A Hybrid and Extendable Self-Reconfigurable Modular Robotic System

Christopher Parrott

Supervisors:
Dr Roderich Groß
Dr Tony J. Dodd

A Thesis Submitted for the Degree of
Doctor of Philosophy

Department of Automatic Control and Systems Engineering

22nd September 2016

Abstract

Modular robotics has the potential to transform the perception of robotic systems from machines built for specific tasks to multi-purpose tools capable of performing virtually any task. This thesis presents the design, implementation and study of a new self-reconfigurable modular robotic system for use as a research and education platform. The system features a high-speed genderless connector (HiGen), a hybrid module (HyMod), an extensions framework, and a control architecture. The HiGen connector features inter-module communication and is able to join with other HiGen connectors in a manner that allows either side to disconnect in the event of failure. The rapid actuation of HiGen allows connections to be made and broken at a speed that is, to our knowledge, an order of magnitude faster than existing mechanical genderless approaches that feature single-sided disconnect, benefiting the self-reconfiguration time of modular robots. HyMod is a chain, lattice, and mobile hybrid modular robot, consisting of a spherical joint unit that is capable of moving independently and grouping with other units to form arbitrary cubic lattice structures. HyMod is the first module, to our knowledge, that combines efficient single-module locomotion, enabling self-assembly, with the ability for modules to freely rotate within their lattice positions, aiding the self-reconfigurability of large structures. The extension framework is used to augment the capabilities of HyMod units. Extensions are modules that feature specialized functionality, and interface with HyMod units via passive HiGen connectors, allowing them to be un-powered until required for a task. Control of the system is achieved using a software architecture. Based on message routing, the architecture allows for the concurrent use of both centralized and distributed module control strategies. An analysis of the system is presented, and experiments conducted to demonstrate its capabilities. Future versions of the system created by this thesis could see uses in reconfigurable manufacturing, search and rescue, and space exploration.

Acknowledgments

I would like to begin by thanking by supervisors, Dr Roderich Groß and Dr Tony J. Dodd for their support throughout my PhD studies. They have assisted in me becoming a researcher, by providing invaluable feedback and helping me become critical of my own work. I give thanks to Roderich for accepting my proposal to work on modular robotics, as well as being supportive of the hardware direction I took the research in. Tony has been of great assistance throughout my PhD, providing high-level feedback on my work, as well as offering an ear to listen to my academic problems. Without my supervisors I would not be where I am today, and for that I immensely grateful.

During my time at Sheffield, I have been privileged to meet and interact with many lab colleagues, giving me great insight in to other areas of robotics. Although not everyone can be mentioned here, special thanks go to Fernando Perez Dias, Matthew Doyle, Melvin Gauci, Yuri Kasubowski Lopes, and Stefan Trenkwalder for their willingness to listen to the many ramblings I have had about the project throughout my research.

I would like to thank the technical team in ACSE and Sheffield Robotics for their support with the hardware for this project, from offering suggestions for components to use, to adapting existing items to my needs. I would also like to thank the financial team in ACSE for their assistance in ordering all of the components needed for this project. Without both teams the project would not be in the position it is today.

My gratitude goes out to Mohamed Marei for undertaking his Masters dissertation project on the creation of a prototype Camera extension module. His research in to camera options, as well as the necessary routines needed to control them and receive video feeds, saved me months of development time when creating the Camera extension presented in this thesis. I wish him the best of luck in his future career.

Thanks go to my friends in the Sheffield University Gaming Society (SLUGSoc), who have helped keep me sane throughout my four years at Sheffield. They have provided a sense of normality to an otherwise abnormal journey. Particular thanks go to Frederich Sonnenwald, for his interest and thoughts on my work, as well as many other random topics, be them gaming related or otherwise. I wish him and the rest of the members of the society all the best for the future.

Finally, I would like to thank my parents for supporting me along this journey to becoming a researcher. They have always been there for me with help and advice, be it via their visits or over the phone. To them I dedicate this work.

Contents

1. Introduction

Since their inception, robotic systems have transformed the world we live in. Having machines capable of repeatedly and reliably performing repetitive tasks has allowed many manufacturing sectors to transition from human labor to machine labor, enabling higher quality and faster production of the many products our society now relies upon. A prime example of this is the car industry, where although many roles are still performed by humans, much of the precision assembly of car chassis and components has been performed by robots for decades.

Despite the advantages of robots for manufacturing, many systems are limited in their uses, typically only designed to perform a single or small set of tasks within a well defined or structured environment, such as an assembly line. Additionally, such robots can be a single point of failure within a system, placing great importance on their continued operation. Transitioning robotic systems to unstructured environments, expanding their task set, and increasing their robustness is an ongoing mission for robotics researchers.

One way this transition is being performed is through modularity. Rather than having a single robot for a given task, instead multiple robots work together to perform the same task. The intention behind this is that by having a collective of robots, the importance of any single one is reduced, allowing such systems to be more adaptable to failure, as in-operable robots can be replaced. Additionally, collectives of robots are potentially capable of collaborating to achieve tasks that may not be achievable by a single robot alone, expanding the range of tasks such a system can perform and the number of environments they can be performed in.

There are two main research areas focused on increasing the modularity of robotic systems, swarm robotics and modular robotics. In swarm robotics [1], many small and sometimes simple robots are placed in an environment together in order to complete a task. In some cases, each robot can complete the task on their own, so having multiple robots increases the systems redundancy. In other cases, the task may only be achievable if multiple robots work together. The reference to swarms comes from this collective behavior, as it likens the systems to natural swarms such as insect colonies, flocks of birds or schools of fish [2]. Examples of tasks being achieved with swarm robotics include object clustering [3, 4] and cooperative transport [5, 6]. In modular robotics [7, 8], tasks are performed via the use of multiple robotic units or modules. Each module can be small and simple like the robots of swarm systems; however, unlike swarm robots, modular robots are capable of joining or being joined together to form connected structures. Once connected, said modules effectively become a single robot, with the combined functionality of all its modules, be it sensing, actuation or processing. This gives modular systems an advantage over swarms, as with appropriate numbers and types of modules, it is possible to create a structure that matches the shape and functionality of virtually any traditional robotic system. Additionally, with appropriate connection hardware, modular robots are capable of reconfiguring their own structure on demand to suit their current task or environment, increasing their flexibility over swarm systems.

1.1. Motivation

Modular robotics has the potential to change our perspective on robotics, from being bespoke machines intended for specific tasks to tools that can be assembled or self-assemble to perform an endless array of tasks. Such tools could benefit a number of application areas, including reconfigurable manufacturing [9], search and rescue [10], and space exploration [11]. In the distant future modular robotics could even give rise to programmable matter [12], materials consisting of millimeter-scale robots capable of replicating the shape, appearance, and properties of virtually any object.

For reconfigurable manufacturing, the objective is to create manufacturing processes that can adapt to changing product demands more efficiently than is possible with current manufacturing processes [13]. Modular robotics has the potential to take reconfigurable manufacturing further, by introducing self-reconfigurability. Having manufacturing systems that are able to self-reconfigure depending on the product that is to be produced, opens up the opportunity for increased customization of products, as the system is able to change its structure to account for their different manufacturing requirements.

For search and rescue, the ability for a response team to quickly react to a disaster is critical for saving lives. To this end robotic systems have been employed to aid response teams [14], by allowing them to explore areas that may have become inaccessible to humans. Modular robots have the potential to benefit in this regard, as rather than teams needing to transport every available search and rescue robotics platform with them to a disaster, or worse, get to the disaster and then need to have a specialized platform shipped in, a large number of modules can be transported instead. Once arrived, the team can assess the situation and assemble the modules into an appropriate configuration to begin search and rescue operations with, potentially saving valuable time. Additionally, if a self-reconfigurable modular system is used, it could adapt its own configuration to the current situation it is facing, for example, by sprouting legs to overcome rough terrain or transforming in to a snake to navigate through small tunnels.

For space exploration, robotic systems have been used for decades to assist astronauts on orbit and enable our species to explore other planets. Despite this, it is still prohibitively expensive to get payloads to orbit and beyond, meaning that the more uses a robotic payload has the easier it is to justify the launch cost. Modular robotics has the potential to assist in this regard, by offering a platform that is capable of being reconfigured in to different structures to allow for many more tasks to be performed than a traditional robotic system. Additionally, the ability for modular robots to deal with failures could benefit planetary exploration, as such robots could self-reconfigure to discard damaged modules in order to continue with their objective. Although such an application may seem distant, a form of modular robot is already in operation on the International Space Station. The Canadarm2, the stations main manipulator, acts as a normal arm for most activities but when required can be relocated to any point on the station. This is possible due to the symmetric nature of the arm, allowing either end to be docked into the many grapple fixtures around the station's outer surface [15], giving it the ability to move end-over-end from one fixture to another. Additionally, when required, the arm can be used to reconfigure the various sections of the space station as new sections are delivered and old ones retired, effectively making the station itself a modular robot.

1.2. Problem Definition

Great strides have been made over the past decades in creating modular systems that show the potential of the field; however, the field is yet to see systems being used in the three main application areas that motivate it. One possible reason for this is the lack of systems that excel in all the aspects of modular robotics [7] that would benefit such application areas over existing solutions. For example, some systems that feature advanced self-reconfigurability [16, 17, 18, 19] lack the advanced individual locomotion capabilities of other systems [20, 21, 22, 23], or vice versa. Another possible reason for the lack of adoption of modular robotic systems is the absence of available platforms for conducting research on. Unlike fields like swarm robotics, which have multiple platforms available for researchers to purchase, there are relatively few for modular robotics, with those that are available primarily being intended for teaching environments such as schools. This means that the barrier to entry in to the modular robotics field is higher than other robotics fields, as research either has to be confined to simulation or needs to be conducted at institutions that already has their own systems, restricting advancement. What is needed is a platform that integrates many of the successful features presented by past works, whilst being open to allow other researchers to assemble their own.

1.3. Aim and Objectives

Out of the three main application areas of modular robotics, search and rescue is the closest to seeing a benefit from modular robotics, with research already being conducted in to dealing with potential challenges that may be faced [10, 27]. As such, the long-term aim of this research project is to develop a modular robotic system primarily for use in search and rescue applications. This system would need to be both robust to survive the harsh environments that natural and man-made disasters present, and versatile in order for it to add value over currently available platforms. For robustness, hazards the modular robot would need to be designed to deal with include sharp obstacles, abrasive dust, water, and corrosive agents [14]. For versatility, the modular robot would need to be designed to include many of the features of past successful platforms. To be usable in reconfigurable manufacturing and space exploration applications, the system would need to be robust and versatile like for search and rescue, as well as be capable of operating in semi-structured environments, be it production lines or on orbit assemblies. For space exploration, the system would also need to be designed to withstand the cold temperatures and solar radiation that are present in the vacuum of space. All of these requirements are challenging to achieve together in a single iteration of a robotic system, so because of this a short-term aim was selected for this thesis, focusing on versatility.

The overall aim for this thesis is to design, implement and study a new self-reconfigurable modular robotic system for use as a research and education platform. The platform is intended to enable scientific experiments to be conducted in to the self-assembly, self-reconfiguration, and collective locomotion of robotic modules within a laboratory environment, as well as for demonstrations of tasks relevant to search and rescue as well as reconfigurable manufacturing and space exploration to be illustrated. To this end the system produced shall incorporate many features of past successful modular robotic systems and, where possible, improve upon them to advance the state of the art beyond the theoretical and algorithmic contributions the finished system would facilitate. Specific focus shall be given to improving the mobility, self-reconfigurability, and extendibility of modular robots, three attributes that relate to their versatility. To aid in platform adoption, the system shall be made using off-the-shelf or easily acquirable bespoke components as well as 3D printing technology.

To aid in the creation process of this new modular robotic system, four objectives have been selected. These are:

- To develop a connection mechanism for self-reconfigurable modules that addresses limitations of existing mechanisms presented by the literature. This connector shall be designed, constructed, and tested to verify its abilities.

- To design and construct a new self-reconfigurable modular robot. The module design shall be analysed to assess its movement capabilities and potential for self-reconfiguration, and its mechanics and electronics shall be discussed. Finally, experiments showing the capabilities of the module shall be presented.

- To extend upon the connector and module designs through the use of extensions, to enable specialized tasks to be performed. A number of extension modules shall be produced to demonstrate their ability to augment the capabilities of the new modular robot.

- To implement a control architecture for the new modular robotic system that allows for centralised and distributed algorithms to be employed to facilitate the system in performing tasks. A number of tasks shall be demonstrated.

1.4. Contributions

The contributions of this thesis are:

- A gender classification of modular robot connection mechanisms that addresses an inconsistency in the terminology of various genderless designs. Under this new classification connectors are either gendered, bi-gendered (hermaphrodite), or genderless. The distinction between bi-gendered and genderless connectors is how their elements join, with bi-gendered having the active element of one connector joining with the passive element of another, and genderless having the active elements of the two connectors themselves join together. This means that single-sided disconnect is an implicit property of genderless connectors, as only one active element needs to disconnect for a connection to be broken. As such, most existing connectors in the literature that are referred to as genderless fall under this new bi-gendered classification.

- A new high-speed genderless mechanical connection mechanism for modular robots, called HiGen, which is capable of joining with other HiGen connectors in a manner that allows either side to disconnect in the event of failure. The mechanism is capable of extending out of and retracting in to its housing, allowing for electrical connections for communication and power sharing to be made and broken, as well as clearance to be created between two neighboring connectors. The connection mechanism's actuation speed is, to our knowledge, an order of magnitude faster than existing mechanical genderless approaches that feature single-sided disconnect (i.e. those that comply with the new genderless classification), offering benefits for the reconfiguration time of self-reconfigurable modular robots.

- A new robotic module that is a hybrid between chain, lattice and mobile self-reconfigurable robots, called HyMod. The module is based on a three degrees of freedom spherical joint and features four HiGen connectors, enabling it to not only rotate freely in place within a cubic lattice position, but also act as a differential wheel setup for individual mobility when away from a connected structure. By allowing free in place rotation, connected modules are able to change their orientation within their lattice positions without colliding with adjacent modules, aiding the self-reconfigurability of large structures. Additionally, when operating independently, wheeled mobility allows the modules to efficiently locomote around flat environments, much like the robots of a swarm robotic system. HyMod is the first modular system, to our knowledge, that combines these two capabilities. An analysis of the geometry and reconfigurability of HyMod is presented, and details of the mechanics and electronics of the module are discussed. Experiments were conducted to show the module's driving and lifting capabilities.

- A hardware framework for the creation of extensions to the HyMod system. Extensions are modules that augment the capabilities of a set of connected HyMod units, by introducing functionality that they may lack or could be impractical to replicate with a collection of units. The framework covers a passive variant of the HiGen connector, a pick-up location template, and internal electronics. To demonstrate the framework, a set of four extensions were developed, covering the areas of manipulation, mobility, perception, and support. Examples are given of how these extensions could be used in combination with HyMod units to produce configurations suited to performing real-world tasks.

- A software architecture for the control of sets of connected HyMod units and extensions. The architecture allows for the concurrent use of both centralized and distributed module control strategies, and is built around the concept of message routing, enabling information to be exchanged between modules in a transparent manner. Details of the architecture and its operational logic are presented, and a self-reconfiguration scenario involving two HyMod units and a modular surface is proposed and demonstrated in simulation.

1.5. Publications

This thesis represents the author's own work, and includes a number of original contributions to scientific knowledge. The work presented herein has so far led to three peer-reviewed papers:

1. **C. Parrott**, T. J. Dodd, and R. Groß, "HiGen: A high-speed genderless mechanical connection mechanism with single-sided disconnect for self-reconfigurable modular robots", in *Proceedings, 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 3926-3932.

2. **C. Parrott**, T. J. Dodd, and R. Groß, "Towards a 3-DOF mobile and self-reconfigurable modular robot", in *Proceedings, IROS 2014 Modular and Swarm Systems Workshop*, 2014.

3. **C. Parrott**, T. J. Dodd, and R. Groß, "HyMod: A 3-DOF hybrid mobile and self-reconfigurable modular robot and its extensions", in *Proceedings, 13th International Symposium on Distributed Autonomous Robotic Systems*, Springer Tracts in Advanced Robotics, Springer, 2016. (to appear)

The material in Publication 1 corresponds to the contents of Chapter 3, and was used as a basis for Section 2.3. The content of Chapter 4 is derived from the material in Publication 3. Additionally, Publication 3 overviews concepts expanded upon in Chapter 5.

During the course of his PhD studies, the author also contributed to other projects that are not featured in this thesis. These have led to the following publication:

1. M. J. Doyle, X. Xu, Y. Gu, F. Perez-Diaz, **C. Parrott**, and R. Groß. “Modular Hydraulic Propulsion: A robot that moves by routing fluid through itself”, in *Proceedings, 2016 IEEE International Conference on Robotics and Automation*, IEEE, 2016, pp. 5189-5196.

1.6. Thesis Outline

This thesis is structured as follows:

- Chapter 2 explores the literature of modular robotics to place this thesis in context. Section 2.1 presents a brief history of modularity and how it relates to modular robotics. Section 2.2 surveys existing modular robotic systems, covering the categories of chain (2.2.1), translational lattice (2.2.2), rotational lattice (2.2.3), morphable lattice (2.2.4), fixed lattice (2.2.5), self-mobile (2.2.6), and heterogeneous (2.2.7) systems. Section 2.3 examines existing connection mechanisms for modular robots, and presents them under a new classification of gendered (2.3.1), bi-gendered (2.3.2), and genderless (2.3.3). Section 2.4 explores how modular robots can be controlled, and details centralized (2.4.1) and distributed (2.4.2) control strategies.

- Chapter 3 presents the high-speed genderless (HiGen) connection mechanism. Section 3.1 presents the chapter and introduces the HiGen connector. Section 3.2 lists the mechanical (3.2.1), electrical (3.2.2), environmental (3.2.3), performance (3.2.4), and reliability (3.2.5) requirements used in creating the connector. Section 3.3 provides details of the HiGen connector's mechanism (3.3.1) as well as the electrical connections made between connectors (3.3.2). Section 3.4 discusses experiments conducted with the connector, such as its actuation and connection time (3.4.1), electrical connectivity (3.4.2), connection repeatability (3.4.3) and load capacity (3.4.4). Section 3.5 summarizes the chapter.

- Chapter 4 details the hybrid modular (HyMod) robot. Section 4.1 introduces the chapter and presents the HyMod unit. Section 4.2 lists the mechanical (4.2.1), electrical (4.2.2), environmental (4.2.3), performance (4.2.4), and reliability (4.2.5) requirements used in creating the module. Section 4.3 covers the theory and development of the module, by analysing its geometry (4.3.1) and ability to self-reconfigure (4.3.2), examining the clearance considerations for free in place rotation (4.3.3), and detailing the module's hardware and electronics (4.3.4). Section 4.4 presents experiments conducted with a single module. Section 4.5 summarizes the chapter.

- Chapter 5 expands upon the work presented in the previous chapters to create extensions. Section 5.1 introduces the chapter and shows the four extensions developed. Section 5.2 lists the mechanical (5.2.1), electrical (5.2.2), environmental (5.2.3), performance (5.2.4), and reliability (5.2.5) requirements used in creating extension modules. Section 5.3 provides details of how the HiGen connector can be expanded for use with extensions (5.3.1), presents an electronics framework for the rapid creation of extensions (5.3.2), and shows a method for creating pick-up locations for said extensions (5.3.3). Section 5.4 covers the development of a gripper extension. Section 5.5 presents the mecanum wheel extension, driven by HyMod units to allow a robot to move in any direction on a flat surface. Section 5.6 covers a camera extension, giving a modular robot vision as well as additional processing capabilities. Section 5.7 shows a modular surface extension, allowing for a regular grid to be produced for modules to self-reconfigure across. Section 5.8 showcases two example configurations of modules and extensions that could be used for performing tasks. Section 5.9 summarizes the chapter.

- Chapter 6 provides details of the architecture used for the centralized and distributed control of HyMod units and extensions. Section 6.1 introduces the chapter. Section 6.2 provides details of the control architecture, including its message structures and routing method (6.2.1), its implementation of multi-byte serial communication (6.2.2), and the logic of its controller (6.2.3). Section 6.3 presents a self-reconfiguration scenario, detailing a 3D simulator (6.3.1), a surface traversal algorithm for addressing the scenario (6.3.2), and the experimental setup to be used for applying the algorithm to the physical system (6.3.3). Section 6.4 summarizes the chapter.

- Chapter 7 concludes the thesis. Section 7.1 presents a discussion of the achievements (7.1.1) and limitations (7.1.2) of the work, as well as a research vision for the field (7.1.3). Section 7.2 ends the thesis by offering number of avenues for further research.

2. Background and Related Work

This chapter begins with a brief history of modularity (Sections 2.1), and presents a literature review of existing modular robotic systems (Section 2.2) and their connection mechanisms (Section 2.3). Finally the chapter is concluded with a review of modular robot control strategies (Section 2.4).

2.1. A Brief History of Modularity

Modularity of mechanical systems originated during the Industrial Revolution with the advent of interchangeable parts in the early years of the 1800s. During this period, advances in automation allowed for parts to be created with increased precision, and for the first time, to specification. By having standardized parts, not only were machines easier to construct, they could also have their components replaced if damaged or broken. By the 1900s, the creation of mechanical systems had transitioned from a task requiring few skilled workers to one that could be achieved by many unskilled workers, each focused on a small aspect of the overall construction. This forms the basis of the assembly line concept, with Henry Ford pioneering the moving assembly line for the mass production of motor vehicles in 1913. With the arrival of computing towards the 21st century, workers have slowly been replaced with robotic systems, capable of performing the same tasks repeatedly and without breaks, further increasing the ability for machines and products to be mass produced. More recently, there has been a shift away from mass production towards specialized parts with the introduction of computer numerical control (CNC) routers and 3D printers, allowing for companies or individuals with modelling skills to produce one-off items or small runs of parts that may be impractical to mass produce with traditional methods, due to setup costs for instance.

A limitation of robotic systems used in manufacturing and other areas is that, although they are made of standardized parts that can be replaced, they are only capable of performing a specific set of tasks for which they are equipped. In contrast, a human is capable of performing a wide variety of tasks if suitable tools are provided. This has partially been addressed by the inclusion of interchangeable end-effectors, allowing robots to swap from object grasping to drilling or welding, for example, but such robots are still limited by the capabilities of their underlying platform. In 1988 Fukuda et al. proposed the concept of a dynamically reconfigurable robotic system as a means of overcoming robot platform limitations [26]. Consisting of intelligent cells with basic mechanical functions, each cell is capable of joining with other cells to form larger structures to match the mechanical needs of virtually any task. By having a robot composed of cells, the robustness of the structure they create is increased due to each cell being an interchangeable part, and the robot can be low-cost by taking advantage of mass production techniques. Because of this, such reconfigurable robotic systems have the potential to replace traditional robotic systems in many existing settings, as well as offer increased flexibility and adaptability by allowing for the rapid reconfiguration of cells to changing tasks or environments. Additionally, the act of reconfiguration could be handled by the robots themselves, enabling their structures to change on demand. This approach to modularity could see robots change from being machines used to perform specific tasks to tools that can perform many tasks, in much the same way that 3D printing is allowing many specialized items to be produced today. It is this potential that has given rise to the current field of self-reconfigurable modular robotics.

2.2. Existing Reconfigurable Robotic Systems

Over the past decades many modular robotic systems have been developed, exploring the various challenges of the field [7]. Initial efforts focused on what are now referred to as chain-type systems, in which robotic modules are joined together via connection mechanisms to form linear and branching structures reminiscent of snakes or other animals. Shortly after this, lattice-type systems emerged, in which robotic modules reside in a two or three dimensional grid structure, with each module connecting to multiple of their neighbors. The advantage of chain-type modules is that they are able to produce kinematic chains, allowing them to be used as manipulator arms or execute locomotion like that found in nature; however, because of this it can complex for modules to self-reconfigure, that being to change their own connected structure without external influence. The advantage of lattice-type modules is that they are able to self-reconfigure efficiently, due to modules being in known grid positions; however, their ability to locomote is limited by the size and shape of the grid used (e.g. cubic, hexagonal) [7]. Later developments aimed to combine these two classes, creating hybrid-type systems consisting of modules that can reside both in-lattice to simplify reconfiguration, and off-lattice to perform locomotion [8]. Additionally, some systems explored the ability of having modules be capable of independent locomotion, allowing them to break away from a lattice and re-join at a different location.

Since the initial classification of chain, lattice, hybrid, and mobile modular systems, it has become increasingly difficult to identify common traits between modules. For the purpose of this review, the following classifications will be used, with some systems falling under multiple classes:

- ***Chain*** - Systems with modules that are only capable of forming linear or branching structures, due to the number and placement of their connection mechanisms. Joint actuations can be translational and/or rotational. Branching is achieved either as part of the standard module design, or via the addition of a secondary module.

- ***Lattice*** - Systems with modules that reside in a grid, with the number of connectors and their positions allowing for each module to connect with multiple neighbors at once. Four variations of this exist:

  - ***Translational*** - Modules only feature translational joints, allowing them to move parallel to the lattice directions. The orientation of modules can never change, preventing their axes from being rotated off-lattice.

  - ***Rotational*** - Modules feature at least one rotational joint, allowing them to change their orientation and be rotated between lattice positions. All hybrid systems fall under this category, as they are able to use rotation to produce chains of modules that reside off-lattice.

  - ***Morphable*** - Modules feature mechanisms allowing them to morph the shape of the lattice they would normally form, allowing modules to be brought together without needing to move to a neighboring grid position.

  - ***Fixed*** - Modules have no mechanism for moving between grid positions and instead require mobility or an external force to reposition them, be it from a user or a stochastic process.

- ***Self-Mobile*** - Systems with modules that are capable of efficient locomotion (that being, wheels or other drive mechanisms) outside of a chain or lattice structure.

In addition to the above, some systems may feature multiple module designs, making them capable of forming chain, lattice, or mobile systems depending on the modules used. Such systems are known as heterogenous and have been grouped together in this review. A list of all the modular robots covered by this review can be seen in Table 2.1.

2.2.1. Chain

The Polypod system [33, 65] developed by Yim et al. features cube-shaped joint modules with two passive connection surfaces, capable of being connected together to form long chains. Two motors reside in these modules, connected via lever mechanisms on either side of the unit in such a way as to allow both translation and rotation to occur; a design that is unique in the field. To allow for more varied configurations separate branch modules are used with six connection surfaces but no actuation, enabling tree-like structures to be formed. Many potential assemblies have been visualized with the Polypod modules, ranging from snakes to bipeds. One interesting assembly shown was the rolling track, which replicates the efficient motion of tank-treads by moving modules end-over-end. Although no rolling track experiments were conducted with this system, later works have shown the rolling track to be a viable method of locomotion [66].

The CONRO [28] by Castano et al. is one of the first chain systems to feature an active connection mechanism. Two prototypes of this system exist, the first consisting of a single module with one degree of freedom and two passive connection surfaces, and the second having two degrees of freedom perpendicular to each other and four connection surfaces (Figure 2.1(a)). This use of four surfaces in the revised prototype, one of which being actuated, allows for branching structures to be produced with a single module; however, the specific arrangement and types of connectors prevent it from forming lattice structures.

Advancing upon their previous work, Yim et al. developed the PolyBot [32, 67, 68]. Three versions were created, with the G3 being the latest (Figure 2.1(b)). The modules are cube-shaped similar to Polypod, but loose the compression and expansion ability in favor of a single rotational degree of freedom with double the range ($\pm 90°$ versus $\pm 45°$). This offers much greater freedom of movement for both locomotion and reconfiguration, as only four modules are required to produce a closed loop.

Table 2.1.: Classifications of the various modular systems covered in this review, grouped by similarity and ordered alphabetically. The HyMod system produced by this project is included for comparison.

System	Dim.	Self-Reconfig.	Chain	Lattice Class	Self-Mobile	Hetero-genous
CKbot [27]	3D	-	✓	-	-	-
CONRO [28]	3D	✓	✓	-	-	-
GZ-I [29]	3D	-	✓	-	-	-
ModRED [30]	3D	✓	✓	-	-	-
Molecube [31]	3D	✓	✓	-	-	-
PolyBot [32]	3D	✓	✓	-	-	-
Polypod [33]	3D	-	✓	-	-	-
RobMAT [34]	3D	-	✓	-	-	-
YaMoR [35]	3D	-	✓	-	-	-
CHOBIE II [36]	2D Vert.	✓	-	Trans.	-	-
Crystalline [37]	2D	✓	-	Trans.	-	-
Meta. Square [38]	2D	✓	-	Trans.	-	-
Telecubes [39]	3D	✓	-	Trans.	-	-
3-D Unit [40]	3D	✓	✓	Rot.	-	-
ATRON [16]	3D	✓	✓	Rot.	-	-
Fracta[41]	2D	✓	-	Rot.	-	-
M-Blocks [42]	3D	✓	-	Rot.	-	-
M-TRAN I/II [43]	3D	✓	✓	Rot.	-	-
M-TRAN III [17]	3D	✓	✓	Rot.	-	-
Roombots [19]	3D	✓	✓	Rot.	-	-
Soldercube [44]	3D	✓	✓	Rot.	-	-
SuperBot [45]	3D	✓	✓	Rot.	-	-
UBot [18]	3D	✓	✓	Rot.	-	-

System	Dim.	Self-Reconfig.	Chain	Lattice Class	Self-Mobile	Hetero-genous
Catoms [12]	2D	✓	-	Morph.	-	-
Meta. Hex [38]	2D	✓	-	Morph.	-	-
Odin [46]	3D	-	-	Morph.	-	-
Slimebot [47]	2D	✓	-	Morph.	-	-
Pebbles [48]	2D	✓	-	Fixed	-	-
Stochastic [49]	2D	✓	-	Fixed	-	-
CEBOT [50]	2D	✓	-	-	✓	-
Distributed Flight Array [51]	2D	✓	-	Fixed	✓	-
iMobot [52]	3D	-	✓	-	✓	-
M^3 Express [53]	3D	✓	✓	Rot.	✓	-
Sambot [20]	2D	✓	✓	-	✓	-
SMORES [21]	3D	✓	✓	Rot.	✓	-
S-bot [54]	2D	✓	-	-	✓	-
T.E.M.P [55]	2D	✓	-	Fixed	✓	-
Automatic Assembly [56]	3D	✓	✓	Fixed	-	✓
Cubelets / roBlocks [57]	3D	-	-	Fixed	✓	✓
EDHMoR [58]	3D	-	✓	-	-	✓
Fable II [59]	3D	-	✓	-	-	✓
I-Cube [60]	3D	✓	-	Rot.	-	✓
Molecube Ex. [24]	3D	-	✓	Rot.	-	✓
Molecule [61]	3D	✓	-	Rot.	-	✓
SMART [62]	3D	✓	✓	-	-	✓
Swarmanoids [63]	2D	✓	-	-	✓	✓
Symbrion & Replicator [64]	3D	✓	✓	Rot.	✓	✓
Thor [25]	3D	-	-	Fixed	-	✓
HyMod	**3D**	✓	✓	**Rot.**	✓	✓

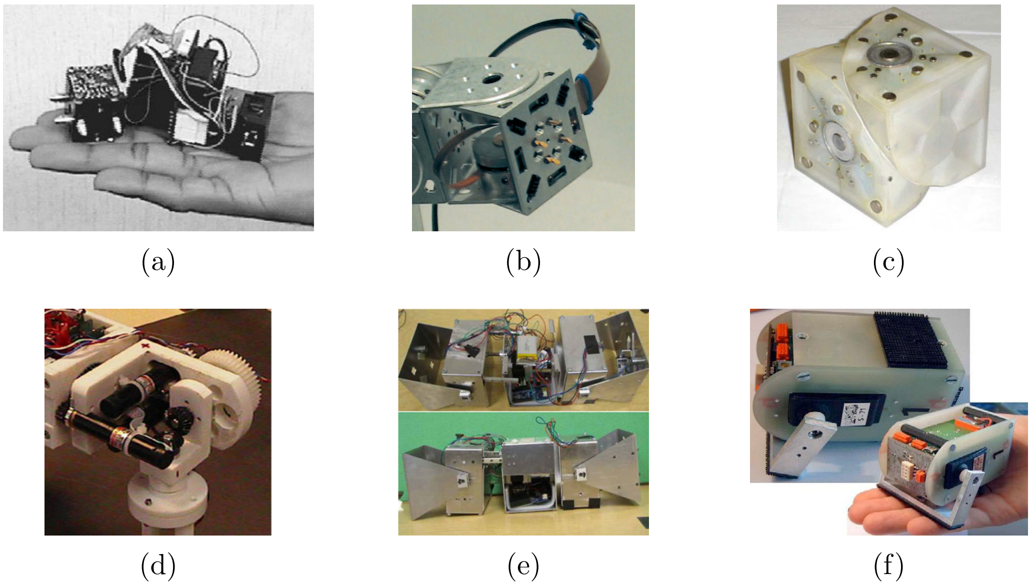


Figure 2.1.: A selection of chain modular robots; (a) CONRO © 2002 IEEE, (b) PolyBot v3 © 2007 IEEE, (c) Molecube © 2007 IEEE, (d) RobMAT © 2008 IEEE, (e) ModRED [1], and (f) YaMoR © 2005 IEEE. Reprinted from [28, 7, 31, 34, 30, 70], respectively.

A system by Zykov et al., called the Molecube [31] (Figure 2.1(c)) tackles the challenges of reconfigurable modular robotics in a unique way. Rather than having its rotational degree of freedom along one of the X, Y or Z axes, it is instead placed along the diagonal of all three, going from one corner of a cube to another. The module is then divided into two halves with three surfaces each, which can continuously rotate relative to each other. By applying fixed rotation amounts of ±120° to the axis, the positions of the surfaces of one half are exchanged, allowing an X axis oriented surface to become Y or Z axis oriented. This offers novel motion at the cost of increased reconfiguration complexity, due to the rotation not being a direct path between two axes. Although designed as a cube and having six surfaces, this prototype only features two connection mechanisms, restricting it to purely chain-like constructs. The connection mechanisms used in this design are active and incorporate both permanent and electro-magnets to hold modules together, with the former creating a strong connection and the latter applying a force to break the connection when required. A revised system of the same name introduces connectors on all surfaces, but removes the ability to self-reconfigure to focus on locomotion strategies [69].

[1] Reprinted from Robotics and Autonomous Systems, 62 / 7, Baca, José and Hossain, SGM and Dasgupta, Prithviraj and Nelson, Carl A and Dutta, Ayan, ModRED: Hardware design and reconfigura-

Recent chain system developments have focused on the use of higher numbers of degrees of freedom within modules to increase their usefulness when dealing with lower module counts. RobMAT [34, 71] developed by Escalera et al. features a joint module with a total of three degrees of freedom in an elbow-like arrangement, giving one across-axis rotation and two along-axis rotations that emulate a spherical joint (Figure 2.1(d)). This allows for the same motion of the PolyBot, but with the addition of the two connectors being able to rotate freely around their normal axes. The ModRED [30, 72] by Chu et al. too features three rotational degrees of freedom, but in a different arrangement (Figure 2.1(e)). Designed as a double cube, the two end rotations allow the connectors to move $\pm 90°$ between cube faces, with the third rotation applying an angular offset between the two cubes. Additionally the module features a translational degree of freedom at the cube intersection, giving it the extension ability of the Polypod. This use of four degrees of freedom allows for many different motions with very few modules, but at the increased cost and complexity of each module.

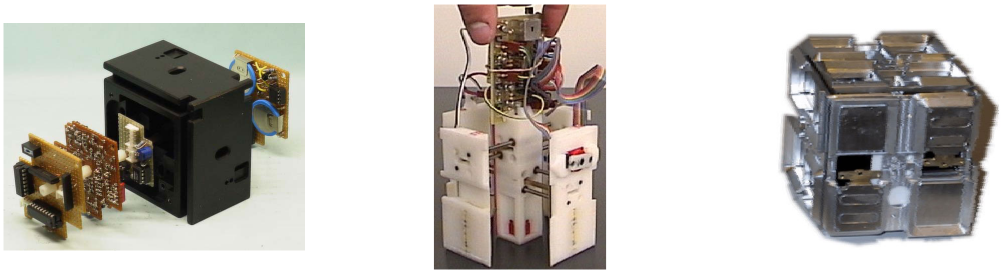
Other chain systems worth mentioning are the CKbot [27], GZ-I [29] and YaMoR [35, 70]. These systems are manually reconfigurable, and designed to be relatively low cost, using hobby grade components in some places. YaMoR can be seen in Figure 2.1(f))

2.2.2. Translational Lattice

Unlike chain and other lattice classes, relatively few translational lattice systems have been developed thus far. The oldest translational lattice system is the Metamorphic Square by Pamecha et al.. Featured as part of a paper discussing metamorphic robots [38], the square design consists of a 2D module that is capable of sliding around neighboring modules in a lattice. This is achieved using a series of rails on the outer perimeter of each module that other modules are able to lock in to and drive along. To allow for modules to transition around the corners of other modules, the rail mechanism can be translated half way into the adjacent lattice position, effectively emulating a module being in that position. This has the result that a module is able to transition itself in to all eight of their neighbor's adjacent lattice positions.

Following a similar concept to that of the Metamorphic Square is the CHOBIE II system [36] (Figure 2.2(a)). Like the Square, all modules feature rails that form a surface other

tion planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration, 1002–1015, Copyright 2013, with permission from Elsevier.



(a) (b) (c)

Figure 2.2.: A selection of translational lattice modular robots; (a) Chobie [2], (b) Crystalline © 2000 IEEE, and (c) Telecubes © 2002 IEEE. Reprinted from [36, 73, 39], respectively.

modules can use to slide along. This requires that there be at least two other modules aligned in the direction of travel to enable movement between two lattice positions. This constraint prevents modules from positioning themselves on corners, as with the Square, meaning a complete line of modules is not possible without removing the ability for the system to self-reconfigure. Interestingly, CHOBIE II operates in a vertical plane, rather than horizontal, allowing the system to be used to demonstrate the creation of overhanging structures to bridge gaps, for example.

An alternative approach to modules sliding within a square grid is to perform expansion and contraction of faces. The first system to feature this is the Crystalline (Figure 2.2(b)), by Rus & Vona [37, 73]. Consisting of 2D modules with four connection surfaces arranged in a square, the system is able to self-reconfigure by compressing two modules in to a single lattice position. This has the effect of making an adjacent lattice position vacant, allowing a neighboring module to extend in to that position. It was shown that this method of self-reconfiguration could be used as a viable method for transforming large lattice structures. Rus & Vona's work also discussed its applicability to three-dimensional lattices; however, it was not until the development of Telecubes (Figure 2.2(c)) by Suh et al. [39, 74] that a platform existed that allowed this concept to be explored. Unfortunately, no real-life demonstrations of 3D self-reconfiguration using Telecubes have been presented.

[2]Springer Distributed Autonomous Robotic Systems, Cellular robots forming a mechanical structure, 6, 2007, 139–148, Koseki, Michihiko and Minami, Kengo and Inou, Norio, © 2007 "With permission of Springer"

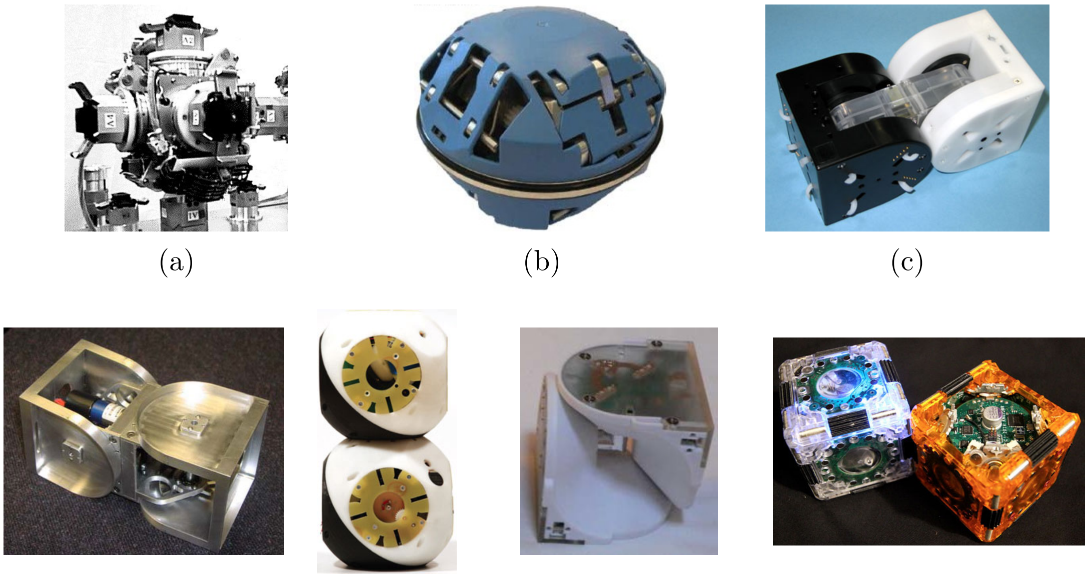
2.2.3. Rotational Lattice

The first rotational lattice system to cover is the Fracta [41], a 2D hexagonal modular robot. Each hexagonal module features connection mechanisms in its corners that other modules connect to and can use and anchor points to roll from one module face to another. This is achieved by using permanent and electro-magnet pairs (one half on either module), oriented vertically. Depending on the field applied to the electro-magnet, the permanent magnets are either attracted to or repelled from them. By coordinating the attraction and repulsion of the magnet pairs, it is possible for a Fracta module to roll around the perimeter of a set of connected modules. Unfortunately, the use of electro-magnets in this manner means the exact angle of rotation cannot be controlled, and therefore the Fracta system cannot reliably form kinematic chains of modules, meaning it is not a hybrid system.

Shortly after the creation of Fracta came 3-D Unit [40, 75], a cube-like module with arms protruding from each of its six faces (Figure 2.3(a)). Each arm features an active connection mechanism and rotational degree of freedom along its axis, allowing modules to assemble into cubic lattices. A novel feature of 3-D Unit for the time was that when the connectors are released, clearance is created between a module and its neighbors, allowing a single unit to rotate in place within a cubic lattice position without needing neighboring modules to be moved from adjacent lattice positions. Additionally, due to motors being used to drive the rotational degrees of freedom of each module, although not demonstrated, it is technically possible for modules to go off-lattice and form kinematic chains, meaning 3-D Unit also falls under the chain-type category and can therefore be considered a hybrid system.

Another system that provides rotation clearance between modules when connectors are retracted is ATRON [16, 76, 77] (Figure 2.3(b)). Designed as two hemispheres with a single continuous rotational degree of freedom joining them, each half houses two active connectors and two passive connectors. The active connectors can extend hooks out from their surface to mate with passive rails on a neighboring module in such a way as to place the two module's rotational axes at right-angles to each other. This unusual formation allows large lattice structures to be produced and for off-lattice configurations to be created. One showcased example was a car-like robot, taking advantage of the continuous rotation ability of the two hemispheres.

One of the most well known modular systems in the field is the M-TRAN [43, 78, 79]. It is designed as a double-cube with two parallel rotational degrees of freedom, placed



(a) (b) (c)

(d) (e) (f) (g)

Figure 2.3.: A selection of rotational lattice modular robots; (a) 3-D Unit © 1998 IEEE, (b) ATRON [3], (c) M-TRAN III (used with permission from H. Kurokawa), (d) SuperBot © 2006 IEEE, (e) Roombots © 2010 IEEE, (f) UBot © 2011 IEEE, and (g) 3D M-Blocks © 2015 IEEE. Reprinted from [40, 16, 80, 81, 82, 18, 42], respectively.

perpendicular to the module's longest axis, and six connection surfaces. Initial versions were shown to perform both locomotion and self-reconfiguration; however, the latter was limited due to the slow actuation time of the connection mechanism, which used shape memory alloy to separate permanent magnets. Later versions refined this concept, with number III forming connections via the use of mechanical hooks, offering greater actuation speed over its predecessors. M-TRAN III (Figure 2.3(c)) was subsequently demonstrated performing quadruped walking motions, self-reconfiguring to a snake and crawling, and transforming back [17]; a significant step for the field. Much of the later work in the field has been influenced by this design.

One such influenced system is the SuperBot [81], developed by B. Salemi et al.. Using the same double-cube structure as M-TRAN, it features an additional degree of freedom along the module's longest axis, capable of rotating the cubes relative to each other (Figure 2.3(d)). This extra degree opens up a number of options for locomotion beyond

[3]Springer Autonomous Robots, Design of the ATRON lattice-based self-reconfigurable robot, 21, 2006, 165–183, Østergaard, Esben Hallundbæk and Kassow, Kristian and Beck, Richard and Lund, Henrik Hautop, © 2006 "With permission of Springer"

that of M-TRAN as it can be used as a primitive wheel, or to make the two other rotational axes become perpendicular, allowing pan and tilt operations. The main focus of the SuperBot platform was to explore locomotion, with crawling, walking and rolling gaits all being demonstrated [45]. Self-reconfiguration was never performed due to the system, at the time, only featuring manual connection mechanisms.

Roombots [19, 82], developed by Spröewitz et al. combines the concept of a module spanning two lattice positions, like M-TRAN, with that of diagonal rotational axes, like Molecubes, into a single module (Figure 2.3(e)). Two cubes with diagonal axes are joined together by a continuous rotational degree of freedom. This arrangement of axes allows Roombots to house ten connection mechanism; however, like with Molecubes, the use of diagonal axes increase the self-reconfiguration complexity of the platform.

Adopting a traditional single cube design is the Ubot by Tang et al.. Similar in function to the PolyBot, the Ubot features two module halves, each with two connectors, joined together via two ±90° rotational degrees of freedom (Figure 2.3(f)). These rotations are oriented perpendicular to each other, allowing a connector to be rotated ±90° in line with an opposing connector as well as raised and lowered to become perpendicular. Unfortunately, only certain rotations can be performed together, otherwise there is risk of the module halves colliding with each other.

Soldercubes are small cubic modules developed by Neubert et al. to demonstrate the use of a phase-change connection mechanism using solder. Each module has six connectors, but only a single continuous rotation axis, oriented through one of the cube's faces (similar to 3-D Unit). As such, the self-reconfiguration ability of the module is limited, instead relying on the system's simplicity to enable many modules to be produced at relatively low cost to provide sufficient degrees of freedom.

A final system to mention in this category is M-Blocks [83], and the recent 3D M-Blocks [42], by Romanishin et al.. Consisting of a single cube with six passive magnetic connectors, each module moves via the use of an inertial mass within the modules themselves (Figure 2.3(g)). By spinning the mass at high-speed and applying a sudden braking force, the momentum of the mass is transferred to the module's structure. This momentum transfer has the effect of overcoming the attractive force of the module's magnetic connectors and pivoting the module around one of its edges, where others magnets and guide teeth aid the module in transitioning between lattice positions. Additionally, by spinning the mass at its maximum speed and then braking, it becomes possible for M-Blocks to jump sections of an assembly. Although, both the M-Blocks and their 3D variant are in fact 3D cubic lattice modules, the non-3D version is only capable of spinning its mass around a single axis. The 3D variant on the other hand, features a diagonal rotational axis internal to it, allowing the axis of the spinning mass to be changed between X, Y and Z, allowing it to move in all three directions, hence being called 3D M-Blocks.

2.2.4. Morphable Lattice

The first morphable system to consider is the Metamorphic Hex, the other platform discussed by Pamecha et al. in their paper on metamorphic robots [38]. This module is hexagonal in nature and capable of forming 2D lattice structures. The design of the module consists of six links joined together in a loop, with actuators at every other corner, allowing them to change their shape from a regular hexagon to fit through or around obstacles. The connection between modules consists of a hook and claw setup, capable of accepting moderate amounts of lateral motion. This enables modules to self-reconfigure around their neighbors, by morphing their shape so that the connectors of an adjacent lattice position are brought together, effectively enabling one module to roll around the perimeter of others.

Following a similar concept of a deformable lattice as the Metamorphic, the Odin (Figure 2.4(a)) is a modular robot comprised of two module types, joints and telescopic links [46, 84]. These modules are manually reconfigurable and can be assembled into a 3D triangular lattice, with joints acting as the branching points for the system . Links have a rotational offset at either end of $\pm 23°$ and can extend from their default length of 60 mm to 132 mm, allowing sides of the triangles to extend. By extending and contracting links in sequence, locomotion of an Odin modular robot can be achieved.

An example of a 2D morphable system using electro-magnets is the Catoms platform [85] (Figure 2.4(b)). Featuring modules just 44 mm across, each contains 24 electro-magnets arranged as two rings of 12, one above the other, to allow fine grained control over the movement of neighboring modules. Due to their cylindrical nature, Catoms can be packed into a hexagonal lattice but can easily form irregular structures when needed, allowing a configuration of modules to morph to fit through openings or around obstacles. The future goal of this technology is to create programmable matter, in which sufficiently small modules can reconfigure into physical structures and be manipulated like clay to adapt new objects [12].

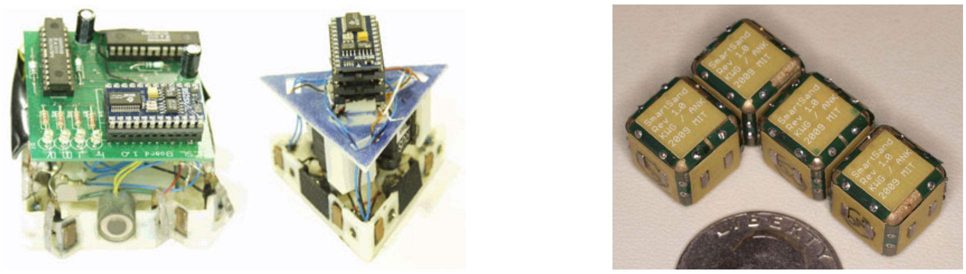
(a) (b) (c)

Figure 2.4.: A selection of morphable lattice modular robots; (a) Odin © 2008 IEEE, (b) Catoms © 2005 IEEE, and (c) Slimebot © 2007 IEEE. Reprinted from [46, 12, 47], respectively.

A final 2D morphable system to mention is the Slimebot [47] (Figure 2.4(c)). Following a similar premise to the Catoms, each module is circular in design, with genderless hook-and-loop fasteners, as found on many clothing items, placed around their perimeter. Unlike electro-magnets this method of connection is passive, so to allow for Slimebots to self-reconfigure their outer surface is divided into six sections capable of extending and retracting independently. This allows one module to effectively push another around its perimeter, as well enable a collection of Slimebots to morph the hexagonal lattice the modules would typically form.

2.2.5. Fixed Lattice

For modules that lack the ability to self-reconfigure by moving joints to transfer modules between lattice positions, stochastic processes can be adopted instead. A stochastic modular system consists of modules with the ability to connect to and disconnect from other modules, but lack any joints or motion capabilities of their own, instead relying on environmental effects to produce module encounters. This concept is demonstrated by the square and triangular platforms of White et al. [49] (Figure 2.5(a)). To allow for the two module types to self-reconfigure stochastically, an air table is used. This table reduces the surface friction experienced by the modules, allowing them to float around randomly. External fans can be used to keep modules in motion [86]. When two modules experience an encounter, they can choose to attract each other using their active magnetic connectors. As such, a set of modules can form a given structure by setting their connector polarities to encourage other modules to attach at the correct locations. Additionally, reconfiguration can occur at any time, by breaking existing connections



(a) (b)

Figure 2.5.: A selection of fixed lattice modular robots; (a) Stochastic Square & Triangle © 2004 IEEE, and (b) Pebbles © 2010 IEEE. Reprinted from [49, 48], respectively.

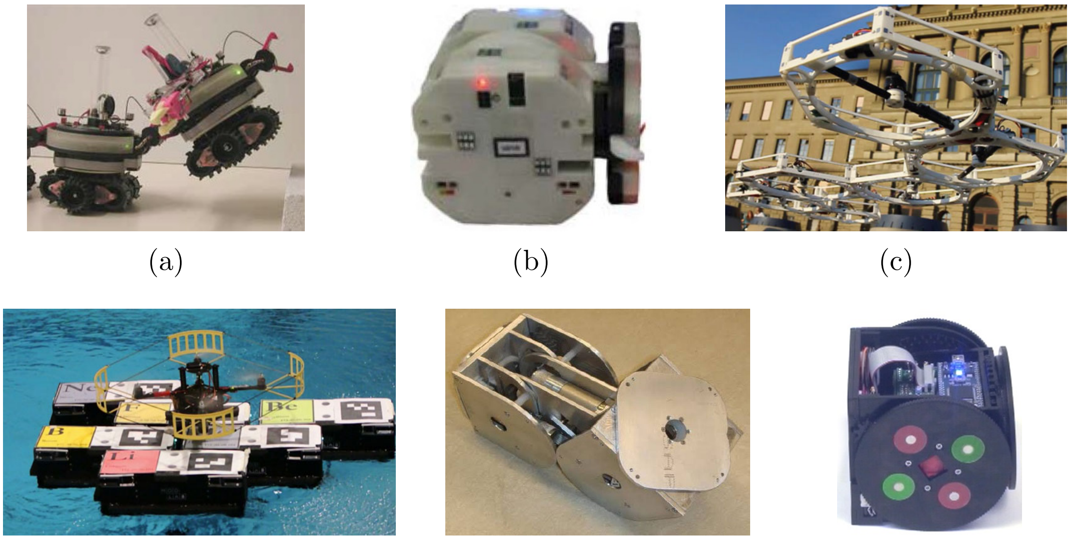
and encouraging connections at new locations.

Pebbles [48] are centimeter sized 2D modular robots, developed by Gilpin et al.. Each module is capable of attaching to other modules via the use of four electro-permanent magnets, one per face (Figure 2.5(b)). The use of electro-permanent magnets allows the attractive force of the connection mechanism to be enabled and disabled on demand, and only consume energy when their state is switched. Structures are created with Pebbles using a process of self-disassembly. Firstly, modules are brought together by an external force such as gravity to produce a grid, with all units' connectors enabled. Once a grid is formed, the modules selectively disable their connectors in order to disconnect modules that do not correspond to the shape required. This shape can then be manually removed from the grid. Details of algorithms for subtractive shape formation with Pebbles can be found in [87].

Two other fixed lattice systems of note are the Distributed Flight Array [51] and the Tactically Expandable Maritime Platform [55]. As both of these system feature drive mechanism that allow for the independent motion of modules, they are covered in more detail in the Self-Mobile modular robots section.

2.2.6. Self-Mobile

One of the first systems created to explore the field of self-reconfigurable modular robotics was CEBOT [50]. Developed by Fukuda et al., their initial work shows the concept of how robotic modules could be used to allow for the examination of a container, with each module being inserted through an inlet and assembling into a larger structure once inside.



(a) (b) (c)

(d) (e) (f)

Figure 2.6.: A selection of self-mobile modular robots; (a) S-bot © 2006 IEEE, (b) Sambot © 2010 IEEE, (c) Distributed Flight Array © 2014 SAGE Publications, (d) T.E.M.P © 2014 IEEE, (e) iMobot © 2010 IEEE, and (f) SMORES © 2010 IEEE. Reprinted from [88, 20, 51, 55, 52, 21], respectively.

On the physical side, prototype modules were produced to test the basic feasibility of the concept, by having a self-mobile robot with a differential drive mechanism move up to and dock with a stationary module, thereby forming a larger configuration. Although the experiments conducted may appear simplistic by today's standards, they were the first verification that self-reconfigurable modular robotics was a worthwhile field to explore.

A more recent showcase of the self-mobile concept is the s-bot platform by Dorigo et al. [54]. Each s-bot (Figure 2.6(a)) features a differential wheel setup that combines wheels and tracks to enable locomotion over uneven terrain whilst maintaining the ability to efficiently rotate on the spot. Intended as a robot for swarm experiments, the s-bot was used as part of the Swarm-Bot project [88], in which groups of s-bots would come together and self-assemble to form large connected structures capable of overcoming obstacles. This was achieved by each robot featuring an illuminated ring that all other robots could attach to via a gripper mechanism, with the illumination being used by each robot's on board vision system to autonomously navigate towards other s-bots.

Following a similar premise to the s-bot is the Sambot by Wei et al. [20, 89], a mobile robot that too can connect to other modules via a gripper mechanism. The Sambot (Figure 2.6(b)) features four passive connectors around its side and an active connector on its top surface. This top surface is able to rotate from the vertical by ±150°, allowing it to move down to the plane of the other connection surfaces, to enable it to join with other modules. The use of a single active connector limits the Sambot to only chain and branching structures. Experiments have shown that a line of modules can perform a snake-like motion to move, and concepts have showcased rolling track and quadruped structures as possible forms.

One of the most unique self-mobile modular systems is the Distributed Flight Array [90]. Developed by Oung et al., this system is comprised of hexagonal modules that fit together into a lattice (Figure 2.6(c)). Each module contains three omni-directional wheels allowing it to independently move in any direction on a flat surface. Magnets on each face are used to attach the modules together, and in the center is housed either a clockwise or counter-clockwise rotating propeller driven by an electric motor. By connecting a minimum of four modules together with equal quantities of propeller rotations, the system is able to take flight. Equal numbers of propellers are required to cancel out the aerodynamic torque each set of spinning blades causes, and at least four modules are needed to offer stability in all planes. Experiments have been conducted showing modules coming together and driving on the ground as single units, with select cases of up to 12 modules being shown to take-off and hover [51].

T.E.M.P, or the Tactically Expandable Maritime Platform [55], developed by O'Hara et al. is a modular robotic system composed of 2D rectangular modules that can float and manoeuvre on water (Figure 2.6(d)). Modules connect together via a flexible hook and rope based mechanism, that allows for platforms to be constructed that can be rigid when required as well as flex to adapt to strong waves. Demonstrations of T.E.M.P modules have been conducted showing floating bridges being formed to allow for small robotic vehicles to cross, as well as landing platforms for quadcopters.

There are three further mobile systems to mention; the iMobot, SMORES, and M^3, all featuring rotating connection surfaces that act as wheels to offer efficient locomotion.

The iMobot [52] by Ryland et al. consists of a double-cube structure with six passive connection surfaces similar to the M-TRAN and SuperBot (Figure 2.6(e)). Its difference lies in the addition of two continuous rotation end-plates that are used to propel the module forward, in addition to rotating adjacent modules. This gives the module the flexibility to perform fast locomotion over even ground, and slower inchworm locomotion over rough ground.

SMORES (Figure 2.6(f)), developed by Davey et al. consists of modules that fit within single cubes much like the PolyBot [21]. It features two primary degrees of freedom forming a tilt and roll mechanism, and two secondary actuations that rotate side plates to provide wheeled locomotion. Connections between modules are achieved with magnets, with a unique rod-based mechanism being used to perform disconnection. A recent revision to SMORES sees this connection mechanism replaced with electro-permanent magnets [91].

M^3 [92] and the subsequent M^3Express [53], developed by Kutzer et al. and Wolfe et al, respectively, consist of L-shaped modules with three driving wheels arranged such that when two modules come together they form the left and right sides of a hinge joint, capable of turning $\pm 120°$. The main difference between the M^3 and its Express variant is the cost of manufacture, with the latter using cheaper components and simpler construction techniques.

2.2.7. Heterogeneous

Heterogeneous systems with two module types are referred to as bipartite, and tend to share a common theme of one module being a construction block and another being a manipulator to move and assemble the blocks. Both the I-Cube [60, 93] and Automatic Assembly System [56] demonstrate this. Another bipartite system of note is the Molecule [61, 94]. Unlike the previous two, the relationship between the modules is more traditional with both having the same shape but only one featuring active connectors. Visually it resembles a molecule of two atoms (cubes) along a diagonal, with an actuated link between them (Figure 2.7(a)). The actuated link consists of two continuous rotational degrees of freedom, allowing each of the Molecule's cubes to be rotated within their respective lattice positions. Simulations and experiments have shown this design to be capable of locomotion and reconfiguration.

Expanding upon previous work with the original Molecube [31], Zykov et al. developed a revised version of the module, with a smaller form-factor, this time featuring six connectors rather than two [69]. Unlike their previous system that focused on self-healing and replication, the new Molecubes were intended as a platform for exploring locomotion capabilities of modules. Because of this the new version forgoes the ability to self-reconfigure, by only using passive connectors, instead offering a number of specialised module types for mobility and manipulation [24] (Figure 2.7(b)).

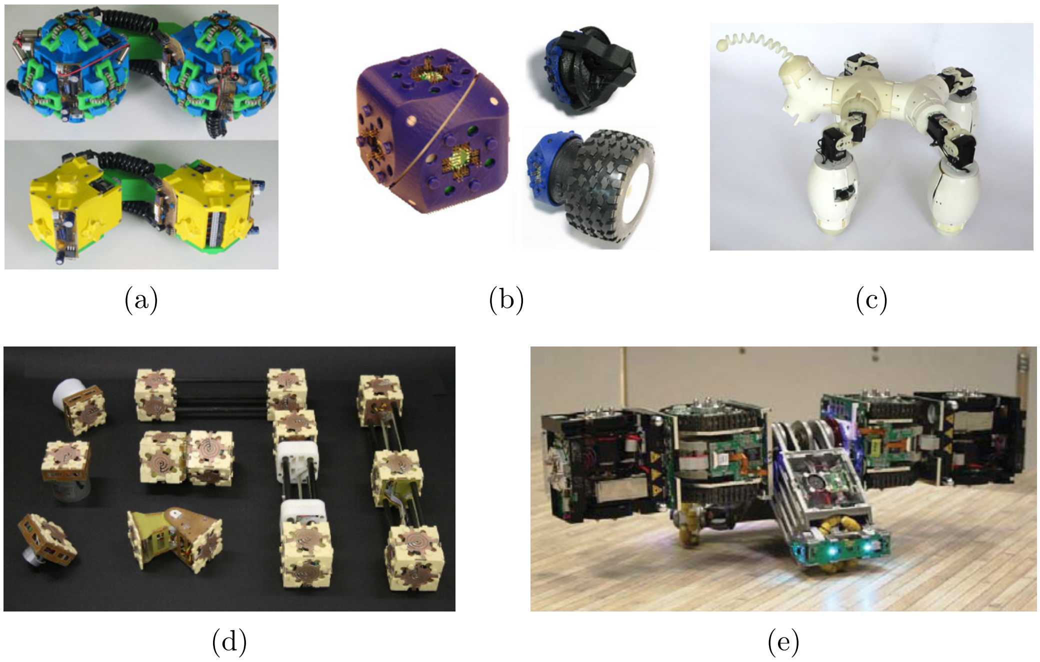


Figure 2.7.: A selection of heterogeneous modular robots; (a) Molecule © 2006 IEEE, (b) Molecubes Extended (used with permission from V. Zykov & H. Lipson), (c) Fable II © 2015 IEEE, (d) EDHMoR [4], and (c) Symbrion & Replicator © 2013 IEEE. Reprinted from [61, 24, 59, 58, 95], respectively.

Moving towards larger numbers of module types are the Thor [25, 84] and SMART [62] systems. Thor, developed by Lyder et al. is comprised of six varied module types, ranging from rotational actuators and structural components to grippers and wheels. Featuring only passive connectors, Thor is intended for manual assembly into rovers and other forms. One unique feature of Thor's design is the ability for multiple rotation modules to be linked together to provide torque to the same drive shaft, passing it through other modules to drive wheels, for instance. Following on, the SMART system is an extension of the RobMAT system [71] developed by Escalera et al.. Featuring the same core module types, the new team have introduced active connectors to the design as well as a range of tool modules for various tasks that can be quickly swapped for others when required.

Cubelets, or roBlocks as they were formerly called, is a commercial education platform that uses small cubic modules that magnetically attach together to form robotic struc-

[4]Reprinted from Robotics and Autonomous Systems, 63 / 2, Faíña, Andrés and Bellas, Francisco and Orjales, Felix and Souto, Daniel and Duro, Richard J, An evolution friendly modular architecture to produce feasible robots, 195–205, Copyright 2015, with permission from Elsevier.

tures with behaviors [57, 96]. Each cube has a specific purpose, be it a sensor input, output (e.g. motor or light), battery, or logic unit. The logic units take data from an input cube or another logic unit and apply an operation to it based on their type before passing it to other logic units or output cubes. By combining multiple inputs, outputs and logic units, relatively complex behaviors can be created, allowing the assembled blocks to react to light or obstacles, for example. The main advantage of Cubelets for educational purposes is that users do not need to program the blocks, and instead construct them to produce the behaviour required.

Fable II [59, 97], by Pacheco et al. is a modular robotics platform intended for creative learning. The platform consists of both active and passive modules that connect together via passive connection mechanisms to form chain and branching structures. Each active module contain electronics, power, and wireless communication, as well as up to two rotational degrees of freedom. By combining the various active and passive modules together, walking robots can be formed (Figure 2.7(c)).

A further heterogeneous system is the EDHMoR [58, 98], developed by Faíña et al.. The EDHMoR system consists of four primary actuating module types, as well as a number of tools and passive modules, such as an electromagnet (Figure 2.7(d)). Each actuating module offers a different motion capability, be it hinge rotation, continuous rotation, extension, or translation. These types allow the creation of structures with fewer modules than would perhaps be required with a homogeneous system.

A system that takes full advantage of the heterogeneous concept is that of the Symbrion & Replicator project [64]. It features three modules types, a Scout, Backbone and Wheel (Figure 2.7(e)). The Scout [22] module is a self-mobile robot featuring tracks around its perimeter for locomotion and sensors to perceive its environment, allowing groups to operate as a swarm robotic system. It is capable of joining with other modules, via four active connectors, one of which can be angled up and down. The Backbone module, called the CoSMO [23], is comparable in size to the Scout, with four active connectors as well as a stronger rotational joint in its center. Additionally, the module has a unique screw drive mechanism, allowing it to translate in two dimensions on a flat surface, as well as perform limited turning [99]. The Wheel module, called Active Wheel [95], features omni-directional wheels allowing it to move around and turn in any direction on a flat surface, giving it an advantage over the Backbone. In addition, its body can raise and lower, allowing its two active connectors to match the height of those on a module the Wheel is attempting to join with. The combination of these three module types shows the advantages of the heterogeneous approach to modular robots, as by working together they are able to self-assemble into a variety of configurations, with Backbone modules acting as the main structure, Wheel modules providing efficient locomotion, and Scout modules providing sensor coverage.

A final heterogeneous system to mention is Swarmanoids [63]. Expanding upon the work of the Swarmbot project, Swarmanoids focused on exploring the collaboration of three different robots, foot-bot, hand-bot, and eye-bot. A foot-bot is a revised version of the s-bot, with the same ability to join with other foot-bots. A hand-bot is an immobile robot with large grippers and a harpoon mechanism, allowing it to anchor itself to a ceiling in order to ascend. An eye-bot is a quadcopter capable of attaching to ceilings and relaying environmental information to the other robots. Demonstrations of these robots showed how they can co-operate to navigate a hallway in order to collect a book from a shelving unit.

2.3. Module Connection Mechanisms

The success of reconfigurable modular systems relies heavily upon the connection mechanism used to join their separate modules together. Such mechanisms need to be capable of withstanding the forces expected by the intended system, provide accurate alignment, and in some cases enable inter-module communication. This has led to a variety of solutions to the connection problem:

- ***Mechanical*** - A traditional approach to the connection of modules is to use a method of mechanical latching to lock them in place. A motor or other form of actuating element, such as a shape memory alloy (SMA), is used to extend hooks or clamp on to posts. Systems such as CONRO [28] and Crystalline [37] employ an approach of passive posts mating with active holes (post-hole), and active latches mating with passive grooves (hook-groove), respectively. A limitation of the CONRO design is that releasing the latch does not automatically disengage the two connectors, instead requiring a separate operation to be performed.

  The M-TRAN III platform [17] uses extendible hooks to overcome the limitation of the post-hole approach. When retracted M-TRAN's connection surfaces lie completely flat with its neighbor's, allowing for translation parallel along the surface, and thus removing the need for a separate operation to pull the connectors apart.

This surface-to-surface connection, while allowing for translation, prevents modules from rotating on axes perpendicular to the surface, requiring clearance to be gained first. This shortcoming is addressed by 3-D Unit [40] and ATRON [16], which emulate a point-to-point connection with their neighboring modules, by extending hooks a significant distance out of their active surfaces. When retracted a gap is produced sufficient for a collection of 3-D Unit or ATRON modules to rotate in place within their lattice position. Examples of actuated mechanical connectors can be seen in Figures 2.8(a, b, and c).

- ***Magneto-Mechanical*** - Mechanisms that employ both a mechanical actuator and permanent magnets can be considered as magneto-mechanical connectors. Their defining feature is that magnets are used to make the connection between neighboring modules, and a mechanical element is used to separate them. A well known example of this technique is on the original M-TRAN and its version II update [43]. It features north and south polarity connectors, arranged on the module such that north always aligns with south within a lattice structure, allowing connections to be automatically made when any two meet. To separate the magnets a mechanical force is applied on the north connectors to make the magnets recede into the surface. This is achieved by the use of SMA coils that, when heated, apply a strong pulling force to overcome the strength of the magnets. Due to the properties of the SMA material used, this process can take over one minute to perform.

  Another method of separating two permanent magnet connectors, as employed on the SMORES platform [21], is to twist one connector relative to another. When a module wishes to disconnect from its neighbor, it extends a rod out of its connector and in to the neighbor's connector to temporarily lock its orientation. The module then proceeds to rotate its connector in order to separate the magnets, before finally retracting the rod. This approach allows for the same rotation method used for joint motion to disconnect the connectors, with a small additional mechanism needed for the actuation of the rods.

  The M-Blocks system [42, 83] uses a unique solution to disconnecting two module surfaces containing permanent magnets. Instead of having an actuated element on the surface, it uses an inertial mass to exert an abrupt momentum transfer onto the module. This transfer allows the module to overcome the magnetic attraction, allowing it to roll from one face to another, as well as jump sections of an assembly. Using an internal actuator to produce an external force allows M-Blocks to be completely enclosed, increasing their robustness.

- ***Electro-Magnetic*** - The use of connectors based upon electro-magnets allows for faster connection and disconnection compared to mechanical based solutions, and enables modules to be created without moving parts, potentially giving them increased robustness. Electro-magnets can be used in one of two ways; either on their own with power applied to create an attractive or repulsive force, as with Catoms [100], or combined with permanent magnets to cancel out the normal attractive force of a connection surface, as used on Molecubes [31].

  A method for overcoming the power requirement of electro-magnets to maintain state is to employ electro-permanent magnets. These are magnets with two different materials, one of which can be influenced by an external coil. When an electro-magnetic field is applied, the direction of one material's field is flipped to either add to or subtract from the other's field, creating a magnet that can be switched on and off. This is put into practice on the Pebbles platform [48], and the updated SMORES system [91].

- ***Electro-Static*** - In a similar manner to electro-magnetic connectors, electrostatics can also be used for joining modules together without the need for moving parts [101]. By applying a voltage to a set of electrodes that form the connector, a charge is created that attracts an opposing connector towards it. This voltage can then be removed once a charge has formed, allowing the connector to be unpowered. In reality however, the charge will leak over time, so will need to be replenished to maintain a given attraction level.

- ***Phase-Change*** - A recent development in module connection is the idea of phase-change connectors [102], that being connectors which join together by melting a material, such as a low melting point solder (Figure 2.8(d)), in order to create a bond with a neighboring connector. The advantage of a connector such as this that there are no moving parts, meaning their size can be relatively small. Additionally, the use of a conductive material allows for electrical connections to be created between two connectors as part of the melting process.

The above list of connection methods cover those used by self-reconfigurable systems. For modular systems that are only intended to be reconfigured by an external operator, three main connection methods exist:

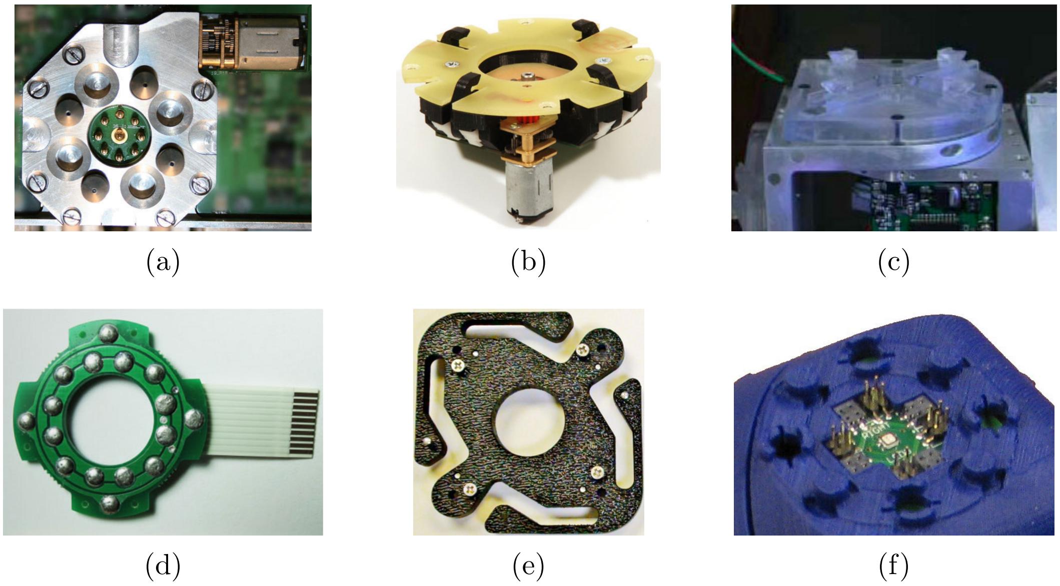


Figure 2.8.: A selection of modular robot connection mechanisms; (a) CoBoLD © 2011 IEEE, (b) Roombots connector [5], (c) SINGO © 2009 IEEE, (d) Solder connector © 2014 IEEE, (e) ModLock © 2012 IEEE, and (f) Molecubes connector (used with permission from V. Zykov & H. Lipson). Reprinted from [104, 82, 105, 102, 103, 24], respectively.

- ***Mechanical*** - Identical to the mechanical and magneto-mechanical connection mechanisms used for self-reconfigurable robots, except the actuating element is replaced by a component that allows a user to grasp and actuate the mechanism manually [62]. Additionally, connection mechanisms can be designed that are only intended for manual actuation, such as ModLock [103] (Figure 2.8(e)).

- ***Magnetic*** - A connection in which permanent magnets are used to hold two connectors together [21, 25]. The polarity and arrangement of magnets can be arbitrary, depending on the intended application.

- ***Friction*** - Modules are joined together by the friction between the features of two connectors. These features are typically posts that fit inside holes on an opposing connector, with the walls creating the friction surface [69] (Figure 2.8(f)). This method of connection can be combined with magnets to create a stronger connection [59].

[5] Reprinted from Robotics and Autonomous Systems, 62 / 7, Spröwitz, A and Moeckel, R and Vespignani, M and Bonardi, S and Ijspeert, AJ, Roombots: A hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot, 1016–1033, Copyright 2013,

Table 2.2.: A comparison of connectors supporting self-reconfiguration in modular robots, with known actuation times. The HiGen connector produced by this project is included for comparison.

System / Connector	Category	Gender	Pathways	Actuation Time (s)
ATRON [76]	Mechanical	Gendered	2	2.4
M-TRAN I/II [43]	Mag.-Mech.	Gendered	3	60 to 180
M-TRAN III [17]	Mechanical	Gendered	5	5
DRAGON [107]	Mechanical	Bi-gendered	12	0.2
Roombots [19]	Mechanical	Bi-gendered	0	2
SMORES [21]	Mag.-Mech.	Bi-gendered	0	0.8 to 2.3
Pebbles [48]	E.-Magnetic	Genderless	1	0.0003
RoGenSiD [108]	Mechanical	Genderless	2	12
SINGO [105]	Mechanical	Genderless	0	25
Solder [102]	P.-Change	Genderless	3	30
HiGen	**Mechanical**	**Genderless**	**12**	**0.2**

Regardless of the method of connection used, all connectors can be categorised as either gendered, bi-gendered, or genderless. A comparison of various connectors and their gender is shown in Table 2.2. For a review of latching mechanisms beyond the area of modular robotics, please refer to [106].

2.3.1. Gendered

Connection mechanisms are gendered if they feature two distinct types of connectors that mate together. One connector type contains an active element, such as a latch or electromagnet, whilst the other contains passive elements, like posts or permanent magnets. These connector types are typically referred to as male and female, although either one can contain the active element depending on the specific implementation. Manual connectors with only passive elements can also be considered gendered, depending on whether one type contains a different set of passive features to the other. An example

with permission from Elsevier.

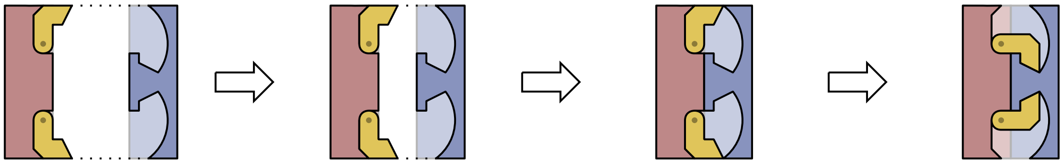


Figure 2.9.: An example connection sequence for a gendered hook-groove mechanical connection mechanism. The red and blue shapes represent the two connectors, with the yellow shapes being their hooks.

connection sequence between active and passive mechanical hook-groove connectors is shown in Figure 2.9.

In the case of magnetic connectors, the polarities of the magnets in each connector type determines their respective gender, with north being used to denote male and south used to denote female, for example. A connection is therefore achieved by aligning the north facing magnets of one connector with the south facing magnets of the opposing connector. To make magnetic connectors in to an active connection mechanism, either side would need to feature an actuator to displace the magnets in order to break the connection [78, 21], or use electro-magnets to cancel out the polarity of the permanent magnets [49].

A key aspect of active gendered connectors is that only the side containing the active element is able to initiate a connection or disconnection from a neighboring module. As such, if the module with the active connector fails, the one with the passive connector has no means of detaching the failed module, potentially restricting a modular robot's ability to perform a task.

Examples of modules that feature gendered connectors include: ATRON [16], CEBOT [50], CONRO [28], Crysalline [37], Fracta [41], Sambot [20], S-bot [54] and Molecule [61].

2.3.2. Bi-Gendered (Hermaphrodite)

Often referred to as genderless in much of the literature, bi-gendered or hermaphroditic connectors extend upon the gendered approach by combining both gender elements in to a single connector design, with the male elements of one side connecting to the female elements of the other side. This allows for a connection between modules to be established using just a single set of male-female elements, opening up the possibility for inactive modules to be docked with and manipulated.

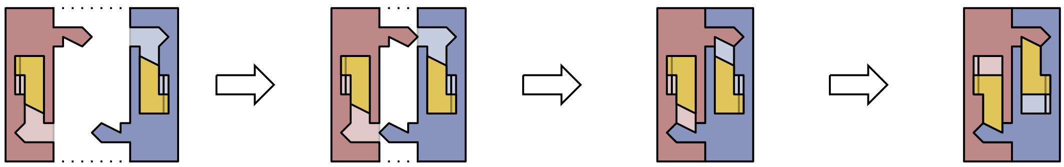


Figure 2.10.: An example connection sequence for a bi-gendered post-hole mechanical connection mechanism. The red and blue shapes represent the two connectors, with the yellow shapes being their latches. Only a single side needs to actuate in order for a connection to be made; however, if both sides actuate and one subsequently fails, it is no longer possible for the remaining side to separate from the failed side.

When two active bi-gendered connectors activate, a connection is produced that is equivalent to having two sets of gendered connectors joined, offering increased strength and redundancy. Unfortunately, such bi-gendered connection mechanisms suffer from the same limitation as gendered connectors, in that if one side fails when connected, it is not possible for the two connectors to be separated. The inability for modules to disconnect if one side fails is the key distinction between the bi-gendered approach and genderless mechanisms, as it may prevent self-reconfigurable modular robots from performing self-healing, whereby they discard damaged modules for new ones to allow the system to continue with its objectives. An example connection sequence between two active post-hole bi-gendered connectors is shown in Figure 2.10.

In the case of manual connectors based on magnets or friction, as there are no active elements, the disadvantage of not being able to disconnect becomes inapplicable. This makes these designs genderless despite containing male elements that join with female elements. As such, arrangements like north facing magnets joining with south facing magnets on SMORES [21], and posts fitting inside holes on Molecubes Extended [69] (or both on Thor [25]), can correctly be referred to as genderless.

Examples of active bi-gendered connection mechanisms include CoBoLD [104] and DRAGON [107], as well as the connectors on the 3-D Unit [40], Roombots [19], and PolyBot [32] platforms.

2.3.3. Genderless

Recent works have seen the creation of connection mechanisms that overcome the limitation of active bi-gendered designs, by offering single-sided disconnect, that being the ability for either side to freely disconnect from the other without mediation. This is achieved by the connectors containing active elements that join with the active elements of an opposing connector, rather than passive elements. As such, when one side wishes to disconnect from its neighbor, only that side's active element needs to be actuated. This makes single-sided disconnect an inherent property of all genderless connection mechanisms. An example connection sequence between two active hook-hook genderless connectors is shown in Figure 2.11.

Two main methods for mechanical latching have been developed so far, contracting hooks and rotating hooks. The SINGO connector [105] for the Superbot platform [45], and GHEFT [109] achieve genderless latching using a chuck-like arrangement of hooks that translate in and out from a central point along the surface. This design allows an opposing connector to contract its hooks around those of the other whilst the other simultaneously expands its hooks to meet at a mid-point. If one side fails the other can actuate its mechanism in the appropriate direction to separate. Unfortunately, the use of a chuck requires mediation between connectors prior to connecting in order to assign movement roles, meaning the operation of each connector is not strictly genderless. The RoGenSiD connector [108] for the ModRED platform [30] creates a genderless connection using a rotating plate with hooks arranged around it. This plate is able to turn in a clockwise direction relative to its surface normal to mate with an opposing connector performing the same relative operation, removing the need for prior role mediation. To prevent unwanted disconnection as a result of rotational forces, a number of posts are used to maintain alignment. These posts introduce the mentioned issues of gendered mechanical designs, by requiring an operation external to the connector to fully separate the two surfaces, which in this case relies on a translational actuator within the ModRED [30] platform on which RoGenSiD features.

Genderless magnetic connections are achieved by mounting magnets perpendicular to a connector's surface, allowing both their north and south poles to be exposed. This arrangement allows two connectors to attach in a genderless manner, and halves the overall number of magnets needed by a bi-gendered design. By using electro or electro-permanent magnets in this arrangement, the active element of the connection becomes the magnetic field from each connector, meaning that both fields need to be active in order for a connection to be formed, but only one needs to deactivate in order to disconnect, thus fulfilling the single-sided disconnect property. Additionally, the perpendicular magnet allows the same field polarity to be applied to both connectors, making their control genderless as well. The use of perpendicularly mounted electro-permanent magnets

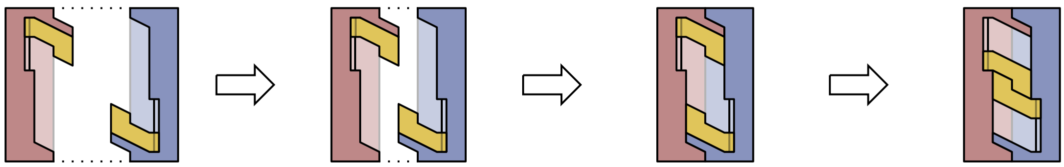


Figure 2.11.: An example connection sequence for a genderless hook-hook mechanical connection mechanism. The red and blue shapes represent the two connectors, with the yellow shapes being their hooks. Of the three mechanism types, genderless is the only one to allow for single-sided disconnect.

can be see on the Pebbles [48] and SMORES-EP [91] platforms.

The recently developed Solder connector [102] for the Soldercubes platform [44] creates a genderless connection via the use of a phase-change material. Each connector has several solder balls on its surface that are heated by a resistor array on the reverse of the mechanism. The solder balls act as the active element of the connector, and join with those of an opposing connector when brought together. Due to the solder being shared between the two connectors once joined, only a single connector needs to apply its heater in order for disconnection to occur, thus fulfilling the single-sided disconnect property. The Solder connector does however require an external operation in order to separate the two connectors, so that the solder can separate and re-form in to balls.

2.4. Module Control

Unlike traditional robotic systems, the control of modular robotic systems requires the coordination of many independent but connected robotic units in order for a given task to be performed. The methods in which modular robots can be controlled are dependent on the processing capabilities of their individual units, with units ranging from having no processing at all [93] to units having full-featured computers [23]. Control of modular robots falls in to two categories, centralized and distributed.

Centralized control of modular robots involves each individual module acting upon instructions sent by a single control unit, be it external to the robot or a designated module within the robot itself. The implementation of centralized control is dependent on the processing capabilities of each individual module, with those lacking any processing having their sensors and actuators controlled directly from an external unit via tethers [38, 94, 75], and those with processing either using a dedicated control module [24, 29] or having an arbitrary module assigned as a control unit, that other modules are communicated with by a shared bus. The advantage of centralized control is that it allows for algorithms to be employed that require global knowledge of the state and configuration of a modular robot, for example, inverse kinematics for having modules manipulate objects. The disadvantage of centralized control is that it compromises the robustness of a modular robot, as a failure of the control unit, either on a software or hardware level, results in a failure of the entire modular robot.

Distributed control of modular robots involves each individual module being in control of itself. The implementation of distributed control requires that each module contain some degree of processing, as well as the ability to communicate locally with other modules. Local communication allows the state of one module to affect that of its neighbors, enabling collective behaviors to emerge from a configuration of modules regardless of the specific order the modules are arranged in. The advantage of distributed control is that it does not require each module to be aware of the complete state of a modular robot, instead acting based upon local interactions. This enables robust operation, as the failure of one module has little impact on the whole robot, unlike centralized control. The disadvantage of distributed control is that producing some behaviors can be more challenging as existing algorithms may assume global information of the robot, which each module may not have.

Both centralized and distributed control have their place in modular robotics. As such, a number of control strategies have been developed that use these two methods of control.

2.4.1. Centralized Control Strategies

Centralized control of modular robot locomotion can be performed via the use of gait control tables [65].

A gait control table consists of a list of steps that represent a complete motion cycle of a modular robot. Each step contains a movement entry for each module and a list of trigger modules. When a gait starts, a central control unit instructs each module to move based on their specific entry in the current step. The control unit then waits for the modules listed as triggers to report their motion as being completed, at which point the next step is moved to and new instructions are sent out. This process repeats until the end of the table is reached, at which point the process loops back around. The advantage of gait control tables is that they offer a method for implementing locomotion that consists of quantized steps; however, they are static meaning that such locomotion cannot adapt to disturbances or changes in the environment. Simulated demonstrations of gait control tables have been shown, where a set of Polypod modules were able to produce an inchworm and rolling-track motion [65].

2.4.2. Distributed Control Strategies

Distributed control of modular robot locomotion can be performed in two main way, either by using hormone-inspired control [110], or by using central pattern generators [111].

With hormone-inspired control, each module in a modular robot has a table of actions they can perform. These actions can be triggered by each module either receiving a hormone from their neighbors or by some internal logic. Once a module completes an action it can send a hormone to its neighboring modules. The result of this is that behaviors emerge from a set of connected modules regardless of their specific order, with it being possible for the behavior to scale as modules are added and removed. This method of locomotion control does require that there be a module to initiate hormone sequences, introducing an element of centralized control, but this can be determined base on its position within a configuration. For example, with a snake-like configuration, a module with no module in front of it could consider itself to be the snake's head, with all other modules being the middle or tail of the snake. Which ever way the assignment of the hormone initiator is chosen, this control method remains distributed as no single module is directly in control of all the others. Note that like with gait control tables, hormone-inspired control is based upon each module having quantized steps they perform.

Using central pattern generators (CPGs) for module locomotion removes the restriction on quantized steps that hormone-inspired control has. Each module features their own CPG that produces an oscillating output signal that is used to drive their joints. The CPGs of neighboring modules are linked together such that the output of one affects the amplitude, phase, and frequency of another. As such, walking gaits and other motions can be formed using this method, as has been demonstrated by the Roombots platform [82]. An advantage of this approach is that CPG parameters can also be controlled by sensors, allowing the gait of the robot to adapt to environmental changes, unlike hormone-inspired control. It should be noted however that depending on the complexity of each CPG, correct assignment of parameters to neighboring module outputs may be non-trivial, requiring some form of learning algorithm such as a genetic algorithm to be employed.

3. Genderless Connector

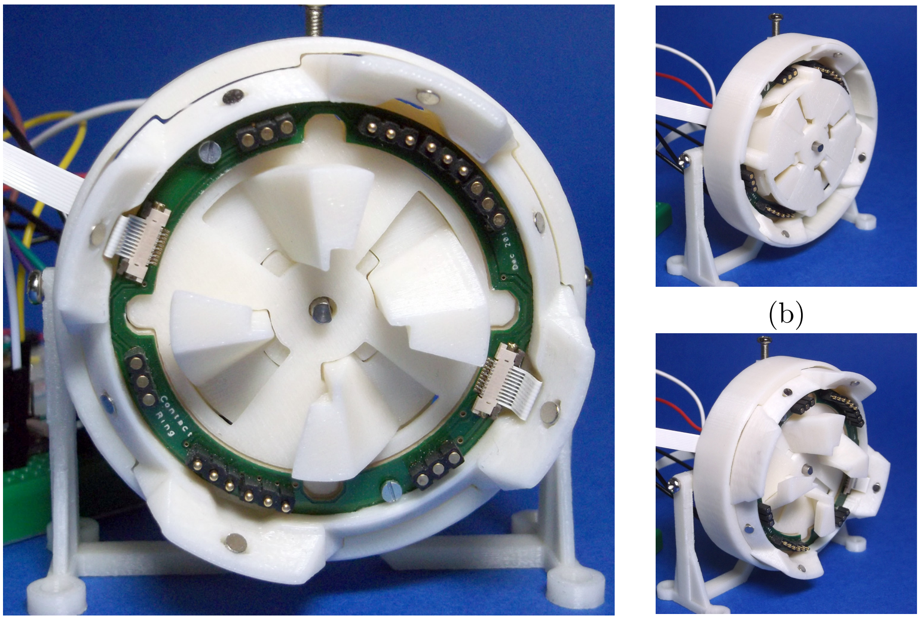
The content of this chapter is derived from the author's following published work [112]

3.1. Introduction

The concept behind modular robotics is that rather than building a bespoke system for a given task, a kinematic approximation is assembled out of a number of ready-made entities, known as modules, instead. These modules have the advantage that they can be easily replaced when damaged or inoperable, as well as rearranged when their task or environment changes.

Modular systems can be identified as either manually reconfigurable or self-reconfigurable [8], based on the method used for connection. The former of these requires an external operation or user to separate and reattach modules, whereas the latter gives modules the ability to perform this action themselves. Self-reconfigurability presents a challenge for the field, as reliable connection mechanisms need to withstand the expected forces, provide accurate alignment, and in many cases feature inter-module communication.

This chapter presents a novel 90 degree symmetric connection mechanism for self-reconfigurable modular robots, called HiGen (see Figure 3.1). It is capable of actuating in a short time, and features a genderless latching method that allows for independent detachment from a neighbor, without mediation. This is important for the self-repair of modular systems, as malfunctioning modules can be discarded to allow the remaining modular assembly to continue with a given task. To benefit modular systems that form lattice structures, HiGen is capable of extending and retracting its latching mechanism as part of the actuation process, creating clearance between two neighboring modules, easing self-reconfiguration. This is demonstrated in Figures 3.1(b) and 3.1(c). A further



(a) (b) (c)

Figure 3.1.: The HiGen connector, shown (a) face-on, and to the side in its (b) retracted and (c) extended states. © 2014 IEEE.

feature of the connector is the integration of multiple electrical pathways between connected modules, allowing for both local communication between neighbors and global communication to all modules within an assembly, as well as power transfer. These pathways are automatically made and broken as part of the extension and retraction operations. Many of these aspects have been demonstrated on systems in the past (e.g. [107, 76, 105]); however, HiGen is the first connector to combine them all into a single unit.

The remainder of this chapter presents the requirements that resulted in the HiGen connector (Section 3.2), details the connector's mechanical and electrical design (Section 3.3), and presents experiments conducted with two units (Section 3.4). Finally, Section 3.5 concludes the chapter.

3.2. Requirements

From exploring the literature surrounding connection mechanisms for self-reconfigurable systems, it is identified that a mechanical or magneto-mechanical solution would be preferred for the new connector being developed. This is because mechanical solutions can be easier to implement than other solutions, can offer decent connection strength, and can be designed to maintain a connection without consuming power. They can also be genderless so as to avoid limitations in the ways in which modules that feature them can be connected together. A desirable quality for the new connector is full separation from a neighbor so as to open up the possibility for free in place rotation of any module that incorporates it. On the electrical side, it is important for modules to be able to pass power and communication through the new connector as well as for some level of neighbor identification to be performed. These high-level goals for the connector give rise to the following systematic requirements:

3.2.1. Mechanical

The connector shall:

C.1.1 - Be primarily constructed using 3D printing technology. A Stratasys Mojo [113] is available in-house, capable of printing parts in ABS plastic with soluble supports

C.1.2 - Feature a self-actuating mechanism that is able to form a mechanical connection with a neighboring connector

C.1.3 - Be genderless in both its design and its operation

C.1.4 - Be four times symmetric to allow for its use by cubic lattice modules

C.1.5 - Be able to maintain a connection with a neighboring connector without power

C.1.6 - Fully separate itself from a neighboring connector when disconnecting, such that an external movement is not required to:

a) Enable translation along and away from the connector's surface

b) Enable rotation in an arc away from the connector's surface

c) Fully disconnect any electrical pathways

C.1.7 - Feature a mechanism design that can be adapted to a passive version

3.2.2. Electrical

The connector shall:

C.2.1 - Contain its own microcontroller for controlling the actuation process of the mechanism

C.2.2 - Feature a motor and driver for actuating the mechanism

C.2.3 - Be able to detect:

a) The actuation state of the mechanism

b) If it is joined to a neighboring connector

c) The relative orientation of a neighboring connector

C.2.4 - Allow for multiple electrical signals to be passed through it to a neighboring connector

a) Both power and communication pathways shall be incorporated

C.2.5 - Allow for actuation of the mechanism to be triggered directly by a user and by an external control signal

C.2.6 - Be externally powered

C.2.7 - Feature indicator LEDs to visually report operation state

C.2.8 - Be reprogrammable

3.2.3. Environmental

The connector shall:

C.3.1 - Operate at a standard humidity for an indoor laboratory environment

C.3.2 - Operate within a temperature range of 20 to 30 degrees Celsius

3.2.4. Performance

The connector shall:

C.4.1 - Be no larger than 100 mm in any direction

C.4.2 - Actuate in a time that is no longer than one second

C.4.3 - Tolerate translational misalignment of at least ±2 mm in any direction

C.4.4 - Tolerate rotational misalignment of at least 5° in any direction

C.4.5 - Be able to support a payload of at least 2 kg

C.4.6 - Become operational within one second

3.2.5. Reliability

The connector shall:

C.5.1 - Be capable of actuating hundreds of times before component failure

C.5.2 - Be able to detect if the mechanism is failing to actuate and stop the operation

3.3. The HiGen Connector

The HiGen connector consists of five 3D printed ABS plastic components, a custom connection board, a DC geared motor, two contact switches, and control circuitry. A breakdown of the connector is shown in Figure 3.2. The design is cylindrical, measuring 71 mm in diameter, with a depth of 32 mm at its thickest point, and 16 mm at its thinnest, with a weight of 67 g. The dimensions fall below the 100 mm specified by requirement C.4.1. The motor is housed directly in the center of the design and features a 298:1 gearbox, giving it a quoted speed and torque at 6 V of 79 rpm and 338 mNm, respectively. Contact switches are used to detect the connector's retracted and extended states, satisfying requirement C.2.3a.

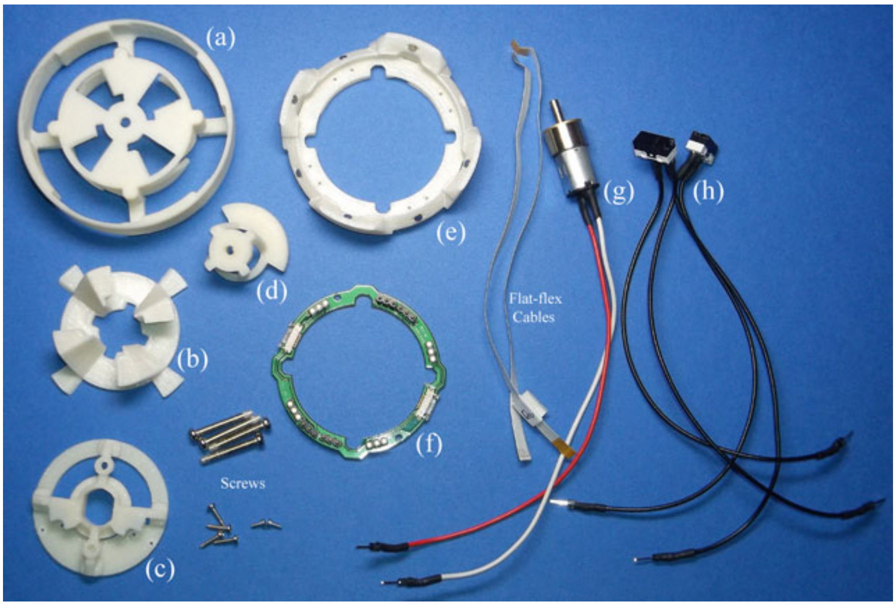


Figure 3.2.: A breakdown of the HiGen connector, showing the (a) housing, (b) docking hooks, (c) motor and switch mount, (d) drive shaft, (e) shroud, (f) connection board, (g) DC geared motor, and (h) contact switches. © 2014 IEEE.

3.3.1. Mechanism Details

The design of HiGen features four hooks placed radially around a central axis, with the motor rotating them between the two states. These hooks mate with an identical set of hooks on an opposing connector by passing over each other, forming a hook-to-hook relation. This arrangement enables single-sided disconnect in the HiGen design. The use of rotational latching, as opposed to translational chuck latching as in the SINGO [105], allows for the operation of two joining connectors to be identical, simplifying the control involved in creating a connection. Both the design and operation of the connector's hooks were a result of requirement C.1.3.

A shroud component is used to avoid any rotational forces around the central axis causing unwanted disconnects. This element mates with the opposing connector via the use of four protrusions, which are tapered to provide a degree of auto-alignment. The arrangement of hooks and protrusions allow for connections at 90 degree intervals, satisfying requirement C.1.4. Once two connectors are joined they can maintain a connection even when power is removed, as per requirement C.1.5. An enclosed area is created by the shroud that prevents external manipulation of the hooks, and motor gearbox friction acts to limit any momentum transfer from indirectly separating them.

In addition to aiding alignment, the shroud also houses the connection circuit board to allow for electrical signals to be passed between modules, as per requirement C.2.4. This board features a number of spring-loaded pins (covered in more detail in Section 3.3.2), each with a quoted force of 0.6 N at half compression, enough to cause strain on the motor. To overcome this force 8 neodymium magnets are included within the protrusion surfaces, enough to counteract the spring force but not enough to hold a connection, unlike magneto-mechanical designs. A caveat of using magnets within the shroud is that a force needs to be applied in order to electrically disconnect two joined connectors. To apply the necessary disconnection force, as well as to facilitate the connector's use on modules within large configurations, the hooks and shroud extend out of and retract into their housing as part of the actuation process. Translation of these elements is achieved via the use of helical guides within the connector's housing, causing the hooks to spiral in and out of the mechanism. Animation steps of the process during connection are shown in Figure 3.3. This approach not only allows electrical contacts to be made and broken without modules being required to move, it also produces a clearance between neighbors of 12 mm. These two aspects satisfy all parts of requirement C.1.6. Inspiration for this functionality was taken from the ATRON [16] and 3-D Unit [40] systems, which also connect with their neighbors by extending hooks some distance above their respective surfaces. This concept, as applied to HiGen, is illustrated in Figure 3.4, where a central module is free to rotate within a lattice structure.

3.3.2. Electrical Details

Electrical connections between two HiGen connectors are made by a custom circuit board housed within the shroud, featuring 12 spring-loaded pins, 12 static pads, and 2 flat-flex cable connectors for interfacing with external circuitry. The flat-flex cable connectors have a maximum quoted current per contact of 500 mA, whereas the pins and pads can handle up to 3.5 A each. In total the connection board offers six wired channels between neighboring modules, all with separate incoming and outgoing pathways. The roles of these are: *Ground, Power, Connection Sense, Two-wire Global Communication* (e.g. I^2C, CAN bus), and *Local Communication* (e.g. serial). Not all of these roles require separate directional pathways, so instead the directions can be combined to add redundancy or increase current capacity, as in the case of power transfer. Note that the

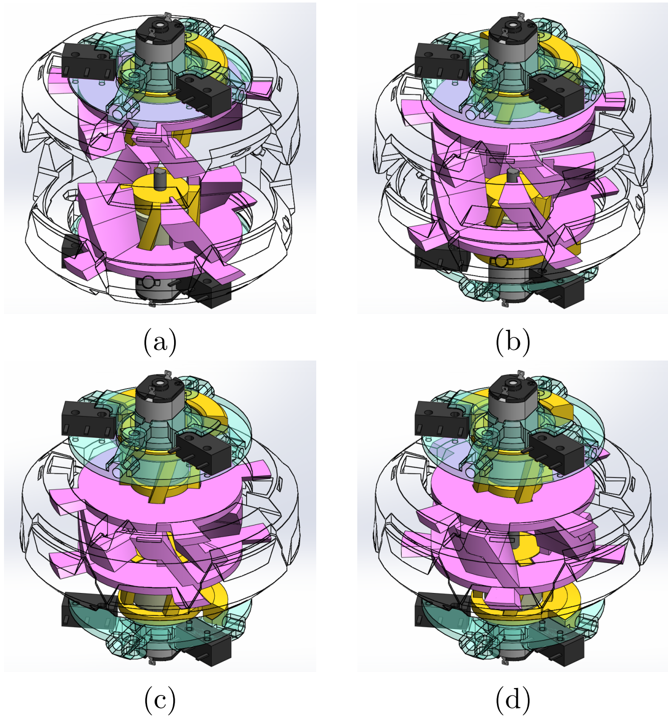


Figure 3.3.: Snapshots of the connection sequence of two HiGen connectors, showing the hooks (in purple) extending behind each other and locking in place. The shroud is transparent to help show the motion of the hooks. © 2014 IEEE.

connection sense pathway was included to allow the connector to detect if it is joined to a neighboring connector, satisfying requirement C.2.3b.

To account for the four times symmetric nature of the HiGen connector, a staggered placement of contacts is used similar to that of M-TRAN III [17], but in a bi-gendered formation. In this case, the outgoing pins are duplicated by 180 degrees, and the incoming pads by 90 degrees, resulting in double the number of contacts necessary for a single orientation interval (24 versus 12). In addition to this, the incoming pads for half of the channels are 180 degrees offset from the rest to ensure that the spring-loaded contact force of two joining boards remains roughly central regardless of connection orientation. The specific placement and intended role of each contact is shown in Figure 3.5. Figure 3.6 shows how the pins and pads of two boards make contact at each orientation interval.

To perform orientation detection, as per requirement C.2.3c, the local communication channel is separated out into two outgoing and two incoming pathways (A and B). Depending on the orientation of the connection, a unique arrangement of these pins

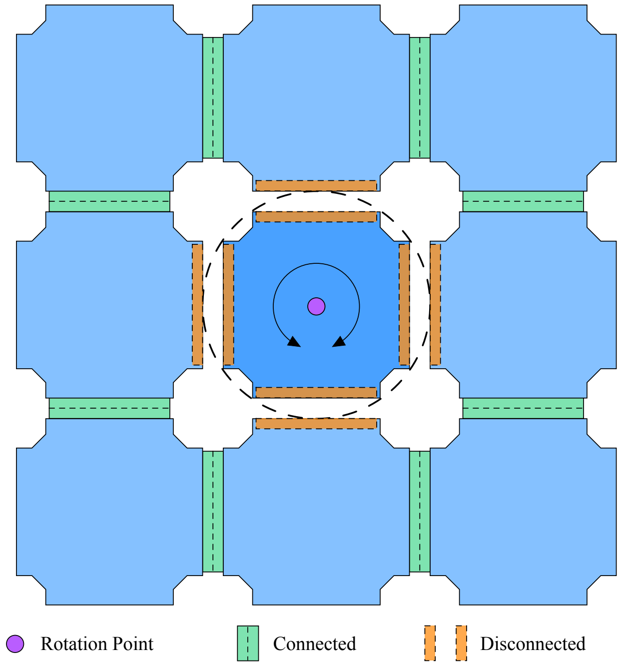


Figure 3.4.: A grid of square modules featuring HiGen, showing how a central module is able to rotate in place within a lattice structure, without neighbors being required to move to provide clearance. © 2014 IEEE.

and pads is produced between one connector and its neighbor (e.g. A-A, A-B, B-A, B-B), such that only a single pathway is formed. By identifying the transmitting and receiving pair of contacts the orientation between two connectors can be determined. Once identified the channel can return to being used for local communication. This is in contrast to solutions such as that on the UBot modular platform [114], which has dedicated pins that can be read to discover the orientation state.

3.4. Experiments

In order to validate the HiGen connection mechanism two complete units were manufactured. The units are controlled using two separate circuits consisting of an Arduino Pro Mini 16 MHz microcontroller, a motor driver, and a number of buttons and light emitting diodes (LEDs). Power is provided by a bench supply running at 6 V, with electronics regulated down to 5 V on board each Arduino. The connectors were tested for actuation and connection time, electrical connectivity, connection repeatability, and load capacity. Figure 3.7 shows the setup used for conducting connection trials.

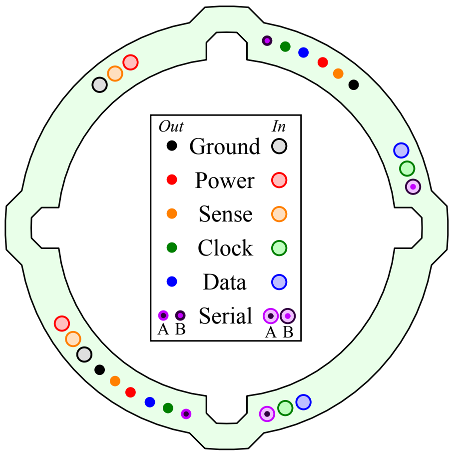
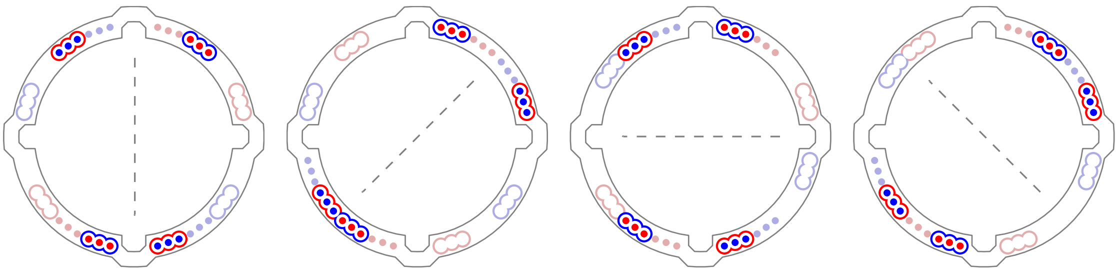


Figure 3.5.: The arrangement and roles of the pins (dots) and pads (circles) on the contact circuit board. Note that the local communication channel, labelled Serial, has all its pins and pads separately accessible. © 2014 IEEE.



(a) 0° (b) 90° (c) 180° (d) 270°

Figure 3.6.: An illustration of the pins (dots) and pads (circles) that make contact when two connectors are joined, at each orientation interval. The bottom board contacts (red) remain fixed while the top board contacts (blue) are flipped and rotated in a clockwise direction from (a) to (d). The dashed lines indicate the resultant mirror axes from these combined operations. © 2014 IEEE.

3.4.1. Actuation and Connection Time

To measure the transition time between HiGen's retracted and extended states, a logic analyser was connected to the motor control lines and the two contact switches of each unit in turn. The time from the initial trigger event until the related contact switch gets pressed and settles is used as the actuation time measure.

A series of 10 actuations were conducted with each connector in isolation. The results of these trials are:

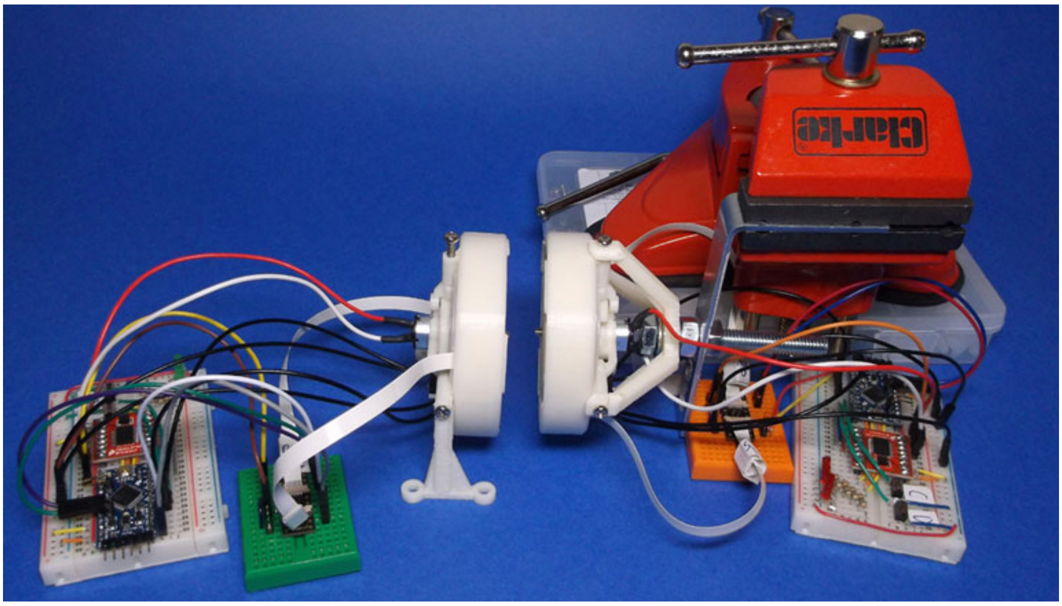


Figure 3.7.: The apparatus used to perform connection trials (excluding power supply and logic analyser). The left connector is free to move on the surface, whereas the right is fixed but able to rotate around the connection axis. © 2014 IEEE.

- Unit One - Average extending time 0.239 s ±0.005. Average retracting time 0.242 s ±0.003.
- Unit Two - Average extending time 0.196 s ±0.001. Average retracting time 0.189 s ±0.001.

The discrepancy between the two units' times may be attributed to differences in the surface friction of the mechanisms as a result of using 3D printing, affecting the final motor speeds when subjected to the same 6 V supply.

The motor within HiGen offers high actuation speed at the cost of low connection torque. To determine if this torque has any detrimental effect on the connection process the two HiGen units were connected and disconnected over a series of 10 further trials, with both units receiving a simultaneous trigger pulse. The results of these tests are:

- Unit One - Average connection time 0.252 s ±0.004. Average disconnection time 0.248 s ±0.006.
- Unit Two - Average connection time 0.198 s ±0.003. Average disconnection time 0.205 s ±0.001.

These timings show that connecting to and disconnecting from a neighboring mechanism only marginally effects the actuation time of each unit. The results demonstrate that the HiGen design surpasses performance requirement C.4.2 by being significantly faster than the existing selection of genderless mechanical connection mechanisms, and on par with the fastest bi-gendered designs.

3.4.2. Electrical Connectivity

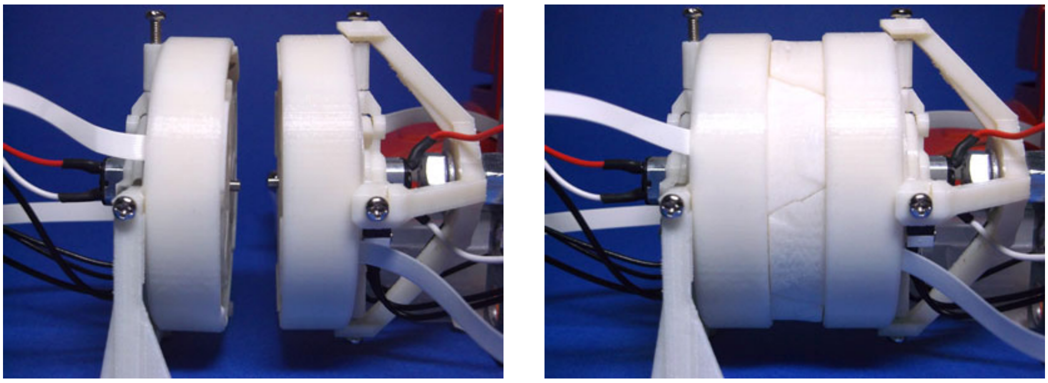
Experiments were conducted with the two HiGen units to verify the successful connection of the electrical pathways at each 90 degree orientation interval. For this the connection sense line of each unit was wired up such that connecting with a neighbor would pull the line low. This way, either microcontroller is able to know if it is electrically connected to another mechanism even if that mechanism is not receiving power. This is useful for situations where a tool has no internal power, and is instead powered via the connection.

Orientation detection was tested by mating the two connectors at different orientation intervals. Each Arduino microcontroller, upon detecting a connection via the sense line, initiates a transmission along both its serial communication lines. The other microcontroller then determines its orientation by detecting which message is received on which pathway. For instance, if message A is received by input B then the connectors are 90 degree offset from each other. The microcontrollers are then able to communicate via the connected input and output pathways, until they detect a connection being broken. Reconnection automatically initiates the orientation detection routine.

3.4.3. Connection Repeatability

The two HiGen units were brought together to test their ability to connect under different alignment conditions. Initially 10 trials were conducted with both units actuating simultaneously, to test the designed separation distance of 12 mm (see Figure 3.8). This presents the well aligned case, and was successful for 100% of the trials. An additional 10 trials were conducted at a closer distance of 6 mm to verify the connector's ability to push its neighbor towards the designed separation distance, with all being successful.

To test the connector's ability to handle other forms of misalignment, and whether the amounts match or surpass the values specified by requirements C.4.3 and C.4.4,



(a) (b)

Figure 3.8.: Two HiGen connectors being tested for connection repeatability at the designed separation distance, (a) detached and (b) connected together. © 2014 IEEE.

Unit One was placed in its extended state, whilst Unit Two repeatedly connected to it. The misaligned cases considered here are detailed in Figure 3.9. A connection was deemed to be a success if the two units fully joined together and were able to exchange handshake messages, as indicated by LEDs illuminating on both microcontroller boards. 200 misalignment trials were conducted in total. Table 3.1 shows the success of these trials, and includes the earlier 20 trials. Note that resetting the experiment after each misalignment trial involved the single-sided disconnect of Unit Two from Unit One.

3.4.4. Load Capacity

A final test was performed on the HiGen connector to determine its load carrying capacity. This was achieved by connecting the two units together, suspending them in a vertical orientation and hanging a mass below. A vertical orientation in this context means that the surface normals of the connectors are parallel to the axis of gravity. A spring balance was used to measure the load. During this test the connectors were disconnected from the bench power supply, due to limited cable length to the microcontrollers.

The combined assembly of two connectors and mounting hardware weighed 145 g. When suspended, it was capable of supporting a load at the limit of the measuring instrument, which was 2 kg. Although this is not a thorough test of the load capacity of the mechanism, it gives an indication that 3D printed parts are already suitable for use on

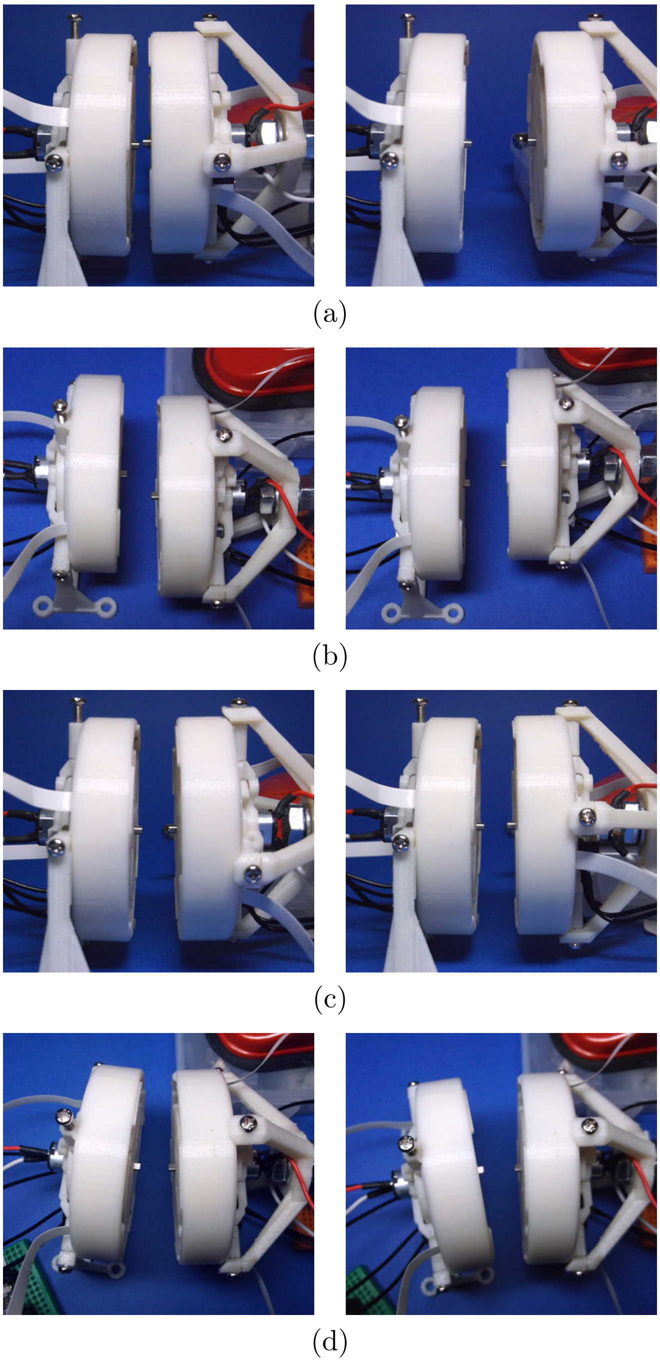


Figure 3.9.: Four ways two HiGen connectors may be misaligned: (a) parallel translation, (b) perpendicular translation, (c) roll rotation, (d) yaw rotation. Each pair of images shows the range of misalignment tested. Unit One is on the left, and Unit Two is on the right of each image. © 2014 IEEE.

Table 3.1.: Results of performing misalignment trials on HiGen

Test	Parameter	Successes
Parallel Translation	6.0 mm	10 of 10
Well Aligned	12.0 mm	10 of 10
Parallel Translation	13.5 mm	10 of 10
Parallel Translation	15.0 mm	6 of 10
Parallel Translation	16.5 mm	5 of 10
Parallel Translation	18.0 mm	0 of 10
Perp. Translation	+5.0 mm	1 of 10
Perp. Translation	+2.5 mm	4 of 10
Perp. Translation	-2.5 mm	3 of 10
Perp. Translation	-5.0 mm	0 of 10
Roll Rotation	+12°	2 of 10
Roll Rotation	+8°	9 of 10
Roll Rotation	+4°	10 of 10
Roll Rotation	-4°	10 of 10
Roll Rotation	-8°	9 of 10
Roll Rotation	-12°	3 of 10
Yaw Rotation	+15°	3 of 10
Yaw Rotation	+10°	10 of 10
Yaw Rotation	+5°	10 of 10
Yaw Rotation	-5°	10 of 10
Yaw Rotation	-10°	9 of 10
Yaw Rotation	-15°	2 of 10

connectors, as they support the mass specified by performance requirement C.4.5. If the mechanical parts were made of metal or another high-strength material, the load carrying capacity of the HiGen mechanism could be further increased.

3.5. Summary

This chapter presented HiGen, a novel mechanical genderless connection mechanism for self-reconfigurable modular robots. The mechanism is four times symmetric and genderless in both its design and operation. This is achieved using a rotary hook-to-hook relation between connectors that does not require power to be maintained. The use of hooks in this manner allows HiGen to separate from a neighboring connector without mediation, benefiting the self-repair capabilities of modular systems. It also removes the need for gender roles to be assigned to each connector, as is a limitation of chuck-based genderless designs. External manipulation of the hooks is avoided by use of a shroud, which also aids in connector alignment.

In contrast to previous genderless mechanisms, HiGen features multiple outgoing and incoming electrical pathways, enabling the concurrent use of several communication protocols as well as power sharing techniques. Additionally, a subset of these pathways have special roles that allow for connectors to not only detect the presence of a joined connector, but also its relative orientation, aiding in modular robot configuration discovery.

HiGen is the first genderless connector in which latching elements are able to retract into the mechanism. Not only does this ability disconnect the electrical pathways between connectors without requiring an external operation to separate them afterwards, it also creates clearance between the connectors that can be exploited by robotic modules for free in place rotation within lattice structures.

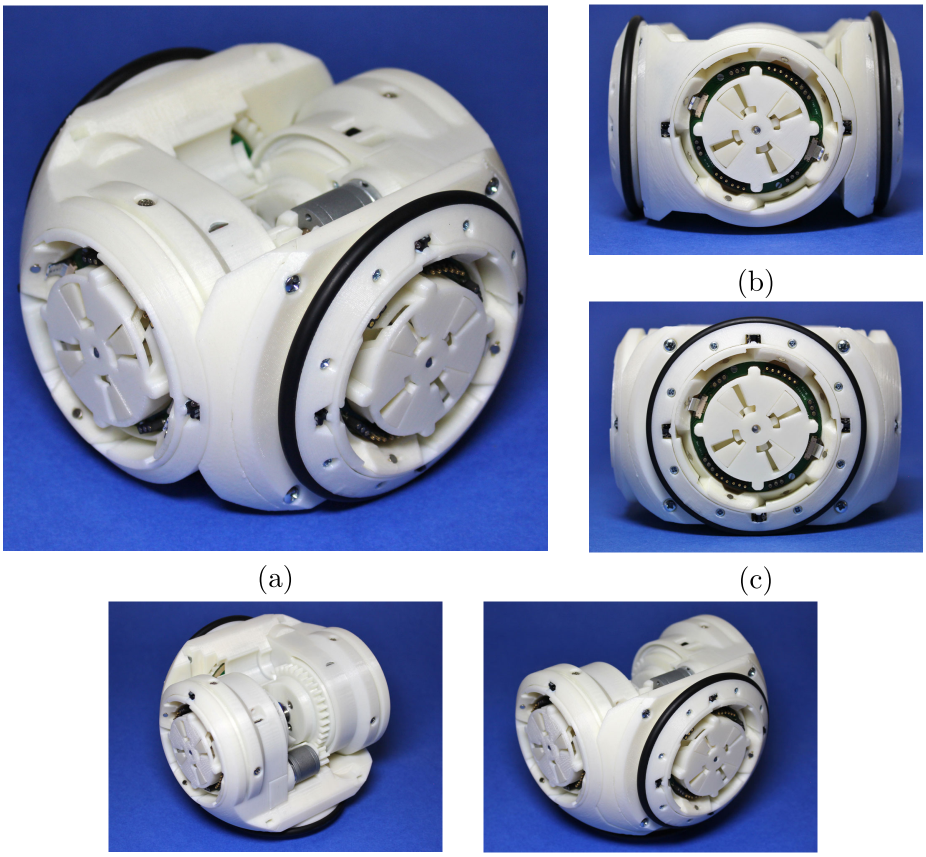
Two full HiGen prototypes were built, and over 200 trials conducted to validate their capabilities. From this it was discovered that HiGen is the fastest connection mechanism of its kind, minimising the time taken for modules to connect and disconnect when performing complex self-reconfigurations. Additionally, despite it being constructed of 3D printed parts, the design is capable of holding a vertical load of at least 2 kg.

4. Hybrid Module

4.1. Introduction

Modular robotics has seen numerous advances over the past decades, with the likes of M-TRAN III [17] and ATRON [16] successfully demonstrating the self-reconfiguration and collective motion of large chain and lattice structures. Each module within a modular robot is relatively simple, with typically only one or two degrees of freedom (DOF), allowing many modules to be produced at relatively low cost. Unfortunately, this means that such modules have limited or no mobility outside of a configuration. Efforts have been made to address this, with swarm systems gaining the ability to self-assemble [88], and modular systems gaining dedicated drive mechanisms to provide efficient single module locomotion [20, 23]. These systems demonstrate the advantages of mobile modular robots, but compromise module self-reconfigurability in favor of individual autonomy. To our knowledge, only one modular system features efficient module mobility without sacrificing on self-reconfigurability [21, 91]; however, it lacks important features from the field such as inter-module communication and power sharing. This highlights the need for further modular robots that retain the features and reconfigurability of past successful systems, whilst also offering efficient single module locomotion.

This chapter presents HyMod (Figure 4.1), a self-reconfigurable modular robot that is a hybrid between mobile, chain, and lattice reconfigurable robots [7]. Inspired by systems such as PolyBot [67] and CKbot [27], HyMod features a central rotational DOF capable of moving ± 90 degrees, and is designed to form arbitrary cubic lattice structures. Two further rotational DOFs are mounted perpendicular to the central rotational joint, serving the dual purpose of emulating a spherical joint and enabling the module to drive around using a differential wheel setup. The arrangement of rotational axes shares similarities with the RobMAT [34] platform, and the use of reconfiguration joints as wheels has been explored on the iMobot [52], M^3 [92], and SMORES [21] platforms.



(a) (b) (c) (d) (e)

Figure 4.1.: HyMod: a new self-reconfigurable modular robot with three degrees of freedom (two of which form differential wheels), and four genderless connectors with single-sided disconnect. The module is shown from an (a) isometric, (b) front, and (c) side view, with its central rotational joint at zero degrees. The central joint is also shown at (d) -90, and (e) +90 degrees.

This implementation removes the need for a separate drive mechanism for locomotion, as is the case with the modules of the Symbrion / Replicator project [64].

Connections to neighboring modules are achieved using four high-speed genderless (HiGen) connectors (Chapter 3), one for each wheel and two along the central rotational axis. The use of HiGen connectors gives the module several advantages over other connection mechanisms, most notably the ability to independently disconnect from as well as produce clearance between neighboring modules. The choice of connector gave rise to the module's spherical design, which allows all three degrees of freedom to actuate simultaneously without colliding with neighboring modules.

The remainder of this chapter presents the requirements that resulted in HyMod (Section 4.2), the design and implementation of the module (Section 4.3), and experiments conducted with a single unit (Section 4.4). Finally, Section 4.5 concludes the chapter.

4.2. Requirements

By examining the range of existing self-reconfigurable modular robots, a number of beneficial features for the new modular robot emerge. On the mechanical side, the ability for the module to reside in both and chain and lattice structures, with the ability to rotate its connectors between lattice faces, is beneficial for both self-reconfiguration and collective locomotion. In addition, having connectors able to move freeing between lattice faces can increase the self-reconfiguration capabilities of the module. To have the module capable of self-assembling, a form of wheel-based locomotion would be necessary. In terms of the electronics, wired communication between modules as well as wireless to an external computer would be desirable, along with the ability for modules to share power between each other so that less active modules can support the energy demands of more active modules. These high-level goals for the module give rise to the following systematic requirements:

4.2.1. Mechanical

The module shall:

M.1.1 - Be primarily constructed using 3D printing technology. A Stratasys Mojo [113] is available in-house, capable of printing parts in ABS plastic with soluble supports

M.1.2 - Feature more than two self-actuating genderless connectors to allow for its use in both chain and lattice configurations

M.1.3 - Conform to the dimensions of one or more positions of a regular 3D lattice structure, be it cubic or hexagonal

M.1.4 - Feature at least one rotational degree of freedom that moves a connector from one face of a lattice position to another

a) The degrees of freedom shall be able to stop at any angle between these positions so as to form kinematic chain structures

M.1.5 - Be able to rotate its connectors away from occupied neighboring lattice positions without being impeded by structural geometry

M.1.6 - Feature wheel-based locomotion to allow for independent movement in a flat environment

4.2.2. Electrical

The module shall:

M.2.1 - Contain its own microcontroller that interfaces with sensors, actuators, and connectors of the module

M.2.2 - Feature a motor and driver for actuating each of the module's degrees of freedom

M.2.3 - Be able to detect:

a) The current angle of its rotational degrees of freedom

b) Its orientation with respect to gravity

c) The distance to nearby obstacles in a flat environment

M.2.4 - Be able to communicate with neighboring modules through its connectors

a) Both neighbor-to-neighbor and network-based communication shall be incorporated

M.2.5 - Be able to communicate wirelessly with an external computer

a) The chosen method shall have the option to be turned on and off by the microcontroller to save power when not required

M.2.6 - Be powered by a common bus shared between modules through their connectors

a) An on-board power supply shall draw from and contribute to the bus

b) One or more rechargeable batteries shall act as the internal power source for the power supply

c) The batteries shall be removable for quick change-over during experiments

d) The batteries shall be rechargeable from within the module

M.2.7 - Feature indicator LEDs to visually report operation state

M.2.8 - Be reprogrammable

4.2.3. Environmental

The module shall:

M.3.1 - Operate at a standard humidity for an indoor laboratory environment

M.3.2 - Operate within a temperature range of 20 to 30 degrees Celsius

4.2.4. Performance

The module shall:

M.4.1 - Be no greater than 1 kg in weight

M.4.2 - Have 3D printed parts that are no larger than the 127 x 127 x 127 mm build volume of the in-house Stratasys Mojo [113]

M.4.3 - Be able to lift at least two modules in line with at least one of its rotational degrees of freedom

M.4.4 - Be able to operate on battery for at least 30 minutes

M.4.5 - Be able to measure distances to nearby obstacles of at least 5 cm

M.4.6 - Become operational within three seconds

4.2.5. Reliability

The module shall:

M.5.1 - Be able to actuate its degrees of freedom hundreds of times without failing

M.5.2 - Be able to repeatedly move its degrees of freedom between given positions

4.3. The HyMod Unit

An objective for HyMod was to create a module to address the division between mobile and self-reconfigurable systems, by integrating an efficient locomotion method that could also have a use on modules within chain or lattice structures (e.g. as a degree of freedom in a kinematic manipulator). Although the modules of systems such as M-TRAN can move independently, they are slow and have limited control over their heading when moving. A more efficient method of locomotion is that of wheels, as these can provide a constant velocity to a robot and allow for controlled turning.

To incorporate wheels into HyMod, as specified by requirement M.1.6, the concept of a spherical joint was adopted (Figure 4.2, left). Typically modules designed to reside in a cubic lattice have a central rotational DOF that goes from -90 to +90 degrees, allowing for a free end to move between three faces of a cube, relative to a fixed end. By adding a rotational DOF to the fixed end, the free end is able to move between five faces. Additionally, by applying a rotational DOF to the free end, any item attached to it can be oriented arbitrarily. If the central rotation axis of this spherical joint is set to zero degrees (Figure 4.2, center), the remaining axes become in-line. By placing wheels on these axes a differential wheel setup is created (Figure 4.2, right), granting HyMod locomotion capabilities on par with various mobile swarm robotic systems available.

To allow for HyMod to form both chain and cubic lattice structures, four connectors are used; one in each wheel, and two along the central rotational axis. This arrangement gives rise to two options for how the three degrees of freedom of the module are connected; either the two central axis connectors reside on the same structure, or each resides in separate module halves, capable of rotating relative to each other. Although both options satisfy requirement M.1.2, the latter was chosen for HyMod as allows for the two central axis connectors to be rotated relative to each other, which could benefit self-reconfiguration.

4.3.1. Geometry Analysis

From examining the 3-DOF spherical joint (Figure 4.2, left), it is apparent that an element of symmetry exists, as swapping which end is fixed can result in the same movements, provided appropriate control remapping occurs. By discovering what these

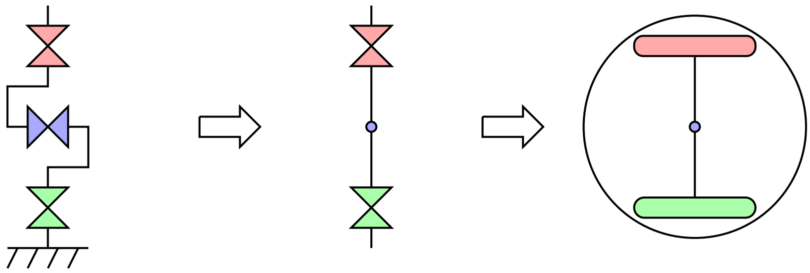


Figure 4.2.: The transition from the side view of a 3-DOF spherical joint (left) to a top view of a differential wheel setup (right), via an intermediate step where the middle DOF is locked at 0 degrees.

symmetries are, the isomorphic configurations that can be created with a given number of HyMod units can be determined, thereby reducing the search space complexity of any self-reconfiguration algorithm that may be employed on the system.

Figure 4.3 shows the eight possible orientations of a HyMod unit. The orientations are depicted on a 2D plane with the central rotational joint set to zero degrees. This can either be thought of as a top-down view of the modules resting on their wheels, or a side view with the modules anchored to a surface via their bottom connectors. The module has a rotational symmetry of two, meaning that of the eight orientations shown, only four are unique. The connector and joint mapping to go between one orientation and its symmetric version are shown in Table 4.1. As an example, to map orientation A to C, commands that would be sent to connectors 0, 1, 2 and 3, would instead need to be send to connectors 2, 3, 0 and 1. Similarly, commands to joints X and Y would instead be sent to joints Y and X, with Z remaining unchanged.

Using the knowledge of module symmetry and the four times symmetry of HiGen connectors, the number of isomorphic configurations of two modules can be determined. By applying the mapping and discarding configurations where a connector symmetry offset (e.g. 90°) is equivalent to a wheel rotation, six isomorphic configurations are produced. These can be seen in Figure 4.4. Of the six, the two configurations labelled α offer a higher number of quantized joint angle combinations, 36 (3 x 4 x 3) versus the 9 (3 x 3) of the four other configurations. This is because those two configurations contain at least one continuous rotational degree of freedom between the two modules, featuring four quantized angles versus the three of the central joint. Note that rotations of wheels not connected to another module were discounted here, as they can be cancelled out by connector symmetry. Similarly, when two wheels are connected together their rotational degrees of freedom are in-line and can therefore be considered as a single joint. Renders

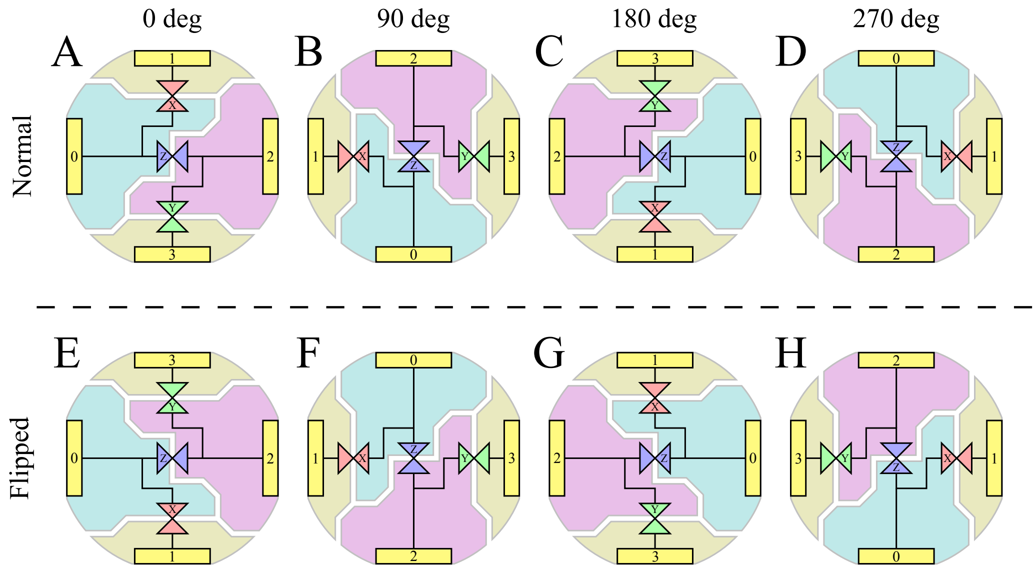


Figure 4.3.: The eight ways a HyMod unit can be oriented, as viewed on a 2D plane. Connectors are depicted using yellow rectangles, and are labelled 0 to 3. Rotational DOFs are depicted using connected triangles and are labelled X to Z. In this arrangement, connectors 1 and 3 can be rotated continuously, whereas connectors 0 and 2 can only be rotated ±90 degrees.

Table 4.1.: The connector and joint index changes when mapping one HyMod orientation to another.

Map				Connectors				Joints		
A	B	E	F	0	1	2	3	X	Y	Z
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
C	D	G	H	2	3	0	1	Y	X	Z

of the joint angle combinations for all six configurations are shown in Appendix A.

As the design of HyMod is based on a spherical joint, it only occupies a single cubic lattice position. This means that in order to self-reconfigure, either four modules are needed so that a loop can be formed, or two modules and some kind of support surface (either a custom made structure or a grid of modules). By using a support surface, and provided both modules are adjacent to it, all of the isomorphic configurations of two modules (Figure 4.4) can transform in to each other without moving between lattice positions [1].

[1]Note, if a surface is made up of passive HiGen connectors, that being connectors without the ability to

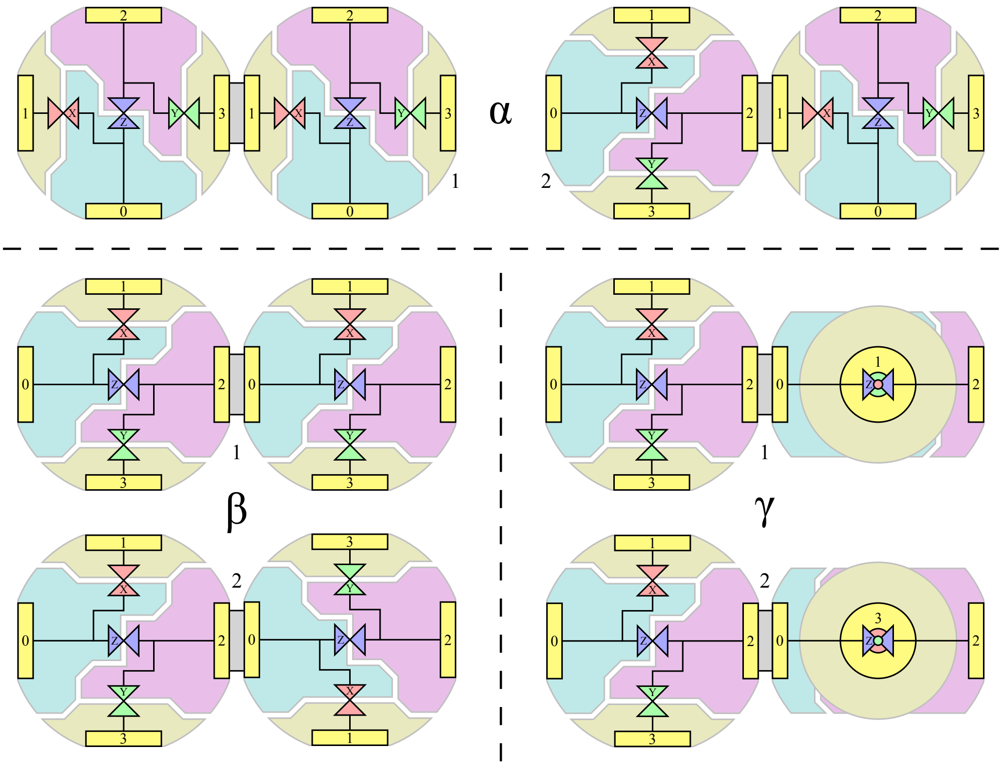


Figure 4.4.: All six of the isomorphic configurations that exist for two connected HyMod units.

If only one module is adjacent however, and the configuration is one of the four labelled β or γ, self-reconfiguration is not possible as there are no perpendicular rotational axes available to move the other module to be adjacent to the surface. This suggests that one or both of the α configurations should be considered the metamodules [115] of the HyMod system. By using these metamodules, arbitrary connected 3D structures can be formed. For example, a cube structure can be formed with $n^3/2$ metamodules, opening up the possibility for configurations of HyMod units to be constructed within the cube, with the remaining modules acting as a form of scaffold to support the construction process. Figure 4.5 shows a cube formed out of 32 HyMod unit metamodules.

actuate from their extended state, then it is not possible to self-reconfigure in to, out of, or between the two configurations labelled γ even if they are adjacent to the surface, as clearance cannot be created between the surface and the modules to allow for such a rotation.

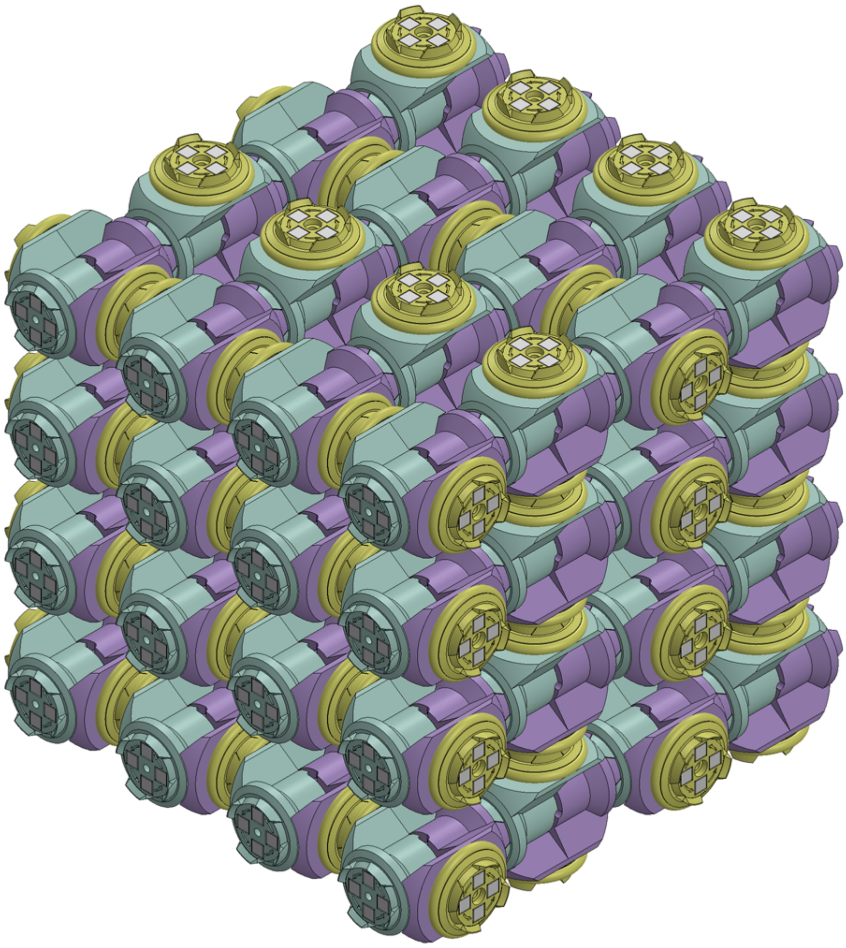


Figure 4.5.: A 4 x 4 x 4 cube formed out of 32 HyMod unit metamodules, using model files from a 3D printable scale module.

4.3.1.1. Example Configurations

To explore the possibilities of the HyMod unit's design before production, six scale models were produced. They consist of four 3D printed components each, connected together by screws, and four perpendicularly mounted permanent magnets per face. This arrangement of magnets emulates the genderless property of the HiGen connector. Examples of common modular robot configurations using the scale modules are shown in Figure 4.6. Additionally, renders of the 3D model files from the scale modules can be seen in Figure 4.5, all the figures within Section 4.3.2, and Tables A.1, A.2, and A.3 of the Appendix.

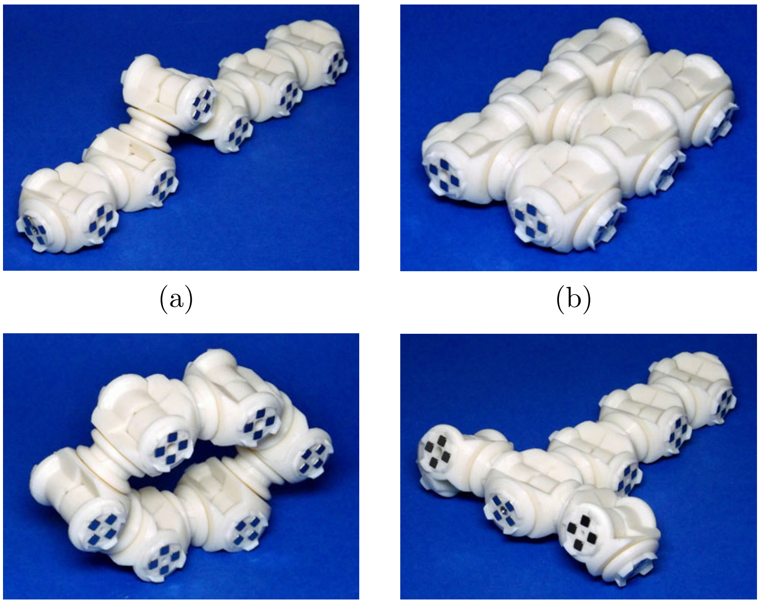


Figure 4.6.: Examples of four possible HyMod robot configurations, using scale models: (a) snake, (b) 6-wheeled vehicle, (c) rolling track, (d) crawler.

4.3.2. Self-Reconfiguration Analysis

The ability for a HyMod unit to self-reconfigure within a cubic lattice position is dependent on its orientation with respect to the lattice's coordinate system, its current connections to neighboring modules, and what other modules are available to connect to. From these starting conditions, a sequence of transitions, that being quantized joint actuations accompanied by connectivity changes, can be applied to a module to change it from one orientation to another. Note that the vacancy of adjacent lattice positions can limit the available transitions, as although a particular transition between orientations may be allowed by the module's current and available connectivity, the motion may result in its geometry colliding with an adjacent module. This limitation can be overcome by designing the module geometry to avoid such collisions, enabling free in place rotation.

By taking a HyMod unit with its central rotational joint at zero degrees and rotating it through all axes of a cubic lattice, 24 orientations of the module are produced. These shall be referred to as planar orientations. Additionally, by setting the central joint to -90 and +90 degrees and applying the same process, 48 non-planar orientations are produced, giving 72 orientations in total. Due to the symmetric nature of the module, only 36 of these orientations are unique, as control remapping allows for it to effectively reside in two orientations at once. From each of these orientations it is possible to transition to exactly six others. For example, each planar orientation has ± 90 degree non-planar orientations relative to either side of its central joint, as well as ± 90 degree rotations of the module's body relative to both of its wheels.

All 72 orientations and the transitions between them can be thought of as the vertices and edges of an undirected graph, respectively. To help depict this graph, Figure 4.7 shows the orientations grouped in to six faces of a cube. Each face features the orientations that can be reached by actuating the central joint of HyMod, with either the cyan side or purple side of the module being fixed relative to the lattice coordinate system. Transitioning between orientations via wheel rotations results in moving to orientations in the same grid position on an adjacent face of the cube. These three transition types are depicted as cyan, purple and yellow lines to match the piece of module geometry that remains stationary with respect to the cubic lattice. Alongside each transition line is an isometric cube icon that shows the maximum connectivity required for the associated joint actuation to occur, with the minimum being just one connection. This means that some transitions may first require module connectivity to be changed before they can be performed, whereas others may already match all or a subset of the connectivity. To better illustrate this concept, four example transition sequences are shown in Figure 4.8. To aid in understanding the 3D nature of the orientation cube, a printable version can be found in Appendix B.

Assuming a HyMod unit has neighboring modules that it can connect to in all six neighboring lattice positions, transitioning the unit from one planar orientation to another by ± 90 degrees around the lattice's X, Y, or Z axis requires a minimum of one joint actuation and zero connectivity changes, and a maximum of three joint actuations and four connectivity changes. This is shown in the first three examples from Figure 4.8. In the case of non-planar orientations, the minimum is the same and the maximum is only two joint actuations and three connectivity changes. This is because the axes of all three joints are perpendicular for non-planar orientations, whereas for planar orientations the two wheel axes are parallel. The maximum number of connectivity changes is dependent on whether the starting and ending connectivities of a module match that required by the starting and ending transitions, respectively. Additionally, some transitions may use the same connectivity as a previous transition, as shown in Figure 4.8(d), reducing the total count for self-reconfiguration operations that follow that path.

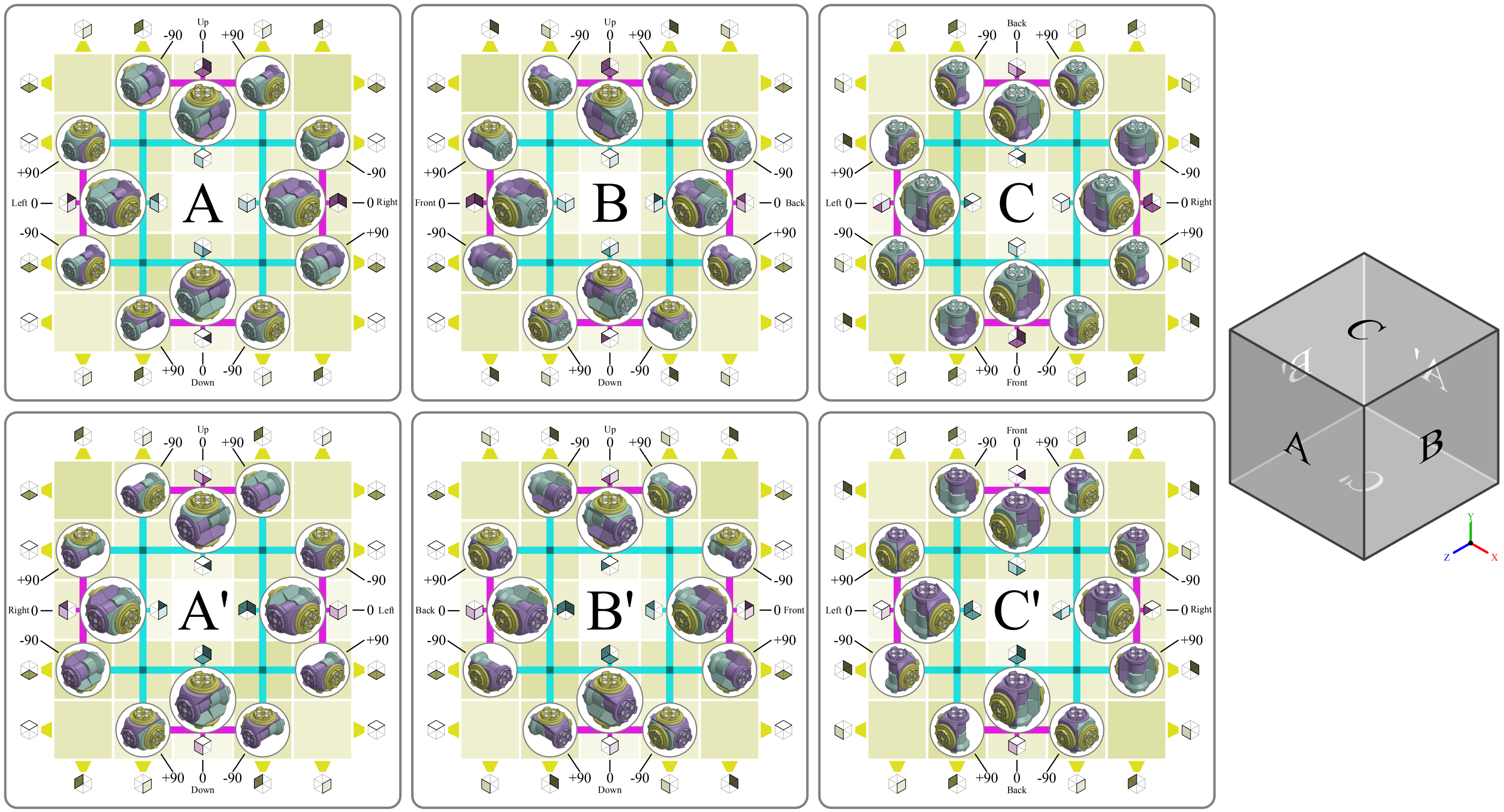


Figure 4.7.: Orientation Cube: Shows isometric renders of all 72 orientations of a HyMod unit within a single lattice position, along with the joint actuations and connectivity changes required to self-reconfigure between each of them. An isometric cube and coordinate axes are shown to help illustrate the relative placement of the six cube faces. The three faces marked with an apostrophe are the symmetric versions of the first three. Further explanation of this diagram can be found in the related section.

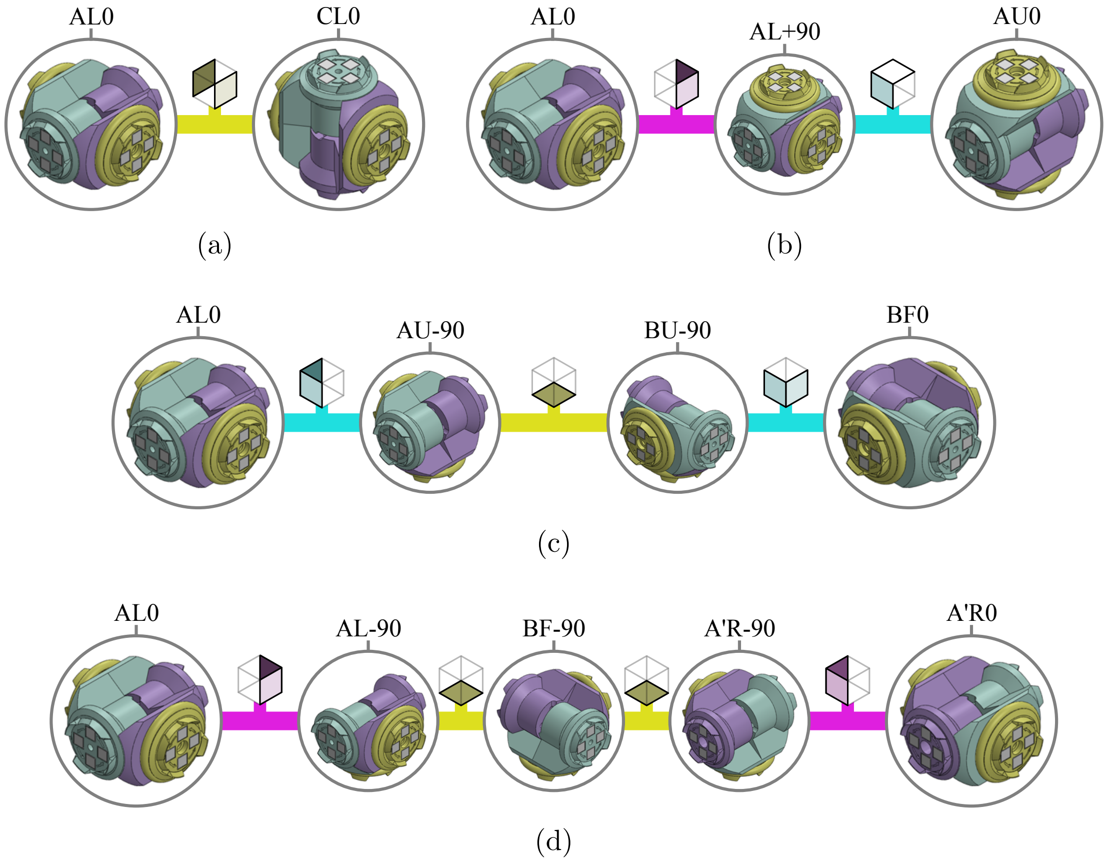


Figure 4.8.: Example transition sequences from the orientation cube, showing the joint actuations and connectivity changes required to rotate a HyMod unit 90 degrees around the (a) X axis, (b) Z axis, and (c) Y axis, as well as (d) switch the unit to its symmetric version. All four examples use the starting orientation AL0, a planar orientation on the left side of the cube's A face.

To get from an orientation to its symmetric version without control remapping, as shown in Figure 4.8(d), takes four transitions. By analysing the graph represented by the orientation cube for the shortest paths between orientations, it is discovered that four is the maximum number of transitions needed for a HyMod unit to self-reconfigure between any two orientations in a cubic lattice. This is provided there are six neigbors available to change connectivity with. The result of this analysis can be seen in Table 4.2, and the adjacency matrix depictions of the graphs used to derive them are presented in Appendix C. Observe that removing symmetric orientations from the graph does not increase the maximum number of transitions required to reach all orientations, and instead reduces the number of transitions in several cases. This highlights the benefit of the symmetric

Table 4.2.: The number of planar and non-planar orientations that are reachable from any planar or non-planar starting orientation with available connectivity on all six cubic faces, for increasing numbers of transitions. The results for the full 72 orientation graph are shown, as well as those for the graph with the 36 symmetric orientations removed. P and N denote planar and non-planar, respectively. Revisited orientations are not counted.

	Planar		Non-Planar	
Transitions	Symmetries	No Symmetries	Symmetries	No Symmetries
0	1 (1P)	1 (1P)	1 (1N)	1 (1N)
1	6 (2P + 4N)	6 (2P + 4N)	6 (2P + 4N)	6 (2P + 4N)
2	19 (3P + 16N)	15 (3P + 12N)	23 (8P + 15N)	15 (6P + 9N)
3	38 (10P + 28N)	13 (5P + 8N)	32 (14P + 18N)	13 (4P + 9N)
4	8 (8P)	1 (1P)	10 (10N)	1 (1N)
Total	72 (24P + 48N)	36 (12P + 24N)	72 (24P + 48N)	36 (12P + 24N)

nature of the HyMod design for self-reconfiguration. Note that the difference between planar and non-planar starting orientations on both graphs can be attributed to four of the transitions from the first non-planar orientation resulting in ±90 degree rotations of the module. In contrast, only two transitions from the first planar orientation result in ±90 degree rotations of the module.

4.3.2.1. Example Sequences

To demonstrate the self-reconfiguration capabilities of the HyMod design before production, two example sequences are presented. The first is a quadruped formation of nine modules transforming in to a line formation, and the second is the 4 x 4 x 4 cube of modules from Figure 4.5 relocating a single module from one of its corners to another. The sequence of self-reconfiguration steps for the quadruped and the cube can be seen in Figures 4.9 and 4.10, respectively. Additionally, the list of orientation changes each step performed for the two sequences are shown in Tables 4.3 and 4.4. Note that with the cube example, free in place rotation of HyMod units is required to achieve the sequence of self-reconfiguration steps shown.

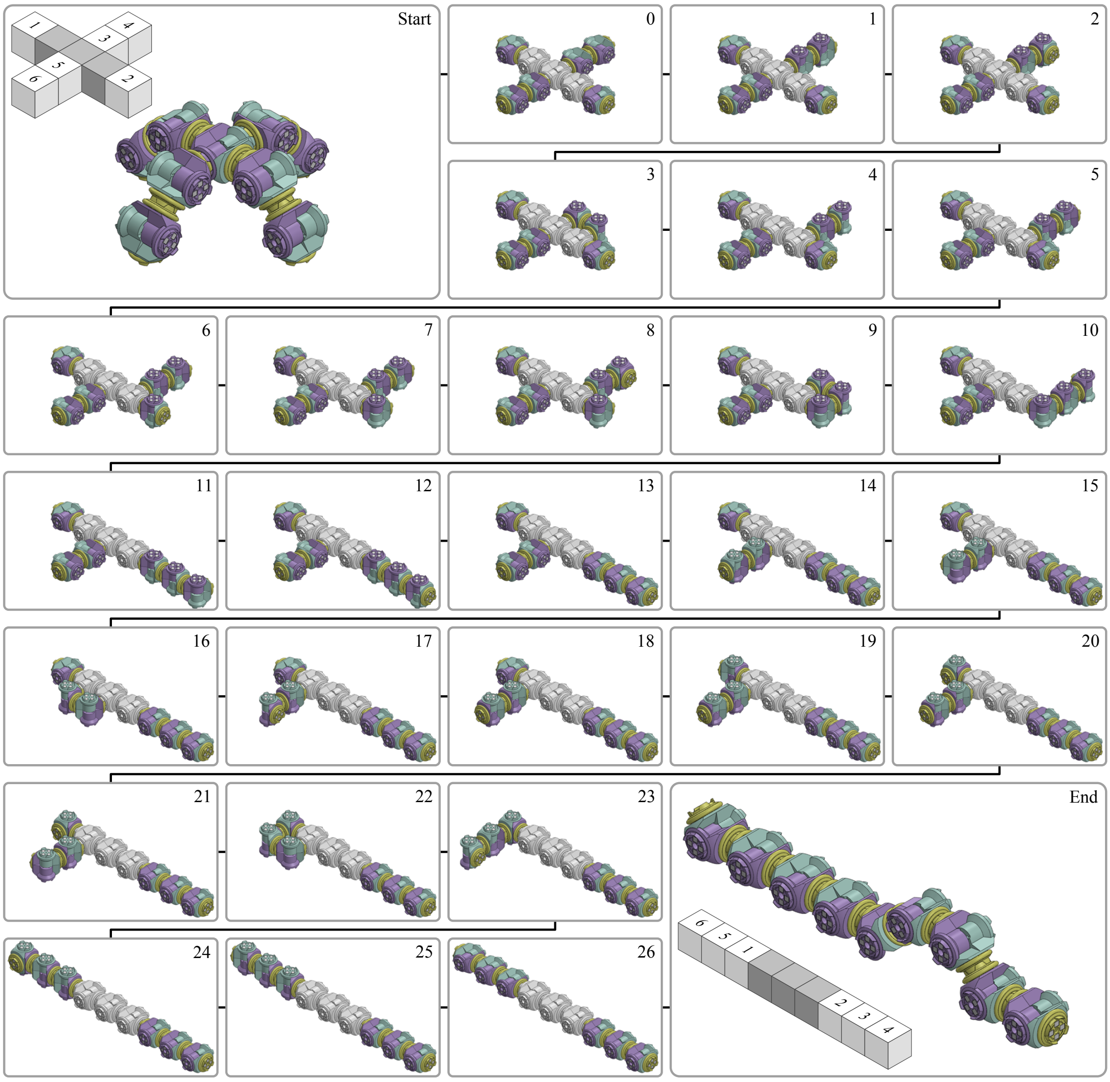


Figure 4.9.: A sequence of self-reconfiguration steps that can be performed to transform a quadruped formation of nine HyMod units in to a line formation. Modules coloured in grey are not modified during the sequence, and as such are not given identifiers.

Table 4.3.: The orientation changes of each of the modules in the quadruped to line self-reconfiguration example. Orientations in blue italics are those that are caused by a module being moved by a neighbor, and | denotes sequence steps where a module's orientation has not changed from a previous step.

	Module					
Step	1	2	3	4	5	6
0	A'L0	A'R0	B'B0	B'B0	B'F0	B'F0
1	|	|	C'B0	*C'B0*	|	|
2	|	|	|	C'B-90	|	|
3	|	|	C'B-90	*C'R-90*	|	|
4	|	|	*C'R-90*	C'F0	|	|
5	|	|	C'F0	|	|	|
6	|	C'R0	|	|	|	|
7	|	C'R+90	|	|	|	|
8	|	|	C'L+90	|	|	|
9	|	|	*C'B+90*	C'L+90	|	|
10	|	|	C'B0	*C'B+90*	|	|
11	|	C'R0	*C'R0*	*C'R+90*	|	|
12	|	|	|	C'R0	|	|
13	|	A'R0	*A'R0*	*A'R0*	|	|
14	|	|	|	|	CF0	*CF0*
15	|	|	|	|	|	CF+90
16	|	|	|	|	CF+90	*CL+90*
17	|	|	|	|	*CL+90*	CB0
18	|	|	|	|	CB0	|
19	CL0	|	|	|	|	|
20	CL-90	|	|	|	|	|
21	|	|	|	|	|	CR-90
22	|	|	|	|	CR-90	*CF-90*
23	|	|	|	|	*CF-90*	CF0
24	CL0	|	|	|	*CL-90*	*CL0*
25	|	|	|	|	CL0	|
26	A'L0	|	|	|	*A'L0*	*A'L0*

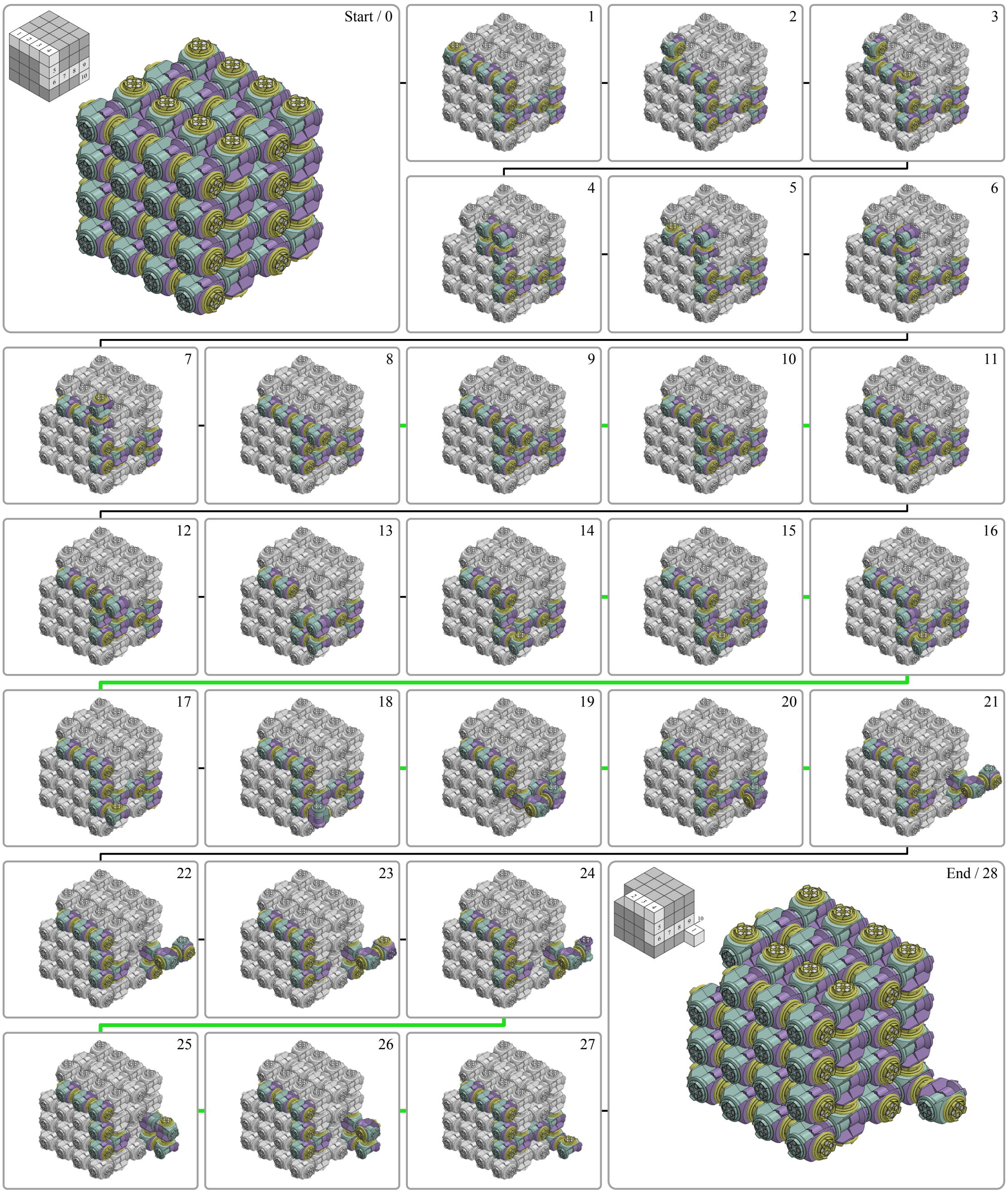


Figure 4.10.: A sequence of self-reconfiguration steps that can be performed to relocate a single module from one corner of a 4 x 4 x 4 cube to another. Modules coloured in grey are not modified during the sequence, and as such are not given identifiers. Note that transition lines shown in green indicate transitions that can only be achieved using free in place rotation.

Table 4.4.: The orientation changes of each of the modules in the 4 x 4 x 4 cube relocation example. Orientations in blue italics are those that are caused by a module being moved by a neighbor, and | denotes sequence steps where a module's orientation has not changed from a previous step.

	Module									
Step	1	2	3	4	5	6	7	8	9	10
0	AL0	AL0	AL0	AL0	AL0	AL0	AU0	AL0	AU0	AU0
1	AL+90	|	|	|	|	|	|	|	|	|
2	*AU+90*	AL+90	|	|	|	|	|	|	|	|
3	|	|	|	AD+90	|	|	|	|	|	|
4	*AR+90*	*AU+90*	AL+90	|	|	|	|	|	|	|
5	|	AL+90	AL0	|	|	|	|	|	|	|
6	|	AL0	|	|	|	|	|	|	|	|
7	AD0	|	|	|	|	|	|	|	|	|
8	*AL0*	|	|	AL0	|	|	|	|	|	|
9	|	|	|	AL-90	|	|	|	|	|	|
10	|	|	|	|	AL+90	|	|	|	|	|
11	|	|	|	|	AU0	|	|	|	|	|
12	AU-90	|	|	|	|	|	|	|	|	|
13	*AR-90*	|	|	*AU-90*	AU+90	|	|	|	|	|
14	|	|	|	*AL-90*	AU0	|	|	|	|	|
15	|	|	|	AL0	|	|	|	|	|	|
16	|	|	|	|	AU-90	|	|	|	|	|
17	|	|	|	|	AL0	|	|	|	|	|
18	CR-90	|	|	|	|	|	|	|	|	|
19	*CB-90*	|	|	|	|	*BF0*	BU0	|	|	|
20	|	|	|	|	|	*AL0*	AU0	|	|	|
21	*CL-90*	|	|	|	|	|	|	*BF0*	BU0	|
22	|	|	|	|	|	|	|	|	|	AL+90
23	B'D-90	|	|	|	|	|	|	|	|	|
24	C'R-90	|	|	|	|	|	|	|	|	|
25	*AR-90*	|	|	|	|	|	|	*BU0*	BU+90	|
26	|	|	|	|	|	|	|	*BF0*	BU0	|
27	|	|	|	|	|	|	|	*AL0*	AU0	|
28	AR0	|	|	|	|	|	|	|	|	|

4.3.3. Clearance Considerations

To allow for a 3-DOF spherical joint module like HyMod to freely rotate in place within a cubic lattice position, and satisfy requirement M.1.5 in the process, sufficient clearance needs to exist such that the module's furthest geometric point from the 3-DOF rotation center can move freely without colliding with neighboring modules. The space for collision-free rotation can be imagined as a sphere around the module. The design of spherical joint modules can be adapted to ensure that the sphere of one module does not intersect with the sphere of a neighbor; however, as most connectors require a surface-to-surface connection (see Section 2.3), a mechanism such as HiGen is needed that is capable of retracting in to its housing when disconnecting, thus staying within the sphere when unused.

The distance a connector's surface needs to be inset from the sphere of a spherical joint module is dependent on the size of the surface and the minimum sphere size, as shown in Figure 4.11. By applying Pythagoras' Theorem, the relation between these parameters can be calculated using the following equations,

$$M_r^2 = C_r^2 + (M_r - I_s)^2, \tag{4.1}$$

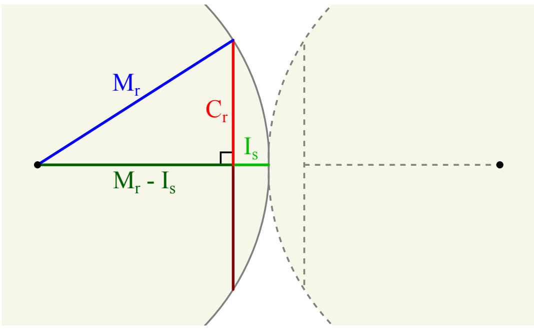
$$C_r = \sqrt{M_r^2 - (M_r - I_s)^2}, \tag{4.2}$$

$$I_s = M_r - \sqrt{M_r^2 - C_r^2}, \tag{4.3}$$

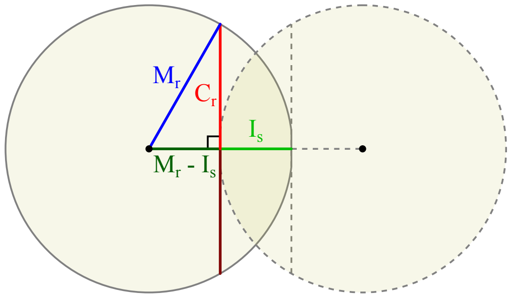
$$M_r = \frac{C_r^2 + I_s^2}{2I_s}, \tag{4.4}$$

where M_r is the radius of the module sphere, C_r is the radius of the connector surface, and I_s is the inset amount.

There are two options for applying the clearance equations to a module; either the module can be designed so that neighboring module spheres do not intersect (Figure 4.11(a)), or by using the knowledge that connectors are inset, the spheres can be made to



(a)



(b)

Figure 4.11.: Diagrams showing the relation between module sphere radius M_r, connector surface radius C_r, and inset amount I_s, for the (a) non-intersecting and (b) intersecting clearance options.

intersect by the inset amount (Figure 4.11(b)). Using the non-intersecting option allows two neighboring modules to rotate simultaneously without colliding with each other, whereas the intersecting option allows for smaller module sizes for a given connector size (by doubling the value supplied to I_s in equation 4.4), at the cost of only one neighboring module being able to rotate at a time. Applying the two clearance options to HyMod, by setting the connector radius to 35.5 mm (half the diameter of HiGen) and the inset amount to 6 mm (half HiGen's actuation distance), results in a non-intersecting module radius of 108.0 mm and an intersecting module radius of 58.5 mm. Due to the smaller value, the intersecting option was chosen as the minimum module size for the development of HyMod.

4.3.4. Hardware Details

The module is built from two mirrored halves, forming a rotational hinge joint. This arrangement of identical halves is common with several modular robots, such as ATRON [16], Molecubes [69], UBot [114], and CoSMO [23]. Each half consists of a chassis housing two HiGen connectors; one parallel to, and the other perpendicular to the hinge axis. The parallel connector is fixed to the chassis whereas the perpendicular connector has a rotational degree of freedom through its center, forming a wheel. This gives a total of four connectors and two wheels per module.

HiGen connectors (described in more detail in Chapter 3) operate by using a central drive motor to translate and rotate four hooks. These hooks latch on to hooks of an opposing connector, creating a genderless connection that allows for single-sided disconnect. As part of this latching process, electrical connections are made, allowing for communication and power transfer across the connectors.

Each HyMod unit consists of sixteen custom ABS plastic components (excluding the four connectors) created using 3D printing technology, fifteen custom circuit boards, two slip rings, two battery packs, and several off-the-shelf items. Four DC geared motors are used to drive the three degrees of freedom of the module (two paired together for the hinge joint), each with a ratio of 154:1 and a quoted torque of 847 mNm at 6 V. An additional 5:1 gear ratio is applied on top of each motor gearbox, increasing the torque of the rotational joints and allowing the motors to be offset from each drive axis. This setup is what facilitates the use of two motors to drive the hinge joint, enabling all motors to be identical whilst allowing the hinge joint to offer effectively twice the torque of the other degrees of freedom. This also simplifies their control because the same driver electronics can be used for each motor. The housings of the four connectors are modified from the original design to allow for extra mounting points for the wheel hubs and the addition of infrared sensors for distance sensing (requirement M.2.3c). Internal sensing, as specified by requirements M.2.3a and M.2.3b, is achieved using a potentiometer, two optical encoder setups, and an Inertial Measurement Unit (IMU). To allow for continuous rotation of the wheels whilst passing power and communication to their connectors, slip ring components are used. This is a solution adopted by past systems [16, 19].

The module weights 810 g, falling below that specified by requirement M.4.1, and measures 128 mm x 128 mm x 94 mm when its hinge is at zero degrees. The size is governed by the part dimension restrictions imposed by requirement M.4.2, the dimensions of the HiGen connector, the height of the slip rings, and the chosen wheel diameter of 94 mm. This wheel diameter gives the module a 4 mm ground clearance when oriented for driving. The separation between modules in a cubic lattice is 140 mm due to the connectors extending out of their housings by 12 mm during connection. To take advantage of this ability the module is designed to fit within a spherical volume, allowing for rotation around three axes without risk of colliding with neighboring lattice modules (Figure 3.4). As such the module shares visual similarity with the Roombots [19] platform, which uses its spherical design to enable the wheel-based locomotion of modules, rather than to provide clearance for self-reconfiguration. Figure 4.12 shows renders of the three main sections that form a complete HyMod unit (Figure 4.13). Additionally, a breakdown of the main HyMod unit properties is shown in Table 4.5.

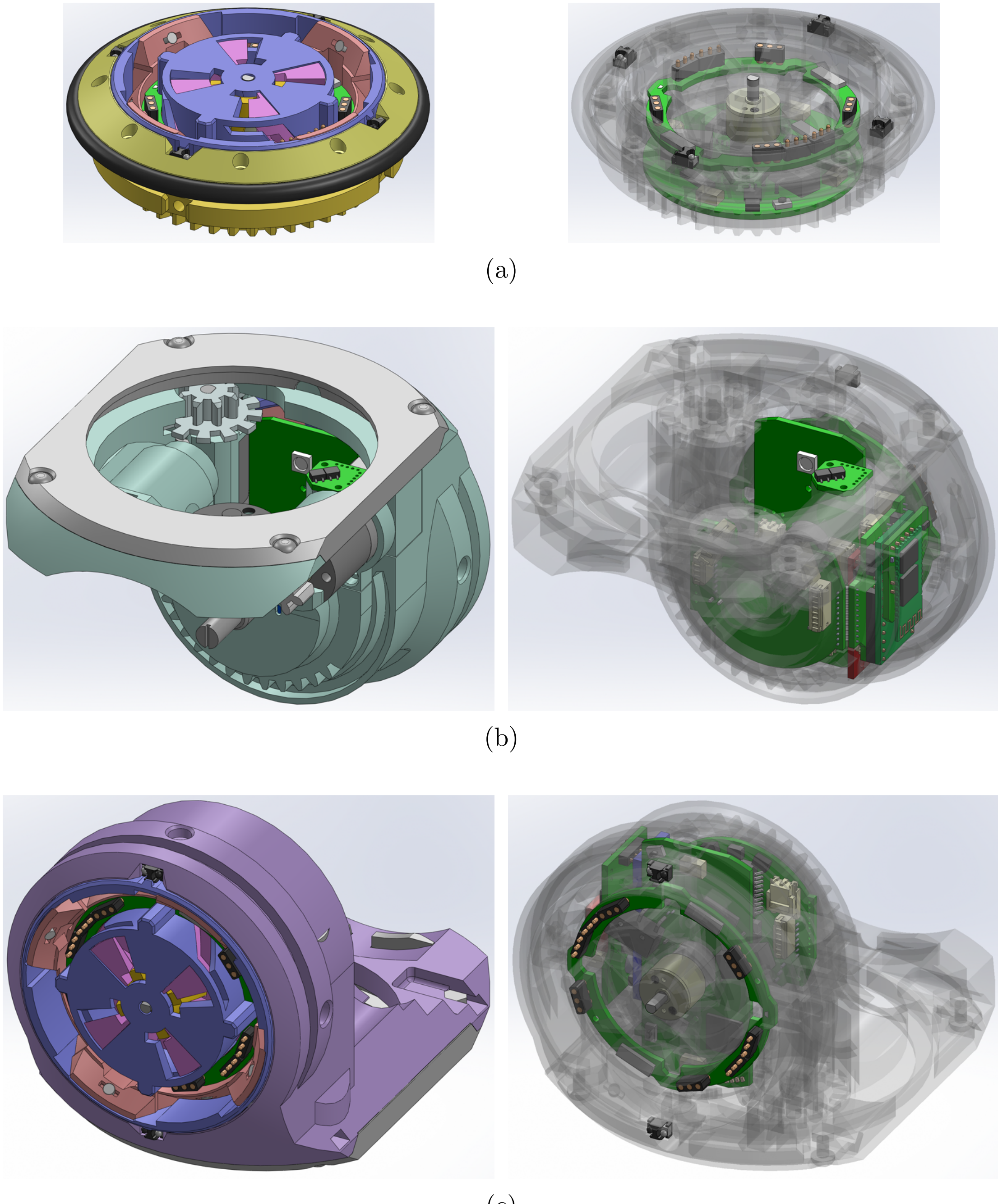
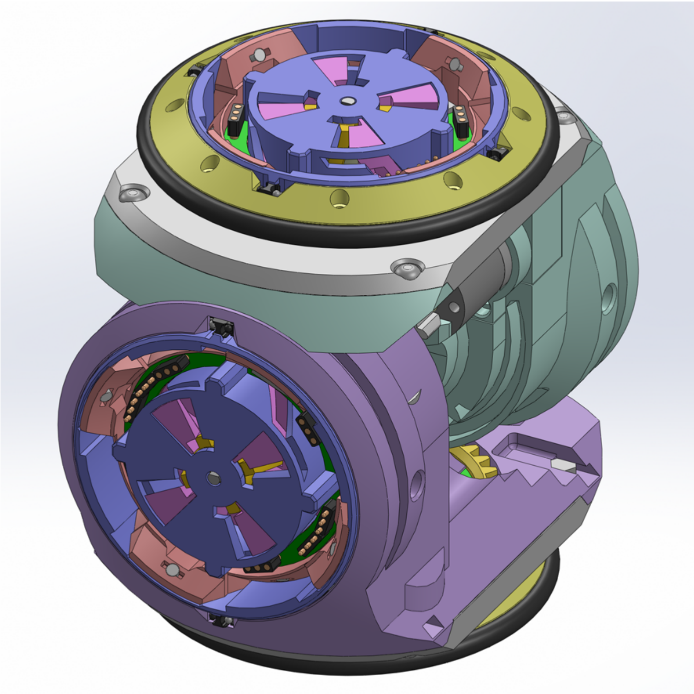
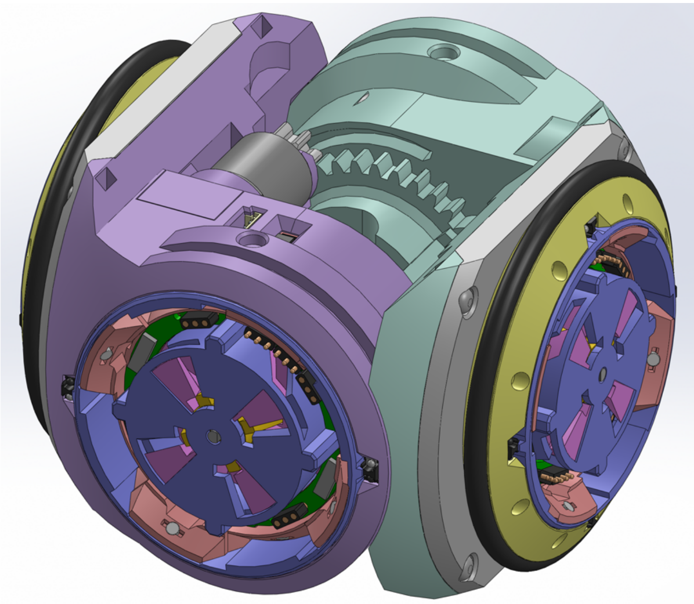


Figure 4.12.: 3D renders of the three main components of a HyMod unit. The external structure (left) and internal electronics (right) of the unit's (a) wheel, (b) processing half and (c) power half, are shown.



(a)



(b)

Figure 4.13.: 3D renders of an assembled HyMod unit, oriented (a) vertically and (b) horizontally.

Table 4.5.: Properties of a HyMod unit

Property	Value
Size	128 x 128 x 94 mm
Lattice spacing	140 mm
Ground clearance	4 mm
Weight	810 g
Controllers	1x *PJRC* Teensy 3.2
	4x *Atmel* ATmega324P (*HiGen controller*)
Communication	1x EGBT-046S Bluetooth modem
	1x *NXP* fault-tolerant CAN transceiver
Sensors	1x *Sparkfun* 9 DOF sensor stick IMU (accelerometer, gyro, magnetometer)
	12x *Vishay* reflective optical sensor (infrared proximity)
Motors	4x *Pololu* 154:1 metal gearmotor
	4x *Solarbotics* 298:1 mini metal sealed gear motor
Power supply	1x *Pololu* step-up voltage regulator (set to 9 V)
Batteries	2x *Turnigy* 3.7 V, 750 mAh round li-po cells (total 7.4 V, 750 mAh)

4.3.4.1. Electronics

HyMod contains 15 custom circuit boards: 1x *processing board*, 1x *Bluetooth board*, 1x *power board*, 4x *HiGen controller*, 2x *motor driver*, 2x *encoder board*, and 4x *contact ring*. The arrangement of boards is shown in Figure 4.14(a).

The main microcontroller for each HyMod unit is a Teensy 3.2, a 32-bit ARM Cortex-M4 based development board running at 96 MHz. This board has built-in USB for communication and programming (requirement M.2.8), a Controller Area Network (CAN) controller, and can interface with the popular Arduino development environment. The Teensy is sandwiched between the *Bluetooth board* and *processing board*; the former acts as an adapter to an off-the-shelf modem for wireless communication to an external computer (requirement M.2.5), and the latter houses additional CAN components and connects to an off-the-shelf Inertial Measurement Unit. Unfortunately, the Bluetooth

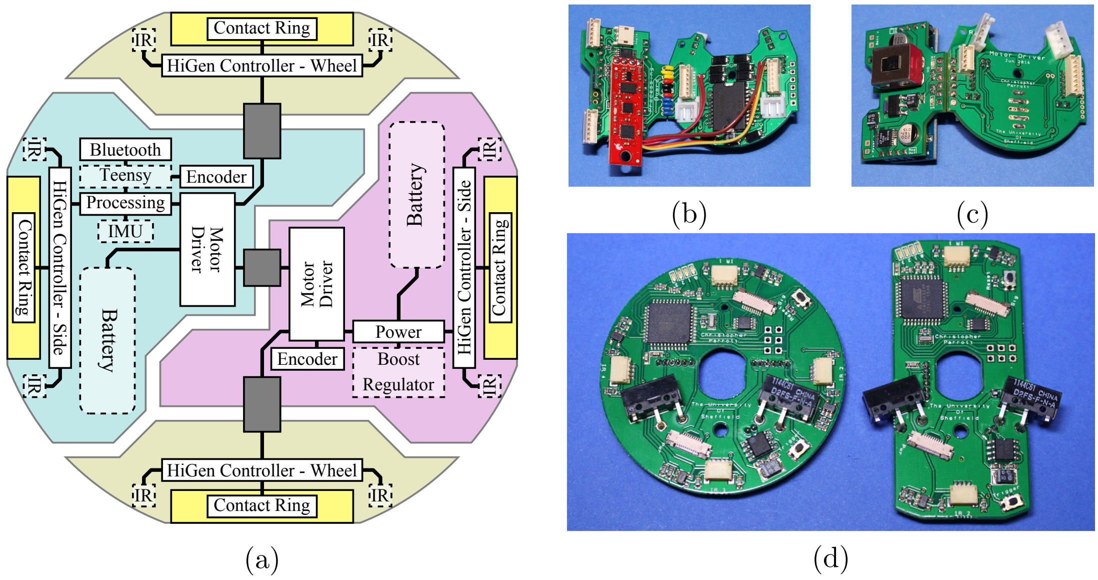


Figure 4.14.: (a) Block diagram showing how the circuit boards and other components within a HyMod unit connect together. White blocks are the custom boards created for this project. Assembled (b) *processing*, (c) *power* and (d) *HiGen controller* boards are also shown.

modem chosen cannot natively be enabled and disabled on-demand, as specified by requirement M.2.5a, so a small MOSFET was added to the *Bluetooth board* to allow power to the modem to be toggled. Figure 4.14(b) shows the assembled board stack.

Each HyMod unit is powered by two 750 mAh lithium polymer battery packs, as per requirement M.2.6b. The packs reside in each half of the module and are connected in series to give 7.4 V. The *power board* (Figure 4.14(c)) takes this voltage and, via a boost regulator, produces a 9 V output. This output is used to power the two *motor driver* boards, which each drive two joint motors. Additionally, to enable power sharing between modules and satisfy requirement M.2.6a, the *power board* passes the 9 V output through an ideal diode to create a power bus. The diode prevents the current of one power supply from feeding back in to another and potentially causing damage. The power bus is then used to produce a 5 V supply for the rest of the electronics with a module. To facilitate the repeated operation of HyMod units, their batteries can be exchanged (requirement M.2.6c) by detaching their wheels from each side and pulling each pack out, allowing for new packs to be inserted. Additionally, the batteries can be charged in place (requirement M.2.6d) by connecting a charging unit to a dedicated port on the side of each unit.

The connectors in HyMod units are controlled using custom *HiGen controller* boards. These boards feature an ATmega324P, a motor driver, two contact switches, an analogue switch, *contact ring* connections, a programming header, a manual trigger, and indicator LEDs. These features satisfy connector requirements C.2.1, C.2.2, C.2.5, C.2.7 and C.2.8. The use of a separate microcontroller allows for each connector to be treated as a device on an internal communication network. Additionally, it reduces the number of connections that need to be passed through the slip rings. There are two versions of the *HiGen controller* board in each module (Figure 4.14(d)), one for the wheel connectors and one for the side connectors. Both boards perform the same basic functions (e.g. connector actuation, infrared proximity sensing) but differ in geometry and specialized features. For instance the wheel *HiGen controller* has a grey code disc etched into it for absolute positioning of the wheel, whereas the side controller has a RGB LED for state indication and general debugging of a module, as per requirement M.2.7.

4.3.4.2. Communication

Modular systems can be thought of as computer networks, where each module acts as a node, able to communicate with other nodes. There are two main ways this can be achieved, referred to as local and global communication [116]. Local communication allows each module to communicate with its immediate neighbors, but requires that messages be relayed in order to reach modules other than direct neighbors. Global communication allows each module to send messages directly to any other module on the same network, but the identifier of the recipient must be known in advance. Due to the different use cases of local and global communication, both are implemented by HyMod, fulfilling requirements M.2.4 and M.2.4a. In addition, each unit features an internal I^2C network to communicate between components, with the Teensy acting as the master.

Local communication between two HyMod units is achieved using a serial link. Messages sent from one module to another are first sent from the Teensy over I^2C to the *HiGen controller* in question. This controller buffers the message and sends it over the serial link to the neighboring module's *HiGen controller*, which stores the message until the neighboring Teensy is ready to collect it.

Global communication between HyMod units is achieved using CAN. CAN allows for multiple connected nodes to communicate with each other by broadcasting messages on

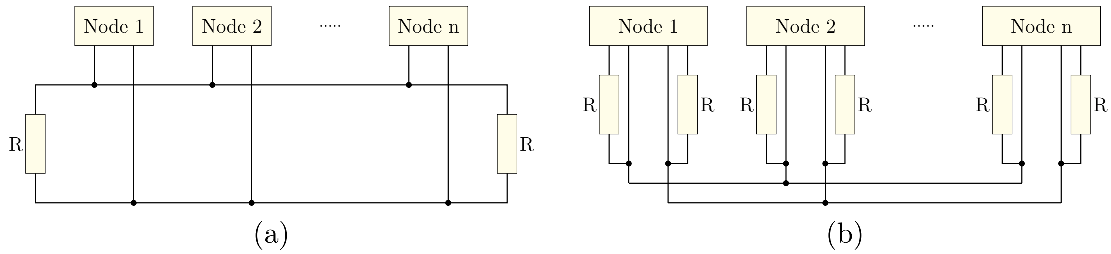


Figure 4.15.: The placement of the termination resistors, R, for the (a) normal and (b) fault-tolerant Controller Area Network implementations.

a common bus. The messages are picked up by all other networked nodes, which can then act upon the data based on an identifier. By default CAN is designed for fixed networks where there is a single line with termination resistors at the ends. Because HyMod units are self-reconfigurable, fault-tolerant CAN was used, as this places the termination resistors at each node instead. A comparison of these two implementations is shown in Figure 4.15. By using digital potentiometers along with FT CAN, the network resistance can be dynamically adjusted based on the number of nodes, maintaining a stable network. Additionally, to avoid looping CAN networks that get created during self-reconfiguration, HyMod employs analogue switches at its connectors to break the network. The use of these switches also allows for hybrid networks to exist [116], whereby the global network is divided in to smaller sub-networks for task processing, with local communication being used to bridge sub-networks when necessary. Figure 4.16 shows both the power and communication networks produced between two HyMod units.

4.4. Experiments

To verify the capabilities of HyMod, a single unit was used. Three main experiments were performed using the unit, examining driving speed, lifting capability and connector actuation. For the purpose of these experiments the unit was tethered to a bench power supply set to 8.4 V (replicating the maximum battery voltage).

The driving speed of HyMod was determined by placing the robot on the ground and timing how long it took for it to travel 2 m in a straight line. The result of this is that the module has a driving speed of $0.1\,\mathrm{m\,s^{-1}}$. The experimental setup and snapshots of the driving experiment can be seen in Figure 4.17.

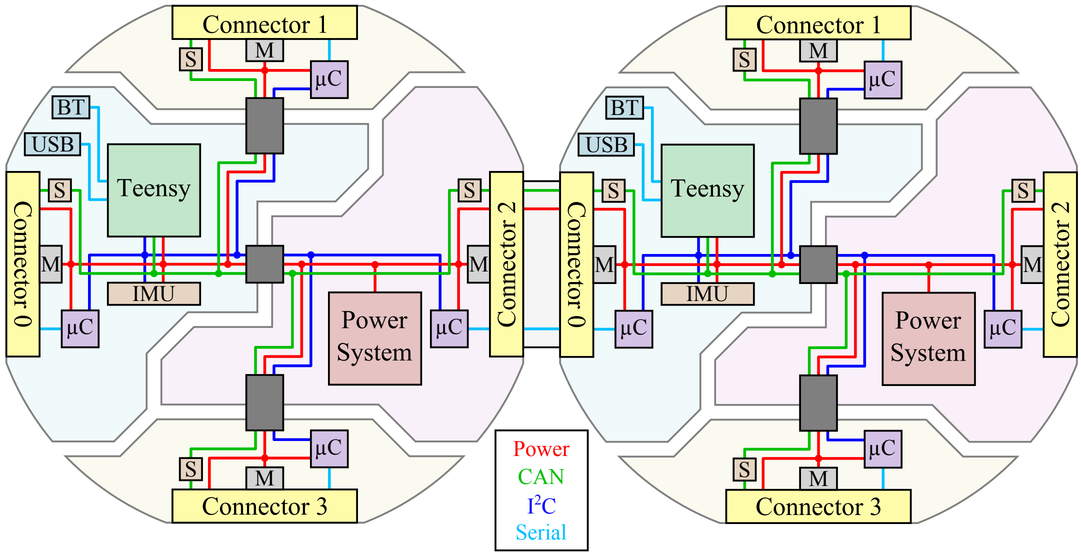
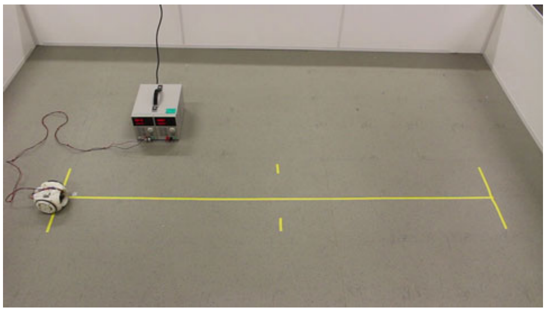


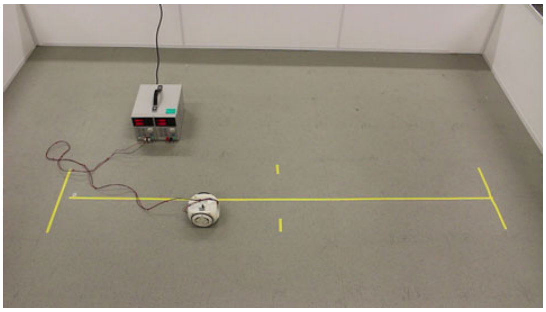
Figure 4.16.: The power and communication network formed between two HyMod units. BT, μC, M, and S denote Bluetooth, microcontrollers, connector motors, and bus switches, respectively.

The lifting capability of HyMod was tested using a 3D printed variable mass holder that attaches via a HiGen connector. The holder weighs 520 g, and supports up to 1000 g (in 100 g increments) of additional weight. The distance from the center of the HyMod unit to the center of mass of the holder is 280 mm (two lattice spacings). Lifting tests were conducted by clamping the HyMod unit to a table and having its hinge joint rotate between -90 and +90 degrees (decelerating on the downward arc). The unit was tested lifting masses up to 1120 g, which is equivalent to lifting 1.8 modules in-line. Greater masses than 1120 g were attempted, but resulted in the failure of the 3D printed gears on the hinge joint's motors, followed by the docking hooks on the HiGen connectors themselves. If these components were constructed with stronger materials, the stated torque value of the motors suggests that higher lifting capacities would be achievable. The experimental setup and snapshots of the lifting experiment can be seen in Figure 4.18.

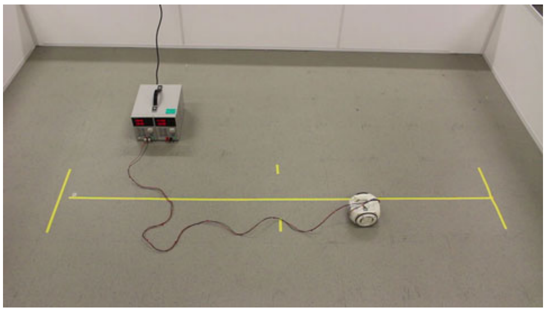
A final test was performed with HyMod, verifying that the two *HiGen controller* boards were able to operate the connectors as intended. Each connector was programmed to drive their motors between retracted and extended states every 2 s. The result was that both controller boards were able to successfully actuate the connectors. Further experiments involving HiGen can be found in Section 3.4.



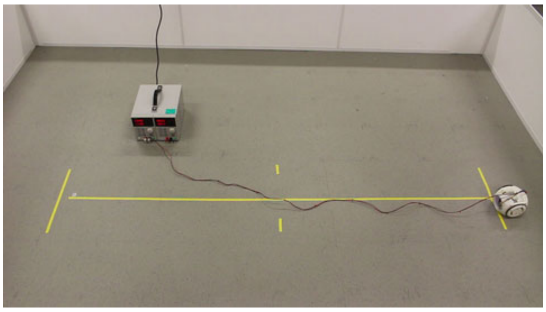
(a) 0 s



(b) 7 s



(c) 14 s



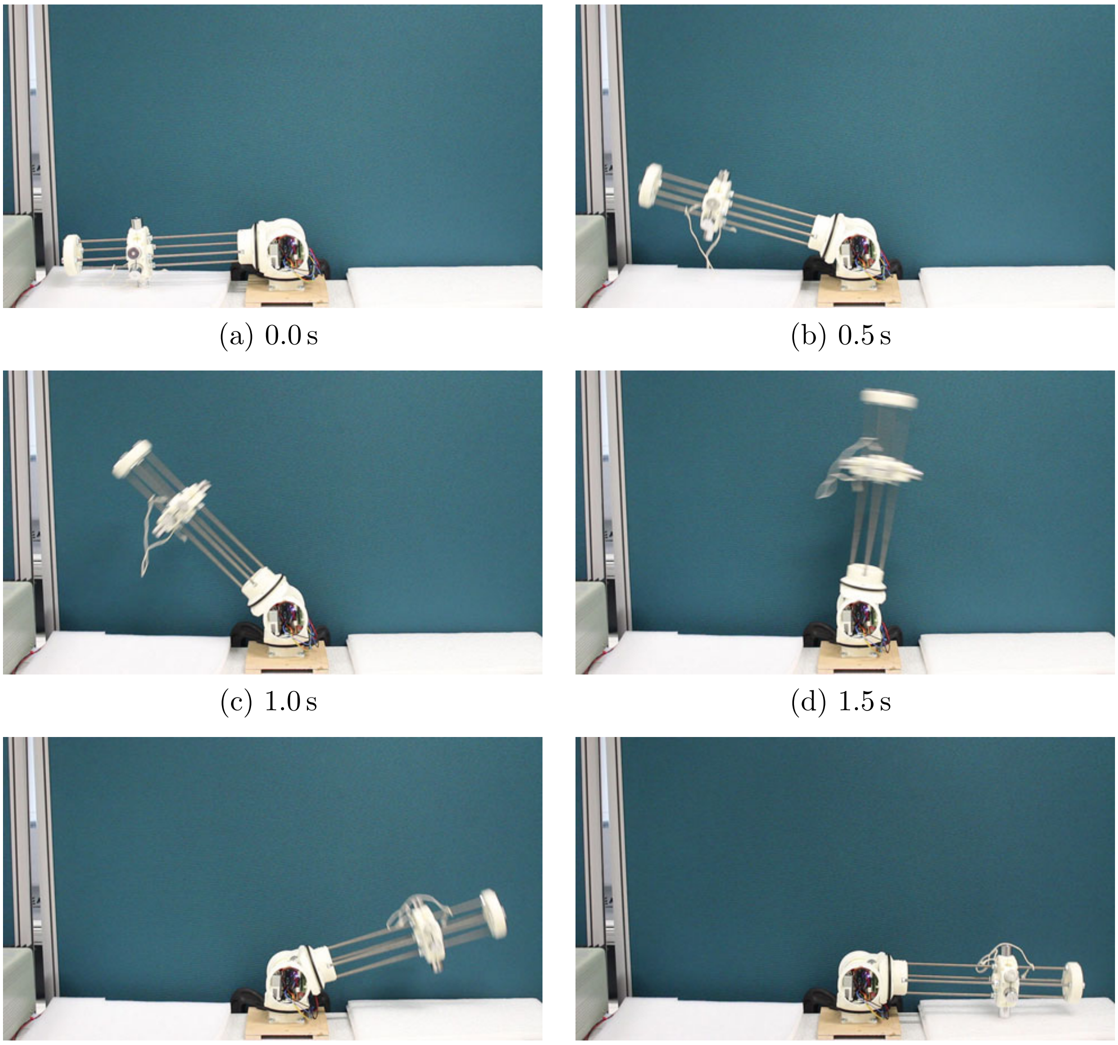
(d) 21 s

Figure 4.17.: Snapshots of a tethered HyMod unit driving a distance of 2 m.

4.5. Summary

This chapter presented HyMod, a new robotic module that is a hybrid between mobile, chain and lattice reconfigurable robots. This is achieved using a novel arrangement of rotational degrees of freedom that serve the dual purpose of emulating a spherical joint and enabling independent module mobility using differential wheels. This contrasts with previous hybrid modular robot implementations that feature separate drive mechanisms for motion, which result in increased module weight and complexity. Four HiGen connectors are integrated in to HyMod to enable the formation of arbitrary cubic lattice structures and, because of their ability to retract, allow the module to freely rotate in place. HyMod is the first module, to our knowledge, that combines independent mobility and free in place rotation.

An analysis of the HyMod design was performed, identifying the module's symmetry and motion capabilities, as well as its ability to self-reconfigure in general and between two example configurations. This analysis highlights the advantages of HyMod's joint and connector arrangement. In addition, the considerations for free in place rotation were



(a) 0.0 s (b) 0.5 s (c) 1.0 s (d) 1.5 s (e) 2.0 s (f) 2.5 s

Figure 4.18.: Snapshots of a tethered HyMod unit lifting a mass of 1120 g. A connector assembly composed of a passive HiGen connector and sheets of Medium-Density Fiberboard (MDF) is clamped to the table to fix the unit in place. Foam padding is positioned either side of the unit to cushion the variable mass holder upon reaching the end of the experiment's rotation arc.

addressed, identifying two clearance options for the design of spherical joint modules.

Details of HyMod's hardware and electronics were given, covering its actuation, sensing, communication, and power system. The module is able to communicate locally with neighboring modules via serial links, communicate globally with specific modules through a common Controller Area Network bus, and wirelessly to an external computer via Bluetooth. The CAN bus implementation on HyMod has the novel ability to dynamically adjust its resistance to account for the number of connected modules, as well as divide the bus into smaller sections to avoid loops and allow for focused communication between modules, enabling hybrid communication. The module also features power sharing, allowing one module to power the electronics of others.

Experiments were conducted examining the movement and lifting capabilities of a single HyMod unit, with it being able to drive at $0.1\,\mathrm{m\,s^{-1}}$ and lift 1120 g at a distance at of 280 mm with its central joint.

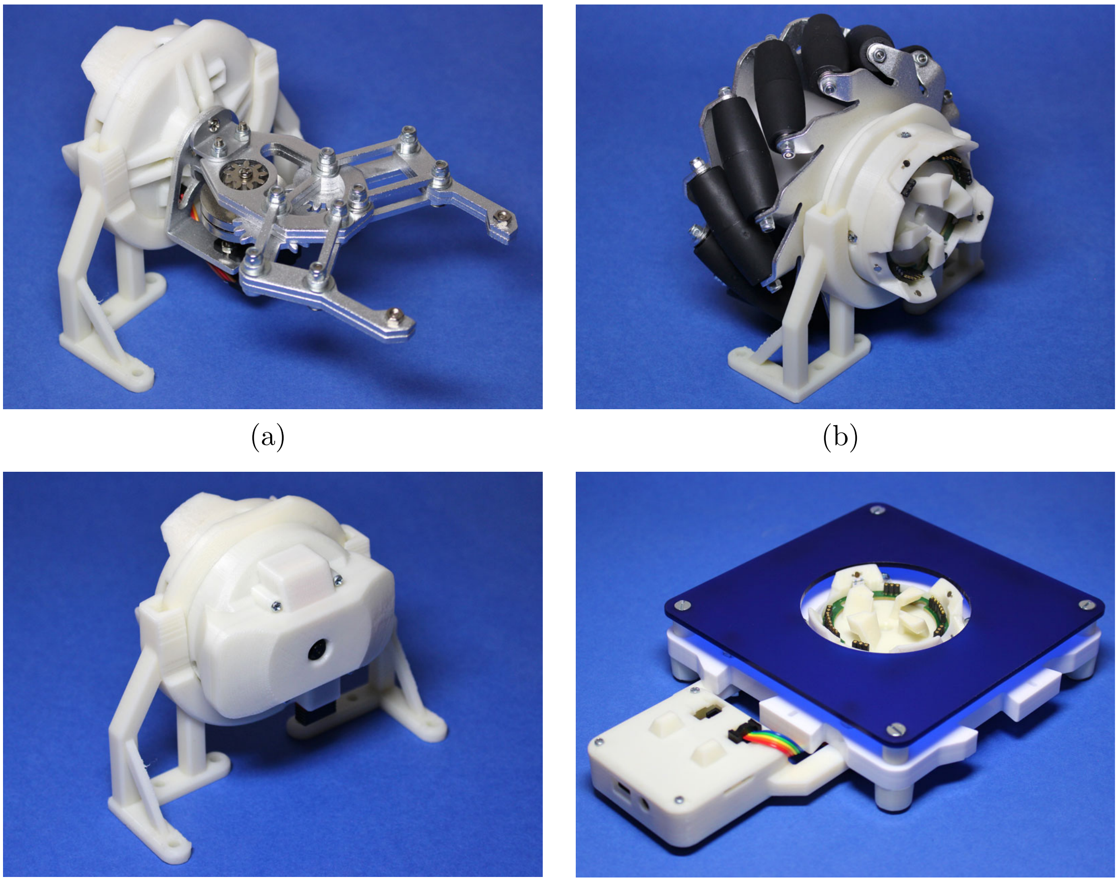
5. Tools and Extensions

5.1. Introduction

Unlike bespoke robotic systems made for specific tasks, modular robotic systems are intended to perform a wide variety of tasks. Some of these tasks may require specialized hardware, meaning that all modules in a homogeneous modular robot would need to feature this hardware in order for said tasks to be accomplished. This would increase the cost and complexity of each module. A solution to this is to develop specialized tools, lacking most of the functionality of the main module of a system, but gaining other application-specific functionality.

This chapter presents HyMod extensions; modules built to add specialized capabilities to a modular robot. Past systems to employ specialized modules include [24, 25, 44]. An extension module must contain processing and local communication (primarily for identification purposes), as well as at least one passive HiGen connector. A passive HiGen connector is one that is in a constant extended state, allowing for an active HiGen connector to attach to it without prior communication. This removes the need for extensions to contain their own power source. Extensions could therefore reside in known pick-up locations to be collected by modular robots when needed. Four extensions have been developed for the HyMod system (see Figure 5.1), covering the areas of manipulation, mobility, perception and support.

The remainder of this chapter presents the requirements that resulted in extension modules (Section 5.2), the common structural and electronics elements of extensions (Section 5.3), the four extensions developed (Sections 5.4, 5.5, 5.6, and 5.7), and a number of configurations such extensions allow the HyMod system to form (Section 5.8).



(a) (b) (c) (d)

Figure 5.1.: Four of the extensions created for the HyMod system; (a) Gripper extension, (b) Mecanum Wheel extension, (c) Camera extension, and (d) Modular Surface extension. (a), (b), and (c) are placed on an extension holder, which can be attached to the side of (d) to create a pick-up location.

5.2. Requirements

To introduce specialized functionality to the HyMod system, the extension framework should offer standardized elements for the structure and electronics of extension modules. These elements should provide all the features necessary to produce functional modules that can integrate with the HyMod system, with only minor modifications being required for any specialized functionality. In terms of electronics, it would be beneficial if the framework did not enforce any requirements on power source. These high-level goals for extension modules give rise to the following systematic requirements:

5.2.1. Mechanical

Extension modules shall:

E.1.1 - Be primarily constructed using 3D printing technology. A Stratasys Mojo [113] is available in-house, capable of printing parts in ABS plastic with soluble supports

E.1.2 - Feature a passive genderless connector that is four times symmetric

E.1.3 - Feature a profile that allows for them to reside in pick-up locations for other modules within a lattice to connect to (if appropriate for their use case)

5.2.2. Electrical

Extension modules shall:

E.2.1 - Be capable of being powered through their connector(s) and (depending on their use case) via external power

E.2.2 - Contain their own microcontroller that interfaces with their connector(s)

a) Connections for sensors and servos shall be included to allow for basic extension functionality

b) Additional inputs and outputs shall be exposed to headers to allow for expanded functionality, such as via Arduino-like add-on boards

E.2.3 - Feature indicator LEDs to visually report operation state

E.2.4 - Be reprogrammable

5.2.3. Environmental

Extension modules shall:

E.3.1 - Operate at a standard humidity for an indoor laboratory environment

E.3.2 - Operate within a temperature range of 20 to 30 degrees Celsius

5.2.4. Performance

Extension modules shall:

E.4.1 - Become operational within three seconds of receiving a connection from a non-extension module or from an external power source

E.4.2 - Have 3D printed parts that are no larger than the 127 x 127 x 127 mm build volume of the in-house Stratasys Mojo [113]

5.2.5. Reliability

Extension modules shall:

E.5.1 - Extensions should handle hundreds of repeated connections and disconnections without failing

5.3. Expanding HiGen to Extensions

The purpose of extensions is to allow for application-specific functionality to be introduced to a modular robot with relative ease. As such, it is important that the number of essential elements for an extension be minimal. The primary element of HyMod extensions is their connection interface. Using a normal HiGen connector, due to its active mechanism, means that any module that features it must contain a power source in order to initiate the actuation process. For the HyMod unit this is acceptable; however, for extensions it may be impractical to introduce a power source. To overcome this problem and satisfy requirement E.1.2 in the process, a passive variant of the HiGen connector was devised, which can be connected to by an active connector but does not contain its own actuation mechanism. This allows extensions to be made that are unpowered and are activated once connected to, drawing power from the HyMod system's shared power bus. Additionally, an electronics framework was devised. This framework deals with the communication requirements of HiGen whilst also allowing for common functionality, such as reading sensors or driving servos, to be performed without requiring extra circuitry.

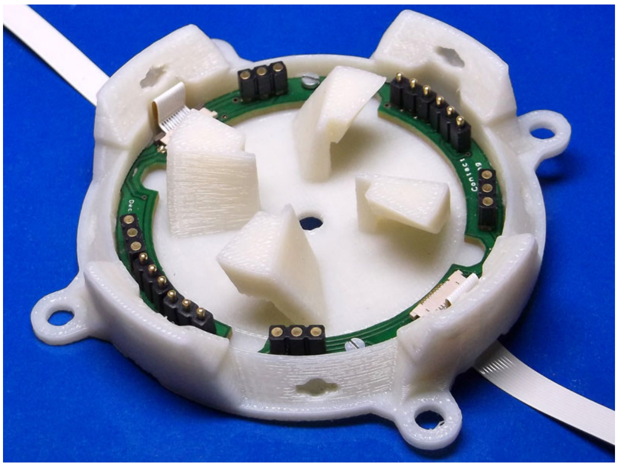


Figure 5.2.: A passive HiGen connector.

5.3.1. Passive Connector

A passive HiGen connector takes advantage of the genderless and single-sided disconnect nature of the active design. When an active connector is in its extended state, another active connector can freely connect to and disconnect from the first connector without the first needing to perform any actions. In fact, the first connector does not need to be powered in this situation. This realisation makes the development of a passive connector trivial, as the extended state of HiGen can be taken and modified to produce a single 3D printable component, containing docking hooks, the alignment shroud, and the top surface of the upper housing. A passive connector is shown in Figure 5.2. It consists of just two components, the passive connector itself and a *contact ring*. This is in contrast to the nine components of an active connector. Animation steps of the process of an active connector joining to a passive connector are shown in Figure 5.3.

Due to a passive HiGen connector being functionally identical to an extended active HiGen connector, the clearance relation presented in Section 4.3.3 is broken. As a result, any HyMod unit that is a neighbor to an extension must translate either away or along the connector surface before it is able to rotate in place. It was decided that the ability to have extensions be composed of few parts and not require their own power source outweighed this clearance limitation.

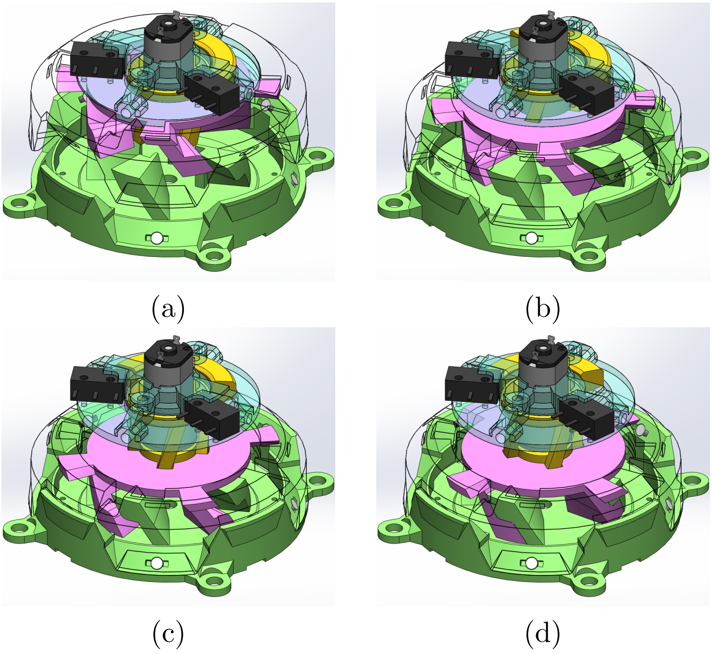


Figure 5.3.: Snapshots of the connection sequence of an active to a passive HiGen connector, showing the hooks (in purple) extending behind the hooks on the passive side and locking in place. The shroud is transparent to help show the motion of the hooks.

5.3.2. Electronics Framework

To facilitate in the rapid development of HyMod extensions, an electronics framework was devised, consisting of two custom circuit boards, a *tool controller*, and a *tool extender*. These can be seen in Figure 5.4.

The *tool controller* board serves two purposes for the framework; it contains the necessary electronics to interface with and perform local communication across a HiGen connector, as per requirement E.2.2, and it is capable of acting as the central microcontroller of simple extensions that do not require much processing power or access to the global communication bus. To create this board, the *HiGen controller* presented in Section 4.3.4 was modified to remove unneeded functionality, such as the motor driver, contact switches, analogue switch, and infrared distance sensors. Distance sensors were removed due to them serving no purpose in the primary use-case of extensions, of being connected to a module that provides them with power. Additionally, the analogue switch used to divide the global communication bus was removed as it is guaranteed that the connecting HiGen will be active and therefore contain a switch, making one on the *tool*

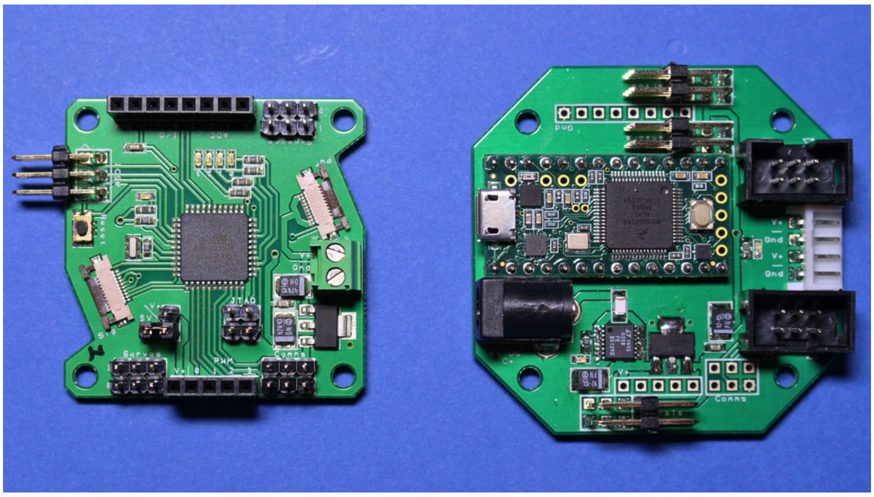


Figure 5.4.: The two custom printed circuit boards that form the HyMod extensions' electronics framework, with the *tool controller* on the left, and the *tool expander* on the right.

controller unnecessary. The programming header and indicator LEDs from the *HiGen controller* remain, fulfilling requirements E.2.3 and E.2.4. One piece of functionality the *tool controller* has over the *HiGen controller* boards is a 5 V regulator, that takes the voltage from the shared power bus and uses it to power extension. This allows for the board to be powered either through its connector or from an external source that is compatible with the shared bus, satisfying requirement E.2.1.

To make the *tool controller* usable for a variety of applications, the concept of add-on boards was adopted from the Arduino hardware platform [117], as requested by requirement E.2.2b. Add-ons are circuit boards that stack on top of a main processor board by a number of header connections. For the *tool controller* all but four unused pins from the microcontroller are routed to such headers. To fulfil requirement E.2.2a and allow for some basic uses of the *tool controller* without requiring an add-on board, the four unused pins are routed to two sets of connections, one set for controlling up to two servo motors, and another set for reading values from up to two sensors. This allows for basic extensions to be developed without requiring additional circuitry.

Although the *tool controller* is sufficient for simple extensions, some extensions may require more capabilities. To address this, an add-on board called the *tool extender* was developed. The primary purpose of the *tool extender* is to give extensions the option of having the same processing capabilities as a HyMod unit, as well as access to the global communication bus. This is achieved by the board housing a Teensy 3.2 microcontroller and related CAN bus components. Additionally, the board can accept an external 9 V power source and contribute it to the shared power bus of the HyMod system. This is beneficial as it enables static extensions such as docking stations to be developed, capable of powering the electronics of a module until it needs to disconnect to perform a task. Additionally, by having the board feature a Teensy microcontroller, multiple I^2C devices such as connectors can be addressed. To take advantage of this, the *tool extender* has connections for two ribbon cables, allowing any number of *tool controller* boards to be commanded from a single extender, opening up the option for connector grid extensions. In such a situation, the board would not be an add-on and instead act in a stand-alone manner. Similar to the *tool controller*, the extender also has unused pins routed to connections for servos and sensors, as the board's shape blocks those connections from any *tool controller* mounted below. Additionally, a general header exists for communicating to other microcontrollers over serial links.

5.3.3. Extension Holder

The passive HiGen connector and electronics framework allows for extensions to be created that have a single connection point, meaning that in order for a HyMod unit to connect to such extensions, they need feature a profile that allows them to reside in some kind of pick-up location, as specified by requirement E.1.3.

To create a pick-up location, a holder and accompanying chassis template were designed (Figure 5.5). The template consists of a ring that slots in to the holder to keep the extension in place whilst leaving the HiGen connector unobstructed. Both the holder and template conform to the dimension constraints of requirement E.4.2, with the width of the holder being 1 mm below the maximum at 126 mm. The centerline of any extension placed in to a holder is raised 64 mm relative to the holder's base, half the width of a HyMod unit. To aid in the rotational alignment of an extension placed in to the holder, the template's ring has four indentations. These indentations match a protrusion in bottom of the holder, allowing the extension to rest at 0, 90, 180, or 270 degrees.

To enable a single HyMod unit to pick an extension out of a holder using only rotation, the outer surface of the template's ring conforms to the surface of a sphere centered around the position of the HyMod unit. This is in addition to a group of modules being able to lift an extension out via translation, by coordinating their motions. Animation steps of a single HyMod unit lifting the template out of an extension holder can be seen in Figure 5.6.

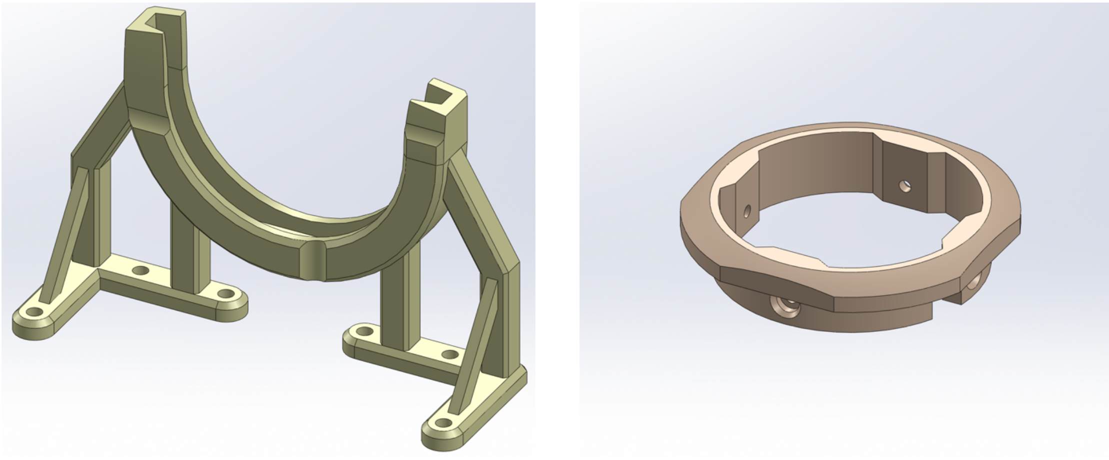

(a) (b)

Figure 5.5.: 3D renders of the (a) holder and (b) chassis template used to create a pick-up location for extensions.

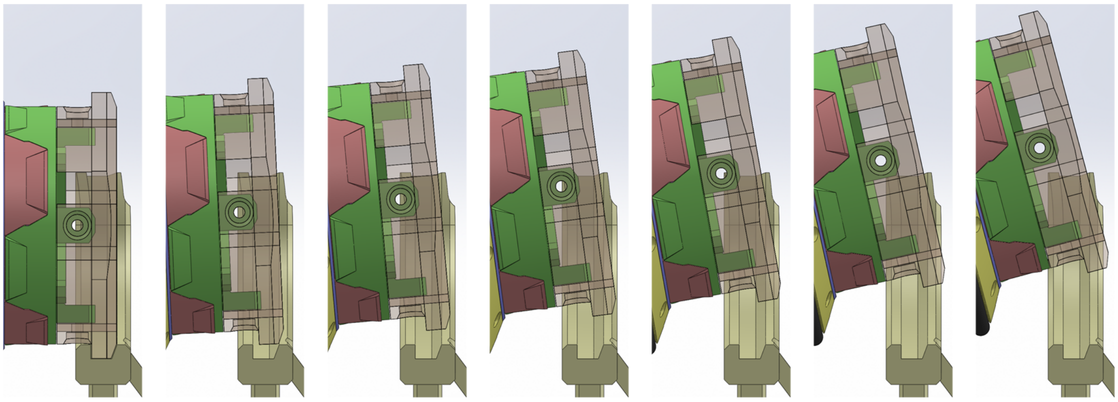
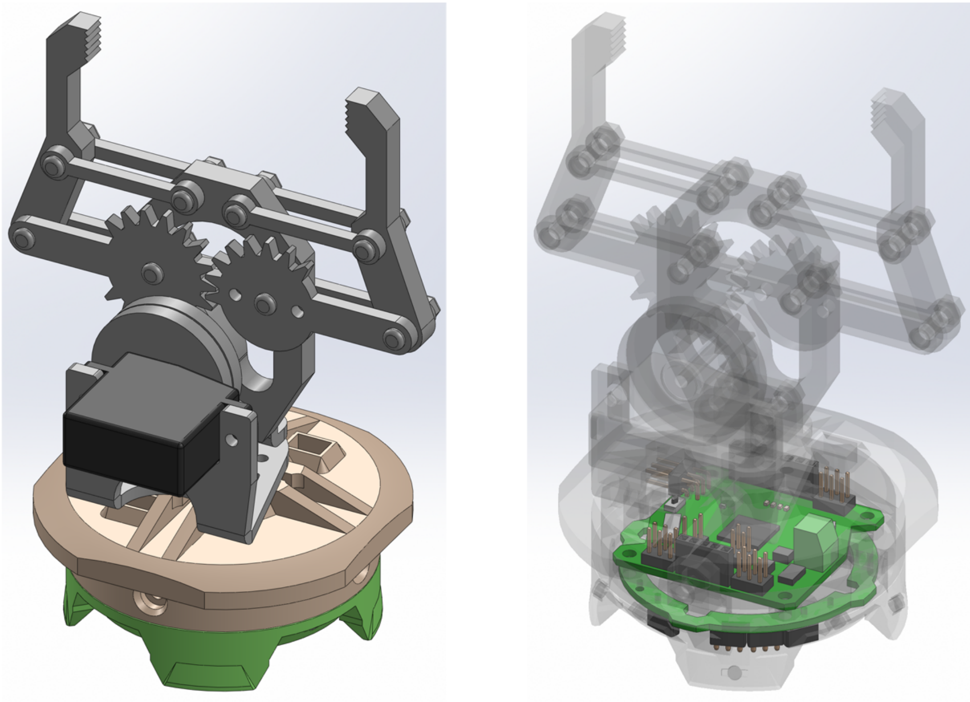

(a) 0° (b) 2.5° (c) 5° (d) 7.5° (e) 10° (f) 12.5° (g) 15°

Figure 5.6.: Snapshots of a HyMod unit lifting an extension out of a holder. The template is transparent to show how its shape interacts with that of the holder.

5.4. Gripper Extension

The ability to manipulate objects is an important feature of many robotic systems. To create an extension for the HyMod system that is capable of manipulation, an off-the-shelf gripper was sourced. The gripper chosen was the MKII Robot Gripper by Dagu [118], due to it being readily available and fitting within the dimensions of a single HyMod lattice position. It is built from aluminium and consists of two parallel actuated gripping fingers driven by a servo motor. A 2:1 gear ratio is applied to the servo output



(a) (b)

Figure 5.7.: 3D renders of the Gripper extension, showing the (a) external structure and (b) internal electronics.

to increase the capable force of the gripper. Additionally, a spring and magnet based clutch is placed between the servo and the output to prevent damage to the motor when attempting to apply an excessive force to an object. The maximum finger separation of the Dagu gripper is 55 mm.

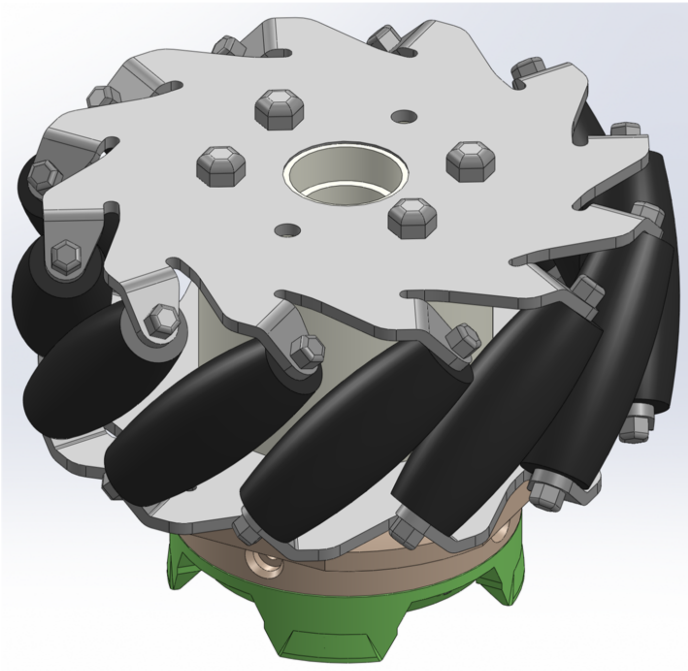
Due to the Dagu gripper only using a servo motor to actuate, minimal control and processing is required by the extension. As such only the *tool controller* from the electronics framework is necessary. A chassis was designed to interface the passive HiGen connector with the gripper, via its rear mounting screws. Square cut outs were added to the chassis to allow for the servo to be connected, as well as sensors for possible future expansion, such as adding force sensing to the fingers. Note that although the gripper itself fits within a single lattice position, with the chassis and connector mounted, the gripper's fingers need to be in their fully separated position for this to remain true. Figure 5.7 shows renders of the extension and its internal electronics. Additionally, Figures 5.12(a) and 5.13(a) show block and network diagrams of the extension's electronics, respectively. The weight of the Gripper extension is 220 g.

5.5. Mecanum Wheel Extension

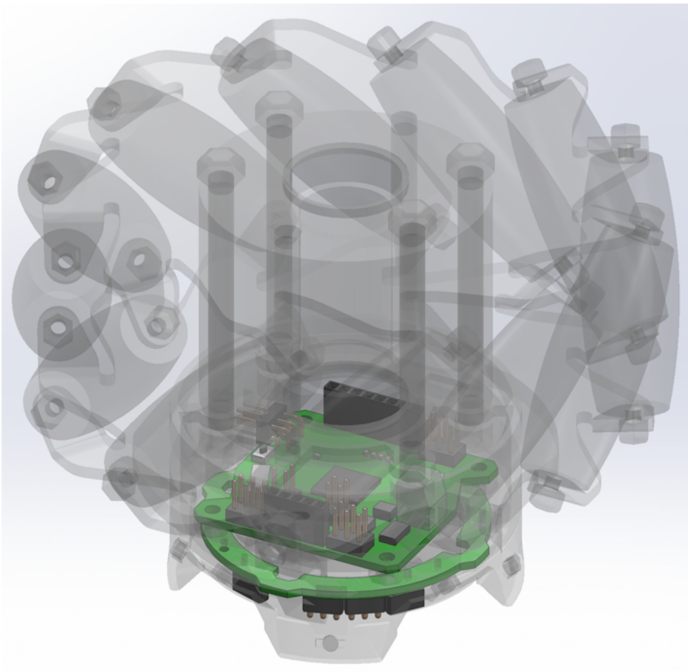
HyMod units are capable of independent locomotion via differential wheels, allowing for individual modules to efficiently navigate a flat environment. Unfortunately, this ability can be diminished as modules are connected together in different configurations. For instance, a line of HyMod units arranged end-to-end would be capable of driving forward and backward, but would have difficulty turning on the spot due to friction. To overcome this, a Mecanum Wheel extension was developed (inspired by [119]), allowing for omni-directional motion when four or more are placed on a system [120].

Mecanum wheels consist of a central hub with a number of free-spinning rollers around their perimeter. These rollers are placed at 45 degrees to the hub's rotation axis, creating the effect of the wheel applying a force in a diagonal direction when rotated. By placing four or more mecanum wheels on a robot (two left-handed and two right-handed), these diagonal vectors can be combined to create a net translational vector in any direction, as well as allow for rotation on the spot. Details of the geometry and kinematics of mecanum wheels can be found in [121].

To create a Mecanum Wheel extension, an off-the-shelf wheel was sourced. A four-pack of 5 inch (127 mm) aluminium wheels was selected from a company called Nexus Robots [122]. This size was chosen due to it being larger than the HyMod units' wheel diameter, and less than the dimensions of a single HyMod lattice position, offering higher ground clearance and allowing two to be placed on neighboring modules without interference. Due to HyMod units containing two continuous rotational degrees of freedom, it is not necessary for Mecanum Wheel extensions to feature their own drive mechanism, instead they can be turned by the module they are attached to. This made the creation of a chassis for the extension relatively simple because, like the gripper, all that was required was a component that interfaced between a passive HiGen connector and the wheel itself. Although a Mecanum Wheel extension is mechanically passive, it is still necessary for it to contain a *tool controller*, to allow for the extension to be identified by neighboring modules. Having a *tool controller* on board also opens up the possibility for future expansion, such as adding distance sensing inside the wheel hub to report the distance the wheel may be away from an object when translating towards it, for example. Figure 5.8 shows renders of the extension and its internal electronics. Additionally, Figures 5.12(b) and 5.13(b) show block and network diagrams of the extension's electronics, respectively. The weight of the Mecanum Wheel extension is 560 g.



(a)



(b)

Figure 5.8.: 3D renders of the Mecanum Wheel extension, showing the (a) external structure and (b) internal electronics.

5.6. Camera Extension

Vision is an important feature for many tasks a robot may be required to perform, be it as a video feed for tele-operation, or for environmental details to automatically be identified and reacted to.

To create a Camera extension, a Raspberry Pi Zero 1.3 [123] along with the Raspberry Pi Camera were chosen. This small board is a linux-based computer running at 1 GHz and with 512 MB of RAM. The Pi Zero 1.3 is an improvement over previous versions as it features a connection for a Pi Camera, allowing for small video streaming platforms to be developed. The choice of a Raspberry Pi was based on research conducted by a Masters student by the name of Mohamed Marei, who was tasked with creating a prototype for a Camera extension. The Camera extension presented here extends upon his work.

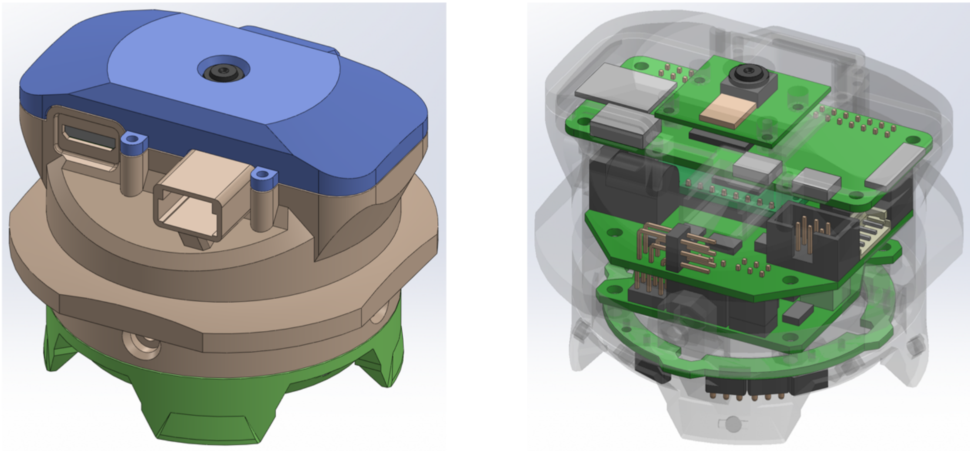
The use of a Raspberry Pi opens up the possibility for the Camera extension to also be used as the central brain of a modular robot composed of HyMod units, as its processor is an order of magnitude faster than the Teensy 3.2 microcontroller. To take advantage of this possibility, the Camera extension houses both the *tool controller* and *tool extender*, enabling the Raspberry Pi to indirectly (via the extender's Teensy) communicate over the global network to all connected modules.

To allow for a video feed to be received from the Raspberry Pi, a USB WiFi dongle is used. This dongle enables the processor to be connected to as a computer on a network, bypassing the need for video to be sent through the global communication bus of the HyMod system and out of a unit's Bluetooth modem, which that network would be unsuitable for due to its relatively low bit rate.

Figure 5.9 shows renders of the extension and its internal electronics. Additionally, Figures 5.12(c) and 5.13(c) show block and network diagrams of the extension's electronics, respectively. The weight of the Camera extension is 125 g.

5.7. Surface Extension

In addition to wheel-based locomotion, HyMod units can also locomote via self-reconfiguration, using neighboring modules to form a grid structure. For some tasks or environments such



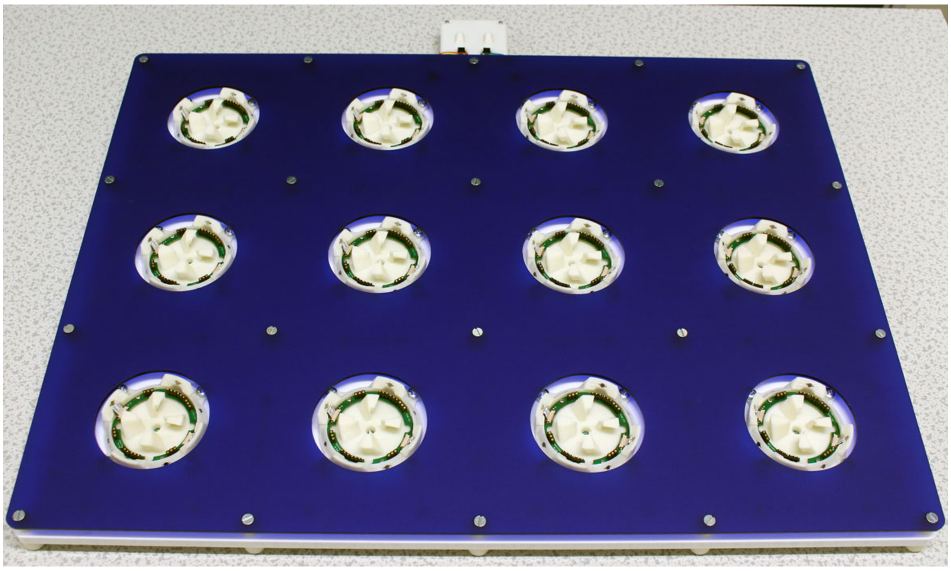
(a) (b)

Figure 5.9.: 3D renders of the Camera extension, showing the (a) external structure and (b) internal electronics.

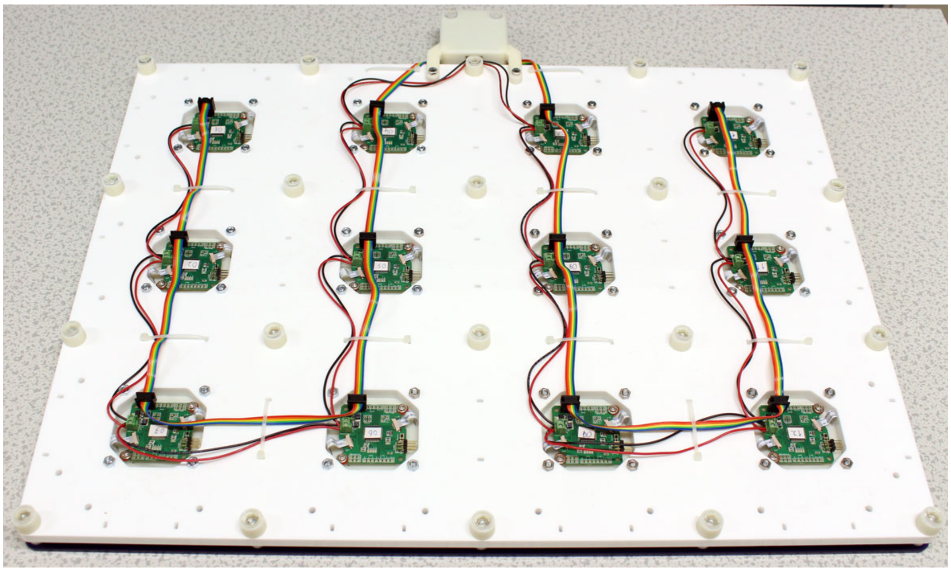
a grid may be required, but the number of modules available may be insufficient, due to cost for example. This can be overcome using a modular grid of passive connectors. As such, a Surface extension was developed (inspired by [124]).

A Surface extension consists of a *tool extender* in a control box connected via ribbon cables to one or more *tool controller* boards, one under each passive connector. The passive connectors are mounted on laser cut acrylic, with each being 140 mm from its neighbor (the cubic lattice size of the HyMod system). An additional laser cut acrylic layer is mounted on top of the first to recess the passive HiGen connectors and allow for HyMod units to drive over the surface should the need arise.

Two versions of the surface were designed, a modular grid, and a single 4 x 3 grid. The cells of a Modular Surface extension were designed to fit together in much the same way as the pieces of a jigsaw puzzle, allowing for any shape or size of 2D grid to be formed. The single grid on the other hand is intended for a single use-case, but offers increased rigidity and lower cost due to it only being comprised of two acrylic pieces, versus the two pieces per cell of a modular grid. Images of the fixed 4 x 3 grid are shown in Figure 5.10. With additional 3D printed components it is possible to mount the holder presented in Section 5.3.3 on the side of either Surface extension, enabling HyMod units to self-reconfigure to a position adjacent to an extension to connect to it.



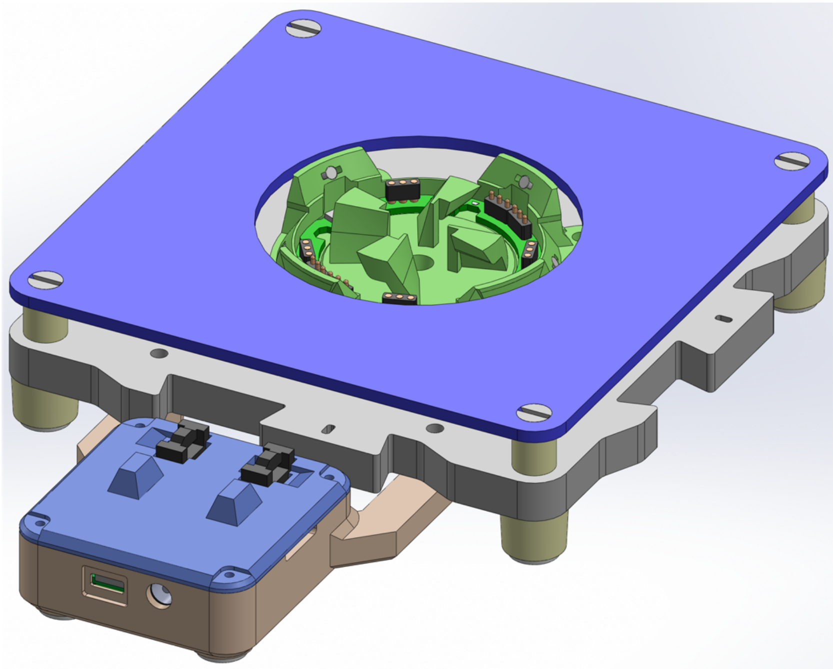
(a)



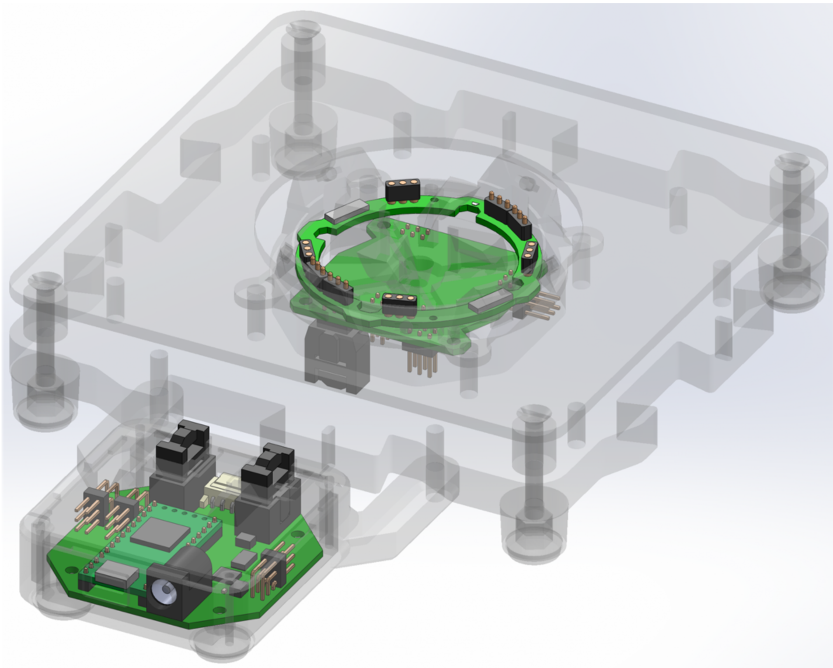
(b)

Figure 5.10.: A single 4 x 3 Surface extensions, viewed from the (a) top, and (b) bottom. The underside shows the 12 passive HiGen connectors wired to the control box via two ribbon cables and daisy-chained power cables.

Figure 5.11 shows renders of the extension and its internal electronics, for a single modular cell. Additionally, Figures 5.12(d) and 5.13(d) show block and network diagrams of the extension's electronics, respectively.



(a)



(b)

Figure 5.11.: 3D renders of the Modular Surface extension, showing the (a) external structure and (b) internal electronics.

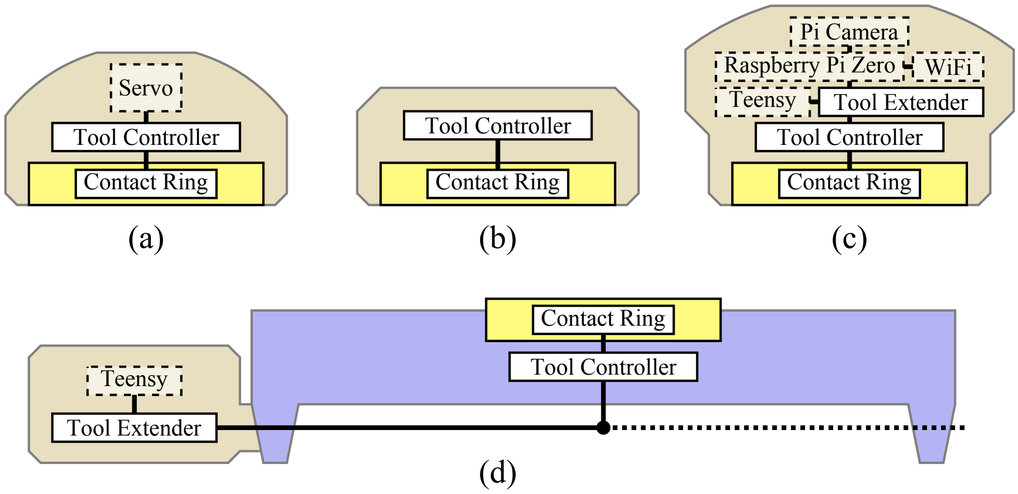


Figure 5.12.: Block diagrams showing the circuit boards and other components within the (a) Gripper extension, (b) Mecanum Wheel extension, (c) Camera extension, and (d) Modular Surface extension. White blocks are the custom boards created for this project.

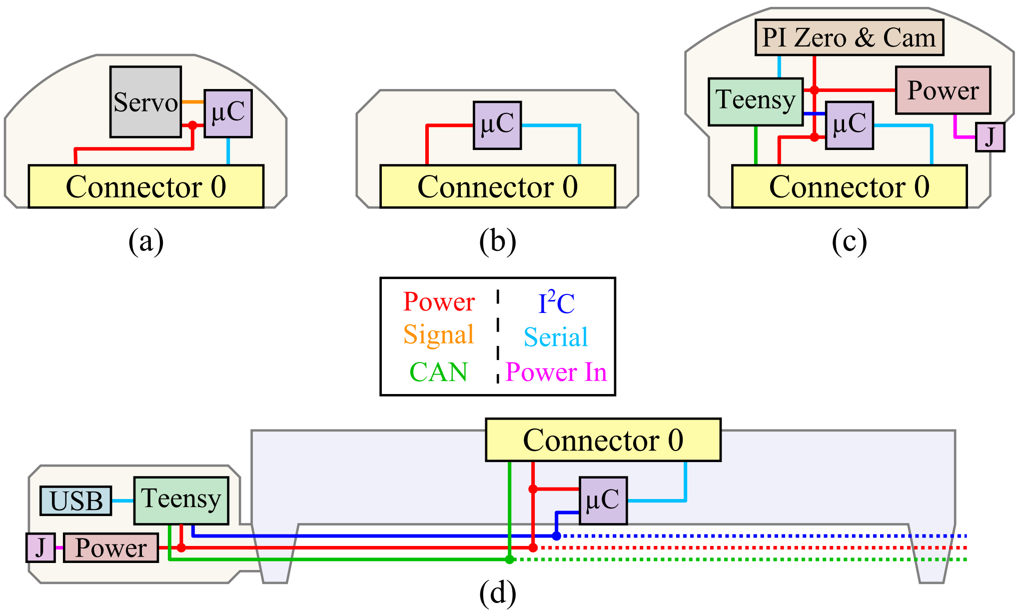


Figure 5.13.: The power and communication network within the (a) Gripper extension, (b) Mecanum Wheel extension, (c) Camera extension, and (d) Modular Surface extension. μC and J denote the microcontroller and power input jack, respectively.

5.8. Example Extension Configurations

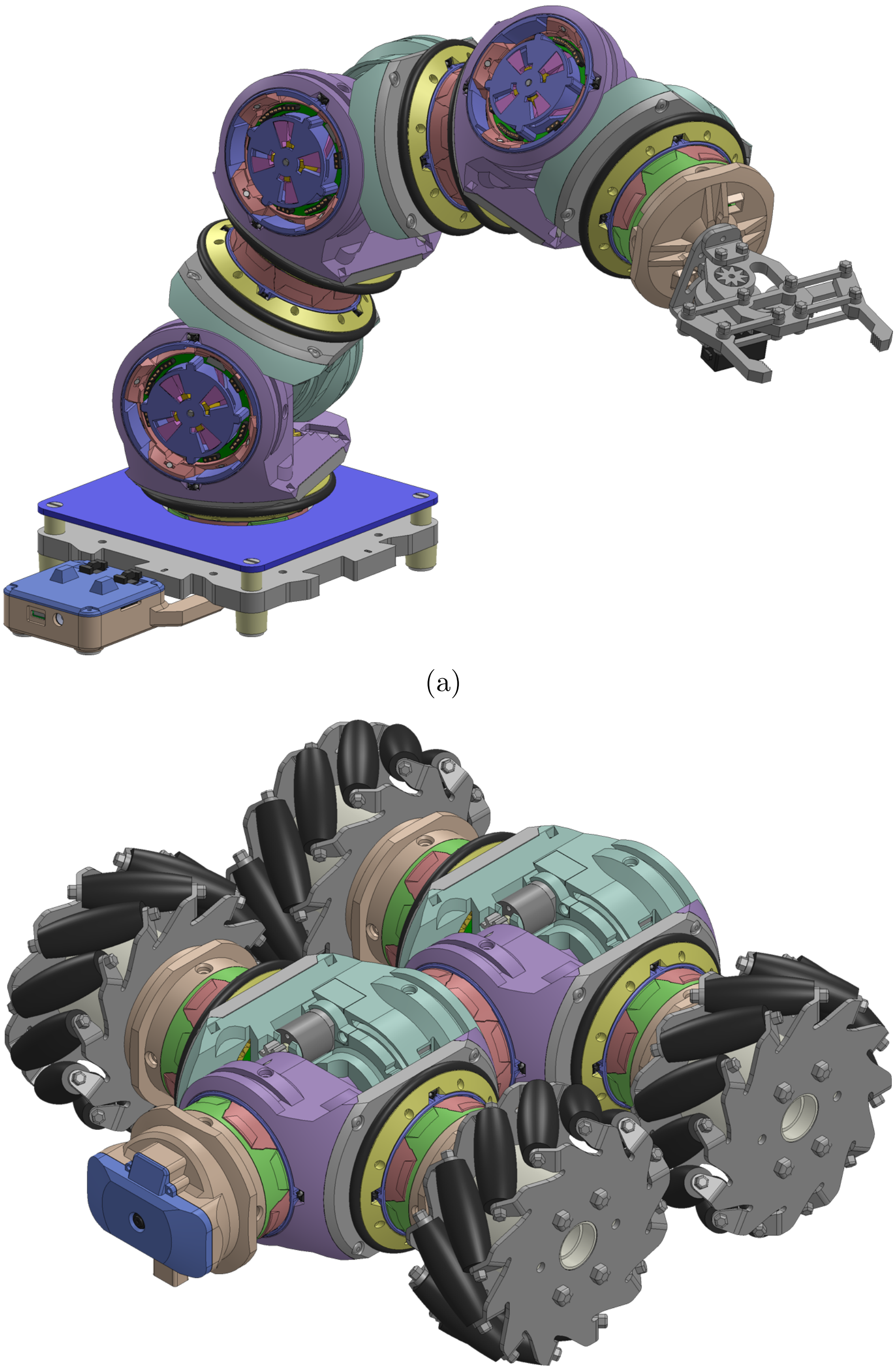
Figure 5.14 illustrates how (a) a manipulator arm and (b) an omni-directional rover could be constructed out of combinations of HyMod units and extensions. The manipulator arm takes advantage of the 3-DOF of each unit to create a 7-DOF manipulator, with the Gripper extension mounted as the end-effector. The omni-directional rover takes advantage of the two continuous rotational degrees of freedom of each HyMod unit to drive Mecanum Wheel extensions to produce motion in any direction on a flat surface. A Camera extension is mounted at the front of the rover to allow for either tele-operation or autonomous operation.

5.9. Summary

This chapter presented HyMod extensions, modules that augment the capabilities of the HyMod system. Extension modules allow for the introduction of application-specific functionality to a modular robot, without requiring that each main module include said functionality, increasing their complexity.

For the creation of HyMod extensions, a novel framework was devised that consists of a passive variant of the HiGen connector, a chassis template, and control circuitry. The passive connector allows for extensions to be developed that can be connected to without needing their own source of power, reducing their complexity. Such extensions, via use of the chassis template and a custom holder, can reside in known pick-up locations for HyMod units to collect as and when required. The control circuitry for extension modules is modified from the electronics present in HyMod to provide common functionality such as HiGen control, local and global communication, and power sharing, as well as support for connections to sensors and actuators. By combining the three elements of the framework, extension modules can be developed much faster than would otherwise be possible with traditional module development methods.

To demonstrate the potential of the framework, four extension modules were built that expand the capabilities of HyMod; a Gripper extension for manipulation tasks, a Mecanum Wheel extension for mobility tasks, a Camera extension for perception tasks, and a Modular Surface extension for supporting other tasks. Details of the four extensions were presented, as well as examples shown of how they may be used together with HyMod units as part of useful configurations.



(a)

(b)

Figure 5.14.: Two example configurations of HyMod system modules; (a) three units and two extensions (one Gripper extension and one Modular Surface extension) forming a manipulator arm, and (b) two units and five extensions (one Camera extension and four Mecanum Wheel extensions) forming an omni-directional rover.

6. Control and Self-Reconfiguration

6.1. Introduction

Traditionally in robotic systems, control is achieved using a central processor, capable of reading in sensor data and outputting commands to motors, either directly or via the use of a hierarchy of sub-processors. This arrangement of processors makes such robots straightforward to operate, as the central processor acts as a single point of control, where instructions can be sent, or programs and algorithms uploaded. For modular robotic systems, although the arrangement of processors still applies to individual modules, bringing several modules together results in a distributed system being created, with multiple points of control. This makes modular robots more complex to operate than traditional robots, as either techniques need to be employed to elect a leader module to command other modules, creating a single point of failure, or distributed algorithms need to be created to allow for modules to perform actions using only information from themselves and possibly their neighbors.

This chapter presents a centralized and distributed control architecture for the HyMod system. Based on message routing [125], the architecture allows for modules to send messages to and receive messages from other modules in a transparent manner, with messages being sent across multiple communication channels and forwarded by any intermediary modules as necessary. Both neighbor-to-neighbor and source-to-destination communication are possible, with the former communication not requiring knowledge of module identifiers, benefiting distributed algorithms.

The remainder of this chapter describes the HyMod control architecture, including its message structures and routing implementation (Section 6.2), and presents a self-reconfiguration scenario being performed in simulation (Section 6.3). Finally, Section 6.4 concludes the chapter.

6.2. Control Architecture

A set of connected HyMod units and extensions share similarity with nodes in a computer network, capable of sending messages to other connected nodes. There are two common ways such nodes can be connected; a Bus topology or a Star topology [126]. In a Bus topology, each node is connected to a central communication bus. Nodes can broadcast messages on to this bus for all other nodes to receive, as well as listen for messages of interest from any other node on the network. This topology allows for one node to communicate directly with another on the network regardless of distance, as every node will receive the message almost simultaneously, and can use address filtering to only act upon messages intended for them. In a Star topology, each node is connected to a central connection point. This point receives all messages and either broadcasts them on to all other nodes, or selectively routes the messages to the intended nodes via the use of addresses. To enable larger networks, multiple Star connection points can be joined together, with each routing all or a subset of messages to their neighbors to handle. This, at a most basic level, is how the internet operates.

As described in Section 4.3.4, HyMod units are capable of both local and global communication with neighboring modules. Each module can broadcast messages on to their CAN bus as well as send messages directly to adjacent modules via their connectors. By using these two approaches to communication, a combination of the Bus and Star topologies is produced. A network topology diagram of this combination, for multiple connected modules, is shown in Figure 6.1. This topology allows for messages to be routed from an external device, such as a computer, on to the global bus to be sent to other networked modules. Additionally, for modules that lack access to the global bus, such as the Gripper extension (Section 5.4), messages can be routed to them via neighboring modules with bus access.

6.2.1. Message Structures and Routing

There are two types of messages used by the HyMod system, commands and responses. A *command* is a *message* that a module will act upon, and a *response* is a *message* that a module returns as an acknowledgement. This distinction allows for messages to be filtered based upon whether the data they contain is to be processed, or is information that was previously requested. In addition to type, messages also contain an identifier

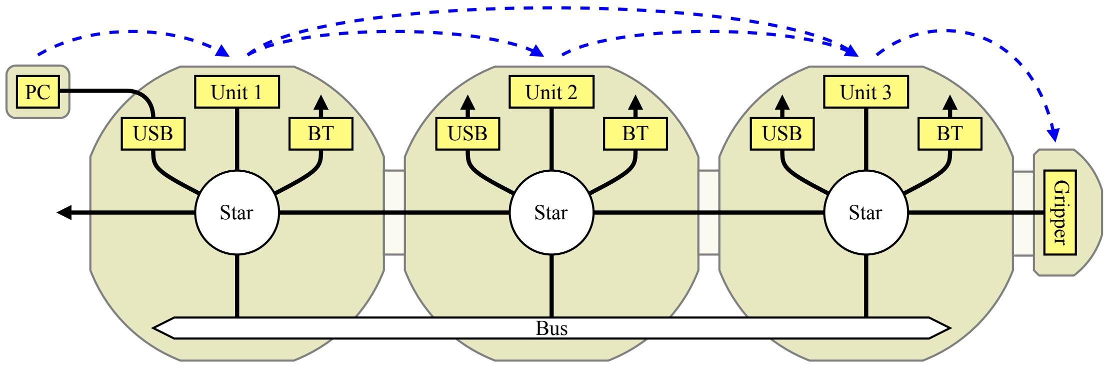


Figure 6.1.: The network topology produced by three HyMod units, a Gripper extension, and an external computer (PC). Blue dashed lines show how messages from the computer can be routed via the Star and Bus topologies to reach all of the modules. BT denotes the Bluetooth serial link in each HyMod unit.

that defines their purpose. The type and identifier form a one byte message header, with one bit for the type and seven for the identifier (allowing for 128 unique message purposes per module). This byte is followed by an arbitrary number of data bytes to create a full message (see Figure 6.2). As an example of a message transfer, suppose the external computer in Figure 6.1 wants the current accelerometer reading from Unit 1's IMU. To retrieve this information the computer would send a command message with the identifier *GetAccelerometerData* to Unit 1, which would then send back a *GetAccelerometerData* response message containing the readings. This approach to message exchange is sufficient for the distributed control of modules, as it allows for information to be passed between neighbors; however, it is lacking details about which module the message is intended for, making it unsuitable for centralized control.

To allow for centralized control of the HyMod system, packets are used. A *packet* is a wrapper around a *message* that includes module source and destination addresses. When a module receives a packet, the destination address is compared with the module's own address. If the two addresses match then the message can be extracted and dealt with like messages from distributed control. If such messages are commands then any resulting response is placed in a new packet with the original source and destination addresses exchanged to allow for it to be returned to sender. If the packet is not for the module, it can be forwarded to another module to bring it closer to its destination.

Due to the self-reconfigurable nature of the HyMod system, it was opted to have modules forward packets across all available communication channels (e.g. connectors, CAN bus

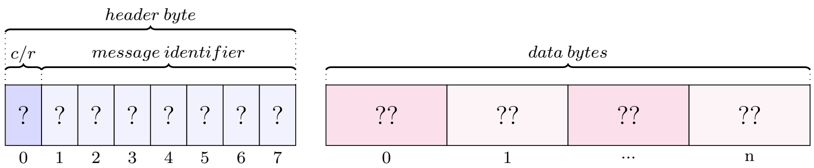


Figure 6.2.: The byte structure of a *message*. The first byte is the header and contains a *command* / *response* bit and seven identifier bits (allowing for 128 unique message purposes). Following that are an arbitrary number of data bytes.

etc) as opposed to selectively forwarding them to destination modules via a routing table. This choice presents two problems; how to avoid a module acting on echoes of command packets it has previously received, and how to prevent packets from being forwarded through the network indefinitely. To overcome these problems, the header of a packet contains two additional values, a sequence number and a jump count. These are adopted from the Transmission Control Protocol / Internet Protocol (TCP/IP) used in computer networking [127]. The sequence number provides a means for a receiving module to compare the relative age of a packet. For instance, if a module receives a packet over the CAN bus with sequence number 3, and then receives another identical packet from a connector, the second packet can be ignored as its number no longer matches the next expected sequence number of 4. For this to work, both the sending and receiving modules need to keep a record of the current sequence number (with there being the ability to synchronise the two modules' numbers if necessary). Additionally, the size of the numbers available should be large enough such that a repeat sequence number from wrapping around is unlikely to be encountered soon after the last. The jump count serves the purpose of limiting the amount of times a packet can be forwarded through the network. The count starts at a high value and is decremented each time the containing packet is forwarded. Once the count reaches zero the packet can no longer be forwarded and is discarded if the module it arrives as is not the intended destination.

A consideration for packets is the number of bits available for their header information. Unlike communication over serial and I^2C, which have no imposed limit on the number of bytes that can be sent at a time, CAN only supports eight data bytes per CAN frame. Additionally, the standard requires that any data sent must have a unique frame identifier of 29 bits. This identifier is used to handle any communication conflicts that may occur, such as multiple modules attempting to communicate at the same time. This is referred to as arbitration [128]. If two modules send a frame with the same identifier (due to insufficient uniqueness), both will believe they have won arbitration and attempt to

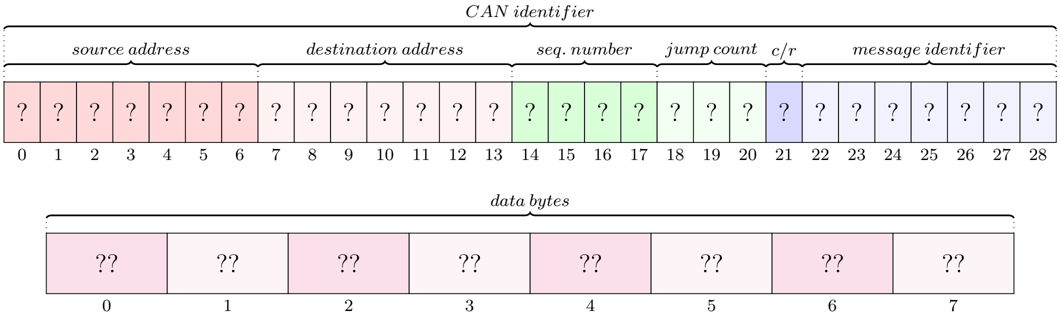


Figure 6.3.: The byte structure of a *packet*. 29 bits are used for the CAN identifier, which contains both the packet header and message header. The packet header consists of seven bit source and destination addresses, a four bit sequence number and a three bit jump count. The message header contains a *command* / *response* bit and seven identifier bits. Following the identifier are up to eight data bytes.

transmit their data, potentially resulting in data corruption. To ensure the uniqueness of CAN frames within the HyMod system, both the packet and message headers are used to form the frame identifier (see Figure 6.3). Because message headers are a byte in length, only 21 bits are available for the four properties of a packet header. As such, it was decided to allocated seven bits to both the source and destination addresses (allowing for 128 modules on a single network), four to the sequence number, and three to the jump count. This division between sequence number and jump count was chosen so that forwarded packets expire before the same sequence number is repeated. Three bits are used for the jump count, giving a maximum of seven jumps, to reduce the likelihood of packets expiring before they reach their destination in a modular network combining both local and global communication routes. For example, with only two jump bits, giving a maximum of three jumps, it would not be possible for packets to reach the Gripper extension in the configuration shown in Figure 6.1 if the global bus is disabled between two of the HyMod units.

6.2.2. Serial Frame Communication

Asynchronous serial is a bit-level communication protocol widely used by microcontroller electronics for transferring data between pairs of devices. The protocol allows for data to be sent from one device to another using a single communication line (two for bi-directional communication). To deal with the communication being asynchronous, each

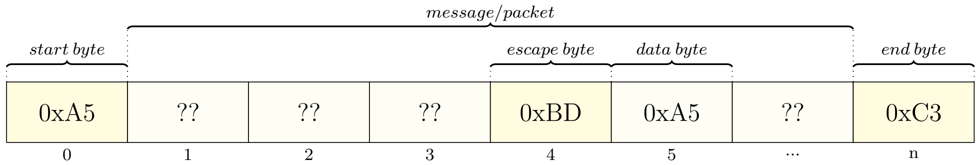


Figure 6.4.: The byte structure of a message or packet serial frame, including an escaped byte.

transmitted byte is preceded and followed by a number of start and stop bits, respectively. These bits allow the receiving device to synchronise itself with the data without requiring a separate clock signal to accompany it, as is the case for synchronous serial. Additionally, a parity bit can follow each byte to enable basic data validation. The advantage of serial protocols is that they allow streams of characters to be easily output from a device to, for example, be viewed on a computer terminal; however, for multi-byte data that needs to arrive as a contiguous block, such as HyMod messages and packets, additional logic is required.

To allow for multi-byte communication of packets and messages over asynchronous serial, be it between connectors or over USB or Bluetooth, a byte-level communication protocol was devised, following the information presented in [129]. Data sent via this protocol is preceded by a *start byte*, instructing the receiving device to buffer any bytes that follow it. Once the data is transmitted an *end byte* is sent, instructing the receiving device to close the buffer and forward its content on to other sections of the program. The grouping of *start byte*, *data bytes* and *end byte* form a serial frame. Because the start and end bytes are specific binary values, it is possible for the data being transmitted to also contain these values. To overcome this problem the protocol precedes any such value by an *escape byte*, instructing the receiving device to ignore the special meaning of the following *data byte*. In addition, if a *start byte* that is not escaped is encountered during a frame, the receiving device assumes that a problem occurred during communication and flushes the previous buffer ready for the new data. The structure of a serial frame, including an escaped byte, is shown in Figure 6.4.

An advantage of implementing a frame-based protocol is that, if bi-directional communication is available, it allows for acknowledgements to be returned once data is received. This is an ability natively supported by protocols such as I^2C and CAN, but not asynchronous serial. When a serial frame is transmitted, the receiving device can choose to send an *acknowledgement byte* back, informing the sender that the data arrived. The number and meaning of the acknowledgements can be arbitrary, but for the purposes

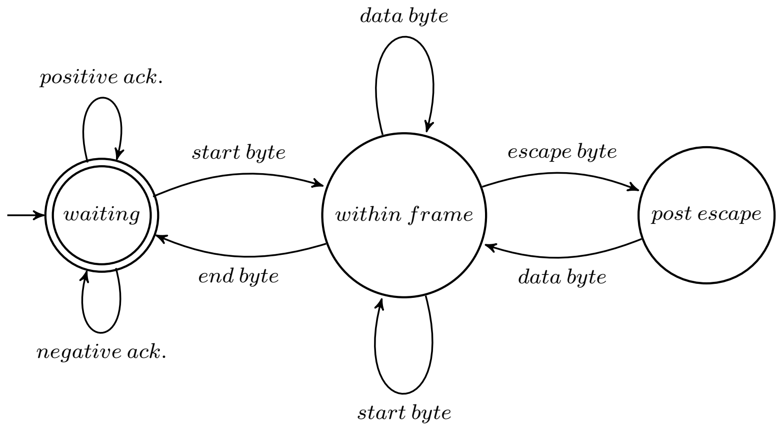


Figure 6.5.: The state machine used to parse serial frames. The machine starts in the *waiting* state until a *start byte* is received, and returns to it once an *end byte* is received. Acknowledgements are only checked for in the waiting state, as one device cannot respond to a frame from another device until its own frame has finished transmitting.

of the HyMod system positive and negative acknowledgements were chosen. A *positive acknowledgement* informs the sender that the data arrived and has been acted upon, whereas a *negative acknowledgement* informs the sender that the data arrived but has not been acted upon for some reason, for example, if it was for an unsupported command. Figure 6.5 shows the state machine used by receiving devices for handling serial frames and acknowledgements.

To make use of serial frame communication with acknowledgements, a handshaking and connection routine was implemented. This routine performs two main functions; it allows for data, such as the relative orientation of two connectors, to be transferred between devices at a level that is transparent to any external code running on a module, and enables communication channels that lack a physical connection state, such as USB and Bluetooth, to have a logical connection state that can be queried. To perform handshaking, three serial frames are used (see Figure 6.6). First a *handshake request* frame is sent from a device to its neighbor. The device then waits a set amount of time for a *handshake data* frame to be returned, containing data about the connection. If a frame is returned, the connection can be considered to be established, otherwise after a time-out the handshake process begins again. To keep a connection established, the device keeps a record of the time data was last received, be it a serial frame or an acknowledgement to a frame that was sent. If no data is received within a given time period (e.g. 5 seconds), a *handshake poke* is transmitted, forcing the neighboring device

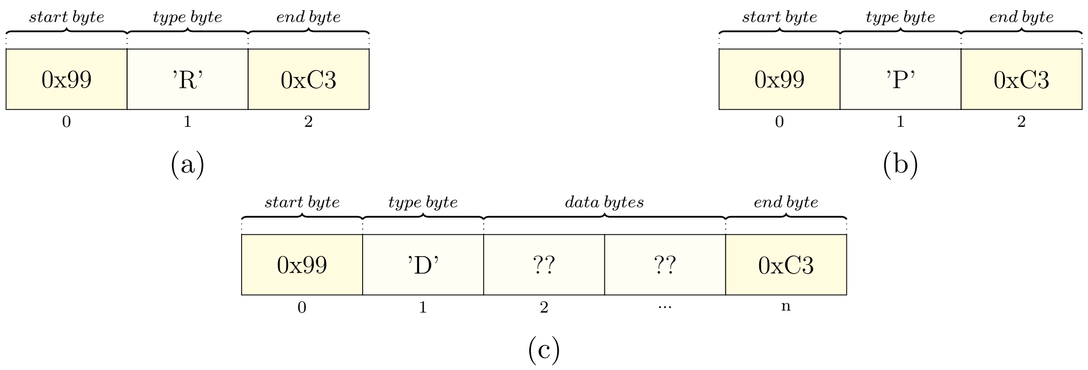


Figure 6.6.: The byte structure of the three serial frames used for handshaking; (a) *handshake request*, (b) *handshake poke*, and (c) *handshake data*. To differentiate handshakes from message and packet serial frames, a different *start byte* value is used.

to issue an acknowledgement to confirm that it is still active. If an acknowledgement is not received, the connection is considered to be broken and the handshake process begins again. The state machine that performs the handshaking and connection routine is shown in Figure 6.7. This routine is used by the HyMod system in all the situations where asynchronous serial communication is used, allowing for the robust transmission of messages and packets between modules in a network and from an external computer to a set of connected modules.

6.2.3. Module Controller Operation

Control of individual HyMod units and extensions is performed by a class called a *ModuleController*. This class is responsible for taking messages and packets received by a module's various communication channels and either acting upon them, or routing them on to other modules as described in Section 6.2.1. To facilitate this behavior, and allow for the control architecture to easily be deployed on multiple module types, module functionality is divided in to two object types that get registered with the *ModuleController*, *Components* and *Communicators*. A component is an element of module functionality that is wished to be read from, written to, or triggered by other modules on the HyMod network. A communicator is an element of functionality that is capable of communicating with other modules in some manner. Note that for consistency in operation, an external computer application is considered to be a module, with its own unique HyMod network address. The code definitions for the component and communicator objects

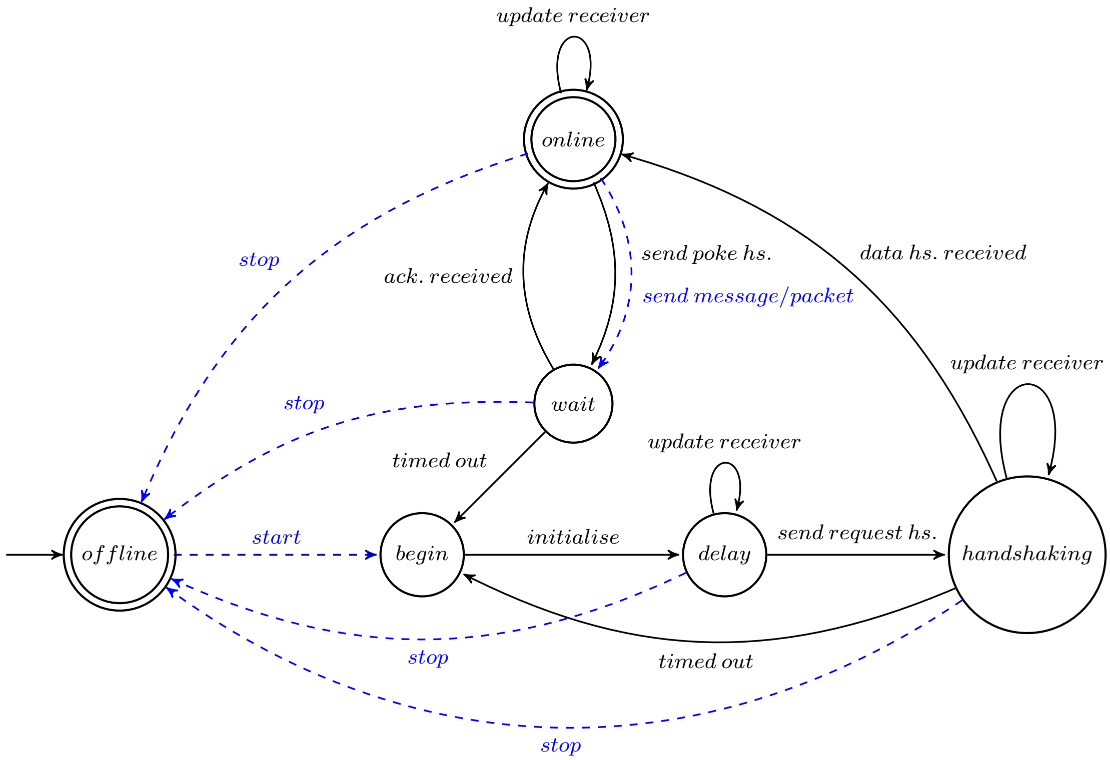


Figure 6.7.: The state machine used to perform handshaking and establish a logical connection between two serial frame devices. Blue dashed lines denote functions that are triggered by external code running on a module.

are shown in Figure 6.8. In this code implementation both the *command* and *response* message types have an equivalent object type. This avoids the need for components to verify a message is a command before acting upon it (as this is performed earlier in the code), and ensures that only responses are returned to the *ModuleController* from its registered components.

There are five component objects implemented by HyMod units; *ModuleMemory*, *JointManager*, *ConnectorManager*, *SensorStick*, and *BluetoothSwitch*. The module memory component gives read access to the internal EEPROM of a module, containing persistent information such as its unique address and type. The joint and connector managers are used to group multiple of their respective objects together, allowing for joints and connectors to referenced by index, rather than requiring that each has its own unique set of message identifiers. The sensor stick component gives read access to the Inertial Measurement Unit within a HyMod unit. Finally, the Bluetooth switch component allows for

```
////////////////////////////////////////////////////////////////////////
class Component
{
public:
    virtual void Initialise(void) = 0;

    //---------------------------------------------------

    virtual bool HandleCommand(const Command& rIn, Response& rOut) = 0;
};

////////////////////////////////////////////////////////////////////////
class Communicator
{
public:
    virtual void Update(void) = 0;

    //---------------------------------------------------

    virtual bool Send(const Packet& rPacket) = 0;
    virtual bool Receive(Packet& rPacketOut) = 0;

    virtual bool Send(const Message& rMessage) = 0;
    virtual bool Receive(Message& rMessageOut) = 0;
};
////////////////////////////////////////////////////////////////////////
```

Figure 6.8.: The C++ definitions for the *component* and *communicator* objects.

the modem of each module to be enabled and disabled, allowing for energy to be saved when wireless communication is not required. For communication, four communicator objects are used, *USBComms*, *BTComms*, *CANComms*, and *HiGenComms*. The USB and Bluetooth communicators are wrappers around a common *SerialComms* class, with different definitions for which serial link is used. The CAN communicator deals with transferring packets across the CAN bus to other modules (messages cannot be sent over CAN as they lack address information). Finally, the HiGen communicator deals with sending and receiving messages and packets across HyMod's internal I^2C communication bus to the *HiGen Controller* boards. Four HiGen communicators are registered with the *ModuleController*, one per connector. Figure 6.9 shows a block diagram of how the components and communicators of a HyMod unit link to the *ModuleController* class.

The main operation of the *ModuleController* is performed by calling the function *Update*. This function is responsible for updating all registered communicators and processing any messages and packets they receive. Processing of messages consists of converting them to commands (if possible) and passing them to each registered component in turn. If the command is handled by a component, a response is created and sent out via the same communicator the original message came in on. If the command is not handled, it gets discarded. Note that if a response message is received, it is ignored by this function

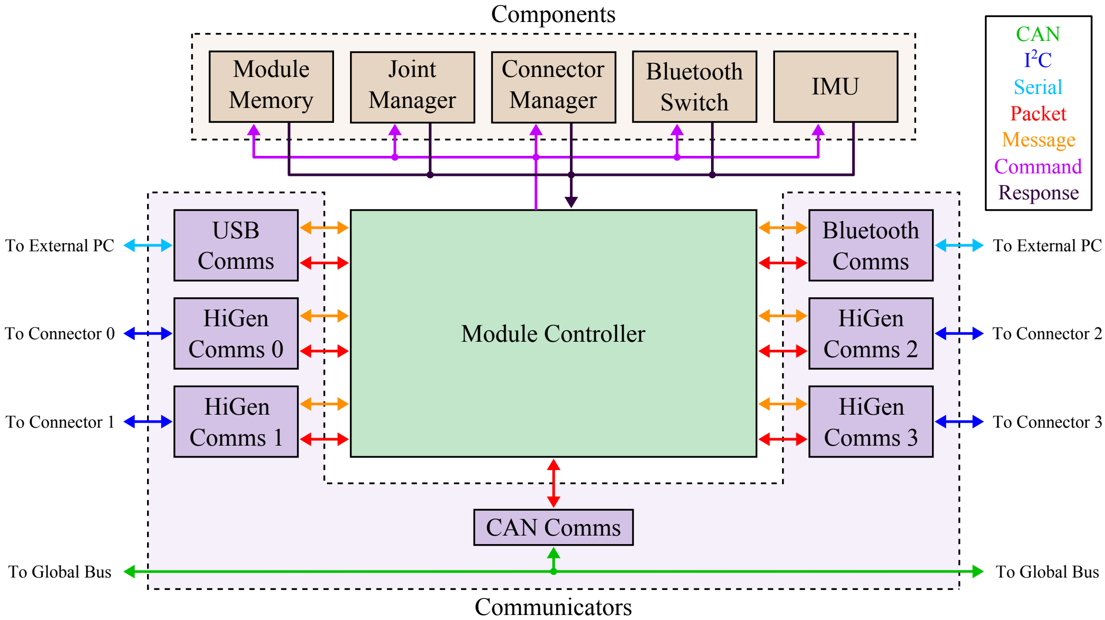


Figure 6.9.: A block diagram of how a *ModuleController* connects to the components and communicators of a single HyMod unit. The arrows indicate the direction data can be exchanged.

so that code external to the *ModuleController* can process it. To send command messages to a neighboring module, the *Send* function on the specific communicator needs to be called directly. Continuing with the *Update* function, processing of packets is performed by first decrementing the jump count of each one received, then checking whether the current module is the intended destination. If the packet is not for the current module, and the jump count has not reached zero, then it is forwarded via all communicators except the one that received it. If the packet is for the current module, then their type is checked, with command packets being dealt with in the same manner as command messages, and response packets being stored for recall by the *WaitForResponseFrom* function. During both these paths, the sequence numbers of the packets are checked to discard any echoes caused by packets taking multiple network routes. Sending of command packets is performed by calling *SendCommandTo* on the *ModuleController*, which forwards them via all registered communicators, unless they are intended for the current module, in which case they are handled like command message. This process of sending and handling messaged and packets allows for the concurrent use of distributed and centralised control of the HyMod system. Pseudocode for *Update*, *SendCommandTo* and *WaitForResponseFrom*, as well as key sub-functions, can be seen in Appendix D.

6.3. Self-Reconfiguration Scenario

Self-reconfiguration is a complex problem for the control of modular robots [130], as on the hardware side it requires the coordination of multiple modules and on the software side it requires algorithms capable of planning the sequence of module motions needed to transform one configuration of modules in to another. Due to the algorithmic complexity of self-reconfiguration, there is yet to be a generic solution to the problem that can be applied to any type of module.

To demonstrate the capabilities of the HyMod control architecture, a self-reconfiguration scenario involving two HyMod units and the Modular Surface extension has been chosen. The two units are arranged in the α_1 configuration (Figure 4.4) and connect to the surface via their end wheel connectors, forming a tower. The objective of the scenario is for the two modules to self-reconfigure from one arbitrary grid position on the surface to another. Although this scenario is relatively simple compared to other demonstrations of self-reconfiguration [37, 16, 17, 18], it is sufficient to show that the HyMod system has the capability to self-reconfigure.

6.3.1. 3D Simulator

Simulations play an important role in the development of reconfigurable modular robots. Not only do they offer the ability to trial module concepts before investing in hardware, they can also demonstrate the capabilities of a system when scaled up to many hundreds of modules, well beyond what may be currently achievable [131] (e.g. due to technology or financial limitations). Examples of simulators for modular robots include CubeInterface for the Molecubes platform [69, 24], and the Unified Simulator for Self-Reconfigurable Robots [132].

To aid in the creation of an algorithm for performing the described self-reconfiguration scenario, an interactive 3D simulator was produced. The simulator is built in C++ and uses the Object-Oriented Graphics Rendering Engine (OGRE) [133] for real-time 3D rendering, and wxWidgets [134] for the graphical user interface. To accommodate the multiple module types of the HyMod system, each is represented in memory as a tree data structure consisting of three object types:

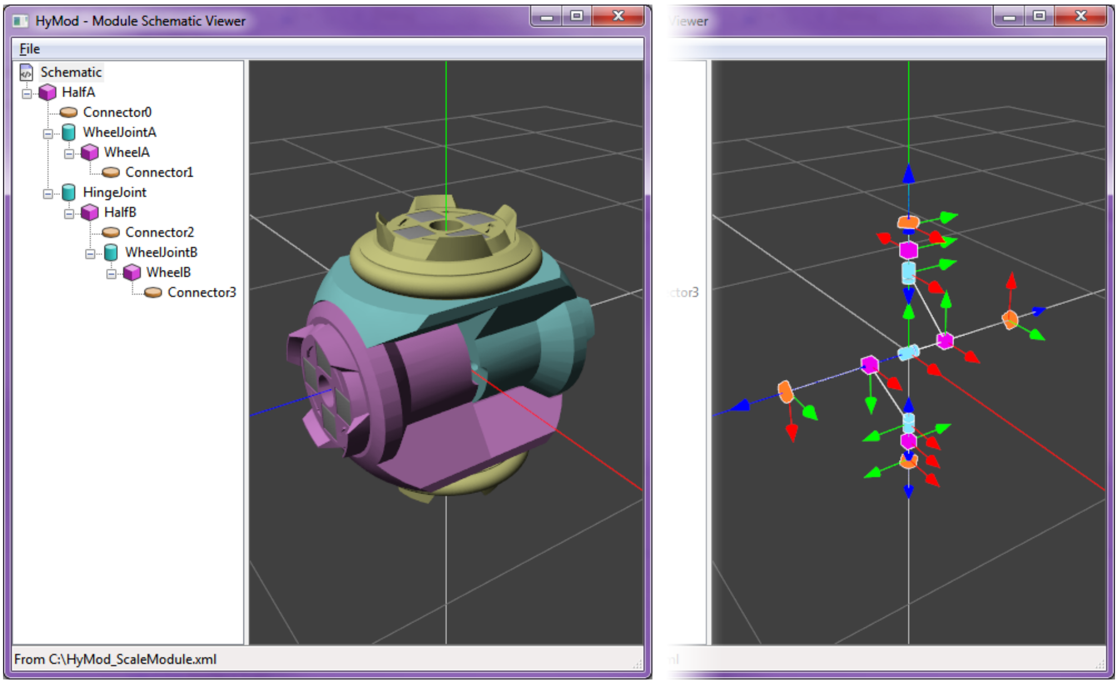


Figure 6.10.: The schematic viewer window for a scale model of the HyMod unit, showing the tree structure and 3D render of the module (left), and the placement of the parts that form the module (right). Magenta cubes, cyan cylinders, and orange discs depict model, actuator, and connector parts, respectively. The X, Y, and Z axes of each part are shown in red, green, and blue, respectively.

- *ModelPart* - A structural element, supporting the assignment of geometric data. Multiple actuator and connector parts can be children of a model part.
- *ActuatorPart* - A degree of freedom between two structural elements of a module. Only a single model part can be a child of an actuator part.
- *ConnectorPart* - A connection mechanism that is used to attach to other module's connection mechanisms. No object can be a child of a connector part.

These object types form a *Schematic*, that can either be hard-coded or loaded in from a file. The file format used for module schematics is the eXtensible Markup Language (XML), as its structure of tags matches that of a tree. Loaded or generated schematics can be opened in a viewer window to confirm their tree structure and object placement is as intended. The viewer window of a scale model of HyMod (Section 4.3.1.1) can be seen in Figure 6.10, and the XML file it was loaded from can be seen in Figure 6.11.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Schematic name="HyMod">
  <ModelPart name="HalfA" file="SimScaleModule_HalfA.mesh" z="-0.08">

    <ConnectorPart name="Connector0" type="genderless" symmetry="4" z="-0.2325" yaw="180" roll="90">
    </ConnectorPart>

    <ActuatorPart name="WheelJointA" type="continuous" y="0.15" pitch="-90"  values="0,90,180,270">
      <ModelPart name="WheelA" file="SimScaleModule_Wheel.mesh" yaw="180" y="0.19" pitch="90">
        <ConnectorPart name="Connector1" type="genderless" symmetry="4" y="0.2325" pitch="-90">
        </ConnectorPart>
      </ModelPart>
    </ActuatorPart>

    <ActuatorPart name="HingeJoint" type="angular" rangeMin="-90" rangeMax="90" values="-90,0,90">
      <ModelPart name="HalfB" file="SimScaleModule_HalfB.mesh" z="0.08" pitch="180">

        <ConnectorPart name="Connector2" type="genderless" symmetry="4" z="0.2325" roll="-90">
        </ConnectorPart>

        <ActuatorPart name="WheelJointB" type="continuous" y="-0.15" pitch="90" values="0,90,180,270">
          <ModelPart name="WheelB" file="SimScaleModule_Wheel.mesh" y="-0.19" pitch="-90">
            <ConnectorPart name="Connector3" type="genderless" symmetry="4" y="-0.2325" pitch="90">
            </ConnectorPart>
          </ModelPart>
        </ActuatorPart>

      </ModelPart>
    </ActuatorPart>

  </ModelPart>
</Schematic>
```

Figure 6.11.: The XML file used to generate a HyMod scale model in the 3D simulator. All part types have the following common attributes: name, x, y, z, yaw, pitch, roll. Default values are assigned to any missing part attributes.

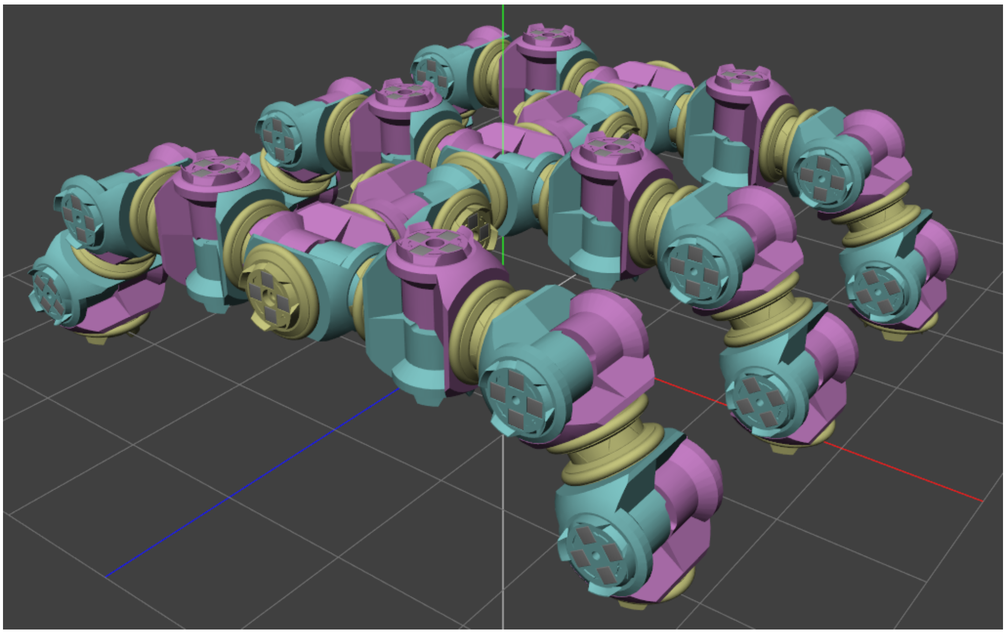


Figure 6.12.: A 3D render from the simulator of a configuration of 23 HyMod scale models forming a hexapod. Three modules are used for each leg, and five for the spine.

To perform simulations with multiple connected modules, a graph data structure is used, with modules being vertices and connections between modules being edges. Each vertex contains a module schematic reference, current actuator values, and matrix information for distance checking and rendering purposes. Each edge contains details of the specific modules and connectors they are linking together. By using this representation, any connected structure of modules can be created. As an example, Figure 6.12 shows a hexapod configuration composed of 23 HyMod scale models.

6.3.2. Surface Traversal Algorithm

To traverse a tower of two HyMod units in an α_1 configuration from one arbitrary position on a Modular Surface extension to another requires the tower to perform a number of self-reconfiguration steps. The number of steps is dependent on the relative difference between the x and y coordinates of the two positions (their Manhattan distance), and can be calculated using the following equation,

$$S = abs(X_a - X_b) + abs(Y_a - Y_b) \tag{6.1}$$

Table 6.1.: The state sequence performed to translate a two HyMod unit tower in a (a) forward and (b) reverse direction along a Modular Surface. T_0 and T_1 denote towers with unit 0 and 1 at the top, respectively. $\downarrow$ and $\uparrow$ represent lowering and raising, respectively.

State	T_0	$\downarrow$	L	$\uparrow$	T_1	$\downarrow$	L	$\uparrow$	T_0
Hinge Angle	0°	$\rightarrow$	+90°	$\leftarrow$	0°	$\leftarrow$	-90°	$\rightarrow$	0°
Top Module	0	0	-	1	1	1	-	0	0

(a)

State	T_0	$\downarrow$	L	$\uparrow$	T_1	$\downarrow$	L	$\uparrow$	T_0
Hinge Angle	0°	$\leftarrow$	-90°	$\rightarrow$	0°	$\rightarrow$	+90°	$\leftarrow$	0°
Top Module	0	0	-	1	1	1	-	0	0

(b)

where S is the number of steps, X_a and Y_a are the coordinates of position a, and X_b and Y_b are the coordinates of position b. Each step moves the tower closer to the goal position by translating it into an adjacent grid position along whichever axis it is facing. To change axis, a rotation of the wheel connector attached to the surface is required between steps.

Self-reconfiguration steps involve four operations; lowering the tower to form a line, connecting the line to the adjacent grid position, disconnect the line from the current grid position, and raising the line to form a tower. Performing this sequence of operations reverses the orientation of the tower, requiring joint motions be inverted and module references switched in order for the tower to continue moving in the same direction. For example, a tower starting with a HyMod unit labelled 0 on top (T_0), can transform to a tower with a unit labelled 1 on top (T_1) by actuating the joints of both units to +90 or -90 degrees to form a line, then reversing that motion. Table 6.1 shows the corresponding joint angles and top modules for a tower moving forward and backward along a Modular Surface. Note how the hinges of the modules only rotate in a single direction whilst a given HyMod unit is on top.

To implement the surface traversal algorithm in simulation, the simulator was modified to create the required configuration of tower and surface, and specify random goal locations for the tower to move to. Movement of the tower is performed using the state machine shown in Figure 6.13, with lower, disconnect and turn commands being issued by the

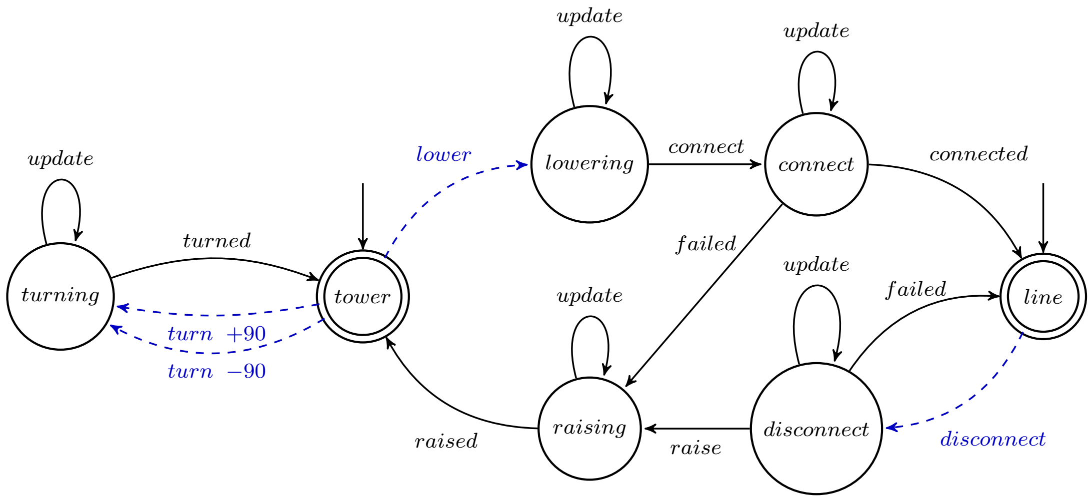


Figure 6.13.: The state machine used to control the self-reconfiguration of two connected HyMod units in the α_1 configuration (Figure 4.4) across a Modular Surface extension. Blue dashed lines denote functions that are triggered by the traversal algorithm.

algorithm based on the tower's current position and orientation. Inversion of the tower is accounted for with variables rather than separate states. This state machine has been designed to work for controlling both the simulation and the physical system via the control architecture, hence the failure paths. Snapshots of the simulated HyMod tower performing surface traversal on a 5 x 5 Modular Surface can be seen in Figure 6.14.

6.3.3. Experimental Setup

To illustrate the applicability of the surface traversal algorithm, the experimental setup shown in Figure 6.15 was assembled. The setup consists of two HyMod units and the 4 x 3 Surface extension. The HyMod units are tethered to a bench power supply set to 8.4 V (replicating the maximum battery voltage), and the Surface extension is powered by a 9 V DC wall adapter. A USB cable is connected between the Surface and an external computer to allow for centralised control of the setup.

Due to time constraints on the project, work on integrating the surface traversal algorithm with the experimental setup will be the focus of future research efforts.

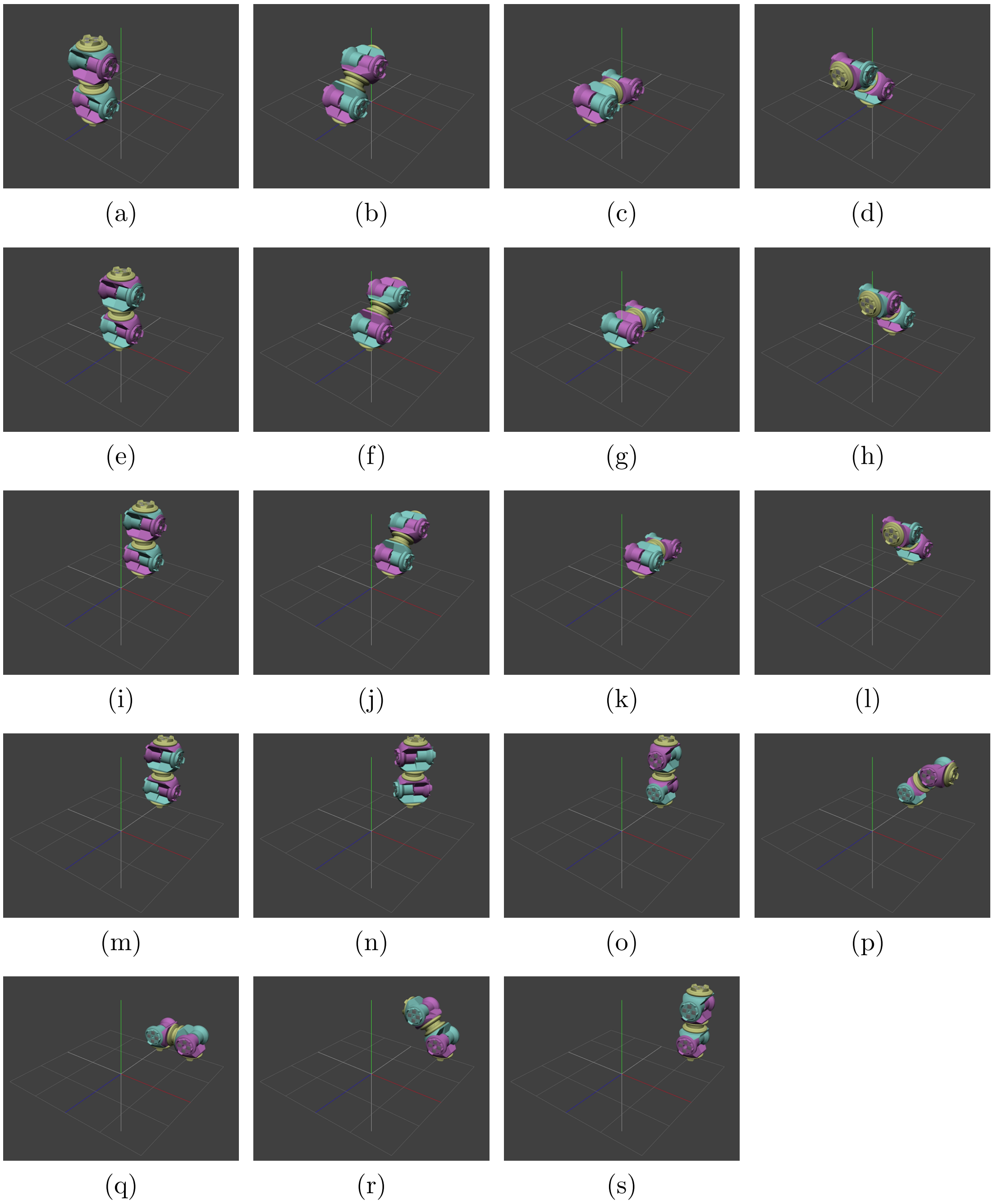
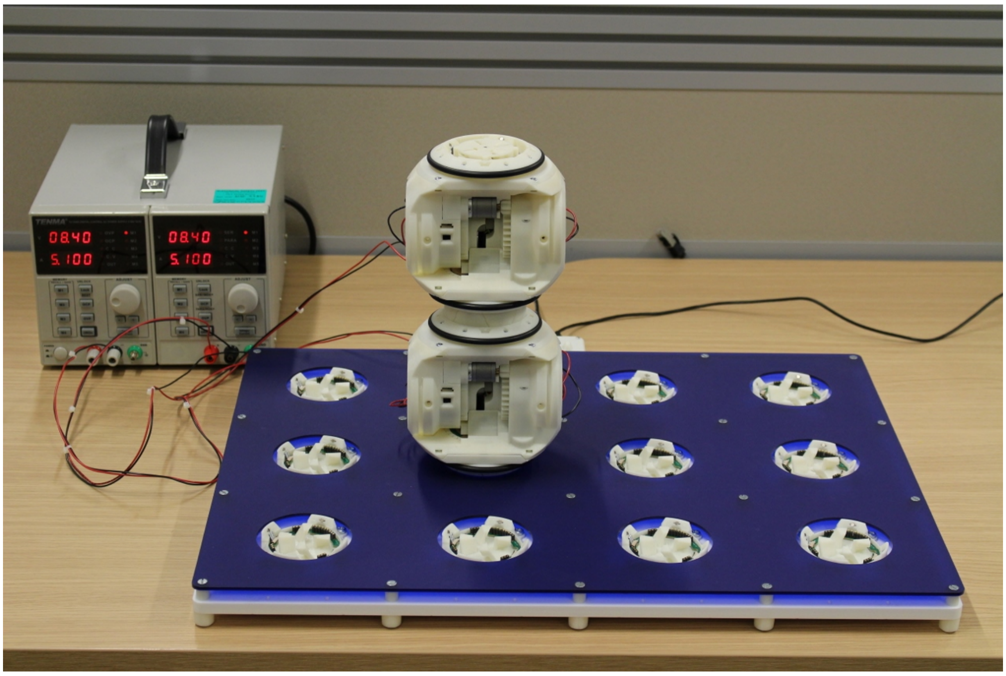
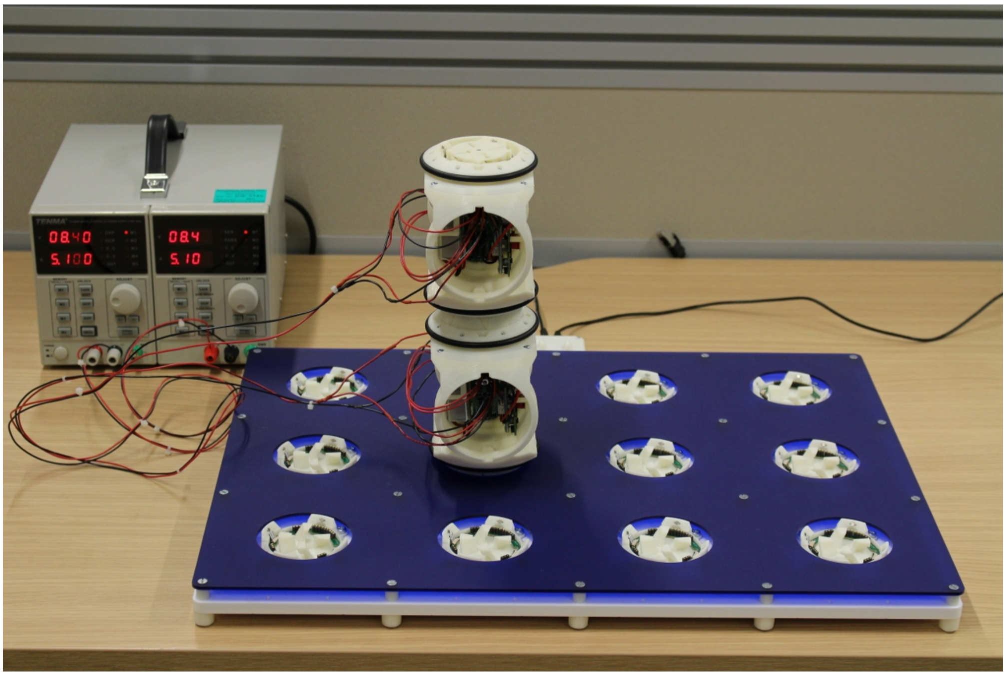


Figure 6.14.: Snapshots of a two HyMod unit tower self-reconfiguring from an arbitrary starting position of (0, 1) to an arbitrary goal position of (1, -2) on a 5 x 5 Modular Surface, in simulation. Joints are moved in 45° increments.



(a)



(b)

Figure 6.15.: The experimental setup that will be used for future demonstrations of the surface traversal algorithm, with the HyMod tower oriented towards the (a) front, and (b) side.

6.4. Summary

This chapter presented a novel software architecture for the control of sets of connected HyMod units and extensions. The architecture allows for the concurrent use of centralized and distributed module control strategies, via the use of a custom message routing protocol that works across HyMod's network of communication channels, forwarding messages between modules when necessary. Each module type implements a standardized controller for dealing with the protocol, and adapts it to expose their specific functionality to the network. This approach offers an advantage over previous approaches to module control, by allowing modules with varying capabilities to co-exist in the same network.

To aid in the deployment of the messaging protocol, a frame-based communication protocol for asynchronous serial was developed, for use between modules and from modules to external computers. This protocol allows for messages to be transmitted as contiguous blocks, with acknowledgements returned to confirm they were received. These are used as a basis for a custom handshaking routine, allowing serial devices to not only know if their neighbor is operational, but also allow for essential information such as identifiers and orientation to be exchanged during an initial connection phase. The frame-based serial protocol and handshaking routing enable robust transmission of messages between modules in a network.

To demonstrate the control of multiple modules, a self-reconfiguration scenario was presented, consisting of two HyMod units in a tower formation traversing a Modular Surface extension. An algorithm for the traversal process was devised that takes advantage of the novel joint and connector arrangement of HyMod, and a novel 3D simulator created to verify the algorithm's effectiveness. The simulator allows for multiple module types to be defined in files and loaded in to create a connected structure that can be controlled by the algorithm in real-time. Finally, an experimental setup for recreating the results of the simulator on the physical system was shown.

7. Conclusion

This thesis presented the design, implementation and study of a new self-reconfigurable modular robotic system for use as a research and education platform, called HyMod. The HyMod system is build using off-the-shelf and easily acquirable bespoke components, such as printed circuit boards, and makes extensive use of 3D printing technology. The system is intended for experiments relating to self-assembly, self-reconfiguration, and collective locomotion, as well as for demonstrating tasks relevant to the areas of reconfigurable manufacturing, search and rescue, and space exploration. In addition, the HyMod system improves upon the state of the art in the areas of mobility, self-reconfigurability, and extendibility. This is achieved by a combination of a high-speed genderless connection mechanism (HiGen), a hybrid mobile and self-reconfigurable robotic module (HyMod), a hardware and electronics framework for extension modules, and a software architecture for the centralized and distributed control of modules.

The HiGen connector is four times symmetric and is capable of joining with other HiGen connectors in a manner that allows either side to disconnect in the event of failure. The mechanism is capable of extending out of and retracting in to its housing, allowing for electrical connections to be made and broken, as well as clearance to be created between two neighboring connectors. The electrical contacts allow for the concurrent use of local and global communication protocols, as well as power sharing techniques. Additionally, the relative orientation of connectors can be determined based on the unique pattern of connections formed at each symmetry interval. Experiments were conducted with a pair of connectors, testing their actuation speed, electrical connectivity, connection repeatability, and load capacity. Actuation results showed that the mechanism is able to actuate in under 0.3 s, a speed that is, to our knowledge, an order of magnitude faster than existing mechanical genderless connection mechanisms that feature single-sided disconnect.

HyMod is a new robotic module that is a hybrid between chain, lattice and mobile self-reconfigurable modular robots. Its hybrid nature is achieved using a three degrees of freedom spherical joint and four HiGen connectors, enabling it to not only rotate freely in place within a cubic lattice position, but also act as a differential wheel setup for individual mobility when away from a connected structure. To our knowledge, HyMod is the first robotic module to combine these two capabilities. An analysis of the module is presented, detailing its symmetry, ability to form configurations and self-reconfigure between them, as well as the clearance considerations for free in place rotation. The mechanical and electrical properties of the completed module are also presented. Experiments were conducted with a single HyMod unit, testing its driving and lifting capabilities. Results show that the module is able to drive at $0.1\,\mathrm{m\ s^{-1}}$ and can lift a weight equivalent to 1.8 modules in-line.

A framework for extension modules has been devised for HyMod. Extensions introduce functionality to a modular robot that its modules may lack or could be impractical to replicate with a collection of modules. This framework accelerates the creation process of specialized or task-specific modules for the HyMod system, by providing a passive variant of the HiGen connector, a pick-up location template, and internal electronics. Four extension module types have been built using the framework; a Gripper extension, a Mecanum Wheel extension, a Camera extension, and a Modular Surface extension. Details of each extension are given, as well as which aspects of the framework each of them adopts.

A software architecture has been implemented for the HyMod system. The architecture allows for the concurrent use of both centralized and distributed module control strategies, and is built around the concept of message routing, enabling information to be exchanged between modules in a transparent manner. Details of the message structures and routing procedures of the architecture are given, as well as the logic for reliable communication between neighboring modules. A self-reconfiguration scenario involving two HyMod units and the Modular Surface extension was proposed, and demonstrated in simulation.

We believe that the HyMod system has the potential to advance the field of self-reconfigurable modular robotics, by being a platform for research in to modular robotics that others can adopt, and highlighting modular robot functionality that future systems should aim to incorporate, such as high-speed genderless connection, free in place rotation, and independent module mobility.

7.1. Discussion

With the current iteration of the HyMod system being complete, now presents an opportunity to discuss how successful it is in achieving the requirements derived from the original objectives. The discussion is divided into achievements and limitations, with a vision for the future of the field also presented.

7.1.1. System Achievements

Beyond the successful creation of a self-reconfigurable modular robot, there are a number of achievements of the HyMod system that are worth a specific mention:

- ***Connection Mechanism*** - Developed HiGen, a new genderless mechanical connection mechanism for modular robots. The connector incorporates many useful features from past designs such as identical operation (requirement C.1.3), four times symmetry (requirement C.1.4), no power required to maintain a connection (requirement C.1.5), and electrical pathways (requirement C.2.4). The main advantage of HiGen over other genderless designs is its high-speed actuation, which not only surpassed performance requirement C.4.2 by a factor of four, is also an order of magnitude faster than other genderless designs. This speed has the potential to significantly reduce the self-reconfiguration time of any modular robots that include HiGen. Another advantage of the connector is its ability to fully separate from a neighboring connector (requirement C.1.6), as not only does this disconnect the multiple electrical pathways of the connector, it also creates clearance between connectors that is exploited by HyMod units to allow for free in place rotation. This is a beneficial ability for modular robots that relatively few systems have incorporated thus far. On the electrical side, HiGen is able to detect its actuation state (requirement C.2.3a), as well as the presence of a neighboring connector (requirement C.2.3b) and its relative orientation (requirement C.2.3c). These latter two pieces of information are useful for module configuration discovery.
- ***Connector Reliability*** - Of all the parts of the HyMod system, HiGen has proved to be the most reliable, having its actuation be formally tested in over 200 trials and informally tested hundreds of times more during the development of HyMod and its extension. The only time actuation fails is when the misalignment between

connectors is greater than its designed limits (requirements C.4.3 and C.4.4), which the mechanism can detect and terminate the actuation to prevent damage (requirement C.5.2).

- ***Hybrid Module*** - Developed HyMod, a new self-reconfigurable modular robot capable of independent locomotion. The module features four genderless connection mechanisms, allowing it to form both chain and lattice structures (requirement M.1.2). To make the module a hybrid, wheel-based locomotion was introduced (requirement M.1.6) by having the module feature three rotational degrees of freedom arranged in to a spherical joint. This arrangement achieves independent locomotion without introducing dedicated drive mechanics that would otherwise add additional complexity and weight to the module, as well as benefits self-reconfiguration when combined with free in place rotation (requirement M.1.5). In terms of electronics, HyMod incorporates various features of past successful systems, most notably, neighbor-to-neighbor and network-based communication, wireless communication, and power sharing (requirements M.2.4, M.2.5, and M.2.6, respectively). It also integrates orientation and environmental sensing (requirement M.2.3). This is all achieved whilst keeping HyMod below a 1 kg target weight (requirement M.4.1).

- ***Module Compactness*** - Given the design chosen for HyMod, the functionality and dimensional requirements imposed on it (M.4.2), as well as the wish for it to be composed primarily of off-the-shelf components, HyMod is relatively compact. Although HyMod lies on the larger end of the modular robot size spectrum, to make it smaller whilst retaining all of the included functionality would have required the introduction of specialist or bespoke components, increasing the overall cost of the project. Reducing the module's size could perhaps be a research focus for a second iteration of HyMod, if one is developed.

- ***System Extendibility*** - Developed four extension modules for the HyMod system. These extension modules benefit from the consideration that was made for how HiGen could be adapted to a passive design (requirement C.1.7), allowing for them to be tools that are connected to rather than full-featured modules. This consideration influenced the requirements for extensions, such as requiring passive connectors (requirement E.1.2), being powered through their connector (E.2.1), and being able to reside in pick-up locations (requirement E.1.3). It also resulted in a framework for extensions being developed, as the common functionality, such as driving actuators or reading sensors (requirement E.2.2a), could be designed once and incorporated in to multiple extension types faster than would otherwise be possible. As extension modules are based on the HiGen design, they too have proven to be reliable for hundreds of repeated actuations (requirement E.5.1).

- ***Control System*** - Developed a control system that successfully supports both the centralized and distributed control of a set of connected HyMod units and extensions. The architecture created allows for easy adaptation to the varied functionality of these modules, and can be expanded to new module types in the future.

7.1.2. System Limitations

Despite the achievements of the HyMod system, there are a few limitations of the current iteration that should be resolved by any future iterations:

- ***Ease of Assembly*** - No explicit priority was given to the complexity of the assembly process of the system, meaning that HyMod units are difficult and time consuming to put together and require the intricate manoeuvring of components. This is due to there being no requirement for ease of assembly imposed on the system, as well as the component dimension constraints specified by module requirement M.4.2. As such, to incorporate all of the module's required functionality some assembly steps became more complicated than originally intended. In hindsight, ease of assembly should have been a requirement of the system from the outset, with a threshold for complexity to compare against and trigger design revisions for assembly steps that exceeded it.

- ***Power Sharing*** - Although HyMod units feature a power supply that allows for them to be powered by a shared bus between modules, as per requirement M.2.6, there are caveats to the final implementation that limit the usefulness of this functionality. Specifically, only the 5 V regulator that powers the module's electronics and the HiGen connector motors are connected to this shared bus, not the motors that drive HyMod's three joints. The reason for this is because the amount of current these motors draw under load exceeds the maximum supported by the components of the HiGen connectors, such as the ribbon cables joining the control board to the contact ring. As the HiGen connector was developed before HyMod, it was unknown what the final power draw of the system would be. Rather than redesign the HiGen connector to accommodate this higher power draw, which would have been a significant time investment at the stage when this was discovered, it was opted to design the power system with the joint motors and associated drivers excluded from the shared bus. This means that less active modules in a configuration are unable to assist other modules with movement activities, potentially limiting the duration of any self-reconfiguration or collective locomotion experiments that are wished to be performed with the modules.

- ***Battery Charging*** - Even though the requirement for battery charging within a module was achieved (M.2.6d), the solution adopted fell short of the original intention. This intention was for each HyMod unit to incorporate a charging circuit that would connect to the shared power bus so that one could be charged by other units or specialised charging station modules. Unfortunately, after researching the electronics required to implement charging circuitry for the lithium polymer batteries, along with potential safety concerns, it was deemed unsuitable to adopt this solution at the time. Instead, ports were added to the side of HyMod units to allow for commercially available charging units to be connected, which still satisfies the requirement but lacks the functionality benefits of the original planned solution.

- ***Operation Time*** - Due to delays in the project caused by the complexities of hardware development, experiments have yet to be conducted on untethered HyMod units. As such, their operation time under various use cases has yet to be tested, meaning that it has not been possible to verify module requirement M.4.4. Unfortunately, it is expected that the 30 minute goal will not be achieved. This is because HyMod was originally intended to feature two 1800 mAh batteries (one in either side of the module) but during early assembly tests it was discovered that the module's wiring occupied more space than was anticipated, and it was too late to increase the module size to resolve it. This led to smaller batteries being sourced, both in size and capacity, whilst still matching the performance properties of the originals. In hindsight, more internal space should have been reserved for the battery compartments in HyMod, or the module should have been increased in size, so that the original batteries could have been retained.

- ***Lifting Capability*** - As highlighted by the experiments conducted with a single tethered HyMod unit, the goal of lifting at least two modules in-line with its main joint (M.4.3) was not achieved. The power system, joint motors, and motor drivers within HyMod were selected so that one module would be able to lift two other modules; however, it was not anticipated that the material properties of the 3D printed plastic components in the connector and on the lifting mechanism would be the limiting factor. This can be attributed in part to the lifting capability of the HiGen connector not being tested beyond the requirement (C.4.5), meaning that the material limitations were not discovered as early as they perhaps should have been so that alternatives could be sourced.

- ***System Reliability*** - Due to project delays it has not been possible to assess the reliability of the various aspects of HyMod units beyond the HiGen connector. Specifically, no experiments have been conducted to determine the reliability (M.5.1) and repeatability (M.5.2) of the module's joints, beyond that observed during the lifting and driving tests. These tests should be conducted in the future as their outcome could impact the ability for HyMod units to successfully self-reconfigure and may require design revisions to resolve.

7.1.3. Research Vision

From undertaking this research, it is observed that modular robotics is now reaching a point where many of the technical challenges involved in the hardware itself have been addressed to varying extents, with new research improving upon past designs. As such, the technology is at a level of maturity where it is already being used as educational platforms and can begin to be considered for use in the various application areas of the field. Because of this, it is anticipated that research over the next decade will focus more on domain specific challenges rather than platform specific challenges, with researchers adopting a number of standard research platforms to accelerate this effort.

In the decades beyond it is foreseeable that the field will reach the goal of creating modular robotic systems that act as tools for performing virtually any task; however, it is not expected that a single modular robotic system will emerge. Instead there will likely be several classes of systems, based on the scales of the tasks to be performed as well as the environments they are being performed in. It could be possible that during this time technologies such as rapid manufacturing may advance to the point where specialized robotic tools can be made to perform the tasks modular robots would otherwise be assembled to perform, but ultimately the adaptability and self-reconfigurability of modular systems will remain desirable properties for many application areas.

7.2. Future Work

With the modular robotic system presented in this thesis, there are several areas where improvements and future work can be performed. In the immediate term, the intention for this project is to open-source all of the hardware design files and program source files used to create the HyMod system. It is hoped that by releasing these files, other researchers will examine and learn from the work produced here, and maybe even adopt the HiGen connector and HyMod unit for their future research with modular robotics.

On the side of improvements, and as mentioned in the discussion section, although the motors and power system of HyMod are theoretically capable of lifting greater than two modules in-line, it has not been possible to verify this as the plastic 3D printed hinge joint gears and the HiGen connector's docking hooks are not capable of handling more than 1.8 modules in-line without failing. With additional time and resources, alternative manufacturing methods could be explored, such as injection moulding, or metal 3D printing, to make these components and other critical parts stronger.

One main area for future research would be on the algorithmic side of self-reconfiguration. Despite the HyMod system being capable of self-reconfiguration and an analysis having been performed, to truly unlock its potential, algorithms need to be developed that take advantage of this knowledge to allow for many HyMod units (both in simulation and reality) to self-reconfigure between different structures autonomously. This would allow for the benefits of free in place rotation to be further demonstrated, by having a shape be formed within a structure of modules acting as a scaffold, for example.

Another area for future research would be on increased task complexity. With the use of the Modular Surface, Gripper and Camera extensions, it should be possible to implement algorithms that allow for a robot arm constructed of modules to pick up an object from one arbitrary location to another, by recognising the shape, and therefore how to grasp the object, for example. This and other tasks could be explored.

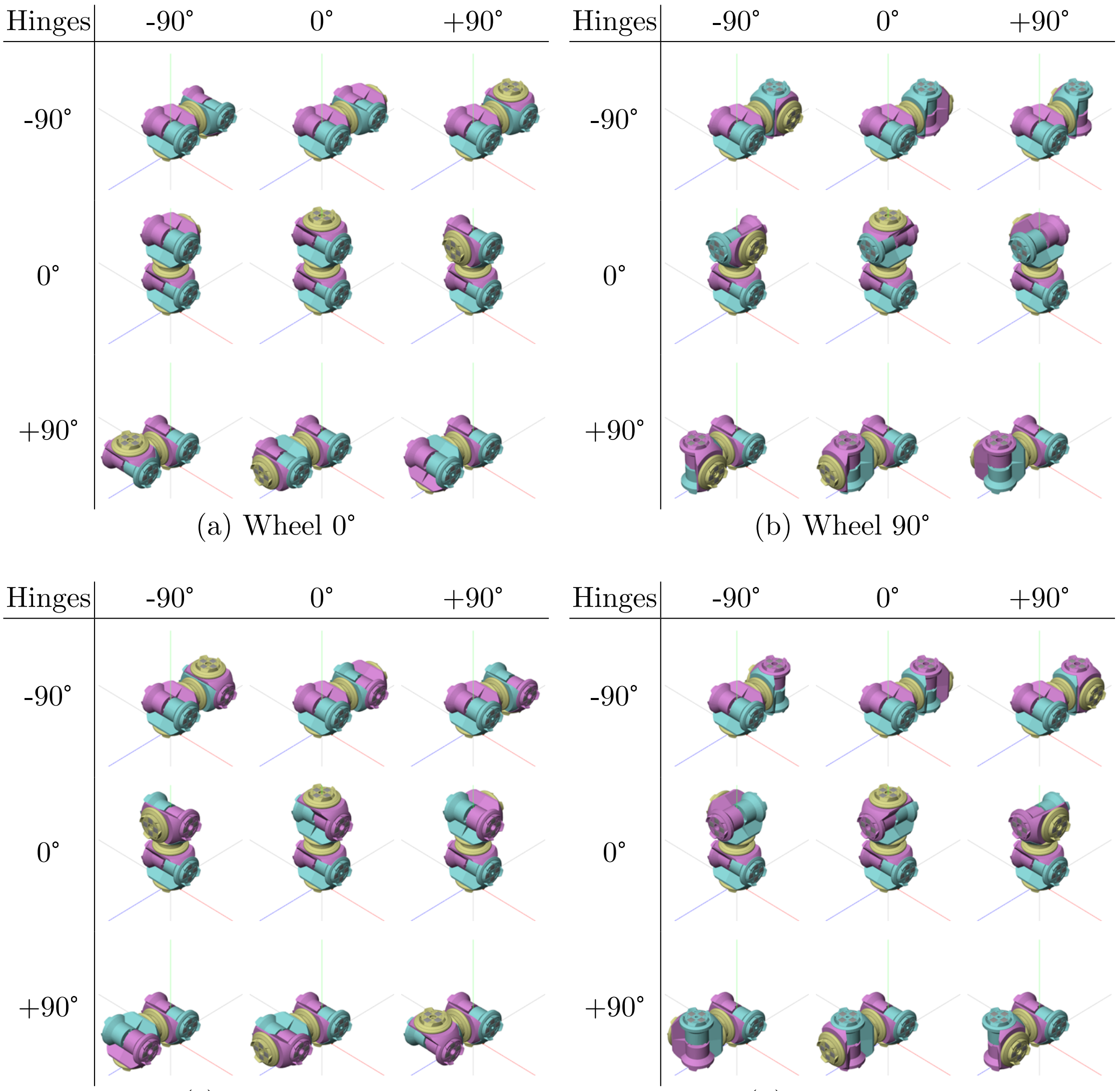
A final area for future research is that of fault tolerance. The system features a connector capable of single-sided disconnect, but so far this ability has only been of benefit for interfacing with extension modules. By exploring fault tolerance, algorithms could be developed that allow for a set of connected modules to adapt to failure, either by self-reconfiguring or discarding failed modules and changing their operation procedure. Both avenues could demonstrate the capability for the HyMod system to continue performing tasks whilst gracefully responding to damage, for example.

Appendices

A. Two-Module Motion Capabilities

This appendix presents 3D renders of all the possible quantized joint angle combinations (90 degree increments) for the two-module configurations presented in Section 4.3.1, Figure 4.4. Note, rotations of wheels not connected to another module are discounted from the combinations, as they can be cancelled out by connector symmetry. Similarly, the configurations with two wheels connected together are considered as a single joint, as their rotational degrees of freedom are in-line, reducing the number of configurations. For these renders, model files from a 3D printable scale module of a HyMod unit were used (Section 4.3.1.1).

Table A.1.: The quantized joint angle combinations for the α_1 configuration of two modules. In total there are 36 combinations, resulting from there being two hinge joints and one wheel joint, giving 3 and 4 angles, respectively.



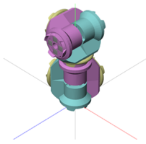
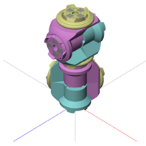
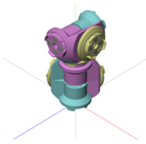
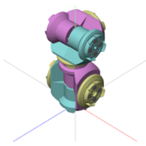
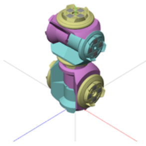
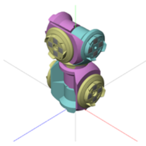
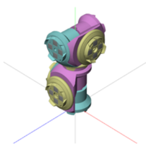
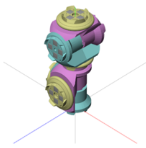
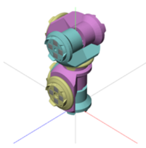
(a) Wheel 0°

(b) Wheel 90°

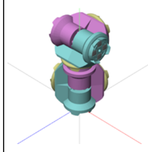
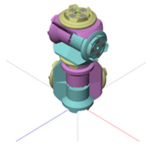
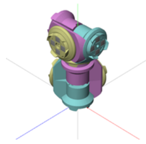
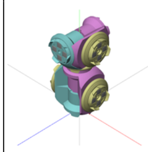
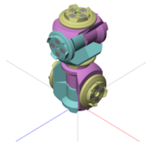
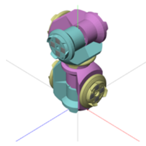
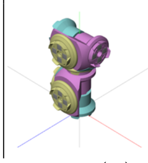
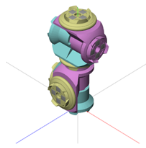
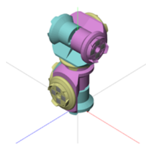
(c) Wheel 180°

(d) Wheel 270°

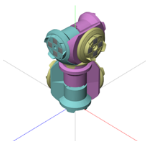
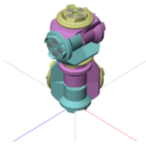
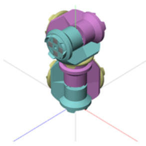
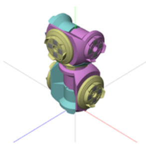
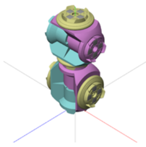
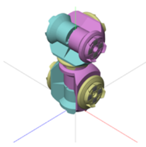
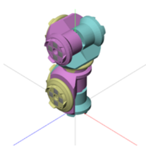
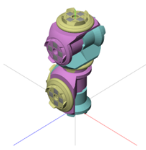
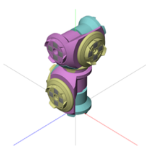
Table A.2.: The quantized joint angle combinations for the α_2 configuration of two modules. In total there are 36 combinations, resulting from there being two hinge joints and one wheel joint, giving 3 and 4 angles, respectively.

Hinges	-90°	0°	+90°
-90°			
0°			
+90°			

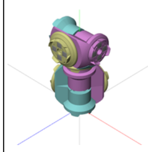
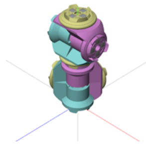
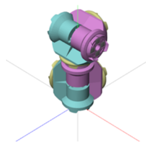
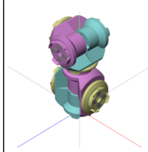
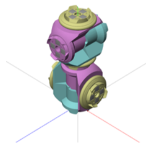
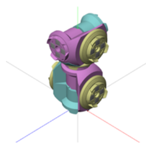
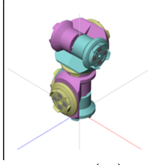
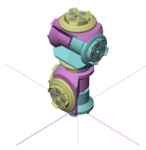
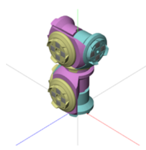
(a) Wheel 0°

Hinges	-90°	0°	+90°
-90°			
0°			
+90°			

(b) Wheel 90°

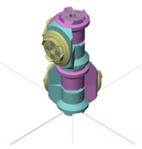
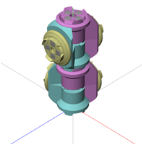
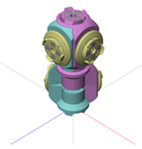
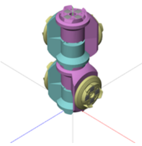
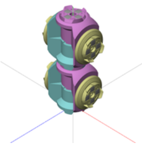
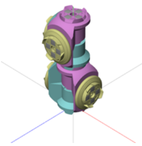
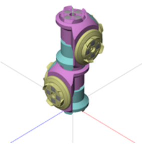
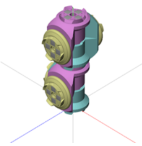
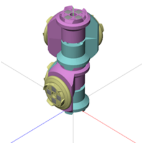
Hinges	-90°	0°	+90°
-90°			
0°			
+90°			

(c) Wheel 180°

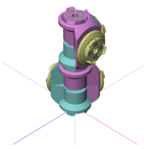
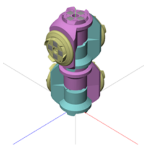
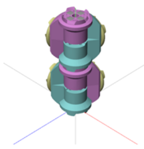
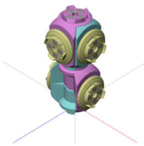
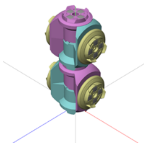
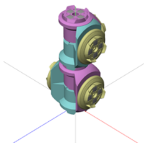
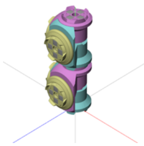
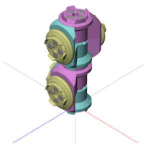
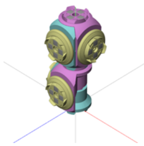
Hinges	-90°	0°	+90°
-90°			
0°			
+90°			

(d) Wheel 270°

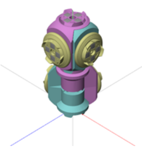
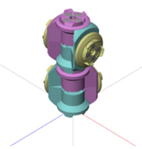
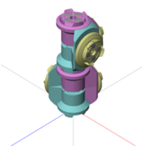
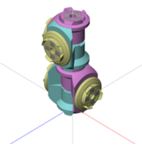
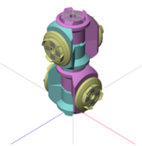
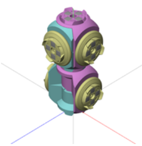
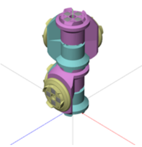
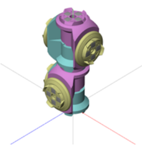
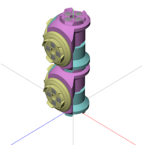
Table A.3.: The quantized joint angle combinations for the remaining configurations of two modules. In total each configuration has 9 combinations, resulting from there only being two hinge joints, giving 3 angles each.

Hinges	-90°	0°	+90°
-90°			
0°			
+90°			

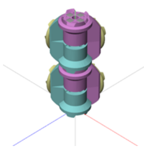
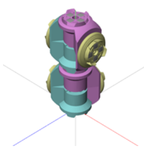
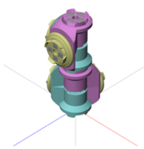
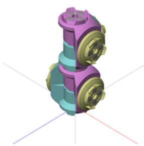
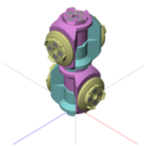
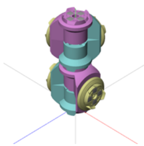
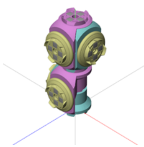
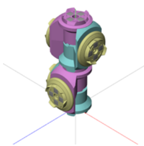
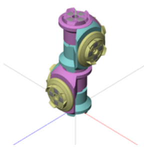
(a) β_1 Configuration

Hinges	-90°	0°	+90°
-90°			
0°			
+90°			

(b) β_2 Configuration

Hinges	-90°	0°	+90°
-90°			
0°			
+90°			

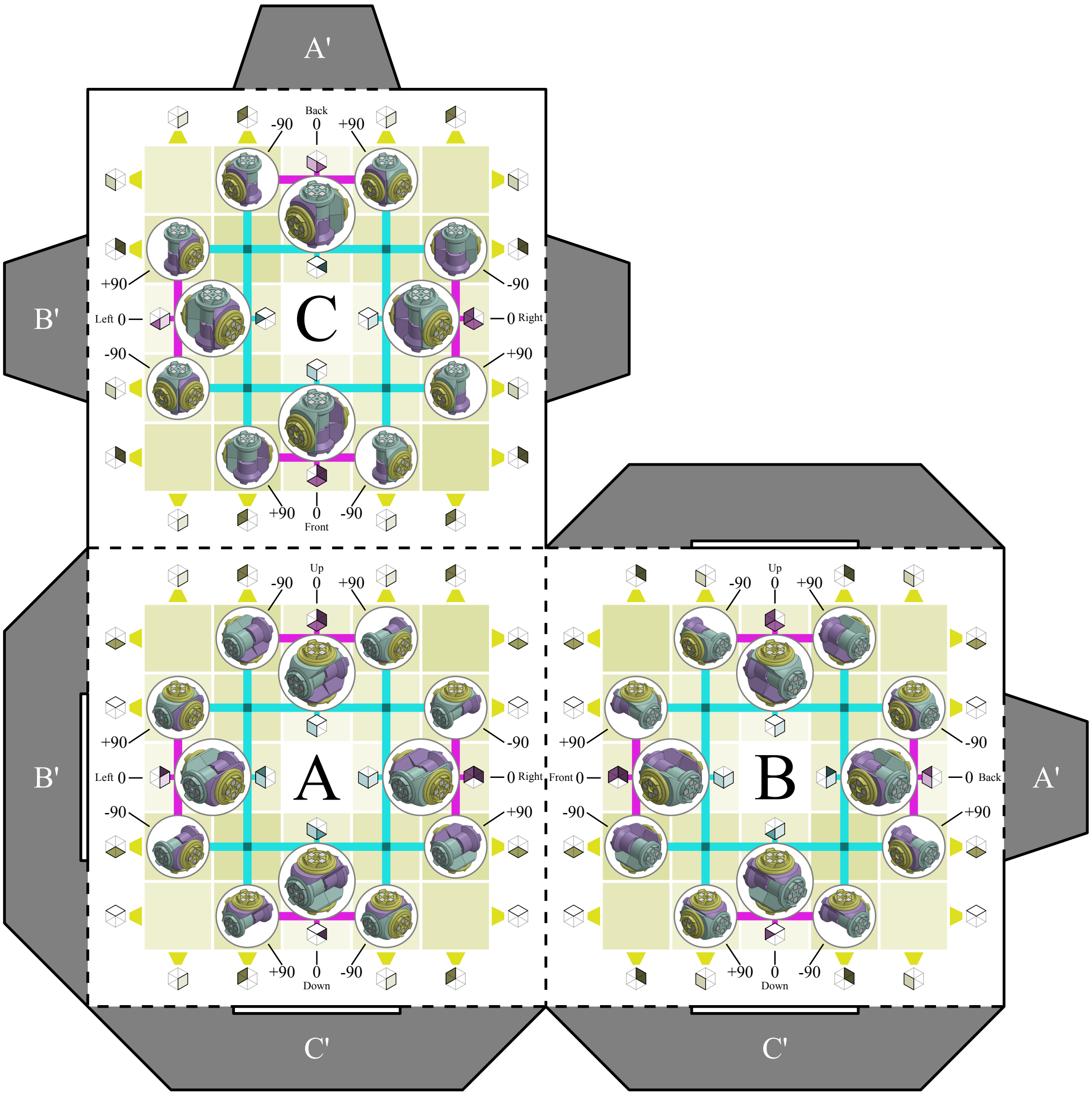
(c) γ_1 Configuration

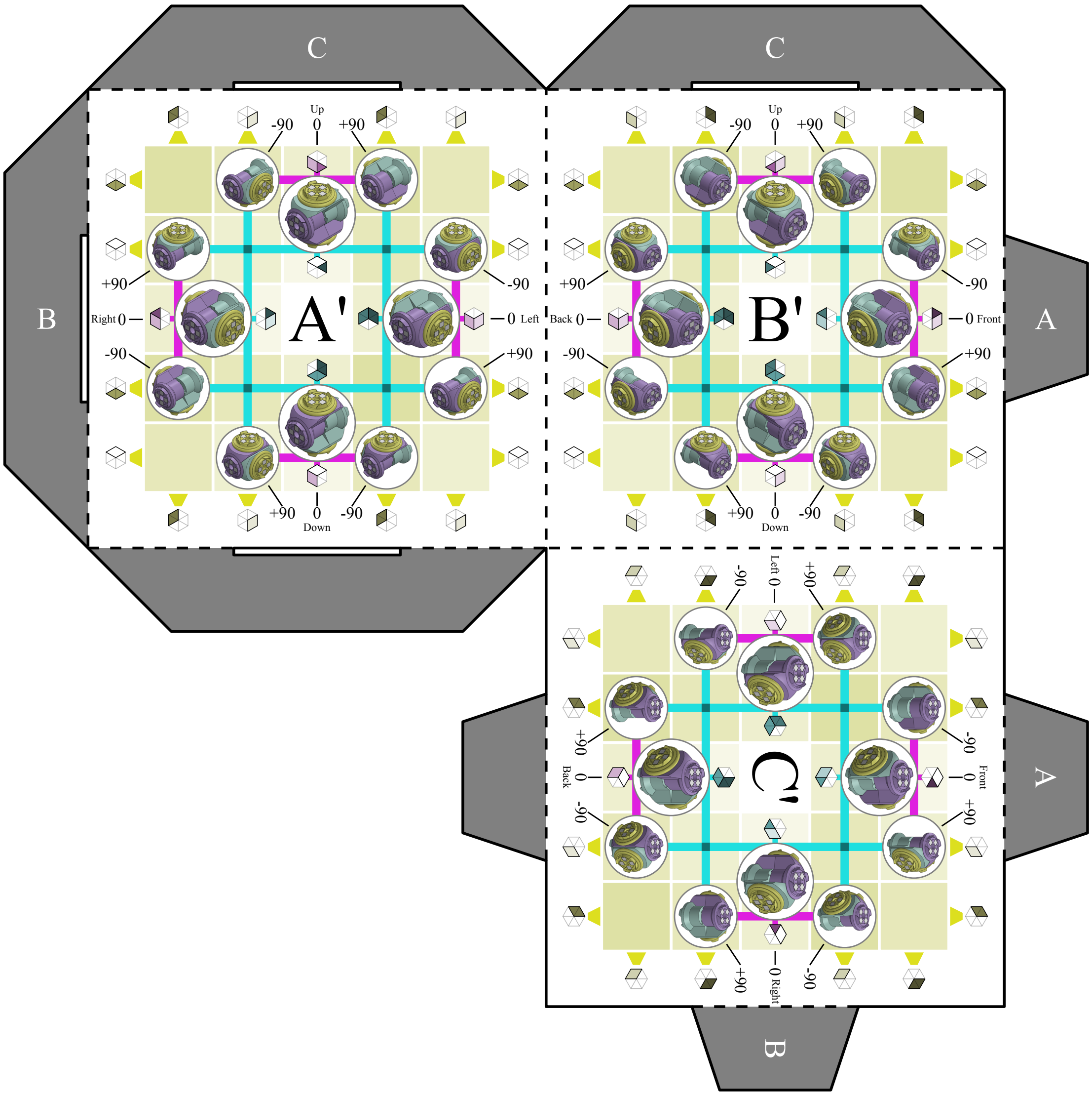
Hinges	-90°	0°	+90°
-90°			
0°			
+90°			

(d) γ_2 Configuration

B. Printable Orientation Cube

This appendix includes a version of Figure 4.7 presented in Section 4.3.2 that can be printed out and assembled in to a cube.




C
C
B
A
A
B
A'
B'
C'
Up
Down
Left
Right
Front
Back
-90
0
+90

C. Orientation Adjacency Matrices

This appendix shows the adjacency matrices of the graphs used to derive the results presented in Table 4.2 of Section 4.3.2.

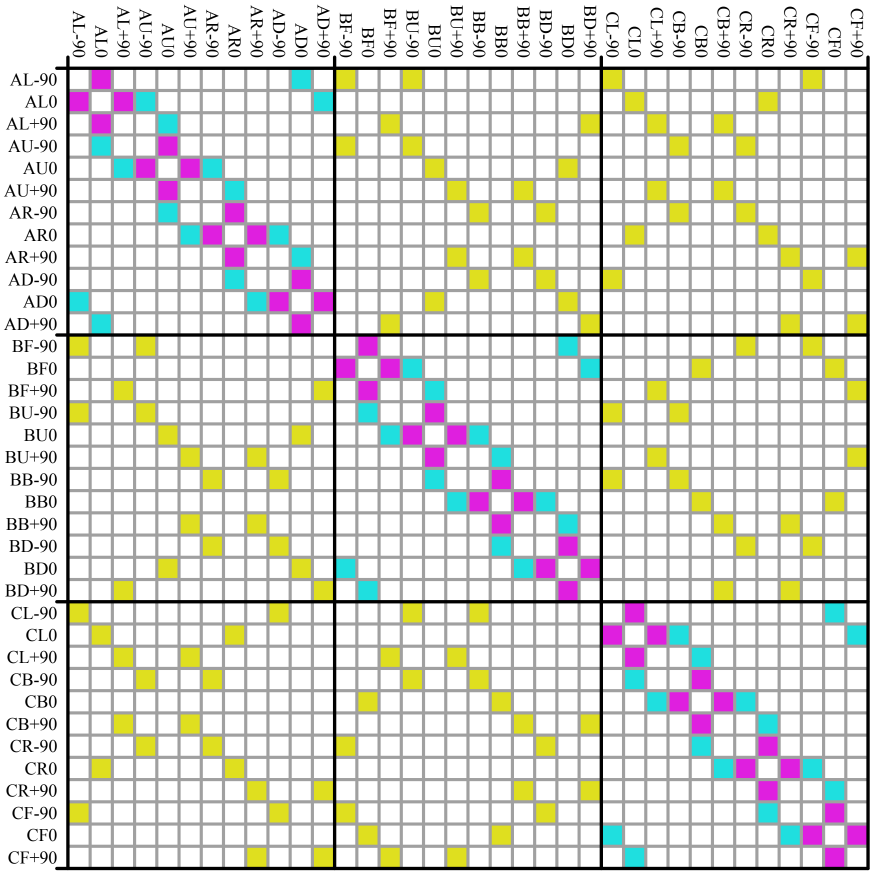


Figure C.1.: The adjacency matrix for the no symmetries HyMod orientation graph, covering the case in which a single unit has modules in all neighboring cubic lattice positions. Each populated cell is a valid orientation transition, with their colors matching those used by the orientation cube.

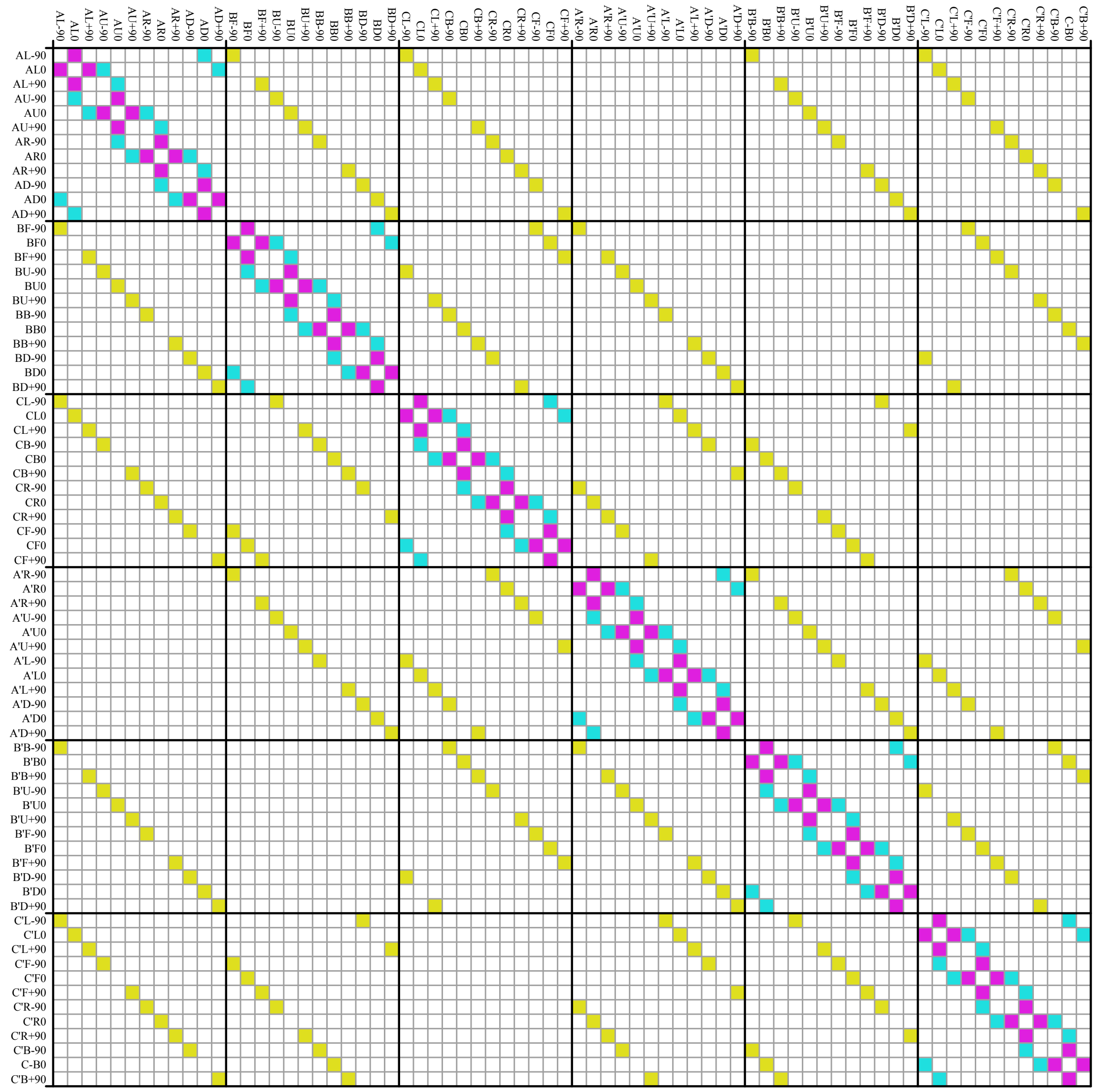


Figure C.2.: The adjacency matrix for the full HyMod orientation graph, covering the case in which a single unit has modules in all neighboring cubic lattice positions. Each populated cell is a valid orientation transition, with their colors matching those used by the orientation cube.

D. ModuleController Pseudocode

This appendix contains pseudocode for the message and packet handling functions of the *ModuleController* class detailed in Section 6.2.3.

Algorithm 1 The functions used during the ModuleController's update loop, to process incoming messages and packets and issue responses. Capitalized function names denote those that are included in this code extract.

```
function UPDATE()
    for all comm ∈ Communicators do
        UpdateCommunicator(comm)
        PROCESSRECEIVEDMESSAGES(comm)
        PROCESSRECEIVEDPACKETS(comm)
    end for
end function

function PROCESSRECEIVEDMESSAGES(comm)
    for all m ∈ Messages(comm) do
        if mCommand ← ConvertToCommand(m) then
            if mResponse ← COMMANDCOMPONENTS(mCommand) then
                SendMessage(mResponse, comm)
            end if
        end if
    end for
end function

function COMMANDCOMPONENTS(mCommand)
    for all comp ∈ Components do
        if mResponse ← HandleCommand(mCommand, comp) then
            return mResponse
        end if
    end for
    return null
end function
```

```
function ProcessReceivedPackets(comm)
    for all p ∈ Packets(comm) do
        p.JumpCount ← p.JumpCount − 1
        ourAddr ← GetAddress()
        if p.DestinationAddress = ourAddr then
            if p.Type = COMMAND then
                if pResponse ← ProcessCommandPacket(p, ourAddr) then
                    SendPacket(pResponse, comm)
                end if
            else if p.Type = RESPONSE then
                ProcessResponsePacket(p)
            end if
        else
            ForwardPacket(p, comm)
        end if
    end for
end function

function ProcessCommandPacket(pCommand, ourAddr)
    address = pCommand.SourceAddress
    currentNumber = pCommand.SequenceNumber
    pLastResponse =LastResponseSent(address)
    lastNumber = pLastResponse.SequenceNumber
    nextNumber =CalculateNext(lastNumber, currentNumber)

    if currentNumber = nextNumber then
        pResponse ←CreateEmptyPacket(ourAddr, address, currentNumber)
        if mCommand ← ExtractCommand(pCommand) then
            if mResponse ← CommandComponents(mCommand) then
                pResponse.Message ← mResponse
            end if
        end if
        LastReceivedJumpCount(address)← pCommand.JumpCount
        LastResponseSent(address)← pResponse
        return pResponse                                  ▷ Return a new response

    else if currentNumber = lastNumber then
        if pCommand.Identifier = pLastResponse.Identifier then
            if pCommand.JumpCount =LastReceivedJumpCount(address) then
                return pLastResponse                      ▷ Return the previous response
            end if
        end if
    end if
end function
```

```
function ProcessResponsePacket(pResponse)
    address ← pResponse.SourceAddress
    if pResponse.SequenceNumber =GetSequenceNumber(address) then
        StoreReceivedResponse(pResponse)
        AdvanceSequenceNumber(address)
    end if
end function

function ForwardPacket(p, sourceComm)
    if p.JumpCount > 0 then
        for all comm ∈ Communicators do
            if comm ≠ sourceComm then
                SendPacket(p, comm)
            end if
        end for
    end if
end function
```

Algorithm 2 The function used to have a ModuleController either act on or forward a command to other modules. Capitalized function names denote those that are included in this code extract.

```
function SendCommandTo(mCommand, address)
    ourAddr ← GetAddress()
    number ←GetSequenceNumber(address)
    pCommand ←CreatePacket(ourAddr, address, number, mCommand)
    if address = ourAddr then
        if pResponse ← ProcessCommandPacket(pCommand, ourAddr) then
            ProcessResponsePacket(pResponse)
        end if
    else
        ForwardPacket(pCommand, null)
    end if
end function
```

Algorithm 3 The function used to have a ModuleController wait for a response to a command. Capitalized function names denote those that are included in this code extract.

```
function WaitForResponseFrom(address, millis)
    prevTime ← GetTime()
    repeat
        Update()
        currentTime ← GetTime()
    until currentTime − prevTime > millis or ResponseReceivedFrom(address)
    if ResponseReceivedFrom(address) then
        return LastResponseReceived(address)        ▷ Return the received response
    end if
end function
```

Bibliography

[1] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[2] S. Camazine, N. R. Franks, J. Sneyd, E. Bonabeau, J.-L. Deneubourg, and G. Theraula, *Self-organization in biological systems.* Princeton University Press, 2003.

[3] O. Holland and C. Melhuish, "Stigmergy, self-organization, and sorting in collective robotics," *Artificial life*, vol. 5, no. 2, pp. 173–202, 1999.

[4] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Gross, "Clustering objects with robots that do not compute," in *Proceedings, 2014 International Conference on Autonomous Agents and Multi-Agent Systems.* International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 421–428.

[5] C. R. Kube and E. Bonabeau, "Cooperative transport by ants and robots," *Robotics and autonomous systems*, vol. 30, no. 1, pp. 85–101, 2000.

[6] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Groß, "Occlusion-based cooperative transport with a swarm of miniature mobile robots," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 307–321, 2015.

[7] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 43–52, 2007.

[8] K. Støy, D. Brandt, and D. J. Christensen, *Self-reconfigurable robots: An introduction.* The MIT Press, 2010.

[9] B. Benhabib and M. Dai, "Mechanical design of a modular robot for industrial applications," *Journal of Manufacturing Systems*, vol. 10, no. 4, pp. 297–306, 1991.

[10] M. Yim, D. G. Duff, and K. Roufas, "Modular reconfigurable robots, an approach to urban search and rescue," in *Proceedings, The 1st International Workshop on Human-friendly Welfare Robotics Systems*, 2000.

[11] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. Homans, "Modular reconfigurable robots in space applications," *Autonomous Robots*, vol. 14, no. 2-3, pp. 225–237, 2003.

[12] S. C. Goldstein, J. D. Campbell, and T. C. Mowry, "Programmable matter," *Computer*, vol. 38, no. 6, pp. 99–101, 2005.

[13] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel, "Reconfigurable manufacturing systems," *CIRP Annals-Manufacturing Technology*, vol. 48, no. 2, pp. 527–540, 1999.

[14] R. R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, and A. M. Erkmen, "Search and rescue robotics," in *Springer Handbook of Robotics.* Springer, 2008, pp. 1151–1173.

[15] M. Stieber, C. Trudel, and D. Hunter, "Robotic systems for the international space station," in *Proceedings, 1997 IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 1997, pp. 3068–3073.

[16] E. H. Østergaard, K. Kassow, R. Beck, and H. H. Lund, "Design of the ATRON lattice-based self-reconfigurable robot," *Autonomous Robots*, vol. 21, no. 2, pp. 165–183, 2006.

[17] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo, and S. Murata, "Distributed self-reconfiguration of M-TRAN III modular robotic system," *The International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 373–386, 2008.

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

[19] A. Spröwitz, R. Moeckel, M. Vespignani, S. Bonardi, and A. Ijspeert, "Roombots: A hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot," *Robotics and Autonomous Systems*, vol. 62, no. 7, pp. 1016–1033, 2014.

[20] H. Wei, Y. Chen, J. Tan, and T. Wang, "Sambot: A self-assembly modular robot system," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 4, pp. 745–757, 2011.

[21] J. Davey, N. Kwok, and M. Yim, "Emulating self-reconfigurable robots - Design of the SMORES system," in *Proceedings, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4464–4469.

[22] S. Russo, K. Harada, T. Ranzani, L. Manfredi, C. Stefanini, A. Menciassi, and P. Dario, "Design of a robotic module for autonomous exploration and multimode locomotion," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1757–1766, 2013.

[23] J. Liedke, R. Matthias, L. Winkler, and H. Wörn, "The collective self-reconfigurable modular organism (CoSMO)," in *Proceedings, 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 2013, pp. 1–6.

[24] V. Zykov, P. William, N. Lassabe, and H. Lipson, "Molecubes extended: Diversifying capabilities of open-source modular robotics," in *Proceedings, IROS 2008 Self-Reconfigurable Robotics Workshop*, 2008.

[25] A. Lyder, R. F. M. Garcia, and K. Støy, "Genderless connection mechanism for modular robots introducing torque transmission between modules," in *Proceedings, ICRA 2010 Workshop on Modular Robots, State of the Art*, 2010, pp. 77–81.

[26] T. Fukuda and S. Nakagawa, "Dynamically reconfigurable robotic system," in *Proceedings, 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 1581–1586.

[27] M. Yim, B. Shirmohammadi, J. Sastra, M. Park, M. Dugan, and C. Taylor, "Towards robotic self-reassembly after explosion," in *Proceedings, 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 2767–2772.

[28] A. Castano, A. Behar, and P. M. Will, "The Conro modules for reconfigurable robots," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 403–409, 2002.

[29] H. Zhang, J. Gonzalez-Gomez, Z. Me, S. Cheng, and J. Zhang, "Development of a low-cost flexible modular robot GZ-I," in *Proceedings, 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.* IEEE, 2008, pp. 223–228.

[30] J. Baca, S. Hossain, P. Dasgupta, C. A. Nelson, and A. Dutta, "ModRED: Hardware design and reconfiguration planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration," *Robotics and Autonomous Systems*, vol. 62, no. 7, pp. 1002–1015, 2014.

[31] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson, "Evolved and designed self-reproducing modular robotics," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 308–319, 2007.

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

[33] M. Yim, "New locomotion gaits," in *Proceedings, 1994 IEEE International Conference on Robotics and Automation.* IEEE, 1994, pp. 2508–2514.

[34] A. Yerpes, J. Baca, J. A. Escalera, M. Ferre, and R. Aracil, "Modular robot based on 3 rotational DoF modules," in *Proceedings, 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2008, pp. 889–894.

[35] R. Möckel, C. Jaquier, K. Drapel, E. Dittrich, A. Upegui, and A. Ijspeert, "YaMoR and Bluemove - an autonomous modular robot with bluetooth interface for exploring adaptive locomotion," in *Climbing and Walking Robots.* Springer, 2006, pp. 685–692.

[36] M. Koseki, K. Minami, and N. Inou, "Cellular robots forming a mechanical structure," in *Distributed Autonomous Robotic Systems 6.* Springer, 2007, pp. 139–148.

[37] D. Rus and M. Vona, "Crystalline robots: Self-reconfiguration with compressible unit modules," *Autonomous Robots*, vol. 10, no. 1, pp. 107–124, 2001.

[38] A. Pamecha, C.-J. Chiang, D. Stein, and G. Chirikjian, "Design and implementation of metamorphic robots," in *Proceedings, 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference*, vol. 10. ASME, 1996.

[39] J. W. Suh, S. B. Homans, and M. Yim, "Telecubes: Mechanical design of a module for self-reconfigurable robotics," in *Proceedings, 2002 IEEE International Conference on Robotics and Automation*, vol. 4, 2002, pp. 4095–4101.

[40] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji, "A 3-D self-reconfigurable structure," in *Proceedings, 1998 IEEE International Conference on Robotics and Automation*, vol. 1, 1998, pp. 432–439.

[41] S. Murata, H. Kurokawa, and S. Kokaji, "Self-assembling machine," in *Proceedings, 1994 IEEE International Conference on Robotics and Automation.* IEEE, 1994, pp. 441–448.

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

[43] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Murata, and S. Kokaji, "Self-reconfigurable modular robot (M-TRAN) and its motion design," in *Proceedings, The 7th International Conference on Control, Automation, Robotics and Vision*, vol. 1, 2002, pp. 51–56.

[44] J. Neubert and H. Lipson, "Soldercubes: a self-soldering self-reconfiguring modular robot system," *Autonomous Robots*, vol. 40, no. 1, pp. 139–158, 2016.

[45] W.-M. Shen, F. Hou, M. Rubenstein, H. Chiu, and A. Kamimura, "Recent progress of SuperBot," *Modular Robots: The State of the Art*, pp. 13–16, 2010.

[46] A. Lyder, R. F. M. Garcia, and K. Stoy, "Mechanical design of Odin, an extendable heterogeneous deformable modular robot," in *Proceedings, 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2008, pp. 883–888.

[47] M. Shimizu, T. Mori, and A. Ishiguro, "A development of a modular robot that enables adaptive reconfiguration," in *Proceedings, 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2006, pp. 174–179.

[48] K. Gilpin, A. Knaian, and D. Rus, "Robot Pebbles: One centimeter modules for programmable matter through self-disassembly," in *Proceedings, 2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2485–2492.

[49] P. White, K. Kopanski, and H. Lipson, "Stochastic self-reconfigurable cellular robotics," in *Proceedings, 2004 IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 2004, pp. 2888–2893.

[50] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss, "Self organizing robots based on cell structures - CEBOT," in *Proceedings, 1988 IEEE International Workshop on Intelligent Robots and Systems*, 1988, pp. 145–150.

[51] R. Oung and R. D'Andrea, "The Distributed Flight Array: Design, implementation, and analysis of a modular vertical take-off and landing vehicle," *The International Journal of Robotics Research*, p. 0278364913501212, 2013.

[52] G. G. Ryland and H. H. Cheng, "Design of iMobot, an intelligent reconfigurable mobile robot with novel locomotion," in *Proceedings, 2010 IEEE International Conference on Robotics and Automation.* IEEE, 2010, pp. 60–65.

[53] K. C. Wolfe, M. S. Moses, M. D. Kutzer, and G. S. Chirikjian, "M^3 Express: A low-cost independently-mobile reconfigurable modular robot," in *Proceedings, 2012 IEEE International Conference on Robotics and Automation.* IEEE, 2012, pp. 2704–2710.

[54] M. Dorigo, "Swarm-bot: An experiment in swarm robotics," in *Proceedings, 2005 IEEE Swarm Intelligence Symposium.* IEEE, 2005, pp. 192–200.

[55] I. O'Hara, J. Paulos, J. Davey, N. Eckenstein, N. Doshi, T. Tosun, J. Greco, J. Seo, M. Turpin, V. Kumar, *et al.*, "Self-assembly of a swarm of autonomous boats into floating structures," in *Proceedings, 2014 IEEE International Conference on Robotics and Automation.* IEEE, 2014, pp. 1234–1240.

[56] Y. Terada and S. Murata, "Automatic assembly system for a large-scale modular structure-hardware design of module and assembler robot," in *Proceedings, 2004*

IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3. IEEE, 2004, pp. 2349–2355.

[57] E. Schweikardt and M. D. Gross, "roBlocks: A robotic construction kit for mathematics and science education," in *Proceedings, The 8th international conference on Multimodal interfaces.* ACM, 2006, pp. 72–75.

[58] A. Faíña, F. Bellas, F. Orjales, D. Souto, and R. J. Duro, "An evolution friendly modular architecture to produce feasible robots," *Robotics and Autonomous Systems*, vol. 63, pp. 195–205, 2015.

[59] M. Pacheco, R. Fogh, H. H. Lund, and D. J. Christensen, "Fable II: Design of a modular robot for creative learning," in *Proceedings, 2015 IEEE International Conference on Robotics and Automation.* IEEE, 2015, pp. 6134–6139.

[60] C. Ünsal, H. Kiliççöte, and P. K. Khosla, "A modular self-reconfigurable bipartite robotic system: Implementation and motion planning," *Autonomous Robots*, vol. 10, no. 1, pp. 23–40, 2001.

[61] K. Kotay and D. Rus, "Efficient locomotion for a self-reconfiguring robot," in *Proceedings, 2005 IEEE International Conference on Robotics and Automation.* IEEE, 2005, pp. 2963–2969.

[62] J. Baca, M. Ferre, and R. Aracil, "A heterogeneous modular robotic design for fast response to a diversity of tasks," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 522–531, 2012.

[63] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, *et al.*, "Swarmanoid: A novel concept for the study of heterogeneous robotic swarms," *IEEE Robotics & Automation Magazine*, vol. 20, no. 4, pp. 60–71, 2013.

[64] S. Kernbach, F. Schlachter, R. Humza, J. Liedke, S. Popesku, S. Russo, T. Ranzani, L. Manfredi, C. Stefanini, R. Matthias, *et al.*, "Heterogeneity for increasing performance and reliability of self-reconfigurable multi-robot organisms," in *Proceedings, IROS 2011 Reconfigurable Modular Robotics Workshop*, 2011.

[65] M. Yim, "Locomotion with a unit-modular reconfigurable robot," Ph.D. dissertation, Citeseer, 1994.

[66] J. Sastra, S. Chitta, and M. Yim, "Dynamic rolling for a modular loop robot," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 758–773, 2009.

[67] M. Yim, D. G. Duff, and K. D. Roufas, "PolyBot: a modular reconfigurable robot," in *Proceedings, 2000 IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 2000, pp. 514–520.

[68] D. Duff, M. Yim, and K. Roufas, "Evolution of PolyBot: A modular reconfigurable robot," in *Proceedings, International Symposium of the Harmonic Drive*. Citeseer, 2001.

[69] V. Zykov, A. Chan, and H. Lipson, "Molecubes: An open-source modular robotics kit," in *Proceedings, 2007 IROS Self-Reconfigurable Robotics Workshop*, 2007, pp. 3–6.

[70] D. Marbach and A. J. Ijspeert, "Online optimization of modular robot locomotion," in *Proceedings, 2005 IEEE International Conference on Mechatronics and Automation*, vol. 1. IEEE, 2005, pp. 248–253.

[71] J. A. Escalera, R. Saltarén, M. Ferre, R. Aracil, and C. E. García, "Base molecule design and simulation of modular robot RobMAT," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 25–30, 2005.

[72] K. D. Chu, S. Hossain, and C. A. Nelson, "Design of a four-dof modular self-reconfigurable robot with novel gaits," in *Proceedings, ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2011, pp. 747–754.

[73] M. Vona and D. Rus, "A physical implementation of the self-reconfiguring crystalline robot," in *Proceedings, 2000 IEEE International Conference on Robotics and Automation*, 2000, pp. 1726–1733.

[74] S. Vassilvitskii, J. Kubica, E. Rieffel, J. Suh, and M. Yim, "On the general reconfiguration problem for expanding cube style modular robots," in *Proceedings, 2002 IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 2002, pp. 801–808.

[75] H. Kurokawa, S. Murata, E. Yoshida, K. Tomita, and S. Kokaji, “A 3-D self-reconfigurable structure and experiments,” in *Proceedings, 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2. IEEE, 1998, pp. 860–865.

[76] M. W. Jorgensen, E. H. Østergaard, and H. H. Lund, “Modular ATRON: Modules for a self-reconfigurable robot,” in *Proceedings, 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2004, pp. 2068–2073.

[77] D. J. Christensen, D. Brandt, and K. Stoy, “Towards artificial ATRON animals: scalable anatomy for self-reconfigurable robots,” in *The RSS Workshop on Self-Reconfigurable Modular Robots*, 2006.

[78] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, “M-TRAN: Self-reconfigurable modular robotic system,” *IEEE/ASME transactions on mechatronics*, vol. 7, no. 4, pp. 431–441, 2002.

[79] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Kokaji, and S. Murata, “M-TRAN II: Metamorphosis from a four-legged walker to a caterpillar,” in *Proceedings, 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3. IEEE, 2003, pp. 2454–2459.

[80] “M-TRAN III module,” http://unit.aist.go.jp/is/frrg/dsysd/mtran3/mtran3.htm, National Institute of Advanced Industrial Science and Technology (AIST), (Last accessed: 21 September 2016).

[81] B. Salemi, M. Moll, and W.-M. Shen, “Superbot: A deployable, multi-functional, and modular self-reconfigurable robotic system,” in *Proceedings, 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 3636–3641.

[82] A. Spröwitz, S. Pouya, S. Bonardi, J. Van Den Kieboom, R. Mockel, A. Billard, P. Dillenbourg, and A. J. Ijspeert, “Roombots: Reconfigurable robots for adaptive furniture,” *IEEE Computational Intelligence Magazine*, vol. 5, no. 3, pp. 20–32, 2010.

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

[84] A. Lyder, "Towards versatile robots through open heterogeneous modular robots," Ph.D. dissertation, 2010.

[85] B. Kirby, J. Campbell, B. Aksak, P. Pillai, J. Hoburg, T. Mowry, and S. C. Goldstein, "Catoms: Moving robots without moving parts," in *Proceedings, National Conference on Artificial Intelligence*, vol. 20, no. 4. AAAI Press; MIT Press, 2005, p. 1730.

[86] R. Ding, P. Eastwood, F. Mondada, and R. Groß, "A stochastic self-reconfigurable modular robot with mobility control," in *Proceedings, 13th Annual TAROS Conference and the 15th Annual FIRA RoboWorld Congress*, vol. 7429. Springer, 2012, pp. 416–417.

[87] K. Gilpin, K. Koyanagi, and D. Rus, "Making self-disassembling objects with multiple components in the robot pebbles system," in *Proceedings, 2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3614–3621.

[88] R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in Swarm-Bots," *IEEE transactions on robotics*, vol. 22, no. 6, pp. 1115–1130, 2006.

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

[90] R. Oung and R. D'Andrea, "The Distributed Flight Array," *Mechatronics*, vol. 21, no. 6, pp. 908–917, 2011.

[91] G. Jing, T. Tosun, M. Yim, and H. Kress-Gazit, "An end-to-end system for accomplishing tasks with modular robots," in *Proceedings, 2016 Robotics: Science and Systems*, 2016.

[92] M. D. Kutzer, M. S. Moses, C. Y. Brown, M. Armand, D. H. Scheidt, and G. S. Chirikjian, "Design of a new independently-mobile reconfigurable modular robot," in *Proceedings, 2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2758–2764.

[93] C. Ünsal, H. Kiliççöte, and P. K. Khosla, "I(CES)-Cubes: A modular self-reconfigurable bipartite robotic system," in *Proceedings, Photonics East'99*. International Society for Optics and Photonics, 1999, pp. 258–269.

[94] K. Kotay, D. Rus, M. Vona, and C. McGray, "The self-reconfiguring robotic molecule," in *Proceedings, 1998 IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 1998, pp. 424–431.

[95] S. Popesku, E. Meister, F. Schlachter, and P. Levi, "Active wheel-an autonomous modular robot," in *Proceedings, 2013 IEEE Conference on Robotics, Automation and Mechatronics*, 2013.

[96] "Cubelets Robot Construction for Kids," http://www.modrobotics.com/cubelets, (Last accessed: 21 September 2016).

[97] M. Pacheco, M. Moghadam, A. Magnússon, B. Silverman, H. H. Lund, and D. J. Christensen, "Fable: Design of a modular robotic playware platform," in *Proceedings, 2013 IEEE International Conference on Robotics and Automation.* IEEE, 2013, pp. 544–550.

[98] A. Faíña, F. Bellas, F. López-Peña, and R. J. Duro, "EDHMoR: Evolutionary designer of heterogeneous modular robots," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, pp. 2408–2423, 2013.

[99] J. Liedke, L. Winkler, and H. Wörn, "An alternative locomotion unit for mobile modular self-reconfigurable robots based on archimedes screws," in *Mechatronics and its Applications (ISMA), 2013 9th International Symposium on.* IEEE, 2013, pp. 1–6.

[100] B. T. Kirby, B. Aksak, J. D. Campbell, J. F. Hoburg, T. C. Mowry, P. Pillai, and S. C. Goldstein, "A modular robotic system using magnetic force effectors," in *Proceedings, 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2007, pp. 2787–2793.

[101] M. E. Karagozler, J. D. Campbell, G. K. Fedder, S. C. Goldstein, M. P. Weller, and B. W. Yoon, "Electrostatic latching for inter-module adhesion, power transfer, and communication in modular robots," in *Proceedings, 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2007, pp. 2779–2786.

[102] J. Neubert, A. Rost, and H. Lipson, "Self-soldering connectors for modular robots," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1344–1357, 2014.

[103] J. Davey, J. Sastra, M. Piccoli, and M. Yim, "ModLock: A manual connector for reconfigurable modular robots," in *Proceedings, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 3217–3222.

[104] J. Liedke and H. Wörn, "CoBoLD - A bonding mechanism for modular self-reconfigurable mobile robots," in *Proceedings, 2011 IEEE International Conference on Robotics and Biomimetics*, 2011, pp. 2025–2030.

[105] W.-M. Shen, R. Kovac, and M. Rubenstein, "SINGO: A single-end-operative and genderless connector for self-reconfiguration, self-assembly and self-healing," in *Proceedings, 2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 4253–4258.

[106] M. Plooij, G. Mathijssen, P. Cherelle, D. Lefeber, and B. Vanderborght, "Lock your robot: A review of locking devices in robotics," *IEEE Robotics & Automation Magazine*, vol. 22, no. 1, pp. 106–117, 2015.

[107] M. Nilsson, "Connectors for self-reconfiguring robots," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 473–474, 2002.

[108] S. G. M. Hossain, C. A. Nelson, and P. Dasgupta, "RoGenSiD - A rotary plate genderless single-sided docking mechanism for modular self-reconfigurable robots," in *Proceedings, 2013 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2013, p. V06BT07A011.

[109] W. Saab and P. Ben-Tzvi, "A genderless coupling mechanism with 6-dof misalignment capability for modular self-reconfigurable robots," *Journal of Mechanisms and Robotics*, 2016.

[110] W.-M. Shen, B. Salemi, and P. Will, "Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots," *IEEE transactions on Robotics and Automation*, vol. 18, no. 5, pp. 700–712, 2002.

[111] A. Kamimura, H. Kurokawa, E. Yoshida, K. Tomita, S. Kokaji, and S. Murata, "Distributed adaptive locomotion by a modular robotic system, m-tran ii," in *Proceedings, 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3. IEEE, 2004, pp. 2370–2377.

[112] C. Parrott, T. J. Dodd, and R. Groß, "HiGen: A high-speed genderless mechanical connection mechanism with single-sided disconnect for self-reconfigurable modular robots," in *Proceedings, 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3926–3932.

[113] "About the Mojo Desktop 3D Printer," http://www.stratasys.com/3d-printers/idea-series/mojo, Stratasys, (last accessed: 08/02/2017).

[114] S. Tang, Y. Zhu, J. Zhao, and X. Cui, "The UBot modules for self-reconfigurable robot," in *Proceedings, 2009 ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots*, 2009, pp. 529–535.

[115] D. J. Christensen, E. H. Østergaard, and H. H. Lund, "Metamodule control for the ATRON self-reconfigurable robotic system," in *Proceedings, The 8th Conference on Intelligent Autonomous Systems*, 2004, pp. 685–692.

[116] R. F. M. Garcia, K. Stoy, D. J. Christensen, and A. Lyder, "A self-reconfigurable communication network for modular robots," in *Proceedings, The 1st international conference on Robot Communication and Coordination*. IEEE Press, 2007, pp. 23–30.

[117] "Arduino Shields," http://learn.sparkfun.com/tutorials/arduino-shields, Sparkfun Electronics, (Last accessed: 21 September 2016).

[118] "DAGU Mechanical 2013, MK II robotic claw with servo," http://www.dagurobot.com/goods.php?id=75, DAGU Hi-Tech Electronic Robotics, (Last accessed: 21 September 2016).

[119] J. Anderson, "RoboPlay 2014 - Mecanum Linkbot Interlude," http://www.youtube.com/watch?v=GqClygCJpKs, (Last accessed: 21 September 2016).

[120] F. Adăscăliţei and I. Doroftei, "Practical applications for mobile robots based on mecanum wheels-a systematic survey," 2011.

[121] A. Gfrerrer, "Geometry and kinematics of the mecanum wheel," *Computer Aided Geometric Design*, vol. 25, no. 9, pp. 784–791, 2008.

[122] "A set of 127mm aluminium mecanum wheels (4 pieces)," http://www.nexusrobot.com/product31c8.html?id_product=181, NexusRobot, (Last accessed: 21 September 2016).

[123] "Raspberry Pi Zero: The $5 Computer," https://www.raspberrypi.org/blog/raspberry-pi-zero/, Raspberry Pi Foundation, (Last accessed: 21 September 2016).

[124] T.-K. Dinh and A. Ijspeert, "Alignment of a Roombot metamodule & extendable grid," 2010.

[125] T. Leighton, "Methods for message routing in parallel machines," in *Proceedings, 24th ACM Symposium on Theory of Computing.* ACM, 1992, pp. 77–96.

[126] A. S. Tanenbaum, *Computer networks.* Prentice Hall New Jersey, 1996, vol. 4.

[127] M. W. Murhammer, O. Atakan, S. Bretz, L. R. Pugh, K. Suzuki, and D. H. Wood, *TCP/IP Tutorial and Technical Overview.* Prentice Hall Upper Saddle River, NJ, 1998.

[128] B. Hallgren, "CANbus - Hardware description," http://canlab.cz/pages/download/canhard.pdf, CERN ATLAS DSC, (Last accessed: 21 September 2016).

[129] E. Bendersky, "Framing in Serial Communications," http://eli.thegreenplace.net/2009/08/12/framing-in-serial-communications, 2009, (Last accessed: 21 September 2016).

[130] A. Casal and M. H. Yim, "Self-reconfiguration planning for a class of modular robots," in *Photonics East'99.* International Society for Optics and Photonics, 1999, pp. 246–257.

[131] M. P. Ashley-Rollman, P. Pillai, and M. L. Goodstein, "Simulating multi-million-robot ensembles," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on.* IEEE, 2011, pp. 1006–1013.

[132] D. Christensen, D. Brandt, K. Stoy, and U. P. Schultz, "A unified simulator for self-reconfigurable robots," in *Proceedings, 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2008, pp. 870–876.

[133] "OGRE - Open Source 3D Graphics Engine — Home of a marvelous rendering engine," http://www.ogre3d.org, (Last accessed: 21 September 2016).

[134] "wxWidgets: Cross-Platform GUI Library," http://www.wxwidgets.org, (Last accessed: 21 September 2016).