. A hybrid communication framework was proposed and investigated in [Garcia *et al.*, 2009, 2007] with both global and local communications involved. The framework is self-reconfigurable and can be used to form arbitrary communication network topologies. A ZigBee communication mesh network with up to 15 nodes was investigated in [Fitch & Lal, 2009], which possesses some advantages such as simplicity and low cost. However, it cannot be used for topology identification and its scalability is also limited. Fortunately, following effort was made by researchers leading to a scalable multi-radio architecture for neighbour-to-neighbour communication [Kuo & Fitch, 2011, 2014, Kuo, 2013], which can be used for automatic detection of neighbouring modules and is applicable to decentralized controllers.

2.6 Passive Compliance and Multiple-State Actuators

A passive compliance mechanism consists of passive joints that are not motorized but can be moved freely due to external forces. The benefits of passive compliance have been widely studied in the literature [Aghili & Parsa, 2009, Caluwaerts *et al.*, 2013,

2.6 Passive Compliance and Multiple-State Actuators

2014, Christoforou *et al.*, 2015, Grosch *et al.*, 2010, Horchler *et al.*, 2015, Li *et al.*, 2007, Merat *et al.*, 2013, Sato *et al.*, 2011, Watanabe *et al.*, 2009]. Theo Jansen mechanism is one of the solutions to using passive dynamics to achieve locomotion gaits [Jansen, 2008, Nansai *et al.*, 2013]. Strandbeests presented in [Jansen, 2008] are kinetic sculptures powered by wind that can mimic a multi-legged walking pattern. There exist numerous bipedal robots using passive dynamics to walk or run [Owaki *et al.*, 2011, Renjewski *et al.*, 2015]. Such passive-dynamic locomotion is accomplished through the interaction of a compliant robot structure and its working environment. Since the robot has no actuator and controller, this kind of passive movements are energy efficient. Tensegrity robotic structures composed of rigid bars and elastic cables are a kind of highly compliant structures [Caluwaerts *et al.*, 2013, 2014]. Such structures could be extremely lightweight and can passively respond to external forces that stem from their working environments. In [Sato *et al.*, 2011, Watanabe *et al.*, 2009], researchers investigated a snake robot system wherein each joint is equipped with a servo motor and an elastic spring. Due to the elasticity of the spring, each joint does not strictly follow motor commands from the controller but is also somewhat affected by the working environment and the physical constraints of the body. Such a snake robot with compliant joints exhibits adaptive behaviours and can respond to environmental changes. A reconfigurable robot manipulator with passive lockable cylindrical joints was proposed and investigated in [Aghili & Parsa, 2009, Merat *et al.*, 2013], where each cylindrical joint can be treated as a combination of two passive revolute and prismatic joints. The reconfiguration process of the robot manipulator can be completed by following a series of steps related to releasing and locking the cylindrical joints. Grosch *et al.* employed passive lockable revolute joints to reduce the mechanical complexity and approximate the full mobility of the conventional Gough-Stewart platform [Grosch *et al.*, 2010]. Only four legs were used to construct a parallel platform that can approximate the full degrees-of-freedom of its conventional counterpart with six legs. In [Li *et al.*, 2007], a hybrid joint with passive and active modes was proposed to secure human safety in potential collisions between humans and robots. The working modes of the hybrid joint can be exchanged by releasing and engaging an electromagnetic clutch. A compliant revolute actuator composed of a DC motor and a flexible unit with multiple working modes was proposed in [Wang *et al.*, 2016], which not only can serve as a non-compliant actuator but also change its stiffness to adjust to external forces. Besides, the actuator

is claimed to have a totally passive state in which the DC motor cannot transmit torque to the output shaft. However, as mentioned in [Wang *et al.*, 2016], such a passive state is dangerous since the mechanical limit can be reached.

Both the aforementioned robotic modules and the above joints/actuators can be considered as basic elements in a certain robotic system. Therefore, it is worth exploring the modularity of joints/actuators with multiple working states. The features of block-type and strut-type modules are compared in Tables B.1 and B.2 of Appendix B. From the current systems, some useful information can be summarized to obtain a better understanding for developing a modular robotic system.

- 1) Servomotors, DC motors, stepper motors and lead screws are widely adopted in the current actuator systems.
- 2) Connector mechanisms generally employ permanent magnets, electromagnets, hook-hole based mechanical devices or ball-and-socket joint based flexible connectors. Note that a connector mechanism and a locking mechanism could share same mechanical solutions in most cases.
- 3) Inter-module communication can be achieved through SPI and I2C buses while communication between modules and environments usually relies on CAN bus, IR sensors, Bluetooth, RF transceivers, XBee transceivers or ZigBee protocols.
- 4) Biologically inspired methods such as CPG network and digital hormone based control strategies are most popular and effective.
- 5) In most systems, robotic modules have either revolute actuators or prismatic actuators, while in each ModRED module, revolute and prismatic actuators coexist.
- 6) A robotic module can be actively controlled to exhibit multiple working modes. For example, each Roombots module has three actuation modes, i.e., oscillation, rotation and locked modes.

2.7 Conclusions

This chapter mainly introduced and reviewed a series of typical modular robotic modules regarding mechatronics design and hardware implementation. As compared with block-type modular robots, less attention has been given to strut-type modular robots

in the modular robotics community. Strut-type modules are capable of forming parallel truss mechanisms with inherent stability to be more suitable for industrial activities such as load transportation. An ideal strut-type modular robot consists of robotic struts and connector nodes. Ideally, struts connected to a same node have a common centre of rotation located on the intersection of the extended lines of the struts. Physically, an ideal compliant node is difficult to implement. The TETROBOT is a solution to an ideal strut-type robot. However, CMS joints used in TETROBOT must be built with greater precision to avoid undesirable sloppiness and play. Besides, CMS joints cannot be used to form chain-type structures and reconfiguration of MRSs formed by CMS joints is difficult as removing struts usually requires disassembly of the whole robotic structure. A workaround solution is to place the rotation centre of each strut on other places such as the node surface. Although the workaround solution makes physical implementation relatively easier, it complicates the modelling process and increases the control and kinematics complexity. Additionally, since the passively movable joints attached to the node inevitably take up spaces, the connector node solution is hard to achieve miniaturization for potential future applications. These limitations of the existing implemented strut-type modules motivate us to explore a new way of constructing strut-type modular robotic structures wherein struts jointed by a same node are expected to intersect at a same point and the nodes and passive joints are isolated to implement in a simpler way.

The actuators employed by existing modular robotic modules are all actively controlled. Each DOF of a Roombots module can exhibit oscillation, rotation and locked modes. However, the locked state is actively achieved by the position-controlled actuator. These three modes also can be easily obtained by other actuators such as a servomotor using a PID controller. The aforementioned passive lockable joints are not motorized and their motions cannot be actively controlled. Existing revolute actuators with multiple working states cannot achieve all of the expected actuated, locked and passive states. Specifically, the hybrid actuator mentioned in Section 2.6 was inherently designed to have actuated and passive states. Even though the hybrid actuator can be actively controlled to be locked like the Roombots module, it consumes energies during such a state. The previously discussed compliant revolute actuator can achieve the actuated, locked and passive states. Unfortunately, mechanical limits would be reached in the passive state, which is dangerous. The work in this research is motivated by the

lack of a prismatic actuator that allows the actuated, locked and passive states to be physically realizable. Such three-state prismatic actuator has ability to overcome the limitations of lockable passive joints used in the conceptual reconfigurable robotic arm discussed in Chapter 1.

CHAPTER 3

Connectivity of Strut-Type Modular Robots

A fundamental issue of modular robotic systems is how to use modular elements to form a robotic structure. Different connectivity strategies can lead to different kinematic structures. This chapter explores the connectivity of strut-type modular robots.

3.1 Connectivities

In general, a strut-type modular robot is composed of robotic struts and connector nodes. The robotic struts are built from prismatic actuators that can extend or contract linearly, while the connector nodes are elements serving for interconnection of robotic struts and allowing for rotations. Many commercial prismatic actuators can be readily used in a strut-type modular robot [Hamlin & Sanderson, 1997]. The major difficulty arises in the connector nodes of the modular robot. Ideally, a connector node has several passively movable joints to allow rotary motion of robotic struts connected to the connector node. The struts linked by a same connector node share a common centre of rotation indicating that all of the movable joints of a connector node must be concentric [Hamlin & Sanderson, 1997]. Figure 3.1(a) illustrates an idealized strut-type MRS and Figure 3.1(b) shows the deformation of such an idealized structure. The struts jointed by a same node intersect at the node centre and are capable of rotating around the intersection point. Mechanically, such idealized point-like compliant nodes are difficult to implement [Lyder, 2010].

A workaround solution to overcoming the physical implementability problem is to use connectivity method shown in Figure 3.1(c) where each strut has a centre of rotation on the surface of its linked node. As shown in Figure 3.1(d), the structure also has deformation capability. The drawback of this workaround node solution is the introduction of offsets shown in Figure 3.1(c), which increases the kinematics complexity. A new connectivity is put forward as shown in Figure 3.1(e) where prismatic actuators are rigidly connected to the nodes. Unlike the way of placing revolute joints on the nodes, a revolute joint is in the middle of two prismatic actuators to achieve deformation of the structure as illustrated in Figure 3.1(f). Struts connected to a same node truly intersect at the node centre and the offsets appeared in Figure 3.1(c) and (d) are eliminated. More importantly, separate rigid connector nodes are considerably easier to design and manufacture as compared with idealized and the workaround connector nodes with integrated passive revolute joints. For building large-scale MRSs, this solution may facilitate and speed up the mass production process as rigid nodes and passive joints

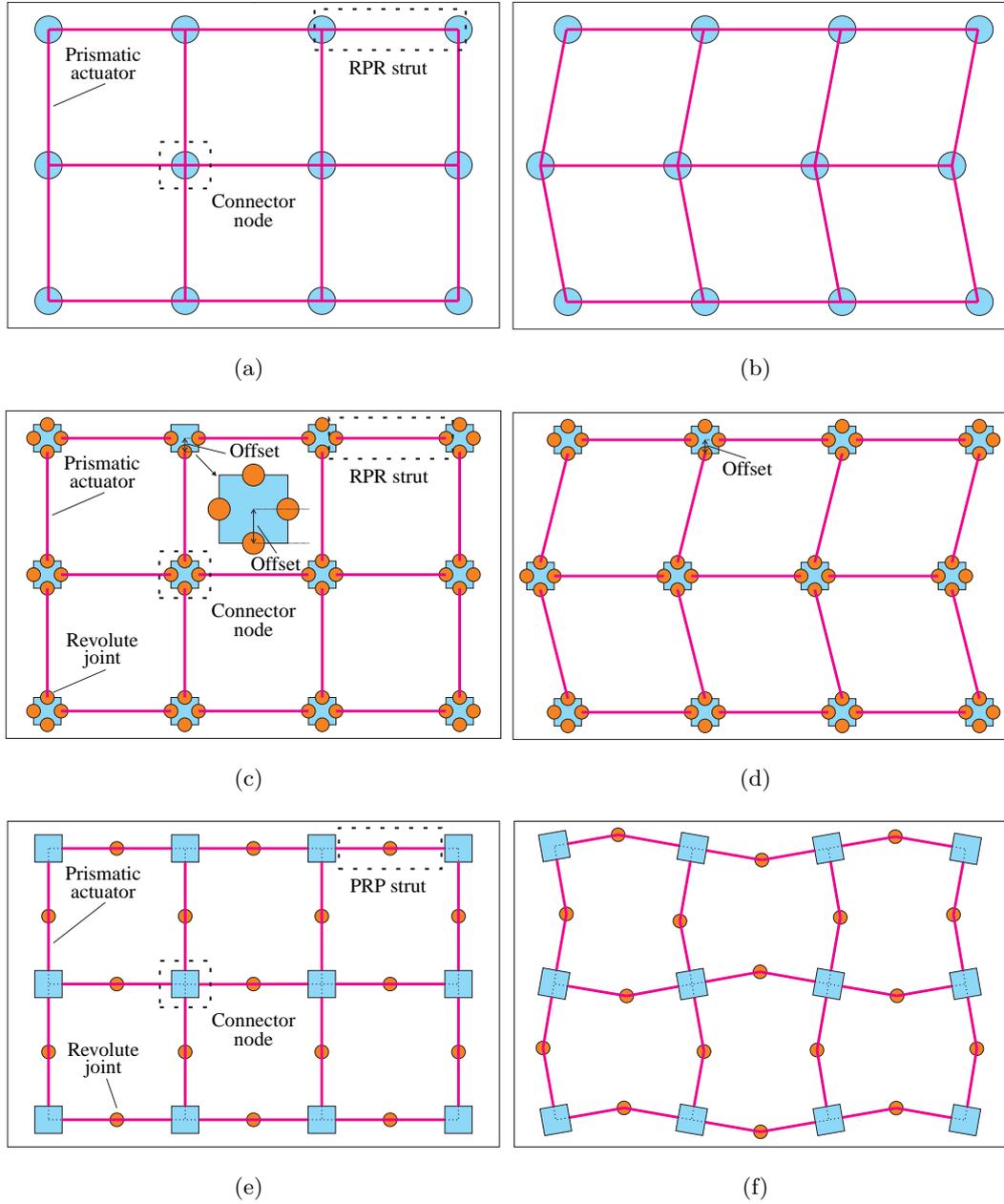


Figure 3.1: Different Connectivities of a Strut-Type MRS. (a) Idealized Connectivity. (b) Deformation of Idealized Connectivity. (c) Connectivity I. (d) Deformation of Connectivity I. (e) Connectivity II. (f) Deformation of Connectivity II.

are separate but this also leads to a research problem: would this new connectivity solution be feasible and/or advantageous over the conventional solution as illustrated

in Figure 3.1(c)?

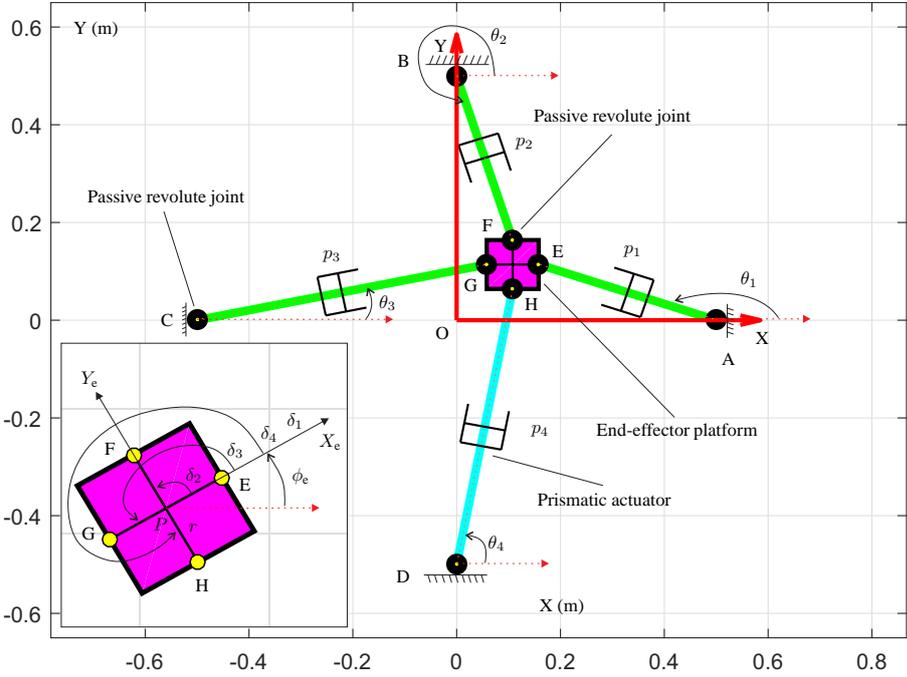
3.2 Kinematics Analysis

This section comparatively analyses kinematics of two parallel structures formed by using the two connectivities I and II illustrated in Figure 3.1(c) and (e). By treating the element between two nodes as a robotic strut, struts shown in Figure 3.1(c) and (e) can be separately termed as RPR and PRP struts, where R and P represent revolute and prismatic, respectively. Two parallel mechanisms, i.e., a 4-RPR and a 4-PRP mechanisms, are then extracted from Figure 3.1 (c) and (e) as basic cells for comparison. Both of the two mechanisms can be used as planar parallel manipulators [Ashok, 2014, Williams II & Shelley, 1997]. Figure 3.2 presents the two simulated parallel mechanisms of which each prismatic actuator has its fully contracted and extended lengths of 0.3 m and 0.6 m, respectively. The central node is considered as an end-effector platform of the parallel mechanisms. The 4-RPR mechanism shown in Figure 3.2(a) consists of four prismatic actuators and eight passive revolute joints. The four outer passive revolute joints are located at four fixed points A, B, C and D on the floor. Hence, the four outer revolute joints are restricted to passively rotate around fixed points. The inner passive revolute joints are floating and can passively rotate and change their positions when the prismatic actuators are engaged. The 4-PRP mechanism presented in Figure 3.2(b) has eight prismatic actuators and four passive revolute joints. Each of the prismatic actuators $p_{1,1}$, $p_{2,1}$, $p_{3,1}$ and $p_{4,1}$ has one tip fixed at points A, B, C and D on the floor, indicating that angles $\theta_{1,1}$, $\theta_{2,1}$, $\theta_{3,1}$ and $\theta_{4,1}$ are constant. The revolute joints are floating and can passively rotate and change their positions owing to the collaborative work of the prismatic actuators. Note that the 4-PRP mechanism adopts a Gammadion cross shape to ensure that it exhibits a non-singular configuration. This section first analyses the DOFs of the two mechanisms and then compares their kinematic equations, workspaces and manipulability.

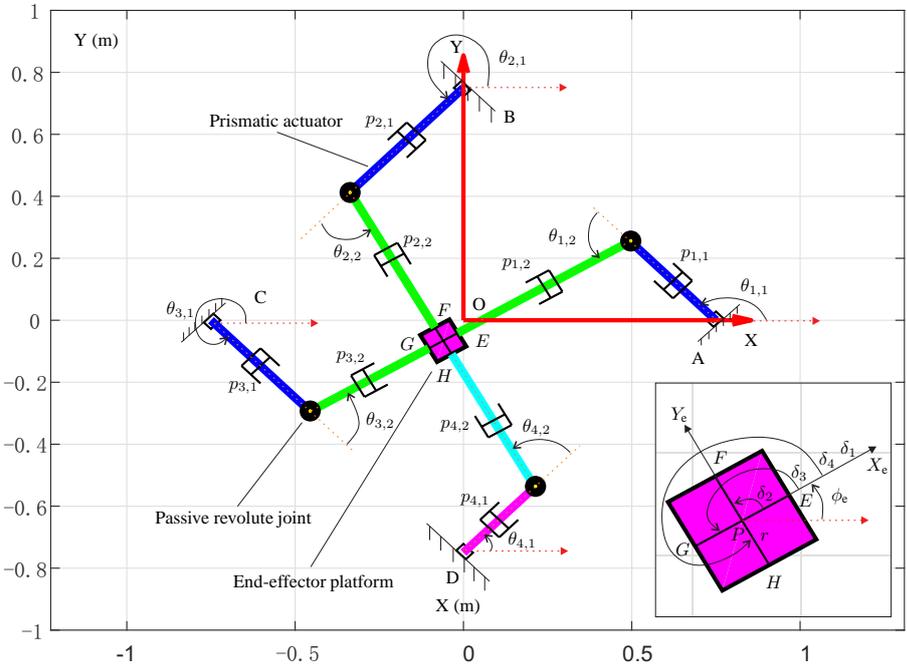
3.2.1 Degrees of Freedom

The first step to investigate a mechanism is to analyse its mobility, i.e., degrees of freedom. The DOF of a two-dimensional (2D) mechanism can be calculated using

3.2 Kinematics Analysis



(a)



(b)

Figure 3.2: Nomenclature of Two Parallel Mechanisms with Fixed Points A, B, C and D. (a) A 4-RPR Parallel Mechanism. (b) A 4-PRP Parallel Mechanism.

Grübler's formula [Hamlin & Sanderson, 1998] of which the expression is depicted as

$$m = 3(l - j - 1) + \sum_{i=1}^j n_i,$$

where m is the DOF of the mechanism, l denotes the number of links, j indicates the number of joints and n_i represents the DOFs of the i th joint. As illustrated in Figure 3.2, both of the 4-RPR and the 4-PRP mechanisms possess $l = 10$ links and $j = 12$ joints with $n_i = 1$ for any $i = 1, 2, 3, \dots, j$. More specifically, in the 4-RPR mechanism shown in Figure 3.2(a), each prismatic actuator has two links. The end-effector platform and the floor are two links. Thus, the total number of links of the 4-RPR mechanism is 10. Similarly, for the 4-PRP mechanism, each passive revolute joint has two links. The end-effector platform and the floor are also two links. Therefore, the total number of links of the 4-PRP mechanism is also 10. This means that the two mechanisms with different connectivities have a same DOF of $3 \times (10 - 12 - 1) + 12 \times 1 = 3$ if the central node is floating and all of the revolute joints are passive.

3.2.2 Kinematics

This subsection mathematically derives the kinematic equations and inverse-kinematics solutions of the two parallel mechanisms to further discuss and compare their performance and differences.

4-RPR Mechanism

Let the pose of the end-effector platform shown in Figure 3.2(a) denote as position $P(x_e, y_e)$ and heading angle ϕ_e . Considering point $E(x_E, y_E)$ shown in Figure 3.2(a) to break the closed loop and using (x_i, y_i) with $i = 1, 2, 3, 4$ to denote positions of fixed points A, B, C and D in Figure 3.2(a), it is readily to have

$$\begin{bmatrix} x_E \\ y_E \end{bmatrix} = \begin{bmatrix} x_1 + p_1 \cos(\theta_1) \\ y_1 + p_1 \sin(\theta_1) \end{bmatrix}. \quad (3.1)$$

The position of point E can also be expressed with respect to point P :

$$\begin{bmatrix} x_E \\ y_E \end{bmatrix} = \begin{bmatrix} x_e + r \cos(\phi_e + \delta_1) \\ y_e + r \sin(\phi_e + \delta_1) \end{bmatrix}. \quad (3.2)$$

Equations (3.1) and (3.2) leads to the following equation [Ashok, 2014, Williams II & Shelley, 1997]:

$$\begin{bmatrix} x_1 + p_1 \cos(\theta_1) \\ y_1 + p_1 \sin(\theta_1) \end{bmatrix} = \begin{bmatrix} x_e + r \cos(\phi_e + \delta_1) \\ y_e + r \sin(\phi_e + \delta_1) \end{bmatrix}.$$

Similarly, the other equations can be obtained by considering points F , G and H to break the closed loop. The complete kinematic equation of the 4-RPR parallel mechanism is readily formulated as

$$\begin{bmatrix} x_i + p_i \cos(\theta_i) \\ y_i + p_i \sin(\theta_i) \end{bmatrix} = \begin{bmatrix} x_e + r \cos(\phi_e + \delta_i) \\ y_e + r \sin(\phi_e + \delta_i) \end{bmatrix}, \quad (3.3)$$

where r and δ_i with $i = 1, 2, 3, 4$ shown in Figure 3.2(a) are constants and for an idealized mechanism, parameter r equals 0 indicating that the workaround connector node solution does increase kinematics complexity.

In the 4-RPR mechanism, there exist eleven variables, that is x_e , y_e , ϕ_e , θ_i and p_i (with $i = 1, 2, 3, 4$). Equation (3.3) has eight equations since the 4-RPR mechanism consists of four serial chains and for each serial chain there exist two equations. If the number of unknown variables is greater than 8, the system is under-determined with no solution or an infinite amount of solutions. In contrast, if the number of unknowns is less than 8, the system is over-determined and may have solutions in a case where it includes some equations that can be eliminated. To obtain a solvable system, eight unknown variables including x_e , y_e , ϕ_e , θ_i (with $i = 1, 2, 3, 4$) and p_4 are defined when considering the forward-kinematics problem: obtaining the eight unknown variables with p_1 , p_2 and p_3 known. It is worth pointing out here that solving the forward-kinematics problem of a parallel mechanism could be very difficult and obtaining a closed-form solution may be impossible. Thus, a numerical technique (specifically, the *fsolve* function) integrated in Matlab is employed to evaluate the unknown quantities.

Unlike the forward-kinematics problem, the inverse-kinematics problem of a parallel mechanism usually has a closed-form solution. In the studied 4-RPR mechanism case, with three priorly known variables x_e , y_e , ϕ_e , the other unknown variables can be obtained as [Ashok, 2014, Williams II & Shelley, 1997]:

$$p_i = \sqrt{(x_e + r \cos(\phi_e + \delta_i) - x_i)^2 + (y_e + r \sin(\phi_e + \delta_i) - y_i)^2} \quad (3.4)$$

and

$$\theta_i = \tan^{-1}\left(\frac{y_e + r \sin(\phi_e + \delta_i) - y_i}{x_e + r \cos(\phi_e + \delta_i) - x_i}\right), \quad (3.5)$$

with $i = 1, 2, 3, 4$, which is quite straightforward.

By differentiating Equation (3.3), the velocity-level kinematic equation of the 4-RPR mechanism can be obtained, which is compactly formulated as [Ashok, 2014, Williams II & Shelley, 1997]:

$$A\dot{\chi} = B\dot{\vartheta} \quad (3.6)$$

where

$$A = \begin{bmatrix} 1 & 0 & -r \sin(\phi_e + \delta_1) \\ 1 & 0 & -r \sin(\phi_e + \delta_2) \\ 1 & 0 & -r \sin(\phi_e + \delta_3) \\ 1 & 0 & -r \sin(\phi_e + \delta_4) \\ 0 & 1 & r \cos(\phi_e + \delta_1) \\ 0 & 1 & r \cos(\phi_e + \delta_2) \\ 0 & 1 & r \cos(\phi_e + \delta_3) \\ 0 & 1 & r \cos(\phi_e + \delta_4) \end{bmatrix}, \quad \dot{\chi} = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\phi}_e \end{bmatrix},$$

$$B = \begin{bmatrix} -p_1 s_1 & 0 & 0 & 0 & c_1 & 0 & 0 & 0 \\ 0 & -p_2 s_2 & 0 & 0 & 0 & c_2 & 0 & 0 \\ 0 & 0 & -p_3 s_3 & 0 & 0 & 0 & c_3 & 0 \\ 0 & 0 & 0 & -p_4 s_4 & 0 & 0 & 0 & c_4 \\ p_1 c_1 & 0 & 0 & 0 & s_1 & 0 & 0 & 0 \\ 0 & p_2 c_2 & 0 & 0 & 0 & s_2 & 0 & 0 \\ 0 & 0 & p_3 c_3 & 0 & 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & p_4 c_4 & 0 & 0 & 0 & s_4 \end{bmatrix} \quad \text{and} \quad \dot{\vartheta} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{p}_1 \\ \dot{p}_2 \\ \dot{p}_3 \\ \dot{p}_4 \end{bmatrix}$$

with $s_i = \sin(\theta_i)$ and $c_i = \cos(\theta_i)$ for $i = 1, 2, 3, 4$. Thus, it is readily to have Jacobian matrix $J = A^\dagger B$ with A^\dagger denoting the pseudoinverse of matrix A such that $\dot{\chi} = J\dot{\vartheta}$ and Yoshikawa manipulability index $w = \sqrt{|JJ^T|}$ for evaluation of the manipulability of the parallel mechanism [Lynch & Park, 2017]. Given a desired end-effector path, three open- and closed-loop control laws can be employed to obtain joint velocities $\dot{\vartheta}$ for tracking a predetermined end-effector path [Agarwal & Vishwajeet, 2009, Davis & Nagasamudram, 2016]:

$$\dot{\vartheta} = J^\dagger \dot{\chi}, \quad (3.7)$$

$$\dot{\vartheta} = J^\dagger \dot{\chi} + \kappa(\vartheta_d - \vartheta_a), \quad (3.8)$$

$$\dot{\vartheta} = J^\dagger (\dot{\chi} + \kappa(\chi_d - \chi_a)), \quad (3.9)$$

where Equation (3.7) is an open-loop control law. Equation (3.8) is a joint space closed-loop law with ϑ_d and ϑ_a indicating desired and actual joint positions, respectively. Equation (3.9) is a task space closed-loop law with χ_d and χ_a separately indicating desired and actual end-effector poses. In addition, positive parameter κ in Equations (3.8) and (3.9) denotes the feedback gain of which the value is tuned manually and empirically ranging from a small value that is larger than 0 to a large value of 200 to finish the path-tracking task as satisfactory as possible. It is worth mentioning that the parallel mechanism would exhibit singular configurations when J^\dagger is not solvable. This research assumes that the solution of J^\dagger always exists and the investigation of singular cases is beyond the scope of this thesis.

4-PRP Mechanism

Following a similar procedure of breaking the closed loop by considering points E , F , G and H shown in Figure 3.2(b), the position-level kinematic equation of the 4-PRP mechanism is derived and obtained as

$$\begin{bmatrix} x_i + p_{i,1} \cos(\theta_{i,1}) + (p_{i,2} + r) \cos(\theta_{i,1} + \theta_{i,2}) \\ y_i + p_{i,1} \sin(\theta_{i,1}) + (p_{i,2} + r) \sin(\theta_{i,1} + \theta_{i,2}) \\ \theta_{i,1} + \theta_{i,2} + \pi \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \\ \phi_e + \delta_i \end{bmatrix}, \quad (3.10)$$

where r , δ_i and $\theta_{i,1}$ with $i = 1, 2, 3, 4$ presented in Figure 3.2(b) are constants.

In the 4-PRP mechanism, there exist fifteen variables, that is x_e , y_e , ϕ_e , $\theta_{i,2}$, $p_{i,1}$ and $p_{i,2}$ (with $i = 1, 2, 3, 4$). Equation (3.10) has eight equations since the 4-PRP mechanism has four serial chains and for each serial chain there are two equations. When considering the forward-kinematics problem, similarly, eight unknown variables including x_e , y_e , ϕ_e , $\theta_{i,2}$ (with $i = 1, 2, 3, 4$) and $p_{4,2}$ are defined with displacements of other prismatic actuators being known quantities to obtain a solvable system. Inverse-kinematics solution of the 4-PRP mechanism is quite straightforward with expressions depicted as [Williams II & Shelley, 1997]:

$$\theta_{i,2} = \phi_e + \delta_i - \pi - \theta_{i,1}, \quad (3.11)$$

$$p_{i,1} = \frac{(x_e - x_i) \sin(\theta_{i,1} + \theta_{i,2}) - (y_e - y_i) \cos(\theta_{i,1} + \theta_{i,2})}{\sin(\theta_{i,2})} \quad (3.12)$$

and

$$p_{i,2} = \frac{(y_e - y_i) \cos(\theta_{i,1}) - (x_e - x_i) \sin(\theta_{i,1})}{\sin(\theta_{i,2})} - r \quad (3.13)$$

for $i = 1, 2, 3, 4$.

By taking the time derivatives of Equation (3.10), the differential kinematic equation is derived as

$$A\dot{\chi} = B\dot{\vartheta} \quad (3.14)$$

where A and $\dot{\chi}$ have same expressions as presented in Equation (3.6) and

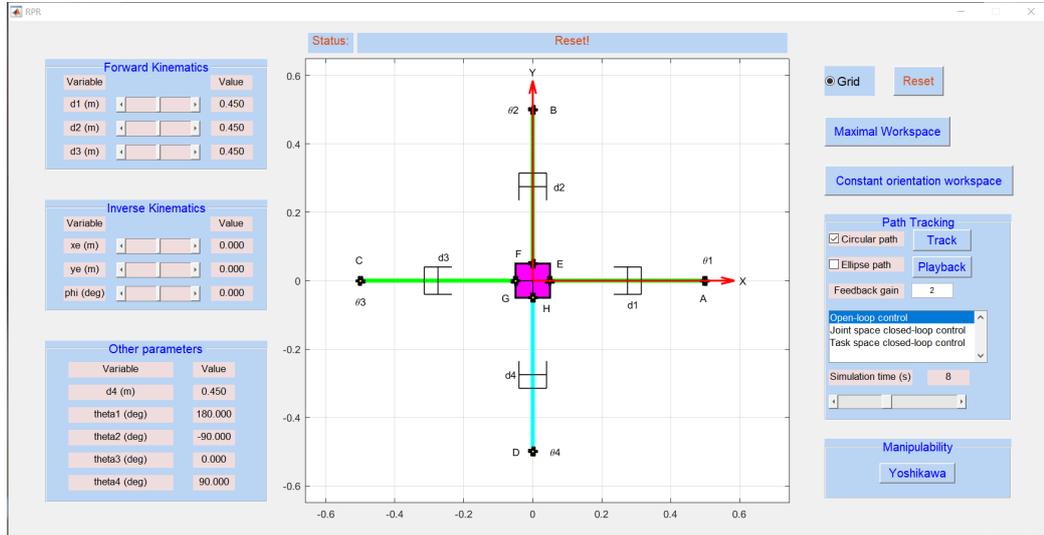
$$B = \begin{bmatrix} -p_{1,2}s_{1,2} & 0 & 0 & 0 & c_1 & 0 & 0 & 0 & c_{1,2} & 0 & 0 & 0 \\ 0 & -p_{2,2}s_{2,2} & 0 & 0 & 0 & c_2 & 0 & 0 & 0 & c_{2,2} & 0 & 0 \\ 0 & 0 & -p_{3,2}s_{3,2} & 0 & 0 & 0 & c_3 & 0 & 0 & 0 & c_{3,2} & 0 \\ 0 & 0 & 0 & -p_{4,2}s_{4,2} & 0 & 0 & 0 & c_4 & 0 & 0 & 0 & c_{4,2} \\ p_{1,2}c_{1,2} & 0 & 0 & 0 & s_1 & 0 & 0 & 0 & s_{1,2} & 0 & 0 & 0 \\ 0 & p_{2,2}c_{2,2} & 0 & 0 & 0 & s_2 & 0 & 0 & 0 & s_{2,2} & 0 & 0 \\ 0 & 0 & p_{3,2}c_{3,2} & 0 & 0 & 0 & s_3 & 0 & 0 & 0 & s_{3,2} & 0 \\ 0 & 0 & 0 & p_{4,2}c_{4,2} & 0 & 0 & 0 & s_4 & 0 & 0 & 0 & s_{4,2} \end{bmatrix}$$

and $\dot{\vartheta} = [\dot{\theta}_{1,2}, \dot{\theta}_{2,2}, \dot{\theta}_{3,2}, \dot{\theta}_{4,2}, \dot{p}_{1,1}, \dot{p}_{2,1}, \dot{p}_{3,1}, \dot{p}_{4,1}, \dot{p}_{1,2}, \dot{p}_{2,2}, \dot{p}_{3,2}, \dot{p}_{4,2}]^T$ with $s_i = \sin(\theta_{i,1})$, $c_i = \cos(\theta_{i,1})$, $s_{i,2} = \sin(\theta_{i,1} + \theta_{i,2})$ and $c_{i,2} = \cos(\theta_{i,1} + \theta_{i,2})$ for $i = 1, 2, 3, 4$. In this case, it is readily to have Jacobian matrix $J = A^\dagger B$ such that $\dot{\chi} = J\dot{\vartheta}$, Yoshikawa manipulability index $w = \sqrt{|JJ^T|}$ and three control laws as presented in Equations (3.7)–(3.9). From the above derivations, it can be observed that the differential kinematics of the two mechanisms are similar. Since they all have 3 DOFs, their end-effector platform can be controlled to follow a given path on a plane using control laws (3.7)–(3.9) as the pose of the end-effector platform is depicted as three variables (i.e., x_e, y_e and ϕ_e).

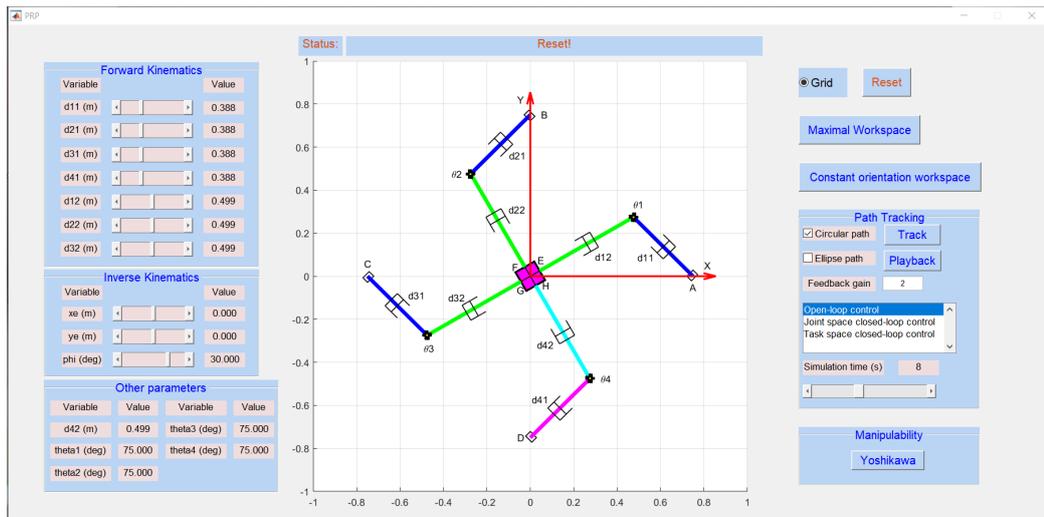
3.3 Simulation and Analysis

To compare the different performance of the two parallel mechanisms, two Matlab graphical user interfaces (GUIs) are created for the two parallel mechanisms as presented in Figure 3.3. The two GUIs cover the workspace, manipulability, forward kinematics, inverse kinematics and path tracking of the two parallel mechanisms. Besides, simulation time and feedback gain for path tracking can be set using the GUIs. Note that this section focuses on demonstrating the mobility of the two parallel mechanisms and locomotion capabilities of modular robotic structures formed by using the two connectivities, therefore finding an accurate analytical forward-kinematics solution is beyond the scope of this thesis.

3.3 Simulation and Analysis



(a)



(b)

Figure 3.3: Matlab GUIs Created for the Two Parallel Mechanisms. (a) GUI for the 4-RPR Parallel Mechanism. (b) GUI for the 4-PRP Parallel Mechanism.

3.3.1 Workspace and Manipulability

As mentioned in the preceding section, the pose of the end-effector platform is defined by three variables: x_e , y_e and ϕ_e . For a given end-effector pose, the unknown variables of the two parallel mechanisms can be determined by solving the inverse-kinematics

3.3 Simulation and Analysis

problem using Equations (3.4), (3.5) and (3.11)–(3.13). For a parallel mechanism, if a given end-effector pose leads to effective joint variables that are within their joint limits, the end-effector pose is within the workspace of the mechanism. A maximal workspace is composed of all the possible effective end-effector poses while a constant orientation workspace is a subset of the maximal workspace with the end-effector orientation ϕ_e being a constant. To obtain the maximal workspace, different combinations of x_e , y_e and ϕ_e are evaluated. The values of x_e and y_e are obtained from an interval $[-0.3, 0.3]$ in a step of 0.05 and the value of ϕ_e is attained from an interval $[-60^\circ, 60^\circ]$ in a step of 1° . For the constant orientation workspace, different values of ϕ_e are considered.

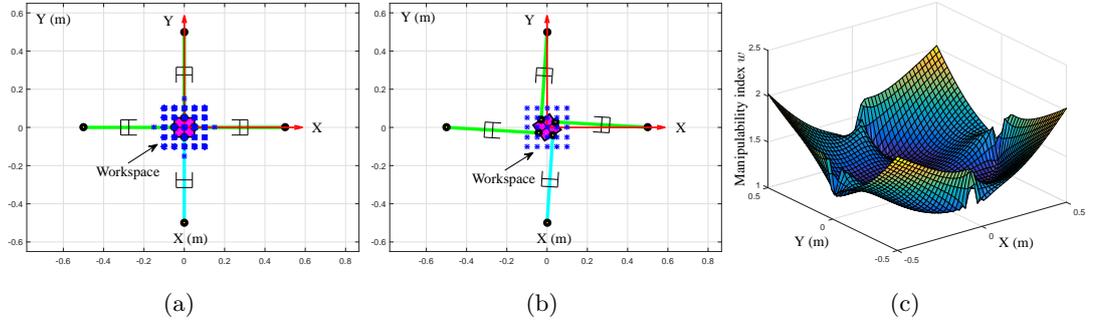


Figure 3.4: Workspace and Manipulability of the 4-RPR Mechanism. (a) Maximal Workspace. (b) Constant Orientation Workspace with $\phi_e = 35^\circ$. (c) Manipulability with $\phi_e = 35^\circ$.

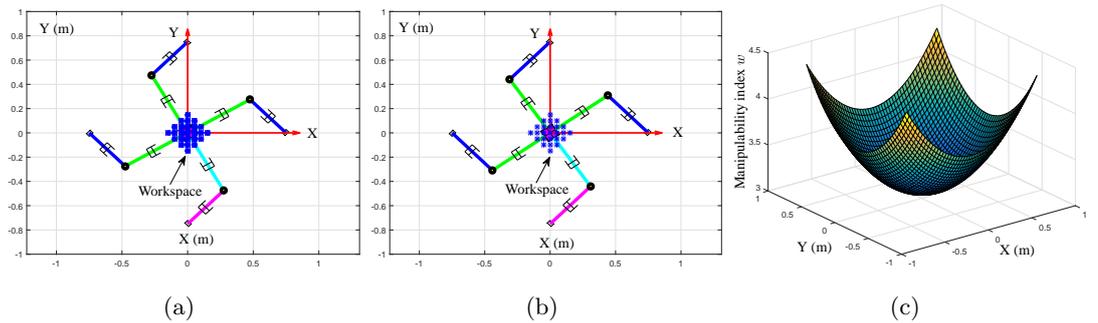


Figure 3.5: Workspace and Manipulability of the 4-PRP Mechanism. (a) Maximal Workspace. (b) Constant Orientation Workspace with $\phi_e = 35^\circ$. (c) Manipulability with $\phi_e = 35^\circ$.

Figures 3.4 and 3.5 show the workspaces and manipulability of the 4-RPR and 4-PRP mechanisms, respectively. As can be observed from the two figures, the maximal workspaces and constant orientation workspaces with $\phi_e = 35^\circ$ of the two mechanisms have almost the same size. The manipulability index w of the 4-PRP mechanism is larger than that of the 4-RPR mechanism when $\phi_e = 35^\circ$, meaning that the 4-RPR mechanism is closer to its singular configurations. For further comparing the two mechanisms, workspaces with different end-effector orientations $\phi_e = 25^\circ, 30^\circ, 40^\circ$ and 45° are studied. Figure 3.6 shows the constant orientation workspace of the 4-RPR mechanism with $\phi_e = 25^\circ$, which is the same as the one presented in Figure 3.4(b). Other constant orientation workspaces of the 4-RPR mechanism with $\phi_e = 30^\circ, 40^\circ$ and 45° are similar to the result shown in Figure 3.6, thus the other results are omitted here. This means that the constant orientation workspace of the 4-RPR mechanism is inde-

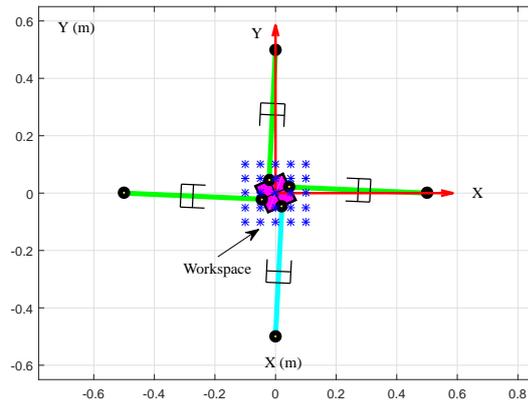


Figure 3.6: Constant Orientation Workspaces of the 4-PRP Mechanism with $\phi_e = 25^\circ$.

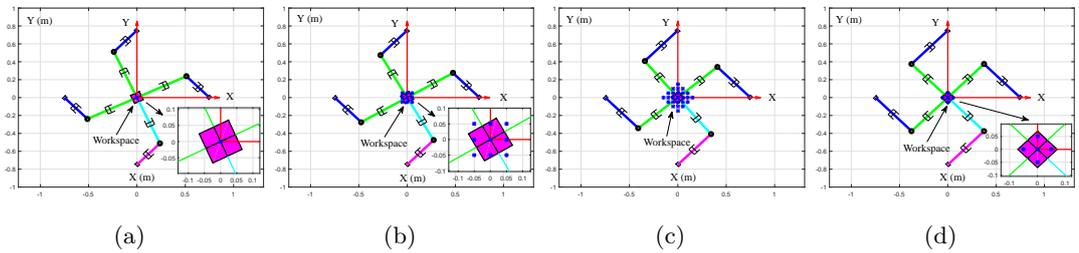


Figure 3.7: Constant Orientation Workspaces of the 4-PRP Mechanism with Different Values of ϕ_e . (a) With $\phi_e = 25^\circ$. (b) With $\phi_e = 30^\circ$. (c) With $\phi_e = 40^\circ$. (d) With $\phi_e = 45^\circ$.

pendent on the values of ϕ_e in the investigated four cases. In contrast, the workspaces of the 4-PRP mechanism with different constant orientations are presented in Figure 3.7, showing that the 4-PRP mechanism has different workspaces corresponding to different values of ϕ_e . When $\phi_e = 25^\circ$, the mechanism can be treated as a static structure as the workspace converges to a single point. When $\phi_e \in (30^\circ, 45^\circ)$ especially when ϕ_e is close to 35° , the workspace is becoming larger. In short, for a specific configuration with $\phi_e = 35^\circ$, workspaces of the two mechanisms have almost the same size and the 4-PRP mechanism wins in terms of manipulability. The 4-PRP mechanism's workspace is closely dependent on the values of end-effector orientation ϕ_e while the 4-RPR mechanism has no such a drawback.

Finally, the two mechanisms are assigned to follow a circular path to show their mobility. The configurations of the two parallel mechanisms are selected under the prerequisite that the parallel mechanisms would not possess singularities during the whole task execution process. This means that the solution of J^\dagger always exists. Control laws (3.7)–(3.9) are applied to the two parallel mechanisms. The feedback gain κ for closed-loop control in each case is simply determined by manual and empirical method to finish the path-tracking task as satisfactory as possible. Simulation results are presented in Figures 3.8 and 3.9 demonstrating that the PRP mechanism has superior performance in terms of accomplishing such tasks. Specifically, when using control laws (3.7) and (3.9), the actual trajectory considerably deviates from the desired path in the 4-RPR mechanism case. Clearer evidences of the deviations can be observed in Figure 3.8(d) and (f) showing the difference e of desired path and the actual trajectory, where e_X and e_Y are the X- and Y-components of position error e , respectively. When control law (3.8) with $\kappa = 130$ is employed, the 4-RPR parallel mechanism can follow the given path quite well with the position error bounded between -4×10^{-4} m and 4×10^{-4} m. As for the 4-PRP mechanism, the actual trajectory and desired path coincide well with each other, which is independent on the employed control laws. Evidently, the position errors synthesized by the 4-PRP mechanism in all the cases are much smaller than those of the 4-RPR mechanism. The feedback gain $\kappa = 130$ in control law (3.8) for the 4-RPR mechanism is selected by testing values ranging from 0.5 to 200. When $\kappa = 120$ or $\kappa = 140$, position error would be slightly larger than the one with $\kappa = 130$ used. For other closed-loop control cases, the 4-RPR and 4-PRP parallel mechanisms would fail to finish the task if κ is too large. Hence, parameter κ in these cases is set

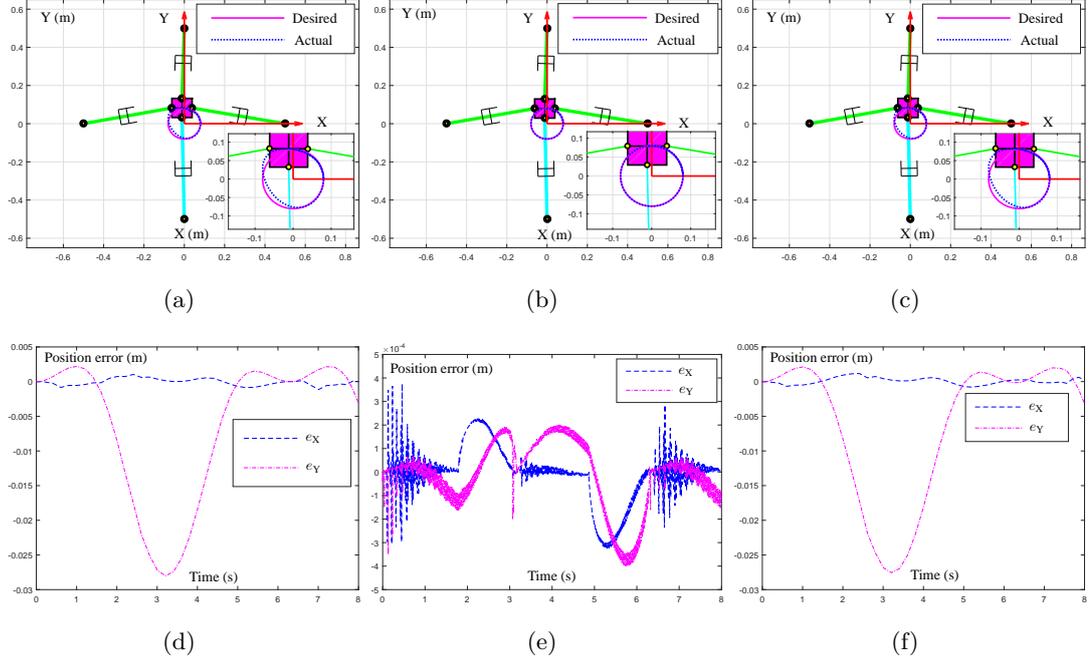


Figure 3.8: Circular Path Tracking using the 4-RPR Mechanism. (a) With Control Law (3.7): $\dot{\vartheta} = J^\dagger \dot{\chi}$. (b) With Control Law (3.8): $\dot{\vartheta} = J^\dagger \dot{\chi} + \kappa(\vartheta_d - \vartheta_a)$ ($\kappa = 130$). (c) With Control Law (3.9): $\dot{\vartheta} = J^\dagger(\dot{\chi} + \kappa(\chi_d - \chi_a))$ ($\kappa = 0.001$). (d) Position Error Corresponding to Control Law (3.7). (e) Position Error Corresponding to Control Law (3.8). (f) Position Error Corresponding to Control Law (3.9).

quite small to ensure that the task can be completed.

According to the above comparisons, each of the two mechanisms has its own advantages and drawbacks. The 4-RPR mechanism outperforms the 4-PRP mechanism in terms of workspace as the workspace of the 4-PRP mechanism is dependent on the orientation of the end-effector platform. Nevertheless, the 4-PRP mechanism can have superior performance for path tracking tasks under the investigated conditions. Imagine a huge structure that is formed by micro-scale 4-RPR or 4-PRP cells, the whole structure may exhibit a large deformation via the slight movement and limited deformation of each cell. One example of such shape-morphing compliant structures has been investigated in [Shaw & Hopkins, 2015]. When considering the fabrication of such structures, using the PRP mechanism would be more advisable as separated rigid connector nodes and passive revolute joints are more implementable especially in terms

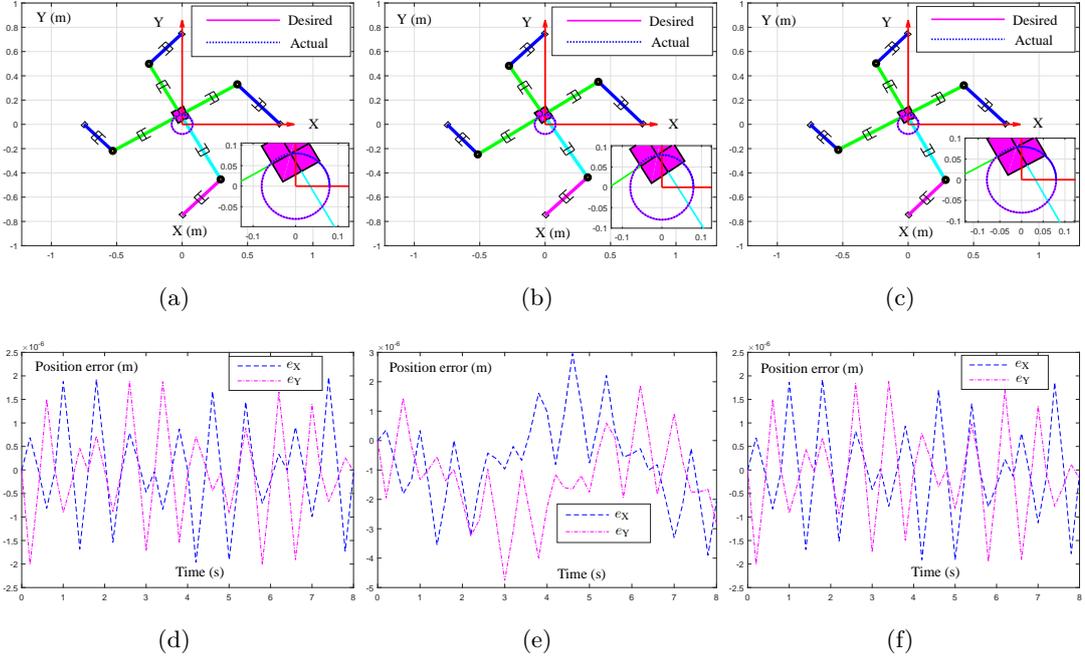


Figure 3.9: Circular Path Tracking using the 4-PRP Mechanism. (a) With Control Law (3.7): $\dot{\vartheta} = J^\dagger \dot{\chi}$. (b) With Control Law (3.8): $\dot{\vartheta} = J^\dagger \dot{\chi} + \kappa(\vartheta_d - \vartheta_a)$ ($\kappa = 0.0001$). (c) With Control Law (3.9): $\dot{\vartheta} = J^\dagger(\dot{\chi} + \kappa(\chi_d - \chi_a))$ ($\kappa = 0.0001$). (d) Position Error Corresponding to Control Law (3.7). (e) Position Error Corresponding to Control Law (3.8). (f) Position Error Corresponding to Control Law (3.9).

of miniaturization for future applications.

3.3.2 Robot Locomotion and Deformation

This section compares the deformation and locomotion capabilities of MRSs constructed using different connectivities. Inspired by [Yu, 2010], two stacked prismatic actuators pointing in opposite directions can form a robotic strut to achieve a worm-like locomotion by changing the mass centre of the whole robotic strut. Specifically, the employed moving principle is illustrated in Figure 3.10. The following three moving steps are cyclically followed to let the robotic strut move rightward:

- 1) Extend prismatic actuator 1 and keep prismatic actuator 2 still;
- 2) Retract prismatic actuator 1 and extend prismatic actuator 2 simultaneously;

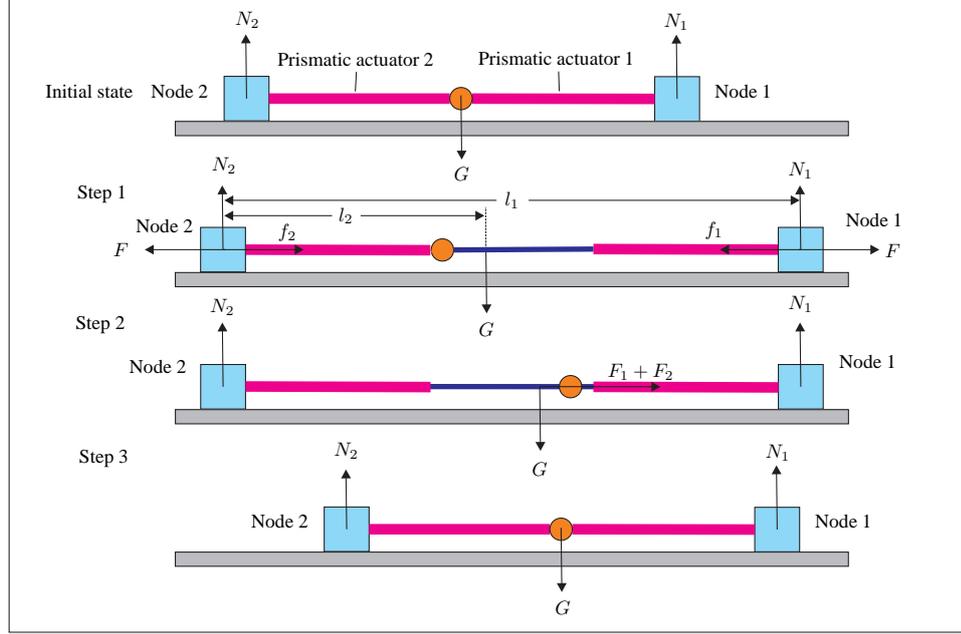


Figure 3.10: Diagram of the Employed Moving Principle for Moving Rightward.

- 3) Retract prismatic actuator 2 and keep prismatic actuator 1 still.

For analysis, the mass centre of a prismatic actuator is supposed to be located at its jointed node and the frictions between the two nodes and the floor are assumed to be the same. In the initial fully contracted stage, the normal forces N_1 and N_2 as well as the gravity force G of the robotic strut satisfy $N_1 + N_2 = G$ and $N_1 = N_2 = G/2$. During Step 1, let μ denote the friction coefficient, when extending prismatic actuator 1, at the first few seconds, friction forces $f_1 = \mu N_1$ and $f_2 = \mu N_2$ would be less than the actuation force F . Hence, Node 2 would move leftward for a short time while Node 1 will move rightward. Since the mass centre of the whole strut moves rightward and $N_1 + N_2 = G$ together with $N_1 \times l_1 = G \times l_2$, N_1 and f_1 decreases while N_2 and f_2 increases. Hence, Node 2 would keep still and Node 1 keeps moving rightward. During Step 2, due to the collective work of actuation forces F_1 and F_2 , the inner shafts of the two prismatic actuators move rightward owing to the resultant force $F_1 + F_2$. The motion analysis of Step 3 is similar to Step 1, during Step 3, Node 2 would keep moving rightward while Node 1 would first keep still and then move leftward for a while.

On the basis of the discussed moving principle, modular robotic structures are

established in a trial version of a physics-based robot simulator named Webots which can automatically consider kinematics and dynamics of a simulated robotic structure using the open dynamics engine. A conventional square robot with compliant nodes and a square robot using rigid nodes are presented in Figure 3.11(a) and (b), respectively. As shown in Figure 3.11(a), each strut of the conventional robot has two prismatic actuators pointing in opposite directions and connects with two compliant nodes with revolute joints. In contrast, each strut of the robot illustrated in Figure 3.11(b) consists of two prismatic actuators jointed by a revolute joint and rigidly connects with two nodes. Hereafter, the struts in Figure 3.11(a) and (b) are termed as RPPR and PRP struts with R and P representing revolute and prismatic, respectively. To achieve a locomotion gait, two parallel struts are actuated using the aforementioned moving principle. In this way, the square robot can move along a certain direction resorting to the frictions between the nodes and the floor. Simulations are performed with corresponding results presented in Figures 3.12 and 3.13 to demonstrate the efficacy of the moving principle. As can be seen from the figures, both of the two square robots move rightward in a worm-like way. The RPPR square robot moves further as compared with the PRP square robot within a single cycle. This results from a fact that the two square robots exhibit different deformations during the locomotion process. The RPPR square robot deforms between rectangle and square shapes and

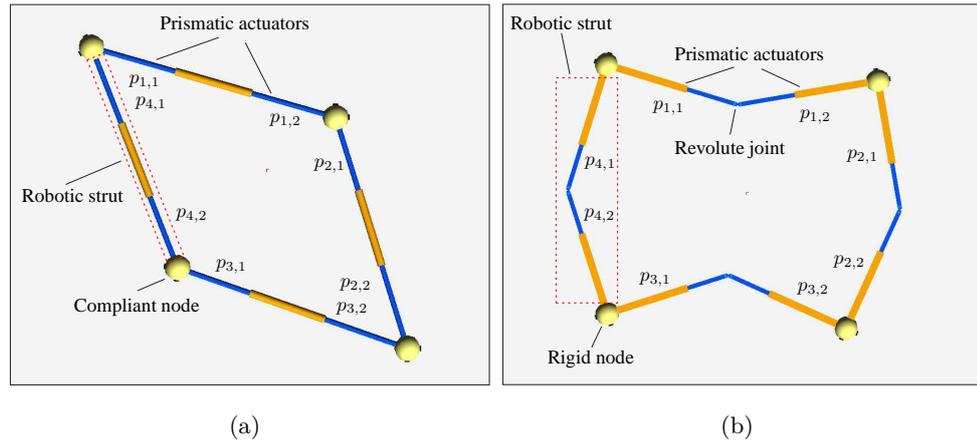


Figure 3.11: Square Robots with Different Connectivities. (a) Square Robot with RPPR Struts. (b) Square Robot with PRP Struts.

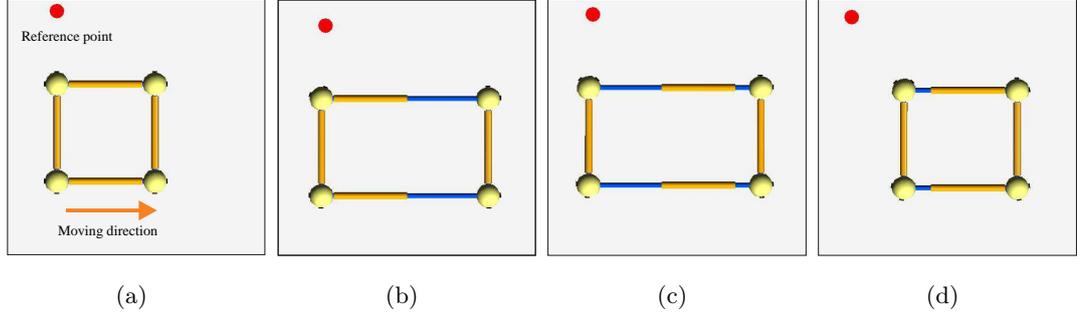


Figure 3.12: Locomotion of a Square Robot with Compliant Nodes. (a) Initial State. (b) Step 1. (c) Step 2. (d) Step 3.

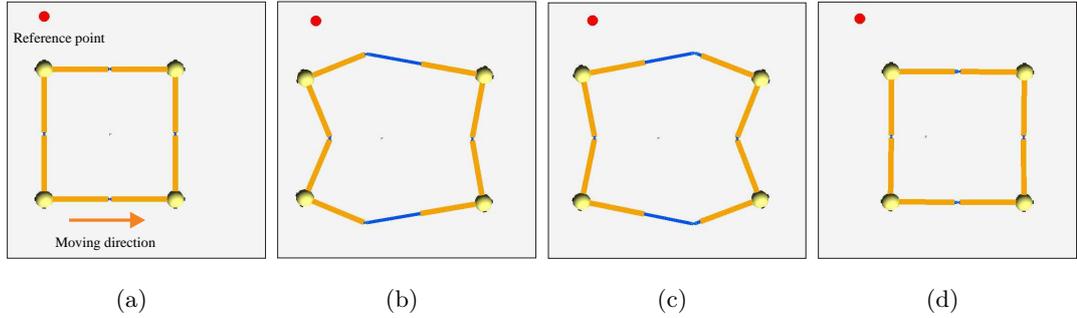


Figure 3.13: Locomotion of a Square Robot with Rigid Nodes. (a) Initial State. (b) Step 1. (c) Step 2. (d) Step 3.

the active struts are always in parallel with the moving direction. The PRP square robot, however, experiences deformed buckling shapes shown in Figure 3.13(b) and (c), leading to the active struts cannot be in parallel with the moving direction.

For further investigation of these two square robots, each square robot is considered as a meta-module and various MRSs can be constructed by rigidly connecting such square meta-modules. Figures 3.14(a) and 3.15(a) show two MRSs consisting of two square meta-modules. Each meta-module can be treated as an agent that can communicate with its neighbouring meta-modules. Starting from an initial state, by coordinating the movements of the square meta-modules, the whole robotic structure can finally moves along a certain direction. The captured snapshots during the locomotion of the two robots are shown in Figures 3.14(b)–(h) and 3.15(b)–(h). Both of the two robotic structures can coordinate the movements of their square meta-modules and fi-

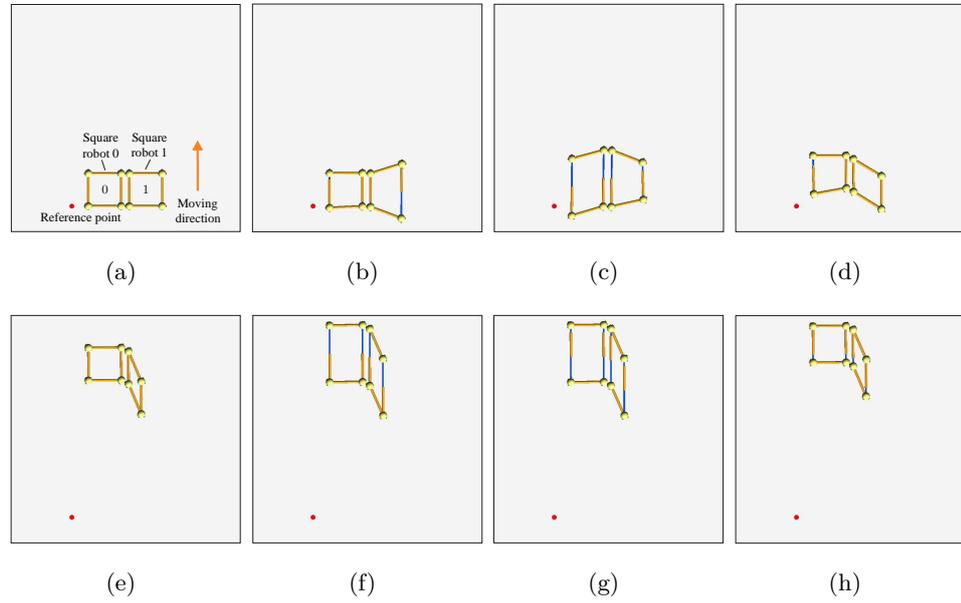


Figure 3.14: Locomotion of a Two-Square Robot with Compliant Nodes. (a) Structure Labels. (b)–(d) Snapshots Captured During Coordination Stage. (e)–(h) Snapshots Captured During Coordinated Locomotion Stage.

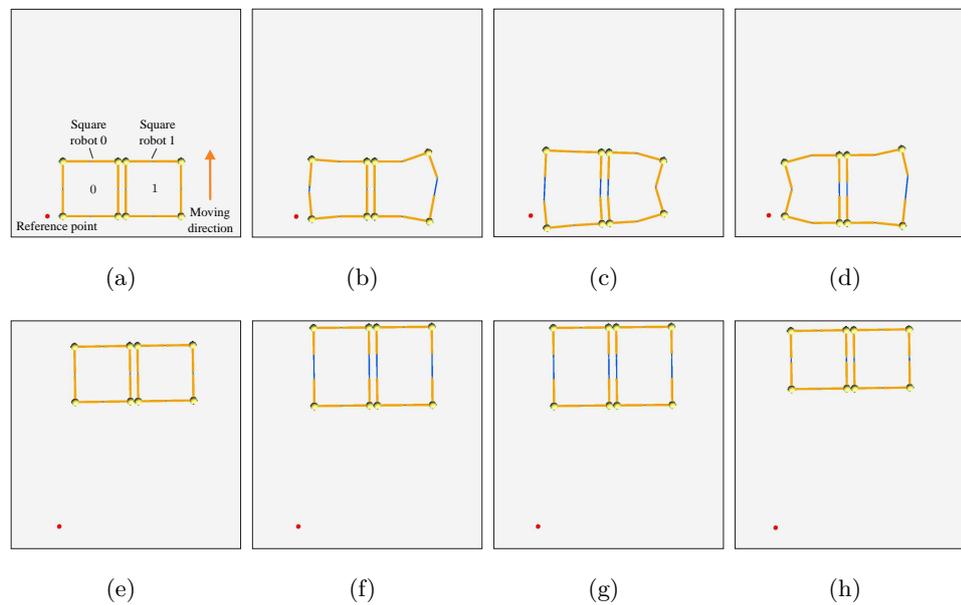


Figure 3.15: Locomotion of a Two-Square Robot with Rigid Nodes. (a) Structure Labels. (b)–(d) Snapshots Captured During Coordination Stage. (e)–(h) Snapshots Captured During Coordinated Locomotion Stage.

nally move along a user-defined direction. Although the RPPR robot still moves faster than the PRP robot, one of the square meta-module maintains a deformed rhombus shape shown in Figure 3.14(d)–(h) when achieving a coordinating locomotion. Unlike the RPPR robot, the square meta-modules of the PRP robot can restore and deform between the rectangle and square shapes after some initial deformations. This is important as undesirable collisions between modules would be more common in a robotic structure with a deformed rhombus shape.

Finally, deformation capabilities of robotic structures composed of RPPR and PRP struts are compared. The two different robotic structures are presented in Figures 3.16(a) and 3.17(a). By actuating struts placed along the top and bottom segments of the perimeter and letting other struts be passive, deformation results are obtained and shown in Figures 3.16 and 3.17. Evidently, both of the two robotic structures are capable of deforming and they exhibit different deformations. Such deformations can be applied to physical display.

In short, modular robotic structures formed by the two connectivities are capable

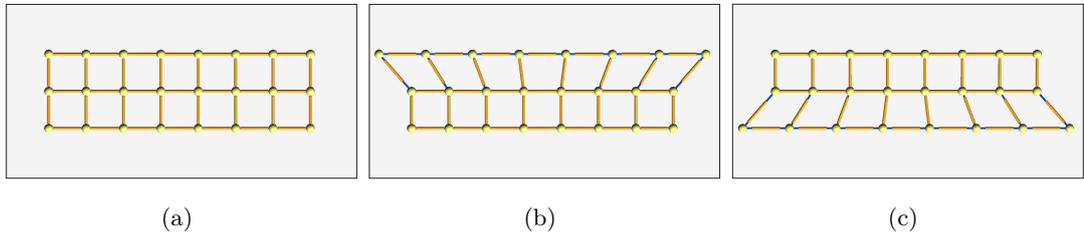


Figure 3.16: Deformation of a Fourteen-Square Robot with Compliant Nodes. (a) Initial State. (b) Snapshot 1. (c) Snapshot 2.

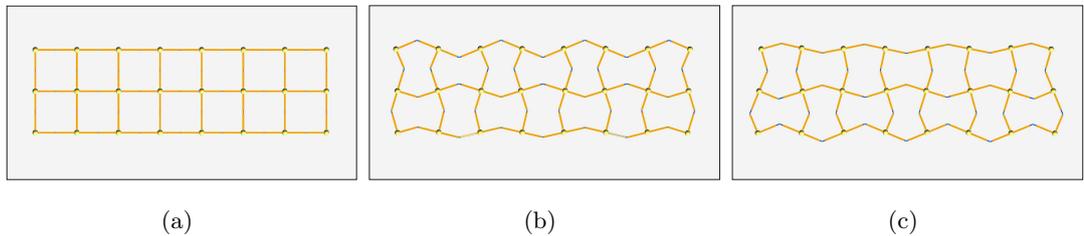


Figure 3.17: Deformation of a Fourteen-Square Robot with Rigid Nodes. (a) Initial State. (b) Snapshot 1. (c) Snapshot 2.

of achieving a worm-like locomotion gait. For a single-module robot, the RPPR robot outperforms the PRP robot as the latter would exhibit deformations that influence its locomotion speed. For robotic structures with multiple squares, the RPPR robot is more susceptible to have undesirable self-collisions due to deformations.

3.4 Conclusions

This chapter investigated and compared two different connectivities for constructing strut-type MRSs. Kinematics analysis of two basic 4-RPR and 4-PRP parallel mechanisms has been studied to show their differences in terms of mobility, workspace and manipulability. Both of the two parallel mechanisms possess 3 DOFs allowing them to track a given path in the 2D plane. The workspace of the 4-PRP mechanism is dependent on the orientation of the end-effector platform while the 4-RPR mechanism has no such a shortcoming. Under a same constant orientation of $\phi_e = 35^\circ$, the 4-PRP mechanism is advantageous over the 4-RPR mechanism in terms of manipulability. The 4-PRP mechanism can also outperform the 4-RPR mechanism when the two parallel mechanisms are applied to tracking a same circular path provided that the given path is within the workspace of the 4-PRP mechanism with a certain constant orientation. After that, locomotion and deformation capabilities of two kinds of MRSs were investigated with comparative simulation results presented. Both of the two types of structures can deform and achieve a worm-like locomotion. For a single square-meta module, the PRP robot tends to exhibit a buckling shape leading to a fact that the two active struts are not in parallel with the moving direction. Hence, the PRP robot moves slower than the RPPR robot. For robots with two square-meta modules, the RPPR robot is more susceptible to experience collisions between modules as the robot would maintain an undesired rhombus shape during locomotion. The PRP robots have potential advantages to realize miniaturization as the rigid nodes and revolute joints can be miniaturized separately. This is significant and crucial to fabricating large-scale MRSs with micro-scale modules for future applications.

CHAPTER 4

Mechatronics Designs and Physical Prototypes

This chapter first conducts a design analysis of strut-type modular robots and then details the mechatronics designs, physical prototypes and implementation considerations of the modular robotic elements.

4.1 Design Analysis

In general, a strut-type modular robot consists of robotic struts and connector nodes. Ideally, the struts should be connected using point-like nodes with passive compliance and struts linked by a same node can rotate around a common centre of rotation. However, such compliant ideal nodes are difficult and even impossible to implement in practice [Lyder, 2010]. This section analyses typical designs of a connector node and a robotic strut. For achieving multiple working states, actuation and locking mechanisms are discussed.

4.1.1 Connector Node

Existing solutions of nodes can be summarized as follows.

- Solution 1: the CMS node shown in Figure 4.1(a) [Hamlin & Sanderson, 1998]. This is a solution to the implementation of an ideal node that can connect numerous struts and the struts jointed by a same node rotate around and intersect at the node centre. Ideal nodes like CMS nodes are beneficial to simplifying the kinematics of robotic structures. However, CMS nodes cannot be used to form chain-type structures and reconfiguration is difficult as removing struts usually requires disassembly of the whole robot [Lyder, 2010]. Additionally, CMS nodes tend to become too weak to sustain a massive robotic structure.
- Solution 2: rigid node with passive universal joint connectors used in TET Walker presented in Figure 4.1(b) [Curtis *et al.*, 2007a]. This is a workaround solution to an ideal node with passive compliance. However, struts connected by such a node have no common rotation centre which increases the kinematics complexity.
- Solution 3: rigid node with passive ball-and-socket joint connectors shown in Figure 4.1(c) [Lyder, 2010, Yu, 2010]. This is another workaround solution to an ideal node. Each strut connected by a node has its own centre of rotation on the node surface and can rotate passively around its rotation centre. Like Solution

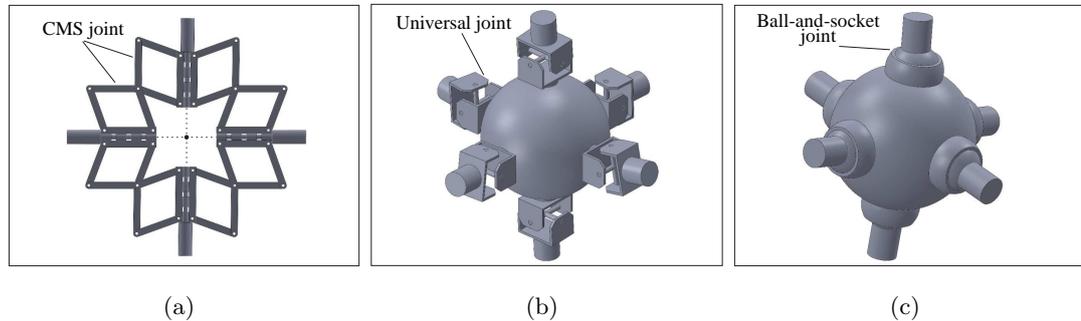


Figure 4.1: Different Solutions to Connector Nodes. (a) CMS Joint. (b) Node with Universal Joints. (c) Node with Ball-and-Socket Joint.

2, such an approach does simplify the physical implementation but makes the kinematics analysis more complex.

Considering the difficulty of the design and implementation of an ideal node and the shortcomings of existing workaround non-ideal nodes, this research focuses on using only rigid nodes to construct strut-type MRSs. Rigid nodes are extremely easy to design and manufacture and struts jointed by a rigid node can meet the requirement of the presence of a common intersection point. The design and implementation of a rigid connector node for this research is detailed in the upcoming Section 4.2.2.

4.1.2 Actuation Mechanism

As mentioned in [Hamlin & Sanderson, 1998], technically, almost any type of prismatic actuator can be used for a strut. Hydraulic and pneumatic actuators exhibit a high strength to weight ratio but they are tricky to work with. Electric prismatic actuators are simpler to use and can be controlled straightforwardly. Considering this fact, the solution of electric prismatic actuator is selected and existing mechanical solutions to a prismatic actuation mechanism are listed as the following.

- Solution 1: lead screw actuator. This solution uses the lead screw transmission mechanism to translate rotary motions into linear motions. Most of the commercial prismatic actuators like the Fircelli actuator shown in Figure 4.2(a) are designed based on a lead screw. In the TETROBOT [Hamlin & Sanderson, 1998] and Amorphous robot [Yu *et al.*, 2010] projects, commercial prismatic actuators

were employed as it is relatively easier to select appropriate prismatic actuators that meet some specific requirements from the market.

- Solution 2: cable-driven antenna actuator. This solution is simple, low-cost, lightweight and easy to achieve a high strain value of 5 but weak to support a large-scale structure. The strut of OctaWorm robot [Zagal *et al.*, 2012] adopts the same principle of a power car antenna shown in Figure 4.2(b), where the prismatic actuator can contract or extend by rotating a drum loaded with a plastic cable.
- Solution 3: telescoping actuator. This solution can be used for achieving a long stroke length. In the Odin robot [Lyder, 2010], a telescoping mechanism that is similar to a camera zoom lens shown in Figure 4.2(c) was designed. The

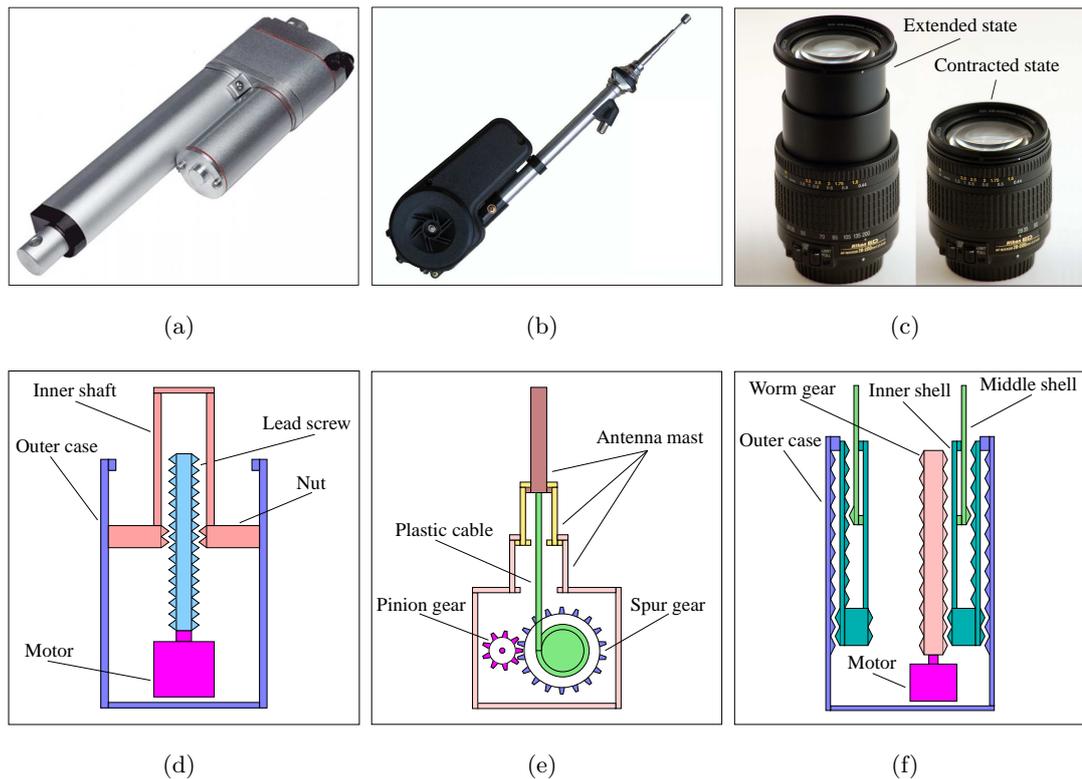


Figure 4.2: Different Prismatic Actuation Solutions. (a) Lead Screw Actuator. (b) Power Antenna Actuator. (c) Camera Zoom Lens. (d) Diagram of Lead Screw Actuator. (e) Diagram of Power Antenna Actuator. (f) Diagram of Camera Zoom Lens.

telescoping body of an Odin robot module has a strain of 1.2. Struts designed for the 12-TET Walker robot [Curtis *et al.*, 2007a] can have a strain of 4.29 due to a telescoping lead screw set.

For better understanding, the actuation mechanisms of the lead screw actuator, the power car antenna and the camera zoom lens are respectively illustrated in Figure 4.2(d)–(f). Since rigid nodes are utilized to construct strut-type modular robot, when forming networked structures, more kinematic constraints are introduced as compared with the cases of moving rotation centre of each strut to another point. These kinematic constraints are restrictions on the motions of struts, which result in difficulties to control the struts for achieving deformation without breaking the connection of modular elements. Passive lockable joints are capable of being locked to work as rigid elements or passive to move freely due to external forces, which can be used to release some of the kinematic constraints and reduce control complexity [Christoforou *et al.*, 2015, Grosch *et al.*, 2010]. A prismatic actuator that can work either in actuated, locked or passive state offers significant advantages or complex multi-linked systems. Then, a passive revolute joint can be used to link two three-state prismatic actuators to form a robotic strut. The passive revolute joint could be a one-DOF hinge joint, a two-DOF universal joint and a three-DOF spherical joint. As a starting point, this research only investigates the one-DOF hinge joint.

A lead screw with a large lead (i.e., the linear distance the nut translates for each one screw revolution) can back drive and the back-drive capability can be used to realize the passive state. The actuation mechanisms of a power antenna and a camera zoom lens are much more expensive and complex than that of a lead screw mechanism, not to mention making modifications to achieve locked and passive states. Hence, Solution 1 is selected and modified to implement a prismatic actuator.

4.1.3 Locking Mechanism

Both of the electromagnetic brake and clutch can work as a locking mechanism [Grosch *et al.*, 2010, Li *et al.*, 2007]. An electromagnetic brake connects a rotating shaft and a stationary body. When the electromagnetic brake is engaged, it brings the rotating shaft to a halt by applying frictional resistance to the moving body. An electromagnetic clutch can be used to transmit motions of one shaft to another shaft if required. The two shafts rotate independently if the electromagnetic clutch is released. When the

electromagnetic clutch is engaged, the two shafts rotate as a single element. A brake is relatively simpler, while a clutch is more versatile. In general, one common drawback of the electromagnetic brake and clutch is that they always require power consumption during the locked state. An economical locking mechanism that only consumes energy during state transitions between locked and unlocked states is desirable in this application. Non-backdrivable gear sets for example a worm gear set have self-locking property [Plooij *et al.*, 2015], meaning that no power supply is needed in the locked state. A lead screw coupled with a non-backdrivable gear set will have a strong actuation force to drive a linear system. However, it cannot back drive, which makes it hard to realize a passive state. Based on the above analysis, it is expected to design and implement an economical mechanical mechanism that works as the locker.

4.1.4 Design Requirements

From the discussed systems in the literature review, one can understand that the existing strut-type modular robots have their shortcomings. Simply put, the TETROBOT robot with ideal compliant nodes (i.e., the CMS joints) requires greater precision to fabricate the CMS joints to avoid unwanted sloppiness and play. Besides, chain-type structures cannot be formed by using CMS joints and reconfiguration of TETROBOT structures is difficult due to the requirement of disassembling the whole structure. The other strut-type modular robots such as the Odin and OctaWorm robots with non-ideal nodes complicate the modelling process and increase the kinematics and control complexity due to the fact that the robotic struts jointed by a same node cannot intersect at a same centre of rotation. The existing joints/actuators in the literature cannot achieve all of the expected actuated, locked and passive states. Considering these points, to fill the gap and overcome the limitations of current systems, the designed modular robotic system must meet the following requirements.

- 1) Each prismatic actuator must be capable of possessing the actuated, locked and passive states and the transitions between the three different working states must be physically achieved.
- 2) When actuated, a prismatic actuator must be capable of lifting at least another module vertically.
- 3) The prismatic-actuator module must have the ability to move passively due to

its own gravity.

- 4) Preferably, the locking mechanism only requires power during state transitions between locked and unlocked states. In the locked state, no power would be consumed and the locker must be capable of supporting another actuator vertically.
- 5) Each prismatic actuator should be self-contained with its own computation, communication, actuation and sensing capabilities. A prismatic actuator must have the ability to exchange information with other prismatic actuators and a low-cost solution to a communication network is preferable.
- 6) Each rigid connector node must be simple, low cost and can connect at least six struts simultaneously. Extended lines of all the actuators connected by a same rigid node must intersect at the centre of the node.
- 7) Each one-DOF passive revolute joint that is simple and low cost. The revolute joint must be easily moved passively.

More specifically, the modular robotic system to be developed must meet the detailed specifications listed in Table 4.1, where modular elements including the prismatic actuator, the rigid connector node and the passive revolute joint are covered.

4.2 Mechatronics Designs and Hardware Implementation

This section first presents the conceived conceptual designs. Then functional prototypes of a three-state prismatic actuator, a rigid connector node and a revolute joint are introduced and detailed.

4.2.1 Three-State Prismatic Actuator

This subsection presents and compares four potential designs of a three-state prismatic actuator capable of achieving the actuated, locked and passive states.

Design Scheme I

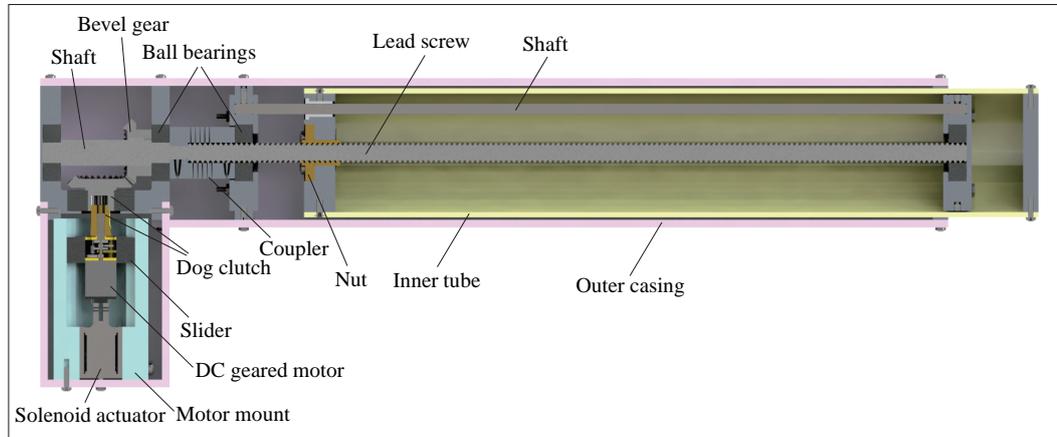
Figure 4.3(a) shows one conceptual design of a three-state prismatic actuator. Such an actuator mainly consists of a back-drivable lead screw, a DC geared motor, a solenoid actuator and a bevel gear set. The DC geared motor is non-backdrivable and has a

4.2 Mechatronics Designs and Hardware Implementation

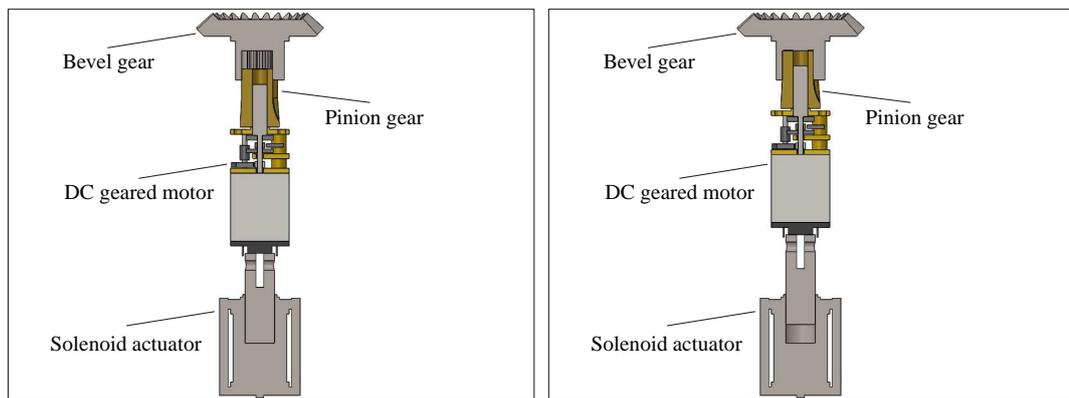
Table 4.1: Requirements and Specifications of the Strut-Type Modular Robotic System to Be Developed

Modular element	Property	
Prismatic actuator	Shape	Rectangular
	Maximum dimension	$100 \times 100 \times 500$ mm
	Weight	< 2.5 kg
	Main materials	Aluminium and plastics
	Actuation force	To lift 25 N
	Locking force	To sustain 25 N
	Passive compliance	To be moved by 10 N
	Transition time of locker	< 1 s
	Maximum actuation speed	≥ 500 mm/s
	Drive type	DC motor and lead screw
	Backdrivable	Yes
	Accuracy in actuated state	< 0.5 mm
	Microcontroller	Arduino
	Position sensor	Rotary encoder
	Communication strategy	RF
	Power source	9 V rechargeable batteries
Battery life per charge	≥ 30 minutes	
Other electronics components	Motor drivers and voltage regulator	
Rigid connector node	Shape	Cubic
	Maximum dimension	$110 \times 110 \times 110$ mm
	Weight	< 0.4 kg
	Main materials	Acrylic and brass
	Number of attachment faces	6
Passive revolute joint	Parts	Male and female parts
	Weight	< 0.2 kg
	Materials	Plastics
	Type	Hinge joint
	Number of attachment faces	2

4.2 Mechatronics Designs and Hardware Implementation



(a)



(b)

(c)

Figure 4.3: Section View of a Three-State Prismatic Actuator with a Dog Clutch. (a) Three-State Prismatic Actuator. (b) Dog Clutch in Released State. (c) Dog Clutch in Engaged State.

pinion gear that can be coupled or decoupled with the bevel gear set through a dog clutch. The engagement and disengagement of the dog clutch are controlled by the extension and contraction of the solenoid actuator as shown in Figure 4.3(b) and (c). When both of the solenoid actuator and the DC motor are engaged, the prismatic actuator is in the actuated state. In the case where only the solenoid actuator is active, the prismatic actuator is in the locked state owing to the non-backdrivable DC motor. When the solenoid actuator is released, a passive state of the prismatic actuator can be obtained due to the back-drive capability of the lead screw. This design gives a

possibility to develop a three-state prismatic actuator but it has an obvious drawback in terms of the difficulty of matching the dog clutch.

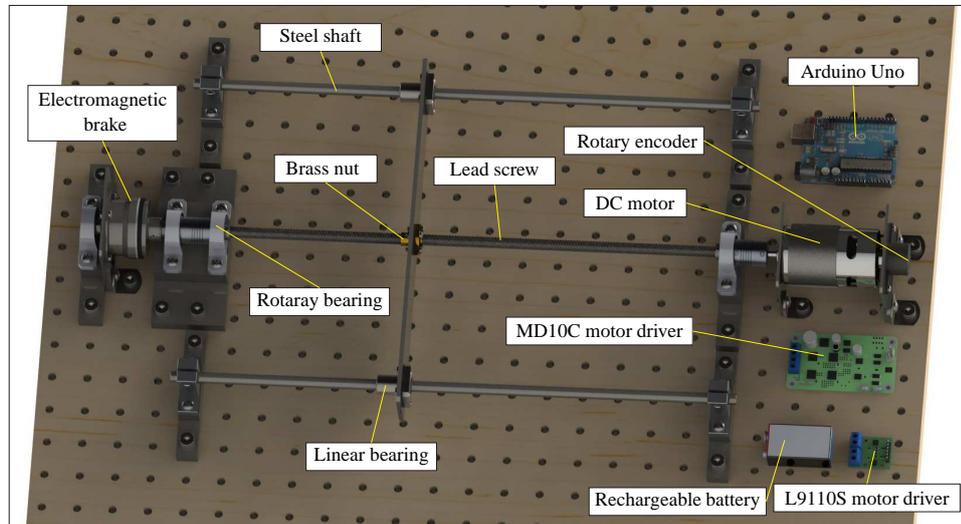
Design Scheme II

This subsection focuses on a relatively simple way of implementing a three-state prismatic actuator. The rendered CAD model is presented in Figure 4.4 from which it can be seen that the whole prismatic-actuator system mainly consists of a back-drivable lead screw, an electromagnetic brake, a DC motor, a rotary encoder, an Arduino Uno board and an MD10C and an L9110S motor drivers. Note that the lead screw must have a high efficiency which ensures its back-drive capability. This means that the lead screw can respond passively to external forces and thus a passive state can be obtained. The electromagnetic brake provides the prismatic-actuator system with a lockable ability indicating that the lead screw can be locked when the brake is engaged. In order for achieving three states, the DC motor is backdrivable as well and it should have low detent and high stall torques. Detent torque is the torque required to rotate the output shaft of a motor when the motor is unenergised, while stall torque is the maximum torque required to let a powered motor stop rotating. The low detent torque indicates one can easily rotate the output shaft of the motor while the high stall torque ensures the motor has enough power to actuate the prismatic-actuator system. The Arduino Uno board and the motor drivers play a role in controlling the DC motor and the electromagnetic brake to achieve three states and realize state transitions. Specifically, the DC motor and the electromagnetic brake are controlled by the MD10C and LS9110S motor drivers, respectively. When the motor is turned on and the brake is released, one can obtain the actuated state. In the locked state, the motor is turned off and the brake is engaged. When both the motor and the brake are inactive, the prismatic actuator is in its passive state.

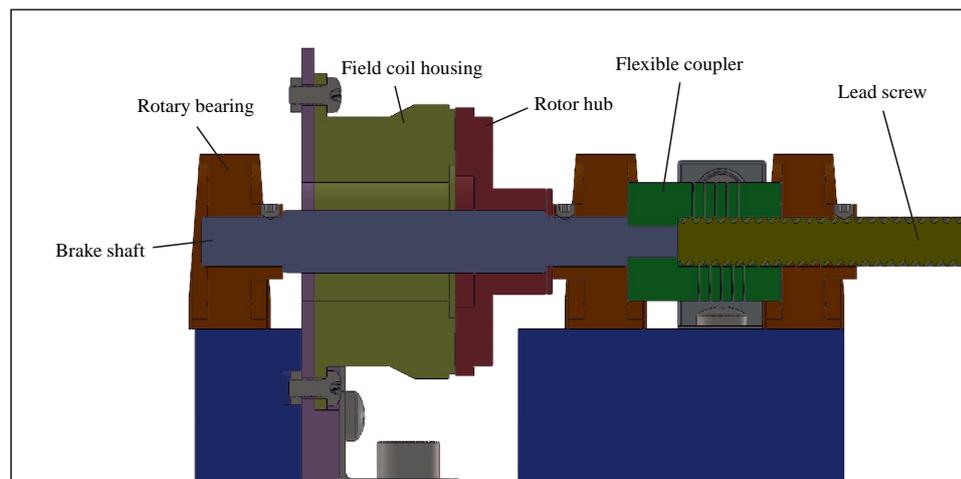
Design Scheme III

In Scheme II, the electromagnetic brake will consume energy all the time when the prismatic actuator is in its locked state. By understanding this point, the electromagnetic brake can be replaced with an economical mechanical locking mechanism. Figure 4.5(a) shows the designed mechanical locker. The locker is mainly composed of three slider arms, a plate with spiral grooves, a casing and a micro DC motor. The three

4.2 Mechatronics Designs and Hardware Implementation



(a)



(b)

Figure 4.4: Rendered CAD Models of a Three-State Prismatic Actuator with an Electromagnetic Brake. (a) Three-State Prismatic Actuator. (b) Section View of the Electromagnetic Brake.

slider arms have round pins mating with the spiral grooves and can move linearly along the tracks of the locker casing owing to the rotary motion of the spiral plate driven by the micro DC motor. For better understanding, Figure 4.5(b) presents the top view of the matched spiral plate and slider arms. The DC motor must have a high torque

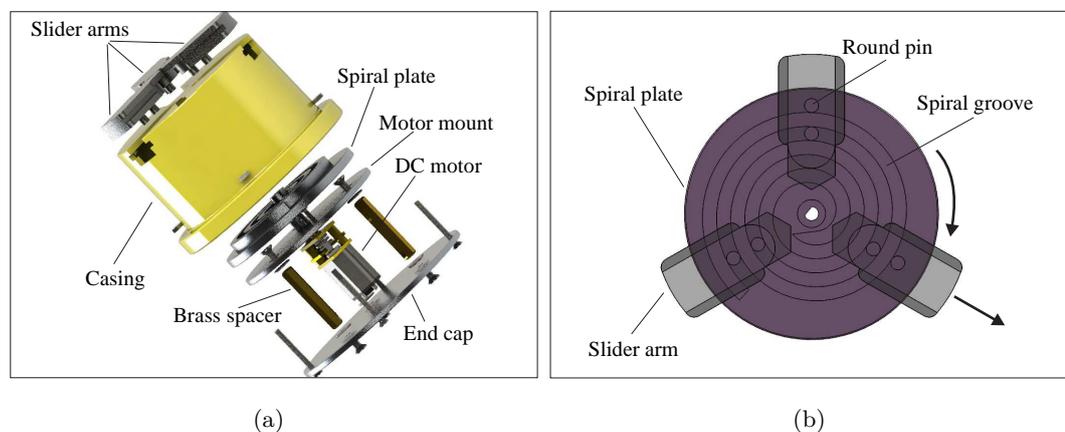


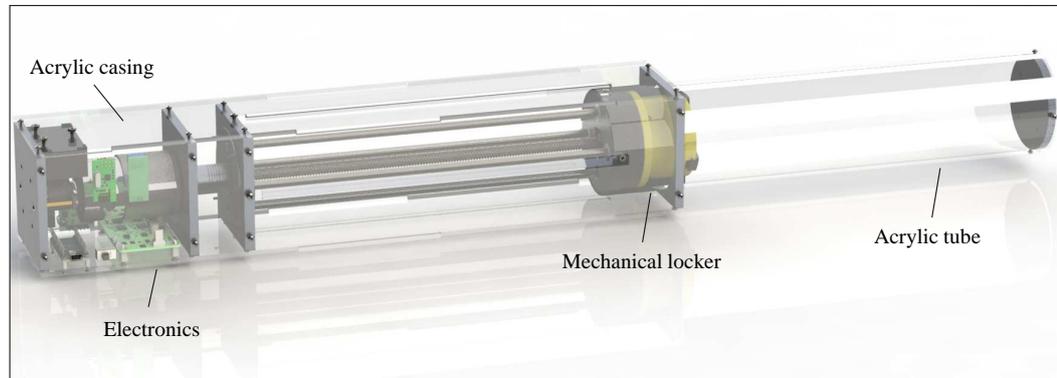
Figure 4.5: CAD Models of the Mechanical Locker. (a) Exploded View of the Mechanical Locker. (b) Top View of the Spiral Plate and Slider Arms.

providing strong force for contracting and extending the slider arms. This indicates the locker system would consume energy only during the state transitions between its released and engaged states, which would considerably reduce the energy consumption as compared with the electromagnetic brake. A whole three-state prismatic actuator conceptual design is shown in Figure 4.6. As can be seen from the figure, the prismatic actuator consists of a DC motor, a backdrivable lead screw, the mechanical locker and some electronics components including an Arduino Nano board, a rotary encoder, an MD10C and an L9110S motor drivers, a rechargeable battery and an nRF24L01+ communication module. Since the designed prismatic actuator has actuation, computation, communication and sensing capabilities, it actually can be considered as a self-contained robotic module when constructing modular robots using such prismatic actuators. Just like Scheme II, the three states can be achieved by the collaborative work of the mechanical locker and the DC motor.

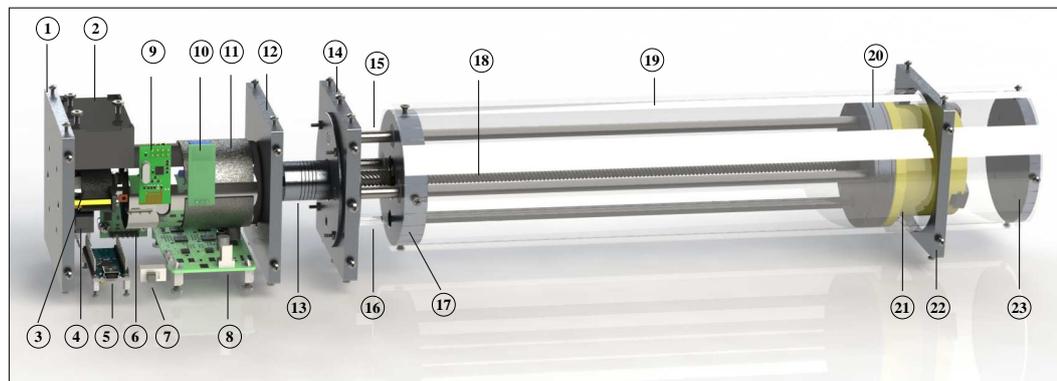
Design Scheme IV

In the above described Schemes II and III, the lead screw and the DC motor are coupled by using a flexible coupler. Therefore, in the passive state, the lead screw and the motor shaft must rotate together when applying appropriate external forces to the prismatic actuator. Hence, a DC motor with low detent torque must be selected for Schemes II and III. Unlike the aforementioned designs, this section presents a potential solution

4.2 Mechatronics Designs and Hardware Implementation



(a)



(b)

Figure 4.6: Rendered Three-State Prismatic Actuator with a Mechanical Locker. (a) Prismatic Actuator with Casing. (b) Prismatic Actuator without Casing. (1) End Cap. (2) Battery Holder. (3) Rotary Encoder. (4) Transistor. (5) Arduino Nano. (6) L9110S Motor Driver. (7) Switch. (8) MD10C Motor Driver. (9) nRF24L01+ Module. (10) ACS712 Current Sensor. (11) DC Motor. (12) Motor Mount. (13) Flexible Coupler. (14) Bearing Mount. (15) Aluminium Shafts. (16) Acrylic Hollow Shaft. (17) Nut and Linear Bearing Mount. (18) Lead Screw. (19) Acrylic Tube. (20) Bearing Mount. (21) Locker Casing. (22) Tube Support Plate. (23) End Cap.

wherein the lead screw and the DC motor can have no coupling relationship. Figure 4.7 shows the CAD model of a gear box that can realize three states. The gear box is mainly comprised of two micro DC motors, a thin section bearing, a slip ring and a planetary gear set. When the locker is released and the actuation motor is turned

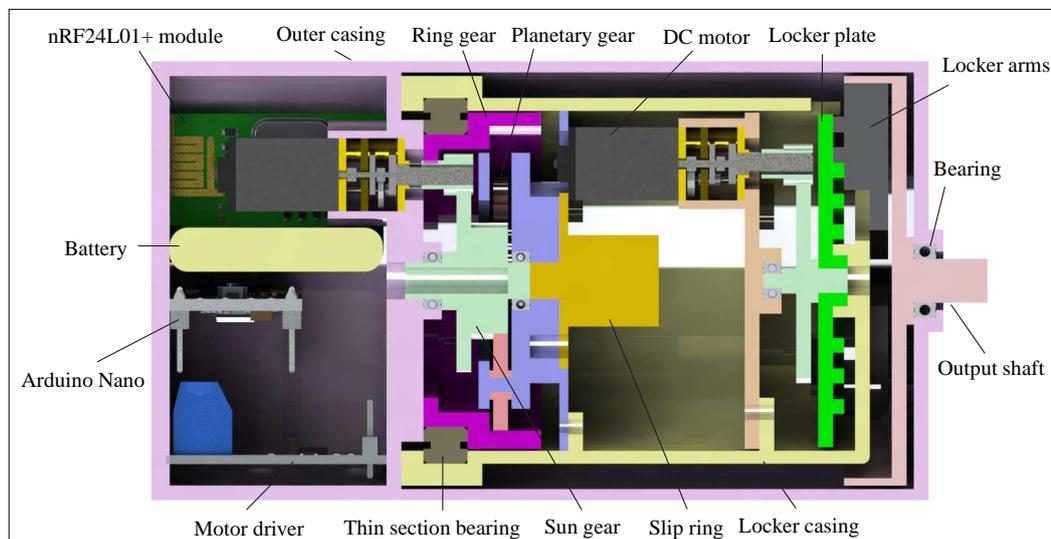


Figure 4.7: Section View of a Rendered Three-State Gear Box.

off, the output shaft can be rotated passively. A three-state prismatic actuator can be obtained by connecting the shaft and a lead screw using a coupler. Since the lead screw has no coupling relationship with the actuation motor, smaller external forces are required to rotate the lead screw in its passive state. When releasing the locker and turning on the actuation motor, the actuated state is acquired. Locked state is achieved by turning on the locker motor while turning off the actuation motor at the same time. Evidently, such a prismatic actuator could be easier to passively respond to external forces. However, the gear box is much more complex than the previous designs, which would lead to more difficulties in physical implementation.

4.2.2 Rigid Connector Node and Passive Revolute Joint

As discussed in the preceding chapter, the connectivity strategy using rigid node in a 2D plane requires a rigid node with four attachment points. To facilitate the future use of the rigid node in 3D space, a rigid node with six attachment points is preferred [Yu *et al.*, 2008, Zagal *et al.*, 2012]. One simple and common design for a node is a spherical ball shown in Figure 4.8(a) with threaded female holes and threaded male connectors. Manufacturing such plastic balls using a 3D printer would be extremely expensive as the balls have a large volume and need a lot of materials. There exist off-the-shelf solid polyoxymethylene (POM) plastic balls with a diameter of 10 cm

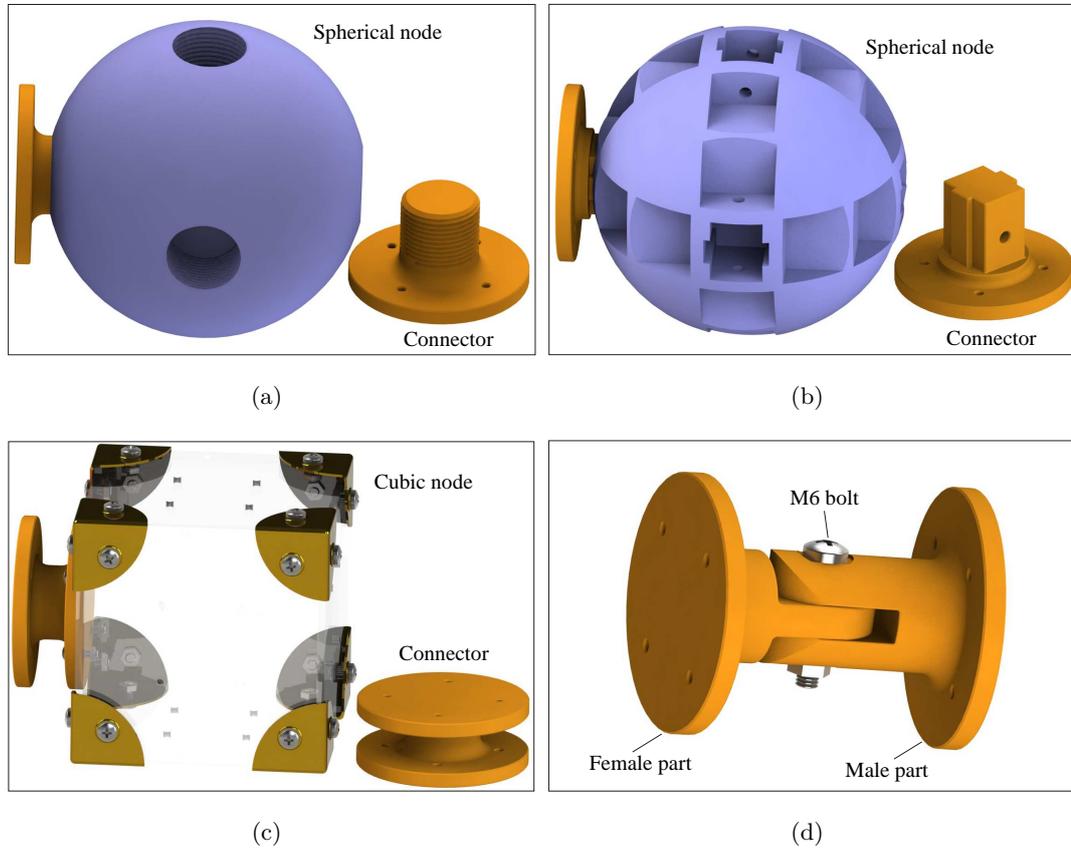


Figure 4.8: Connector Nodes and Revolute Joint. (a) Spherical Node with Threaded Holes. (b) Spherical Node with Tabbed Holes. (c) Cubic Node. (d) Revolute Joint.

which is quite tough and durable. However, since the prismatic actuator expected to be prototyped in Scheme III is rectangular, there would be some difficulties in aligning modules connected using such threaded balls. Another solution to a rigid node as shown in Figure 4.8(b) is to use tabbed holes but it needs a more complicated procedure to manufacture it. Figure 4.8(c) presents a more elegant solution in which a cubic node is made up from acrylic boards. The acrylic boards are easy to be cut using laser cutting technique which is much more time-efficient than computer numerical control (CNC) machining or 3D printing in this case. More importantly, it is easier to ensure the alignment between rectangular prismatic actuators and cubic nodes. Therefore, the cubic type connector node is selected to implement for the modular robotic system.

In the connectivity strategy, a passive revolute joint is placed in the middle of two

prismatic actuators. In a 2D plane, a one-DOF passive hinge joint with two attachment faces can meet the connectivity requirements. A more complex passive revolute joint could be a two-DOF universal joint or a three-DOF ball-and-socket joint. As a starting point, this research only considers the 2D case and therefore a one-DOF hinge joint is designed. Figure 4.8(d) shows a simple solution to a passive hinge joint consisting of two parts that are jointed using a bolt and a nut. Each part of the revolute joint has 4 holes for connecting with a prismatic actuator using bolts. Such a design is easy for manual attachment and detachment.

4.2.3 Physical Prototypes

Three prototypes are built iteratively during the implementation process. The first one is a prototype implemented for design scheme II, which works as a platform for testing and selecting electronics components. The second prototype is built using aluminium to verify the feasibility of design scheme III and the third prototype is a modified and updated 3D printed version of prototype II.

Prototype I

Figure 4.9 presents an implemented prismatic actuator for Scheme II. The employed lead screw has seven start threads with each start having a pitch of 2 mm, providing a quick 14 mm translation. This guarantees its high back-drive capability to passively respond to external forces. Gear reduction would weaken the back-drive capability, therefore the lead screw is coupled directly with a DC motor that has a low detent torque to be rotated easily. The DC motor has a back shaft allowing an incremental quadrature encoder to be attached to. The adopted rotary encoder has two channels with its cycles per revolution (CPR) being 200 which can provide 200, 400 or 800 pulses per revolution depending on whether X1, X2 or X4 decoding mode is used. To achieve a high positioning accuracy, X4 decoding mode is chosen which means a displacement of 14 mm of the lead screw corresponds to 800 pulses of the encoder resulting in a precision of 0.0175 mm of the prismatic-actuator system. At the opposite end of the DC motor side, the lead screw is connected with an electromagnetic brake. As for electronics, the DC motor is driven by an MD10C motor driver which has a working voltage ranging from 5 V to 30 V and a maximum continuous current up to 13 A. The brake is controlled by an L9110S motor driver having a maximum drive current

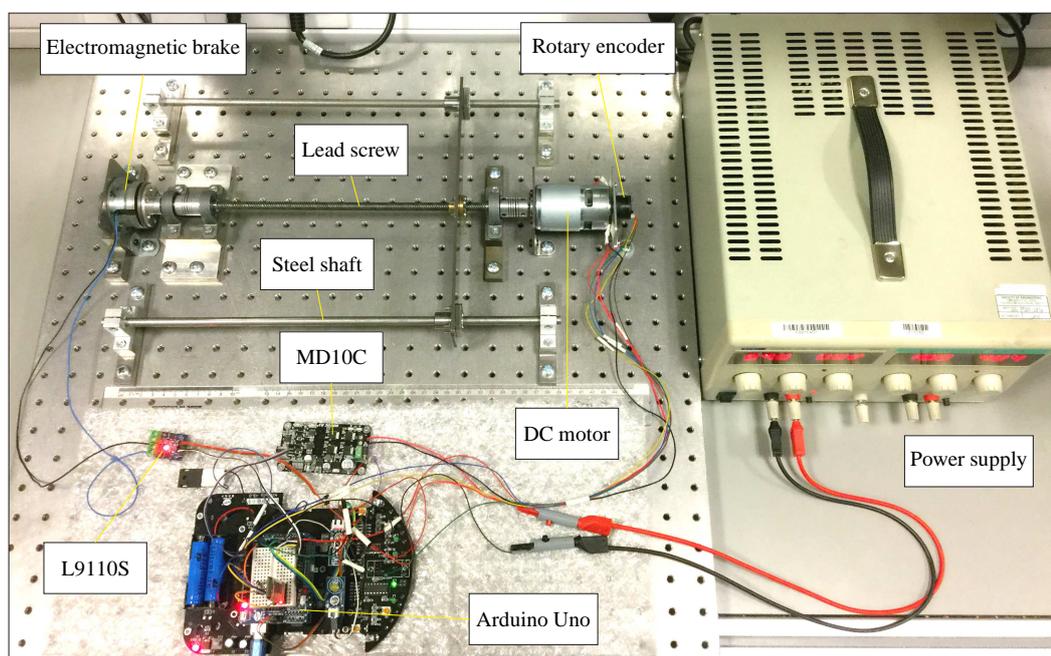


Figure 4.9: Prototype I of a Three-State Prismatic Actuator.

of 0.8 A and can work under a supply voltage between 4.5 V and 20 V. The power supply can offer a voltage ranging from 0 V to 35 V and a maximum current of 4 A. An Arduino Uno board is used to work as a computation unit that can receive remote control signals by an integrated IR receiver. Additionally, the Arduino Uno board is capable of communicating with a laptop through a serial cable to collect experimental data such as the encoder readings.

Prototype II

Figure 4.10 shows the implemented aluminium version prototype for Scheme III with its components presented in Figure 4.11. The prismatic actuator has a dimension of $86 \times 86 \times 480$ mm when fully contracted and it can have a fully extended length of 730 mm leading to a stroke length of 250 mm. The weight of the prismatic actuator is around 2.2 kg. Mechanically, the prismatic actuator consists of an acrylic casing, an acrylic tube, a steel lead screw, an acrylic hollow shaft and three aluminium shafts. The locker casing is a Vero White 3D printed part and the rest of the mechanical parts are made of aluminium. The actuator is powered by a rechargeable battery with 9

4.2 Mechatronics Designs and Hardware Implementation

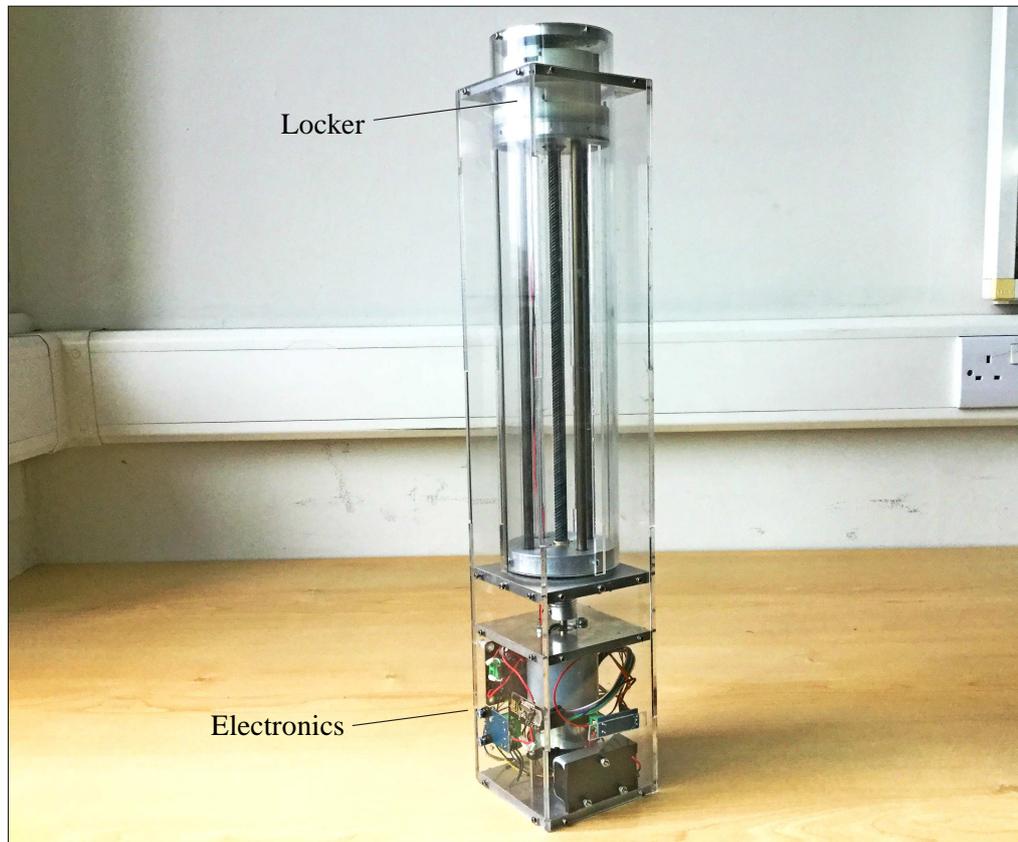


Figure 4.10: Aluminium Version of Prototype II.

V which provides voltage for the Arduino Nano board, the MD10C motor driver and the L9110S motor driver. The micro DC motor inside the locker is connected with the L9110S motor driver via two cables that go through the acrylic hollow shaft. An ACS712 current sensor is used to monitor the current drawn by the actuator. The DC motor of the actuator is driven by the MD10C motor driver which can receive direction and PWM signals from the Arduino Nano board. A transistor, a diode and a resistor are utilized to serve as a switch to control the power for the MD10C motor driver so that a passive state can be achieved as the prismatic actuator cannot move very smoothly when MD10C is powered on even with the PWM value set as 0.

Since the aluminium parts are time consuming and expensive to fabricate, as a cheaper and rapid solution, 3D printing is used to produce the mechanical parts. To ensure the toughness of the printed parts, PolyMaxTM poly-lactic acid (PLA) filament

4.2 Mechatronics Designs and Hardware Implementation



Figure 4.11: Mechatronics Parts of the Aluminium Version of Prototype II. (1) Acrylic Tube. (2) Spiral Plate. (3) Micro Motor Mount. (4) Slider Arms. (5) Lead Screw. (6) Aluminium Shafts. (7) Acrylic Hollow Shaft. (8) Bearing Mount. (9) Plastic Locker Casing. (10) End Cap. (11) Motor Mount. (12) End Cap. (13) Nut and Linear Bearing Mount. (14) Bearing Mount. (15) Tube Support Plate. (16) Acrylic Casing. (17) Rotary Encoder. (18) Brass Nut. (19) Flexible Coupler. (20) Rotary Bearings. (21) Plastic Linear Bearings. (22) L9110S Motor Driver. (23) DC Motor. (24) Battery Holder. (25) 9 V Rechargeable Battery. (26) Micro DC Motor. (27) Transistor. (28) MD10C Motor Driver. (29) ACS712 Current Sensor. (30) nRF24L01+ Module. (31) Arduino Nano. (32) Switch.

with a diameter of 3 mm is selected as such a PLA material is claimed to be extremely tough featuring enhanced impact strength of up to nine times that of regular PLA and 20% superior than ABS. Unlike ABS, the PolyMaxTM PLA material can get rid of the harmful odour and potential safety risk that results from printing ABS. Figure 4.12 shows an assembled prismatic actuator consisting of orange PLA printed mechanical parts. The whole components of a PLA version prototype is presented in Figure 4.13. The PLA version shares almost the same characteristics and components that the aluminium version possesses. It has a same dimension of $86 \times 86 \times 480$ mm and a same stroke length of 250 mm. It is lighter with a mass of 1.8 kg owing to the usage

4.2 Mechatronics Designs and Hardware Implementation

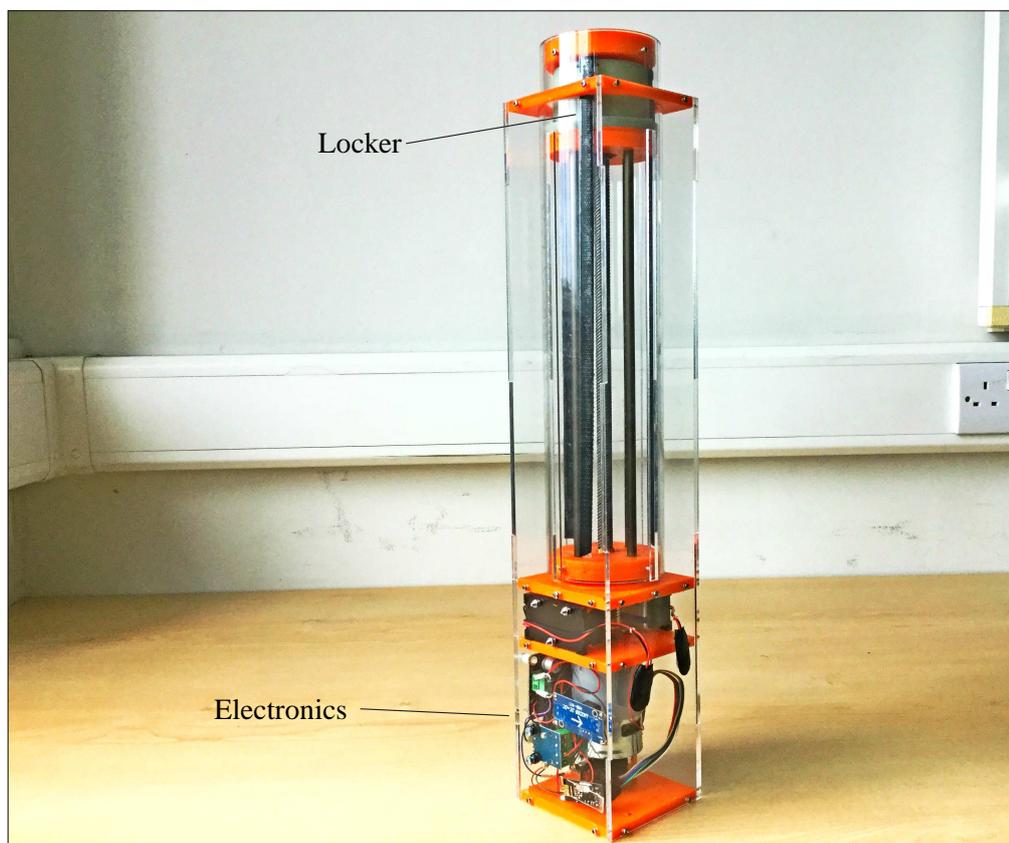


Figure 4.12: PLA Version of Prototype II.

of plastic materials instead of aluminium materials. To increase the friction between the slider arms of the locker and the acrylic tube, the bearing mount near the locker side is keyed to make it possible to put three Velcro strips inside the acrylic tube. The PLA version has one more 9 V battery to enhance its actuation capability and an adjustable voltage regulator to convert the voltage of the batteries into 9 V for powering the Arduino Nano board and driving the locker. The electronics schematic diagram is shown in Figure 4.14. The Arduino Nano breakout board serves as the brain of the actuator. An nRF24L01+ module that can communicate with other actuators using radio is connected with the Arduino Nano through an SPI bus. The rotary encoder is connected with the Arduino Nano via two digital pins (i.e., D2 and D3) that are usable for interrupts to record the encoder readings in a quick manner. For controlling the DC motor of the actuator, the Arduino can send direction and PWM signals to the MD10C

4.2 Mechatronics Designs and Hardware Implementation

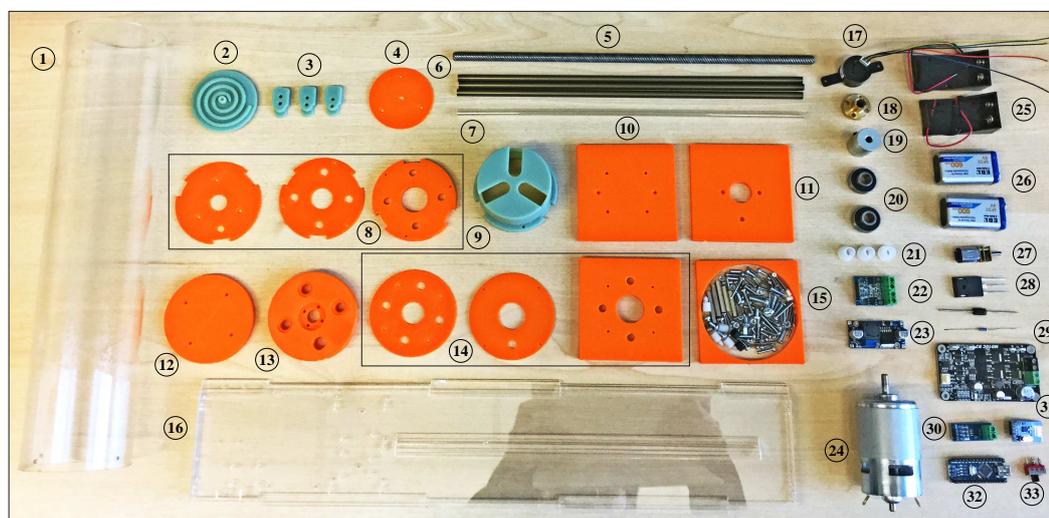


Figure 4.13: Mechatronics Parts of the PLA Version of Prototype II. (1) Acrylic tube. (2) Spiral plate. (3) Slider arms. (4) Micro motor mount. (5) Lead screw. (6) Aluminium shafts. (7) Acrylic hollow shaft. (8) Bearing mount. (9) Locker casing. (10) End cap. (11) Motor mount. (12) End cap. (13) Nut and linear bearing mount. (14) Bearing mount. (15) Tube support plate. (16) Acrylic casing. (17) Rotary encoder. (18) Brass nut. (19) Flexible coupler. (20) Rotary bearings. (21) Plastic linear bearings. (22) L9110S motor driver. (23) Voltage regulator. (24) DC motor. (25) Battery holders. (26) 9 V rechargeable batteries. (27) Micro DC motor. (28) Transistor. (29) MD10C motor driver. (30) ACS712 current sensor. (31) nRF24L01+ module. (32) Arduino Nano. (33) Switch.

motor driver to control the rotational direction and speed of the motor. In order to achieve a pure passive state, a TIP142 bipolar NPN transistor is exploited to work as a switch to turn off the power supply of the MD10C motor driver. An FR307G diode is connected with the positive and negative leads of the MD10C motor driver in parallel to deal with cases where sudden voltage spikes seen across the load when the supply current is suddenly reduced or interrupted. For the locker motor, by changing the two digital input signals of the L9110S motor driver, the direction of the micro motor can be controlled. Constant speed is used for controlling the locker and the displacements of the slider arms are managed by setting the working time.

A cubic node with a rigid connector and a passive revolute joint are shown in Figure

4.2 Mechatronics Designs and Hardware Implementation

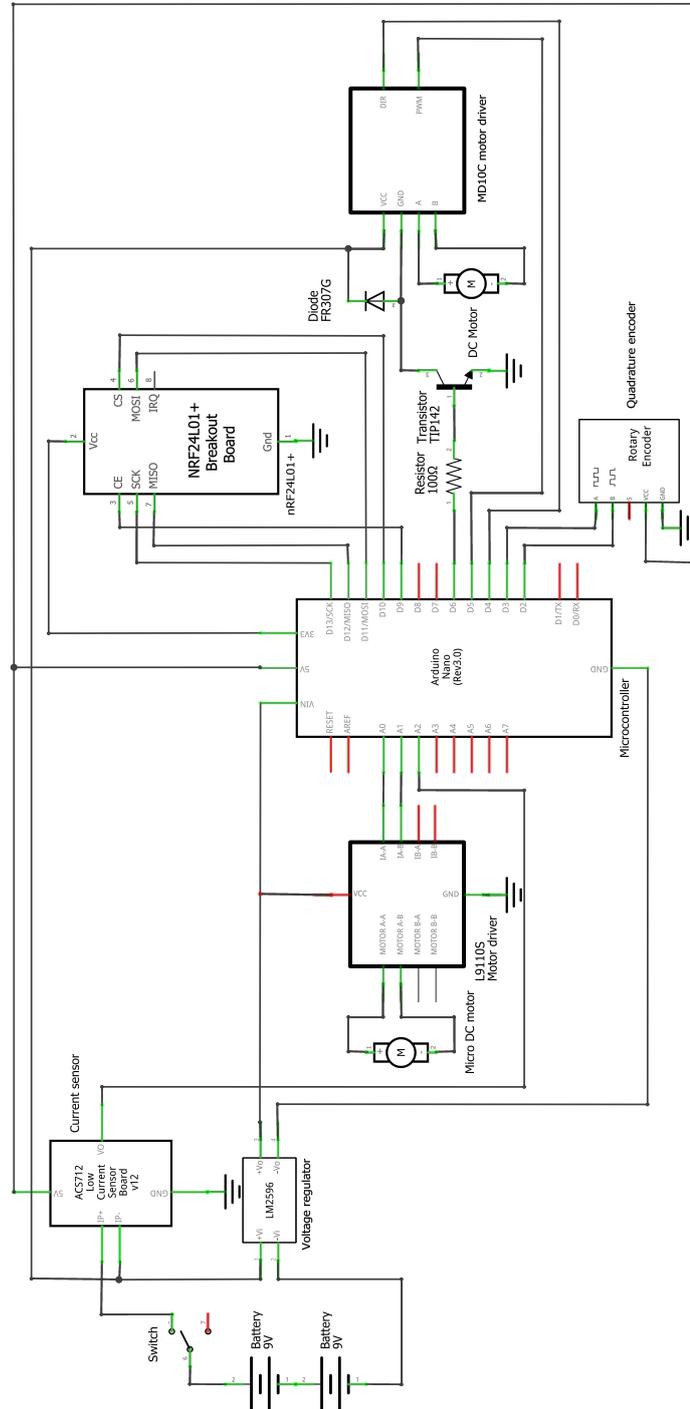


Figure 4.14: Electronics Schematic of the PLA Version Prismatic Actuator.

4.3 Mechatronics Implementation Considerations

4.15. The cubic node is built from six acrylic boards and eight brass corners and the revolute joint consists of two 3D printed parts jointed by a bolt. Both of the cubic node and the revolute joint have faces for interfacing with prismatic actuators. By using a 3D printer, more modular elements including eight identical prismatic actuators, four rigid connector nodes and four passive revolute joints are fabricated and assembled. Figure 4.16(a) and (b) shows one snapshot captured during the assembly process and the assembled eight prismatic actuators.

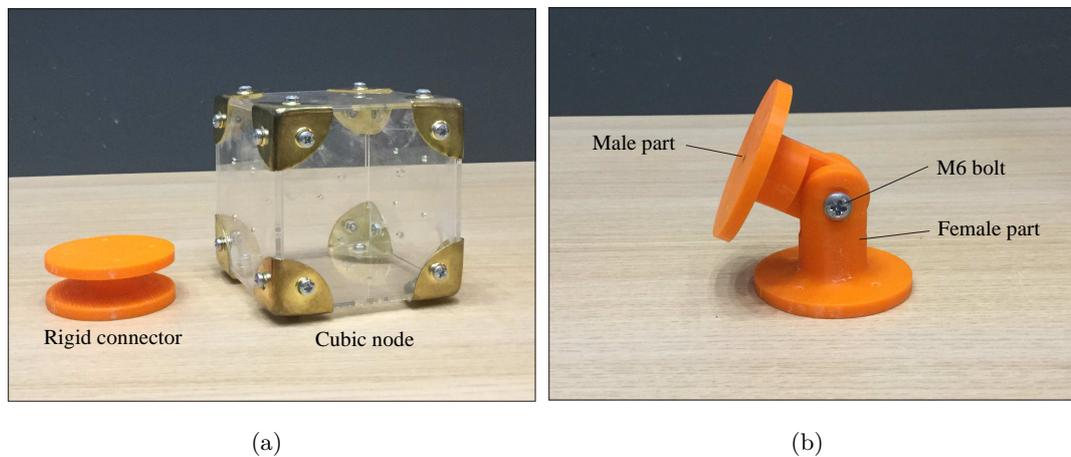


Figure 4.15: Connector Nodes and Revolute Joint. (a) Cubic Node. (b) Revolute Joint.

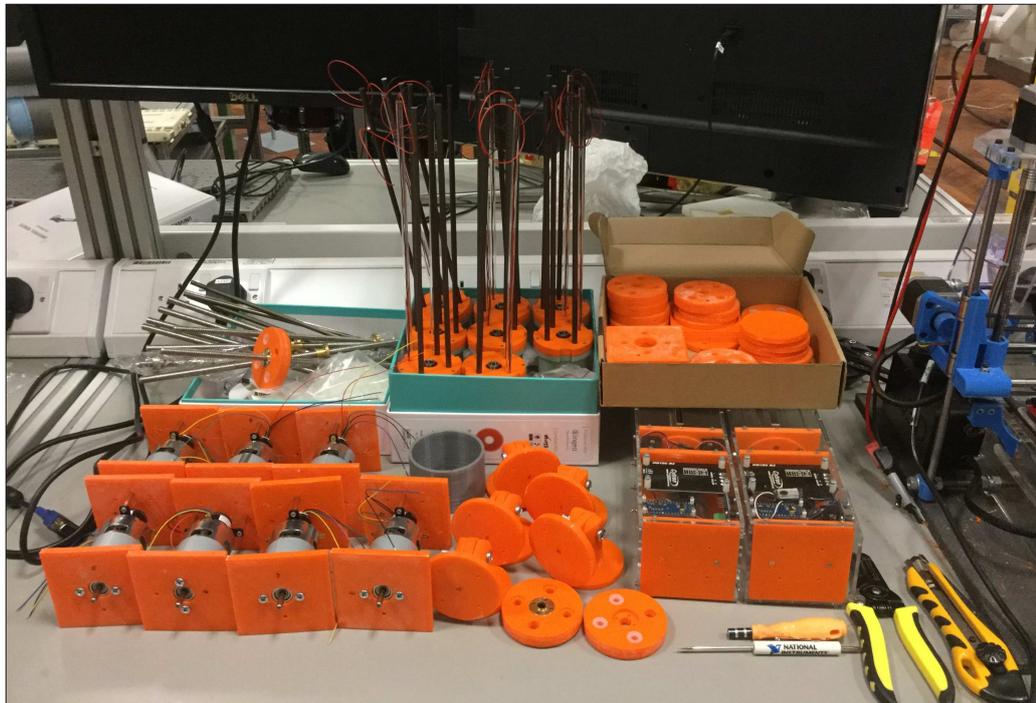
4.3 Mechatronics Implementation Considerations

A trade-off between available budget, prototype performance, physical feasibility and manufacturing time has to be made when designing and implementing a robotic system. This section analyses and presents the selection and fabrication of some key components during the implementation process.

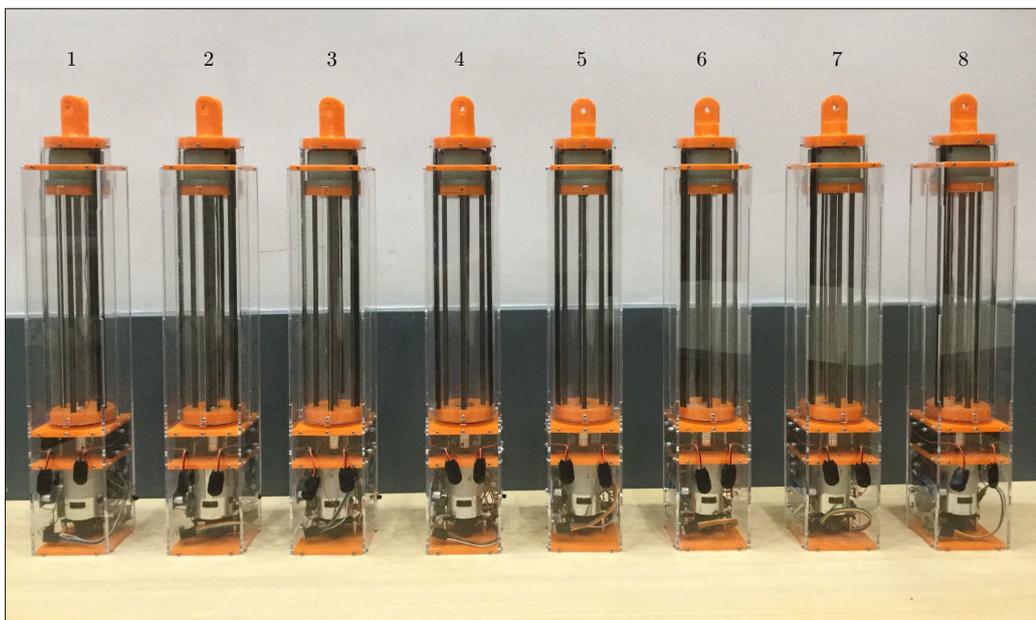
4.3.1 Lead Screw

To drive a linear motion, lead screws and ball screws are two commonly used mechanical solutions which translate rotary motions to linear motions. Generally, lead screws utilize a rod with single- or multiple-start threads and a mating nut while ball screws also use ball bearings that re-circulate in the matching grooves between the threaded

4.3 Mechatronics Implementation Considerations



(a)



(b)

Figure 4.16: Assembly of Eight PLA Version Prismatic Actuators. (a) Assembly Process. (b) Assembled Actuators.

4.3 Mechatronics Implementation Considerations

rod and the nut to eliminate frictions. A lead screw tends to be more lightweight, less efficient and cost-effective as compared with a ball screw. The back-drive capability of a lead screw or a ball screw is closely related to its mechanical efficiency. A higher efficiency leads to a stronger back-drive capability and a more smooth and quicker passive motion. Mathematically, let parameters p , n , l , α , μ , d_m , d_r , d_p denote the pitch, number of threads, lead, lead angle, coefficient of friction, major diameter, root diameter and pitch diameter of the lead screw, respectively. The following relationships between these parameters can be obtained [Hollander & Sugar, 2006]:

$$l = np = \pi d_p \tan \alpha, \quad (4.1)$$

$$\eta_1 = \frac{1 - \mu \tan \alpha}{1 + \mu \cot \alpha} = \frac{l(\pi d_p - \mu l)}{\pi d_p(l + \mu \pi d_p)}, \quad (4.2)$$

$$\eta_2 = \frac{1 - \mu \cot \alpha}{1 + \mu \tan \alpha} = \frac{\pi d_p(l - \mu \pi d_p)}{l(\pi d_p + \mu l)}, \quad (4.3)$$

where η_1 is the efficiency of converting rotary to linear motions and η_2 is the efficiency of translating linear to rotary motions. As can be seen from Equations (4.1)–(4.3), efficiencies η_1 and η_2 are the functions of μ , l and d_p . For better understanding, Figure 4.17(a)–(c) shows the profiles of η_1 and η_2 with respect to μ , d_p and l , respectively. As can be seen from the figure, efficiencies η_1 and η_2 decrease with the increase of μ and d_p while η_1 and η_2 increase by increasing l . Therefore, a lead screw or ball screw with a low coefficient of friction, a small diameter and a long lead is preferable.

For a passive motion, suppose there is an axial load F , the back-drive torque that

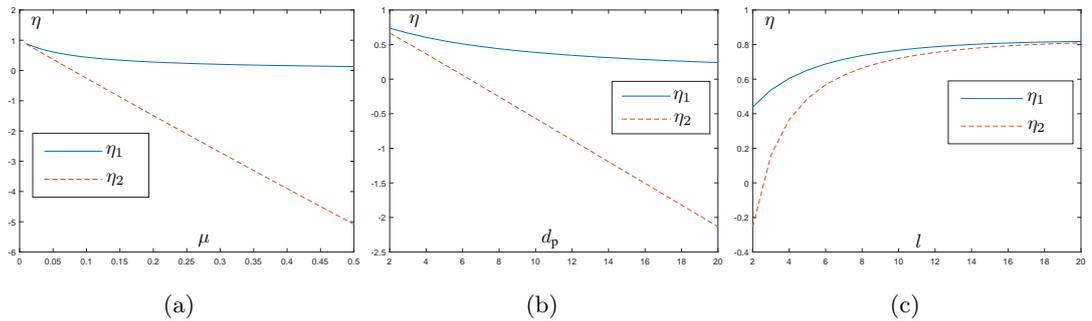


Figure 4.17: Lead Screw Efficiency Profiles. (a) η Versus μ Plot with $l = 2$ mm and $d_p = 8$ mm. (b) η Versus d_p Plot with $\mu = 0.1$ and $l = 2$ mm. (c) η Versus l Plot with $\mu = 0.1$ and $d_p = 8$ mm.

4.3 Mechatronics Implementation Considerations

can passively move the threaded rod can be calculated as [Budynas & Nisbett, 2014]:

$$\tau_b = \frac{Fl\eta_2}{2\pi}. \quad (4.4)$$

If τ_b is large enough to counteract the linear system's friction torques including friction torques between the threaded rod and the nut as well as between the rod and the bearings, then the rod can back drive due to external force F . In practice, torque τ_b is hard to be directly determined by taking into account all the potential frictions in a system. Equation (4.4) lays a basis for experimentally measuring τ_b using different loads F . Considering that one of promising properties of modular robot is low cost, a lead screw that only costs around 5 pounds is selected under the guidance of Equations (4.1)–(4.4) as the final solution to building the prismatic actuators. The selected lead screw has parameters $p = 2$ mm, $n = 7$, $d_m = 8$ mm, $d_r = 6$ mm and $d_p = 7$ mm. In addition, coefficient of friction μ is estimated as 0.15 [Budynas & Nisbett, 2014]. By substituting these values into (4.1)–(4.3), the efficiencies are obtained as $\eta_1 = 73.20\%$ and $\eta_2 = 69.78\%$.

4.3.2 Shafts and Bearings

Shafts and bearings are extremely important for obtaining a smooth motion of a prismatic-actuator system. Since the actuation and locker motors are at the two different ends of the lead screw and the electronics circuits are located in the actuation motor's side, long cables are required to power the locker motor. Due to this fact, hollow shafts are first considered. However, precision hollow shafts are provided by less suppliers and tend to be more expensive as compared with standard solid shafts. Hence, three precision shafts and one hollow acrylic tube are finally adopted. Figure 4.18 shows two different precision shafts made of steel and aluminium with almost the same specifications. An aluminium shaft is more lightweight and cheaper than a steel shaft. Therefore, the final vote is given to the aluminium one. Regarding the bearings, ball bearings and plastic nylon bearings are considered. Since ball bearings need lubrication and are more expensive, nylon bearings are selected.

4.3.3 Microcontroller

Nowadays, there exist many affordable and popular electronics development boards in the robotics community, for example, Arduino, Raspberry Pi and Programmable

4.3 Mechatronics Implementation Considerations

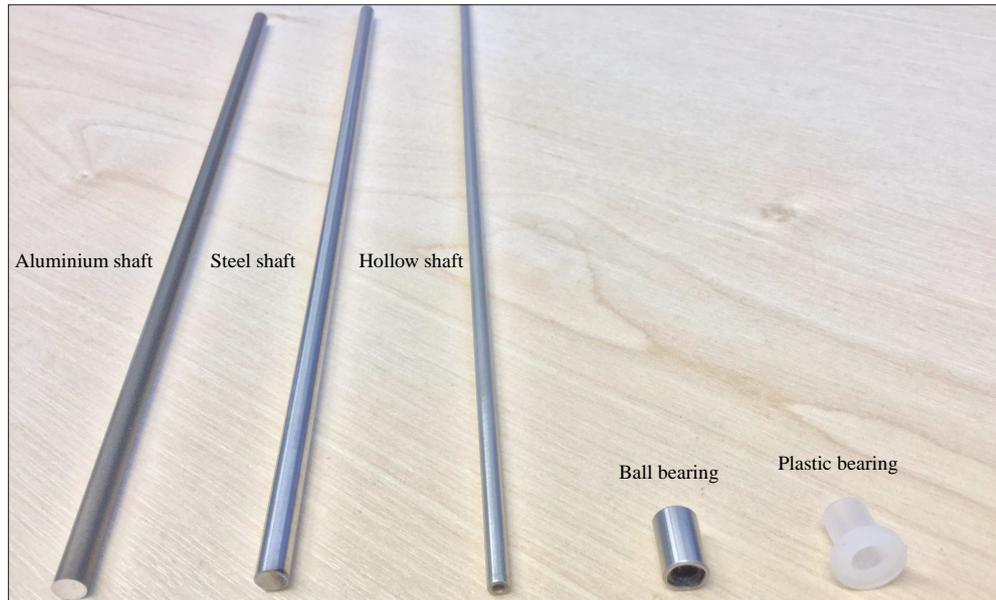


Figure 4.18: Different Shafts and Bearings Considered in the Implementation Process.

System on Chip (PSoC) [Dokter, 2015, Krauss, 2016, Prasanna & Rathore, 2014]. Arduino is an open-source and low-cost electronics prototyping platform together with an integrated development environment (IDE) [Krauss, 2016, Umedachi *et al.*, 2013]. An Arduino board contains a microcontroller that is programmed in C/C++ and a broad range of peripheral devices like sensors and actuators are compatible with Arduino. Up to date, Arduino has a series of products with different sizes and shapes to meet different requirements. It has a large community where one can find a lot of useful tutorials and materials to speed up the development process. Raspberry Pi is a micro-computer with a Linux operating system and can work like a desktop or a laptop with appropriate peripheral devices such as a keyboard, a mouse and a screen, which is much more powerful than a microcontroller like Arduino [Krauss, 2016, Vandeveld *et al.*, 2013]. PSoC from Cypress Semiconductor has one unique characteristic: it has both analogue and digital circuitries that are dynamically reconfigurable [Prasanna & Rathore, 2014]. It is allowed to program multiple configurations in a PSoC chip and then change these configurations in real time when the program is running. In this way, the resources of the chip can be fully exploited.

Table 4.2 compares some typical and popular Arduino, Raspberry Pi and PSoC

Table 4.2: Comparison of Arduino Uno, Arduino Nano, PSoc 5LP CY8CKIT-050 and Raspberry Pi 3 Model B

Boards	Arduino Uno	Arduino Nano	PSoc 5LP CY8CKIT-050	Raspberry Pi 3 Model B
Microprocessor	8-bit ATmega328P	8-bit ATmega328P	32-bit ARM Cortex-M3	64-bit ARM Cortex-A53
Architecture	AVR	AVR	ARMv7-M	ARMv8-A
Clock speed	16 MHz	16 MHz	80 MHz	1.2 GHz
Flash memory	32 KB	32 KB	256 KB	Inapplicable
SRAM	2 KB	2 KB	64 KB	1 GB
EEPROM	1 KB	1 KB	2 KB	Inapplicable
Power supply	Recommended 7 – 12 V	Recommended 7 – 12 V	9 or 12 V	5.1 V from micro USB
Operating voltage	5 V	5 V	1.71 – 5.5 V	5 V
Analogue I/O pins	6	8	72 analogue & digital pins	40 analogue & digital pins
Digital I/O pins	20	22	72 analogue & digital pins	40 analogue & digital pins
External interrupts	2	2	32	40
I/O pin current	40 mA on I/O pins 50 mA on 3.3 V pin	40 mA	100 mA	16 mA
Communication	Serial, UART, SPI, PWM, I2C	Serial, UART, SPI, PWM, I2C	Serial, UART, SPI, PWM, I2C, CAN, RS-232, Bluetooth	Serial, UART, Wi-Fi, Bluetooth
PCB size	53.4 × 68.6 mm	18 × 45 mm	101.6 × 152.4 mm	56.5 × 85.6 mm
Weight	25 g	7 g	100 g	45 g
Cost*	£16.16	£14.40	£74.74	£32.99

* For a fair comparison, all the costs are collected from <http://uk.rs-online.com/web/>.

boards. The PSoC 5LP CY8CKIT-050 development board is the most expensive one even though it has more functionalities than Arduino Uno and Nano. As a micro-computer, Raspberry Pi 3 Model B has the most powerful microprocessor for computation. However, it has no wired buses like SPI and I2C that can be used for communication with other electronics boards. Additionally, it is not as suitable as a microcontroller for executing tasks that need precise timing [Vandeveldel *et al.*, 2013]. Arduino Uno and Nano boards have similar and almost the same functionalities and the Nano board with 2 more available pins is cheaper and much smaller. Based on the above analysis, Arduino Nano with an ATmega328P microprocessor is selected for our application as a functional and cost-effective solution.

4.3.4 Motors

This part analyses the motors that can meet the design requirements. Specifically, DC motors for realizing the actuated and locked states are investigated.

Actuated State As compared with Prototype II, Prototype I needs more power to actuate the prismatic-actuator system as it has a brake rotor hub and a brake shaft that are coupled with the lead screw. Therefore, this research focuses on analysing the actuated state of the Prototype I for a vertical application. Specifically, the DC motor must be capable of providing enough torques for overcoming load inertia and gravity to actuate the prismatic-actuator system vertically. The inertia of the lead screw, the brake and the brake shaft for Prototype I can be obtained according to the following basic formula [Mazurkiewicz, 1995, Voss, 2007]:

$$J = md^2/8,$$

where m and d denote the mass and diameter of a cylindrical object, respectively. Following the above formula, it is easy to acquire the required inertia values: $J_{\text{screw}} = 7.36 \times 10^{-7} \text{ kg} \cdot \text{m}^2$, $J_{\text{brake}} = 6.53 \times 10^{-6} \text{ kg} \cdot \text{m}^2$ and $J_{\text{shaft}} = 4.02 \times 10^{-7} \text{ kg} \cdot \text{m}^2$. Assume a load with mass of $m_{\text{load}} = 2.5 \text{ kg}$ must be lifted, then $J_{\text{load}} = m_{\text{load}}l^2/(2\pi)^2 = 1.24 \times 10^{-5} \text{ kg} \cdot \text{m}^2$. Consequently, the inertia of the whole system is computed as

$$J_{\text{system}} = J_{\text{screw}} + J_{\text{brake}} + J_{\text{shaft}} + J_{\text{load}} = 2.01 \times 10^{-5} \text{ kg} \cdot \text{m}^2.$$

Meanwhile, the inertia moment of the DC motor is estimated as $J_{\text{motor}} = 1.2 \times 10^{-5} \text{ kg} \cdot \text{m}^2$ [Applied Motion Products, 2014, Global Electric Motor Solution, 2015]. After that,

4.3 Mechatronics Implementation Considerations

the torques required to accelerate the system and conquer the gravity can be calculated using the following equations [Budynas & Nisbett, 2014, Voss, 2007]:

$$\tau_a = (J_{\text{system}} + J_{\text{motor}})a, \quad (4.5)$$

$$\tau_g = \frac{m_{\text{load}}gl}{2\pi\eta_1}, \quad (4.6)$$

where a is the acceleration of the system and g represents the gravity acceleration. Suppose an extreme case where the prismatic-actuator system can move a full-stroke distance of 250 mm by following an isosceles triangular velocity profile with execution time of 1 s, then the angular peak velocity and acceleration of the motor will be about $w = 224.40 \text{ rad/s} = 2143 \text{ RPM}$ and $a = 448.8 \text{ rad/s}^2$, respectively. It follows (4.5) and (4.6) that $\tau_a = 14.4 \text{ mN} \cdot \text{m}$, $\tau_g = 74.6 \text{ mN} \cdot \text{m}$ and thus the total required torque $\tau_t = \tau_a + \tau_g = 89 \text{ mN} \cdot \text{m} = 0.089 \text{ N} \cdot \text{m}$.

An appropriate DC motor can be selected on the basis of the above theoretical results. Figure 4.19 shows various motors considered during the implementation process. Table 4.3 lists the measured results of the motors' continuous currents, stall currents and stall torques under a voltage of 3 V without load. The motor torques are measured using a torque meter that can measure a torque up to $1.471 \text{ N} \cdot \text{m}$ with a resolution of $0.001 \text{ N} \cdot \text{m}$. A power supply that can offer a maximum current up to 10 A is employed to energize the five motors and measure the drawn currents. As presented in Table 4.3, the stall current of Motor 3 is larger than 10 A, therefore the torque of Motor 3 is

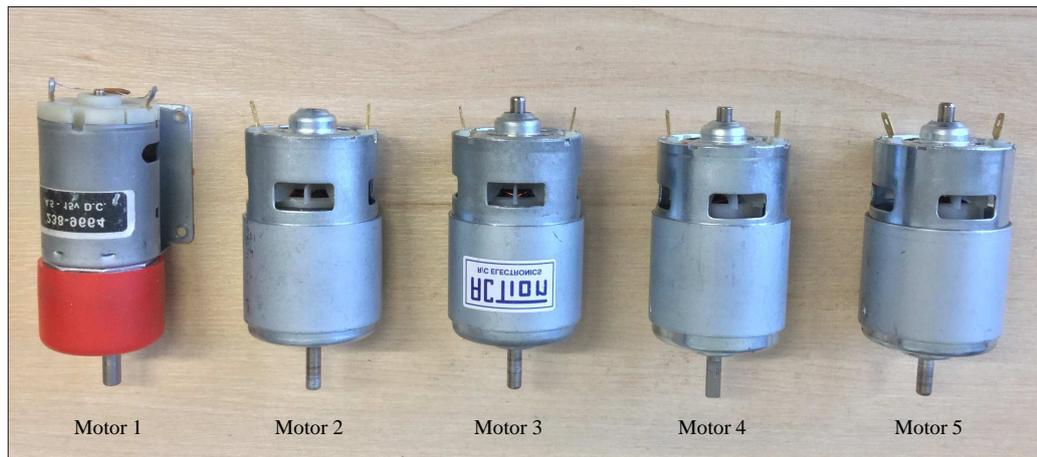


Figure 4.19: Different DC Motors Considered in the Implementation Process.

4.3 Mechatronics Implementation Considerations

Table 4.3: Comparison of Five Different DC Motors

Motors	Motor 1	Motor 2	Motor 3	Motor 4	Motor 5
Continuous current (A)	0.39	0.47	0.72	0.23	0.17
Stall current (A)	5.01	6.58	> 10	3.75	2.76
Stall torque (N · m)	0.28	0.137	–	0.106	0.096
Shaft diameter (mm)	6	5	5	5	5
Shaft shape	D-shaped	Round	Round	D-shaped	Round
Speed (RPM)*	274	1250	3000	1125	875
Required torque (N · m)	≥ 0.089		Required speed (RPM)		≥ 2143

* The motor speed listed in the table corresponds to a power supply of 3 V. If a 9 V battery is used, the speeds of Motors 1–5 are respectively around 822, 3750, 9000, 3375 and 2625 RPM.

inapplicable in this case. Torques of the other four motors are larger than 0.089 N · m even under a low voltage of 3 V, which can meet the analysed torque requirement. For the speed requirement, if a 9 V battery is used, speeds of Motors 2–5 are larger than the required 2143 RPM. Apart from the requirements of speed and torque, a lower continuous current is desirable to reduce the power consumption. The comparative results show that Motor 5 exhibits the smallest continuous current and stall current, meaning that Motor 5 wins in terms of the drawn current. Since the lack of important information especially the stall torque and stall current of Motor 5, experiments are conducted to measure these significant parameters. Figure 4.20(a) shows the setup including two power supplies and one torque meter for measuring the continuous and stall currents as well as the stall torque of the DC motor under different voltages. Power supply 1 is a digital instrument that can provide a voltage ranging from 0 to 42 V with a maximum available current of 10 A. Power supply 2 is an analogue device that can give a voltage between 8.1 and 14.9 V with a maximum available current of 32 A. Two power supplies are used since the stall current of the tested motor would be more than 10 A under 11 V. Therefore power supply 1 is employed to offer a series of voltages between 3.0 V and 10.0 V in steps of 1.0 V and measure the drawn currents of the motor in a relatively accurate manner. Under each voltage, stall currents and torques are measured five times. The corresponding results are presented in Figure 4.20(b) showing the relation between the average stall currents and the applied voltages. Meanwhile, the average

4.3 Mechatronics Implementation Considerations

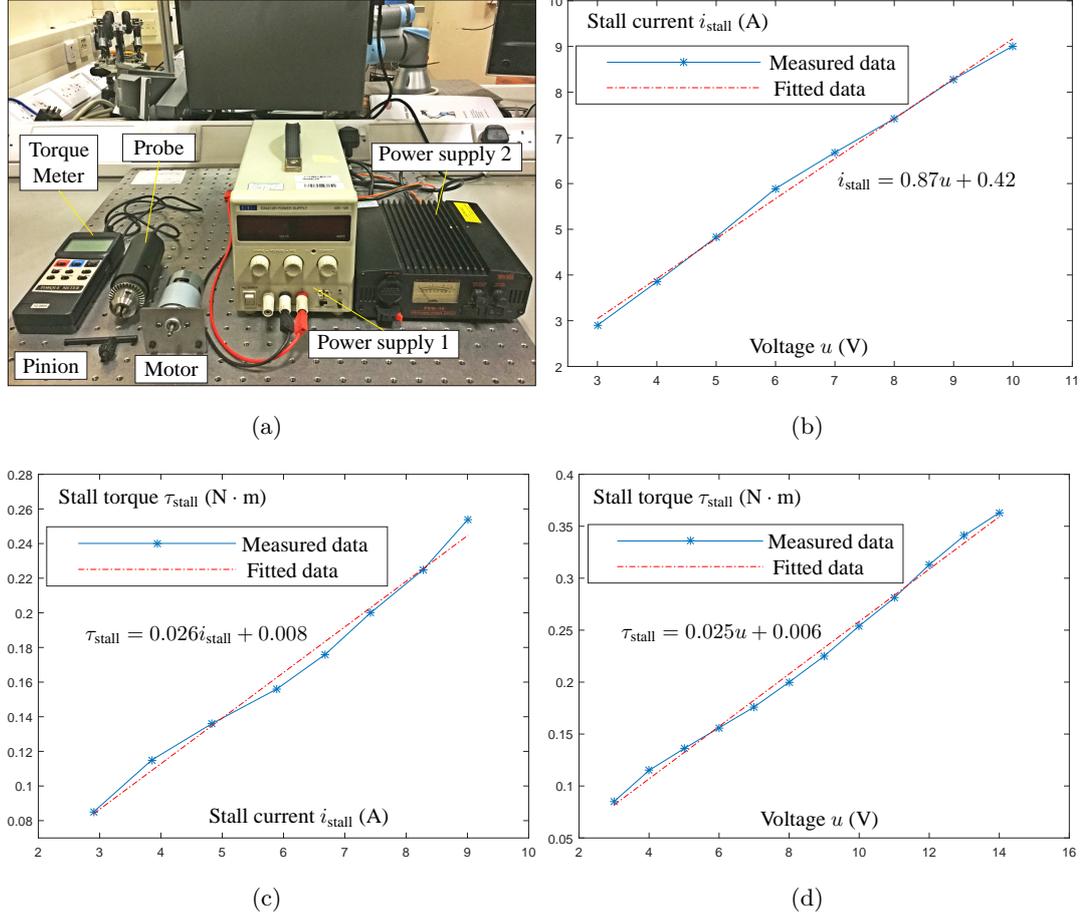


Figure 4.20: Experimental Test of the DC Motor 5. (a) Experimental Setup. (b) Average Stall Current Versus Voltage. (c) Average Stall Torque Versus Average Stall Current. (d) Average Stall Torque Versus Voltage.

stall torques under different voltages are also measured and the relation of the average stall torques and average stall currents is plotted in Figure 4.20(c). The whole measured results can be referred to Appendix C.1. Theoretically, torque τ and current i of a DC motor satisfies the following equation:

$$\tau = \kappa_t i, \quad (4.7)$$

where κ_t denotes the motor torque constant. From Figure 4.20(c), one can obtain an approximation value of κ_t as 0.026 N · m/A. By using power supply 2 and the torque meter, the stall torques and stall currents of the motor under voltages 11.0, 12.0, 13.0

4.3 Mechatronics Implementation Considerations

and 14.0 V are measured. Note that the current cannot be evaluated precisely using power supply 2 as it is an analogue equipment. The stall torque versus voltage plot is presented in Figure 4.20(d). As shown in the figure, the stall torque under 14.0 V is 0.363 N · m. Hence, the motor can meet with the supposed minimum torque of 0.089 N · m. In the PLA version prismatic actuator, two serially connected 9 V batteries are used to offer an actual voltage of around 17 V. The continuous current of the motor without load is measured around 0.25 A under a voltage of 17 V. The stall torque corresponding to a voltage of 17 V can be estimated as 0.431 N · m. In this case, an estimated stall current of 16.58 A is obtained by considering Equation (4.7).

Locked State When using an electromagnetic brake to realize a locked state in Prototype I, no extra motor is required. According to the data sheet, when the electromagnetic brake is engaged, it can provide a torque of 1.8 N · m under 24 V. When using the designed mechanical locker in Prototype II, a force analysis is conducted in order to select a feasible motor. The released and engaged states of the locking mechanism are shown in Figure 4.21(a) and (b), respectively. The rotary motion of the locker plate results in a linear motion of the three slider arms due to the contact of the spiral groove of the locker plate and the round pins of the slider arms. Mathematically, the parametric equation of the Archimedes spiral shown in Figure 4.21(c) is depicted as

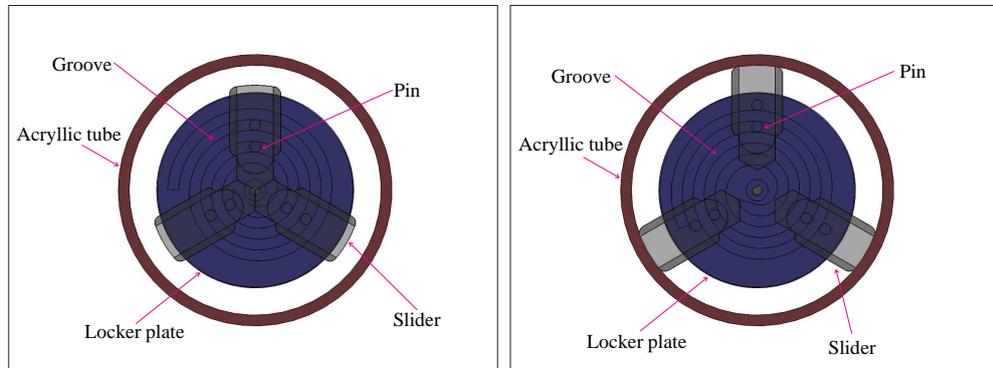
$$\begin{cases} x = (a + b\theta) \cos(\theta), \\ y = (a + b\theta) \sin(\theta), \\ z = c\theta, \end{cases} \quad (4.8)$$

where variable $\theta \in \mathbb{R}$ and a , b and c are positive constants. Figure 4.21(d) presents the projection of the 3D tapered helix on the X-Z plane for better understanding. For such a helix with a pitch p , equation $\tan(\alpha) = (d_{\max} - d_{\min})/2h$ is satisfied where α is the angle of taper, h denotes the height and d_{\max} as well as d_{\min} represent the maximum and minimum diameters of the helix, respectively. The helix used in the locker plate illustrated in Figure 4.21(e) is obtained by setting $z = 0$, which can be depicted in polar coordinates as:

$$\rho = a + b\theta. \quad (4.9)$$

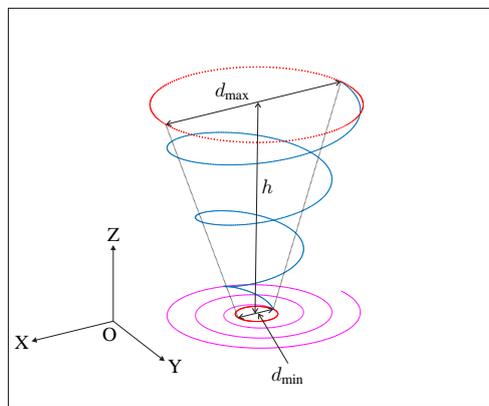
When the plate rotates a full revolution of 2π rad, the slider arms would move $p \tan(\alpha)$ mm or $[a + b(\theta + 2\pi)] - (a + b\theta) = 2\pi b$ mm linearly. Schematic diagrams of the spiral

4.3 Mechatronics Implementation Considerations

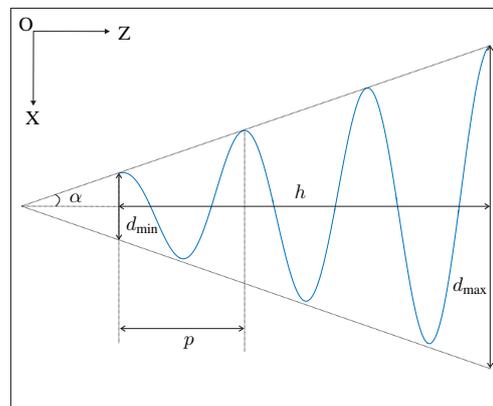


(a)

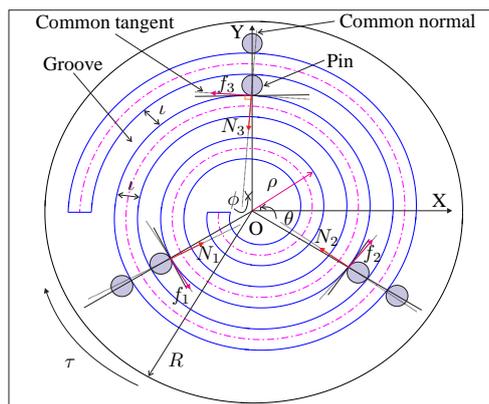
(b)



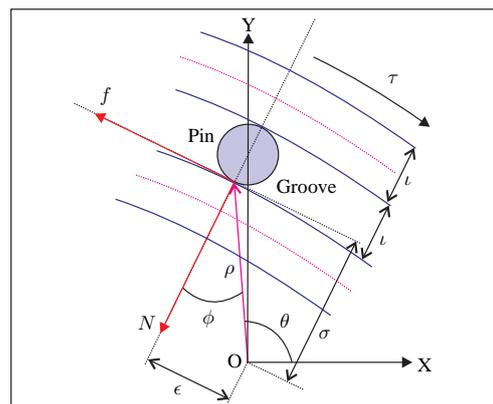
(c)



(d)



(e)



(f)

Figure 4.21: Diagrams Associated with the Spiral Cam Mechanism of the Locker. (a) Released State. (b) Engaged State. (c) 3D Tapered Helix. (d) 2D Tapered Helix. (e) Spiral Cam Mechanism. (f) Free Body Diagram of Cam.

4.3 Mechatronics Implementation Considerations

cam mechanism employed in the mechanical locker system is shown in Figure 4.21(e) where round pins of the three slider arms mate with the groove of the locker plate. The common normal direction can be depicted using angle ϕ when rotating the cam [Durali, 2015] and the following relation

$$\tan(\phi) = \frac{d\rho}{\rho d\theta} = \frac{b}{\rho} \quad (4.10)$$

always holds true [Henriksen, 1973]. Figure 4.21(f) presents the free body diagram of the cam mechanism. For simplicity, suppose normal forces $N_1 = N_2 = N_3 = N$ and frictional forces $f_1 = f_2 = f_3 = \mu_1 N$ with μ_1 being the coefficient of friction between the locker plate and the round pins. Considering the moment equilibrium equation with respect to point O, the following equation is obtained:

$$N_1\epsilon_1 + N_2\epsilon_2 + N_3\epsilon_3 + f_1\sigma_1 + f_2\sigma_2 + f_3\sigma_3 = \tau \quad (4.11)$$

which can be reformulated as

$$N(\epsilon_1 + \epsilon_2 + \epsilon_3) + \mu_1 N(\sigma_1 + \sigma_2 + \sigma_3) = \tau \quad (4.12)$$

that is equivalent to

$$N = \frac{\tau}{\epsilon_1 + \epsilon_2 + \epsilon_3 + \mu_1(\sigma_1 + \sigma_2 + \sigma_3)} \quad (4.13)$$

where $\epsilon_i = \rho_i \sin(\phi_i)$ and $\sigma_i = \rho_i \cos(\phi_i)$ with $i = 1, 2, 3$ are the distances between point O and application points of normal forces N_i and frictional forces f_i , respectively. For holding the actuator using the locker, suppose the mass of the actuator is M and the coefficient of friction between the slider arms and the acrylic tube is μ_2 . Hence, $N(\cos(\phi_1) + \cos(\phi_2) + \cos(\phi_3)) - \mu_1 N(\sin(\phi_1) + \sin(\phi_2) + \sin(\phi_3)) \geq Mg/\mu_2$ must be satisfied to achieve a locked state. By considering Equation (4.13), finally the following inequality can be attained:

$$\tau \geq \frac{Mg(\epsilon_1 + \epsilon_2 + \epsilon_3 + \mu_1(\sigma_1 + \sigma_2 + \sigma_3))}{\mu_2(c(\phi_1 + \phi_2 + \phi_3) - \mu_1 s(\phi_1 + \phi_2 + \phi_3))} \quad (4.14)$$

with $c(\phi_1 + \phi_2 + \phi_3) = \cos(\phi_1) + \cos(\phi_2) + \cos(\phi_3)$ and $s(\phi_1 + \phi_2 + \phi_3) = \sin(\phi_1) + \sin(\phi_2) + \sin(\phi_3)$.

For the implemented mechanical locker, specific values of related parameters are listed in Table 4.4. The slider arms would move a displacement of $2\pi b = 6$ mm linearly to do retraction or expansion when the locker plate rotates a full revolution. As designed,

4.3 Mechatronics Implementation Considerations

Table 4.4: Parameters Related to the Spiral Cam Locking Mechanism

Parameter	Symbol	Value
Angle of taper	α	0.1974 rad
Pitch	p	30 mm
Helix height	h	90 mm
Minimal diameter	d_{\min}	9 mm
Maximal diameter	d_{\max}	45 mm
Helix constant	a	3 mm
Helix constant	b	$3/\pi$ mm
Helix constant	c	$15/\pi$ mm
Helix variable	θ	$[\pi, 7\pi]$ rad
Module weight	M	2.5 kg
Coefficient of friction	μ_1	0.5 [Bruce, 2010, Budynas & Nisbett, 2014]
Coefficient of friction	μ_2	0.2 [Bruce, 2010, Budynas & Nisbett, 2014]

the slider arms must move at least 5.5 mm to touch the acrylic tube, which means the locker plate has to rotate clockwise by $11\pi/6$ rad. In the initial released state, values of θ of the three slider arms are $7\pi/6$, $11\pi/6$ and $15\pi/6$ rad. Therefore, in the engaged state, $\rho_1 \approx a + b(7\pi/6 + 11\pi/6) = 12$ mm, $\rho_2 \approx a + b(11\pi/6 + 11\pi/6) = 14$ mm and $\rho_3 \approx a + b(15\pi/6 + 11\pi/6) = 16$ mm. Taking into account Equation (4.10), $\phi_1 = 0.0796$ rad, $\phi_2 = 0.0682$ rad and $\phi_3 = 0.0597$ rad are obtained. By substituting these values into Equation (4.14) and considering parameters in Table 4.4, the minimum motor torque is calculated as $1.01 \text{ N} \cdot \text{m}$ and the corresponding normal force $N = 42.40 \text{ N}$. The stall torque of the selected micro DC motor is $1.57 \text{ N} \cdot \text{m}$ under 12 V. In our design, a voltage of 9 V is used to drive the motor to engage and disengage the mechanical locker and in this case its stall torque can be roughly estimated as $1.18 \text{ N} \cdot \text{m}$ which is larger than the calculated value of $1.01 \text{ N} \cdot \text{m}$ meaning that the selected motor is suitable to the application. As a summary, Table 4.5 presents specifications of the employed 775 and N20 DC motors which are respectively employed to drive the prismatic actuator and the mechanical locker. Data of the N20 motor is obtained from the vendor, while the the stall current and stall torque of the 775 motor are rated under 9 V.

4.3 Mechatronics Implementation Considerations

Table 4.5: Specifications of 775 and N20 DC Motors

DC Motor	775	N20
Overall length	98 mm	36 mm
Diameter	42 mm	12 mm
Weight	350 g	10 g
Shaft diameter	5 mm	3 mm
Gear ratio	Without gearbox	298:1
No load speed	3500 RPM	100 RPM
No load current	0.21 A	0.04 A
Stall current	8.28 A	0.6 A
Stall torque	0.225 N · m	1.57 N · m
Rated voltage	9 V	12 V
Cost	£8.68	£2.76

4.3.5 Motor Drivers

Once the motors are selected, the next step is to choose appropriate motor drivers. The range of operating voltage and the maximum output current are two important indices that need to be first evaluated. If the maximum available current of a motor driver is too small, the motor driver may be damaged due to overheat. The operating voltages of 775 DC motor, electromagnetic brake and N20 DC motor are respectively 17 V, 9 V and 9 V. The stall currents of 775 DC motor under 17 V and N20 DC motor under 12 V are separately 16.58 A and 0.6 A, and the working current of electromagnetic brake under 9 V is 0.12 A. The physical dimension of the motor driver should be small enough to save more space for the other electronics. According to the above requirements, four affordable motor drivers are considered as shown in Figure 4.22 and compared in Table 4.6. As seen from the table, L9110S and L298N are suitable for the electromagnetic brake and the N20 DC motor. Since L9110S is smaller and cheaper than L298N, L9110S is finally chosen to drive the electromagnetic brake and the mechanical locker. The continuous current of the 775 DC motor under 17 V is around 0.25 A and the MD10C motor driver can offer a continuous current up to 13 A. The maximum peak current of MD10C and the maximum output current of BTS7960

4.3 Mechatronics Implementation Considerations

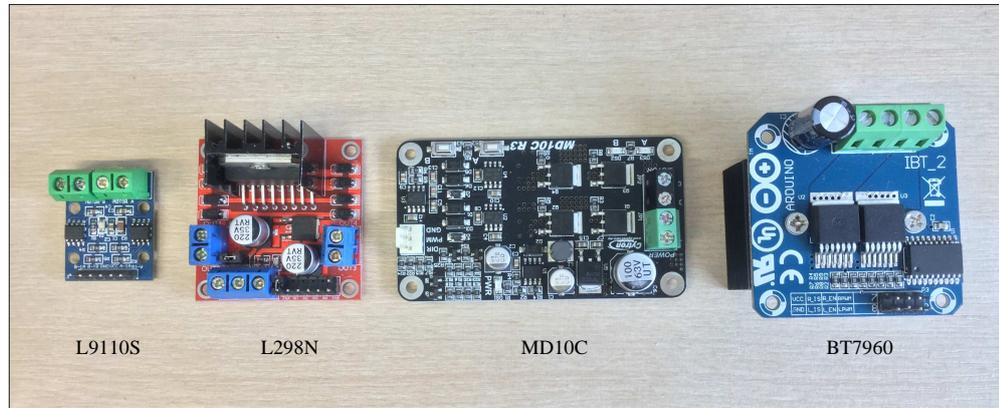


Figure 4.22: Different Motor Drivers Considered in Implementation.

Table 4.6: Comparison of L9110S, L298N, MD10C and BTS7960

Motor driver	L9110S	L298N	MD10C	BTS7960
Applicability	2 DC motors or 1 stepper	2 DC motors or 1 stepper	1 DC motor	2 DC motors or 1 stepper
Operating voltage	2.5 – 12 V	5 – 35 V	5 – 30 V	5.5 – 27 V
Maximum current	0.8 A	2 A	13 A (continuous) 30 A (peak)	43 A
Dimension	29 × 23 × 14 mm	43 × 43 × 28 mm	75 × 43 × 15 mm	50 × 50 × 42 mm
Cost	£1.75	£2.62	£11.06	£10.78

are larger than 16.58 A. Therefore, both MD10C and BTS7960 are functional for the 775 DC motor. Considering that MD10C and BTS7960 cost almost the same and BTS7960 needs more space than MD10C, MD10C is more favourable and thus is selected to drive the lead-screw transmission mechanism of the prismatic actuator.

4.3.6 Sensors

Sensors are indispensable to achieving a closed-loop control and monitoring a robotic system's states. For measuring positions of the prismatic actuator, different types

4.3 Mechatronics Implementation Considerations

of position sensors including linear potentiometers, continuous rotary potentiometers, absolute rotary encoders and incremental rotary encoders can be exploited. Both linear and rotary potentiometers can provide absolute position of the prismatic actuator even the system is powered off. However, potentiometers generally have lower precision and introduce more frictions as compared with encoders, which brings difficulties in a smooth passive motion. Like potentiometers, an absolute rotary encoder can also record the absolute position but it tends to be more expensive than an incremental rotary encoder. Based on these analyses, an incremental rotary encoder from Avago Technologies is chosen for the prismatic actuator. The resolution of the encoder is 200 CPR. Since X1, X2 or X4 decoding mode can be used, such an encoder can provide 200, 400 or 800 positions per revolution. When using X4 decoding mode for recording positions of the employed lead screw with a lead of 14 mm, the positioning precision is $14/800 = 0.0175$ mm. When using a linear potentiometer, an Arduino Nano can read analogue values that range from 0 to 1023 from the potentiometer, which results in a positioning precision of $300/1024 = 0.293$ mm that is much larger than that of the selected encoder, meaning that the employed encoder outperforms the linear potentiometer. The detailed specifications of the employed rotary encoder are listed in Table 4.7. To monitor the currents of the actuator and the locker, an ACS712T ELC-05B current sensor that can measure positive and negative 5 A current is chosen. When the actuation or the locker motor gets stuck and the current is larger than a

Table 4.7: Specifications of the Employed Encoder

Part number	HEDR-5420-ES214
Shaft diameter	5 mm
Operating voltage	-0.5 – 7 V
Resolution	200 CPR
Number of channels	2
Operating temperature	-10 – 85°C
Switching frequency	16 KHz
Maximum RPM	4800
Cost	£17.23

specific threshold for a certain period of time, the power for the MD10C or L9110S motor driver can be turned off by reading values from the current sensor.

4.3.7 Batteries, Switches and Voltage Regulator

Considering the continuous current of the 775 DC motor is rated as 0.25 A under 17 V, a high drain battery is preferable. Experiments need to be performed numerous times, therefore, a rechargeable battery is a wise choice. A cheap solution is to use the EBL 6F22 lithium-ion rechargeable batteries with a claimed voltage of 9 V and a capacity of 600 mAh. As advertised, such batteries are targeted at high drain applications and can be recharged up to 1200 times. Two such EBL batteries are connected serially with an actual voltage being around 17 V to power up the prismatic actuator with an experimentally measured maximum drawn current of 5 A. A slider switch is used for turning on or off the prismatic actuator. For achieving a pure passive state, an NPN transistor is selected to serve as a controllable switch to power off the MD10C motor driver. When considering the transistor, the maximum voltage that can be handled by the collector emitter junction V_{CEO} and the maximum DC collector current I_C that the device can withstand are two crucial factors. For the MD10C motor driver, a TIP142 NPN transistor with $V_{CEO} = 100$ V and $I_C = 10$ A is utilized. Since the recommended working voltage of the Arduino Nano board ranges from 7 to 12 V, an adjustable LM2596 step down voltage regulator is employed to output a voltage of 9 V for the Arduino Nano. Such a voltage regulator can handle an input voltage ranging from 4 V to 40 V and output a voltage between 1.5 V and 34 V with a conversion efficiency of 92%. More importantly, this module has a thermal shut-down and current-limit protection, which can deal with a maximum current of 3 A.

4.3.8 Communication Module

There exist numerous low-cost solutions to realizing the communication between different modules. Table 4.8 lists three different boards that are based on different technologies, more specifically, RF, Wi-Fi and Bluetooth. As summarized in the table, HC-05 Bluetooth module has the shortest range but it is the most expensive one. The ESP8266 Wi-Fi module has the fastest transmitting speed, however, it consumes more power to transmit. Since the performance of these solutions are similar and there exists a big community that can offer open-source materials for using nRF24L01+ modules,

Table 4.8: Comparison of nRF24L01+, ESP8266 and HC-05 Wireless Modules

Module	nRF24L01+	ESP8266	HC-05
Technology	RF	Wi-Fi	Bluetooth
Frequency	2.4 GHz	2.4 GHz	2.4 GHz
Modulation	GFSK	Inapplicable	GFSK
Range	100 m	100 m	10 m
Operating voltage	1.9 – 3.6 V	3.0 – 3.6 V	3.6 – 6 V
Maximum speed	2 Mbps	72.2 Mbps	2 Mbps
Transmit power	4 dBm	19.5 dBm	4 dBm
Communication with MCU	SPI	Serial, SPI, I2C	Serial
Size	29 × 15 × 1.2 mm	25 × 15 × 1.2 mm	28 × 15 × 2.35 mm
Cost	£1.99	£3.99	£5.49

nRF24L01+ is selected to form a communication network as an ultra low power and cheap solution.

4.4 Conclusions

This chapter focused on the mechatronics design and hardware implementation of a strut-type modular robotic system. Four conceptual design schemes of a three-state prismatic actuator have been presented, compared and discussed. Physical prototypes of two different prismatic actuators have been implemented and introduced in terms of fabrication of mechanical parts and implementation of electronics circuits. Other robotic elements including a cubic node with rigid connectors and a passive revolute joint have been designed and built. More modular units have been replicated for constructing modular robotic structures leading to a system with eight prismatic actuators, four rigid connector nodes and four passive revolute joints. Implementation process associated with components selection has been detailed as well. Actuation and locking forces have been analysed mathematically for sizing DC motors of the prismatic actuators and the designed locking mechanism. Based on the implemented modular elements, experiments are then performed in the upcoming chapters.

CHAPTER 5

Experimental Validation of Prismatic Actuators

This chapter experimentally verifies the functionalities and performance of the fabricated prismatic actuators with their communication, three states and state transitions covered. An overview of the experiments is first given. Then, the experimental setups are introduced. Experimental results are finally presented and discussed.

5.1 Overview

In the preceding chapter, a prismatic actuator based on an electromagnetic brake (Prototype I) and a prismatic actuator equipped with a designed mechanical locker (Prototype II) are detailed. In this chapter, experiments are conducted based on the built prototypes ranging from an individual prismatic actuator to systems with multiple prismatic actuators.

The actuation, locking and passive compliance performance of individual prismatic actuators is first investigated. For the actuated state, the positioning accuracy and trajectory-tracking performance of the prismatic actuator are studied. For the locked state, locking forces of the electromagnetic brake and the mechanical locker are measured using experimental devices. For the passive state, the external force required to let the prismatic actuator move passively is experimentally estimated.

The above experimental results lay a foundation to implement the self-contained prismatic actuator, i.e., Prototype II. Three states and state transitions of the self-contained prismatic actuator are then investigated. To test the actuation force, external loads are applied to the prismatic actuator to check how much weight can be vertically lifted by the actuator. In the locked and passive states, external loads are also used to see how much load the actuator can sustain. State transitions of the actuator are finally achieved to prove the feasibility of the conceptual design.

Finally, communication and state transitions capabilities of multiple prismatic actuators are studied. The prismatic actuators communicate with each other and change states according to the received messages.

5.2 Experimental Setups

Experimentally, the performance of the fabricated prototypes in the actuated, locked and passive states are validated. The actuation and locking forces of the prototypes are evaluated. The minimum force required to passively move the prismatic actuators

is also estimated. Correspondingly, various experimental setups are used to conduct the experiments.

5.2.1 Actuated State

Experiments are performed based on Prototype I shown in Figure 5.1. The actual position of the prismatic actuator is measured by a rotary encoder and the position sensor data are collected through serial communication between an Arduino Uno board and a laptop. A laser light sensor is used to measure the displacement externally. Here, the laser light sensor reading works as a reference to check the feasibility of the employed rotary encoder.

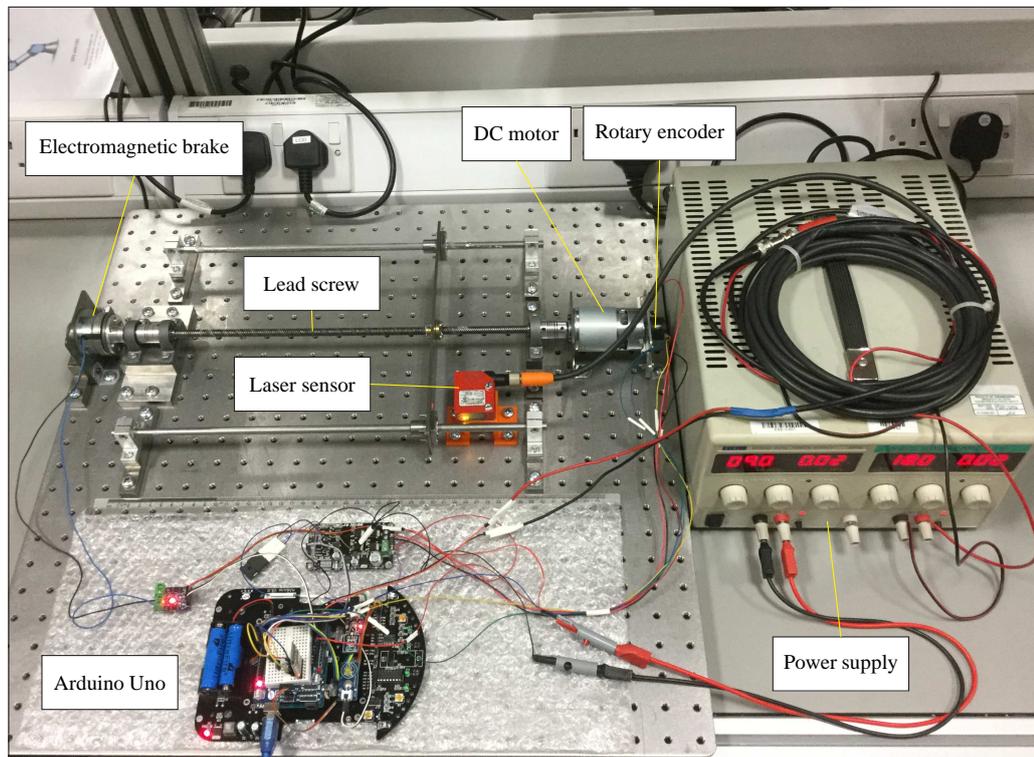


Figure 5.1: Experimental Setup for Verifying the Performance of the Actuated State.

In order for actuating the prismatic actuator to move to a given position and follow a predetermined trajectory, a closed-loop PID controller is developed. PID controllers have been widely and maturely applied to more than 95% of the industrial automation applications [Åström & Hägglund, 1995, Åström & Murray, 2010]. The core idea of a

PID controller is to continuously calculate an error value defined as $e(t) = d(t) - s(t)$ with $d(t)$ and $s(t)$ respectively denoting the desired setpoint signal and the sensory output signal, and apply proportional, integral and derivative terms as a correction to a system. Mathematically, a PID controller can be depicted as

$$u(t) = k_p e(t) + k_i \int_0^t e(t) dt + k_d \frac{de(t)}{dt}, \quad (5.1)$$

where non-negative coefficients k_p , k_i and k_d represent the proportional, integral and derivative gains, respectively. Figure 5.2 presents a schematic diagram of a classic PID controller for better understanding. Different combinations of k_p , k_i and k_d can lead to different PID responses, i.e., different resultant sensory signals of the controlled system. Basically, a PID step-response curve has five meaningful indices (i.e., rising time, overshoot, settling time, steady-state error and stability) that are closely associated with the controller performance [Åström & Hägglund, 1995]. The relationship between PID gains and the five performance indices is summarized in Table 5.1 [Song, 2014]. Note that PID gains are dependent on each other and changing one of them can have

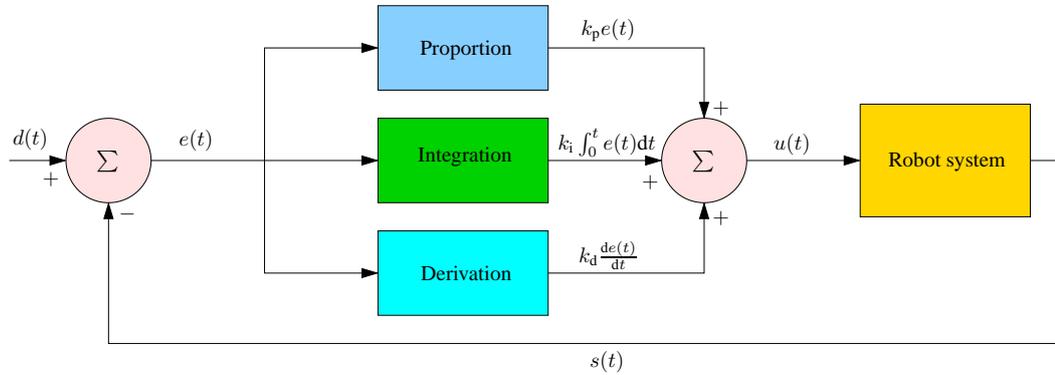


Figure 5.2: Schematic Diagram of a Classic PID Controller.

Table 5.1: Effects of k_p , k_i and k_d on Performance Indices of a PID Response

Gain	Rising time	Overshoot	Settling time	Steady-state error	Stability
$k_p \uparrow$	↓	↑	Small change	↓	↓
$k_i \uparrow$	↓	↑	↑	Eliminate	↓
$k_d \uparrow$	Small change	↓	↓	No change	↑

impact on the other two. Actually, PID gains k_p , k_i and k_d are extremely important for stably and robustly controlling a robot system, and their optimal values must be elaborately determined. There are many methods including the famous Ziegler-Nichols method and empirical method for manually tuning the values of k_p , k_i and k_d based on the step response of a controlled system using Table 5.1 as a reference [Cominos & Munro, 2002].

For the developed prismatic actuator shown in Figure 5.1, a closed-loop system is formed by the 775 DC motor, the rotary encoder, the lead screw and a given setpoint signal. Specifically speaking, the PID controller first calculates the difference between the setpoint signal and the encoder reading. After that the proportional, integral and derivative terms are calculated and combined to generate a control signal (i.e., a PWM signal) which is applied to the system. The lead-screw system takes actions according to the given PWM value and outputs its position information in the form of an encoder reading. By iteratively following the described same procedure, the prismatic actuator can keep still or follow a specific setpoint signal.

5.2.2 Locked State

In the preceding chapter, two prototypes of three-state prismatic actuators are detailed. One of the prototype uses electromagnetic brake to achieve the locked state, while the other one employs a designed mechanical locker to achieve the same purpose. Figure 5.3 shows the setup for testing the electromagnetic brake. The brake is powered by a power supply. Slotted masses and a weight hanger are used to check how much weight an engaged brake can hold under a given voltage. Experimental setup for measuring force of the designed mechanical locker is presented in Figure 5.4. A force sensor is utilized to measure the locking force. Before using the force sensor to evaluate the force of the locker, the force sensor must be calibrated. The calibration and measurement setups are respectively presented in Figure 5.4(a) and (b). Figure 5.4(a) shows the setup including a laptop, a calibration rig, some slotted masses and a circuit for calibrating the force sensor. The schematic diagram of the prototyped circuit is shown in Figure 5.5. The circuit consists of an Arduino Nano microcontroller for obtaining sensory readings, an L9110S motor driver for driving the locker, a TC1044S voltage regulator and an MCP6002 operational amplifier. Besides, an IR receiver is used to do remote control for changing some experimental conditions and an LED light is employed to

5.2 Experimental Setups

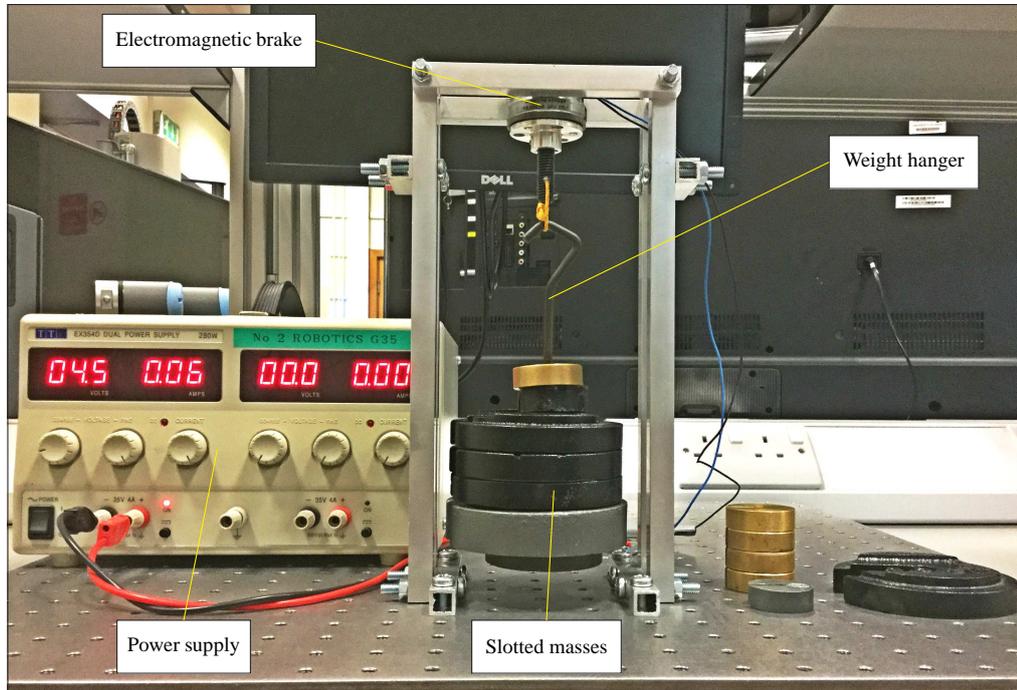


Figure 5.3: Experimental Setup for Testing the Electromagnetic Brake.

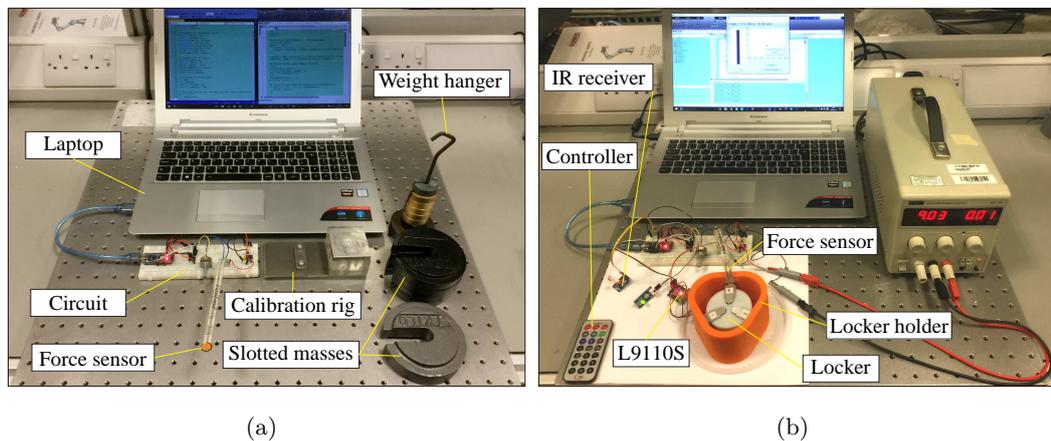


Figure 5.4: Experimental Setup for Measuring Force of the Mechanical Locker. (a) Calibration Setup. (b) Measurement Setup.

indicate that the IR message is received by the microcontroller or not. After finishing the calibration, the locking force is measured using the setup presented in Figure 5.4(b).

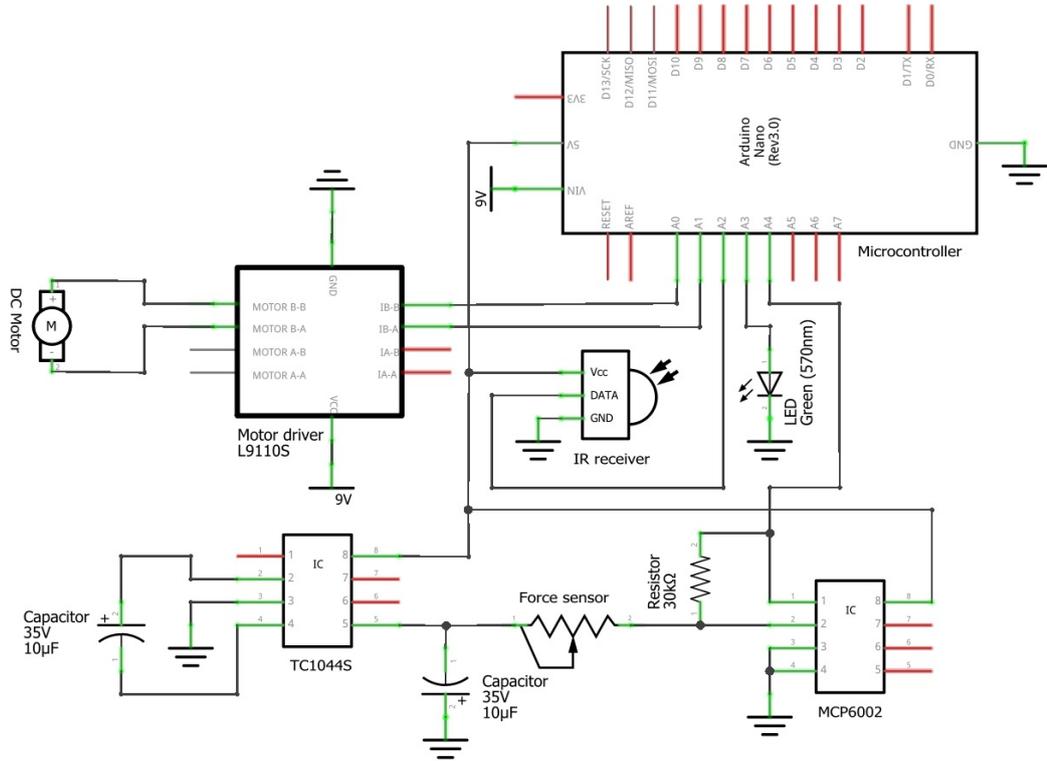


Figure 5.5: Schematic Diagram of a Circuit for Measuring Forces of the Locker.

The locker is put inside a locker holder and the force sensor is glued to the inner wall of the locker holder. Note that the force sensor is fixed so that the slider arms of the locker presses the force sensor when the locker is engaged. During the calibration and measuring processes, the laptop communicates with the Arduino Nano board through a serial cable. Codes for the Arduino Nano are written in C++ while for the laptop side, Matlab codes are written to collect the sensory data and plot the obtained results. The related codes are given in Appendix D.1.

5.2.3 Passive State

The required external force to make the prismatic-actuator system move passively is experimentally measured as it is difficult to estimate the friction torques of the system. Four different lead screws with different leads and/or different lengths are tested. Figure 5.6(a) shows four lead screws of which lead screws 1–3 have the same length of 400

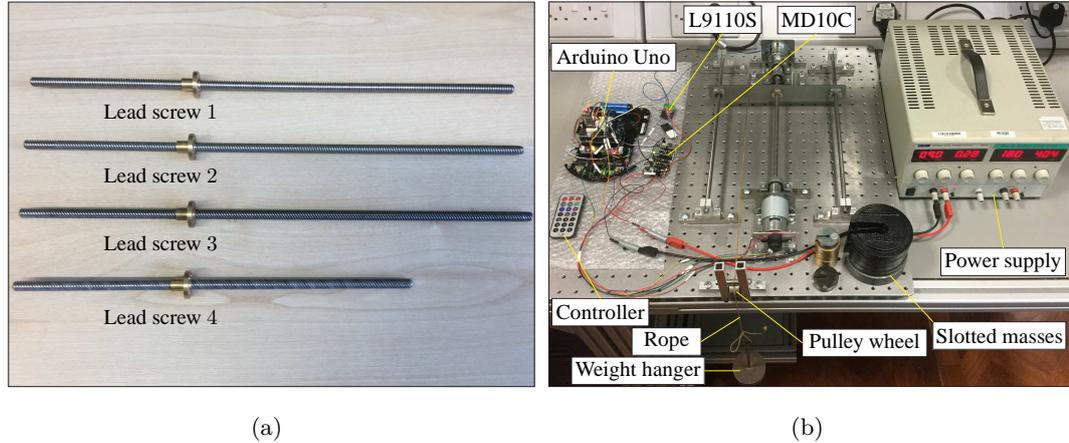


Figure 5.6: Experimental Setup for Testing the Passive State. (a) Lead Screws. (b) Experimental Setup with a Pulley System.

mm. The leads of lead screws 1–3 are different, which are 2 mm, 8 mm and 14 mm, respectively. Lead screw 4 is shorter than the other lead screws and it has a length of 300 mm and a lead of 14 mm. As shown in Figure 5.6(b), a pulley system consisting of a pulley wheel, a rope, a weight hanger and some slotted masses is built. The slotted masses can be placed on the weight hanger to evaluate the minimum force required to passively move the prismatic-actuator system.

5.2.4 Three States of an Individual Prismatic Actuator

The three states and state transitions of the mechanical locker based Prototype II are demonstrated using the experimental setup shown in Figure 5.7. The experimental setup includes two prototyped prismatic actuators, a laptop, an Arduino Uno board, a power supply and some slotted masses. Slotted masses can be placed on the top of a prismatic actuator to check the actuation, locking and passive compliance performance. The prismatic actuators can be controlled remotely through the Arduino Uno board. A human operator can send commands to the Arduino Uno board through IR communication. After receiving the commands, the Arduino Uno board transmits the message to the prismatic actuators through RF communication realized by the nRF24L01+ modules. This way can overcome the light-of-sight problem of IR communication. Experimental data are collected through a serial cable between the Arduino

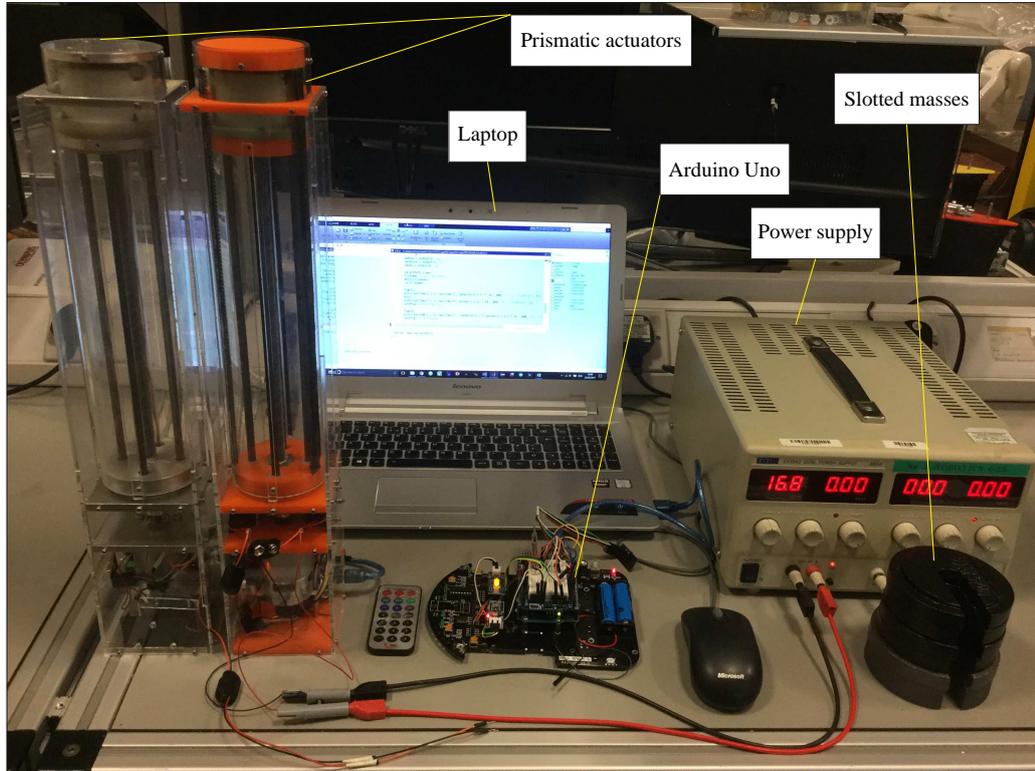


Figure 5.7: Experimental Setup for Testing Three States and State Transitions of Individual Prismatic Actuators.

Nano board and the laptop.

5.2.5 Communication and State Transitions of Multiple Actuators

Communication and state transitions performance of multiple actuators are verified. Two separate actuators and parallel structures consisting of multiple actuators are built to perform the experiments. Class diagrams of the developed software for these physical experiments are presented in Appendix D.2.

Separate Prismatic Actuators

Two separate actuators with one of their ends are rigidly attached to a fixed table as shown in Figure 5.8. The table is higher than a fully retracted actuator so that the actuators can be operated against gravity. Each actuator has its own nRF24L01+ module to perform RF communication. To achieve remote control, an Uno node with an

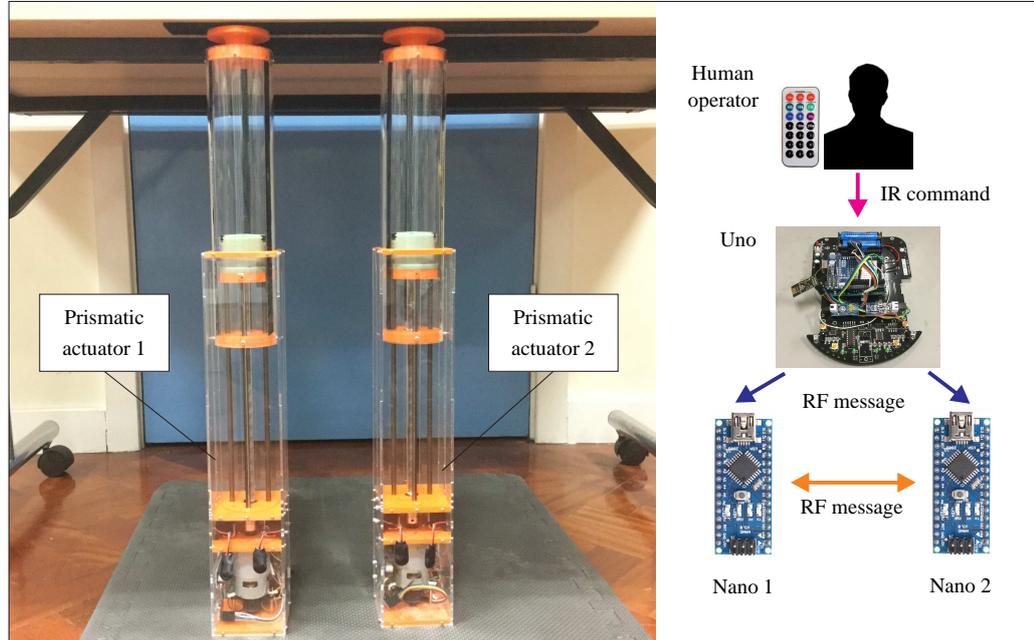


Figure 5.8: Experimental Setup for Testing the Three States and State Transitions.

IR receiver and an nRF24L01+ module is used to receive and then transmit commands sent by a human operator with an IR controller. Actuators 1 and 2 are respectively initialized as actuated and passive when they receive remote control commands to start working. For the active actuator, it lifts itself up as the actuation force supplied by the DC motor is capable of overcoming its own weight. Once the destination setpoint is achieved, the active actuator changes its state from actuated to locked by engaging its locking mechanism and turning off the power for the MD10C motor driver. Then, the active actuator sends a message to the passive actuator which would change its state to be active and send a feedback message out after receiving the message. When the feedback message is received, the previous active actuator converts its locked state into passive. In this way, the two actuators change states periodically and alternatively.

Parallel Structures with Multiple Prismatic Actuators

This section investigates and demonstrates state transitions of parallel prismatic actuators. Three parallel structures with two, three and four actuators are studied. Ends of the parallel actuators are jointed together so that all the actuator motions are physically synchronised. Figure 5.9(a) shows two parallel prismatic actuators of which the

5.2 Experimental Setups

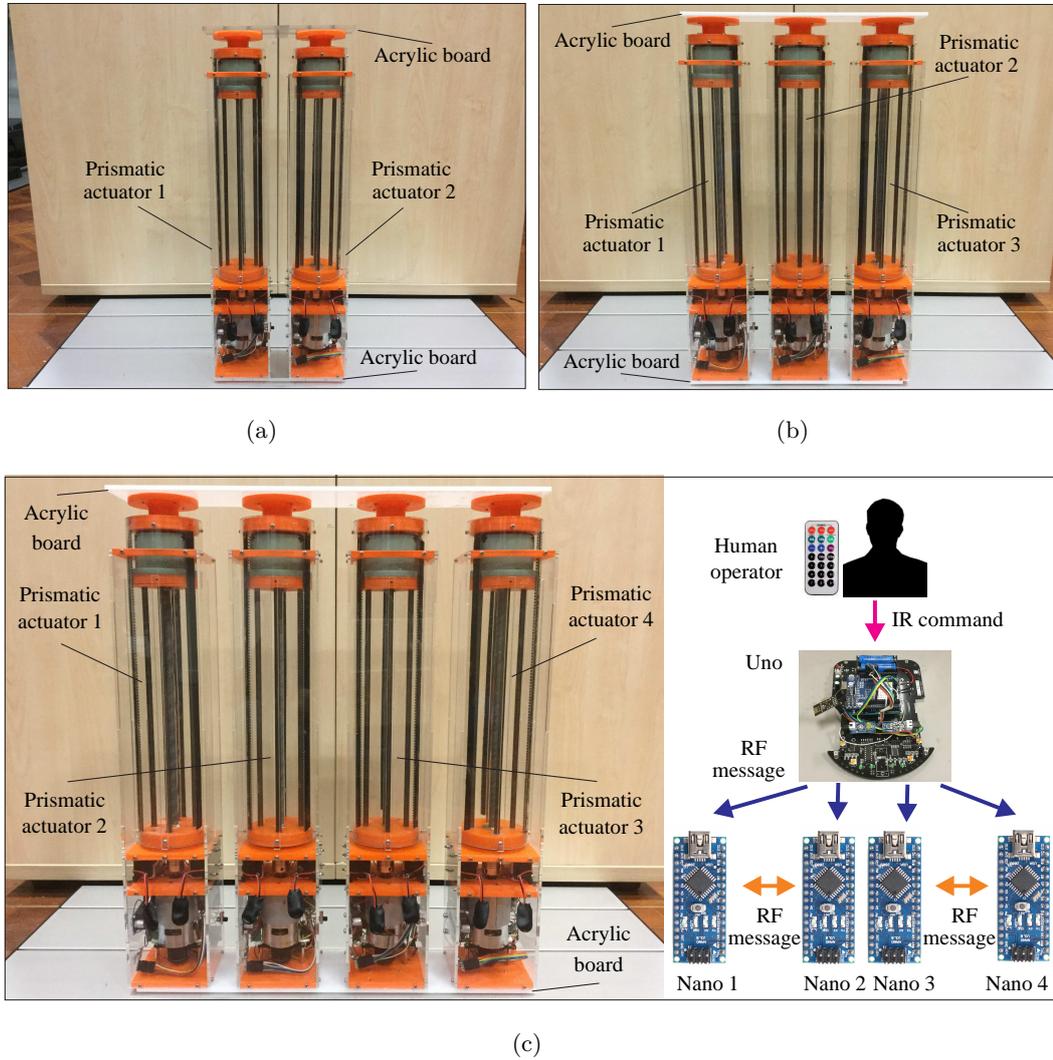


Figure 5.9: Parallel Structures with Multiple Prismatic Actuators. (a) Two Parallel Actuators. (b) Three Parallel Actuators. (c) Four Parallel Actuators.

two ends are connected rigidly using thick acrylic boards. For conventional prismatic actuators, such a parallel structure can move only when both of the two actuators are active at the same time. Owing to the passive state, the structure shown in Figure 5.9(a) can move by actuating one of the actuators and leaving the other one passive. Meanwhile, since the two actuators are identical and can communicate with each other, states of the two actuators can be exchanged periodically according to their received

messages. More specifically, the initial states of prismatic actuators 1 and 2 are first set as actuated and passive, respectively. Once a command is sent through the central Uno node to inform the prismatic actuators to get started, the parallel structure moves vertically due to the actuation of one active actuator. The other inactive actuator moves passively since it is capable of responding to external forces. After a full cycle of extension and contraction of the two prismatic actuators, the active actuator sends a message to the inactive actuator. When receiving the message, the inactive actuator converts its passive state into actuated state and sends a feedback message to the other actuator. The previous active actuator changes its state to be passive once it receives the feedback message. The above described state transitions and communication processes are periodical and the parallel structure keeps completing reciprocal motions in this manner.

Three prismatic actuators are connected by acrylic boards in parallel as presented in Figure 5.9(b) to further investigate the developed prototypes. Initially, the middle prismatic actuator 2 is set as actuated while the other two treated as a group are set as passive. Like the previous case, the prismatic actuator 2 communicates with the other two actuators and then changes its state. In this case, the middle actuator performs communication with two actuators that also change their states according to their received messages.

Finally, a parallel structure consisting of four prismatic actuators connected by acrylic boards is shown in Figure 5.9(c). The four actuators are divided into two groups. Actuators 1 and 3 are paired with actuators 2 and 4 to perform mutual communication, respectively. The communications between the paired actuators are independent on the Uno node. Hence, the established communication network shown in the right side of Figure 5.9(c) is distributed. Initially, actuators 1 and 3 are set as actuated while the other two actuators are passive. Similarly, the two groups of the four parallel prismatic actuators can exchange states periodically.

5.3 Experimental Results

This section presents and discusses the experimental results. The results related to the actuated, locked and passive states are first given and discussed. Then, three states and state transitions testing results of individual prismatic actuators are presented. More results are shown to verify the communication and state-transition capabilities of

multiple prismatic actuators.

5.3.1 Actuated State

The step response of the prismatic actuator is investigated and then trajectory-tracking tasks are assigned to further demonstrate the performance of the prismatic actuator and the PID controller. In this experiment, the PID gains k_p , k_i and k_d are separately empirically tuned as 0.23, 0.00035 and 0.00003. For the step response, the setpoint is set as 140 mm which corresponds to 8000 encoder ticks. The experiment is conducted repeatedly for ten times. Figure 5.10 presents the step responses of the prismatic actuator controlled by the PID controller. The figure shows that the prismatic actuator can stably achieve a specified displacement of 140 mm in all of the ten trials, even though the time spent to achieve the destination is different. This may result from the vibration of the system which leads to different initial conditions of the ten trials. The profiles of the rotary encoder and laser light sensor coincide quite well, which verifies

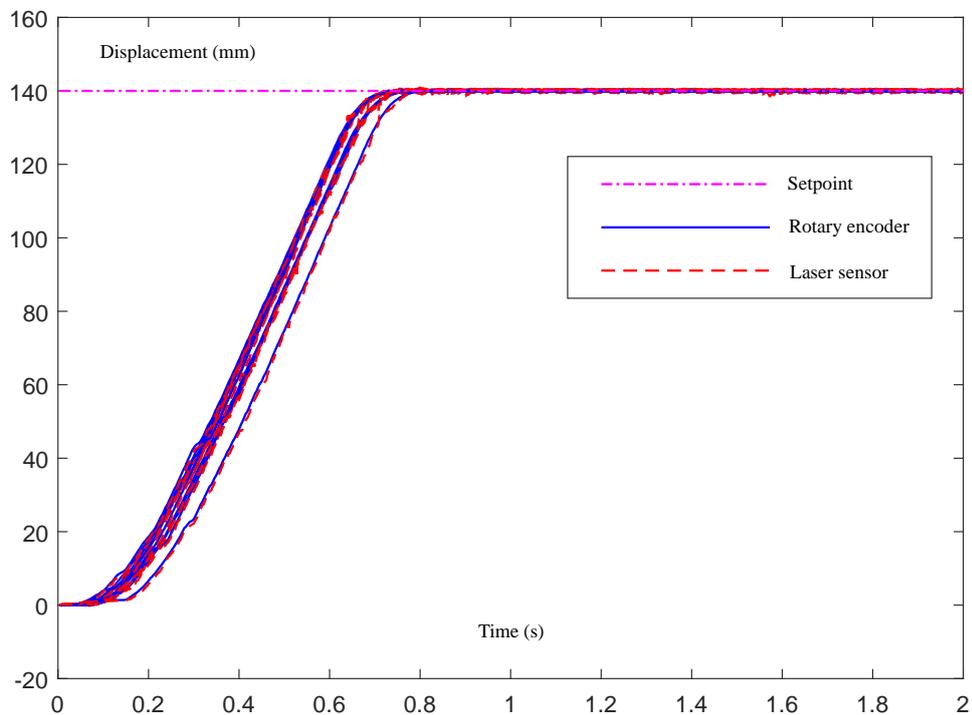


Figure 5.10: Step Responses of the Prismatic Actuator Controlled by a PID Controller.

5.3 Experimental Results

the efficacy of the prismatic actuator and the PID controller.

Then, the prismatic actuator is assigned to track a sinusoidal-like trajectory. The sinusoidal-like trajectory is used to investigate coordination and locomotion of modular robots in the upcoming chapter. Figure 5.11 shows the corresponding experimental results. As can be observed from Figure 5.11(a) and (b), the actual position recorded by the encoder coincides with the setpoint position with the maximum absolute value of the displacement error being less than 0.2 mm. Figure 5.11(c) presents positions recorded by the rotary encoder and the laser sensor showing that they coincide well

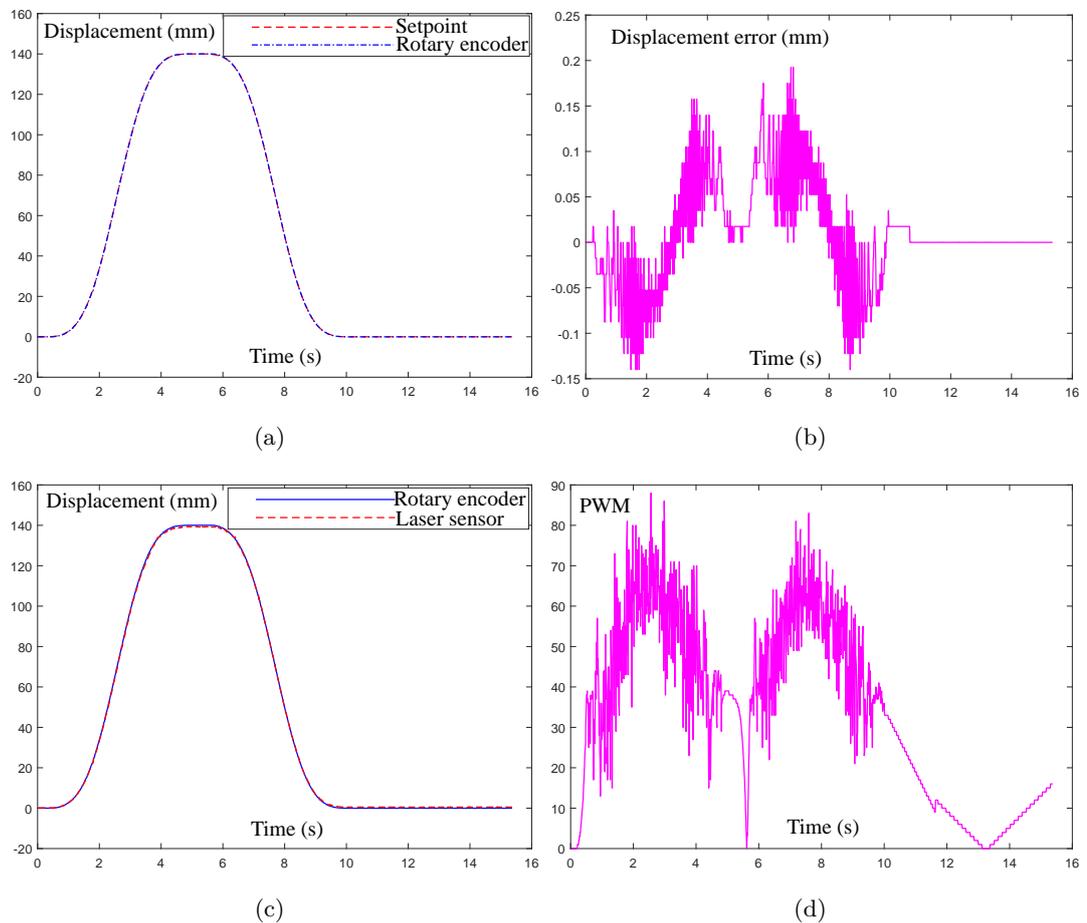


Figure 5.11: Sinusoidal-Like Trajectory Tracking using the Prismatic Actuator. (a) Motion Profiles. (b) Displacement Error. (c) Verification with Laser Sensor. (d) PWM Values.

with each other. The PWM values generated by the PID controller are plotted in Figure 5.11(d). Note that noises are inevitable when collecting data through serial communication. This may be the reason why there exist some abrupt changes in the profiles of the displacement error and the PWM values.

To substantiate the repeatability of the trajectory-tracking performance, an experiment is conducted to let the prismatic actuator follow a periodical sinusoidal profile with an amplitude of 210 mm for 20 cycles. The resultant motion and displacement-error profiles are shown in Figure 5.12. As can be seen from the figure, the actual position can coincide with the desired position with the maximum absolute value of the position error being less than 0.2 mm. This further demonstrates the repeatability, feasibility and positioning accuracy of the prismatic actuator in the actuated state.

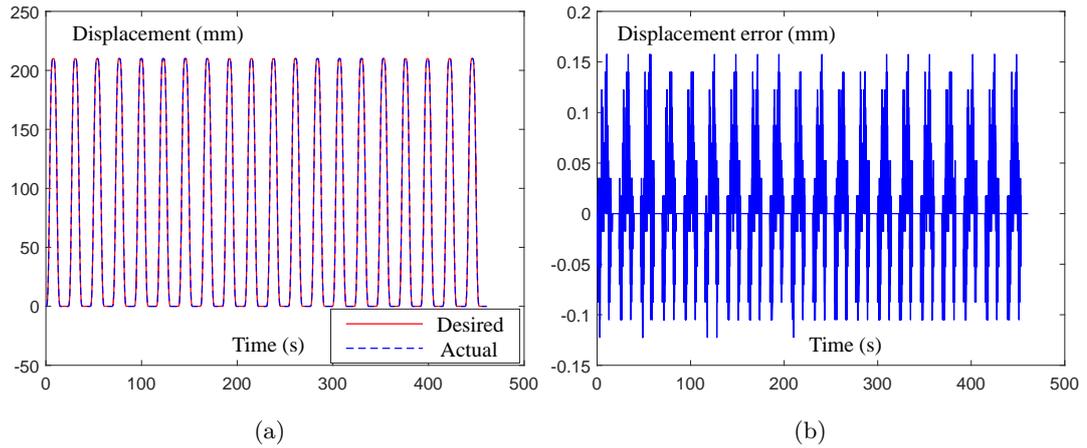


Figure 5.12: Periodical Sinusoidal-Like Trajectory Tracking using the Prismatic Actuator. (a) Motion Profiles. (b) Displacement Error.

5.3.2 Locked State

The electromagnetic brake is first experimentally tested. A series of voltages from 2.0 to 4.5 V in steps of 0.5 V are applied to the brake to find the maximum weights that the brake can hold under different voltages. For each case with a certain voltage, experiments are repeated for three times. The total obtained results are presented in Appendix C.2. Figure 5.13(a) shows the relation between the measured average weights and the applied voltages. In the figure, the solid line with star makers is the measured

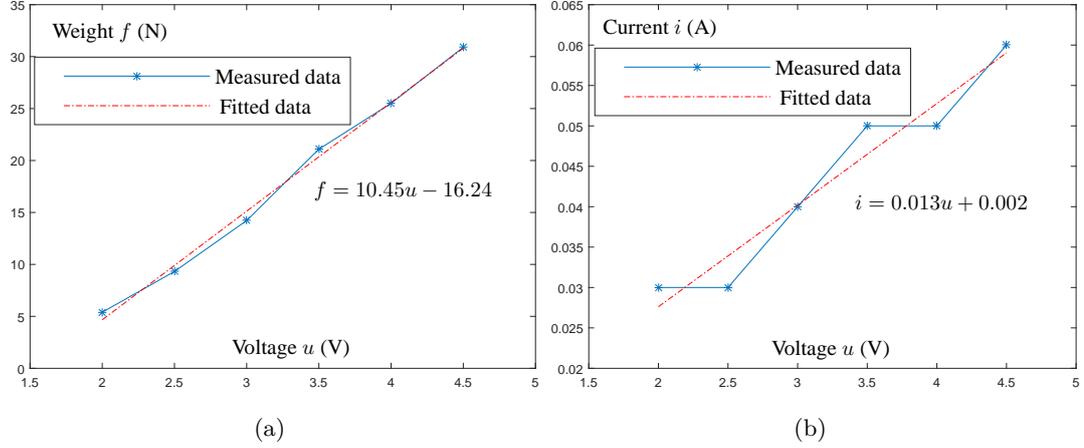


Figure 5.13: Experimental Test Results of the Electromagnetic Brake. (a) Average Weight Versus Voltage. (b) Average Current Versus Voltage.

average values and the dash-dot line is an approximating linear relationship between the applied voltages and the maximum weights. The employed brake can support a load with a maximum weight of 30.90 N under 4.5 V, which is enough to sustain one prismatic actuator with a mass of 2.2 kg or 1.8 kg. When using a voltage of 9 V, the maximum weight that the brake can deal with can be estimated as 77.81 N using the fitted relation. During the locked state, the current drawn by the electromagnetic brake under different voltages is shown in Figure 5.13(b). Using the fitted relation, the current drawn by the brake is estimated as 0.12 A under 9 V. Such a current is much smaller than the maximum drive current of the L9110S motor driver.

The locking force of the mechanical locker is measured using a force sensor. The force sensor is first calibrated using a calibration rig and some slotted masses. The calibration process is presented in Figure 5.14(a). The force sensor is put inside the calibration rig and different weights are placed on the top of the rig. Then, the Arduino reads data output by the force sensor. To attain the relation between the weights and sensing readings, different masses between 0.5 and 3.0 kg are tested. It is worth pointing out that the adopted reading for each trial is an average of 100 sensing values and 25 groups of experiments are repeatedly conducted. The calibration data plotted in Figure 5.14(b) is an average result. More calibration data can be seen from Appendix C.3. Figure 5.14(c) shows a histogram of 100 sensing values in a trial when placing a

5.3 Experimental Results

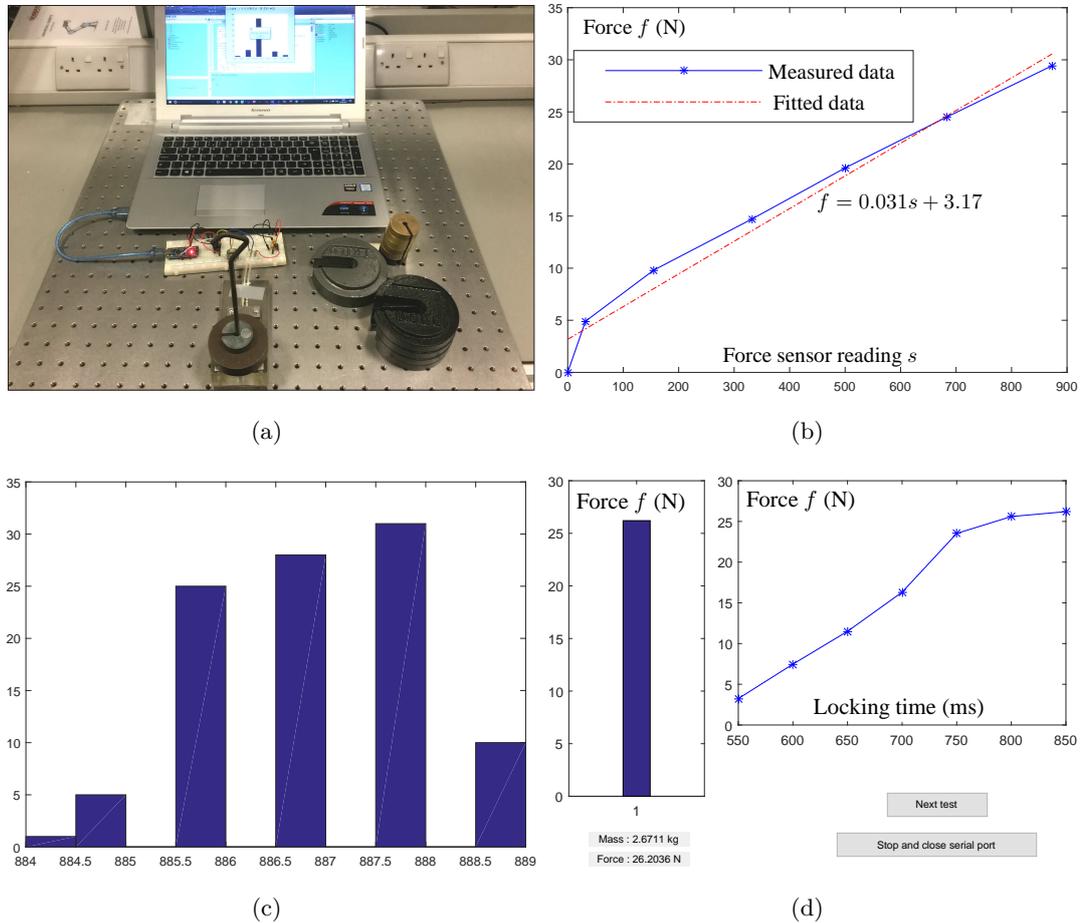


Figure 5.14: Force Measurement of the Mechanical Locker. (a) Calibration Process. (c) Calibration Result. (d) Histogram of One Trial. (d) Measured Result.

mass of 3.0 kg on the rig. After obtaining the calibration result, the locking force of the mechanical locker is measured. Locking forces are evaluated by setting different extension and contraction time of the locker. Specifically, experiments with time $t \in [550, 850]$ ms in steps of 50 ms are performed. The maximum time is set as 850 ms since the motor gears are broken in this case. 10 groups of experiments are repeated to make the result relatively accurate. The final average result is shown in Figure 5.14(d) and for more results please refer to Appendix C.4. As shown in the figure, the maximum force is around 26.2 N and the forces corresponding to 800 ms and 850 ms are almost the same indicating that the gears cannot stand with more forces. Evidently, the resultant

value of 26.2 N is smaller than 42.4 N calculated in the previous chapter. One of the reasons of this fact is that the motor gears are broken resulting in a smaller force that can be measured. Another reason is that frictions are not considered in the theoretical analysis side. Since the locker has three sliders, the total locking force can be estimated as 78.6 N. The coefficient of friction between the slider arms and the acrylic tube is a crucial factor relevant to the amount of weight that can be sustained by the mechanical locker. The performance of the mechanical locker is experimentally tested. Figure 5.15 presents the experimental setup and result showing that the mechanical locker can support a weight of 3.0 kg, which is more than the weight of the designed Aluminium or PLA version prismatic actuator.

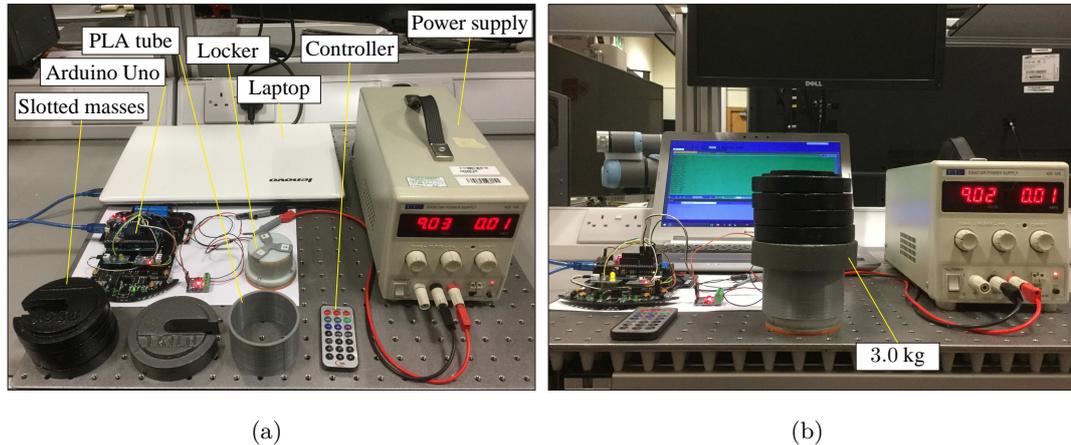


Figure 5.15: Experimental Test of the Mechanical Locker. (a) Experimental Setup. (b) Weight Versus Voltage.

5.3.3 Passive State

Using the pulley system, external forces required to move the prismatic-actuator system are evaluated. Different cases where lead screws without coupled components, lead screws connected with the motor and lead screws coupled with the motor and the brake are considered. Table 5.2 presents the average experimental results of five trials. More data can be referred to Appendix C.5. Without connection with the motor and brake, lead screw 1 with a lead of 2 mm cannot back drive even a weight of 39.24 N is applied. Lead screw 2 with a lead of 8 mm can improve the back-drive capability but it

Table 5.2: Forces Required to Move Different Lead Screws in the Passive State

Only lead screw	Lead screw	1	2	3	4
	Length (mm)	400	400	400	300
	Lead (mm)	2	8	14	14
	Weight (N)	> 39.24	10.30	3.92	3.53
Lead screw with motor	Lead screw	3		4	
	Weight (N)	9.03		8.34	
Lead screw with motor and brake	Lead screw	3		4	
	Weight (N)	11.34		10.50	

needs a much larger force as compared with a lead of 14 mm. Hence, lead screws 1 and 2 are discarded while lead screws 3 and 4 coupled with motor and/or brake are further tested. Evidently, a bigger lead and a shorter length can lead to a stronger back-drive ability of a lead screw. According to these results, the 300 mm long lead screw 4 with a lead of 14 mm is finally selected as it can move passively under an applied weight of only 10.50 N when coupled with the motor and the brake.

5.3.4 Three States of an Individual Prismatic Actuator

The three states of the aluminium version and PLA version prismatic actuators are studied in this subsection. The two versions of prismatic actuators are different in terms of the employed materials and the power source. The aluminium version is powered by a 9 V rechargeable battery, while the PLA version is enhanced with two 9 V rechargeable batteries. As measured, each employed battery has an actual voltage of around 8.4 V. In the experiments, to avoid the damage of the batteries, the batteries are replaced by a power supply that can offer 8.4 V and 16.8 V. The actuation capability of the aluminium version prismatic actuator is tested under a voltage of 8.4 V using the slotted masses. Figure 5.16(a) demonstrates that the aluminium prismatic actuator can lift an external load of 2.0 kg but failed in the case of 2.5 kg. Since the aluminium prismatic actuator has a mass of 2.2 kg, it is hard to lift another aluminium prismatic actuator. In contrast, the PLA version actuator powered by a voltage of 16.8 V can lift an external load of 3.0 kg but failed to lift a load of 3.5 kg as shown in Figure 5.16(b). An external load of

5.3 Experimental Results

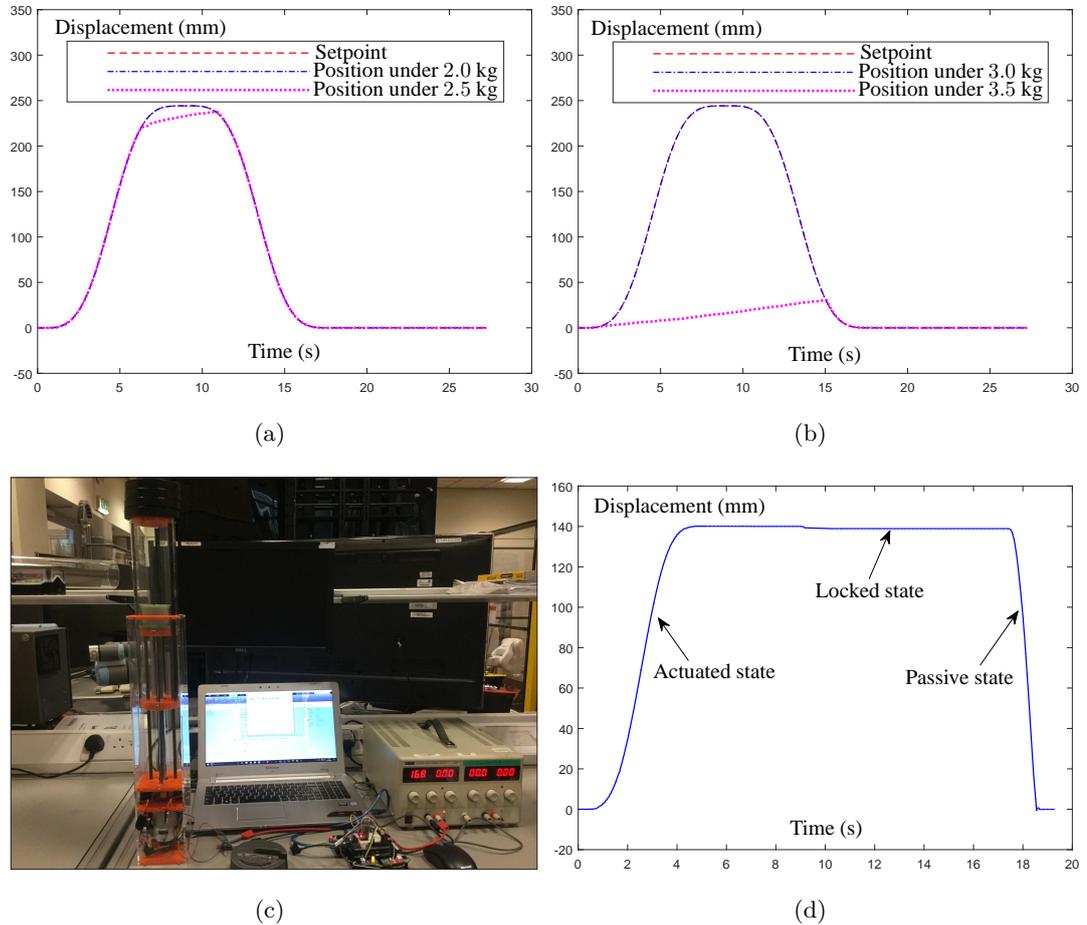


Figure 5.16: Three States of Prototype II. (a) Actuated State of Aluminium Version Prototype Under Loads of 2.0 and 2.5 kg. (b) Actuated State of PLA Version Prototype Under Loads of 3.0 and 3.5 kg. (c) Locked State Under a Load of 2.0 kg. (d) State Transitions Under a Load of 1.0 kg.

3.0 kg is larger than the weight of a PLA version prismatic actuator with a mass of 1.8 kg, meaning that the prismatic actuator can lift another prismatic actuator vertically. These results show that the actuation force is enhanced by using a higher voltage and lighter materials. Velcro tapes are employed to increase the friction between the acrylic tube and the slider arms. Figure 5.16(c) presents the locking capability of the prismatic actuator indicating that the locker can hold an external load of 2.0 kg. It is enough to keep locked and overcome its own gravity in vertical applications. The result of state

transitions between actuated, locked and passive states is shown in Figure 5.16(d). As seen from the figure, the designed and implemented prismatic actuator can achieve the state transitions. Specifically, the actuator is driven by the motor in the actuated state to lift the external load. Then, the locking mechanism is engaged and the power for the MD10C motor driver is turned off in the locked state. Finally, the locking mechanism is released and the actuator moves passively due to the external load.

5.3.5 Communication and State Transitions of Multiple Actuators

This subsection experimentally verifies the communication and the state-changing functionalities of multiple prismatic actuators. Based on RF communication, two separate actuators alternatively and cyclically change their states are first demonstrated. Then, parallel structures composed of two, three and four actuators are studied to show a case where parallel structures can be moved with one or two actuators being passive.

Separate Prismatic Actuators

Communication between the two prismatic actuators is first tested. A human operator sends commands to the Arduino Uno board to start or stop the communication. These command messages are sent and received by an IR controller and an IR receiver, respectively. The Arduino Uno board then forward the command messages to the two prismatic actuators through nRF24L01+ RF modules. The communication between the two actuators is repeatedly conducted for five times. For each time, prismatic actuator 1 sends a message to prismatic actuator 2 every 0.1 s. The communication process is started or stopped by the human operator using the IR controller. Test results of one trial is presented in Figure 5.17. As can be seen from Figure 5.17(a), prismatic actuator 1 tried to deliver 5056 messages, received 5015 feedback messages from prismatic actuator 2, and failed to deliver 1 message. As shown in Figure 5.17(b), prismatic actuator 2 received 5015 messages. This means that $5015/5056 = 99.19\%$ of the messages are successfully exchanged between the two actuators. Prismatic actuator 1 failed to deliver 1 message and managed to deliver 40 messages. However, these 40 messages are not received by prismatic actuator 2. The Uno board outputs information to inform the human operator that the communication is started or stopped as shown in Figure 5.17(c). In the other four trials, similar results are obtained. The whole results of the five trials are presented in Appendix C.6. These results substantiate the efficacy of the

5.3 Experimental Results

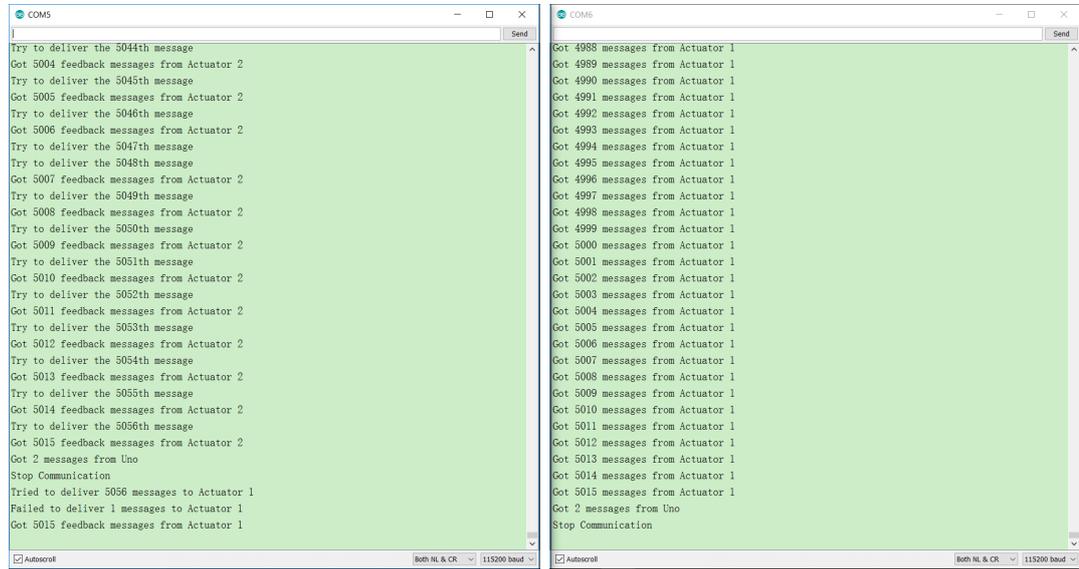


Figure 5.17(a) shows a serial monitor window for COM5 displaying the output of Prismatic Actuator 1. The log consists of alternating 'Try to deliver' and 'Got' messages for messages 5044 through 5015, followed by 'Stop Communication' and a final 'Got 5015 feedback messages from Actuator 1'. Figure 5.17(b) shows a serial monitor window for COM6 displaying the output of Prismatic Actuator 2, with a log of 'Got' messages for messages 4988 through 5015, followed by 'Stop Communication' and 'Got 2 messages from Uno'.

(a)

(b)

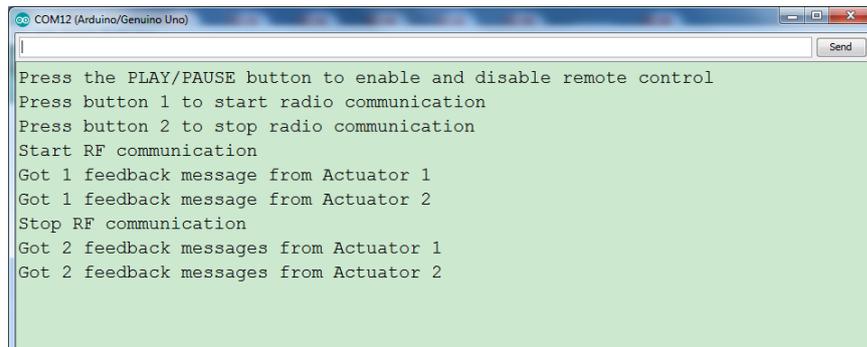


Figure 5.17(c) shows a serial monitor window for COM12 (Arduino/Genuino Uno). The output includes control instructions: 'Press the PLAY/PAUSE button to enable and disable remote control', 'Press button 1 to start radio communication', and 'Press button 2 to stop radio communication'. It then shows 'Start RF communication', followed by 'Got 1 feedback message from Actuator 1' and 'Got 1 feedback message from Actuator 2', then 'Stop RF communication', and finally 'Got 2 feedback messages from Actuator 1' and 'Got 2 feedback messages from Actuator 2'.

(c)

Figure 5.17: Communication of of Two Separate Actuators. (a) Output Information of Prismatic Actuator 1. (b) Output Information of Prismatic Actuator 2. (c) Output Information of the Uno Node.

developed communication architecture using IR and nRF24L01+ modules.

Based on communication, state transitions of the two separator actuators are validated. Two snapshots are shown in Figure 5.18(a) and (b) during the experiments. In Figure 5.18(a), actuator 1 is locked and actuator 2 is passive, while in Figure 5.18(b), the states of the two actuators are exchanged. Motion profiles of the two actuators are illustrated in Figure 5.18(c) and (d) from which it can be observed that the three

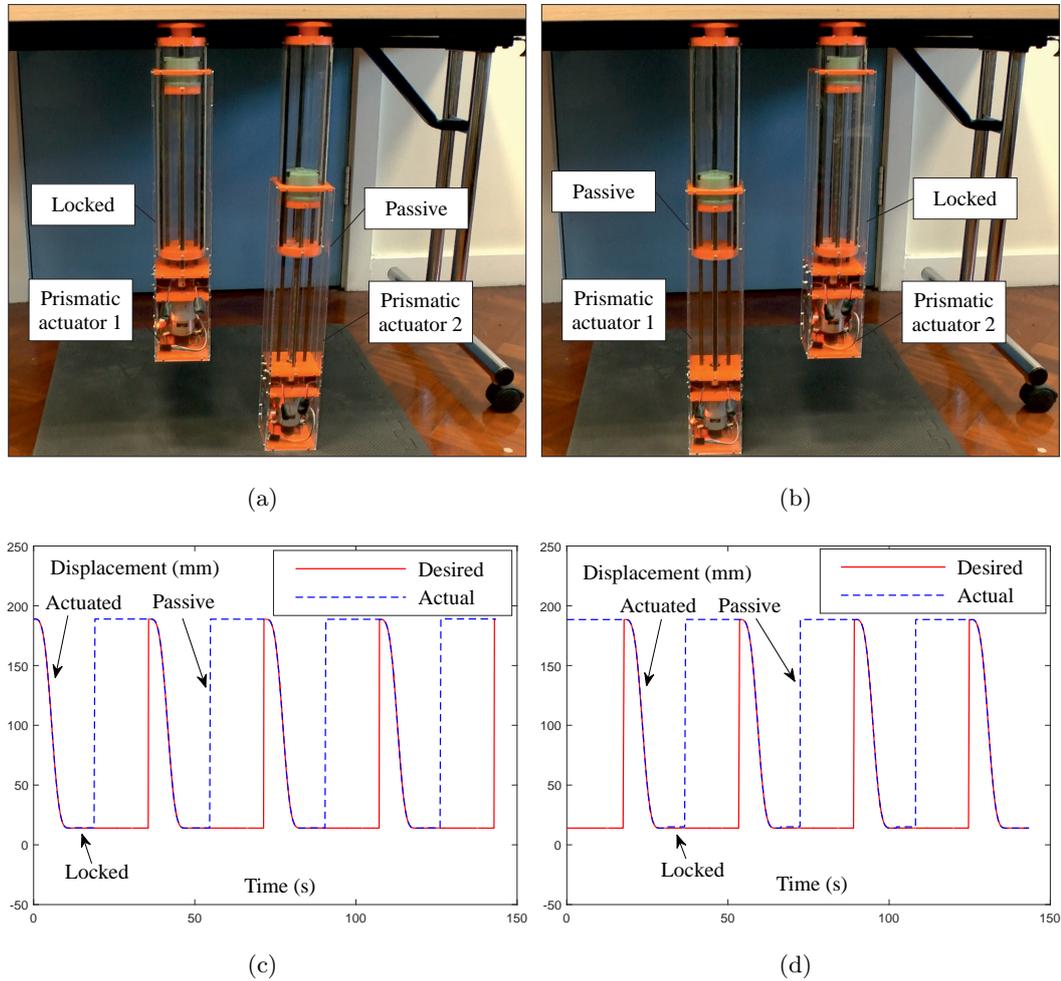


Figure 5.18: State Transitions of Two Separate Actuators. (a) With Actuator 1 Locked. (b) With Actuator 2 Locked. (c) Motions of Actuator 1. (d) Motions of Actuator 2.

states and state transitions can be achieved by the fabricated prismatic actuators. Note that the setpoint of a passive actuator is set as a constant to conveniently identify the passive state. In this experiment, the communication between the two actuators are independent on the Uno RF node during state transitions.

Parallel Structures with Multiple Prismatic Actuators

Figure 5.19 shows the motions of the two-actuator parallel structure. During the passive state, the desired displacement of an actuator is set as 0 to conveniently identify the

5.3 Experimental Results

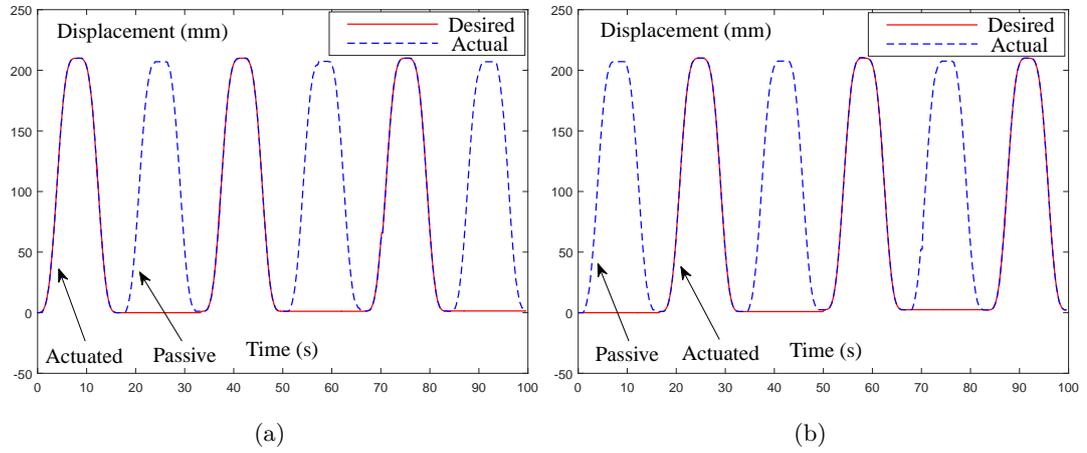


Figure 5.19: Motions of Two Parallel Actuators with Actuated and Passive States. (a) Motions of Prismatic Actuator 1. (b) Motions of Prismatic Actuator 2.

actual displacement of the actuator. Evidently, as shown in the figure, the two actuators can change their states and move reciprocally. For an active actuator, the desired and actual displacements coincide quite well while the passive actuator moves passively. Thus, the state-changing and communication capabilities of the developed prismatic actuators are verified again.

Motion profiles of the three-actuator parallel structure are illustrated in Figure 5.20. As demonstrated, the three actuators can change their states and move up and down

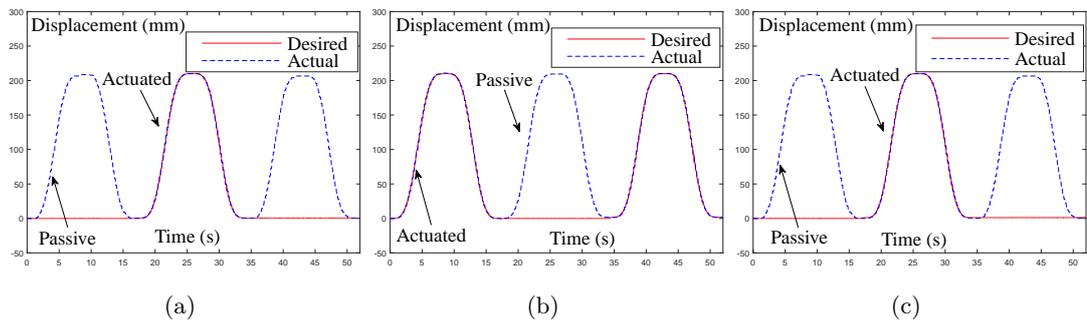


Figure 5.20: Motions of Three Parallel Actuators with Actuated and Passive States. (a) Motions of Prismatic Actuator 1. (b) Motions of Prismatic Actuator 2. (c) Motions of Prismatic Actuator 3.

5.3 Experimental Results

indicating that the two outer actuators (i.e., actuators 1 and 3) can move passively due to the force provided by the middle active actuator (i.e., actuator 2). This feature of the implemented prismatic actuators are of importance as it can be applied to applications where energy consumption requires to be reduced.

Motion profiles of the four-actuator parallel structure are shown in 5.21 from which it can be seen that all of the actuators can achieve state transitions resorting to the communication between the actuators. Motions of actuators 1 and 3 are similar as the two actuators run almost the identical codes and communicate with actuators 2 and

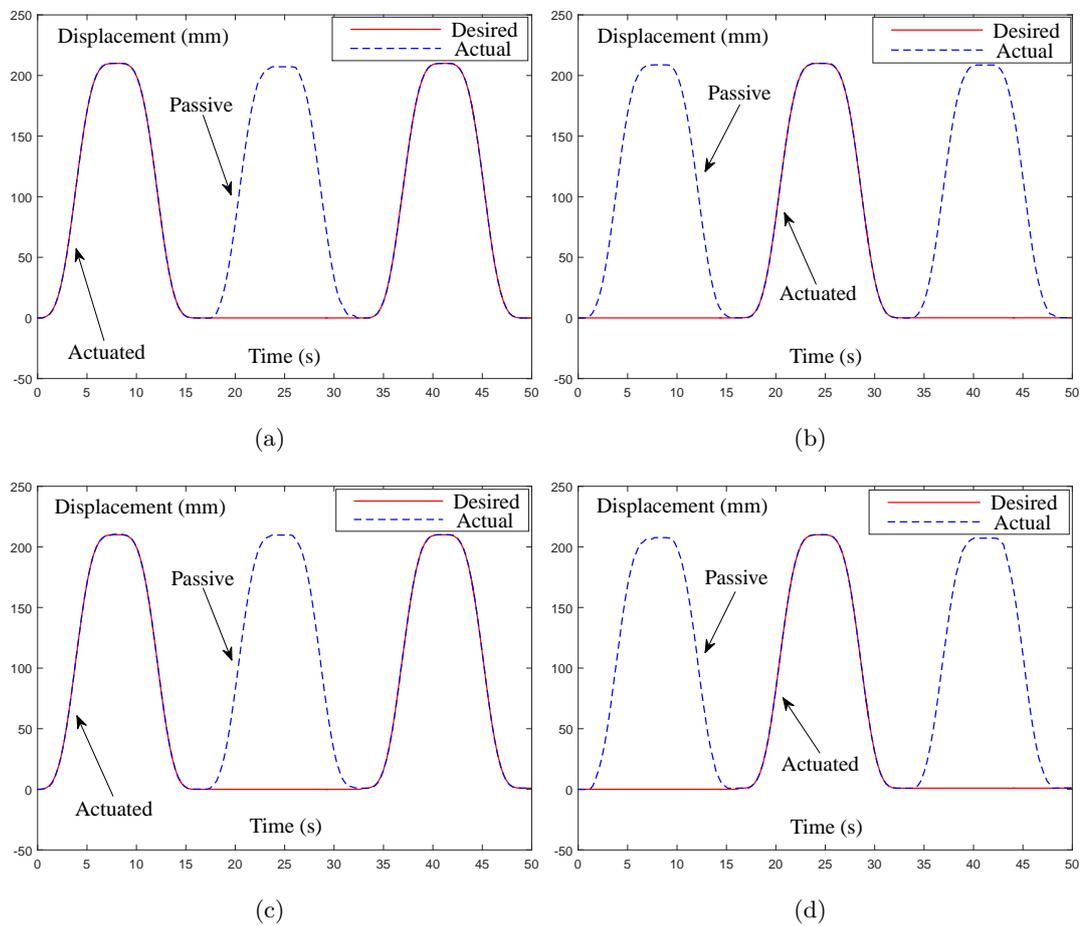


Figure 5.21: Motions of Four Parallel Actuators with Actuated and Passive States. (a) Motions of Prismatic Actuator 1. (b) Motions of Prismatic Actuator 2. (c) Motions of Prismatic Actuator 3. (d) Motions of Prismatic Actuator 4.

4, respectively. Since two actuators can be actuated and the other two can be set as passive simultaneously, this is potentially beneficial to reducing control complexity of some parallel structures composed of the designed prismatic actuators.

5.4 Conclusions

The performance of the implemented prismatic actuators has been investigated in this chapter. In the actuated state, a PID controller has been developed and applied to the prismatic actuators. Based on Prototype I, a laser light sensor was used as reference to verify the effectiveness of the employed rotary encoder. Step response of the control system has been studied and sinusoidal-like trajectory-tracking tasks have been completed using the prismatic actuator. The encoder has 800 pulses for a displacement of 14 mm of the lead screw leading to a positioning precision of 0.0175 mm. As demonstrated, the PLA version Prototype II can lift an external load of 29.43 N in its actuated state. For the locked state, the performance of the electromagnetic brake has been evaluated experimentally and the maximum weight that the brake can hold is estimated as 77.81 N. The force of the mechanical locker has been measured using a force sensor with a resulting maximum force of around 78.6 N obtained. Theoretically, the force of the locker could be more. However, the motor gears would be broken once the force is over a certain threshold in practice. The estimated actuation and locking forces substantiate the theoretically analysed results in the preceding chapter. Different lead screws have been exploited and compared to study the passive compliance capability of the prismatic actuator using a pulley system for Prototype I. Finally, three states and state transitions of individual and multiple actuators have been demonstrated using the Prototype II showing that the prismatic actuators can realize state transitions between the three different states. All of the experimental results verify the feasibility of the designed prismatic actuators and show that the Prototype II is capable of achieving the actuated, locked and passive states.

CHAPTER 6

Control and Locomotion of Modular Robots

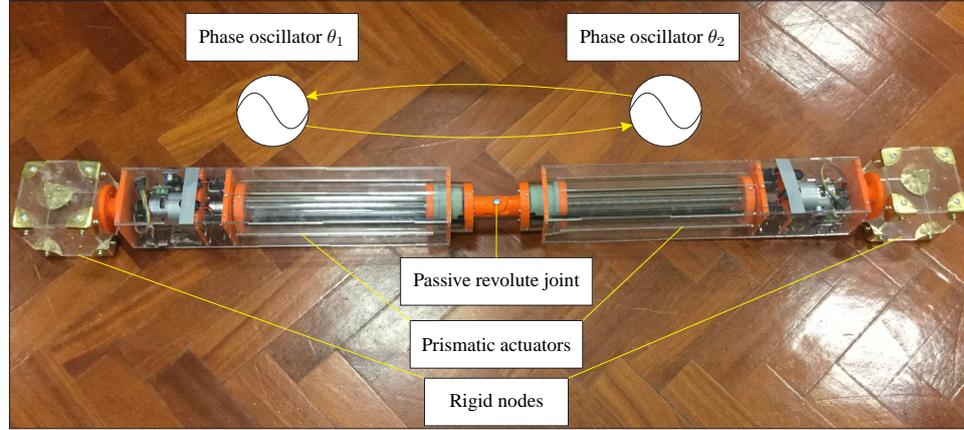
This chapter investigates the control and locomotion of various modular robotic structures composed of three-state prismatic actuators both in a simulated environment and in the real world.

6.1 Amorphous Robots

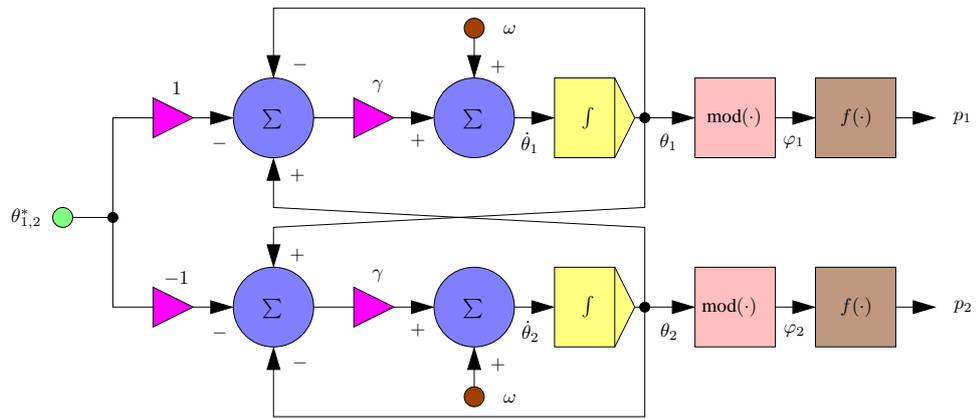
This section focuses on a class of amorphous robots constructed by using rigid nodes and robotic struts consisting of two three-state prismatic actuators linked by a passive revolute joint. Hereafter, such a strut is also termed as a robotic module. This section first studies the control and locomotion of four physical robots, i.e., a single-module worm robot, a two-module worm robot, a four-module square robot and a four-module parallel mechanism. After that, a generalized simulation and control framework is established for this type of amorphous robots.

6.1.1 Single-Module Worm Robot

This subsection investigates a single robotic module shown in Figure 6.1(a). As can be seen from the figure, the robotic module possesses two prismatic actuators and a passive revolute joint. The moving principle introduced in Chapter 3 is applied to the physical robot to obtain a locomotion gait. In the described moving principle, the initial states of the two actuators of the robot are assumed to be fully contracted. However, in practice, the two actuators may have initial displacements. In this case, one way to achieve a locomotion is to let the actuators return to their fully contracted states before following the moving steps. If this way is applied to networked robotic structures wherein each actuator has an initial position, physical constraints may conflict with the specified actuator displacements when letting all the actuators be fully contracted. Therefore, it is worth gradually coordinating movements of the actuators to achieve the worm-like locomotion from an initial state. To this end, a simplified CPG method using phase oscillators is implemented with the schematic diagram illustrated in Figure 6.1(b). The core idea is to adopt a periodical sinusoidal-like position profile to generate setpoint signals p for each actuator. Just like a sine function $p = L \sin(\omega t + \vartheta)$ with p , L , ω , t and ϑ separately representing displacement, amplitude, frequency, time and phase shift, the sinusoidal-like function can also shift its phase. For coordinating movements of two actuators, the phases of two sinusoidal-like functions can be updated and coordinated.



(a)



(b)

Figure 6.1: A Robotic Module Composed of Two Prismatic Actuators and A Schematic Diagram of the Employed CPG Controller. (a) Robotic Module. (b) CPG Controller.

If the phase difference of the two sinusoidal-like functions is constant, then one of the actuator's movements would be delayed or the two actuators' movements will be synchronized.

More specifically, each prismatic actuator of the robot has a phase oscillator as presented in Figure 6.1(a) and (b). Initially, each phase oscillator has an initial phase $\theta_i(0)$ with $i = 1, 2$. The phase difference (i.e., $\theta_{1,2}(t) = \theta_2(t) - \theta_1(t) = -\theta_{2,1}(t)$) of the two oscillators can finally converge to the desired phase difference by exchanging information and updating the phases according to the following control law [Sato *et al.*,

2011, Yu, 2010]:

$$\begin{cases} \dot{\theta}_1(t) = \omega + \gamma(\theta_{1,2}(t) - \theta_{1,2}^*), \\ \dot{\theta}_2(t) = \omega + \gamma(\theta_{2,1}(t) - \theta_{2,1}^*), \end{cases} \quad (6.1)$$

where ω denotes the intrinsic frequency and $\theta_i(t)$ represents oscillator phase of strut i at time t with $\dot{\theta}_i(t)$ being its derivative. Positive parameter γ is related to the convergence speed. Constant $\theta_{1,2}^*$ is the desired phase difference and its value is dependent on the moving direction of the robot. Specifically speaking, if the robot moves leftward, $\theta_{1,2}^* = -\pi/2$ rad, otherwise, $\theta_{1,2}^* = \pi/2$ rad. To force phases $\theta_1(t)$ and $\theta_2(t)$ to be bounded between 0 rad and 1.5π rad and to obtain the actuation signal, the following computations are perform:

$$\varphi_i(t) = \text{mod}(\theta_i(t), 1.5\pi), \quad (6.2)$$

and

$$f(\varphi) = \begin{cases} L \sin^2(0.5\pi \sin^2(0.5\pi\varphi/P)), & \text{if } 0 < \varphi < \pi, \\ 0, & \text{otherwise,} \end{cases} \quad (6.3)$$

where $\text{mod}(\cdot)$ is to compute modulus after division and $f(\cdot)$ is an activation function (AF) with L and P indicating the amplitude and a constant, respectively. Note that the designed AF (6.3) (i.e., AF I) is different from the AF presented in [Yu, 2010] (i.e., AF II). Specifically, the AF II is defined as

$$f(\varphi) = \begin{cases} 2L\varphi/\pi, & \text{if } \varphi \leq \pi/2, \\ 2L(\pi - \varphi)/\pi, & \text{if } \pi/2 < \varphi \leq \pi, \\ 0, & \text{otherwise} \end{cases} \quad (6.4)$$

with $\varphi(t) = \text{mod}(\theta(t), 2\pi)$ indicating a phase bounded between 0 rad and 2π rad.

The profiles of AFs I and II with respect to phase φ are respectively shown in Figure 6.2(a) and (b) for better understanding. To further compare the two AFs with respect to time, parameters $\theta_1(0)$, $\theta_2(0)$, $\theta_{1,2}^*$, ω , L and P are separately set as 0, 0, 0, $\pi/5$, 300 and $\pi/2$ as an illustrative example, making AFs I and II cyclic with their periods being 7.5 s and 10 s, respectively. Figure 6.2(c) and (d) shows displacement and velocity profiles of AFs I and II with time ranging from 0 s to 30 s. As shown in Figure 6.2(c), AFs I and II separately have 4 and 3 cycles within 30 s. This is because AF I is designed to exhibit a shorter time to keep fully contracted between two neighbouring

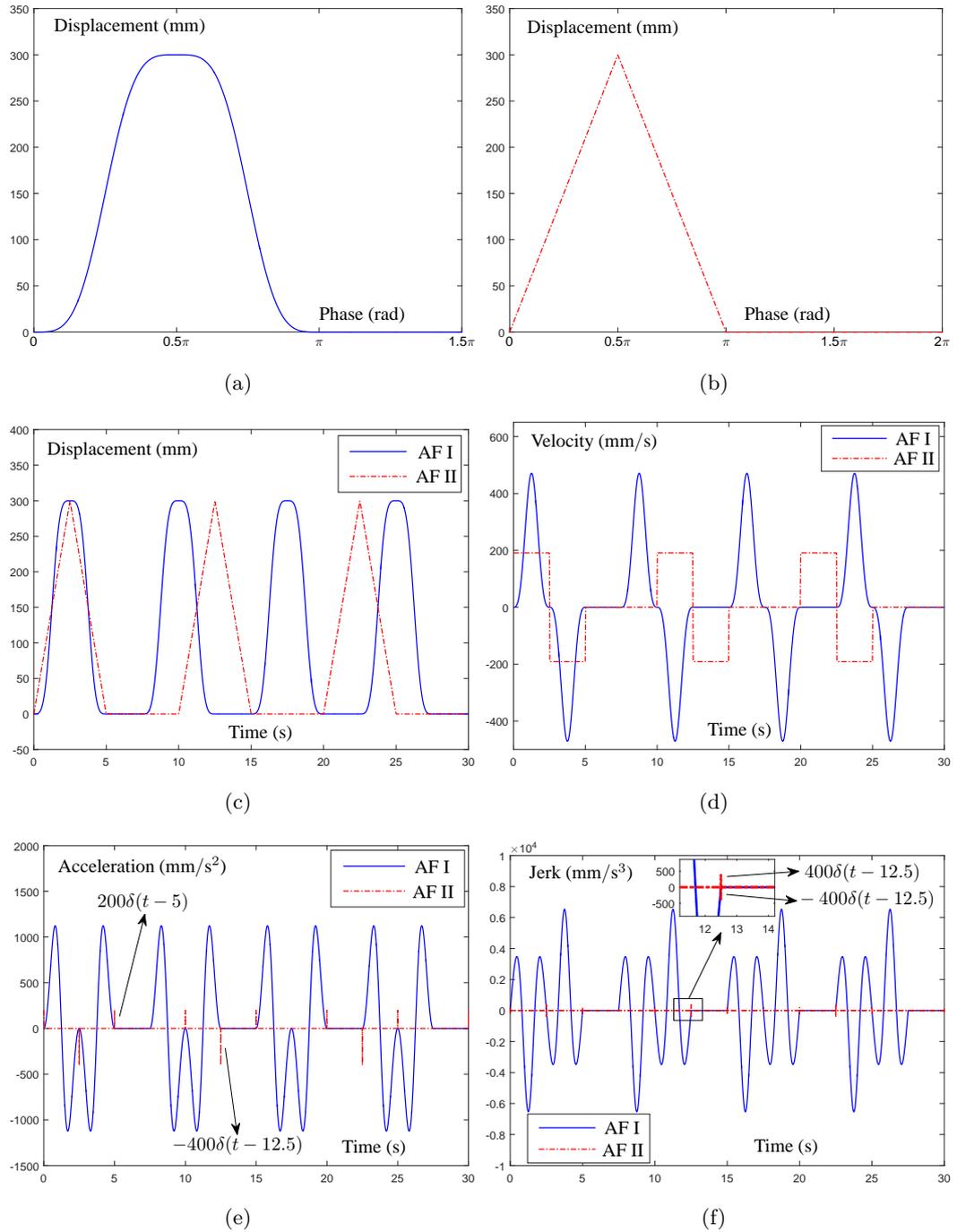


Figure 6.2: Motion Profiles of Two Actuation Functions. (a) Displacement Versus Phase Profile of AF I. (b) Displacement Versus Phase Profile of AF II. (c) Displacement Versus Time Profiles. (d) Velocity Versus Time Profiles. (e) Acceleration Versus Time Profiles. (f) Jerk Versus Time Profiles.

extension-and-contraction cycles. In this way, it can speed up the locomotion process when the actuators are cyclically driven to achieve a locomotion gait. For extension or contraction, AFs I and II have a same average velocity of 120 mm/s. The displacement profile of AF I shown in Figure 6.2(c) is smoother than that of AF II. Actually, AF II is not a differentiable function and its velocity profile shown in Figure 6.2(d) is the so-called constant-velocity-window type profile [Arevalo, 2001]. When using a constant-velocity-window profile for motion control, the system must have power to tolerate undesired acceleration impulses [Arevalo, 2001]. The acceleration profile of AF II is presented in Figure 6.2(e) where the impulses are mathematically expressed as Dirac delta function $\delta(\cdot)$. Note that the unit delta function $\delta(t)$ is defined as infinite at $t = 0$ and 0 for $t \in (-\infty, 0) \cup (0, +\infty)$. There exist other motion profiles such as triangular and trapezoidal velocity profiles to remedy the problem of acceleration impulses, however, triangular and trapezoidal velocity profiles will produce jerk impulses that are also unfavourable [Karlsson, 2009, Tooley, 2009, Voss, 2007]. Such impulses can be eliminated using the differentiable AF I of which the velocity, acceleration and jerk profiles are all continuous as presented in Figure 6.2(d)–(f). Hence, AF I is more suitable than AF II for motion planning in robotics [Saha, 2008].

To coordinate phases of the two prismatic actuators, a communication architecture implemented and employed in the experiment is presented in Figure 6.3. Three different communication strategies (i.e., IR, RF and serial communications) are used in such an architecture. Steps of conducting the experiment can be described as follows.

- 1) An operator sends commands to the Arduino Uno through IR communication and then the Uno as an RF node relays the received commands to the two Nano RF nodes via RF communication. The user-specified commands include information for assigning a travelling direction and letting the robot get started or stop running the program.
- 2) Each of the two prismatic actuators move to an initial position according to its own generated initial phase. If the initial state is achieved, the Nano master node sends a message to query about whether the initialization process of the Nano slave node is finished or not. Once the Nano master node receives a positive feedback, it sends a message to the Uno RF node to inform that the initialization of the whole robot is completed. Note that during this stage, the Nano slave node would wait for the message sent by the Nano master node if it finishes the

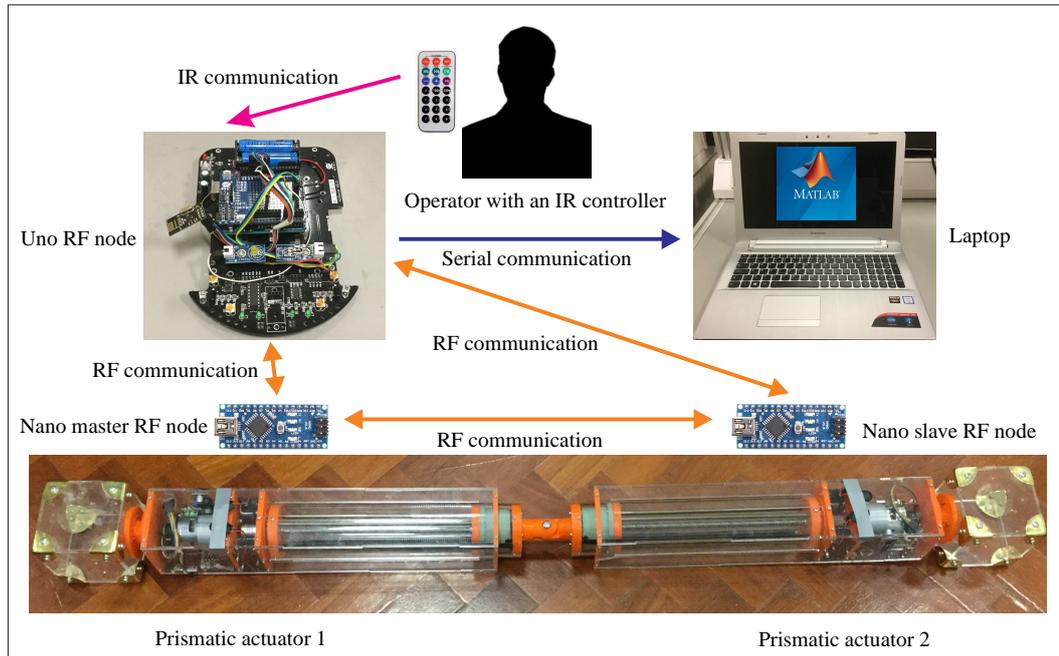


Figure 6.3: Communication Architecture of the Single-Module Worm Robot.

initialization earlier.

- 3) The Uno RF node working as a data hub sends messages including the phases of the two actuators to the Nano nodes every 0.1 s. According to the received message, each Nano node updates its phase and then each actuator moves to a setpoint using a PID controller at each time step. Meanwhile, the two Nano nodes send messages back to the Uno node and then the laptop with Matlab reads data from the Uno node through serial communication.

In this way, the two actuators can coordinate their movements and finally move leftward or rightward as a whole robot.

In this experiment, the PID gains are empirically tuned as $k_p = 55$, $k_i = 162.5$ and $k_d = 0.5$. Parameters ω and γ in control law (6.1) are respectively set as 0.001π and 0.0005 . Experimental results are shown in Figures 6.4 and 6.5. Initially, the worm robot is fully contracted as shown in Figure 6.3. When it receives the command to get started to move leftward, the two actuators of the worm robot first move to an initial position as shown in Figure 6.4(a). Then the two actuators exchange information through the intermediate Uno node and update their phases using control law (6.1). As presented

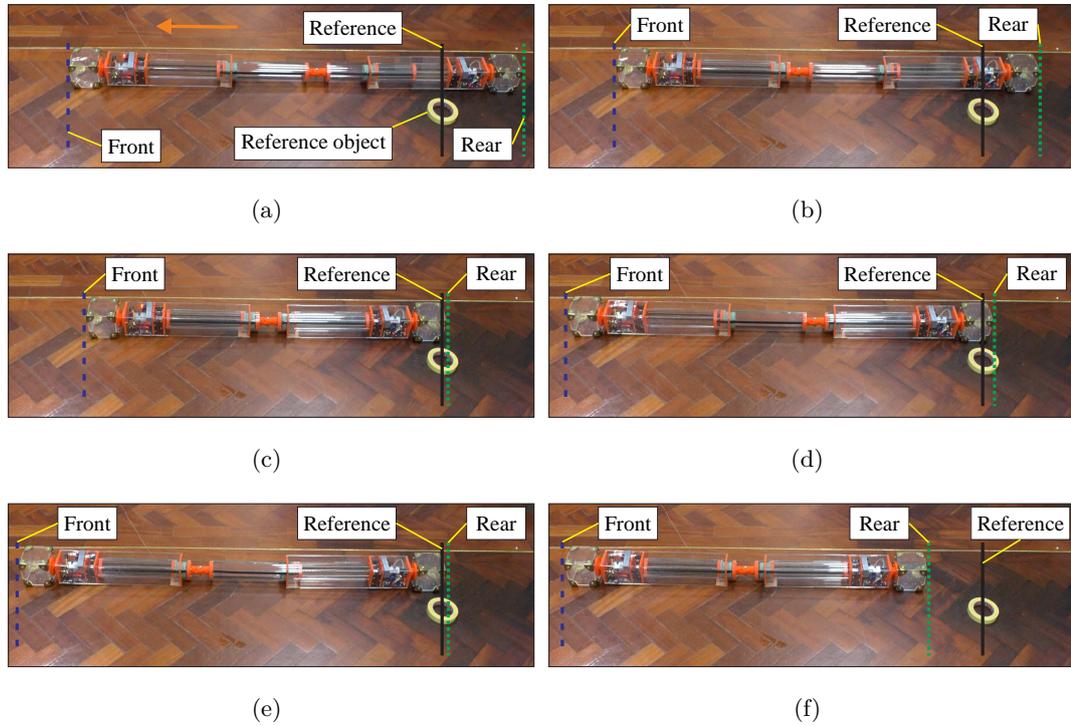


Figure 6.4: Phase Coordination and Locomotion of the Single-Module Worm Robot. (a) Initialized State. (b) and (c) Coordination Stage. (d) Locomotion Step 1. (e) Locomotion Step 2. (f) Locomotion Step 3.

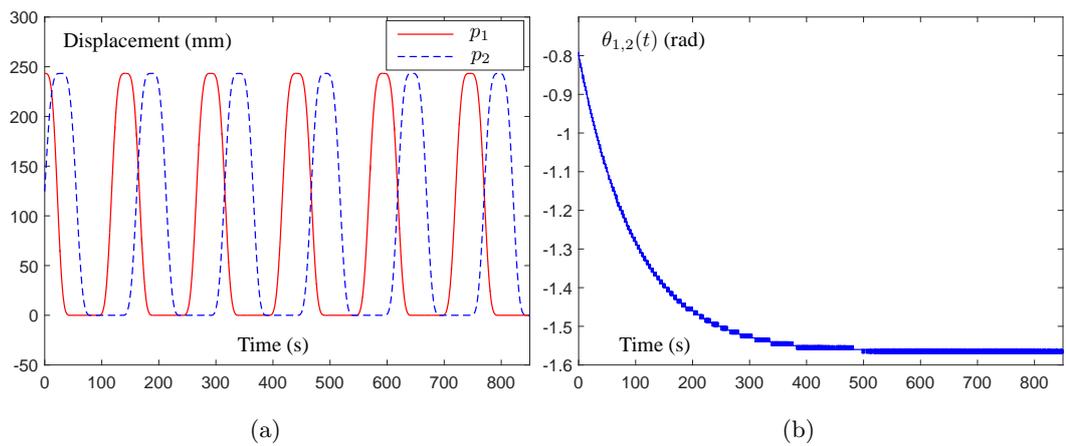


Figure 6.5: Displacements and Phase Difference of the Two Prismatic Actuators. (a) Displacements Measured by Encoders of Actuators. (b) Phase Difference of Actuators.

in Figure 6.4(b)–(f), the robot moves leftward and a worm-like locomotion is obtained. Figure 6.5 shows the displacements measured by encoders and the phase difference of the two actuators p_1 and p_2 . Evidently, a coordinating motion is obtained as shown in Figure 6.5(a) and the phase difference finally converges to $-\pi/2$ rad. In the coordinated locomotion stage, the robot can move leftward around 0.2 m in each three-step cycle that needs about 150 s to be finished. These results substantiate the coordination and locomotion abilities of the physical modular robot and verify the effectiveness of the developed control and communication system.

6.1.2 Two-Module Worm Robot

In order to investigate coordination between modules, a two-module robot consisting of four prismatic actuators that can achieve a worm-like locomotion is built. From the experimental results of the single-module robot, it is readily to know that the robot can move rightward or leftward depending on a phase difference of $\pi/2$ rad or $-\pi/2$ rad. Here, this result is directly applied to the two-module worm robot. Specifically speaking, as shown in Figure 6.6, each module consists of one master RF node and one

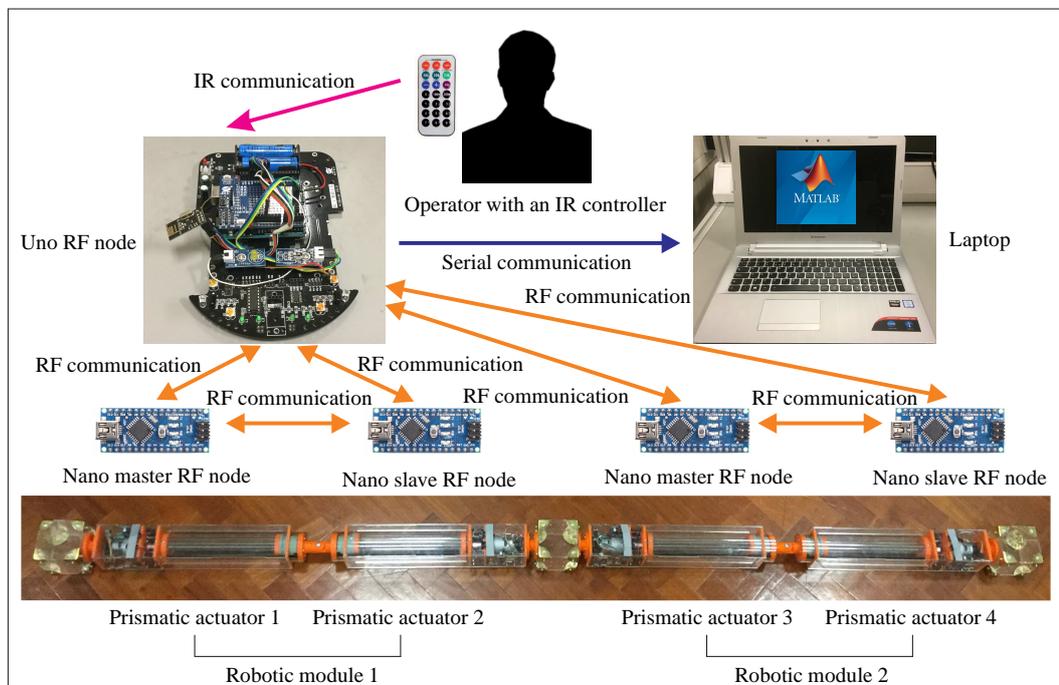


Figure 6.6: Communication Architecture of the Two-Module Worm Robot.

slave RF node. The two master RF nodes communicate and update phases in a similar way used in the single-module robot as shown in Figure 6.1 and each of the two slave RF nodes passively receives message from its paired master RF node and updates its phase depending on the phase difference. Mathematically, phases of the four prismatic actuators are coupled as follows:

$$\begin{cases} \dot{\theta}_1(t) = \omega + \gamma(\theta_{1,3}(t) - \theta_{1,3}^*), \\ \theta_2(t) = \theta_1(t) - \theta_{2,1}^*, \\ \dot{\theta}_3(t) = \omega + \gamma(\theta_{3,1}(t) - \theta_{3,1}^*), \\ \theta_4(t) = \theta_3(t) - \theta_{4,3}^*, \end{cases} \quad (6.5)$$

where $\theta_{1,3}^* = -\theta_{3,1}^* = \pi$ rad and $\theta_{2,1}^* = \theta_{4,3}^* = \pi/2$ rad when the travelling direction is set as rightward. Otherwise, $\theta_{1,3}^* = -\theta_{3,1}^* = -\pi$ rad and $\theta_{2,1}^* = \theta_{4,3}^* = -\pi/2$ rad for moving leftward. By coordinating and updating phases of the four actuators, a worm-like locomotion gait of the two-module robot can be attained.

Figure 6.7 shows the snapshots captured when the two-module robot coordinates its actuators' movements and finally moves rightward. As can be observed from Figure 6.7(a), the four prismatic actuators are initially fully contracted. When the robot receives the command to start running the program from the Uno RF node, an initialization process is first conducted with the initialized state shown in Figure 6.7(b). After that, the four prismatic actuators update their phases according to the control law and finally the whole robot achieves a worm-like locomotion gait presented in Figure 6.7(c)–(o). The motion profiles of the four actuators are shown in Figure 6.8(a) where the motions of prismatic actuators 1 and 4 are finally synchronized as the phase difference between these two actuators is 1.5π that is used in (6.2) to generate the cyclic actuation signal with a period of 1.5π . Besides, the phase difference of prismatic actuators 1 and 3 (i.e., $\theta_3(t) - \theta_1(t)$) as shown in Figure 6.8(b) finally converges to a constant value of π rad. These results again validate the communication and locomotion capabilities of the designed and fabricated modular robots.

6.1.3 Four-Module Square Robot

The single-module and two-module robots can only move along a straight line, to demonstrate that the modular robotic system has the ability to move in a 2D space, a four-module square robot is built as shown in Figure 6.9. Such a robot can travel

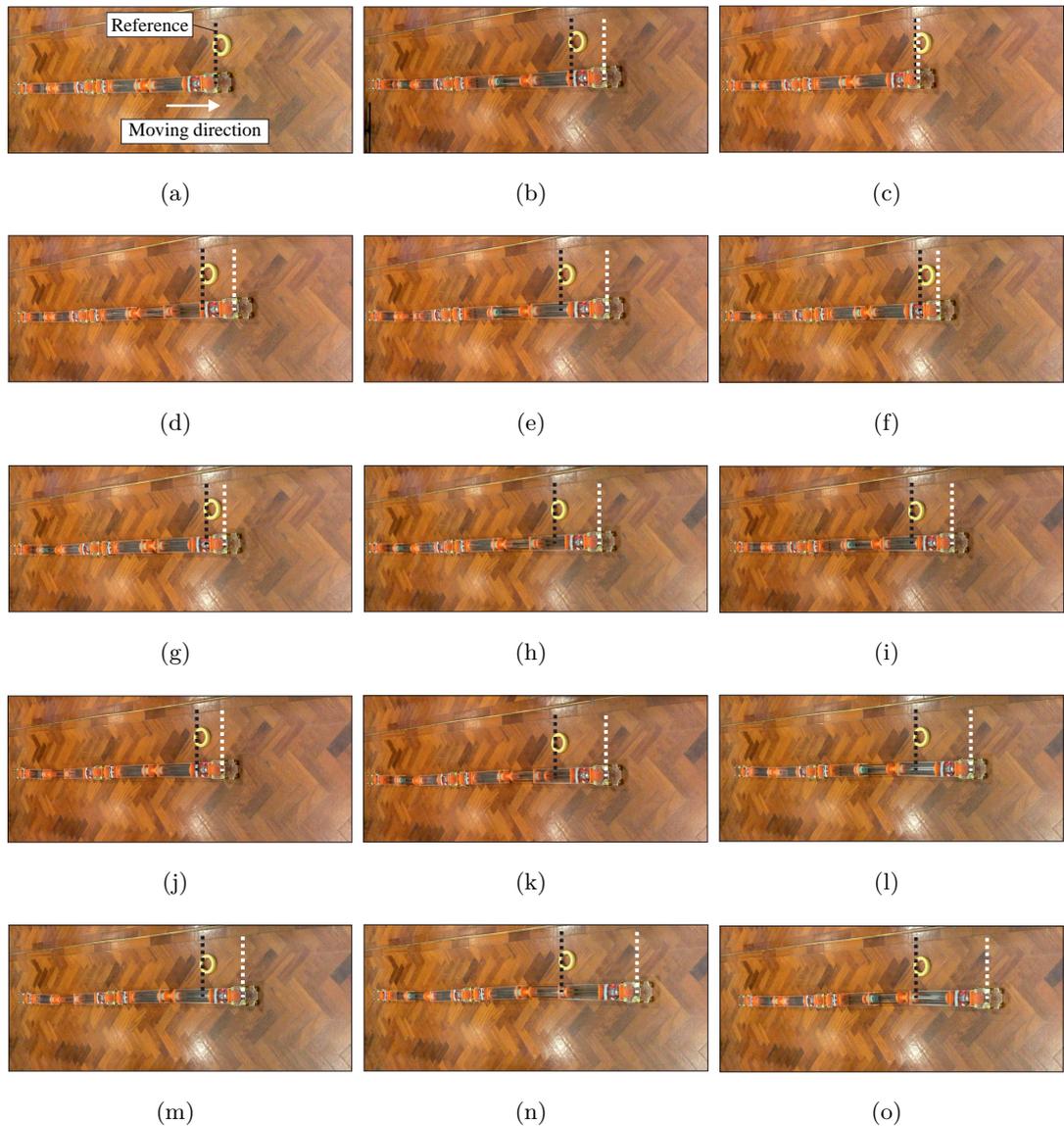


Figure 6.7: Phase Coordination and Locomotion of the Two-Module Worm Robot. (a) Static State. (b) Initialized State. (c)–(o) Snapshots Captured When the Robot Moves Rightward.

along four cardinal directions (i.e., directions 1, 2, 3 and 4) as illustrated in the figure by using a similar control and communication architecture shown in Figure 6.1 that works for the one-module and two-module robots. Each robotic module of the square

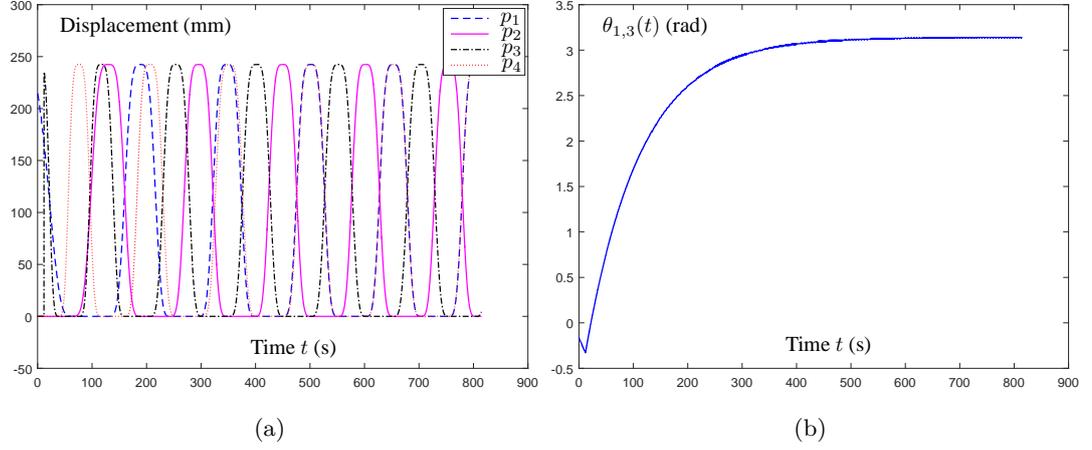


Figure 6.8: Actuator Displacements and Phase Difference of the Two-Module Robot. (a) Actuator Displacements Measured by Encoders. (b) Phase Difference of Actuators.

robot has a Nano master and a Nano slave communication nodes. When moving along a specific direction, the robotic modules that are in parallel with the direction are set as active and the desired phase difference of the two active Nano master nodes is set as 0 rad.

To verify that the moving principle presented in Chapter 3 is applicable to the square-shaped robot, the square robot is first assigned to move along direction 2 as defined in Figure 6.9. Here, the initial phases of the two Nano master nodes are set as 0 rad, which means that the coordination process is not required as the phase difference is already 0 rad. The snapshots captured when the square-shaped robot moves along direction 2 are shown in Figure 6.10. The square robot follows three steps to move along direction 2 in each cycle. Figure 6.10(b)–(j) shows the first three cycles. Snapshots captured when the robot finishes cycles 4 and 5 are shown in Figure 6.10(k) and (l), respectively. From these figures, it can be seen that the robot moves along direction 2 in a worm-like manner by periodically following the three-step moving principle. The locomotion capability of the square-shaped robot is thus demonstrated.

Then, another experiment is performed to validate the coordination and locomotion capabilities of the square-shaped modular robot with captured snapshots presented in Figure 6.11. In this experiment, the square robot is commanded to move along direction 3 and each active phase oscillator has its own initial phase. Starting from the fully

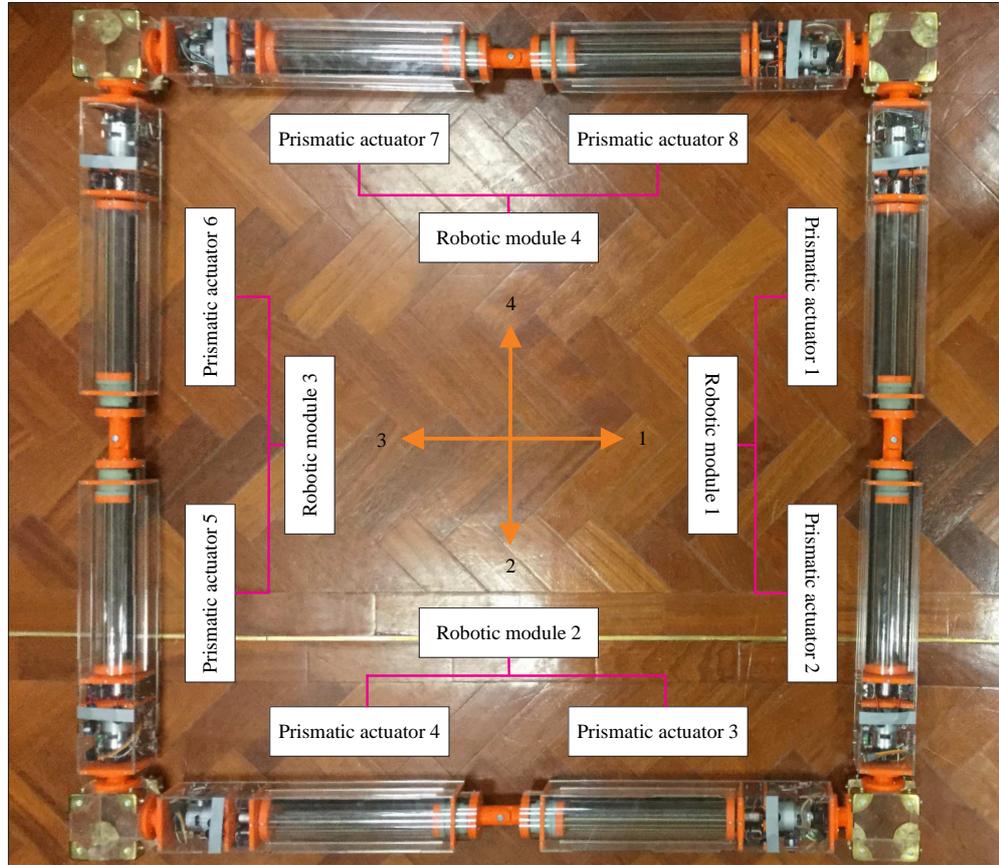


Figure 6.9: A Four-Module Square Robot.

contracted state as shown in Figure 6.11(a), the robot initializes the positions of the active prismatic actuators p_3 , p_4 , p_7 and p_8 according to their initial oscillator phases. After that, the two robotic modules update phases and movements at each time step of 0.1 s through the RF communication with an Arduino Uno RF node. The square structure as a whole robot finally can achieve a worm-like locomotion and travel along direction 3. Position data collected by the Arduino Uno node are plotted in Figure 6.12. As can be seen from the figure, finally, the motions of prismatic actuators p_3 and p_8 are synchronized and the motions of prismatic actuators p_4 and p_7 also coincide well with each other. These results verify the developed control and communication system as well as the locomotion capability of the square robot. Just like the simulated square robot in Chapter 3, the physical square robot would also exhibit buckling shapes during coordination and locomotion as shown in Figure 6.11(1). This does affect the

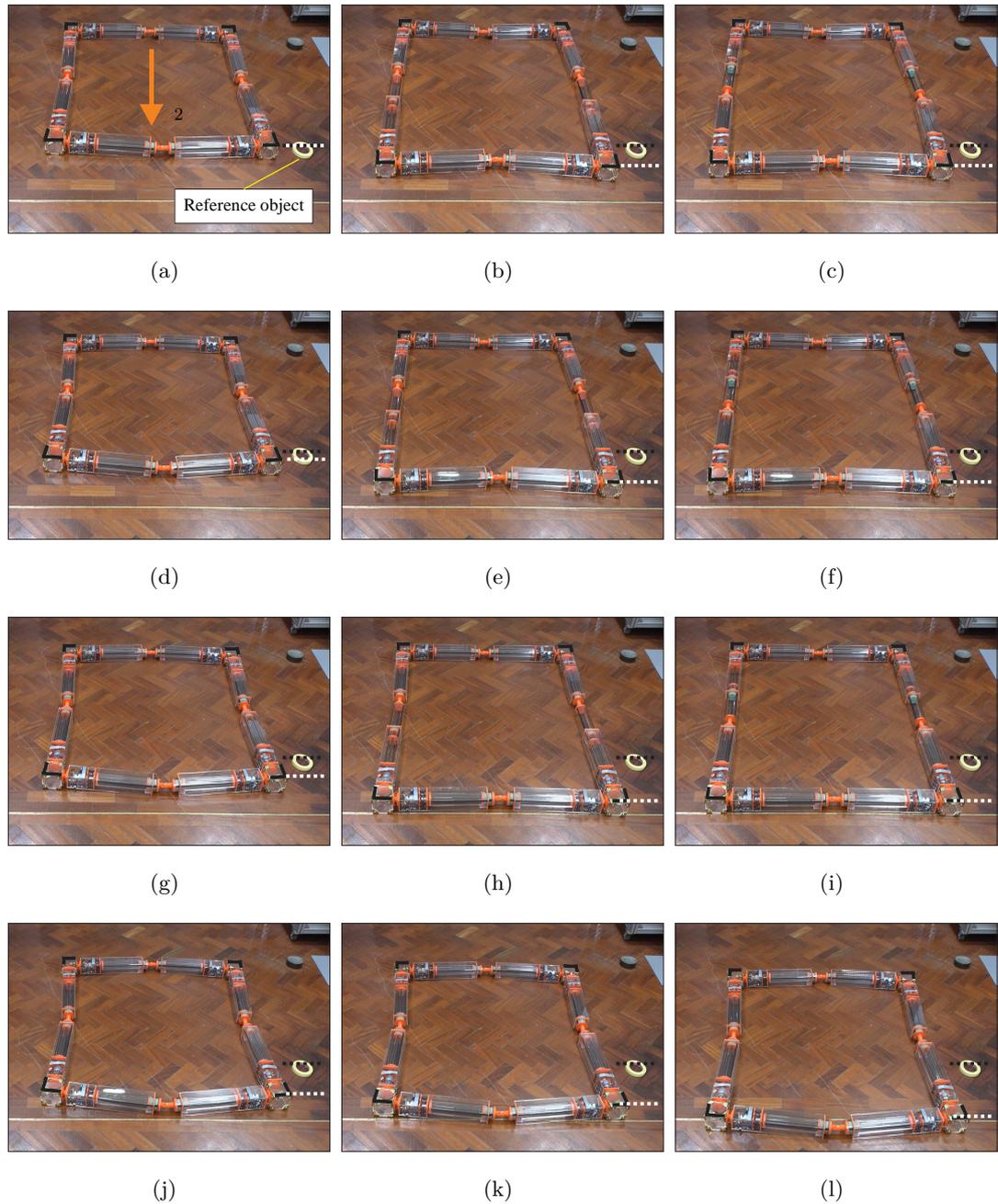


Figure 6.10: Basic Locomotion of the Square Robot When the Robot Moves Along Direction 2. (a) Static State. (b)–(l) Snapshots Captured During Locomotion Stage.

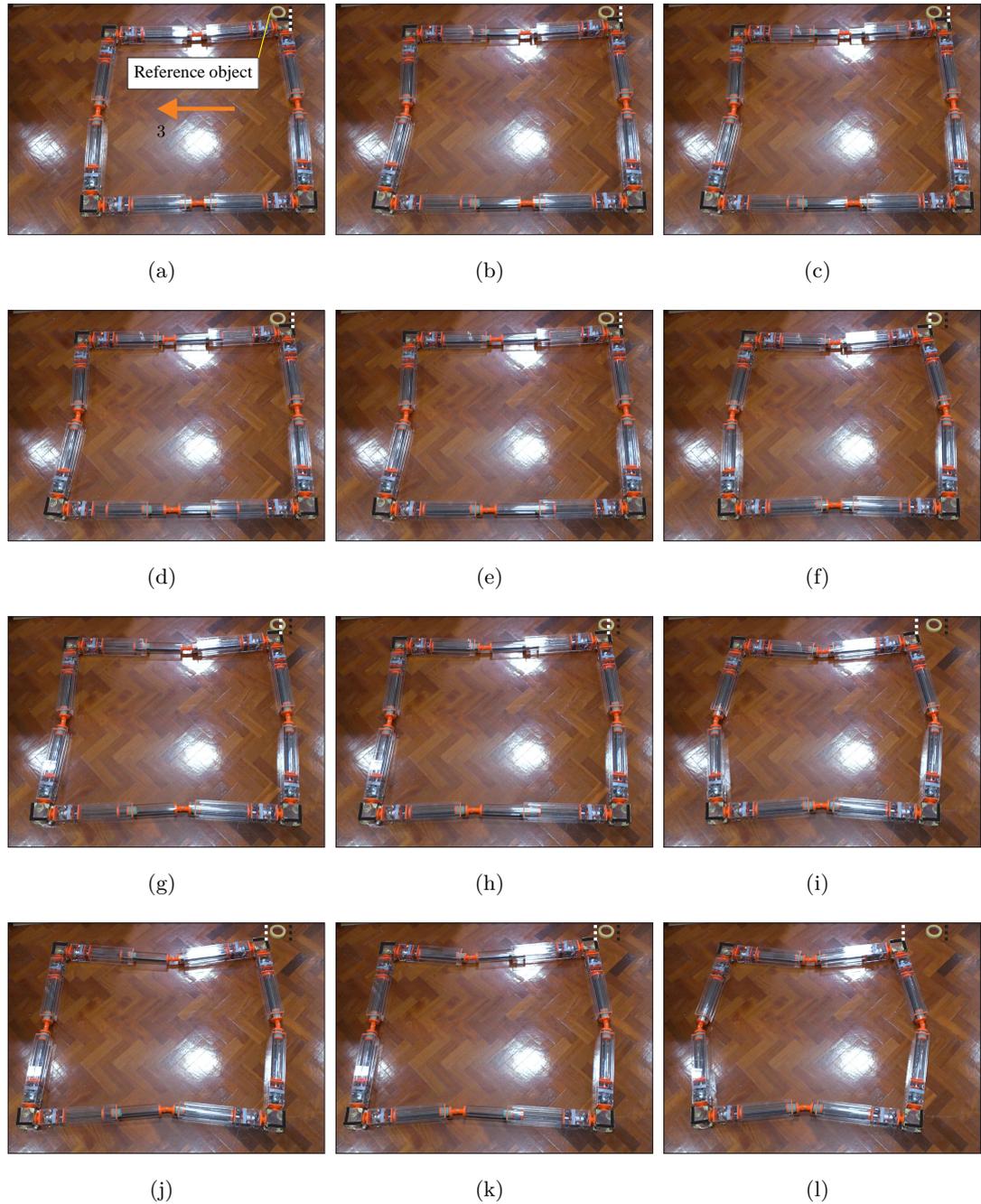


Figure 6.11: Coordination and Locomotion of the Square Robot When the Robot Moves Along Direction 3. (a) Static State. (b) Initialized State. (c)–(l) Snapshots Captured During the Coordination and Locomotion Stages.

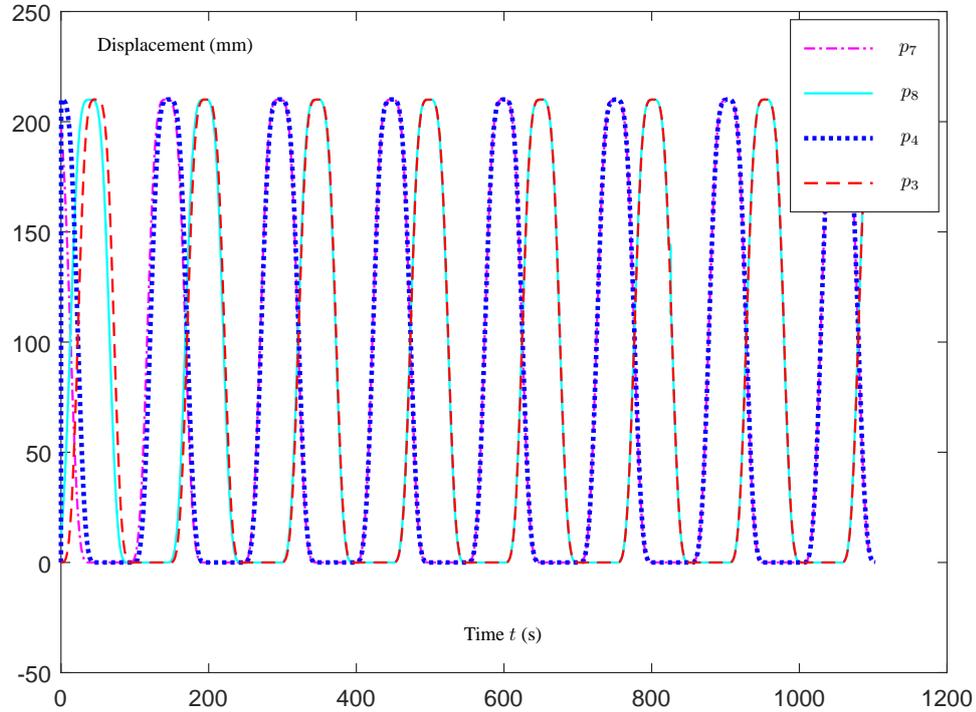


Figure 6.12: Displacements Measured by Encoders and Phase Difference of the Square Robot's Actuators.

locomotion speed of the robot. In the future work, the middle passive revolute joint can be designed to have multiple working states: actuated, locked and passive states. Then, in the coordination stage, the revolute joint can be set as passive to release physical constraints. Once coordinating movements of the actuators are achieved, the revolute joint can be actuated to change the shape of the robot to make it move faster. When a desired shape is obtained, the revolute joint can be locked to save power consumption. In short, the experimental results of the investigated modular robots tentatively demonstrate the feasibility of constructing strut-type modular robots using rigid connector nodes and robotic struts that are comprised of two prismatic actuators linked by a passive revolute joint.

6.1.4 Movements of a 4-PRP Parallel Mechanism

As discussed in Chapter 3, complex robotic structures can be formed by considering the square robot as a meta-module. From the connectivity of these complex structures,

a 4-PRP parallel mechanism can be extracted. To validate the kinematics analysis results presented in Chapter 3, a physical 4-PRP parallel mechanism is built as shown in Figure 6.13. Such a mechanism consists of four robotic modules of which each module has two prismatic actuators linked by a revolute joint leading to a total number of eight actuators. Considering that an nRF24L01+ module can only connect with up to six other nodes in a multiceiver mode, one of the actuators within each robotic module is assigned as a master node to exchange messages with the Uno node and transmit commands to the other actuator serving as a slave node. In this manner, the physical structure can be remotely controlled by an operator. Each robotic module has a fixed node that is attached to the floor using strong Velcro tapes. The other end of the robotic module is connected with a floating node that is located in the middle of the structure. Like in the simulations presented in Chapter 3, the floating node can be treated as an end-effector platform of the 4-PRP parallel mechanism.

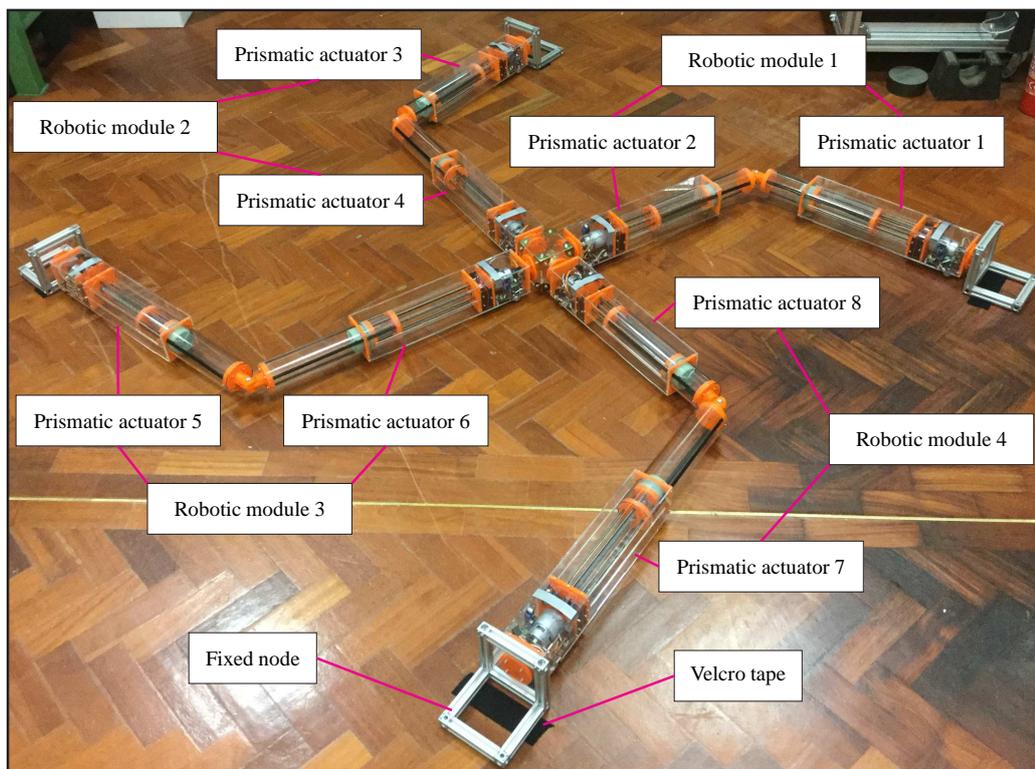


Figure 6.13: A Physical 4-PRP Parallel Mechanism.

To verify the simulation results presented in Chapter 3, a circular path is designed to be followed by the central node by actuating the eight prismatic actuators. The motion snapshots of the 4-PRP mechanism are presented in Figure 6.14 showing that the central node of the structure moves to track the circular path, which demonstrates the mobility of the 4-PRP mechanism and thus verifies the correctness of the kinematics analysis presented in Chapter 3. For better understanding, the desired circular path in the X-Y plane is presented in Figure 6.15(a). Collected motion profiles of the eight prismatic actuators during the task execution process are shown in Figure 6.15(b). The actual trajectory shown in Figure 6.15(a) well coinciding with the desired path can be

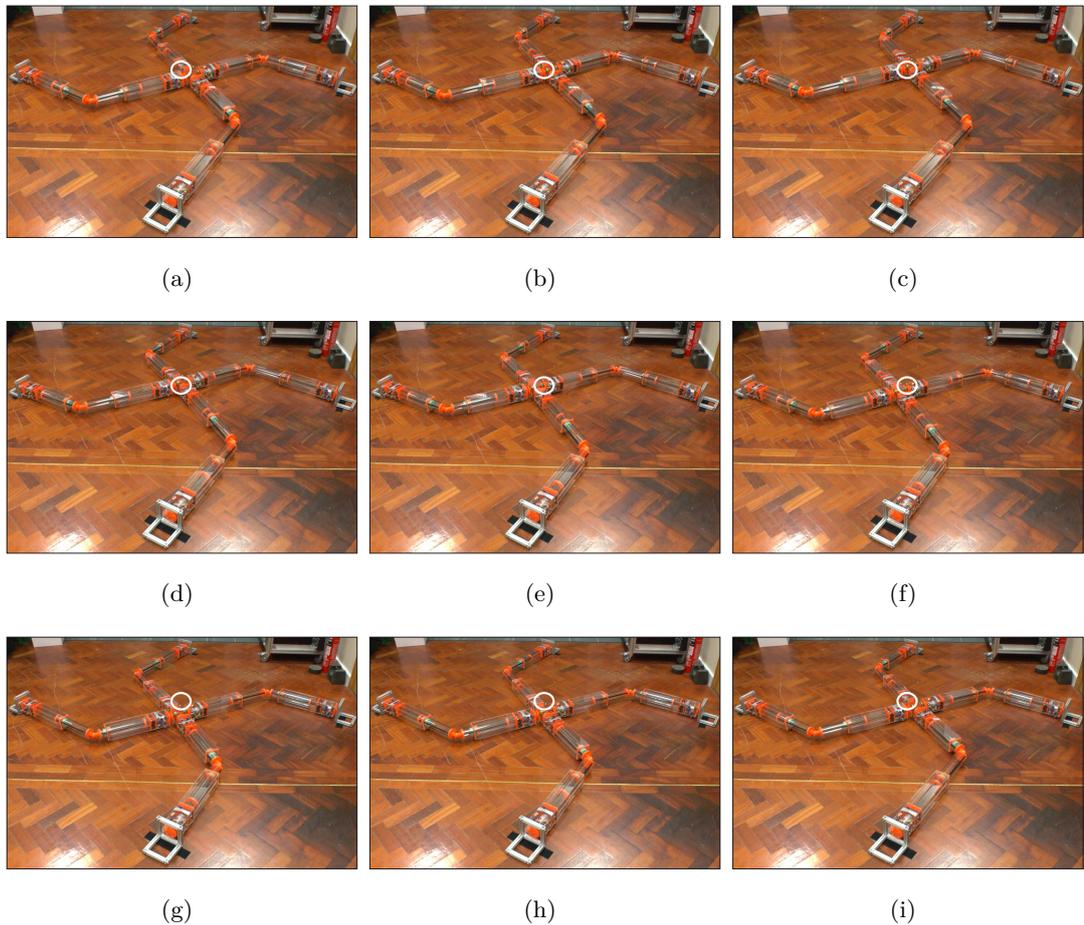


Figure 6.14: Movements of the 4-PRP Parallel Mechanism. (a) Static State. (b)–(i) Snapshots Captured When the Central Node Tracks a Circular Path.

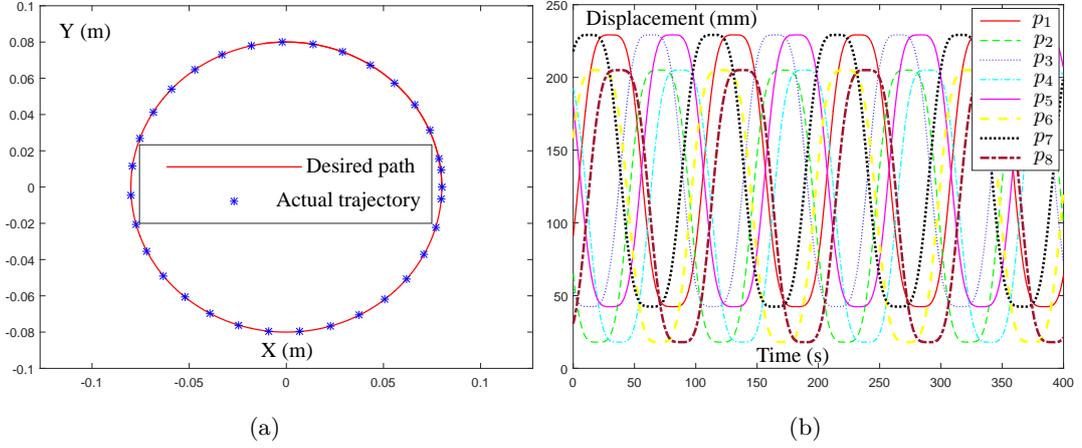


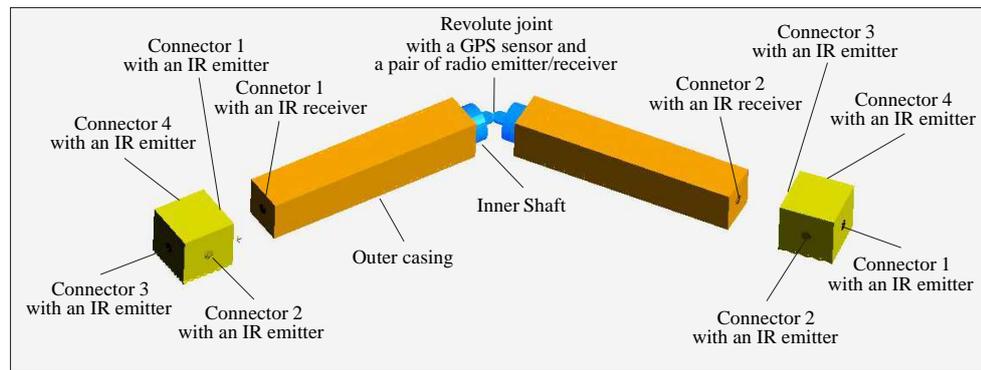
Figure 6.15: Circular Path of the End-Effector Platform and Displacements of Prismatic Actuators. (a) Circular Path Tracking. (b) Displacements Measured by Encoders.

estimated from the resolved joint angles by solving the forward-kinematics problem. Since the 4-PRP parallel mechanism is not a tailored robot for completing path-tracking tasks, the physical mechanism unavoidably has shortcomings including the facts that the nodes could not be strictly rigid, the actuators also could not be rigidly connected by the nodes, and the nodes attached to the floor also exhibit flexibility to some extent. Nevertheless, all the motion profiles of the eight actuators are smooth and continuous and the central node does move in the 2D plane, indicating that the resolved joint angles are suitable and the 4-PRP mechanism has 3 DOFs as analysed in Chapter 3.

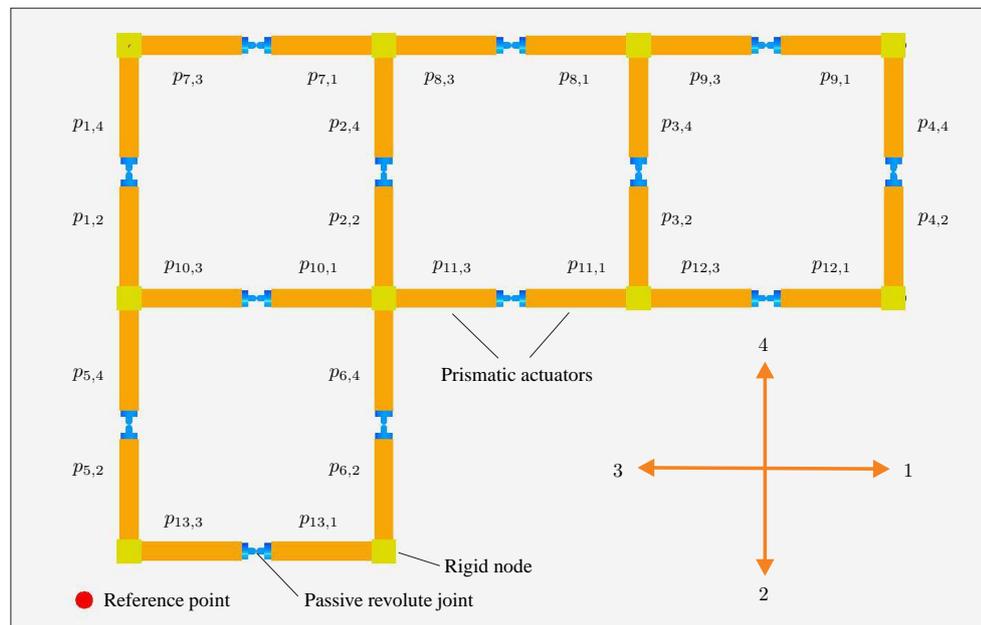
6.1.5 Generalized Computer Simulations

The single-module and two-module worm robots as well as the four-module square robot share a same CPG control strategy. Based on the above successful physical experiments, a general simulation framework is established to investigate more complex robotic structures. Simulated robots are constructed in a physics-based simulator named Webots [Michel, 2004] on the basis of physical characteristics of the developed robotic elements. Besides, virtual sensors are incorporated into the simulated robots to make the developed framework applicable to a class of amorphous robots with different configurations. Specifically, each cubic node with a mass of 0.32 kg has four connectors with numbered labels 1, 2, 3 and 4 and each connector also possesses an IR emitter.

Each strut consists of two prismatic actuators linked by a passive revolute joint. The outer casing and the inner tube of each prismatic actuator weigh 1.48 kg and 0.36 kg, respectively. Additionally, the strut has a connector with an IR receiver located at each of its ends. A global positioning system (GPS) sensor and a pair of radio emitter and receiver are integrated into the middle revolute joint. For better understanding, Figure 6.16(a) illustrates a simulated robotic strut and two cubic nodes with sensors.



(a)



(b)

Figure 6.16: Simulated Amorphous Robot. (a) Simulated Rigid Node and Robotic Strut with Sensors. (b) Illustrative Example of a Simulated Amorphous Robot.

Unlike the way of treating each square meta-module robot as an independent agent in [Yu, 2010], each node and each strut are considered as an agent that can communicate with other agents in the simulation. An illustrative example of an amorphous robotic structure is shown in Figure 6.16(b). The prismatic actuators are labelled as $p_{i,d}$ where i denotes the strut ID and d is 1, 2, 3 or 4 related to the travelling directions. For travelling along a specific direction, each occupied connector of a node sends a message including its numbered label to its coupled strut. Since each strut is connected with two nodes, a strut receives two messages including two numbered labels. For example, the strut composed of prismatic actuators $p_{1,4}$ and $p_{1,2}$ can get numbered labels 2 and 4 from its jointed nodes. By using the received numbered labels, a strut is capable of identifying whether it is in parallel with or perpendicular to a predetermined moving direction. If a strut is in parallel with the moving direction, it keeps active, otherwise it exits the program and becomes inactive to reduce computational burden. Then, each active strut sends a message to its nearest neighbouring struts located in the four cardinal directions with respect to itself. Such a message includes the ID and the GPS sensor values of a strut. After receiving its neighbouring struts' GPS messages, each active strut establishes a table to store these data which are used to determine whether a message is sent from a neighbouring strut or not during the locomotion stage. Once the GPS information table is constructed, each active strut moves to its initial position according to its phase oscillator's initial phase $\theta_i(0)$. To achieve a coordinating locomotion gait of a whole modular robotic structure, each strut updates and evolves its oscillator phase according to the following control law:

$$\dot{\theta}_i(t) = \omega + \gamma \sum_{j \in N_i} (\theta_{i,j}(t) - \theta_{i,j}^*), \quad (6.6)$$

where N_i denotes a set containing neighbouring struts of strut i . Constant $\theta_{i,j}^*$ is the desired phase difference between struts i and j . With respect to a certain travelling direction, the desired phase difference is defined as

$$\theta_{i,j}^* = \begin{cases} \pi, & \text{if } j \text{ is in front of } i, \\ -\pi, & \text{if } j \text{ is at back of } i, \\ 0, & \text{if } j \text{ is in parallel with } i. \end{cases}$$

The distance of struts i and j is determined by using the previously established GPS information table. By applying the Euler-forward difference rule to (6.6), the discrete-

time equation is obtained as

$$\theta_i(k+1) = \theta_i(k) + \delta\omega + \delta\gamma \sum_{j \in N_i} (\theta_{i,j}(k) - \theta_{i,j}^*), \quad (6.7)$$

where $\theta_i(k) := \theta_i(t = k\delta)$ with $\delta > 0$ and $k = 0, 1, 2, \dots$ representing the sampling gap and iteration index, respectively.

Let d and d' represent the moving and opposite directions, respectively. For each strut i , the following computations are performed:

$$\begin{cases} \varphi_{i,d}(k) = \text{mod}(\theta_i(k), 1.5\pi), \\ \varphi_{i,d'}(k) = \text{mod}(\theta_i(k) - 0.5\pi, 1.5\pi). \end{cases} \quad (6.8)$$

Consequently, oscillator phase $\theta_i(k)$ is converted into signals $\varphi_{i,d}(k)$ and $\varphi_{i,d'}(k)$ that are bounded between 0 and 1.5π . Finally, desired positions $p_{i,d}(k)$ and $p_{i,d'}(k)$ are calculated using activation function (6.3) which is repeated here for convenience:

$$f(\varphi) = \begin{cases} L \sin^2(0.5\pi \sin^2(0.5\pi\varphi/P)), & \text{if } 0 < \varphi < \pi, \\ 0, & \text{otherwise.} \end{cases} \quad (6.9)$$

Class diagrams of the control framework is shown in Appendix E.1. A series of simulations are performed with simulation results presented in Figures 6.17–6.23. Figure 6.17 shows a single-module robot that moves rightward. Starting from a fully-contracted state, the robot first completes an initialization process in accordance with its initial oscillator phase. After that the robot is engaged to move along direction 1 in a cyclical manner. Unlike the real single-module robot, the whole simulated one is treated as an agent for simplicity which means that the two prismatic actuators of the simulated robot have no communication during the locomotion task. The GPS values of the single-module robot is illustrated in Figure 6.17(i) from which it can be observed that the robot can keep moving rightward by using the developed control strategy.

Figures 6.18 and 6.19 present synthesized results when a square robot moves along direction 2. As shown in these figures, the square robot can achieve a worm-like locomotion from an initialized state by coordinating phases of its struts. The absolute displacement errors of the prismatic actuators are less than 0.5 mm demonstrating that the simulated robot can achieve a high positioning accuracy. There always exist gaps between the experiment and the simulation. The simulative results outperform the experimental ones as frictions between the four rigid nodes and the floor are impossible to

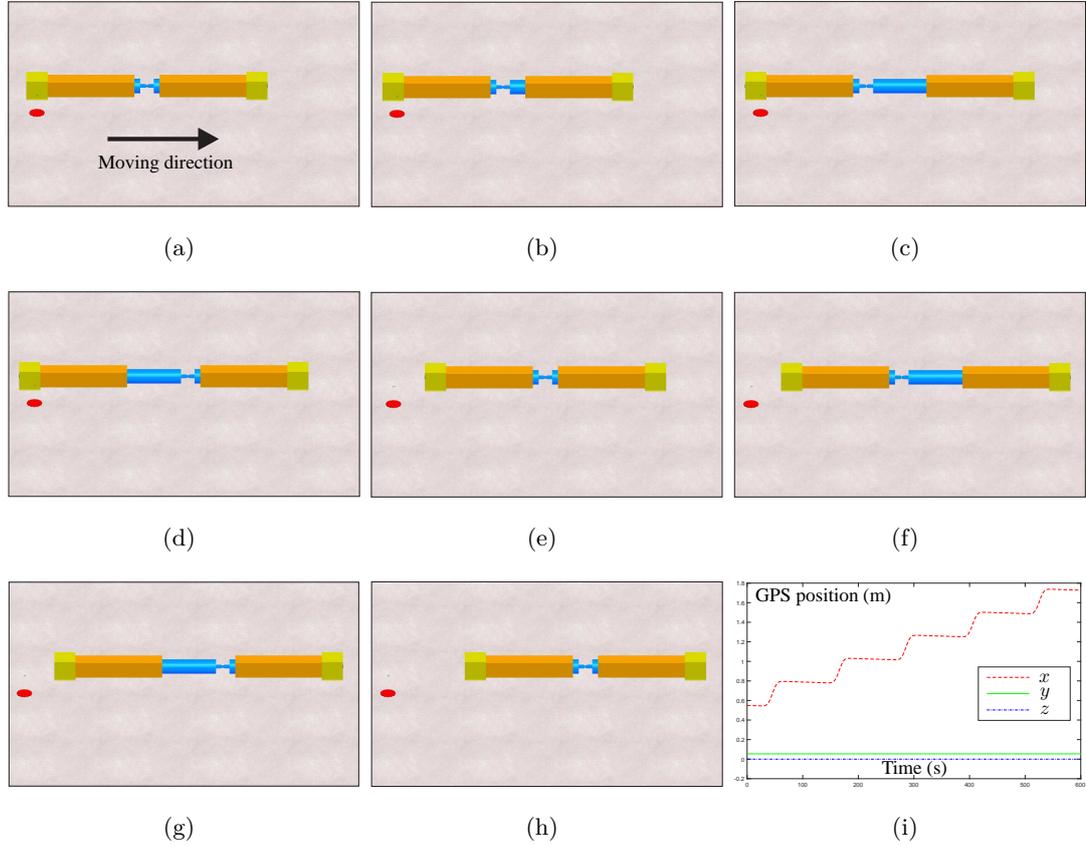


Figure 6.17: Simulation of a Single-Module Worm Robot. (a) Static State. (b) Initialized State. (c)–(h) Snapshots Captured During Coordination and Locomotion Stage. (i) Profiles of GPS Position.

be identical in the real world. Besides, the implemented cubic node cannot be coupled with a prismatic actuator in a strictly rigid way in practice.

A two-square robot is constructed and investigated to further demonstrate the applicability of the developed control framework. As can be observed from Figure 6.20, the whole robot finally moves along direction 3 and related phase differences converge to 0 rad or π rad. In addition, displacement errors of the eight active actuators are quite small with the maximum absolute value being around 0.8 mm. Generally, the maximum errors appear at the first few seconds during the phase coordination stage. Initial phases of the struts are randomly generated in the simulation and some of the combinations of initial phases cannot meet the physical constraints of a networked

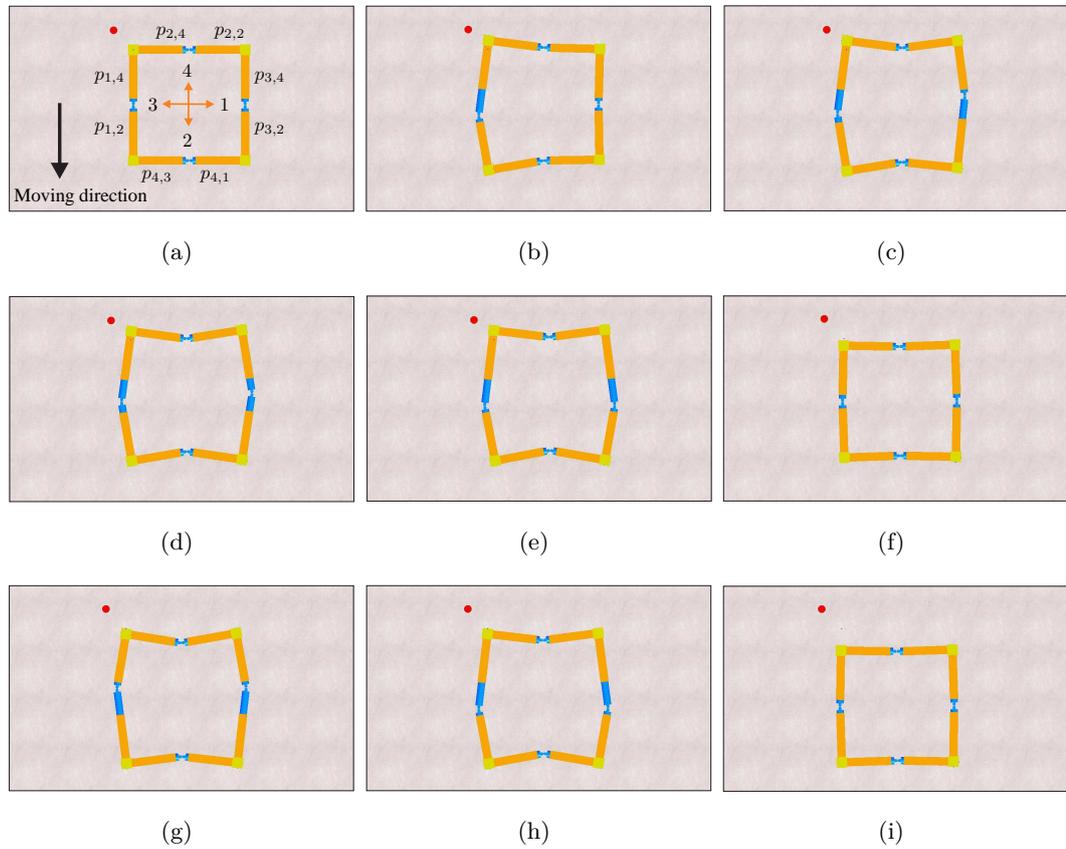


Figure 6.18: Simulation of a Square Robot. (a) Static State. (b) Initialized State. (c)–(i) Snapshots Captured During Coordination and Locomotion Stage.

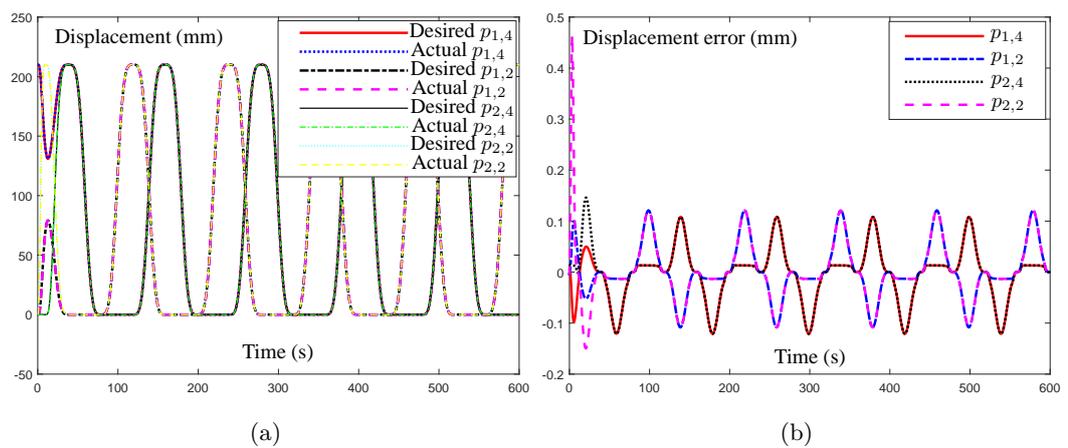


Figure 6.19: Motion and Position Error Profiles Synthesized by the Square Robot. (a) Motion Profiles. (b) Position Error Profiles.

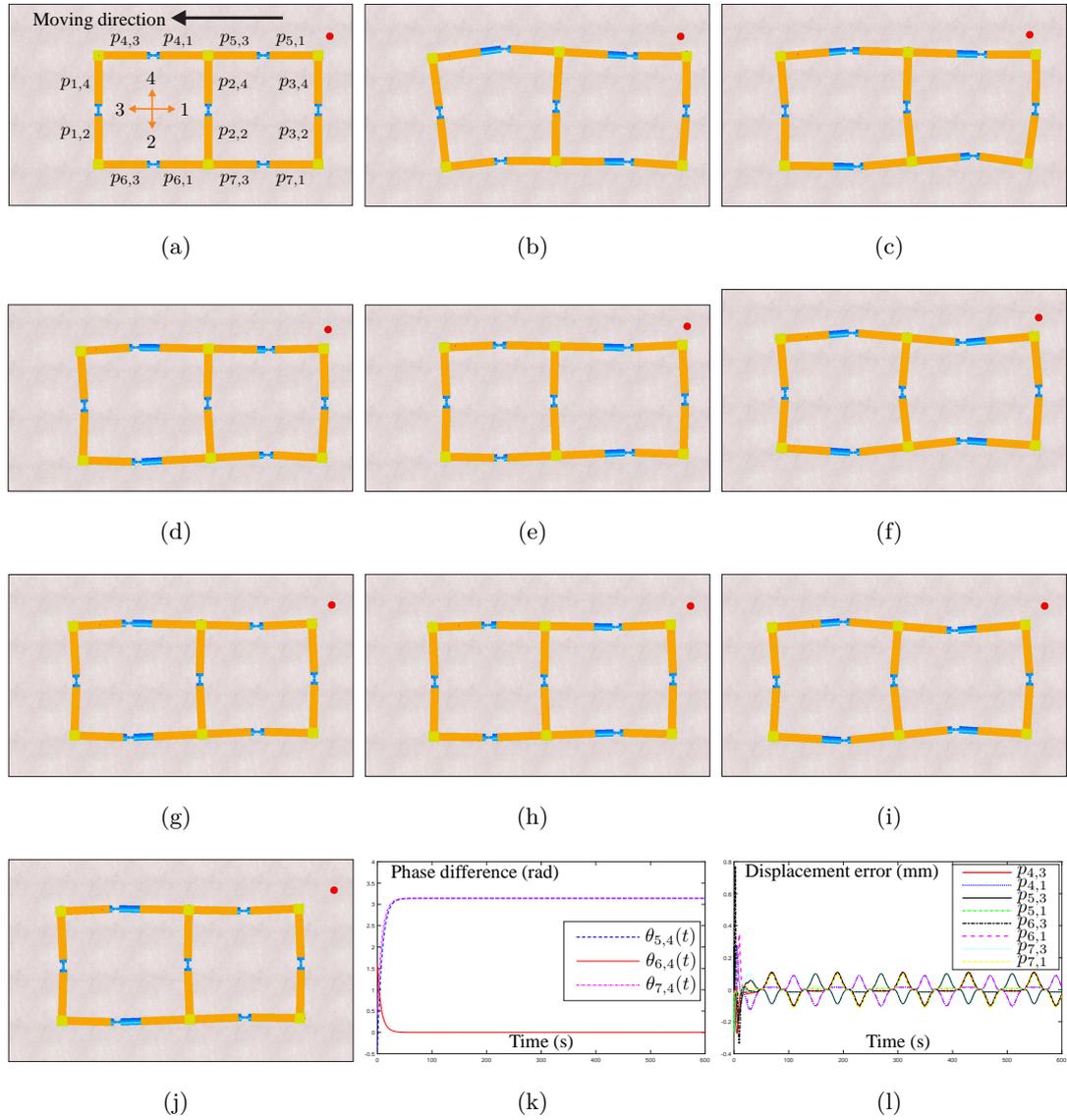


Figure 6.20: Simulation of a Two-Square Robot. (a) Static State. (b) Initialized State. (c)–(j) Snapshots Captured During Coordination and Locomotion Stage. (k) Phase Difference Profiles. (l) Displacement Error Profiles.

structure like the two-square robot. Therefore, an appropriate random initial condition is significant in this case.

Finally, a more complex four-square robotic structure shown in Figure 6.16(b) is studied. The whole robot is assigned to move along direction 4 from an initial state.

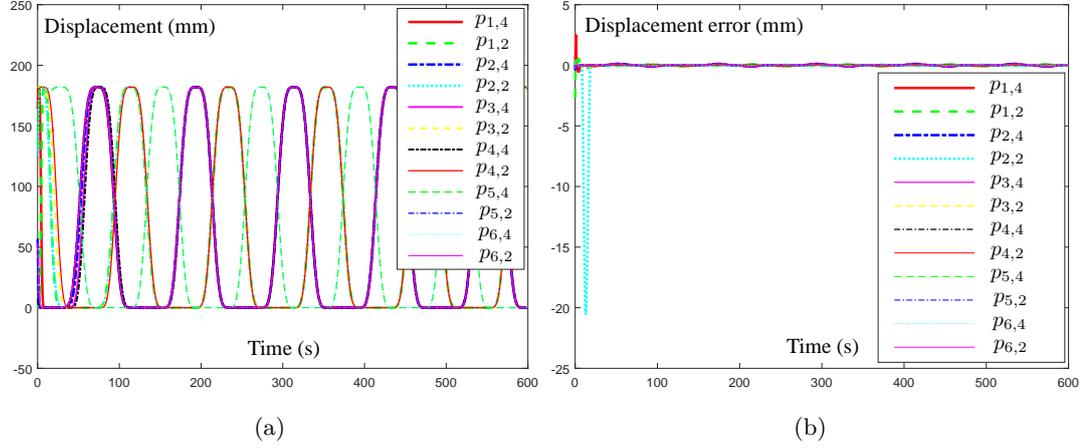


Figure 6.21: Motion and Position Error Profiles Synthesized by the Four-Square Robot without Using Passive State of Actuators. (a) Motion Profiles. (b) Position Error Profiles.

Firstly, the displacements of all the active actuators are controlled using AF (6.9) with the corresponding motion profiles and displacement errors of active actuators presented in Figure 6.21. As can be seen from the figure, the maximum absolute value of displacement error of actuator $p_{2,2}$ is around 20 mm which is too large in practical applications and this unwanted large deviation may lead to the damage of some struts and the failure of completing a task in practice. To overcome this problem, the prismatic actuator is endowed with passive compliance by resorting to a variable *state*. Specifically, when the absolute value of displacement error is less than 0.5 mm, parameter *state* = 1 and the prismatic actuator is controlled using AF (6.9); otherwise, parameter *state* = 0 and in this case the prismatic actuator is passive. Note that the desired position of a passive prismatic actuator is invalid and in the passive state the desired positions are set to be constant for better identification. Meanwhile, the position error of a passive prismatic actuator is set as 0 mm. Results for the same structure are shown in Figures 6.22 and 6.23. From these figures it can be observed that the previous undesired position deviation problem is remedied and a coordinating locomotion is acquired. Initially, the robotic structure cannot move and each controlled strut determines its state according to the position errors of its two prismatic actuators. By changing the states adaptively, the resultant maximum absolute value of displacement errors is less than 0.6 mm. Moreover, each strut updates its phase through the communication with its neighbour-

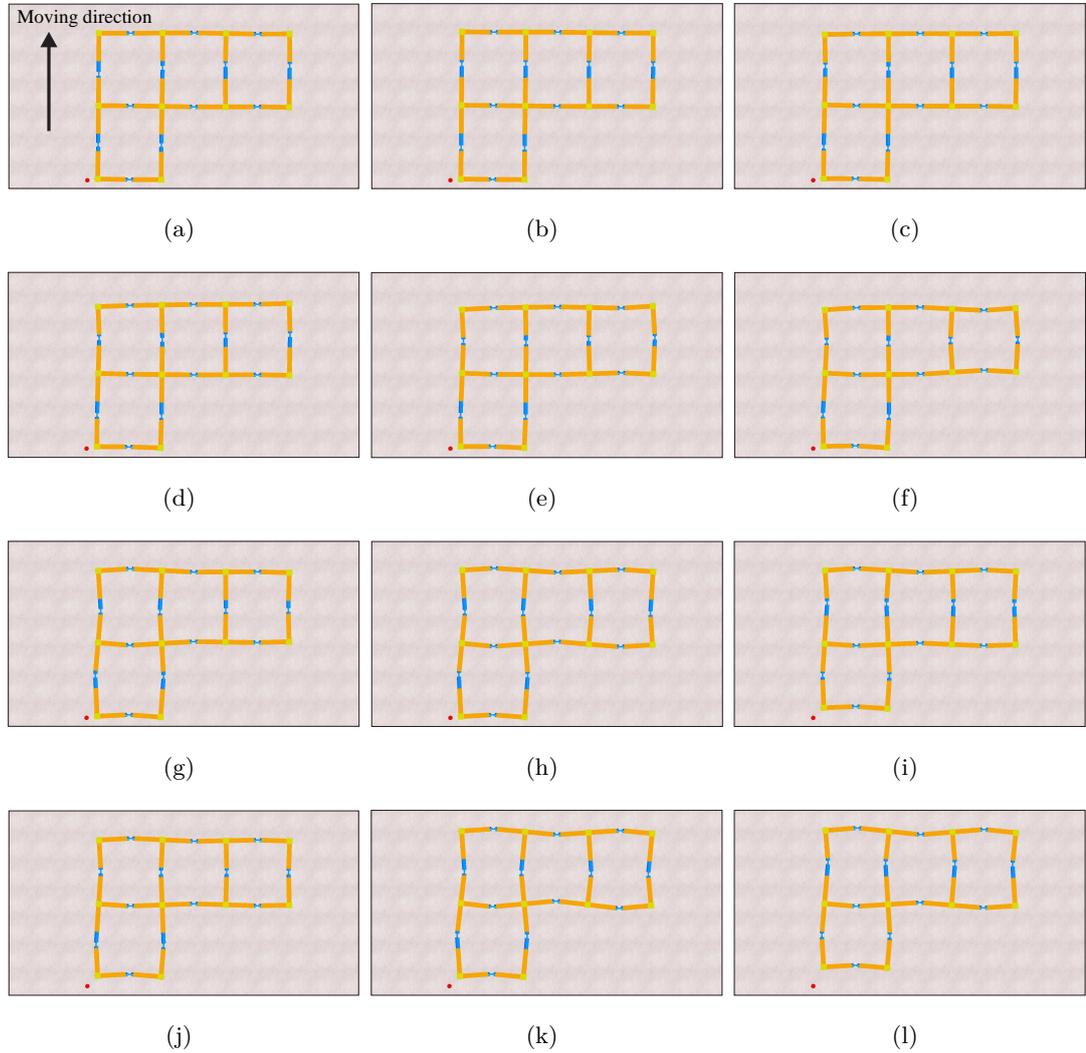


Figure 6.22: Locomotion of a Simulated Four-Square Robot with Passive State of Actuators. (a) Static State. (b) Initialized State. (c)–(l) Snapshots Captured During Coordination and Locomotion Stage.

ing controlled struts. Finally, the robotic structure can move along direction 4 in a coordinating way. The illustrative example of this four-square robot demonstrates a scenario where the passive state can be applied to reducing kinematic constraints of a robotic structure during locomotion. All of the above simulation results substantiate the performance of the implemented simulation and control framework, which lays a foundation and provides inspiring direction for future research with more physical

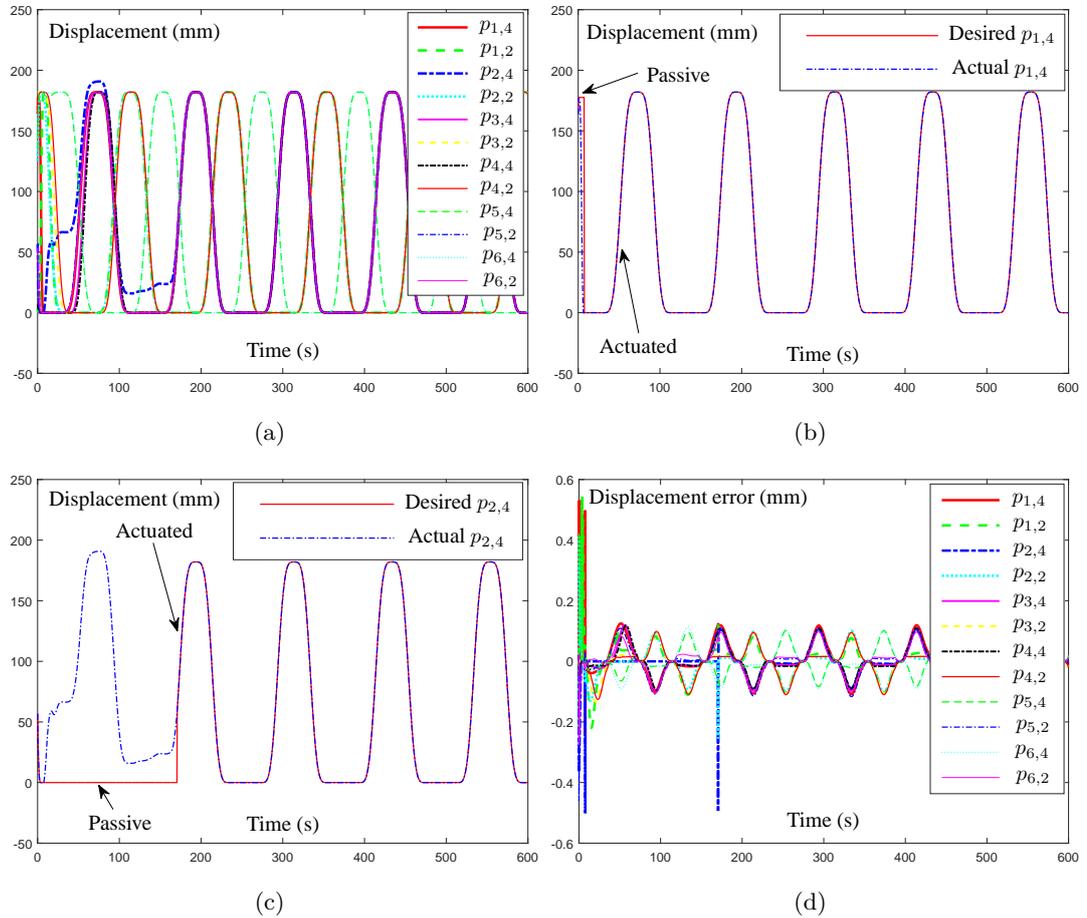


Figure 6.23: Motion and Position Error Profiles Synthesized by the Four-Square Robot with Passive State of Actuators. (a) Motion Profiles. (b) Position Error Profiles. (c) Motion Profiles. (d) Position Error Profiles.

robotic modules.

6.2 Cross-Shaped Robot

Without revolute joints, robotic structures also can be formed by using the prismatic actuators and the rigid nodes. A cross-shaped robot consisting of four prismatic actuators jointed by one rigid cubic node is shown in Figure 6.24. Each prismatic actuator has its own phase oscillator and the four actuators are coupled to coordinate movements. Experimental results are illustrated in Figure 6.25. Each actuator of the robot first

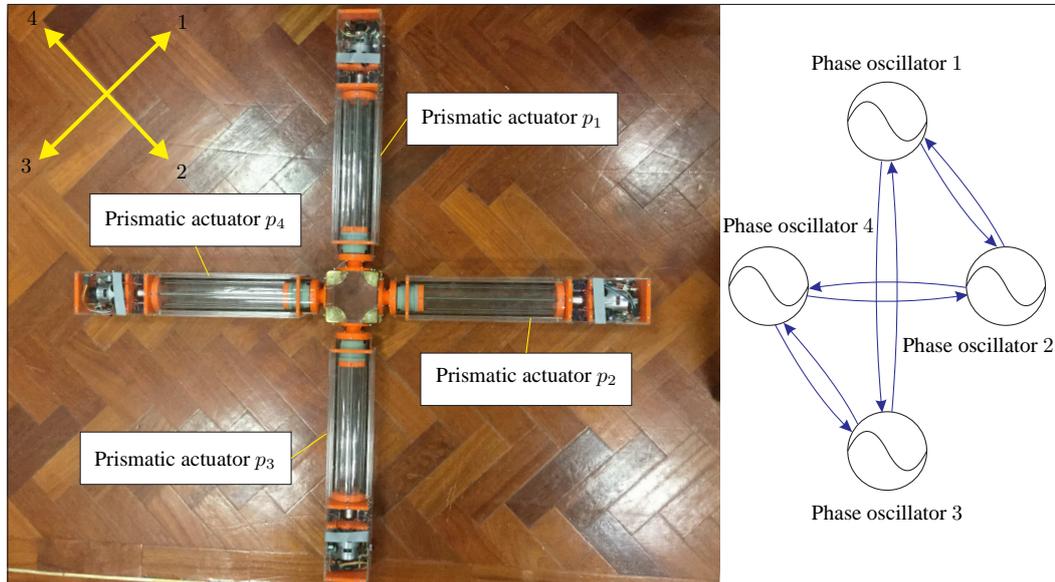


Figure 6.24: A Cross-Shaped Robot.

initializes its state according to its phase oscillator's initial phase with the initialized robot presented in Figure 6.25(b). After that, each actuator communicates with its coupled actuators resorting to a central Uno RF node and updates its phase at each time step of 0.1 s. As presented in Figure 6.25(c)–(i), the robot cannot move during the initial coordination stage. A coordinating motion of the robot is finally achieved by synchronizing the movements of prismatic actuators 1 and 2 as well as the movements of prismatic actuators 3 and 4. Meanwhile a constant delay between prismatic actuators 1 and 3 as well as prismatic actuators 2 and 4 is obtained. Finally, the robot moves along direction 1 owing to cyclical motions of its four actuators, which can be seen from Figure 6.25(j)–(o). The experimental results demonstrate the locomotion ability of the cross robot and validate again the communication capabilities of the modular robotic system. Note that centralized communication is employed in the cross robot and the previous amorphous robots as the actuators have to perform communication frequently to coordinate their movements. If distributed communication is used, communication congestion would be a problem since the Arduino Nano may have limited computation power to process information swiftly. In future work, more powerful microcontrollers can be selected to improve the performance.

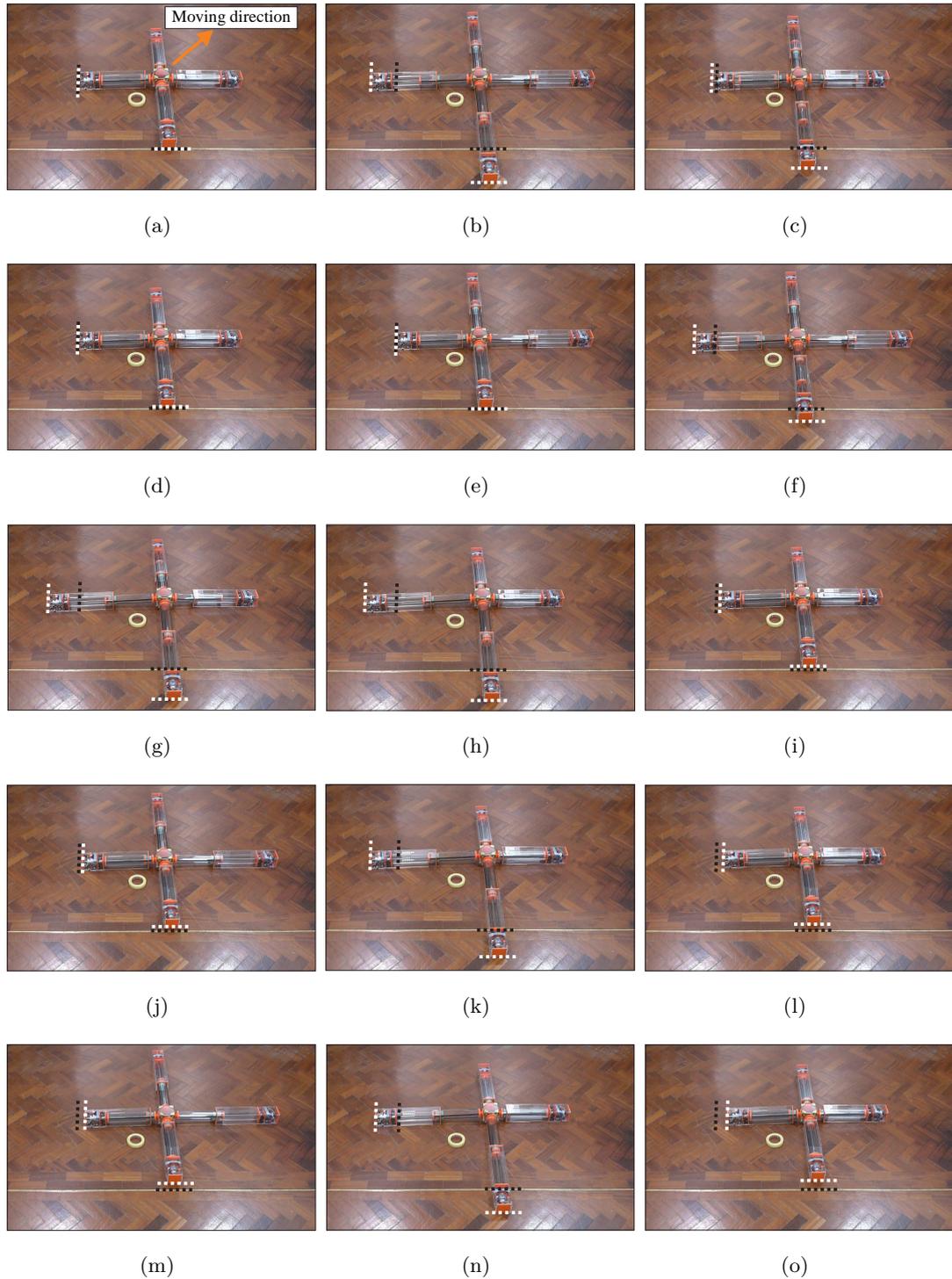


Figure 6.25: Locomotion of the Cross-Shaped Robot. (a) Static State. (b) Initialized State. (c)–(o) Snapshots Captured During Coordination and Locomotion Stage.

6.3 Simulated Table-Shaped Robot

This section concentrates on the locomotion of a table-shaped structure composed of prismatic actuators and rigid nodes shown in Figure 6.26 using three states. Unlike the amorphous robot, each strut of the table-shaped robot only has one prismatic actuator. The table-shaped robot is composed of controlled struts and inactive struts of which the controlled struts are labelled as neck, shoulder, inner leg and outer leg and the inactive struts are structural elements that are rigid all the time.

Undoubtedly, one can use a manually designed gait table with sequential steps to obtain a locomotion gait of such a robotic structure. Instead of using a gait table, a role-based control strategy using neighbour-to-neighbour local communication is developed to achieve the same purpose. Before introducing details about the control strategy, for better understanding, related parameters with detailed explanations are listed in Table 6.1. Parameters *isLocked*, *isPassive* and *state* are relevant to the following strut behaviours:

$$\begin{cases} isPassive = 1 : & \text{strut is passive,} \\ isPassive = 0 : & \begin{cases} isLocked = 1 : & \text{strut is locked,} \\ isLocked = 0 : & \text{strut is actuated,} \end{cases} \end{cases}$$

and for the actuated state, an activation function is defined with its mathematical

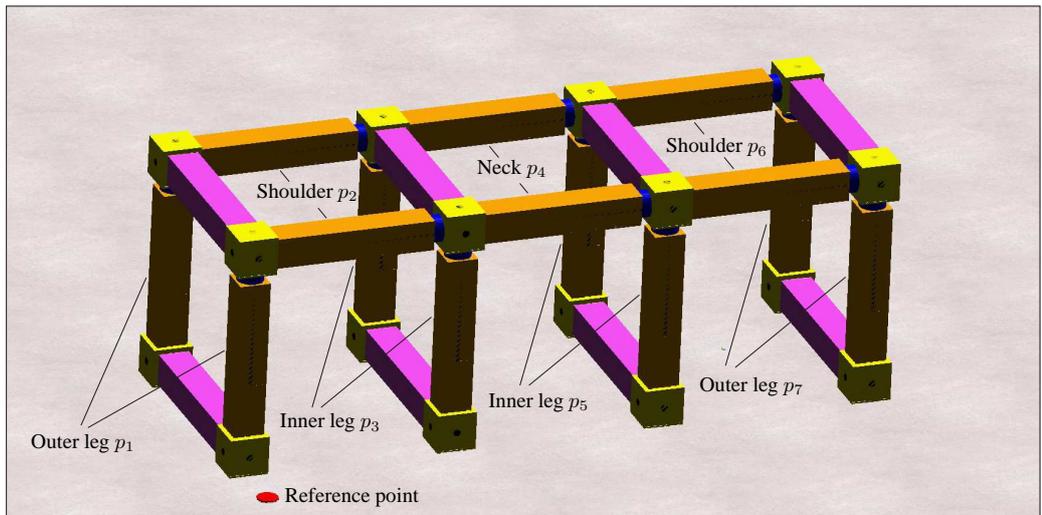


Figure 6.26: A Table-Shaped Robot Consisting of 22 Struts.

Table 6.1: Related Parameters of the Role-Based Control Strategy

Parameters	Explanations
<i>role</i>	Outer leg, inner leg, shoulder or centre with value being 1, 2, 3, 4 respectively
<i>state</i>	Fully contracted or fully extended with value being 0 or 1 respectively
<i>strutID</i>	Identity number of active struts with value being 1, 2, \dots , 7
<i>isLocked</i>	Boolean variable with value being 0 or 1, strut is in locked state when $isLocked = 1$
<i>isPassive</i>	Boolean variable with value being 0 or 1, strut is in passive state when $isPassive = 1$
<i>isInitialized</i>	Boolean variable with value being 0 or 1, initialization stage is completed when $isInitialized = 1$
<i>stepCounter</i>	Counter of time steps
<i>commCounter</i>	Counter of communication times

expression depicted as:

$$!state \times L + (-1)^{!state} \times L \sin^2(0.5\pi \sin^2(0.5\pi t/T)),$$

where L denotes the amplitude. This means that the strut contracts and extends when $state$ equals 0 and 1, respectively. Therefore, state transitions can be achieved by controlling the values of parameters $isLocked$, $isPassive$ and $state$ and an effective controller for each strut should have the ability to update these values appropriately according to messages received from neighbouring struts. The whole control process is divided into two stages: an initialization stage and a cyclical motion stage. The initialization stage initializes the structure from its fully contracted state shown in Figure 6.26 to the initial state of the cyclical motion stage. Figure 6.27 shows the control flowchart during the initialization stage. After setting initial values for some parameters, each strut conducts communication to send messages to its neighbouring struts. Then, each strut determines its own role. After that, struts with different roles process messages received from their neighbouring struts and take actions. Once the initialization stage is finished, the robot repeats cyclical motions based on the control

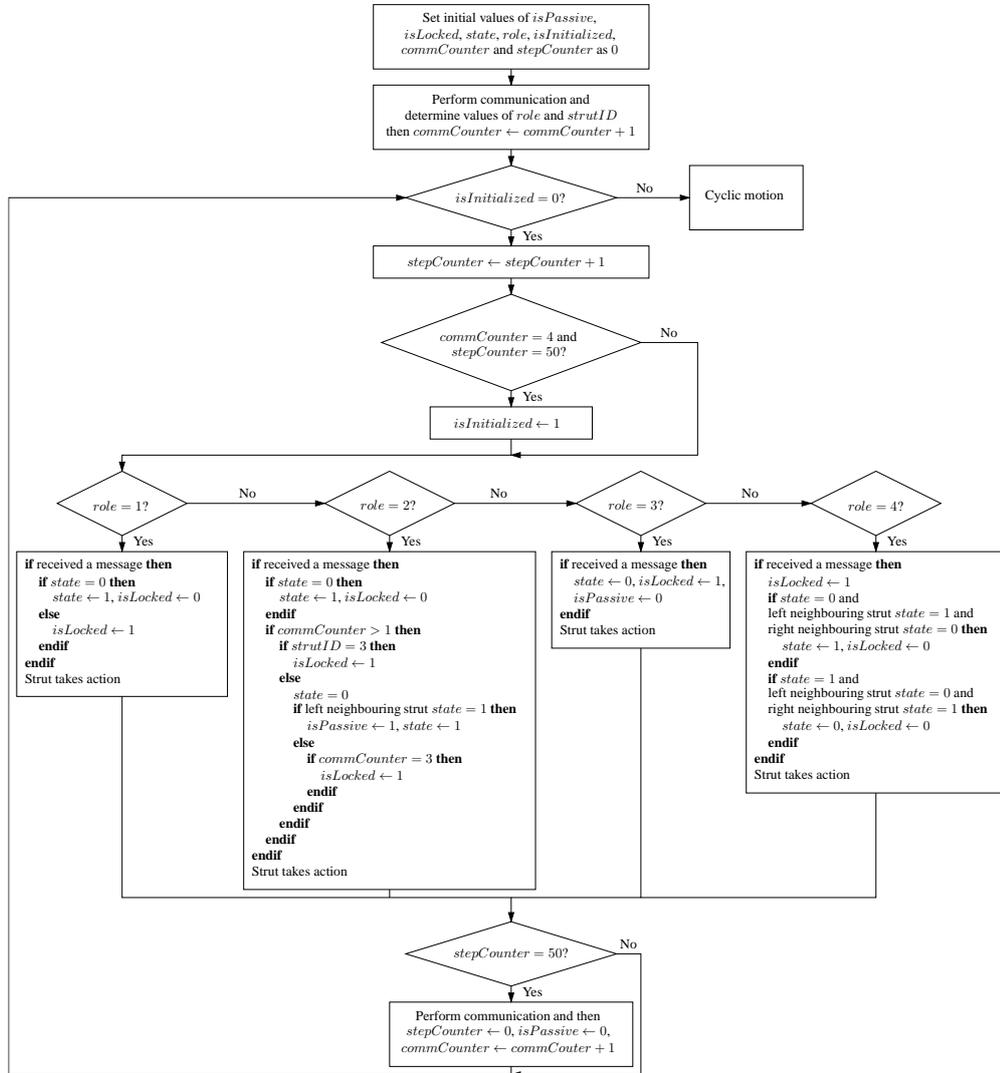


Figure 6.27: Flowchart of the Role-Based Control Strategy in the Initialization Stage.

flowchart illustrated in Figure 6.28. In the cyclical motion stage, each strut also takes action by processing its received messages. Appendix E.2 shows the class diagrams of the role-based control strategy.

Simulation results are presented in Figures 6.29–6.31. The initialization stage is completed by treating the fully contracted state as the initial state and taking four steps as shown in Figure 6.29(a)–(d). Then the final state of the initialization stage is treated as an initial state of the cyclical motion stage. In this way, after twelve steps the robot as presented in Figure 6.30 (a)–(l) moves over a distance and returns to its

6.3 Simulated Table-Shaped Robot

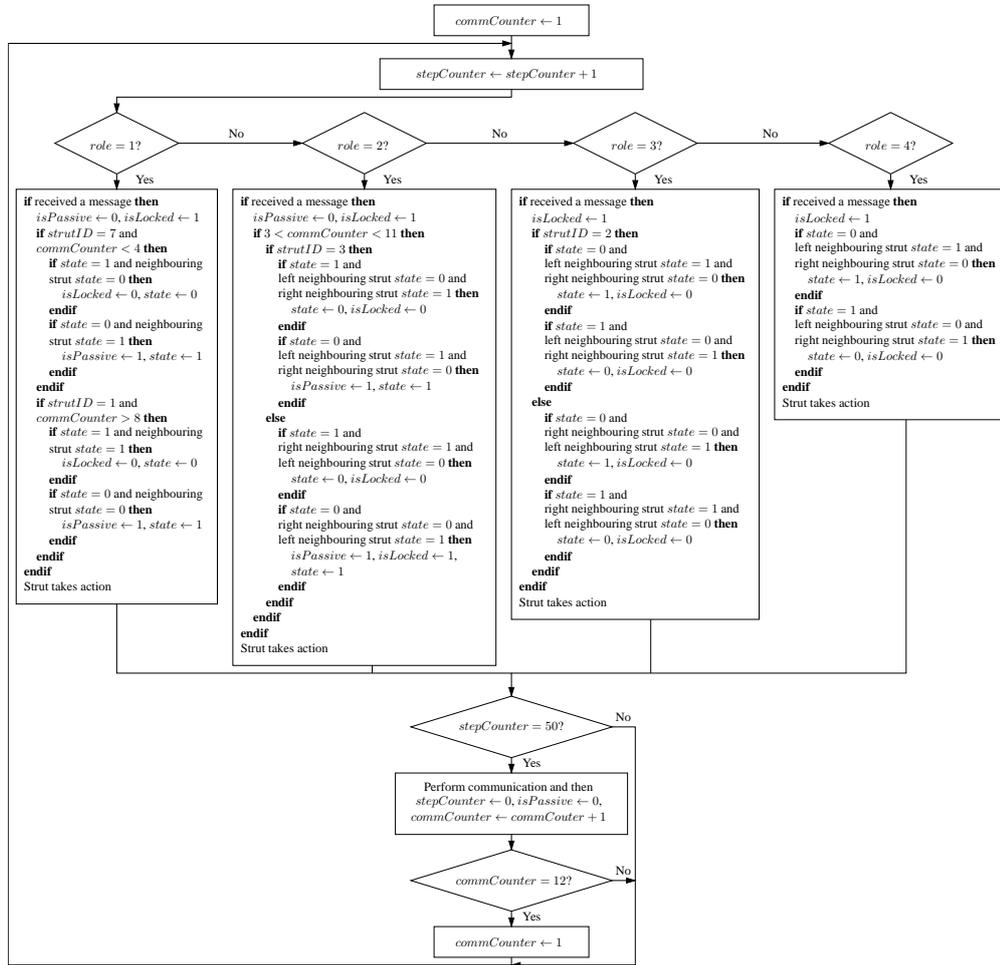


Figure 6.28: Flowchart of the Role-Based Control Strategy in the Cyclic Motion Stage.

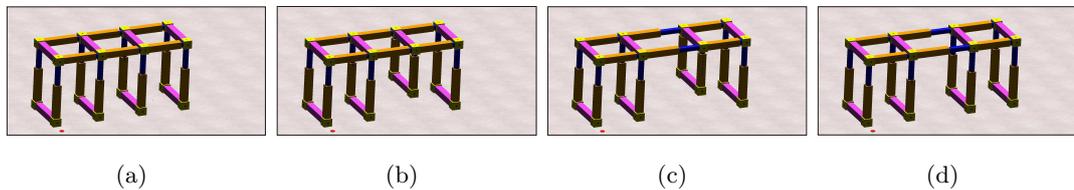


Figure 6.29: Initialization Stage of the Table-Shaped Robot. (a)–(d) Initialization Steps 1–4.

initial state of the cyclical motion stage. Evidently, by repeating the same twelve steps, the robot can keep walking along a certain direction. Differing from using conventional

6.3 Simulated Table-Shaped Robot

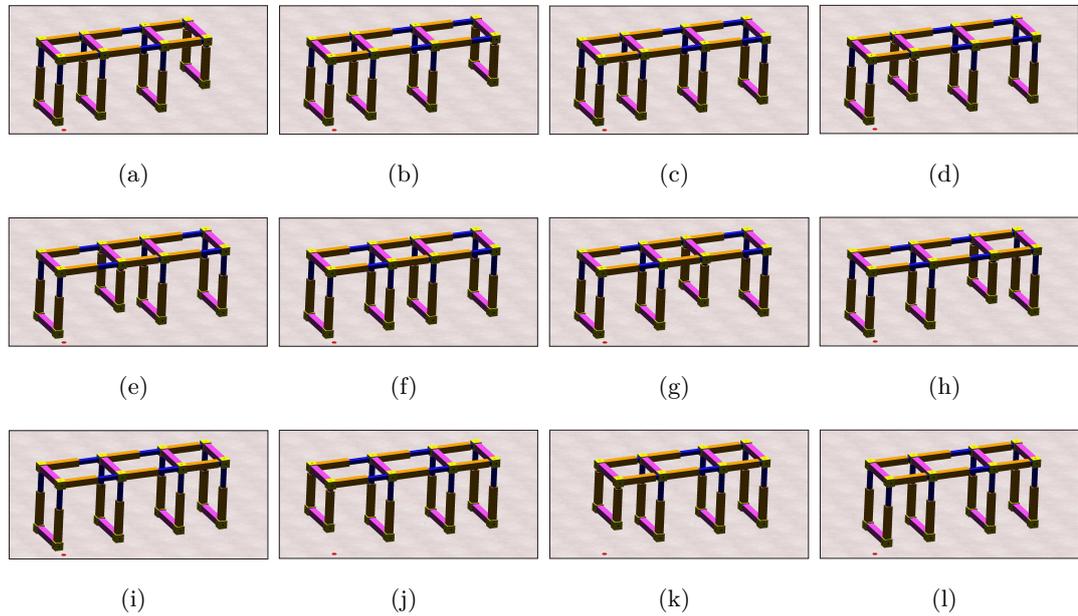


Figure 6.30: Cyclical Motion Stage of the Table-Shaped Robot for Walking Rightward. (a)–(l) Motion Steps 1–12.

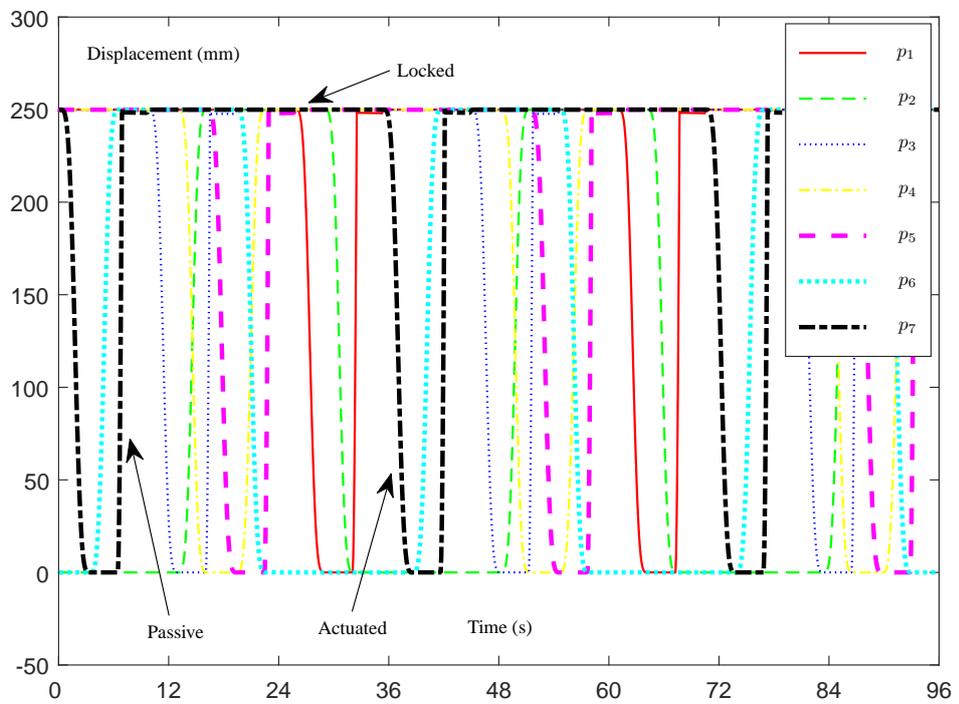


Figure 6.31: Actuator Motion Profiles of the Table-Shaped Robot.

struts, a passive state is utilized to extend the strut resorting to the gravity force. Figure 6.31 shows the position profiles of seven prismatic actuators during the first 96 s of the cyclical motion stage. All of the above results demonstrate that, based on local neighbour-to-neighbour communication, the table-shaped robot can walk using three states and state transitions achieved by the role-based control strategy.

6.4 Conclusions

This chapter has investigated the control and locomotion of some physical and simulated modular robotic structures composed of the proposed prismatic actuators. A class of amorphous robots was first studied including a single-module robot, a two-module robot and a square robot. These robots are comprised of prismatic actuators, passive revolute joints and rigid nodes and can be controlled using a same CPG control strategy to achieve a coordinating locomotion gait. Centralized communication method was employed in these experiments as the coordination between prismatic actuators is quite often, which leads to difficulties in synchronization. Then, a physical 4-PRP parallel mechanism was built to validate the theoretical analysis conducted in Chapter 3. As demonstrated by the experimental results, the parallel mechanism has mobility to let the central node follow a circular path when actuating the eight actuators. Since the outside connector nodes are fixed, this is an extreme case where the structure is physically constrained. Based on the successful experiments, an extended simulation was performed to show a case where the passive state of prismatic actuators can be used to release physical constraints of a structure during the coordination process. The control framework developed for the extended simulations is scalable as it is applicable to various amorphous robotic structures. After that, robotic structures formed by rigidly connecting prismatic actuators were investigated. A cross-shaped robot was first built and the same CPG control strategy was applied to this robot for achieving a locomotion gait. Then, a simulated table-shaped robot was constructed in Webots and a role-based control method was implemented to let the robot walk through distributed neighbour-to-neighbour communication. Each active prismatic actuator labelled as out leg or inner leg of the table-shaped robot achieved state transitions between actuated, locked and passive states. All of the simulative and experimental results substantiated the efficacy of the developed modular robotic system.

CHAPTER 7

Conclusions and Future Work

The design, manufacture and control of a strut-type modular robotic system have been detailed in this thesis. This chapter details conclusions drawn from the current research findings and points out directions for further research.

7.1 Assessment of Research Objectives

Six research objectives were outlined in Chapter 1. This section evaluates the extent to which they have been achieved during this research.

7.1.1 Literature Review on Related Work

Mechatronics design and hardware implementation of typical robotic modules have been analysed and compared to gain prototyping techniques from the state-of-the-art robotic modules. Existing popular control strategies in modular robotics have been studied to lay a foundation for the control of the developed strut-type modular robot. Popular communication methods have been discussed and robots with passive compliance and actuators with two or more working states have been reviewed.

7.1.2 Development of Connectivity Strategy

Instead of conventional compliant nodes, rigid connector nodes have been exploited to construct robotic structures. Meanwhile, passive compliance was added to the prismatic actuators and passive revolute joints were placed in the middle of two prismatic actuators to release some physical constraints. Kinematics of structures formed using the conventional and the proposed connectivities is comparatively conducted. The locomotion and deformation capabilities of robotic structures formed using the two different connectivities are also compared and analysed.

7.1.3 Design and Fabrication of Modular Elements

Four design schemes have been put forward to develop a prismatic actuator that can be actively controlled and passively respond to external forces. Two physical prototypes were first fabricated based on two of the design schemes for selecting electronics components and checking the feasibility and performance of the developed physical systems in terms of their actuation, locking and passive compliance capabilities. Eight prismatic actuators and elements for interconnection including four rigid connector nodes and four

passive revolute joints were finally manufactured resorting to 3D printing technologies to achieve rapid prototyping of the investigated strut-type modular robotic system.

7.1.4 Investigation of Communication Strategies

For a central Uno node, IR communication has been employed to receive remote control commands from a human operator. RF communication using nRF24L01+ modules has been implemented to exchange messages from the Uno node and prismatic actuators with Nano nodes. Both centralized and distributed communication strategies were implemented and tested for different tasks.

7.1.5 Construction of Simulated Modular Robots

Robot modelling has been completed using a trial version of a physics robot simulator named Webots which is based on open dynamics engine. The physics characteristics of the simulated robotic elements were constructed based on their real counterparts to ensure a relatively high fidelity. In addition, virtual sensors including a GPS sensor and IR emitters/receivers were incorporated into the simulated nodes and struts.

7.1.6 Implementation of Control Strategies for Locomotion

A PID controller was developed for controlling positions of a prismatic actuator. Programs for driving the rotary encoder, nRF24L01+ module, MD10C and L9110S motor drivers were implemented. A CPG control strategy has been implemented and applied to numerous modular robotic structures to achieve a coordinating worm-like locomotion. A scalable CPG controller and a role-based control method were developed for simulated modular robots with more complex structures.

7.2 Conclusions

The following conclusions are relevant to the specific tasks of work performed.

7.2.1 Design

- 1) The overall design of the prismatic actuator satisfies the design specifications presented in Chapter 4. The selected motors and fabricated mechanical locker

enables the prismatic actuators to exhibit three states, i.e., the actuated, locked and passive states. Thus, the concept of a three-state prismatic actuator has been physically proved. However, the actuation force is not enough for 3D cases to support more modules. The passive state depends on a back-drivable lead screw which is coupled with a DC motor directly. Besides, the locking force may become weak for some applications and the prototyped prismatic actuator is bulky for some scenarios.

- 2) The employed position sensor is an incremental rotary encoder which is capable of measuring relative displacement of the prismatic actuator but it is unable to sense the absolute position of a prismatic actuator especially in the case where the power is off. This means that the actuator must be reset to ensure the correctness of the positioning information when it runs out of power.
- 3) The adopted Arduino Nano microcontroller and nRF24L01+ module is suitable to implementing a centralized or distributed communication network for accomplishing some research tasks. The microcontroller can handle the real-time control of the developed robotic system. However, for message-based control method which is closely dependent on a high frequent communication between modules, distributed communication is difficult to implement on hardware. This may result from the limited computation ability of the Arduino microcontroller which cannot process messages swiftly. Another possible reason is that communication congestion appears among the nRF24L01+ modules.
- 4) The designed passive revolute joint is simple and easy to fabricate. It is also strong enough and competent to serve as a modular unit for interconnection of two prismatic actuators. Since it is purely passive, the motion of the revolute joint cannot be controlled directly.
- 5) The fabricated rigid connector node is simple and functional enough to achieve manual reconfiguration for constructing modular robotic structures by connecting with prismatic actuators. This manual way of reconfiguration brings limitations of the autonomy of the modular robotic system.
- 6) The employed connectivity strategy is effective to form strut-type modular robotic structures. By using the connectivity strategy, connector nodes and passive

movable joints are isolated, which improves the implementability of the modular robotic system as compared with the conventional ideal compliant nodes. However, the square robot formed using the connectivity strategy moves slower than the conventional square robot with compliant nodes. This is caused by the fact that the square robot formed using rigid nodes and passive revolute joints would exhibit buckling shapes leading to that the active struts are not in parallel with the specified moving direction.

7.2.2 Modelling

- 1) The simulated parallel mechanisms created in the Matlab GUIs are suitable for comparing the two different mechanisms and verifying the kinematics analysis results. These parallel mechanisms are simplified robot models without considering physics. Nevertheless, these models facilitate the investigation and comparison of their performance in terms of mobility, manipulability and workspace.
- 2) The simulated robots in Webots exhibit almost the same behaviours of the physical real robots. However, a gap between a simulated robot and an actual counterpart would always exist as uncertain factors in the real world cannot be fully considered in a simulated environment. Besides, in the simulation side, there are a lot of combinations of significant parameters need to be tuned, which complicates the modelling process.

7.2.3 Control

- 1) The developed PID controller has a quite well performance in terms of its step response and positioning accuracy. A very slight instability phenomenon appears in the robotic structures' coordination and locomotion experiments. In these experiments, a prismatic actuator has to communicate with its paired prismatic actuators and then update its phase to obtain a new setpoint for its PID controller at each time step of 0.1 s. However, communication sometimes needs more time to be completed and data loss is unavoidable, which results in a fact that the update rate of the PID controller's setpoint signal varies in practice.
- 2) The developed CPG control strategy is capable of coordinating motions of prismatic actuators and achieving a worm-like locomotion of a robotic structure as

a whole. Variables like the intrinsic frequency of each phase oscillator and the convergence parameter are of great significance to obtaining satisfactory experimental results. These parameters must be tuned in an empirically manner.

- 3) The distributed role-based control method is effective to make the simulated table-shaped robot walk depending on neighbour-to-neighbour communication. It provides a case where three states (i.e., actuated, locked and passive states) can be employed to generate a locomotion gait of a robotic structure composed of the developed prismatic actuators. However, its efficacy needs to be further verified using experiments by producing more physical prototypes in the future.

7.3 Future Directions

This research has provided a starting point for undertaking research into a strut-type modular robotic system. This section points out additional work that might be undertaken to improve the further research.

7.3.1 Design

- 1) The actuation force of the actuator can be improved by using a power source with a larger voltage as the nominal voltage of the 775 DC motor is 36 V and the current batteries used can offer only 16.8 V. Meanwhile, a more powerful DC motor can be used to replace the 775 DC motor to further increase actuation force. Decoupling the DC motor from the lead screw would enhance the passive compliance capability of the prismatic actuator. A potential solution is to use a clutch and then a smaller force is required to make the prismatic actuator move passively. This means that the prismatic actuator in its passive state would exhibit timelier response to the environment. Replacing the mechanical locker with another locking mechanism such as an electro-permanent magnet based mechanism may provide a stronger locking force. Like the designed mechanical locker, the electro-permanent magnet also consumes energy during the transitions between engaged and disengaged states. However, the electro-permanent magnet solution tends to be more difficult to implement as compared with the designed mechanical locker. Modifying and miniaturizing the current design would bring more applications

of the proposed modular robotic system, although it is challenging to fabricate mechanical parts in a micro- or nano-scale.

- 2) Mounting an absolute rotary encoder is a solution to measure absolute position of a prismatic actuator. This will get rid of the requirement of resetting an actuator when it runs out of battery, though it would increase the overall cost.
- 3) The usage of a more powerful microprocessor would improve the computation efficiency and may solve the communication congestion problem appeared during the coordination of modules.
- 4) The passive revolute joint can be improved to achieve three working states like the three-state prismatic actuators. Then, the unwanted shapes of the robots during locomotion can be adjusted by actuating the revolute joint.
- 5) The rigid connector node can be equipped with active connector mechanism to achieve automatic reconfiguration of the modular robotic system, which would enhance the autonomy of the robotic system.
- 6) Designing more types of modular elements would result in more structures that can be formed. 3D structures can be explored by adding elements such as ball-and-socket joints and universal joints. The modular robotic system can be equipped with more sensors to do more effective interactions with humans and its working environments.

7.3.2 Modelling

- 1) Conducting robot modelling process for the parallel mechanisms using Webots or other physics-based robot simulator will improve the fidelity of the simulated parallel mechanisms.
- 2) Developing evolutionary algorithms to optimize significant parameters would narrow the gap between the simulated environment and the real world. An accurate simulated robot model would facilitate and speed up the controller development process as creating simulated robots is time-saving and cost-effective as compared with fabricating physical prototypes.

7.3.3 Control

- 1) The PID gains can be evolved using a genetic algorithm and a self-adaptive PID controller like a Fuzzy-PID controller could be implemented to adjust the PID gains in real time.
- 2) Parameters associated with the CPG control strategy could be optimized using evolutionary algorithms to improve the performance of the robotic system. Besides, such evolutionary algorithms can be developed to evolve controller and morphology of robotic structures simultaneously.
- 3) Investigating more complex distributed communication networks and manufacturing more modules to verify the applicability of the developed role-based control method to a physical table-shaped robot.

APPENDIX A

Paper Published

This section contains published work from this thesis.

- Li, W., Richardson, R.C., & Kim, J. (2016). A novel strut-type modular robotic structure using rigid node. In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics*, 261–268.

A Novel Strut-Type Modular Robotic Structure Using Rigid Node

Weibing Li, Robert C. Richardson and Jongrae Kim

School of Mechanical Engineering, University of Leeds, LS2 9JT, United Kingdom
{*mnwl, r.c.richardson, menjkim*}@leeds.ac.uk

Keywords: Modular Robots, Rigid Nodes, Central Pattern Generators, Distributed Control, Physics-Based Simulation

Abstract: This paper proposes a novel way of constructing strut-type modular robotic structures to avoid some difficulties of designing and implementing ideal compliant nodes. Rigid nodes are employed to replace the ideal compliant nodes and to reduce the structural complexity while the feasibility of hardware implementation is dramatically improved. To release some kinematic constraints caused by the rigid nodes, we introduce robotic struts that consist of two prismatic actuators linked by a passive revolute joint. Physics-based robot models are constructed using a robot simulator. A scalable distributed control method is implemented using coupled central pattern generators. And, for comparison, the same control method is applied to conventional and the proposed strut-type modular robotic structures. Simulation results show that the proposed strut-type structures have several advantages over the conventional ones including less number of passive joints and shape-maintenance property.

1 INTRODUCTION

A modular robotic structure (MRS) consists of separate identical or different modules that can attach to or detach from each other to make the whole robot achieve manual or self-adaptive reconfiguration (Yim et al., 2007). One outstanding characteristic of MRSs is their shape-changing capability. In modular robotics, the following two shape-changing methods have been widely studied:

- **Reconfiguration:** an MRS can change its configuration (i.e., connectivity) by attaching and detaching robotic modules manually or self-adaptively.
- **Deformation:** an MRS with a specific configuration can change its shape without changing the connectivity of robotic modules.

Reconfiguration endows an MRS with a wide range of robotic structures which can emulate conventional monolithic robots and are suitable for different tasks under different working environments. Being different from reconfiguration, deformation can be used to adjust the MRS shape to internal and external forces exerted on the robotic structure.

To utilize the benefits generated from reconfiguration and deformation, numerous MRSs have been designed and developed. Most of the existing MRSs have block-like modules fitted with only revolute actuators (Zhang et al., 2003; Kurokawa et al., 2006; Østergaard et al., 2006; Salemi et al., 2006; Yim

et al., 2007; Sprowitz et al., 2014), which are suitable for reconfiguration. Relatively less attention is given on strut-type MRSs using prismatic actuators (Curtis et al., 2007; Lyder, 2010; Yu, 2010; Zagal et al., 2012), which are adept at deformation. A design case of using both prismatic and revolute actuators can be found in (Baca et al., 2014). Usually, a revolute actuator can only rotate around its axis, while a prismatic actuator can elongate its body to reach some positions in the workspace directly. Prismatic actuators can form parallel truss-based structures that provides inherent stability. Hence, prismatic actuators may be more suitable for industrial activities such as load transportation than revolute actuators (Ramchurn et al., 2006).

In recent years, researchers have designed different strut-type MRSs using prismatic actuators for investigating their deformation and locomotion capabilities. Ideally, a strut-type MRS should have an ideal node connector mechanism which can connect numerous robotic struts. More importantly, robotic struts can rotate around the node center with some passive three degrees-of-freedom spherical joints and robotic struts connected by a same node should have a common center of rotation. In most simulations of the existing literature, robotic struts are jointed by point-like ideal nodes. Such point-like ideal nodes is helpful to reducing the complexity of kinematics (Hamlin and Sanderson, 1998) and providing compliant capability

for strut-type MRSs, however, physical implementation of such ideal nodes is highly difficult and even impossible (Lyder, 2010).

A lot of efforts have been made to design and implement an ideal compliant node. In the Tetrobot project (Hamlin and Sanderson, 1998), a centric multilink spherical (CMS) joint mechanism was designed, which can let the extended lines of its connected struts intersect at a same point. Such a CMS joint design can not only make a homogeneous design of Tetrobot possible but also simplify the kinematics of structures. However, CMS joint cannot be used to form chain-type structures. Additionally, CMS joints tend to become too weak to sustain a massive robotic structure. Moreover, the Tetrobot is hard to reconfigure as adding or removing struts usually need to disassemble the whole robotic structure (Lyder, 2010).

A workaround for constructing strut-type MRSs is to use struts that have their own center of rotation on the node surface. In such a solution, ball-and-socket joints and universal joints are commonly employed. In (Yu, 2010; Lyder, 2010), passive ball-and-socket joint based designs were adopted to provide compliant movements. Another well-known connector mechanism is the one designed by NASA for a 12-tetrahedron (12-TET) robot (Curtis et al., 2007). Specifically, NASA researchers developed two types of connectors, one is a wheel-shaped node for locomotion and the other one is a special payload node. The nodes endow itself with compliant flexibility by using passive universal joints. It is worth pointing out that the above non-ideal compliant nodes do simplify the physical implementation but make the kinematic analysis more complex (Lyder, 2010). This may be the reason why prototypes (e.g., Odin and 12-TET modular robots) using such compliant nodes are difficult to control and can only complete simple locomotion and/or deformation tasks.

Apart from the node design, another challenge in modular robotics is to construct a unified control framework that is both suitable to different modular robotic systems and scalable to robot size. Due to the modularity of modular robots, distributed control methods are intrinsically more scalable than centralized control methods. The scalability of a phase-automata based distributed control method developed for chain-type PolyBot modular robots has been validated by using a physical snake robot with 55 modules (Zhang et al., 2003). In (Yu, 2010), a scalable control framework for realizing coordinating locomotion of amorphous MRSs was established, analyzed and verified. Such a scalable control framework is based on a central pattern generator (CPG) based distributed control method.

Based on the above understanding, we focus on using rigid nodes for constructing strut-type MRSs to avoid the difficulty of implementing ideal compliant nodes. Unlike passive compliant nodes, struts connected using rigid nodes can not rotate passively around the nodes. Rather, by connecting struts rigidly using rigid nodes, the extended lines of struts intersect at a same point, which simplifies the kinematics complexity. To release some kinematic constraints caused by using rigid nodes, we use robotic struts that are comprised of two prismatic actuators linked by a passive revolute joint. For validating the proposed way of constructing strut-type MRSs, a scalable distributed control method is developed inspired by (Yu, 2010).

This paper is organized as follows: firstly, a novel strut-type modular robotic structure is presented; secondly, a control method using central pattern generator is designed based on a moving principle of which the performance is demonstrated by a prototype; thirdly, locomotion and deformation capabilities are verified by simulations; and finally, conclusions are presented.

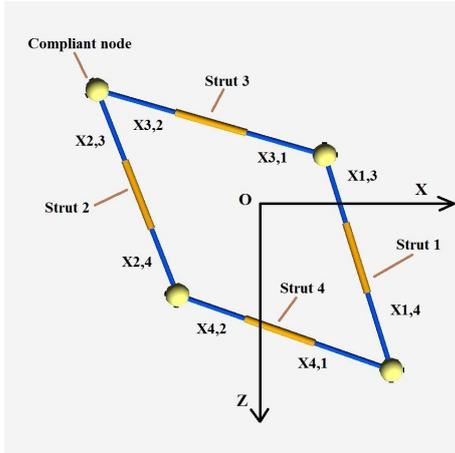
2 ROBOTIC STRUCTURE & CONTROL

In this section, a novel strut-type MRS is to be introduced and details about the robot modeling and controller development environments including a CPG control method are to be presented.

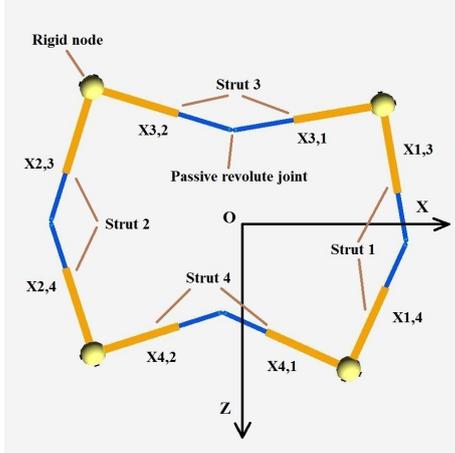
2.1 Strut-Type MRSs

Two strut-type MRSs are illustrated: one is conventional MRSs using ideal nodes and the other one is the proposed MRSs using rigid nodes. Figure 1(a) shows a conventional square-shaped MRS of which each strut has two prismatic actuators. The four robotic struts are connected using ideal compliant nodes equipped with passive revolute joints. Hence, the struts jointed by the same node can rotate around a common center of rotation. Each revolute joint has a rotation range of 80 degrees (Yu, 2010). As mentioned before, such ideal nodes are very difficult to design and implement. To avoid this implementation difficulty and reduce the kinematics complexity, rigid nodes are proposed for connecting robotic struts as shown in Figure 1(b).

Besides, differing from conventional struts that only have prismatic actuators, each strut is comprised of two prismatic actuators and one revolute actuator. To release the kinematic constraints introduced by rigid nodes, we let the revolute actuator be passive



(a) A square robot with ideal compliant nodes



(b) A square robot with rigid nodes

Figure 1: Square robots with compliant and rigid nodes.

to add compliance for MRSs. To the authors' best knowledge, such a hybrid strut design has never been investigated and reported in the literature. Considering the type of actuators and joints used within each strut of the square robots, for convenience, hereafter, we term the robots shown in Figure 1(a) and (b) as RPPR (with R and P separately representing revolute and prismatic actuators) and PRP square robots, respectively. For constructing more complex structures, we consider each square robot as a meta-module, then arbitrary robotic structures can be constructed by rigidly connecting such square meta-modules.

To obtain movements of a robotic strut, one can let the two connected nodes work alternatively as a fixed anchor resorting to a friction-changing mechanism on the node bottom (Cheng et al., 2010). In this paper, to prevent from designing a friction-changing mechanism, we use the following moving steps to achieve a worm-like locomotion of a robotic strut with two prismatic actuators (Yu, 2010):

- Step 1: extend one of the prismatic actuators and keep the other one still;
- Step 2: retract the fully extended prismatic actuator and extend the other one simultaneously;
- Step 3: retract the fully extended prismatic actuator and keep the other one still.

For better understanding, we have tested such a moving principle using a physical strut controlled by an Arduino Uno board. Figure 2 shows the experimental test results. Initially, the prismatic actuators are fully retracted. Then, following the above moving steps periodically, the robotic strut can obtain a worm-like locomotion due to the change of its mass center during the task execution process.

Remarks. As illustrated in Figure 3, initially, the robotic strut keeps still and the normal forces N_1 and N_2 as well as the gravity force G should satisfy $N_1 + N_2 = G$ and $N_1 = N_2 = G/2$. During Step 1, when actuating the left prismatic actuator, at the first few seconds, friction forces $f_1 = \mu N_1$ (with μ denoting the friction coefficient) and $f_2 = \mu N_2$ would be less than the actuation force F . Hence, Node 2 would move rightward for a short time while Node 1 will move leftward. Since the mass center of the whole strut moves leftward and $N_1 + N_2 = G$ together with $G \times l_1 = N_1 \times l_2$, N_2 and f_2 increases while N_1 and f_1 decreases. Therefore, Node 2 would keep still and Node 1 keeps moving leftward. During Step 2, due to the collective work of actuation forces F_1 and F_2 , the horizontal resultant force exerted on the two nodes would be around 0 (we assume $F_1 = F_2$). Since such a force is less than the maximum static friction forces of the two nodes, the two nodes would not move. For the outer casing, owing to its resultant force $F_1 + F_2$, it will move leftward. The motion analysis of Step 3 is similar to Step 1, during Step 3, Node 2 would keep moving leftward while Node 1 would first keep still and then move rightward for a few seconds.

2.2 CPG Control Method

Inspired by (Yu, 2010) and (Sato et al., 2011), a CPG based distributed control method is implemented for comparing and investigating the conventional and proposed strut-type MRSs. Each square meta-module runs the developed identical controller. Initially, each CPG oscillator for each square robot has its own phase $\varphi(0)$. To achieve a coordinating movement of a whole structure comprised of several square meta-modules, the following control law is used to update the oscillator phase:

$$\varphi_i(k+1) = \varphi_i(k) + \gamma \sum_{j \in N_i} (\varphi_j(k) - \varphi_i(k) - \varphi_{ij}^*), \quad (1)$$

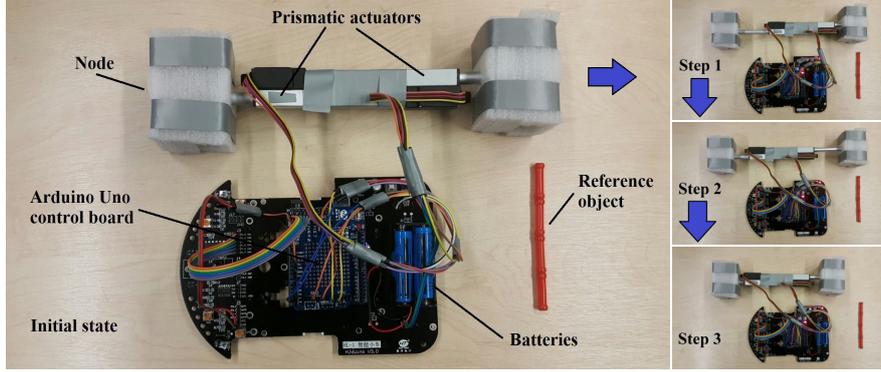


Figure 2: Experimental test of the employed moving principle within one cycle.

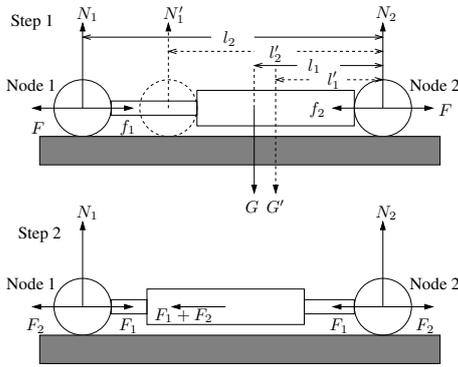


Figure 3: Force analysis of the employed moving principle.

where $\varphi_i(k)$ represents a part of the i th oscillator's phase at the k th time step, parameter γ is related to the convergence speed of (1), and N_i denotes a set containing square meta-module i 's neighboring modules. Constant φ_{ij}^* is the desired phase offset between square meta-modules i and j . With respect to a moving direction, we have

$$\varphi_{ij}^* = \begin{cases} \pi, & \text{if } j \text{ is in front of } i \\ -\pi, & \text{if } j \text{ is at back of } i \\ 0, & \text{if } j \text{ is in parallel with } i \end{cases}$$

then, by considering intrinsic frequency of the CPG oscillator, we can have

$$\theta_i(k) = \omega k + \varphi_i(k), \quad (2)$$

where ω and $\theta_i(k)$ denote the oscillator frequency and oscillator phase, respectively.

For a square meta-module shown in Figure 1, we can have four cardinal traveling directions, i.e., a square robot can move along the positive and negative directions of X- or Z-axis. We use index 1, 2, 3, 4 for representing the traveling direction, which is listed in Table 1. Let d and d' denote the traveling and opposite directions, respectively. Set Ω represents the struts that can enable the square robot move along the

traveling direction once they are actuated. By using the index schemes shown in Figure 1 and Table 1, for all $i \in \Omega$, we do the following computation:

$$\phi_{i,d}(k) = \text{mod}(\theta_i(k), \frac{3\pi}{2}), \quad (3)$$

$$\phi_{i,d'}(k) = \text{mod}(\theta_i(k) - \frac{\pi}{2}, \frac{3\pi}{2}). \quad (4)$$

In this way, the oscillator phase $\theta_i(k)$ is forced to become cyclic signals $\phi_{i,d}(k)$ and $\phi_{i,d'}(k)$ with a period of $3\pi/2$. Finally, the following activation function (AF) is exploited to obtain the corresponding set points $x_{i,d}(k)$ and $x_{i,d'}(k)$:

$$f(\phi) = \begin{cases} L \sin^2(\frac{\pi}{2} \sin^2(\frac{\pi}{2}\phi/P)), & \text{if } 0 < \phi < \pi \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where L indicates the fully extended length of a prismatic actuator and P is a constant parameter related to the period of the output signal. With respect to time, Figure 4 shows profiles of the designed AF (5) (i.e., AF I) and the AF presented in (Yu, 2010) (i.e., AF II). Note that the profile of (5) is smoother than that of AF II. Actually, AF II is not continuous, which may damage the physical motor as the motor velocity has to change abruptly at time 0s, 2.5s, 5s, 10s, 12.5s and 15s. Such a case can be avoided using the continuous function (5). One can easily prove that the n th (with $n = 1, 2, 3 \dots$) order derivative of (5) would be 0 when $\phi = 0$ or $\phi = P$. Hence, the AF (5) is more suitable than AF II for motion planning in robotics. Besides,

Table 1: Index scheme for traveling directions.

Direction	Index
+X	1
-X	2
-Z	3
+Z	4

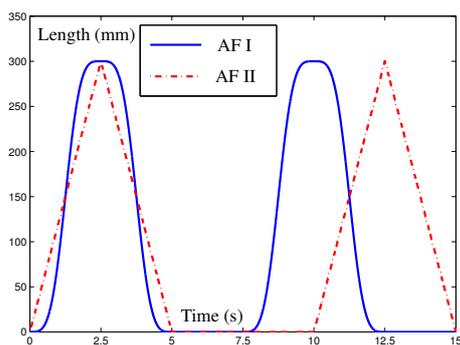


Figure 4: Two different activation functions.

AF (5) has a period of 7.5s (5s for actuation and 2.5s for keeping still) while AF II has a period of 10s (5s for actuation and 5s for keeping still). This is because we find it can speed up the locomotion process by decreasing the time for keeping still.

3 SIMULATIONS

To validate the performance of the proposed way of constructing strut type MRSs, simulations are conducted comparatively. Specifically, MSR with one and two square meta-modules are investigated first. Then, we study the locomotion of an MSR with six PRP square meta-modules. After that, a tentative deformation test is finally performed using an MSR with 37 PRP struts.

Inspired by some successful work (Spröwitz et al., 2014), Webots is used as the robot simulator. Webots is a physics-based simulator developed by Cyberbotics. By using a scene hierarchical tree, a robot model in Webots directly describes the geometric, kinematic and dynamic relationships of the robotic components as well as between the robotic system and its working environment.

3.1 Locomotion

This subsection investigates the locomotion capability of strut-type MRSs. We first study a single square meta-module, then MRSs with two and six square meta-modules are studied. Note that, for the comparative simulations, same initial phase values are employed for a relatively fair comparison.

3.1.1 Single Square Module

Figures 5 and 6 show an RPPR and a PRP square robots moving rightward (i.e., along the +X direction) using the mentioned moving principle. The conventional RPPR square robot can move faster than

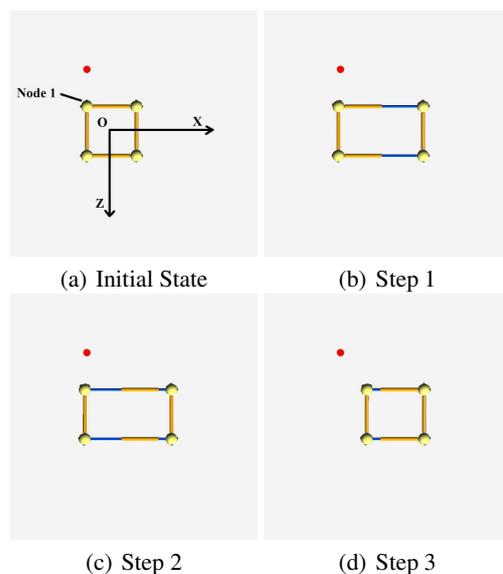


Figure 5: Conventional square robot obtain a worm-like locomotion within one cycle.

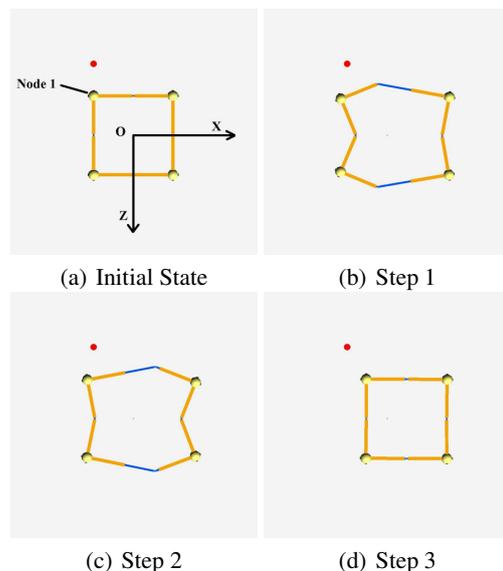


Figure 6: Proposed square robot obtain a worm-like locomotion within one cycle.

the proposed one, since the PRP square robot experiences deformations as shown in Figure 6(b) and (c). Such a fact can be seen in Figure 7 from which we can observe that, after actuating the square robots for 32s, the RPPR and PRP square robots move rightward around 1.0m and 0.7m, respectively.

3.1.2 Two Square Modules

In this part, we let MRSs with two square meta-modules move forward (i.e., along the -Z direction).

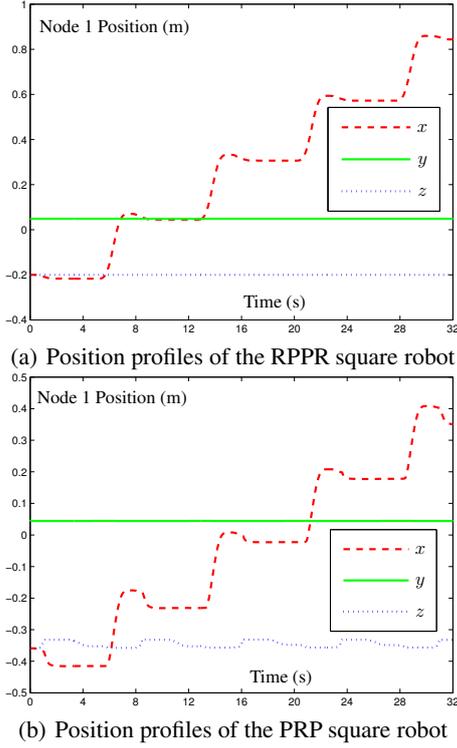


Figure 7: Position profiles synthesized using CPG neural network based control method.

The corresponding results are shown in Figures 8 and 9. As seen from Figure 8(a)–(d) and Figure 9(a)–(d), initially, both the two kinds of MRSs can not move efficiently. After coordinating the phases between the two square meta-modules, a worm-like locomotion can be obtained as shown in Figure 8(e)–(h) and Figure 9(e)–(h). Even though the conventional MRS can still move slightly faster than the proposed MRS, it keeps an unstructured shape when achieving a coordinated locomotion. Unlike the conventional MRS, after some initial deformations, the proposed PRP MSR can restore and maintain its structured shape. This is important, as collisions between modules are more common when moving with an unstructured shape.

3.1.3 Six Square Modules

For further validation, we established MRSs with six square meta-modules. The corresponding simulation results are presented in Figure 10. From the figure, we can see that the MRSs can obtain a coordinating locomotion using the CPG control method. Specifically, oscillator phases are updated with desired phase offsets achieved and actuators' actual profiles finally coincide with their desired counterparts. Note that, during the task execution, square meta-module 0's prismatic actuators $x_{2,3}$ and $x_{2,4}$ are locked. Therefore,

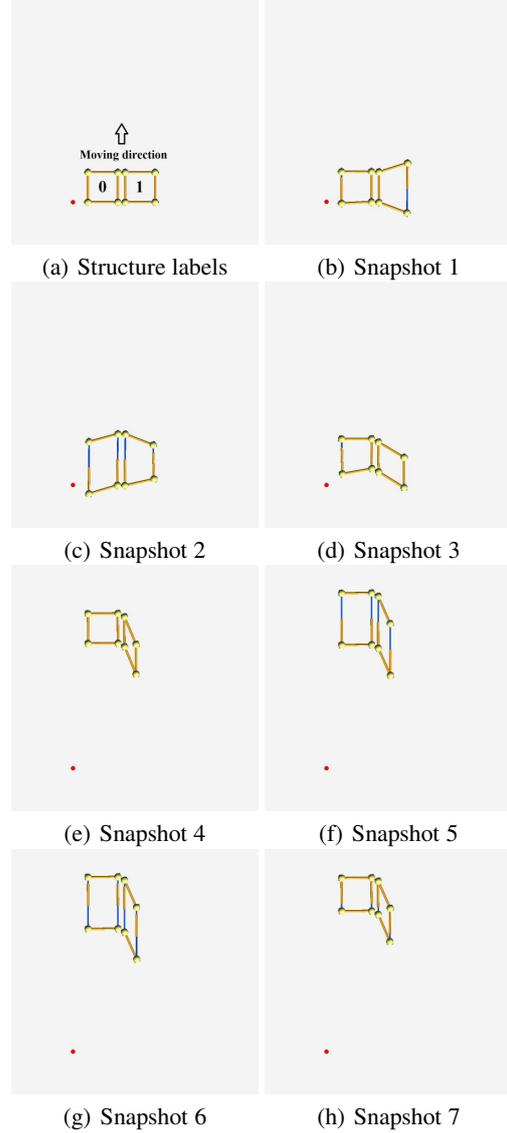


Figure 8: Locomotion of conventional MSR with two square meta-modules.

desired values for $x_{2,3}$ and $x_{2,4}$ are 0. For readability, we only show the first 60s profiles of actuated actuators $x_{4,1}$ and $x_{4,2}$ of square meta-module 0. The deviation in the $x_{4,1}$ profile may result from unexpected kinematic singularities. In future work, we will devote time to coping with this phenomenon.

3.2 Deformation

A simulation is performed to show the deformation capability of the proposed strut-type MRSs. Specifically, we construct an MRS with 37 PRP struts as shown in Figure 11(a). By actuating PRP struts placed along the top and bottom segments of the

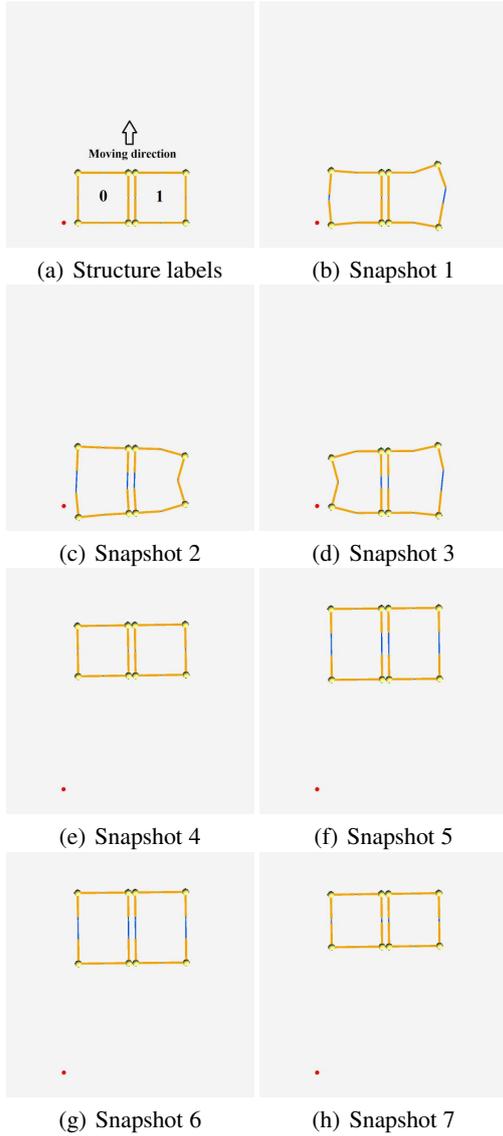
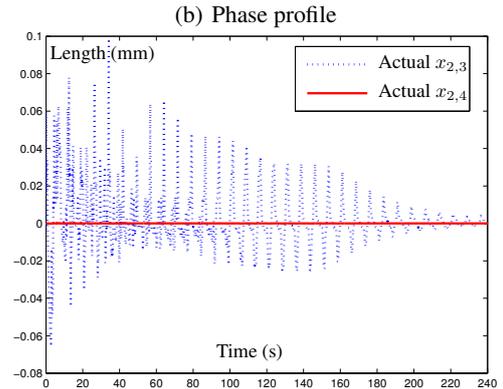
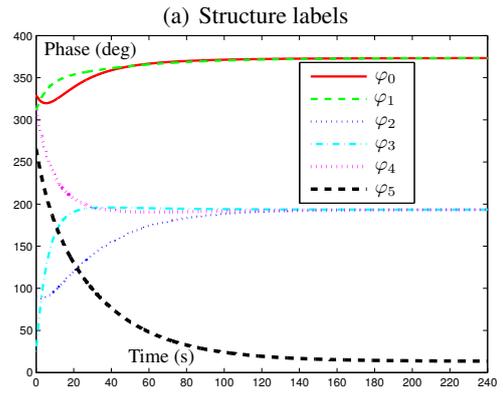
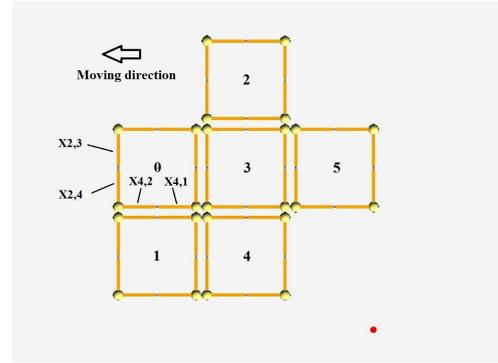


Figure 9: Locomotion of proposed MSR with two square meta-modules.

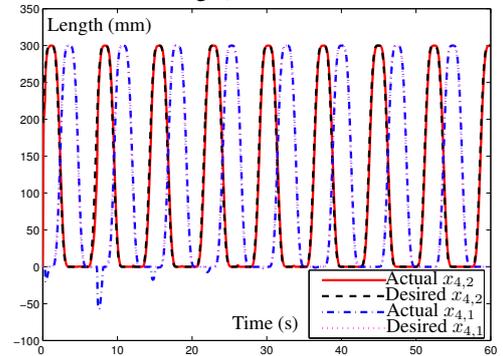
perimeter using (2)–(5) (with $L = 50\text{mm}$ and $\varphi_i(k) = 0\text{rad}$) and letting other struts be passive, we obtain the deformation result shown in Figure 11(b). Such a deformation can be used for physical display.

4 CONCLUSIONS

This paper presents a novel way of constructing strut-type MRSs using rigid nodes and robotic struts equipped with two prismatic and one revolute actuators. For testing such conceptual structures, Webots is used to construct the physics-based robot models.



(c) $x_{2,3}$ and $x_{2,4}$ profiles of square module 0 (with desired values being 0)



(d) $x_{4,2}$ and $x_{4,1}$ profiles of square module 0

Figure 10: Synthesized simulation results of an MSR with six square meta-modules.

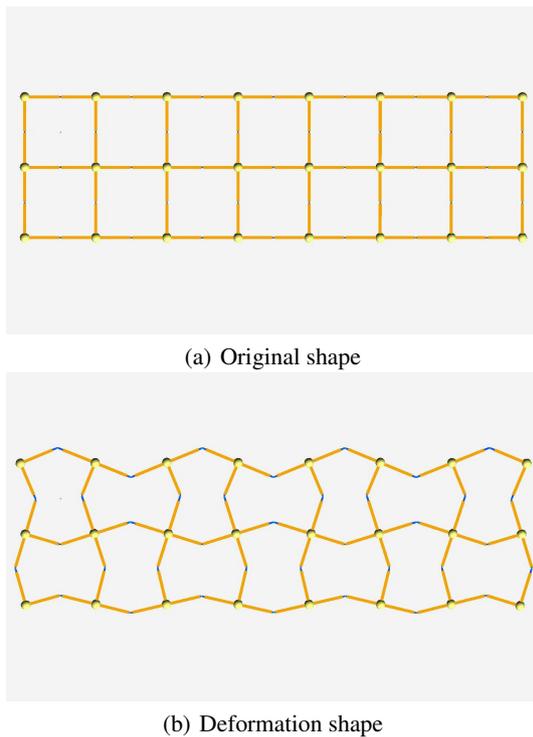


Figure 11: Deformation test of an MRS with 37 PRP struts.

Then, a CPG based control method is implemented for verifying the performance of the proposed MRSs. Comparative simulation results demonstrate the efficacy of the control method and the proposed MRSs as compared with conventional MRSs. Note that, by using rigid nodes, the difficulty of implementing ideal compliant nodes has been avoided, thus simplifying the mechanical design process. Future work will focus on investigating useful deformation of the MRSs, designing and building the proposed MRSs, and verifying the control method using physical MRSs.

REFERENCES

- Baca, J., Hossain, S. G. M., Dasgupta, P., Nelson, C. A., and Dutta, A. (2014). ModRED: Hardware design and reconfiguration planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration. *Robotics and Autonomous Systems*, 62(7):1002–1015.
- Cheng, N., Ishigami, G., Hawthorne, S., Chen, H., Hansen, M., Telleria, M., Playter, R., and Iagnemma, K. (2010). Design and analysis of a soft mobile robot composed of multiple thermally activated joints driven by a single actuator. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5207–5212.
- Curtis, S., Brandt, M., Bowers, G., Brown, G., Cheung, C., Cooperider, C., Desch, M., Desch, N., Dorband, J., Gregory, K., Lee, K., Lunsford, A., Minetto, F., Truszkowski, W., Wesenberg, R., Vranish, J., Abrahantes, M., Clark, P., Capon, T., Weaker, M., Watson, R., Olivier, P., and Rilee, M. L. (2007). Tetrahedral robotics for space exploration. In *Proceedings of the IEEE Aerospace Conference*, pages 1–9.
- Hamlin, G. J. and Sanderson, A. C. (1998). *TETROBOT: A Modular Approach to Reconfigurable Parallel Robotics*. Springer, New York.
- Kurokawa, H., Yoshida, E., Tomita, K., Kamimura, A., Murata, S., and Kokaji, S. (2006). Self-reconfigurable M-TRAN structures and walker generation. *Robotics and Autonomous Systems*, 54(2):142–149.
- Lyder, A. (2010). *Towards Versatile Robots Through Open Heterogeneous Modular Robots*. PhD thesis, University of Southern Denmark.
- Østergaard, E. H., Kassow, K., Beck, R., and Lund, H. H. (2006). Design of the ATRON lattice-based self-reconfigurable robot. *Autonomous Robots*, 21(2):165–183.
- Ramchurn, V., Richardson, R. C., and Nutter, P. (2006). ORTHO-BOT: A modular reconfigurable space robot concept. In Tokhi, M., Virk, G., and Hossain, M., editors, *Climbing and Walking Robots*, pages 659–666. Springer, Berlin Heidelberg.
- Salemi, B., Moll, M., and Shen, W.-M. (2006). SUPER-BOT: A deployable, multi-functional, and modular self-reconfigurable robotic system. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3636–3641.
- Sato, T., Kano, T., and Ishiguro, A. (2011). On the applicability of the decentralized control mechanism extracted from the true slime mold: a robotic case study with a serpentine robot. *Bioinspiration and Biomimetics*, 6(2):026006.
- Spröwitz, A., Moeckel, R., Vespignani, M., Bonardi, S., and Ijspeert, A. J. (2014). Roombots: A hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot. *Robotics and Autonomous Systems*, 62(7):1016–1033.
- Yim, M., Shen, W.-M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., and Chirikjian, G. S. (2007). Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics Automation Magazine*, 14(1):43–52.
- Yu, C.-H. (2010). *Biologically-Inspired Control for Self-Adaptive Multiagent Systems*. PhD thesis, Harvard University.
- Zagal, J. C., Armstrong, C., and Li, S. (2012). Deformable octahedron burrowing robot. In Adami, C., Bryson, D. M., Ofria, C., and Pennock, R. T., editors, *Artificial Life 13*, pages 431–438. MIT Press, Cambridge.
- Zhang, Y., Yim, M., Eldershaw, C., Duff, D., and Roufas, K. (2003). Scalable and reconfigurable configurations and locomotion gaits for chain-type modular reconfigurable robots. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 893–899.

APPENDIX B

Comparison of Modular Robotic Modules

This appendix compares existing modular robotic modules. Table B.1 presents the comparison of typical block-type robotic modules while Table B.2 shows the differences of typical strut-type robotic modules.

Table B.1: Comparison of Block-Type Modules

Modular robot	YaMoR	Molecubes
Type	Chain	Hybrid
Homogeneous	Yes	No
Number of DOF	1 revolute DOF	1 revolute DOF
Docking mechanism	Screw-pin based mechanism	Pin-socket based mechanism; 6 docking faces
Communication	Bluetooth	Wire (internal); Bluetooth (external)
Actuators	Radio control (RC) servomotor	AX-12 servomotor
Sensors	IR sensor, 3D accelerometer	Ultrasonic or IR sensors (future work)
Microcontroller	FPGA/ARM7TDMI microcontroller	ATmega324P processor
Software environment	Bluemove	AGEIA PhysX and OGRE 3D
Control	CPG network	Distributed control
Physical material	PCB, Velcro	Printed ABS plastic
Power source	Lithium-ion batteries	Samsung SB-L220
Reconfiguration methods	Manual reconfiguration	Manual reconfiguration
Applications	Testing control algorithms; Exploring locomotion abilities; Finding new applications for wireless network	An open source platform developed for the modular robotics community

ATRON	MTRAN	Dtto
Lattice	Hybrid	Hybrid
Yes	Yes	Yes
1 revolute DOF	2 revolute DOFs	2 revolute DOFs
Hook-hole based mechanism	Permanent magnets and SMA springs (I, II); Hook-hole based mechanism (III); 6 docking faces	Hook-hole based mechanism
Infrared neighbour-to-neighbour communication	CAN bus, Bluetooth	Bluetooth, RF
Servo motors	DC motors	Servo motors
IR sensors	IR sensors (proximity), accelerometer, tilt-sensor	Inapplicable
BasicATOM40	ATMega128 microcontroller and ATMega8 microcontroller for each hemisphere	Arduino Nano
Inapplicable	Inapplicable	V-REP
Distributed	CPG network	CPG network
Inapplicable	Plastic, aluminium, steel	Plastic
Inapplicable	Two ion-lithium-polymer cells	Lithium-polymer battery
Manual reconfiguration	Meta-module based self-reconfiguration	Self-reconfiguration
Inapplicable	Search and rescue	Open source project for the modular robotics community

SuperBot	Roombots	ModRED
Hybrid	Hybrid	Chain
Yes	No	Yes
3 revolute DOFs	3 revolute DOFs	4 (RRPR: 3 revolute and 1 prismatic) DOFs
Hook-hole based mechanism; 6 docking faces	Hook-hole based mechanism; 10 docking faces	Solenoid-based mechanism; Genderless; 2 docking faces
SPI bus; IR LEDs	Slip rings; Bluetooth	1 XBee radio modem for wireless communication
DC motors	DC motors	Bipolar stepper motors
Potentiometer, accelerometer and inclinometer sensors	Inapplicable	IR sensors, 9-DOF Razor inertial measurement unit, Bump switches
ATmega128 microcontroller	Inapplicable	2 ATmega328P based Arduino boards
Low-level, behaviour-level and remote-client software	Webots	Webots
Digital hormone based control	CPG network	Fuzzy logic controller
Aluminium	ABS plastic pieces and glass-fiber sheet	Aluminium
Rechargeable lithium-polymer battery	Lithium-polymer battery	Rechargeable lithium-polymer battery
Manual reconfiguration	Force-field guidance, look-up table, collision clouds	Computer vision algorithm; Close-looped and distributed controller with fuzzy logic for dynamic gait election; Gait control tables
Space exploration	Multifunctional and intelligent robotic furniture	Exploration and discovery on planetary surfaces

Sambot	SMORES	GZ-I ¹
Chain	Hybrid	Hybrid
Yes	Yes	Yes
1 revolute DOF	4 revolute DOFs	1 revolute DOF
Hook-hole based mechanism; 1 active docking face and 3 passive docking faces	Permanent magnets; 4 docking faces	Holes, bolts and screw threads based mechanism; 4 docking faces
ZigBee or CAN bus	XBee radio transceivers; slip rings (future)	I2C bus
Micro DC motors	5 motors	1 RC servo
IR sensors	Tactile sensors (future)	Future work
STM32 (master) and ATmega8 (slave) microcontrollers	MBED microcontroller	P89LPC922 microprocessor
Inapplicable	Inapplicable	An assistant software developed by researchers
Behaviour-based control; Gait control table	Inapplicable	Distributed control
Plastic	Plastic	Aluminium
Inapplicable	Power is on-board	Lithium-polymer rechargeable battery
Gait tables	Self-reconfiguration	Manual reconfiguration
Inapplicable	Emulating movement abilities of other modular robots	Search and rescue and space applications

PolyBot ²	SEU Robot ³	ModuKnight ⁴
Hybrid	Chain	Hybrid
Homogeneous or Heterogeneous	Yes	Yes
1 revolute DOF	3 revolute DOFs	2 revolute DOFs
Pin-hole based mechanism; Hermaphroditic; 2 docking faces (segment) and 6 docking faces (node); 4 docking faces without nodes (G1v4)	Pin-hole based mechanism; 2 docking faces	Neutral docking ports; 4 docking faces
RS485 communications bus (G1v4); CAN bus (G2)	ZigBee protocols, wireless sensor network	Inapplicable
A DC motor and an SMA actuator	4 motors	Servo motors
IR sensors (G2)	IR sensors	IR sensors
Motorola 68HC11 board or servo controller boards; PIC 16F877 (G1v4); Motorola PowerPC 555 embedded processor (G2)	ZigBee micro-controller	BasicATOM40
Inapplicable	Simulationxpress	Inapplicable
Massively distributed control network (G3); Phase-automata based control	Distributed control	Distributed control
Plastic (G1); Steel sheet (G2)	Aluminium	Inapplicable
NiMH batteries (G1v4)	Two sets of batteries	Inapplicable
Manual reconfiguration (G1); Self-reconfiguration (G2, G3)	IR-sensor based detach-dock approach	Manual reconfiguration
Space manipulation, space mobility, space station inspection and maintenance	Search and rescue	Inapplicable

¹ GZ-I robot is a low cost chain-type modular robot [Zhang *et al.*, 2008].

² PloyBot is one of the most maturely investigated modular robots with a series of generations [Yim *et al.*, 2003, Zhang *et al.*, 2003].

³ An SEU robotic module can be regarded as a connection of three GZ-I or PolyBot modules [Qiao *et al.*, 2014].

⁴ ModuKnight is a hybrid-type homogeneous asymmetric modular robot [Ko, 2003]. Since GZ-I, PolyBot, SEU and ModuKnight modular robots are similar to YaMoR, ModRED and MTRAN robots to some extent, details of these robots are thus omitted.

Table B.2: Comparison of Strut-Type Modules

Modular robot	TETROBOT	Odin	12-TET Walker	Amorphous Robot
Organisation	Rensselaer Polytechnic Institute	University of Southern Denmark	NASA	Harvard University
Self-reconfigurable	No	No	No	No
Homogeneous	No	No	No	No
Self-contained module	No	No	No	No
Strain*	0.341	1.2	4.29	2
Node device	Scissors based mechanism	Ball-shaped node with 12 connection sockets	Nine-faceted wheel-shaped devices	Ball-shaped node with 3 connection sockets
Connection mechanism	Bolts	Ball-and-socket joint and spring loaded design	Universal joint	Ball-and-socket joint and foam tube
Number of actuators	1	1	2	2
Advantages	Idealized CMS compliant node	Hierarchical design, compliant connector	High strain	Light weight, compliant connector
Disadvantages	Low strain, bulky	Low strain, limited deformation	Bulky, expensive, high control complexity	Limited locomotion gaits

* The strain of a prismatic actuator is defined as the ratio between its workspace and its fully retracted length.

APPENDIX C

Experimental Data

C.1 Stall Current and Stall Torque of the 775 DC Motor

This appendix first shows the measured stall current and stall torque of the 775 DC motor. Force measurement data of the electromagnetic brake and the mechanical locker are then presented. After that, passive state and communication test results are given.

C.1 Stall Current and Stall Torque of the 775 DC Motor

Tables C.1 and C.2 show the resultant stall currents and stall torques of the motor under different voltages, respectively. For each tested voltage, the stall currents and stall torques are measured and recorded five times.

Table C.1: Stall Current of the Selected Motor Under Different Voltages

Voltage (V)	Stall current (A)					
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
3.0	2.95	2.88	2.93	2.85	2.89	2.90
4.0	3.70	3.88	3.95	3.78	3.94	3.85
5.0	4.85	4.78	4.83	4.77	4.92	4.83
6.0	5.94	5.86	5.79	5.96	5.90	5.89
7.0	6.63	6.81	6.73	6.68	6.55	6.68
8.0	7.32	7.46	7.50	7.42	7.40	7.42
9.0	8.24	8.19	8.28	8.37	8.32	8.28
10.0	9.07	8.79	9.03	9.11	8.95	9.01

Table C.2: Stall Torque of the Selected Motor Under Different Voltages

Voltage (V)	Stall torque (N · m)					
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
3.0	0.079	0.090	0.088	0.082	0.086	0.085
4.0	0.109	0.112	0.120	0.117	0.108	0.115
5.0	0.127	0.142	0.134	0.138	0.139	0.136
6.0	0.144	0.158	0.160	0.165	0.153	0.156
7.0	0.188	0.184	0.173	0.165	0.170	0.176

C.2 Electromagnetic Brake Test Results

Voltage (V)	Stall torque (N · m)					
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
8.0	0.196	0.198	0.210	0.203	0.193	0.200
9.0	0.220	0.229	0.225	0.213	0.238	0.225
10.0	0.248	0.262	0.253	0.256	0.251	0.254
11.0	0.288	0.291	0.277	0.281	0.268	0.281
12.0	0.326	0.315	0.317	0.306	0.301	0.313
13.0	0.340	0.342	0.351	0.335	0.337	0.341
14.0	0.358	0.363	0.366	0.370	0.358	0.363

C.2 Electromagnetic Brake Test Results

Table C.3 shows the current drawn by the electromagnetic brake under different voltages. The minimum weights that can break an engaged brake under different voltages are also shown in the table.

Table C.3: Electromagnetic Brake Test Results Under Different Voltages

#	Voltage (V)	2.0	2.5	3.0	3.5	4.0	4.5
1	Weight (N)	5.40	9.32	14.22	21.09	25.51	30.90
	Current (A)	0.03	0.03	0.04	0.05	0.05	0.06
2	Weight (N)	5.40	9.32	14.22	20.60	25.51	30.90
	Current (A)	0.03	0.03	0.04	0.05	0.04	0.06
3	Weight (N)	5.40	9.32	14.22	21.58	25.51	30.90
	Current (A)	0.03	0.03	0.04	0.05	0.06	0.06
Average	Weight (N)	5.40	9.32	14.22	21.09	25.51	30.90
	Current (A)	0.03	0.03	0.04	0.05	0.05	0.06

C.3 Calibration Results of the Force Sensor

A total of 25 groups of experiments are performed to calibrate the force sensor. The calibration data is shown in Table C.4. In each group, different weights are placed on

C.3 Calibration Results of the Force Sensor

Table C.4: Calibration Data of the Force Sensor

Weight (kg)	0.0	0.5	1.0	1.5	2.0	2.5	3.0
Group 1	0	29	158	370	578	764	887
Group 2	0	27	149	360	551	717	929
Group 3	0	37	158	323	477	707	851
Group 4	0	40	182	372	486	694	855
Group 5	0	22	166	324	488	630	854
Group 6	0	20	139	318	475	639	833
Group 7	0	28	163	295	530	673	928
Group 8	0	25	158	317	469	725	865
Group 9	0	29	162	337	516	667	915
Group 10	0	43	159	328	504	662	816
Group 11	0	38	157	352	486	714	890
Group 12	0	30	121	358	512	697	926
Group 13	0	32	161	402	528	754	957
Group 14	0	47	139	295	460	617	791
Group 15	0	37	157	352	508	683	878
Group 16	0	37	154	349	583	726	960
Group 17	0	22	142	287	482	722	930
Group 18	0	33	143	292	476	622	896
Group 19	0	26	204	386	547	723	913
Group 20	0	47	139	300	489	649	792
Group 21	0	36	150	277	479	650	782
Group 22	0	34	140	313	448	605	864
Group 23	0	38	155	331	499	731	880
Group 24	0	25	174	340	474	642	813
Group 25	0	22	142	317	460	699	825
Average	0	32	155	332	500	684	873

the calibration rig. For each case of each group, the sensor value ranging from 0 to 1023 is an average of 100 readings.

C.4 Measuring Results of the Mechanical Locker Force

A total of 10 groups of experiments are conducted to measure the locking force by setting the working time of the locker as time $t \in [550, 850]$ ms in a step of 50 ms. In each case of each group, like in the calibration process, the measuring result is also an average of 100 readings from the force sensor.

Table C.5: Measured Locking Forces (N) of the Mechanical Locker

Time (ms)	550	600	650	700	750	800	850
Group 1	3.1699	6.9033	10.1032	14.5895	21.1150	24.0954	25.5699
Group 2	3.1699	7.6248	10.7621	16.0954	23.2797	26.0405	26.3542
Group 3	3.1699	7.9385	11.7660	15.8758	23.6562	23.4679	26.4169
Group 4	3.1699	6.9974	11.9228	16.5346	22.8091	26.2914	26.2287
Group 5	3.1699	7.6562	12.0483	16.7542	23.4365	25.3503	26.0091
Group 6	3.1699	7.6876	11.6091	16.3150	24.0326	26.1032	25.6012
Group 7	3.1699	7.3425	11.6719	17.0993	24.3464	26.2914	26.6365
Group 8	3.1699	8.3464	12.0170	17.0366	23.0287	26.6365	26.9816
Group 9	3.3895	7.7503	12.1425	16.8797	24.4718	25.6012	26.1346
Group 10	3.6405	6.4954	10.9817	15.9385	25.0993	26.1660	26.1032
Average	3.2389	7.4742	11.5025	16.3119	23.5275	25.6044	26.2036

C.5 Passive State Test Results

The passive state of the prismatic actuator is tested. The external forces that are required to let the actuator move passively are evaluated using slotted masses and the pulley system. Five measurements are recorded for each lead screw under different conditions. The experimental test results are listed in Table C.6.

C.6 Communication Test Results

Communications are performed for five times as shown in Table C.7. In each time, prismatic actuator 1 tries to deliver more than 5000 messages to prismatic actuator 2.

C.6 Communication Test Results

Table C.6: Forces Required to Move Different Lead Screws in the Passive State

Only lead screw	Lead screw	1	2	3	4
	Length (mm)	400	400	400	300
	Lead (mm)	2	8	14	14
	Weight in Trial 1 (N)	> 39.24	10.30	4.41	3.43
	Weight in Trial 2 (N)	> 39.24	9.81	3.92	3.43
	Weight in Trial 3 (N)	> 39.24	10.30	3.92	3.43
	Weight in Trial 4 (N)	> 39.24	10.79	3.92	3.43
	Weight in Trial 5 (N)	> 39.24	10.30	3.43	3.92
	Weight on average (N)	> 39.24	10.30	3.92	3.53
Lead screw with motor	Lead screw	3		4	
	Weight in Trial 1 (N)	8.83		7.85	
	Weight in Trial 2 (N)	8.83		8.83	
	Weight in Trial 3 (N)	9.32		8.34	
	Weight in Trial 4 (N)	9.32		8.34	
	Weight in Trial 5 (N)	8.83		8.34	
	Weight on average (N)	9.03		8.34	
Lead screw with motor and brake	Lead screw	3		4	
	Weight in Trial 1 (N)	11.28		10.30	
	Weight in Trial 2 (N)	11.28		10.79	
	Weight in Trial 3 (N)	11.28		10.30	
	Weight in Trial 4 (N)	11.77		10.30	
	Weight in Trial 5 (N)	11.28		10.79	
	Weight on average (N)	11.34		10.50	

The RF nodes can output information on the screen of its connected laptop to monitor the communication process. An operator stops the communication when the number of messages delivered is over 5000. Then the experimental data are recorded. As can be seen from the table, the lowest successful rate is 98.68% and the average successful rate of the transmitting and receiving process is 99.04%, which verifies the effectiveness of the established communication architecture for the modular robotic system.

C.6 Communication Test Results

Table C.7: Communication Test Results

#	Prismatic actuator 1			Prismatic actuator 2	Successful rate
	Tried to deliver	Delivered but failed to be received	Failed to deliver	Received	
1	5023	49	2	4972	98.98%
2	5019	66	0	4953	98.68%
3	5029	36	3	4990	99.22%
4	5020	42	1	4977	99.14%
5	5056	40	1	5015	99.19%
Average successful rate					99.04%

APPENDIX D

Software Implementation for Experiments

This appendix presents programs for calibrating the force sensor and measuring the locking force of the built mechanical locker. Class diagrams related to physical experiments of the fabricated modular robotic system are also briefly introduced.

D.1 Codes for Measuring Locking Force

This appendix presents the Arduino and Matlab codes implemented for experimentally calibrating the force sensor and evaluating the locking force using the force sensor.

D.1.1 Arduino Codes

LockerForce.ino

```
#include "Brake.h"
#include "IRremote.h"
// Parameters for L9110S motor driver
const int MB_Left = A0;
const int MB_Right = A1;
Brake brake(MB_Left, MB_Right);
unsigned long lockTime = 350;
unsigned long releaseTime = 350;
unsigned long stopTime = 250;
unsigned long time = 0;
boolean motorState = false;
// Parameters for IR receiver
const int irPin = A2;
const int ledPin = A3;
boolean ledState = false;
boolean commandState = false;
IRrecv irrecv(irPin);
decode_results results;
// Parameters for force sensor
int mode = -1;
int readValue;
const int forcePin = A4;
```

```
const int numValues = 100;
// the setup function runs once when you press reset or power the board
void setup() {
  brake.setup();
  irrecv.enableIRIn();
  pinMode(ledPin, OUTPUT);
  pinMode(forcePin, INPUT);
  Serial.begin(115200);
  // Check serial communication: acknowledgement routine
  Serial.println('a');
  char a = 'b';
  // Wait for a specific character from the PC
  while (a != 'a')
  {
    a = Serial.read();
  }
  // the loop function runs over and over again until power down or reset
  void loop() {
    listenIR();
    // Locker test
    if (commandState == true)
    {
      if (motorState == true)
      {
        engageLocker();
      }
      else
      {
        releaseLocker();
      }
    }
    // Check any data has been sent by the PC
```

```
if (Serial.available() > 0)
{
// Check if there is a request for any data
mode = Serial.read();
// Set different modes for various applications
switch (mode)
{
case 'F': // Force sensor
for (int i = 0; i < numValues; i++) {
readValue = analogRead(forcePin);
Serial.println(readValue);
}
break;
}
}
delay(100);
}

void engageLocker() {
Serial.println("Locked state");
brake.engageBrake(lockTime);
brake.stopBrake(stopTime);
}

void releaseLocker() {
Serial.println("Released state");
brake.releaseBrake(releaseTime);
brake.stopBrake(stopTime);
}

void listenIR() {
if (irrecv.decode(&results)) {
switch (results.value)
{
case 0xFFC23D:
commandState = true;

```

```
if (ledState == true)
{
digitalWrite(ledPin, LOW);
ledState = !ledState;
motorState = false;
}
else
{
digitalWrite(ledPin, HIGH);
ledState = !ledState;
motorState = true;
}
break;
case 0xFF6897:
time = 550;
setLockerTime(time, time);
Serial.print("Time is: ");
Serial.println(time);
break;
case 0xFF30CF:
time = 600;
setLockerTime(time, time);
Serial.print("Time is: ");
Serial.println(time);
break;
case 0xFF18E7:
time = 650;
setLockerTime(time, time);
Serial.print("Time is: ");
Serial.println(time);
break;
case 0xFF7A85:
time = 700;
```

```
setLockerTime(time, time);
Serial.print("Time is: ");
Serial.println(time);
break;
case 0xFF10EF:
time = 750;
setLockerTime(time, time);
Serial.print("Time is: ");
Serial.println(time);
break;
case 0xFF38C7:
time = 800;
setLockerTime(time, time);
Serial.print("Time is: ");
Serial.println(time);
break;
case 0xFF5AA5:
time = 850;
setLockerTime(time, time);
Serial.print("Time is: ");
Serial.println(time);
break;
case 0xFF42BD:
time = 900;
setLockerTime(time, time);
Serial.print("Time is: ");
Serial.println(time);
break;
case 0xFF4AB5:
time = 950;
setLockerTime(time, time);
Serial.print("Time is: ");
Serial.println(time);
```

```
break;
case 0xFF52AD:
time = 1000;
setLockerTime(time, time);
Serial.print("Time is: ");
Serial.println(time);
break;
default:
commandState = false;
}
irrecv.resume(); // Receive the next value
}
else
{
commandState = false;
}
}
void setLockerTime(unsigned long time1, unsigned long time2)
{
lockTime = time1;
releaseTime = time2;
if (ledState == true)
{
digitalWrite(ledPin, LOW);
ledState = !ledState;
}
else
{
digitalWrite(ledPin, HIGH);
ledState = !ledState;
}
}
```

Brake.h

```
#ifndef BRAKE_H
class Brake {
public:
Brake(int pinA, int pinB);
virtual ~Brake();
void setup();
void engageBrake(unsigned long time);
void releaseBrake(unsigned long time);
void stopBrake(unsigned long time);
private:
int pinA;
int pinB;
};
#endif
```

Brake.cpp

```
#include "Brake.h"
#include "Arduino.h"
Brake::Brake(int pinA, int pinB) {
this->pinA = pinA;
this->pinB = pinB;
}
Brake::~Brake() {}
void Brake::setup() {
pinMode(this->pinA, OUTPUT);
pinMode(this->pinB, OUTPUT);
}
void Brake::engageBrake(unsigned long time) {
digitalWrite(pinA, HIGH);
digitalWrite(pinB, LOW);
delay(time);
}
```

```
void Brake::releaseBrake(unsigned long time) {
digitalWrite(pinA, LOW);
digitalWrite(pinB, HIGH);
delay(time);
}
void Brake::stopBrake(unsigned long time) {
digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
delay(time);
}
```

D.1.2 Matlab Codes

mainFun.m

```
clc;
clearvars;
close all;
%% Initialize specifications of serial communication
delete(instrfindall);
comPort = 'COM6';
% Connect MATLAB to the Arduino
if (~exist('serialFlag', 'var'))
[arduino, serialFlag] = setupSerial(comPort);
end
fileName = 'meanData.mat';
% Weights in kg for calibration
weights = [0; 0.5; 1.0; 1.5; 2.0; 2.5; 3.0];
if exist(fileName, 'file') ~= 2
numData = 25;
parentPath = pwd;
loadedData = zeros(length(weights), numData);
for i = 1:numData
if (i < 10)
pathName = [parentPath '\Calibration\0' num2str(i)];
```

```

else
pathName = [parentPath '\Calibration\' num2str(i)];
end
cd(pathName);
myFile = matfile('data.mat');
loadedData(:, i) = myFile.meanValue;
end
cd(parentPath);
loadedData = loadedData';
meanData = round(sum(loadedData) / numData);
meanData = meanData';
save meanData.mat meanData
end
% Read a series of values and then obtain the average value
numValues = 100;
initialData = readSensor(arduino, numValues);
if (sum(initialData) ~= 0)
disp('Error: Initial data is not 0.');
```

```

closeSerial;
return;
end
if exist(fileName, 'file') == 2
message = sprintf('Do calibration again?');
caption = 'Yes or No';
calibration = questdlg(message, caption, 'Yes', 'No', 'No');
if strcmpi(calibration, 'Yes')
%% Force sensor calibration
meanValue = forceCalibration(arduino, numValues, weights);
else
msg = sprintf('Use existing calibration results?');
cap = 'Yes or No';
measure = questdlg(msg, cap, 'Yes', 'No', 'Yes');
if strcmpi(measure, 'Yes')
```

```

load(fileName);
else
closeSerial;
return;
end
end
else
message = sprintf('Calibration?');
caption = 'Yes or No';
calibration = questdlg(message, caption, 'Yes', 'No', 'Yes');
if strcmpi(calibration, 'Yes')
%% Force sensor calibration
meanValue = forceCalibration(arduino, numValues, weights);
else
closeSerial;
return;
end
end
figure;
plot(meanData, weights, 'b-*');
P = polyfit(meanData, weights, 1);
wfit = P(1)*meanData+P(2);
hold on;
plot(meanData,wfit,'r-.');
temp=['fig',num2str(length(weights)),'.fig'];
savefig(temp);
msg = sprintf('Measure force?');
cap = 'Yes or No';
measure = questdlg(msg, cap, 'Yes', 'No', 'Yes');
if strcmpi(measure, 'Yes')
%% Initialize a GUI window for visualization of measured force
if (~exist('handle', 'var') || ~ishand(handle))
handle = figure;

```

```
end
if (~exist('axis1', 'var'))
axis1=axes('parent',handle,'position',[0.06 0.19 0.24 0.75]);
end
if (~exist('axis2', 'var'))
axis2=axes('parent',handle,'position',[0.36 0.36 0.58 0.58]);
end
if (~exist('text1', 'var'))
text1 = uicontrol('Style', 'text', 'String', 'Mass: 0 kg',...
'pos', [55 30 100 15], 'parent', handle);
end
if (~exist('text2', 'var'))
text2 = uicontrol('Style', 'text', 'String', 'Force: 0 N',...
'pos', [55 10 100 15], 'parent', handle);
end
if (~exist('button1', 'var'))
button1 = uicontrol('Style', 'togglebutton', 'String',...
'Stop and close serial port', 'pos', [300 20 200 25], 'parent',...
handle);
end
if (~exist('button2', 'var'))
button2 = uicontrol('Style', 'togglebutton', 'String',...
'Next test', 'pos', [350 60 100 25], 'parent', handle);
end
axes(axis1);
hold off;
%% Read data and plot
numExps = 25;
time = 550:50:850;
forceTemp = zeros(2, 1);
forces = zeros(length(time), numExps);
index = 1;
for i = 1:length(time)
```

```

if (get(button1, 'Value') ~= 0)
break;
end
mbox = msgbox(['Press button ' num2str(i-1)...
' to set time for locking as ' num2str(time(i))]);
uiwait(mbox); % Wait for confirmation
for j = 1:numExps
disp(['Experiemnt ' num2str(j) ' for time = ' num2str(time(i))]);
if (get(button1, 'Value') ~= 0)
break;
end
if (get(button2, 'Value') ~= 0)
set(button2, 'Value', 0);
end
numReading = 0;
while (get(button1, 'Value') == 0)
if (get(button2, 'Value') ~= 0)
break;
end
forceValue = readSensor(arduino, numValues);
forceValue = round(sum(forceValue) / numValues);
mass = polyval(P, forceValue);
if (mass > 0)
if (get(button1, 'Value') ~= 0 || get(button2, 'Value') ~= 0)
break;
end
numReading = numReading + 1;
force = mass * 9.81;
% Update the readouts on the figure
set(text1, 'String', ['Mass : ' num2str(mass) ' kg']);
set(text2, 'String', ['Force : ' num2str(force) ' N']);
bar(force, 'BarWidth', 0.2);
forceTemp(index) = force;

```

```

if (index == 1)
index = 2;
else
index = 1;
end
if (numReading >= 2 && abs(forceTemp(2) - forceTemp(1)) < 2)
disp(['Current force is ' num2str(force) 'N'])
forces(i, j) = force;
promptMessage = sprintf('Adopt the measured value?');
titleBarCaption = 'Yes or No';
button = questdlg(promptMessage, titleBarCaption, 'Yes',...
'No', 'Yes');
if strcmpi(button, 'Yes')
break;
else
continue;
end
end
end
end
pause(0.1);
end
end
end
else
closeSerial;
return;
end
if (i == length(time))
axes(axis2);
plot(time, sum(forces, 2)/numExps, 'b-*');
savefig('Force.fig');
end
save force.mat forces

```

```
mbox = msgbox('Finished! Close the serial port?');  
uiwait(mbox);  
closeSerial;
```

setupSerial.m

```
function [serialBus, serialFlag] = setupSerial(comPort)  
serialFlag = 1;  
serialBus = serial(comPort);  
set(serialBus, 'DataBits', 8);  
set(serialBus, 'StopBits', 1);  
set(serialBus, 'BaudRate', 115200);  
set(serialBus, 'Parity', 'none');  
% Open the serial port: Arduino would output a character 'a'  
fopen(serialBus);  
a = 'b';  
% Read the character 'a'  
while(a ~= 'a')  
a = fread(serialBus, 1, 'uchar');  
end  
if (a == 'a')  
disp('Serial read successfully.');end  
% Output character 'a' to start the Arduino program  
fprintf(serialBus, '%c', 'a');  
mbox =msgbox('Serial communication setup.');uiwait(mbox);  
fscanf(serialBus, '%u'); % Read data  
end
```

readSensor.m

```
function force = readSensor(arduino, numValues)  
fprintf(arduino, 'F'); % Let Arduino read and output sensory data  
i = 1;
```

```
temp = zeros(1, numValues);
while (i <= numValues)
temp(i) = fscanf(arduino, '%d'); % Receive data from the serial port
i = i + 1;
end
force = temp;
end
```

closeSerial.m

```
function [] = closeSerial()
clearvars; % Clear all the variables
if ~isempty(instrfind)
fclose(instrfind); % Close serial port
delete(instrfind);
end
allHandle = allchild(0);
allTag = get(allHandle, 'Tag');
isMsgbox = strcmp(allTag, 'Msgbox_', 7);
delete(allHandle(isMsgbox)); % Close all the message boxes
close all; % Close all the windows
disp('Serial port is closed.');
```

forceCalibration.m

```
function [meanValue] = forceCalibration(arduino, numValues, weights)
%% Force sensor calibration
meanValue = zeros(length(weights), 1); % Vector for storing mean values
data = zeros(length(weights), numValues); % Matrix for storing data
for i = 2:length(weights)
mbox = msgbox(['Place ' num2str(weights(i))...
' kg on the force sensor']);
uiwait(mbox); % Wait for confirmation
data(i, :) = readSensor(arduino, numValues);
```

```
meanValue(i) = round(sum(data(i, :)) / length(data(i, :)));
while (meanValue(i) < meanValue(i-1)) || meanValue(i) == 0
data(i, :) = readSensor(arduino, numValues);
meanValue(i) = round(sum(data(i, :)) / length(data(i, :)));
end
figure(i-1);
hist(data(i, :));
temp=['fig',num2str(i-1),'.fig'];
savefig(temp);
meanValue(i)
end
save data.mat meanValue numValues weights data
end
```

D.2 Class Diagrams for Physical Experiments

The whole control and communication architecture includes a central Uno RF node and numerous Nano RF nodes. The Uno node is equipped with an IR receiver and an nRF24L01+ module. Hence, software for the Uno node has dependencies including the Arduino integrated SPI and IR remote control libraries as well as an open-source library for driving the nRF24L01+ module, which can be downloaded from <https://github.com/nRF24/RF24>. Each Nano node is dependent on the SPI and nRF24L01+ libraries to drive the nRF24L01+ communication module. The EEPROM library is also incorporated into the program to store a varying seed number for generating random phases. Besides, for each Nano node, classes for the motor drivers, rotary encoder and current sensor are implemented with class diagrams shown in Figure D.1. A Nano node also possesses an activation function to generate desired setpoints. The PID controller is modified and developed based on an open-source library which can be downloaded from <https://github.com/br3ttb/Arduino-PID-Library/>. Based on the implemented classes and the open-source Arduino libraries, physical experiments including the state transitions of prismatic actuators and locomotion of various robotic structures are performed. For each experiment, a main function is developed to call and use the classes and the libraries to achieve certain purposes.

D.2 Class Diagrams for Physical Experiments

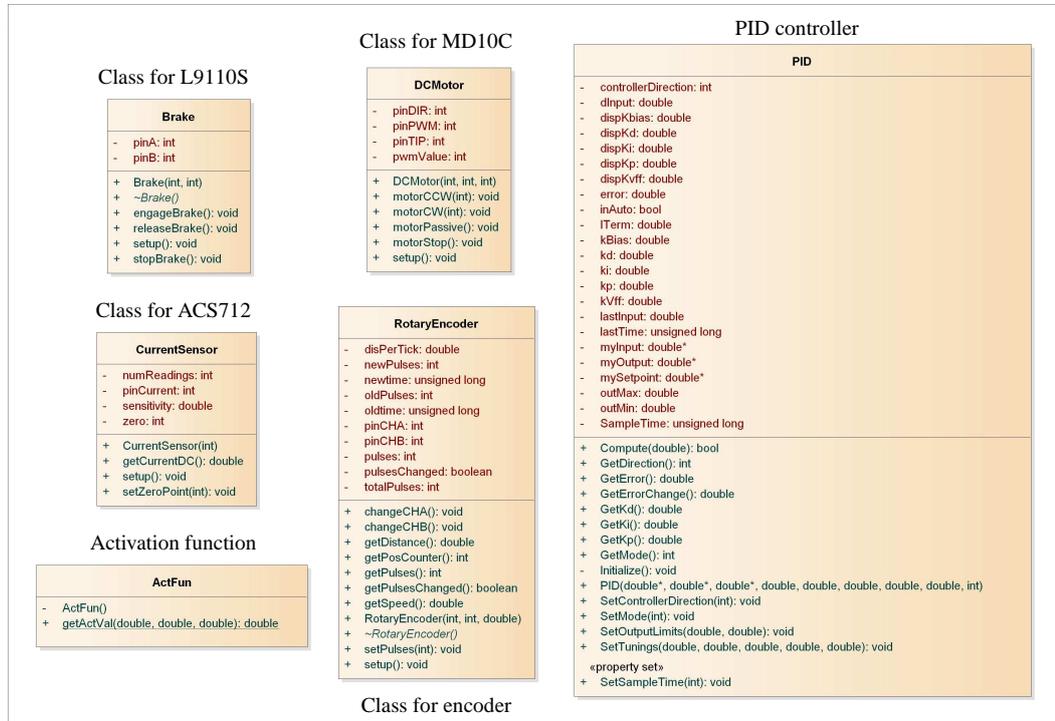


Figure D.1: Class Diagrams of Software for Physical Experiments.

APPENDIX E

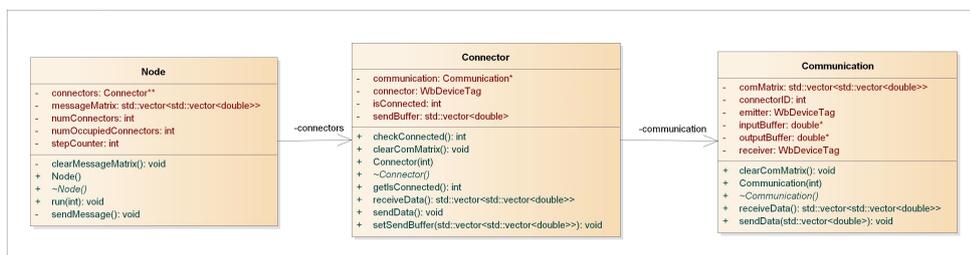
Software Implementation for Simulations

E.1 Class Diagrams for Amorphous Robots

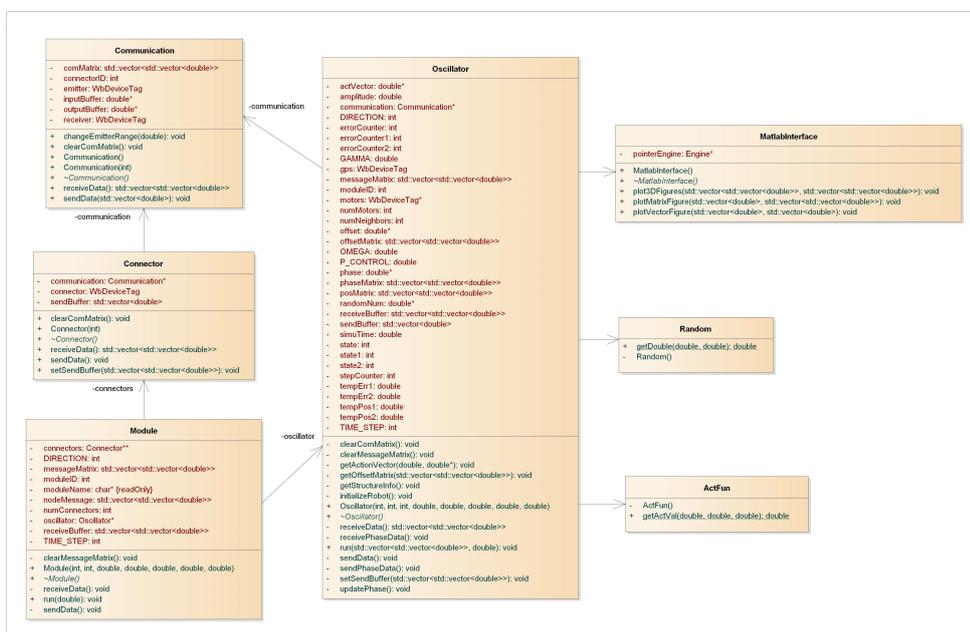
This appendix presents the software implementation for the simulated amorphous robots and the table-shaped robot.

E.1 Class Diagrams for Amorphous Robots

For controlling the simulated amorphous robots in Webots, Two controllers are implemented: one is for each rigid node and the other one is for each robotic strut. Class diagrams of the two controllers are illustrated in Figure E.1. Each rigid node can send



(a)



(b)

Figure E.1: Class Diagrams of Software for Simulated Amorphous Robots. (a) Node Controller. (b) Strut Controller.

E.2 Class Diagrams for the Table-Shaped Robot

messages out through its occupied connector equipped with an IR emitter. Hence, each node class incorporates an array of connector class with each connector class including a communication class. A variety of functions provided by the application programming interface (API) of Webots are used to achieve communications between the robotic elements in the simulated environment. For each strut, it has a module class incorporating two connectors and an oscillator. Similar to the rigid node, each strut has two connectors with IR receivers capable of receiving messages from rigid nodes. Besides, for phase coordination, each oscillator class has a communication class for exchanging information between struts. A random number generator, an activation function and a Matlab interface working as utilities are also incorporated into each oscillator class. The Matlab engine API for C++ is used to call Matlab from the C++ program to plot related figures. Three states of each strut are achieved by using the activation function and the Webots API.

E.2 Class Diagrams for the Table-Shaped Robot

For the table-shaped robot constructed in Webots, each actuator changes its states according to messages received from its neighbouring actuators. Figure E.2 illustrates class diagram of the software. Like the strut of the amorphous robot, each strut of the table-shaped robot requires to perform communication with others and then takes

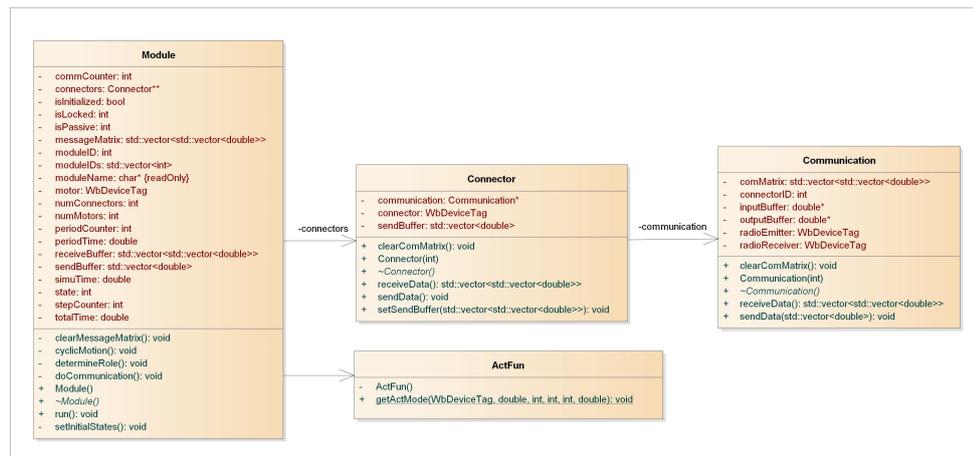


Figure E.2: Class Diagrams of Software for the Simulated Table-Shaped Robot.

E.2 Class Diagrams for the Table-Shaped Robot

actions. Therefore, each module class has two connectors with communication channels and an activation function to realize three states and state transitions.

REFERENCES

- ÅSTRÖM, K. & HÄGGLUND, T. (1995). *PID Controllers: Theory, Design and Tuning*. Research Triangle Park: Instrument Society of America.
- ÅSTRÖM, K. & MURRAY, R. (2010). *Feedback Systems: An Introduction for Scientists and Engineers*. Oxford: Princeton University Press.
- AGARWAL, P. & VISHWAJEET, K. (2009). Kinematic modeling and analysis of 3-PRR parallel manipulator using Matlab GUI. Technical report, University at Buffalo.
- AGHILI, F. & PARSA, K. (2009). A reconfigurable robot with lockable cylindrical joints. *IEEE Transactions on Robotics*, **25**, 785–797.
- APPLIED MOTION PRODUCTS (2014). Datasheet of BL090-H03-G. Available from: <https://www.mclennan.co.uk/product/bl090-h03-g-42mm-brushless-dc-motor-90w>.
- AREVALO, V.M. (2001). Sinusoidal velocity profiles for motion control. Technical report, Augen Ópticos.
- ASHOK, D.A. (2014). *Design and analysis of kinematically redundant planar parallel manipulator for isotropic stiffness condition*. Master thesis, National Institutes of Technology.
- BACA, J., HOSSAIN, S., DASGUPTA, P., NELSON, C.A. & DUTTA, A. (2014). ModRED: Hardware design and reconfiguration planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration. *Robotics and Autonomous Systems*, **62**, 1002–1015.

- BRUCE, J., CALUWAERTS, K., ISCEN, A., SABELHAUS, A. & SUNSPIRAL, V. (2014). Design and evolution of a modular tensegrity robot platform. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 3483–3489.
- BRUCE, R. (2010). *CRC Handbook of Lubrication: Theory and Practice of Tribology, Volume II: Theory and Design*. Boca Raton: CRC Press.
- BUDYNAS, R. & NISBETT, J. (2014). *Shigley's Mechanical Engineering Design*. New York: McGraw-Hill Education.
- CABRERA, M., TRIFONOV, R., CASTELLS, G. & STØY, K. (2011). Wireless communication and power transfer in modular robots. In *Proceedings of the IROS Workshop on Reconfigurable Modular Robotics: Challenges of Mechatronic and Bio-Chemo-Hybrid Systems*.
- CALUWAERTS, K., D'HAENE, M., VERSTRAETEN, D. & SCHRAUWEN, B. (2013). Locomotion without a brain: Physical reservoir computing in tensegrity structures. *Artificial Life*, **19**, 35–66.
- CALUWAERTS, K. *et al.* (2014). Design and control of compliant tensegrity robots through simulation and hardware validation. *Journal of The Royal Society Interface*, **11**.
- CHRISTENSEN, D., SCHULTZ, U. & MOGHADAM, M. (2011). The assemble and animate control framework for modular reconfigurable robots. In *Proceedings of the IROS Workshop on Reconfigurable Modular Robotics*.
- CHRISTENSEN, D.J., CAMPBELL, J. & STØY, K. (2010a). Anatomy-based organization of morphology and control in self-reconfigurable modular robots. *Neural Computing and Applications*, **19**, 787–805.
- CHRISTENSEN, D.J., SPRÖWITZ, A. & IJSPEERT, A.J. (2010b). Distributed online learning of central pattern generators in modular robots. In S. Doncieux, B. Girard, A. Guillot, J. Hallam, J.A. Meyer & J.B. Mouret, eds., *From Animals to Animats 11: Proceedings of the International Conference on Simulation of Adaptive Behavior*, 402–412, Berlin: Springer.

-
- CHRISTOFOROU, E.G., MÜLLER, A., PHOCAS, M.C., MATHEOU, M. & ARNOS, S. (2015). Design and control concept for reconfigurable architecture. *Journal of Mechanical Design*, **137**, 042302.
- CHU, K.D., HOSSAIN, S.G.M. & NELSON, C.A. (2011). Design of a four-DOF modular self-reconfigurable robot with novel gaits. In *Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 747–754.
- COMINOS, P. & MUNRO, N. (2002). PID controllers: recent tuning methods and design to specification. *IEE Proceedings: Control Theory and Applications*, **149**, 46–53.
- CURTIS, S. *et al.* (2006). Mobile science platforms for impassable terrain. In *Proceedings of the IEEE Aerospace Conference*, 1–7.
- CURTIS, S. *et al.* (2007a). Tetrahedral robotics for space exploration. *IEEE Aerospace and Electronic Systems Magazine*, **22**, 22–30.
- CURTIS, S. *et al.* (2007b). Tetrahedral robotics for space exploration. In *Proceedings of the IEEE Aerospace Conference*, 1–9.
- DAVEY, J., KWOK, N. & YIM, M. (2012). Emulating self-reconfigurable robots - design of the SMORES system. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4464–4469.
- DAVIS, P. & NAGASAMUDRAM, R. (2016). Analysis of a 4RPR manipulator. Technical report, University at Buffalo.
- DOKTER, P. (2015). Maker trends: The path of least resistance. In *Proceedings of the IEEE Hot Chips 27 Symposium (HCS)*, 1–21.
- DURALI, L. (2015). *A New Self-Contained Electro-Hydraulic Brake System*. PhD thesis, University of Waterloo.
- FAÍÑA, A., BELLAS, F., LÓPEZ-PEÑA, F. & DURO, R.J. (2013). EDHMoR: Evolutionary designer of heterogeneous modular robots. *Engineering Applications of Artificial Intelligence*, **26**, 2408 – 2423.

-
- FAÍÑA, A., BELLAS, F., ORJALES, F., SOUTO, D. & DURO, R. (2015). An evolution friendly modular architecture to produce feasible robots. *Robotics and Autonomous Systems*, **63**, 195 – 205.
- FANG, Y. & CHEN, X. (2011). Design and simulation of UART serial communication module based on vhdl. In *Proceedings of the International Workshop on Intelligent Systems and Applications*, 1–4.
- FERNÁNDEZ, D.E. (2014). *Learning Locomotion Gait Through Hormone-Based Controller in Modular Robots*. Bachelor thesis, Universidad Carlos III de Madrid.
- FERRO, E. & POTORTI, F. (2005). Bluetooth and Wi-Fi wireless protocols: a survey and a comparison. *IEEE Wireless Communications*, **12**, 12–26.
- FITCH, R. & LAL, R. (2009). Experiments with a ZigBee wireless communication system for self-reconfiguring modular robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1947–1952.
- FRAMPTON, K.D., BAUMANN, O.N. & GARDONIO, P. (2010). A comparison of decentralized, distributed, and centralized vibro-acoustic control. *The Journal of the Acoustical Society of America*, **128**, 2798–2806.
- FUKUDA, T., NAKAGAWA, S., KAWAUCHI, Y. & BUSS, M. (1988). Self organizing robots based on cell structures - CEBOT. In *Proceedings of the IEEE International Workshop on Intelligent Robots*, 145–150.
- GARCIA, R., SCHULTZ, U. & STØY, K. (2009). On the efficiency of local and global communication in modular robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1502–1508.
- GARCIA, R.F.M. (2008). *The Odin Modular Robot: Electronics and Communication*. Master thesis, University of Southern Denmark.
- GARCIA, R.F.M., STØY, K., CHRISTENSEN, D.J. & LYDER, A. (2007). A self-reconfigurable communication network for modular robots. In *Proceedings of the 1st International Conference on Robot Communication and Coordination*, 23:1–23:8.

- GHOSH, A., SANTRA, S.B., BISWAL, P. & CHHOTARAY, P. (2016). Bi-directional converter with modified multi-carrier PWM technique controlled brushless DC motor drive for compressor system. In *Proceedings of the International Conference on Communication and Signal Processing*, 0623–0629.
- GLOBAL ELECTRIC MOTOR SOLUTION (2015). NEMA 23 round brushless DC motor. Available from: <http://gemsmotor.com/nema-23-round-brushless-dc-motor>.
- GROSCH, P., GREGORIO, R.D., LÓPEZ, J. & THOMAS, F. (2010). Motion planning for a novel reconfigurable parallel manipulator with lockable revolute joints. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 4697–4702.
- GROTZINGER, J.P. (2013). Analysis of surface materials by the curiosity mars rover. *Science*, **341**, 1475–1475.
- GUETTAS, C., CHERIF, F., BRETON, T. & DUTHEN, Y. (2014). Cooperative co-evolution of configuration and control for modular robots. In *Proceedings of the International Conference on Multimedia Computing and Systems*, 26–31.
- HACKADAY (2016). Dtto - Explorer Modular Robot. Available from: <https://hackaday.io/project/9976-dtto-explorer-modular-robot>.
- HAM, R.V., SUGAR, T.G., VANDERBORGH, B., HOLLANDER, K.W. & LEFEBER, D. (2009). Compliant actuator designs. *IEEE Robotics and Automation Magazine*, **16**, 81–94.
- HAMANN, H., STRADNER, J., SCHMICKL, T. & CRAILSHEIM, K. (2010). A hormone-based controller for evolutionary multi-modular robotics: From single modules to gait learning. In *Proceedings of the IEEE Congress on Evolutionary Computation*, 1–8.
- HAMLIN, G. & SANDERSON, A. (1997). TETROBOT: A modular approach to parallel robotics. *IEEE Robotics Automation Magazine*, **4**, 42–50.
- HAMLIN, G.J. & SANDERSON, A.C. (1998). *TETROBOT: A Modular Approach to Reconfigurable Parallel Robotics*. New York: Springer.
- HENRIKSEN, E.K. (1973). *Jig and Fixture Design Manual*. New York: Industrial Press.

- HESLINGA, P.M., PADELDFORD, K.W., POPPA, S.M. & SWIESON, C.J. (2012). *Self-Reconfigurable Modular Robot*. Bachelor thesis, Worcester Polytechnic Institute.
- HOLLANDER, K.W. & SUGAR, T.G. (2006). Design of lightweight lead screw actuators for wearable robotic applications. *Journal of Mechanical Design*, **128**, 644–648.
- HORCHLER, A.D. *et al.* (2015). Worm-like robotic locomotion with a compliant modular mesh. In P.S. Wilson, F.P. Verschure, A. Mura & J.T. Prescott, eds., *Proceedings of the International Conference on Biomimetic and Biohybrid Systems, Living Machines*, 26–37, Cham: Springer International Publishing.
- HOSSAIN, S., NELSON, C. & DASGUPTA, P. (2012). Hardware design and testing of ModRED: A modular self-reconfigurable robot system. In J.S. Dai, M. Zoppi & X. Kong, eds., *Advances in Reconfigurable Mechanisms and Robots I*, 515–523, London: Springer.
- HOSSAIN, S.G.M. (2014). *Towards a Sustainable Modular Robot System for Planetary Exploration*. PhD thesis, University of Nebraska.
- HUASONG, M., ZHENGLIN, L., HONGXING, W. & TIANMIAO, W. (2010). Control of a self-assembly modular robot system over a wireless ZigBee network. In *Proceedings of the World Congress on Intelligent Control and Automation*, 1057–1062.
- IJSPEERT, A.J., CRESPI, A., RYCZKO, D. & CABELGUEN, J.M. (2007). From swimming to walking with a salamander robot driven by a spinal cord model. *Science*, **315**, 1416–1420.
- JANSEN, T. (2008). Strandbeests. *Architectural Design*, **78**, 22–27.
- JING, G., TOSUN, T., YIM, M. & KRESS-GAZIT, H. (2016). An end-to-end system for accomplishing tasks with modular robots. In *Proceedings of Robotics: Science and Systems*.
- KARLSSON, P. (2009). Survey of methods of combining velocity profiles with position control. In *IDT Workshop on Interesting Results in Computer Science and Engineering*.
- KO, W.Y.A. (2003). *The Design of a Representation and Analysis Method for Modular Self-Reconfigurable Robots*. Master thesis, University of Hong Kong.

- KRAUSS, R. (2016). Combining Raspberry Pi and Arduino to form a low-cost, real-time autonomous vehicle platform. In *Proceedings of the American Control Conference*, 6628–6633.
- KUNDU, S., CHATTERJEE, D. & CHAKRABARTY, K. (2017). Bifurcation behaviour of PWM controlled DC series motor drive. *IET Power Electronics*, **10**, 279–291.
- KUO, V. & FITCH, R. (2011). A multi-radio architecture for neighbor-to-neighbor communication in modular robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 5387–5394.
- KUO, V. & FITCH, R. (2014). Scalable multi-radio communication in modular robots. *Robotics and Autonomous Systems*, **62**, 1034–1046.
- KUO, V.C.J. (2013). *Enabling Parallel Wireless Communication in Mobile Robot Teams*. PhD thesis, The University of Sydney.
- KUROKAWA, H. *et al.* (2006). Self-reconfigurable M-TRAN structures and walker generation. *Robotics and Autonomous Systems*, **54**, 142–149.
- KUROKAWA, H. *et al.* (2008). Distributed self-reconfiguration of M-TRAN III modular robotic system. *The International Journal of Robotics Research*, **27**, 373–386.
- LEENS, F. (2009). An introduction to I2C and SPI protocols. *IEEE Instrumentation Measurement Magazine*, **12**, 8–13.
- LI, C., NIE, H. & CHEN, J. (2014). Motion analysis and simulation of a 12-tetrahedral walker robot. *Computer Modelling and New Technologies*, **18**, 263–268.
- LI, Z., LUO, J., XI, N. & MING, A. (2007). Development and control of compliant hybrid joints for human-symbiotic mobile manipulators. *International Journal of Advanced Robotic Systems*, **4**, 27–34.
- LYDER, A. (2010). *Towards Versatile Robots Through Open Heterogeneous Modular Robots*. PhD thesis, University of Southern Denmark.
- LYDER, A., GARCIA, R. & STØY, K. (2008). Mechanical design of Odin, an extendable heterogeneous deformable modular robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 883–888.

-
- LYNCH, K. & PARK, F. (2017). *Modern Robotics*. Cambridge: Cambridge University Press.
- MÂNDRU, D. & TEUȚAN, E. (2005). Ankle rehabilitation system based on tetrahedral module. In *Proceedings of the International Conference on Computational Mechanics and Virtual Engineering*, 1–4.
- MARBACH, D. & IJSPEERT, A.J. (2004). Co-evolution of configuration and control for homogenous modular robots. In *Proceedings of the Eighth Conference on Intelligent Autonomous Systems*, 712–719.
- MAYE, J. (2007). *Control of Locomotion in Modular Robotics*. Master thesis, Ecole Polytechnique Fédérale de Lausanne.
- MAZURKIEWICZ, J. (1995). The basics of motion control - Part 1. *Power Transmission Design*, 43–45.
- MERAT, P., AGHILI, F. & SU, C.Y. (2013). Modeling, control and simulation of a 6-DOF reconfigurable space manipulator with lockable cylindrical joints. In J. Lee, M.C. Lee, H. Liu & J.H. Ryu, eds., *Proceedings of the International Conference on Intelligent Robotics and Applications*, 121–131, Berlin: Springer.
- MICHEL, O. (2004). Webots: Professional mobile robot simulation. *Journal of Advanced Robotics Systems*, **1**, 39–42.
- MOECKEL, R. *et al.* (2006). YaMoR and Bluemove - an autonomous modular robot with Bluetooth interface for exploring adaptive locomotion. In M. Tokhi, G. Virk & M. Hossain, eds., *Climbing and Walking Robots*, 685–692, Berlin: Springer.
- MURATA, S. & KUROKAWA, H. (2007). Self-reconfigurable robots. *IEEE Robotics Automation Magazine*, **14**, 71–78.
- MURATA, S., KAKOMURA, K. & KUROKAWA, H. (2007). Toward a scalable modular robotic system. *IEEE Robotics Automation Magazine*, **14**, 56–63.
- MURATA, S. *et al.* (2002). M-TRAN: self-reconfigurable modular robotic system. *IEEE/ASME Transactions on Mechatronics*, **7**, 431–441.

-
- NANSAI, S., ROJAS, N., ELARA, M.R. & SOSA, R. (2013). Exploration of adaptive gait patterns with a reconfigurable linkage mechanism. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4661–4668.
- NGAJIEH, F.N. & WEIBER, C.E. (2015). *Arduino Dynamic Wireless Sensor Network System*. Bachelor thesis, Blekinge Institute of Technology.
- ØSTERGAARD, E.H., KASSOW, K., BECK, R. & LUND, H.H. (2006). Design of the ATRON lattice-based self-reconfigurable robot. *Autonomous Robots*, **21**, 165–183.
- OWAKI, D., KOYAMA, M., YAMAGUCHI, S., KUBO, S. & ISHIGURO, A. (2011). A 2-D passive-dynamic-running biped with elastic elements. *IEEE Transactions on Robotics*, **27**, 156–162.
- PLOOIJ, M., MATHIJSSSEN, G., CHERELLE, P., LEFEBER, D. & VANDERBORGHT, B. (2015). Lock your robot: A review of locking devices in robotics. *IEEE Robotics Automation Magazine*, **22**, 106–117.
- PRASANNA, U.R. & RATHORE, A.K. (2014). Small-signal modeling of active-clamped ZVS current-fed full-bridge isolated DC/DC converter and control system implementation using PSoC. *IEEE Transactions on Industrial Electronics*, **61**, 1253–1261.
- PUTRA, G.D., PRATAMA, A.R., LAZOVIK, A. & AIELLO, M. (2017). Comparison of energy consumption in Wi-Fi and Bluetooth communication in a smart building. In *Proceedings of the IEEE 7th Annual Computing and Communication Workshop and Conference*, 1–6.
- QIAO, G., SONG, G., WANG, W., ZHANG, Y. & WANG, Y. (2014). Design and implementation of a modular self-reconfigurable robot. *International Journal of Advanced Robotic Systems*, **11**, 1–12.
- RENJEWSKI, D., SPRÖWITZ, A., PEEKEMA, A., JONES, M. & HURST, J. (2015). Exciting engineered passive dynamics in a bipedal robot. *IEEE Transactions on Robotics*, **31**, 1244–1251.
- SAHA, S.K. (2008). *Introduction to Robotics*. New Delhi: Tata McGraw-Hill Education.

- SALEMI, B., MOLL, M. & SHEN, W.M. (2006). SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3636–3641.
- SATO, T., KANO, T. & ISHIGURO, A. (2011). On the applicability of the decentralized control mechanism extracted from the true slime mold: a robotic case study with a serpentine robot. *Bioinspiration and Biomimetics*, **6**, 026006.
- SHAW, L. & HOPKINS, J. (2015). An actively controlled shape-morphing compliant microarchitected material. *Journal of Mechanisms and Robotics*, **8**, 021019.
- SHEN, W.M., WILL, P., GALSTYAN, A. & CHUONG, C.M. (2004). Hormone-inspired self-organization and distributed control of robotic swarms. *Autonomous Robots*, **17**, 93–105.
- SHEN, W.M., CHIU, H.C.H., RUBENSTEIN, M. & SALEMI, B. (2008). Rolling and climbing by the multifunctional SuperBot reconfigurable robotic system. In *Proceedings of the Space Technology and Applications International Forum, AIP Conference Proceedings No. 969, American Institute of Physics*, 839–848.
- SHEN, W.M., KOVAC, R. & RUBENSTEIN, M. (2009). SINGO: A single-end-operative and genderless connector for self-reconfiguration, self-assembly and self-healing. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 4253–4258.
- SONG, Y. (2014). *Intelligent PID controller based on fuzzy logic control and neural network technology for indoor environment quality improvement*. PhD thesis, University of Nottingham.
- SPRÖEWITZ, A., MOECKEL, R., MAYE, J. & IJSPEERT, A.J. (2008). Learning to move in modular robots using central pattern generators and online optimization. *The International Journal of Robotics Research*, **27**, 423–443.
- SPRÖWITZ, A. (2010). *Roombots: Design and Implementation of a Modular Robot for Reconfiguration and Locomotion*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne.
- SPRÖWITZ, A., MOECKEL, R., VESPIGNANI, M., BONARDI, S. & IJSPEERT, A. (2014). Roombots: A hardware perspective on 3D self-reconfiguration and locomotion

- with a homogeneous modular robot. *Robotics and Autonomous Systems*, **62**, 1016–1033.
- SPRÖWITZ, A. *et al.* (2010). Roombots: Reconfigurable robots for adaptive furniture. *IEEE Computational Intelligence Magazine*, **5**, 20–32.
- STØY, K., SHEN, W.M. & WILL, P.M. (2002). Using role-based control to produce locomotion in chain-type self-reconfigurable robots. *IEEE/ASME Transactions on Mechatronics*, **7**, 410–417.
- STØY, K., LYDER, A., GARCIA, R.F.M. & CHRISTENSEN, D.J. (2007). Hierarchical robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Self-Reconfigurable Robots, Systems and Applications*.
- STØY, K., BRANDT, D. & CHRISTENSEN, D.J. (2010). *Self-Reconfigurable Robots: An Introduction*. London: The MIT Press.
- TEUȚAN, E., STAN, S.D., VERDES, D. & BALAN, R. (2009). Virtual reality simulation of Tetrobot parallel robot for medical applications. In S. Vlad, R. Ciupa & A. Nicu, eds., *International Conference on Advancements of Medicine and Health Care through Technology*, 177–180, Berlin: Springer.
- TINDELL, K.W., HANSSON, H. & WELLINGS, A.J. (1994). Analysing real-time communications: controller area network (CAN). In *Proceedings of the Real-Time Systems Symposium*, 259–263.
- TOOLEY, M. (2009). *Plant and Process Engineering 360°*. Oxford: Butterworth-Heinemann.
- TOSUN, T., DAVEY, J., LIU, C. & YIM, M. (2016). Design and characterization of the EP-face connector. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 45–51.
- TRUSZKOWSKI, W. *et al.* (2010). Swarms in space missions. In *Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems*, 207–221, London: Springer.
- UMEDACHI, T., VIKAS, V. & TRIMMER, B.A. (2013). Highly deformable 3-D printed soft robot generating inching and crawling locomotions with variable friction legs.

-
- In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4590–4595.
- VANDEVELDE, C., SALDIEN, J., CIOCCI, C. & VANDERBORGHT, B. (2013). Overview of technologies for building robots in the classroom. In *Proceedings of the International Conference on Robotics in Education*, 122–130.
- VOSS, W. (2007). *A Comprehensible Guide to Servo Motor Sizing*. Greenfield: Copperhill Media Corporation.
- WANG, W., LOH, R.N. & GU, E.Y. (1998). Passive compliance versus active compliance in robot-based automated assembly systems. *Industrial Robot: the international journal of robotics research and application*, **25**, 48–57.
- WANG, Z. *et al.* (2016). Design of a novel compliant safe robot joint with multiple working states. *IEEE/ASME Transactions Mechatronics*, **21**, 1193–1198.
- WATANABE, W., SATO, T. & ISHIGURO, A. (2009). A fully decentralized control of a serpentine robot based on the discrepancy between body, brain and environment. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2421–2426.
- WEI, H., CAI, Y., LI, H., LI, D. & WANG, T. (2010). Sambot: A self-assembly modular robot for swarm robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 66–71.
- WEI, H., CHEN, Y., TAN, J. & WANG, T. (2011). Sambot: A self-assembly modular robot system. *IEEE/ASME Transactions on Mechatronics*, **16**, 745–757.
- WILLIAMS II, R.L. & SHELLEY, B.H. (1997). Inverse kinematics for planar parallel manipulators. In *Proceedings of the ASME Design Technical Conferences*.
- WU, A.M., XIAO, J., MARKOVIC, D. & SANDERS, S.R. (1999). Digital PWM control: Application in voltage regulation modules. In *Proceedings of the 30th Annual IEEE Power Electronics Specialists Conference*, vol. 1, 77–83.
- YIM, M., WHITE, P., PARK, M. & SASTRA, J. (2009). Modular self-reconfigurable robots. In R.A. Meyers, ed., *Encyclopedia of Complexity and Systems Science*, 5618–5631, New York: Springer.

- YIM, M. *et al.* (2003). Modular reconfigurable robots in space applications. *Autonomous Robots*, **14**, 225–237.
- YIM, M. *et al.* (2007). Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics Automation Magazine*, **14**, 43–52.
- YU, C.H. (2010). *Biologically-Inspired Control for Self-Adaptive Multiagent Systems*. PhD thesis, Harvard University.
- YU, C.H., HALLER, K., INGBER, D. & NAGPAL, R. (2008). Morpho: A self-deformable modular robot inspired by cellular structure. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3571–3578.
- YU, C.H., WERFEL, J. & NAGPAL, R. (2010). Coordinating collective locomotion in an amorphous modular robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2777–2784.
- ZAGAL, J.C., ARMSTRONG, C. & LI, S. (2012). Deformable octahedron burrowing robot. In C. Adami, D.M. Bryson, C. Ofria & R.T. Pennock, eds., *Artificial Life 13*, 431–438, Cambridge: The MIT Press.
- ZHANG, H., GONZALEZ-GOMEZ, J., ME, Z., CHENG, S. & ZHANG, J. (2008). Development of a low-cost flexible modular robot GZ-I. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 223–228.
- ZHANG, Y., YIM, M., ELDERSHAW, C., DUFF, D. & ROUFAS, K. (2003). Scalable and reconfigurable configurations and locomotion gaits for chain-type modular reconfigurable robots. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, vol. 2, 893–899.
- ZHU, Y. *et al.* (2015). A simplified approach to realize cellular automata for UBot modular self-reconfigurable robots. *Journal of Intelligent and Robotic Systems*, **79**, 37–54.
- ZYKOV, V., MYTILINAIOS, E., ADAMS, B. & LIPSON, H. (2005). Self-reproducing machines. *Nature*, **435**, 163–164.
- ZYKOV, V., CHAN, A. & LIPSON, H. (2007a). Molecubes: An open-source modular robotics kit. In *Proceedings of the IEEE/RSJ International Conference on Intelligent*

Robots and Systems Workshop on Self-Reconfigurable Robots, Systems and Applications.

ZYKOV, V., MYTILINAIOS, E., DESNOYER, M. & LIPSON, H. (2007b). Evolved and designed self-reproducing modular robotics. *IEEE Transactions on Robotics*, **23**, 308–319.

ZYKOV, V., WILLIAMS, P., LASSABE, N. & LIPSON, H. (2008). Molecubes extended: Diversifying capabilities of open-source modular robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Self-Reconfigurable Robots, Systems and Applications.*