



HAL
open science

Modular robotics meet Internet of things : safety, security and performance challenges and countermeasures

Jean-Paul Abs Yaacoub

► **To cite this version:**

Jean-Paul Abs Yaacoub. Modular robotics meet Internet of things : safety, security and performance challenges and countermeasures. Robotics [cs.RO]. Université Bourgogne Franche-Comté, 2024. English. NNT : 2024UBFCD013 . tel-04773981

HAL Id: tel-04773981

<https://theses.hal.science/tel-04773981v1>

Submitted on 8 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT

DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE-FRANCHE-COMTÉ

PRÉPARÉE À L'UNIVERSITÉ DE FRANCHE-COMTÉ

École doctorale n°37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

JEAN PAUL YAACOUB

**Modular Robotics Meets Internet of Things: Safety, Security and
Performance Challenges and Countermeasures**

**La Robotique Modulaire Rencontre l'Internet des Objets : Défis et Contre-Mesures en
Matière de Sécurité, de Sûreté et de Performances**

Thèse présentée et soutenue à Montbéliard, le 05 Juin 2024

Composition du Jury :

Prof. DEMERJIAN JACQUES	Professeur à Lebanese University, Liban	Rapporteur
Prof. ABOUAISSA ABDELHAFID	Professeur à l'Université de Haute Alsace	Rapporteur
Prof. MICHELIN SYLVAIN	Professeur à l'Université Gustave Eiffel	Président du Jury
Prof. NOURA HASSAN	Professeur à l'Université de Franche-Comté	Examineur
Prof. MAKHOUL ABDALLAH	Professeur à l'Université de Franche-Comté	Examineur
Prof. PIRANDA BENOIT	Maître de Conférences HDR de l'Université de Franche-Comté, France	Directeur de thèse

ACKNOWLEDGEMENTS

First of all, I would like to thank all thesis committee members for having accepted to take and dedicate their time to review my dissertation and provide valuable feedback despite their busy schedules. I am also grateful for their insightful feedback, constructive criticism, and dedication to helping me refine my ideas and improve the quality of this thesis.

I'd like to express my thanks to my supervisor, Dr. Benoit PIRANDA, who has helped and assisted me throughout this research project. I am grateful for our constant exchange of information, friendly chats, at the end of our meetings, and his availability and support in my academic endeavors. I would also like to thank him for the countless hours he has spent improving my work in that process.

I would also like to thank my mentor Professor Hassan NOURA, for his dedicated guidance and selfless support by continuously providing encouragement and being always willing and enthusiastic to assist in any way he could throughout the research project. Prof. NOURA is the main person who has inspired me to pursue a career in scientific research, to whom I am forever grateful, as he sparked my interest in various fields, including but not limited to cyber-security, ethical hacking, digital forensics, machine learning, and robotics. His unaccounted guidance, mentorship, and unwavering support throughout every stage of this research endeavor, as well as expertise and encouragement have been instrumental, vital and crucial in shaping this work. I wish him all the best in his research work and projects, as well as in finding the finest PhD students to help and achieve his groundbreaking research visions.

Furthermore, I am seizing this opportunity to thank all other members of the OMNI team at FEMTO-ST for their support and interest at various points of my thesis. I am also thankful to Dr. Abdallah Makhoul for his time, discussions and support, and Frédéric Lassabe, who was involved for some time in the discussions surrounding my work on Blinky Blocks.

Special thanks to my friends and colleagues for their camaraderie and companionship throughout this process. Their shared experiences have made this journey more memorable. These past two and a half years would not have been the same without the good times with fellow PhD students and office mates including our interns, walks, and outdoor activities. I would also like to thank my colleagues for their supporting role throughout this challenging professional and personal experience, especially my colleagues who are national and international PhD graduates and undergraduates, especially members of the

FEMTO-ST OMNI DISC team Montbéliard, with whom we have mutually supported each other during the whole thesis period.

Finally, I would like to thank my family (including my parents) and friends, including my best friend and godson from England (UK), for their enduring support throughout these years, especially while being abroad and away from them. I extend heartfelt thanks for their endless encouragement and understanding during this challenging yet rewarding journey, which has been a constant source of motivation. I would also like to thank the Protection Civil du Doubs (APC25 - Besancon et Montbéliard) and the Sauveteurs en Mer (SNSM 25) for their constant appreciation and support.

In conclusion, I am thankful to everyone mentioned above and countless others who have contributed to my academic and personal development. This thesis would not have been possible without their unwavering support and encouragement.

In conclusion, I am deeply thankful to all the people that I have just mentioned and to everyone else who has contributed in one way or another to the realization of this PhD work, as well as my academic and personal development. This also includes those who took part in the study and enabled this research to be possible:

Thank You. Merci à vous.

Certa Cito - Fear Naught - Fidem Servavi - Aeterna Victrix

CONTENTS

I	Introduction	1
1	Introduction	3
1.1	Context	3
1.2	Objectives Of The Thesis	4
1.3	Outline Of The PhD Thesis Dissertation	6
II	Context & Problems	9
2	Context and Problems	11
2.1	Context: Modular Robotic Systems	11
2.1.1	Components of MSRR	11
2.1.2	Characteristics & Programming Hypothesis	13
2.1.3	Modular Robotic Classification	13
2.1.4	Modular Robot Types	17
2.1.5	Taxonomy of Architectures	22
2.1.6	IoMRT-related Applications	24
2.1.7	Properties: Modular Robots Vs. Traditional Robots	28
2.2	Self-Reconfiguration Process	32
2.2.1	Criteria	34
2.2.2	Characteristics	35
2.2.3	Advantages	36
2.2.4	Self-Reconfiguration Types	38
2.3	Problems IoMRT: Limitations, Failures & Challenges	39
2.3.1	Limitations & Constraints	39
2.3.2	Failures	41

2.3.3	Challenges	43
2.4	IoMRT Security	45
2.4.1	Security Issues & Safety Concerns	45
2.4.1.1	Security Issues	45
2.4.1.2	Security & Safety Concerns	47
2.4.2	Security Threats & Vulnerabilities	50
2.4.2.1	Threat Source	50
2.4.2.2	Threat Type	51
2.4.2.3	Threat Nature	52
2.4.2.4	Security Vulnerabilities	52
2.4.3	Security Attacks	54
2.4.3.1	Attack Source	54
2.4.3.2	Attack Types	55
2.4.3.3	Attacks Classification	61
2.4.3.4	Targeted Components	62
2.5	IoMRT-related Risks	64
2.5.1	Risk Identification	64
2.5.1.1	Risk Types	66
2.5.1.2	Risk Causes	67
2.5.2	Risk Life-cycle	68
2.5.2.1	Risk Planning	69
2.5.2.2	Risk Management	69
2.5.2.3	Risk Assessment	70
2.5.2.4	Risk Analysis	72
2.5.2.5	Mitigation Methods	73
2.6	Presented Modular Robotic Systems Solutions	75
2.6.1	Theoretical Robotic Solutions	75
2.6.2	Mobile Robotic Solutions	77

3.1	Introduction	81
3.2	Detailed state-of-art	83
3.2.1	Frameworks & Module Robotic Solutions	83
3.2.2	Algorithmic Robotic Solutions	86
3.2.3	Simulation-based Solutions	91
3.2.4	Robotic Security Solutions	94
3.3	Conclusion	96
III	Contribution	99
4	The Concept of Internet of Modular Robotic Things	101
4.1	Introduction	101
4.2	Details of the contribution	103
4.3	Communication Nature & Technologies	103
4.3.1	IoMRT Layer Protocols	104
4.3.1.1	IoMRT Physical/Data-Link Layer Communications	104
4.3.1.2	IoMRT Network Layer Protocols	106
4.3.1.3	IoMRT Transport Layer Protocols	108
4.3.1.4	IoMRT Application/Session Layer Protocols	108
4.3.2	Modular Robotic Communication Layers	110
4.3.2.1	Modular Robots Physical/Data-link Layer Communications	110
4.3.2.2	Modular Robots Network Layer	111
4.3.2.3	Modular Robots Session Layer	111
4.3.2.4	Modular Robots Application Layer	112
4.3.2.5	Modular Robots Simulation Framework	112
4.4	IoMRT Algorithms	113
4.4.1	Machine Learning Algorithms	114
4.4.2	Application Algorithms	116
4.4.3	IoMRT Networking Topologies	119
4.5	Suggestion & Recommendation:	122

4.5.1	Security Aspect	122
4.5.2	Safety Aspect	123
4.5.3	Privacy Aspect	124
4.5.4	Availability Aspect	125
4.5.5	Lessons Learnt	126
4.5.6	Future Work	127
4.6	Conclusion	130
5	A new <i>Blinky Block</i> Communication Protocol	133
5.1	Introduction	133
5.2	<i>Blinky Blocks</i> Benchmark & Compression Models	135
5.3	Method and experiments	138
5.4	Conclusion and Future Works	142
6	LCAPBB	145
6.1	Introduction	145
6.1.1	Problem Formulation	146
6.1.2	Related Work	146
6.1.3	Contribution	146
6.1.4	Organisation	147
6.2	<i>Blinky Blocks</i> : Attacks & Countermeasure	147
6.2.1	BB Possible Attacks	149
6.2.2	BB Mitigation Methods	150
6.3	PROLISEAN Protocol	150
6.3.1	First Version: Simple authentication	151
6.3.1.1	Vulnerabilities Of The First Version	151
6.3.2	Second Version: Authentication & Ciphering	151
6.3.2.1	Vulnerabilities Of The Second Version	152
6.3.3	Third Version: PROLISEAN Protocol	153
6.3.3.1	Vulnerabilities Of The Third Version	153
6.3.4	Fourth Version: Improved PROLISEAN Protocol	154

6.3.4.1	Vulnerabilities Of The Fourth Version	154
6.4	Proposed Secure and Robust Protocol	155
6.4.0.1	Proposition Flexible & Robust Secure Transmission Protocol	155
6.4.1	Dynamic key approach	156
6.4.2	Security Requirements vs Application Type	156
6.5	Proposed Lightweight Cryptographic Algorithms	157
6.5.1	Dynamic Cryptographic Primitives	157
6.6	Lightweight Cipher Schemes	158
6.6.1	Substitution Cipher	158
6.6.2	Permutation Cipher Variant	159
6.6.3	Lightweight Stream Cipher: Addition Keystream Cipher	159
6.6.3.1	Multi-Operations Cipher variant	160
6.6.3.2	Updating Cryptographic Primitive Technique	161
6.6.4	Security Analysis	162
6.6.4.1	Uniform Distribution	162
6.6.4.2	Random Recurrence	163
6.6.4.3	Correlation Coefficient	163
6.6.5	Sensitivity Test	164
6.6.6	Key Sensitivity for the Dynamic Approach	164
6.6.6.1	Message Sensitivity Test	165
6.6.7	Evaluation of the Proposed Update Cryptographic Primitives Process	165
6.6.8	Randomness Of The Generation Algorithm Key-Stream	166
6.7	Experimental Performance Results	166
6.8	Cryptanalysis of the Proposed Cipher Scheme	167
6.8.1	Resistance Against Statistical Attacks (Ciphertext Only Attacks) . .	168
6.8.2	Resistance Against Chosen/Known Plain-text/Cipher-text Attacks . .	170
6.8.3	Resistance Against Key-Related Attacks	170
6.8.4	Weak Keys	170
6.8.5	Resistance Against Brute-Force Attacks	171
6.8.6	Resistance Against More Powerful Attacks	171

6.9 Performance Analysis	172
6.9.1 Effect Of Error Propagation	172
6.9.2 Computational Delay	173
6.10 Suggestions & Recommendations	175
6.11 Conclusion	176
IV Conclusion	177
7 General conclusion	179
7.1 Summary Of The PhD Thesis	181
7.2 Perspectives	185



INTRODUCTION

INTRODUCTION

1.1/ CONTEXT

The world is becoming more and more digitized with the rise of modular robotic systems. Therefore, with the increasing demands and needs for robotic systems, the modular robotic domain was introduced as an essential key part of the Internet-of-Things (IoT) [155], where the IoT encompasses a network of interconnected devices and systems that collect and exchange data over the internet, enabling real-time monitoring, automation, and optimization across various domains.

Modular robots [492] are a versatile class of robotic systems made of individual modules that can be reconfigured and combined with their flexible and customizable nature in various configurations [166], enabling adaptability to diverse tasks. Modular self-configurable robotic systems (including swarms [380] and lattice-based robots [321]) are classed as "smart" autonomous machines with kinematic properties which are defined by a set of interconnected links, modules, and algorithms to achieve the required three-dimensional (3D) or two-dimensional (2D) complex shape or hierarchical structure. Unlike the "herd" term [327], which is based on a group of autonomous robots collaboratively operating together under centralized control, often following a predefined leader, "swarms" are groups of autonomous robots that operate collectively under decentralized control, enabling them to accomplish complex tasks efficiently and adaptively. Robots with lattice-based modular structures are composed of individual parts placed in a grid-like pattern that allows for dynamic reconfiguration to build a variety of shapes and carry out complex tasks.

Due to its "intelligent" concept, it has become suitable for its deployment in the Internet of Robotic Things (IoRT) domain. Thus, becoming a key part of it and establishing itself as the newly revolutionized Internet of Modular Robotic Things (IoMRT). The IoRT refers to the networked ecosystem where robotic devices are connected to the internet, enabling them to communicate, share data, and collaborate autonomously or semi-autonomously, while the IoMRT refers to the interconnected network of modular robotic systems within the IoT that autonomously self-configure, integrate and communicate for various applica-

tions and tasks.

This thesis presents a survey that highlights and discusses the IoMRT concept which is a novel concept that focuses on self-reconfigurable modular robots and robotic systems by discussing their criteria, characteristics, architecture, and design. The security, safety, and privacy aspects are also presented and discussed, making it the first thesis to focus on this topic and introduce the novel IoMRT concept. Moreover, the main drawbacks and challenges are also highlighted, while the already available solutions are also presented and analyzed. A brief analysis regarding each solution is also presented with an insight into their future work. Additionally, more work was presented regarding the security aspect(s) of modular robotic systems to protect them from possible cyber-physical attacks. Thus, covering all possible aspects of the newly introduced IoMRT concept (see Table 4.1).

Modular Self-Reconfigurable Robots (MSRR) are innovative robotic systems composed of interconnected modules that can autonomously change their shape and configuration to adapt to various tasks and environments, offering unparalleled versatility and adaptability in IoT-based robotics applications. However, designing MSRR and Modular Self-Reconfigurable Robotic Systems (MSRRS) is a challenging task especially when it is part of IoMRT. This is due to the fact that MSRRSs represent a transformative advancement in robotics technology, comprising interconnected modules equipped with self-reconfiguration capabilities, enabling dynamic morphological changes to efficiently and autonomously navigate different tasks. Therefore, this proves to be also a very rewarding outcome since it can shape the future aspect of modular and non-modular robotic systems in the robotic field and IoT domain alike.

At the heart of the intersection of programmable matter and the IoT, modular self-reconfigurable robotic systems represent an innovative structure in which autonomous modules dynamically modify their configurations, enabling unmatched flexibility and adaptability in tackling a wide range of real-world problems.

1.2/ OBJECTIVES OF THE THESIS

The future of robotic systems lies within these main notions of self-reconfiguration, self-shaping, self-scaling, and self-healing processes to accurately achieve a higher level of robustness, flexibility, and adaptability respectively.

This thesis aims to present a detailed study of the MSRR and MSRRS and their link to IoT to highlight the importance of their adoption in real-world and real-time IoT applications in the foreseen future. This will help with identifying and overcoming unforeseen situations and ensuring long-term self-sustaining robotic systems mainly capable of performing self-healing, self-reconfiguration, and self-replication tasks. At the same time, this thesis

also aims to highlight the main drawbacks, challenges, threats, and attack aspects that may target the modular robotic systems to offer a secure, safe, accurate, and error-free loRT and loMRT domains and shedding light on the future of loMRT which it will be combined with the loSRT to introduce the loMSRT (see Table 4.1).

The integration of this new concept will help to outsource both components and programming and ensure the software compatibility with the Application Programming Interface (API). At the same time, to also visualize the impact of the modular embedded system's total performance, the objective is for it to be achieved at lower computation and programming times and at reduced costs with more affordable solutions being presented. As the world gets more smarter and more connected, more computing power is being added to smaller modular robotic devices. Nonetheless, modular robots now represent the intersection between embedded software and modular robotic components. Both of them include software and hardware modules, high processing power, cameras, multi-purpose/specialized sensors, and electro-mechanical components [175]. Despite them being complex to build and program, except that they are being extensively used to develop new prototypes that allow their deployment in real life to deal with real case scenarios [106] and applications [74, 216]. This is mostly due to the collective behavior of swarm intelligence and decentralized, self-organised systems.

This thesis introduces a novel IoT concept that complements the loRT domain and overcomes its main limitations and challenges. The newly introduced concept is called the Internet of Modular Robotic Things (loMRT), which allows swarms of autonomous robots [72] and distributed systems with local sensing and communication capabilities, which are part of loSRT, to be connected to centralized and well-defined access control to enhance the connection, synchronization and communication with each other, which is one of the main other "collective behavior" limitations for swarms robotics [331, 395]. Another limitation that the loMRT can overcome is the terrain problem and environmental conditions that prove to be challenging. A problem that is mitigated within the loMRT domain, offering more flexibility and adaptability to any environmental and geographical changes.

Moreover, this thesis also examines the newly introduced concept and compares it with the loRT domain, especially in terms of performance, safety, security, accuracy, and privacy aspects that can be linked to the modular robotic domain and IoT, including challenges and vulnerabilities/security gaps [36, 370, 462]. Future work is also presented concerning loMRT, and several lessons that were learned, are being learned, and are to be learned are also presented and explained. Moreover, several suggestions and recommendations are also presented in this regard to take precautionary and protective

measures and procedures, especially when adopting the IoMRT domain and the modular robotic design.

The real aim is to propose this new concept as an advanced version of the IoRT, which takes into consideration the main limitations of robotic systems and aims to solve them through reliance on the modular robotic concept. However, it is also important to highlight the main drawbacks, attacks, and challenges that surround this newly introduced domain with an insight into how the future of IoT is going to be once this IoMRT concept is introduced and applied.

More light was also shed on the swarm robotics which are to be further integrated to form another new IoRT concept which is the IoSRT. Additionally, this work also highlights the future of modular robots and swarms combined which will merge both concepts to allow the emergence of the IoMSRT, which saw recent testing in both civilian and military domains and is said to be deployed in the near future.

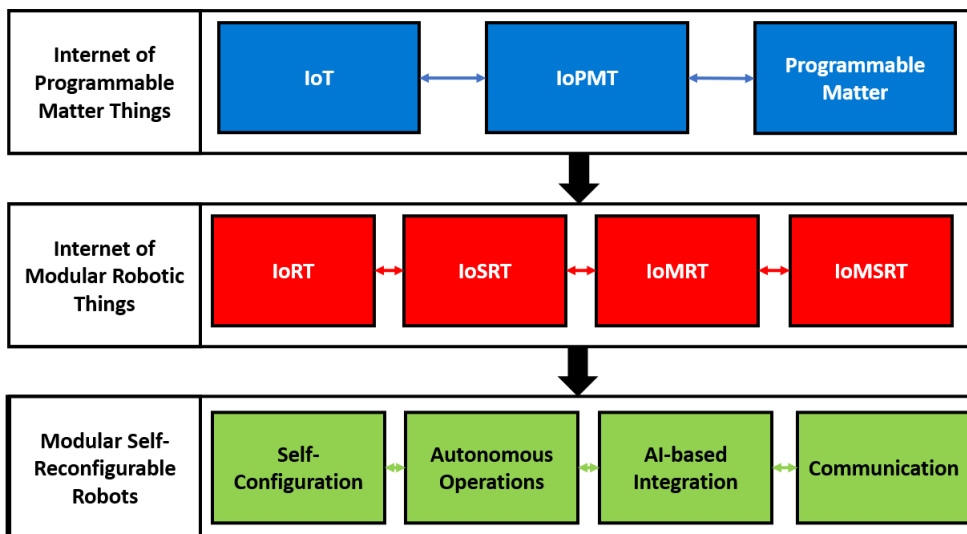


Figure 1.1: IoPMRT: Present Application and Future Use.

1.3/ OUTLINE OF THE PHD THESIS DISSERTATION

As the world has become more digitally connected, modular robotic systems have grown in popularity and are now key parts of the IoT. These self-configurable modular robotic systems, represented as either lattice-based or swarms, are intelligent autonomous machines with interconnected modules, interactions, and algorithms that allow for flexible deployment in the IoRT and the creation of the IoMRT. This study examines and investigates the idea of the IoMRT, with a special focus on lattice-based self-reconfigurable modular robots and robotic systems. It discusses their architecture, design, criteria, and

safety, security, and privacy considerations. It also identifies the main issues and current adjustments and improvements while providing predictions for future advancements, namely in bolstering and reinforcing security against any physical or cyber threats. The integration of robotic systems and MSRR into the IoT under the IoMRT framework offers a range of solutions that address different industries, including law enforcement, armed forces (i.e. military use and counter-terrorism/insurgency operations), and the medical field. These systems outweigh the constraints of traditional robotics with their self-reconfiguration and self-healing capabilities, providing improvements in terms of power consumption, time efficiency, reusability, and adaptability to complex environments with little assistance from humans.

In our thesis research, we placed a significant emphasis on communication, recognizing its key role as the most essential component of programmable matter. Our focus primarily revolved around optimizing communication efficiency, particularly with regard to message length and the associated time considerations. We covered various aspects such as computation and sensing, aiming to optimize communication processes and reduce latency. By addressing these factors, we aimed to enhance the overall performance and functionality of programmable matter systems, ensuring seamless interactions and operations in diverse environments.

Lattice-based modular robots are composed of modules arranged on a lattice and forming 3D shapes, if these robots are small enough and many enough, they form a programmable matter [342]. This work proposes a method for optimizing data communication times between modules by compressing the data. We first analyzed the communication delay between the end device modules, and then a set of recent lossless compression algorithms was tested to select the optimal one to implement with *Blinky Blocks*. Based on the results obtained, we propose to add a lossless data compression scheme to reduce the communicated data size and consequently communication delay. We found that the "Brotli" compression algorithm is the most suitable one for modular robot communication as it achieved a good balance between computing and communication overhead. Then, based on the compression ratio and the communication delay interpolation, a significant gain is achieved by reducing the communication delay by a factor of 5. This protocol was already used in several published papers including [35], and is constantly used by our colleagues.

The introduction of nanorobots as part of modular robots within the IoT domain offers cutting-edge technologies that revolutionize various IoT fields mainly including military, law enforcement, manufacturing, disaster response, industrial, and healthcare domains.

The adoption of modular nanorobots, as part of programmable matter, led us to introduce a new novel concept called the Internet of Programmable Matter of Things (IoPMoT), their adaptable and reconfigurable nature allows their versatile applications in complex environments by fostering real-time data-driven insights and automation. As a result, this work proposes a new Lightweight Cryptography and Authentication Protocol for BBs (LCAPBB), as an enhancement of PROLISEAN, which is within the realm of Lightweight Cryptographic Algorithms and Protocols for Programmable Matter (LCAPPM), and presents the main flaws of this solution and offers proper enhancements by using a solution protocol based on cryptographic approaches (see Table 4.1). This whole proposed concept can be summarised in the following Figure 1.1.

As part of the ongoing future work, we are already working on a lightweight cryptography algorithm to be added to "Brotli", to form a secure compression based on the crypto-compression concept, which allows us to achieve two main objectives. The first one was already achieved in terms of reducing communication delay and overhead, and the second objective is to secure the compressed message to prevent it from being intercepted, altered, and modified. Other future work includes further research on other compression algorithms that cover other than textual data including media (video, audio) or image data. Further research will also surround the introduction of Artificial Intelligence (AI) into modular robots, especially Blink Blocks to achieve higher accuracy and precision in terms of performance, security, and efficiency. Lightweight authentication solutions are also to be further researched before being applied and integrated into the BB module.



CONTEXT & PROBLEMS

CONTEXT AND PROBLEMS

2.1/ CONTEXT: MODULAR ROBOTIC SYSTEMS

Modular robotics can be described as smart autonomous objects that come in the form of a 2D or 3D shape, capable of sensing their surrounding and communicating with each other via actuators to perform a variety of physical tasks (i.e moving), visual (i.e colour changing) or/and audio tasks (i.e sound) [440]. Additionally, these modules are embodied with memory and computational capabilities, necessitating programming to establish correlations between sensor inputs and environmental actions. For this reason, it is important to understand its main components, along with the state of the art of self-configuration algorithms and underlying models in modular robotic and self-organizing particle systems to know how to introduce them into the IoT domain.

2.1.1/ COMPONENTS OF MSRR

Before proceeding any further, it is important to briefly present and highlight the main components of modular self-reconfigurable robotics and robotic systems to be more familiar with the MSRR and the MSRRS concepts ahead of their appliance into the IoT domain. The main differences with classical distributed systems like sensor networks are the number and the density of modules. In a modular robotic system, we consider thousands to millions of small communication objects in the same area (or volume). That communicates to reach a global goal. This is achieved when the modules engage in inter-communication, selectively halting specific information exchanges, and subjecting them to analysis before disseminating fresh data to other modules. The importance of this part is to highlight the main IoT components especially the sub-components that form the key parts of both hardware and software, to gain more insights and details about these components, how they work, operate, and interconnect. For instance, modules may undergo a selection process based on a global criterion, such as the center of the system, leading to the formation of clusters or orchestrating module movements to construct a

new geometry. Therefore, we present these IoMRT components as follows:

Hardware Components: Includes the physical components of any electronic device(s) in any IoT domain, and in many cases can act as a protector to the softer component parts. Thus, offering a tamper-resistant protection, and performing a physical execution of commands issued by the software. **Sensors:** Aim to detect both events and changes in its environment via sensing before sending the sensed information to the computer processor. Sensors are used in everyday objects and for real-time IoT applications. Sensors can either be active or passive, analog or digital. They can also be light or sound sensors, heat, movement, or noise detectors, contact sensors, ultrasonic distance sensors, etc. **Actuators:** Are systems that alter the environment of the module. It can produce soft alteration generating light or sound. However "movers" have more effects by participating in the motion of the module (by rotation or translation). Actuators are usually divided into three main types which are hydraulic, electric, and pneumatic [203, 277]. **Computer Processors:** Include a Central Processing Unit (CPU) to control the electronic circuitry that executes the issued commands and instructions to perform arithmetic, logic, and input/output (I/O) operations. In IoT, processors can take many types such as micro-processors, micro-controllers, embedded processors [88], and (next generation) digital signal processors [362]. They are also combined with memory to store the code to run on its local data. **Power Supplies:** These Are electrical devices including Power Supply Units (PUS) that regulate the incoming power voltage and frequency to supply robotic systems with safe-for-use electrical power. Power supplies can be standalone or built into the robotic system or its IoT devices. **Other Hardware Components:** That make up the logistics or/and mobile part of any given modular robot, such as wheels, magnets, chains, etc, as well as the cover shell that protects the vulnerable components from any physical shock or interaction.

Communication system: Combines sensor and actuator to permit exchange of message between systems. It may be a point-to-point communication system or a wireless system for long or short-distance communications. This also covers the integration of hybrid systems conjoined with clusters, wherein communication channels operate in a point-to-point fashion, facilitating wireless data exchange. **Intra-Communication:** Communication inside the group of robots, based on sharing data with each other. **Inter-Communication:** Communication between clusters and with a central unit to ensure a closer and stronger communication link.

Software Components: Include a set of instructed data and/or programs that are used to operate via hardware to execute specific tasks. The software includes a set of scripts,

programs, and applications that run on a specific device linked to modular robots and robotic systems in IoT. Software categories are divided into two types: application software and system software which are used to design robotic hardware or perform specific tasks. **Operating Systems:** Which are IoT system software that manages the Modular robotics hardware, and software resources, and provides them with common services, commands and tasks. **Firmware:** Is a tangible and mostly updatable electronic component with embedded IoT system or/and software instructions that issues a command that notifies how an electronic device must operate. Firmware includes the full Basic Input/Output System (BIOS) type. **Applications:** Can be introduced by configuring modular robots to perform a variety of complex tasks such as construction/deconstruction, assembly/disassembly, attaching/detaching, and this is due to their modular nature and properties. After highlighting these components and discussing them, the work will focus on analyzing the main IoT characteristics.

2.1.2/ CHARACTERISTICS & PROGRAMMING HYPOTHESIS

In IoT, unlike robots, modular robotic systems have unique characteristics that allow them to be the most adopted robots within the modular robotic domains in complex environments and different scenarios. Therefore, it is important to discuss and analyze them to add more insight and information in this regard. As a result, these characteristics are presented as follows:

- **Unique ID:** Each modular robot has an Identification (ID) that is unique and invariant to them to distinguish them from their other connected robots in IoT. However, it is still possible to modify an ID on a robot.
- **Memory:** Modular robots have memories or/and processing units that allow them to self-reconfigure once obstacles are detected, as well as use intelligent methods to sense their surroundings to identify the location of their neighbours.
- **Communication:** Modular robots can communicate with their connected neighbours by exchanging messages through sending and receiving processes.

Since these characteristics are presented, the next part will discuss the main modular robotic classification.

2.1.3/ MODULAR ROBOTIC CLASSIFICATION

Unlike other robots, modular self-reconfigurable robots have their unique characteristics and properties which make their classification more accurate and much easier. In this thesis, this classification is based on the nature of movement, structure, and module components, along with their communication nature and is represented as follows (see

Figure 2.1):

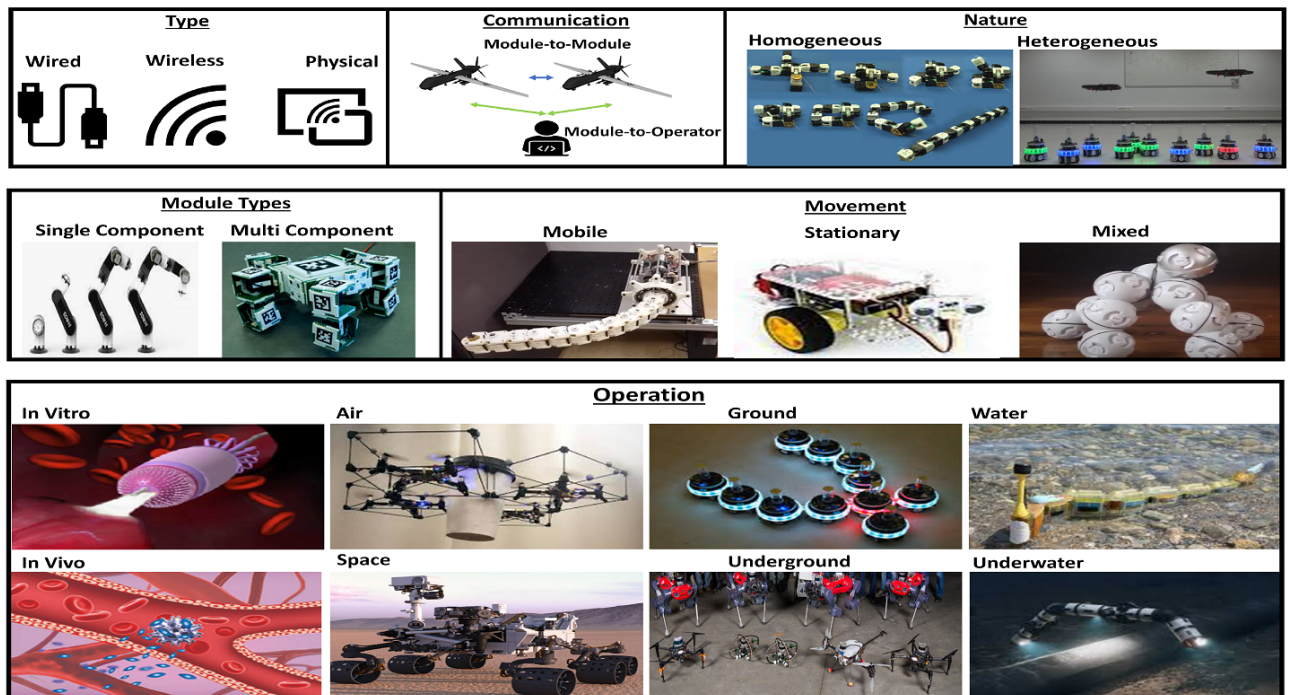


Figure 2.1: Classification of Modular Robots in terms of IoMRT.

Structure: Include how modules can move or slide over a given terrain to achieve the intended modular robotic shape. For this reason, three main movement types are presented and described as follows: **Cyber:** Which allows modules to operate wirelessly, without the need for any physical interaction in order to maintain a communication link between each other [478]. This is similar to wireless communications between robots especially Unmanned Aerial Vehicles (UAVs), Unmanned Underwater Vehicles (UUVs), and Unmanned Ground Vehicles (UGVs) [358, 258]. **Physical:** Which allows the interconnected modules to slide through physical interaction, of which without it, there will be no communication. **Cyber-Physical:** Which is the hybrid communication type, where module communications can be maintained either physically or virtually (wirelessly). In fact, it is a useful method to overcome availability issues, but at the cost of performance.

Movement: The nature of the movement of modular robots is that they can be stationary, mobile, or flexible. This is further explained below as follows: **Stationary:** Modular robots can be stationary, meaning they are fixed without the ability to move. They are mostly used in medical robotic operations [488] and industrial tasks [489]. A similar concept includes the Blinky Blocks that perform stationary operations, meaning they are fixed, cannot move, and can operate on the spot [31]. **Mobile:** Modular robots can also be mobile with the ability to move freely due to their self-reconfiguration capabilities

which allow them to resize and reshape to overcome obstacles. They are primarily used in law enforcement and military operations, search-and-rescue operations, as well as for disaster response and forest navigation without crashing [508]. This includes but is not limited to two different examples, including kilobots (move freely without interconnection) and Roombots (motion in a grid) [270]. **Mixed:** or hybrid, where modular robots have both properties, which allow them to be stationary when required and mobile when needed, in cases like exploration, climate monitoring, reconnaissance/surveillance operations, etc.

Modules Types: Modules are made up of one to many components to form the intended modular robotic system. These modules can either be interconnected to form a single modular robot or split into lattice-based swarms to form a modular shape or resize. **Single Components:** Where several modules tend to form a single united modular robotic body which allows the modular robot to resize and reshape, without being able to detach from the modular robotic body once it formed (i.e NSK Robot Module and single component Angular modules) [193]. **Multi-Components:** These Are usually located to form a singular modular robotic body, but at the same time, they are capable of detaching to achieve a more flexible size, or even reshaping. In this case, modular robots can either attach or detach to achieve the intended form (i.e Atro (Automation Technology for Robots) and EMERGE (Easy Modular Embodied Robot Generator) modular robot [288]). **Mixed Component:** Are very similar to the multi-components parts with an addition that allows them to split into several sub-modular robotic systems to perform several tasks simultaneously, or achieve one united task by forming a united modular robotic body (i.e SMORES [255] and Modquad [382]).

Communication Nature: The nature of communication between modular robotic systems is based on two main interconnected types to form a strong communication link and maintain this channel. These two communication types are presented below as follows: **Module-to-Module communications:** Or M2M-Com are based on point-to-point or peer-to-peer (neighbour-to-neighbour) concepts where two or more modular self-reconfigurable robots can communicate with each other either through physical interaction (i.e Claytronic's Blinky Blocks [302], Sliding cubes, 2D/3D Catoms/Datoms [346], Hexanodes, etc) or wirelessly, or even both. Other cubic P2P examples primarily include Telecubes [426], Miche [160], Pebbles [159], Cubelets [87], and Kubits [181]. The advantage of P2P communication is that it gives information about the presence of neighbours, if a neighbour answers these messages, it means that it is present. However, there's one key drawback which is that the quality of alignment and attachment of the modules is very sensitive. Wireless communication tends to mitigate the previous problem. However, broadcast messages need to filter these messages based on the ID of the sender, except that a high number of simultaneous communications may result in interference.

Module-to-Operator communication: M2O-Com or Human-to-Module (H2M Com) is the communication type where all modules are also connected to the operators who issue the commands and (semi)instruct how the modules should behave. This communication nature can either be a Broadcast: One-to-All (OTA) or Multi-cast: One-to-Many (OTM). This includes but is not limited to Kilobots and droplets. This is further explained in Figure 2.2 and Figure 2.3, respectively.

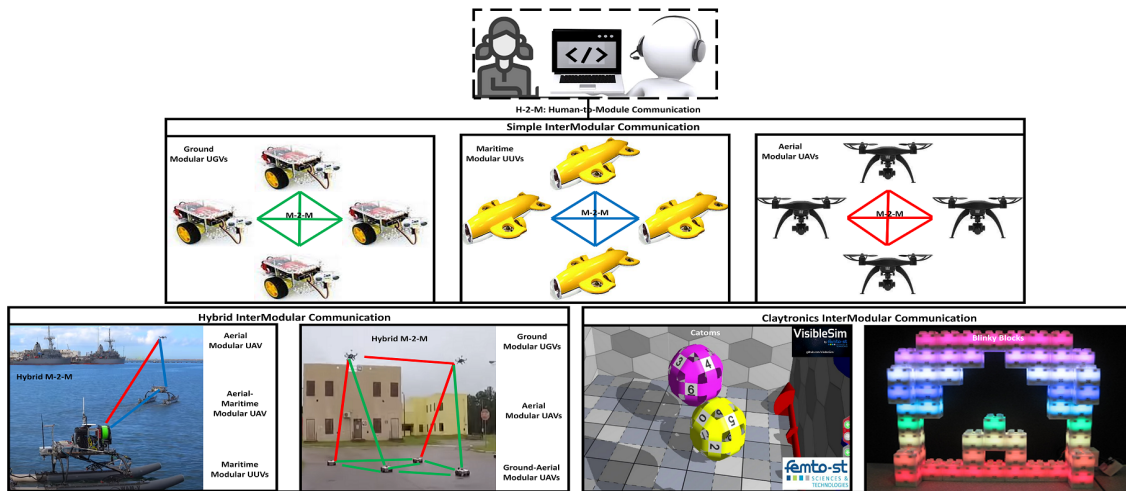


Figure 2.2: Different Modular Module-to-Module (Swarmanoids) and Human-to-Module Communication links.

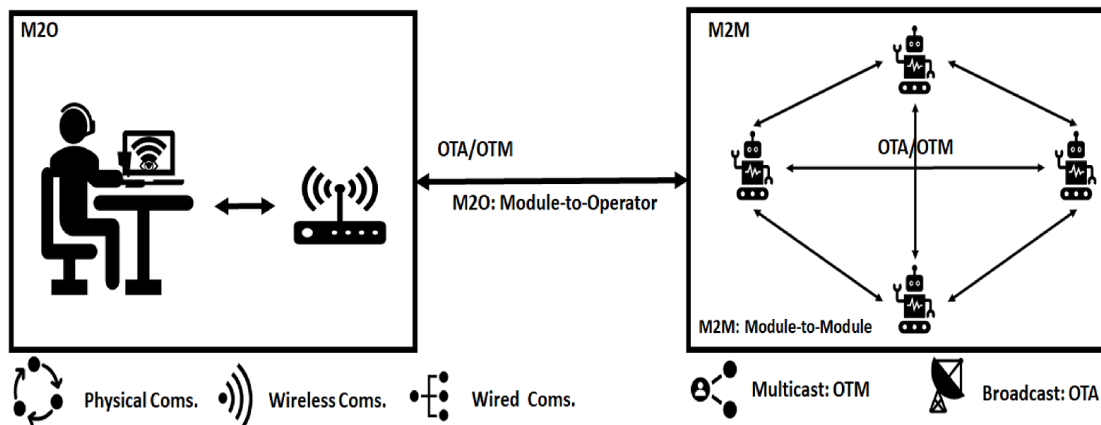


Figure 2.3: Communication Between Operators and Modular Robots.

Communication Type: The communication type is also important to be highlighted as the future of IoMRT and swarm robotics evolve around it, especially when we have a different type of robots communicating together to achieve a much more accurate Collective and Collaborative Behaviour (CCB), as part of the future of Internet of Swarm Robotic Things (IoSRT) which once combined, it will surely introduce the Internet of Modular Swarm Robotic Things (IoMSRT). In other terms, unlike the traditional homogeneous

nature of robotic systems, the heterogeneous nature of Swarm robots and MSRRs, will be characterized by their unique reliance on communication that allows it to communicate together and other with different robot types. **Homogeneous Nature:** Includes the communication between the same type of modular robotic systems, especially lattice-based and swarms robotics. Mostly robots designed by the same company, or robots that operate over the same terrain, similar to the Swarm intelligence (SI) of bird flocks (such as the Bionicswift bird-like robot swarms [256]), as well as bees and ants colonies [39]. **Heterogeneous Nature:** Includes having different modular robotic types from different companies being able to mix and communicate with each other, or with other manned/unmanned robots, forming a heterogeneous swarm robotic nature. This is similar to the spatially targeted communication method for (swarm) multi-robot systems, presented by Mathew et al. in [275]. Another version of heterogeneous communication includes cross-domain modular robots that operate over different terrains (ground, sea, and/or air), communicating together and sharing data with each other for enhanced decision-making. This is primarily used in military (counter-insurgency/terrorism) operations (Joint Surveillance Reconnaissance Target Acquisition (JSRTA)) and search-and-rescue operations. As a result of this Heterogeneous Robotic Swarms type, a novel study concept called "Swarmanoid" [113, 115] was introduced, tested and evaluated [114, 63].

2.1.4/ MODULAR ROBOT TYPES

Shapes of a modular robot may be very different being the result of design with different goals, some of them are made to slide [223, 266], other to roll [469, 472, 271], jump [251, 153], fly [66], swim [89, 41], carry loads [509, 254], or even self-disassemble such as the presented work by Gilpin et al. in [90]. This allows them to be able to reshape themselves depending on the desired design via self-reconfiguration. Therefore, it is important to present and briefly describe them in order to be more familiar with them.

Modular UGV/UUGVs: Includes soft/hard modular robots (ranging from nanometers to meters in length, size, and height) which were also deployed to further extend the swarm robotic (IoSRT) ground capabilities, especially in search and rescue missions, emergency services/first responders, disaster management, construction tasks, as well as medical and military operations. As a result, different architectures and modeling concepts were issued and designed, with more Modular Unmanned Ground Vehicles (MUGVs) and Modular Unmanned Underground Vehicles (MUUGVs) prototypes undergoing realistic tests before their final deployment. Some examples can be seen in Figure 2.4. **Chain-like Modular Movement:** offers the ability to perform the same movement of a tank/armoured personnel carrier (APC) such as moving forward, and backward, as well as rotation and redirection. Mostly used in military, law enforcement, and search and rescue operations.

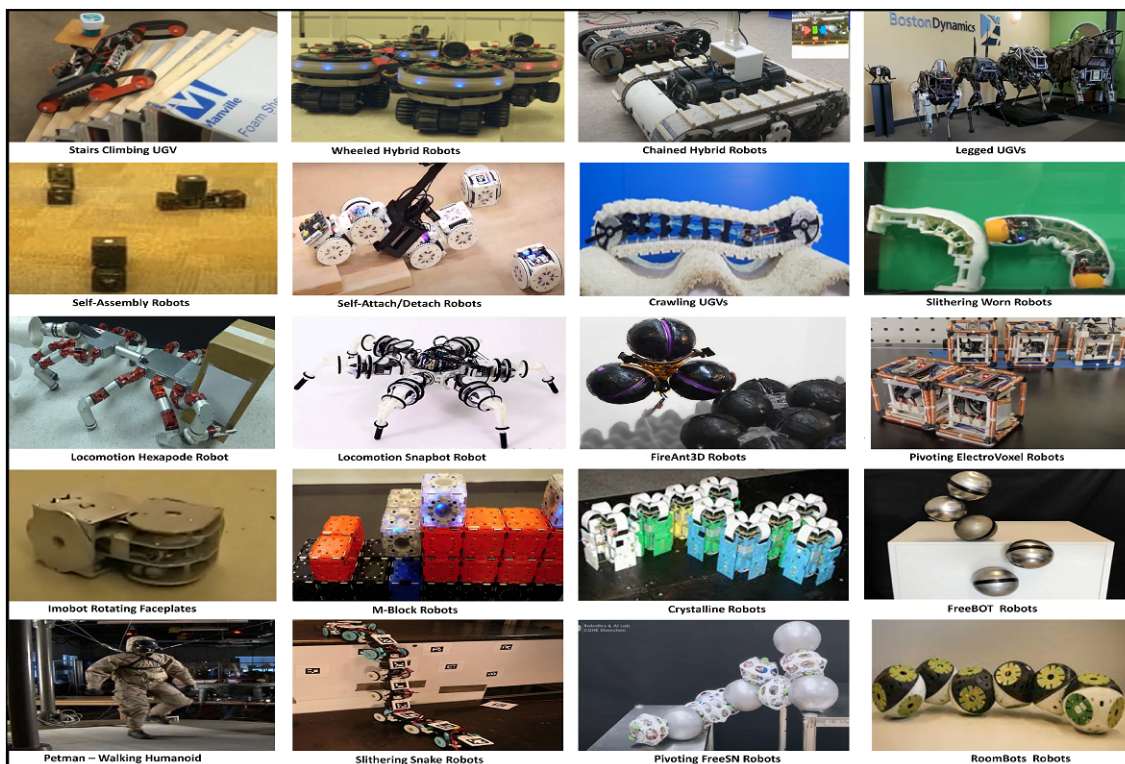


Figure 2.4: Some examples of Modular Unmanned Ground Vehicles.

Wheel-like Modular Movement: is the most adopted modular movement, as robots are assigned with wheels to perform car-like movements, which offer an enhanced version of the movement, with less power consumption and higher speed. It can be used for a variety of tasks aside from the military, law enforcement, and search and rescue operations, such as shopping, food delivery, etc. It can also take the form of a hybrid wheel [410].

Leg-like Movement: legged robots have the ability to move like either crabs or spiders, depending on the number of legs to perform the intended movement, and whether it is a soft or hard modular robotic structure. In this case, the modular robotic design can vary from at least one leg to more than 8 legs on average. But it can also reach a much higher number of legs. Some of these modular robots include but are not limited to BEX's Kawasaki rideable robot goat [412], nor Spot [52] (a Russian armed version was presented called robot dog m-81 series), Cheetah [398], and Bigdog [360] which are developed by Boston Dynamics.

Jumping Movement: modular robots are now modified to have the ability not only to jump in the form of humanoid, but also to cover a higher height and longer distance. This is a technique that is being primarily used by the military and adopted by law enforcement to counter and overpower armed criminals and terrorists alike.

Walking-like Movement: modular robots were adopted to form what is known as "humanoids" that have the ability to perform human-like movements such as moving joints, as well as walking, moving, talking, sitting, etc. Some of these modular robots include but are not limited to the most known Petman and Atlas humanoids, developed by

Boston Dynamics [306, 307]. Paik et al. also presented an autonomous multi-locomotion insect-scale robot called millirobot, inspired by trap-jaw ants, capable of conducting horizontal jumping for distance, vertical jumping for height, somersault jumping to overcome obstacles, walking on textured terrain and crawling on flat surfaces [500].

Slithering Movement: Is another technique that was adopted by modular robots by imitating the worms and snakes movements (i.e. HiBot) [172] by conducting a twisting smooth and unobtrusive surface or underground movement. This is mostly adopted in search and rescue operations.

Rolling Movement: Combines the rotation of a modular robot in the form of an axially symmetric object, which allows it to move over a given simple/difficult ground surface without obstacles.

Sliding Movement: Ensures a frictional motion between the modular robot and the surface with which it is in contact with, which is not the case with the adoption of the rolling movement.

Locomotion Movement: Allows modular robots to make a directional movement from one place/surface location to another to reach the intended/destined location, to perform a specific task such as reshaping or resizing. This also includes the presented reconfigurable swarm of identical low-cost quadruped robots, which are linked either on demand or autonomously, by Ozkan et al. in [323]. FireAnt3D is a well-known example, which was presented by Petras and Rubenstein in [430]. FireAnt3D is a 3D self-climbing robot capable of climbing its peers using non-latticed connections such as docks to locomote over arbitrary peers using 3D arrangements. In [37], Stoy & Nagpal presented a mechanical design and locomotion of modular-expanding robots with locomotion mechanisms such as crawling, rolling, and climbing and their possible applications in space exploration or search and rescue.

Pivoting Movement: Which allows modular robots (mainly Datoms, FreeSNs, 3D Cubes and ElectroVoxels [310]) to self-reconfigure by performing pivoting movements in order to reshape and resize [450].

Obstacle Climbing Movement: Mainly includes, jumping, pivoting and climbing of walls, stairs and ropes to perform tasks that are deemed too challenging to normal and traditional robots. Hence, the reliance on the modular robots to perform this task.

Hybrid Modular Movement: Includes the modular robotic ability to perform any of the ground movement, or mixes the ground movement with aerial or/and surface/underwater activity. Hence, the adoption of the Hybrid Modular Unmanned Ground Vehicles (HMUGV) and the Hybrid Modular Unmanned Multi-Terrain Vehicle (HMUMTV) concepts. Some examples can be seen in Figure 2.5. A well-known example is that presented by Romanishin et al. in [368, 367], where the self-transforming M-Blocks (2.0) robots can jump, spin, flip, and identify each other. Another example includes the presented crystalline robot system by Rus & Marselette in [376], where the robotic movement is based on a set of modules that aggregate together to form the distributed robot systems by actuating, expanding and contracting each unit via self-reconfiguration [377]. Another novel concept for modular robots was introduced by Roderich et al. in [117], called Modular Fluidic Propulsion (MFP), that routes fluid through themselves in order to move.

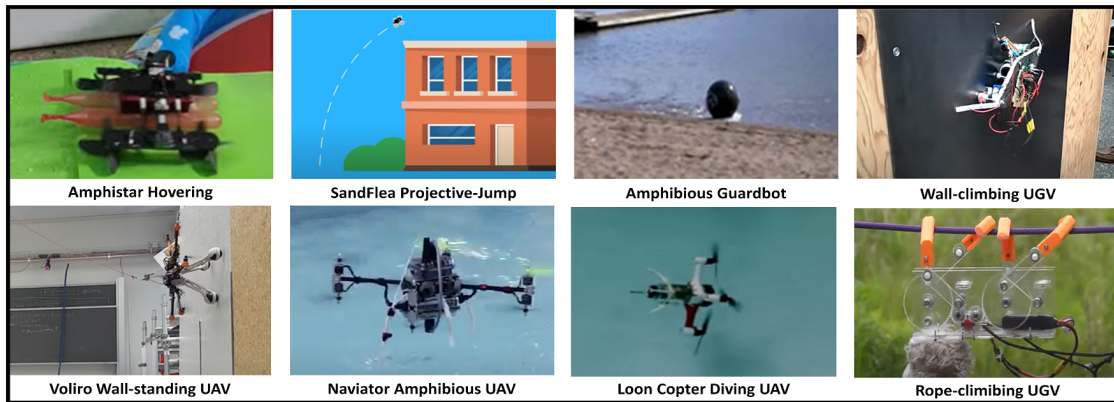


Figure 2.5: Some examples of Hybrid Multi-Terrain Unmanned Modular Vehicles.

Modular UAVs/Drones: (soft/hard) modular robots also evolved to revolutionise the drone/UAV concept, with the introduction of several UAV modules which can be combined to form a single UAV body, or operate in swarms as part of loSRT. Such a modular robotic type ranges from millimetres (i.e Black Hornet variants) to centimetres (Vapor, THOR, Firefly, and BUG variants) and meters in length, size and height. Some examples can be seen in Figure 2.6, which include the airblock quadcopter that uses the Makeblock software [386], HYDRUS 2D Transformable Drone [403], ModQuad/ModQuad-DoF [382, 147], and Pico quadcopters tested in [292] on collision avoidance. In fact, it is important to highlight the following novel modular robots, including the hovering DRAGON UAV, which is a novel modular robot created by Zhao et al. capable of performing multi-degree-of-freedom (DoF) aerial transformation [505]. Another example is the novel Aspect Ratio-Modular Vertical Take-Off and Landing (ARM-VTOL) aerial robot created by Carlson et al. [66], capable of performing both VTOL and Fixed-Wing hybrid missions once combined via magnetic coupling.

Modular FW: Modular Fixed Wing (MFW) include modular drones with the ability to rely on the module concept to perform challenging aerial tasks (i.e search, rescue, demining, reconnaissance, surveillance, etc) depending on weather conditions (i.e storm, wind, rain, snow, etc). **Bird-like UAVs:** are modular robots that imitate the movements of birds, as well as take their shape and size, with a realistic appearance too if needed. They can be launched in swarms too. **Modular QCs:** Modular Quadcopters (MQCs) are usually modular drones in a form of swarm, that once added together, they can form a single uniform a quadcopter rotary wing, with more power and force. They are or can be used for deliveries, reconnaissance/surveillance, bomblets dropping, artillery spotting or intelligence gathering. **Modular UAGC:** or Modular Hybrid Unmanned Aerial-Ground Vehicles (MH-UAGC), including modular robots that can fly and at the same time operate on the ground to perform ground task/work, or move against the wall (i.e Voliro and FSTAR variant) [279]. **Modular HAUV:** or Modular Hybrid Amphibious Unmanned Vehicles (MH-

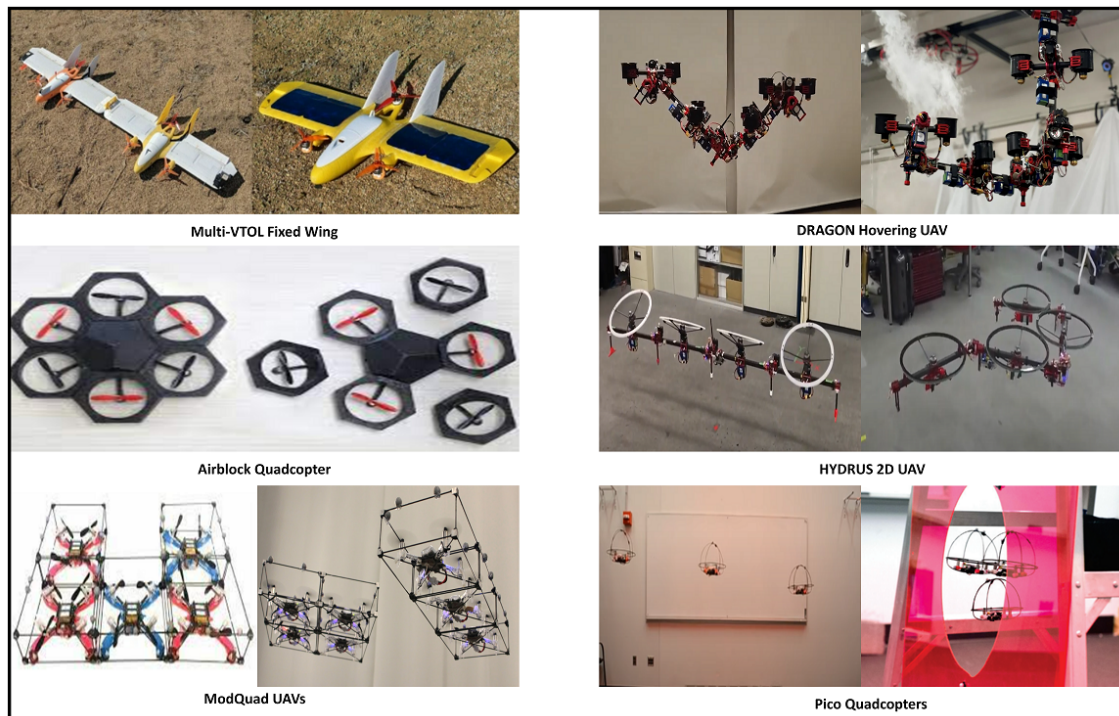


Figure 2.6: Some examples of Modular Unmanned Aerial Vehicles.

AUV) such as Hybrid Unmanned Aquatic-Aerial Quadcopters (H-UAAQ), which is or will be a new form of modular robots that are capable not only of flying, but also of hovering above the water, on the water surface and also underwater (i.e Eagle Ray, Loon Copter [15] and Naviator variants [269, 280]). Primarily used for bio research, search and rescue, and military/navy (amphibious) operations. Another well-known example is Salamandra robotica II, the new generation of amphibious salamander-like robot, presented by Crespi et al. (including Auke Ijspeert) in [91], capable of both swimming with foldable limbs (like a lamprey robot [90]) and walking.

Modular UUVs/USVs: (soft/hard) modular robots also revolutionised the Remote-Operated Surface Vehicles (ROSVs) and Remote-Operated Underwater Vehicles (ROUVs) aspects, bringing a new whole advanced meaning to the swarm robotic domain as part of IoSRT. As a result, a new modular amphibious version is being adopted now to replace the traditional "maritime/underwater" robotic version, introducing the Modular Unmanned Underwater Vehicle (MUUV) and Modular Unmanned Surface Vehicle (MUSV) concepts, respectively. Some examples can be seen in Figure 2.7. **Dolphin/Fishlike Motion:** relies on wriggling its body and mainly modular tail to create a realistic fishlike movement, instead of relying thrusters for movement or rotation [293]. **Starfish-like Motion:** aside the Jellyfish-like movement, this one is achieved by relying on modular soft robots, which can be made or are not made of silicon foam, with the ability to move using

a single low-powered actuator or by relying on the extended tentacles or arms from its central disk-like body [208]. **Octopus-like Motion:** includes the adoption of soft modular robots to conduct an octopus-like movements including the adoption of similar size, shape, forms and arms/tentacles movement [143, 296, 399]. This is primarily adopted and used for bio/maritime researches. **Stingray Motion:** which allows modular robots to have the needed flexibility to perform complex motion using simple servos, linkages and a microcontroller [210], or by using an electro-mechanical software design to ensure a smooth underwater movement [27]. **Amphibious Motion:** which gives modular robots the ability not only to move on dry land, but also the ability to hover over water to also perform a supplementary task. **Submerging Motion:** such as testing modular robots underwater in order to check its performance, or in order to try and imitate sea-life, (i.e fish, anguilliform fish, snakes, starfish, stingrays, etc) before further testing. **Surfing Motion:** modular robots have the ability to operate on the surface of the water to perform necessary task such as search and rescue, military operation or biological/geological operation. **Diving Motion:** ensures that modular robots have the ability to perform exploration tasks [38], excavation tasks, search and recovery operations, as well as covert (sabotage/espionage) tasks without being detected.

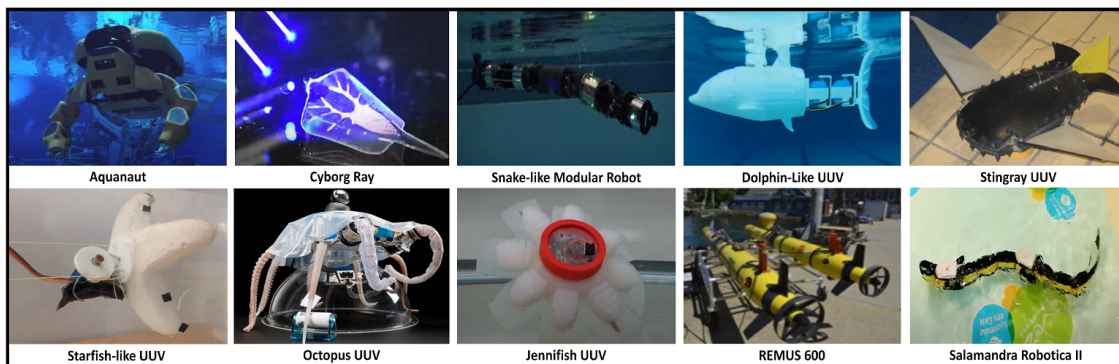


Figure 2.7: Some examples of Modular Unmanned Maritime, Aquatic, and Underwater Vehicles.

After highlighting the main modular robotic types, it is also important for us to discuss the taxonomy of their architecture.

2.1.5/ TAXONOMY OF ARCHITECTURES

The architecture of modular robots is divided into several categories of modules that make the main IoMRT concept or modular IoT systems. These categories are further described in [7, 58]. In this subsection, we aim to identify and discuss them as follows:

Chain Architecture: allows the mutual connection of modules with Degrees Of Freedom (DOFs) to form either complex or flexible structures including loops and trees. Despite being highly flexible, the Chain architecture suffers from several drawbacks such as complex control, computation, and coordination. In other terms, in chain-based architecture, units can reach any point due to their versatile IoT nature. However, unlike the lattice-based architecture, it is unable to accomplish any reconfiguration step due to complex computation.

Deterministic Architecture: is adopted as a deterministic reconfiguration for macro-scale systems where each system's units are either directly moved or manipulated to achieve their required targeted location. Its reconfiguration time is known, and a sophisticated feedback control is strictly recommended to achieve a precise manipulation.

Free-form Architecture: allows free-form systems to aggregate modules in semi-arbitrary positions [159]. This architecture is neither chain nor lattice-based architecture. As a result of free form, Liang et al. developed a FreeBOT as a novel MSRR that can freely be connected with lesser physical constraints to achieve the main tasks of module-independent movement, connector management, and system reconfiguration. The connection between FreeBOTs is instant and genderless due to the freeform and fault-tolerant connector [253].

Hybrid Architecture: unlike the free-form architecture, is based on the combination of both chain and lattice-based architecture. Hybrid modules can interconnect while adopting lattice structures.

Lattice Architecture: can adopt a variety of 2D and 3D modules including hexagonal, polygonal, and rectangular patterns, as well as other complex lattice-based shapes such as sliding cubes. Lattice-based modules can traverse but not move via grid cells. Unlike the chain-based architecture, the Lattice-based architecture type is often adopted for both modeling and problem solving such as the the reconfiguration planning problems. In fact, Lattice-based architecture offers a simple computation and reconfiguration representation and planning and can be scaled easily in complex IoT systems, where a few units are enough to achieve the required reconfiguration step.

Mobile Architecture: In mobile architecture, modules can freely move in an independent manner in a given IoMRT environment. Modules can also attach one to another to form a complex chain or lattice-based structure.

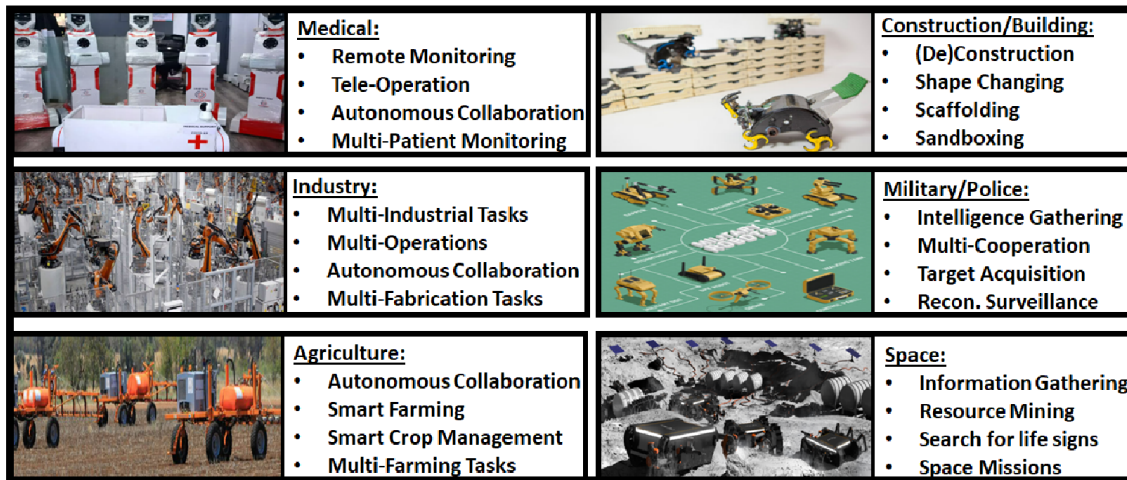


Figure 2.8: Modular Robots in main IoT-related domains and fields.

Stochastic Architecture: Is suitable for micro-scale IoT systems which use stochastic reconfiguration where each unit uses a statistical process to move. The location of each unit is only known upon its connection to the main structure. The time of each reconfiguration is only guaranteed in a statistical manner.

Truss Architecture: Uses both stretchable and contractible (passive/active) struts to form Truss-shaped structures. The adoption of struts in this architecture type is to achieve the ability to reach the required configuration without being affected or prone to any limitations or challenges. **Active Struts:** Are stretchable and contractible struts with a variable length. **Passive Struts:** Are stretchable and contractible struts with a fixed length. **Joint Component:** Acts as a linking medium for both struts. Afterwards, we will discuss and present the IoMRT-related applications based on the taxonomy of the architectures presented above.

2.1.6/ IoMRT-RELATED APPLICATIONS

Due to their advanced natural ability to reshape and resize autonomously, modular robots started seeing increased adoption in the IoT domain, aside the biological (i.e octopus arm [244]) domain, especially in industry, medicine, agriculture, search and rescue, tourism and entertainment, combat, construction and deconstruction. This is illustrated in Figure 2.8. As a result, in our thesis, their adoption is highlighted and presented as follows:

Industrial IoT: Especially Cyber-Physical Systems (CPS), which along with the development of industrial automation have managed to achieve a faster and safer fabrica-

tion/production through its production lines. The adoption of modular robots in the industrial field offered several advantages, which helped sorting shape and size issues of fragile modules and irregular objects. Other advantages include fault tolerance, scalability, low maintenance, reconfiguration capabilities [129] and adaptability, along with a reduced industrial production cost [265].

Medical IoT: soft/hard modular robotic technologies saw a remarkable adoption in the medical field, especially for medical treatment and surgical operations. This includes the adoption of new procedures and techniques such as the design and fabrication of soft units for surgical manipulators, medical (soft) sensors, the adoption of the Minimally Invasive Surgery (MIS) for abdominal interventions [99], as well as many other soft robotic medical applications presented in [199]. One of their main advantages include a high degree of freedom, with life-like material properties, ability to produce/reproduce the joints' motion in humans and animals, in addition to lightweight wearable devices and suitable surgical and rehabilitation devices for all patients [337, 474]. In fact, modular robots were also used to for in vivo and in vitro testing and treatment for human patients such as the case of the presented robot for Intravascular Treatment especially the Chronic Total Occlusion (CTO) disease [326]. Other millimeter-sized were designed with an artificial brain and legs to be inserted into the human body's main artery or digestive organ, while researchers have developed a highly advanced miniaturized smart robots designed to change their shape as they encounter different fluids that can deliver drugs directly to the diseased tissue [202].

Agricultural IoT: modular robots and robotic systems tend to be highly reconfigurable, which makes them suitable for deployment in all weather conditions and over different challenging agricultural conditions (i.e mud, dirt, steep etc) and terrains (i.e open fields, farms, tunnels, greenhouses, and polytunnels). This also include the use of modular robots to ensure a safer yet faster food production and management (i.e livestock). A prime example of that, is the Thorvald II, a high-quality agricultural robot that was presented by [169] to allow a quick customisation for a specific application in a specific environment.

Disaster Management: due to their flexible and tamper-resistant nature, modular robots can also fit in small tiny places that cannot be reached by humans [501]. This is one of the main reasons why this modular robotic type is being used for disaster management purposes such as exploration, inspection, reconnaissance, surveillance, and monitoring, as well as search and rescue tasks and purposes [182].

Tourism and Entertainment: modular robots are more and more now being deployed to enhance tourism, to entertain the people, or to help children with their intellectual development [265]. For example, Lee et al. introduced a modularised design for soft robotics in [245] in order to try and create toys using soft modular units. The introduced design is inspired by LEGO, and other toy robots such as soft gripper [187, 405] and scorpion-inspired robot [449]. Also, drone light shows which are performed by a synchronised group of illuminated and choreographed swarm drones which allows them to achieve the intended aerial formations or desired shape within a well-defined frame, such as the light shows in China [26].

Military IoT: modular robots which are part of IoMRT/IoSRT are seeing an extensive deployment (as early as world war I (i.e automatic planes) [85] and post world war II, using wired or radio-controlled robots and teletanks [92, 127]) on a variety of terrains, timing (day-time/night-time), and weather conditions, as part of wider military operations (i.e covert/overt surveillance (RoboBees [56, 475]), spying, reconnaissance, surveillance, search and rescue, casualty/equipment recovery, medevac, extraction, reinforcement, supplies, demining/mine hunting, bomb defusing/removal, ordnance disposal, navigation, adjustment, sabotage, support, sentry, etc) and battle fields [413], to cover specific mission-need systems [304]. Thus, becoming the spearhead of the Military IoT domain, especially with the introduction of swarm robotics [464]. A prime example is the iRobot PackBot, which is a combat-tested, man-portable UGV, that saw deployment in Afghanistan and Iraq, and a Griffon prototype which is a combined UGV/UAV [490]. Also, SENTINEL-M, a robot that operates as part of individual or swarm robotic military formation(s) to ensure live fire training for troops. Other main operations include precision/Guided Shelling, Counter Arms and Drugs Smuggling (CAADS), Counter Armed/Unarmed-Border Infiltration (CAUBI), anti-submarine warfare, counter-piracy, (multi) bombs dropping/bomblets drooping (including modified US/Russian grenades, thermobaric/vacuum or fuel bombs, Rocket Propelled Grenade (RPG), mines, explosive charges, and Mortar rounds) extensively used by both sides in Ukraine [185, 240] to target troop gatherings, trenches, sniper positions, spotters, artillery pieces, armour, armoured, logistics or/and infantry columns, transport/logistics, and light vehicles [180, 241], listening/visual posts, anti-air defense, munition depots, and small boats (i.e around the Dnieper island [228]) as well as for Surveillance Target Acquisition Reconnaissance (STAR), Joint Surveillance Target Acquisition Reconnaissance (JSTAR), Reconnaissance Surveillance Target Acquisition (RSTA), Intelligence Surveillance Reconnaissance (ISR), Combat Identification and Sensor Requirement (CISR), and Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) purposes [479, 484, 100]. This also includes convoy protection, Target Acquisition, Detection and Identification (TADI), Simultaneous Localisation and Mapping

(SLAM), and Moving Target Localisation and Tracking (MTLAT), in addition to guiding surrendering soldiers, locating missing and wounded personnel, exposing enemy hideouts and trenches (via drone bomblet/loitering attacks, reconnaissance, adjusting artillery and guiding trench defense/assault attacks), de-camouflaging/de-concealing units and drone-to-drone fighting as part of autonomous warfare [46, 186, 320]. Drone Swarms can also be used to reduce civilian casualties, minimize collateral damage, and eliminate "terror" targets via approved "Targeted Killing" (i.e. IDF's Dabla Unit) similar to the last Gaza Operation (May 2021 conflict [352]) and invasion 2023 (i.e. urban warfare, close quarters combat, and tunnel inspection/discovery). Swarms can also be used for real combat operations [204]. Many military-led swarm programs were developed, including the Distributed Battle Management (DBM) program [363], especially as part of the third offset US strategy [496] including the following main projects: Defense Advanced Research Projects Agency's Offensive Swarm-Enabled Tactics (OFFSET) swarms program [235], DARPA's OFFensive Swarm-Enabled Tactics (OFFSET) project [80], LOCUST project [387], Loyal Wingman project [511], British army's 2022 "Swarming Drone" tests on Salisbury Plain, along many others. Swarms of ALTIUS-600/700, Coyote [51], Uvify IFO-S UAVs and Aion Robotics R1 UGVs [354] were used at separate projects. In fact, swarm drones like Coyote UAVs are capable of carrying or being equipped with payloads (becoming swarm munitions [471]) or electronic warfare equipment and are mostly launched from aircraft (i.e. helicopters or fighter jets) or mobile/fixed multi-launching pads, which are mostly used. In fact, they can also be mounted and launched (or even land) on AI-based ground robots (i.e. ROBUST), drones, or unmanned surface vehicles. Swarm drone boats were also deployed by Ukraine against the Russian naval fleet, as well as allegedly targeting the Kerch bridge (July 17, 2023) that links the Russian-annexed (since 2014) Crimea [291] with Russia [335]. The first appearance of these swarm loitering drone boats [241] was reported in late October 2022, when the Sevastopol port that hosts the Russian Black Sea fleet was targeted by a swarm of Ukrainian Uncrewed Surface Vessels (USV), damaging a frigate and a minesweeper [240], in addition to other recorded and unrecorded similar events. In fact, this also includes the most recent introduction of the Military space force and the arms race for space control, outer space geopolitics, and interstellar space colonisation [209, 285].

Internet of Space Things: including space exploration and terrestrial deployments [9], as swarm modular robots (aside rovers, landers and satellites) saw astronomical uses and deployment in outer space and on celestial objects missions. Swarm robots and hive-mind algorithms are to be used by NASA as part of the annual Swarmathon Challenge, to mine the moon, excavate and build simple structures by 2025, ahead of their training on Earth before making them fully autonomous and collaborative [4]. A move that can be achieved by using a training model called Explainable Autonomous Robotic

System (HEART). This is an ongoing race between America, Russia and China. The exploration is not be limited to the moon, but to other asteroids as well, in search for water, ice, air, potential life, and rocket fuel, in addition to mining minerals, metals, helium-3, and other scattered valuable resources, and taking samples, pictures and videos for analysis [309]. In fact, Correll et al. created swarms of tiny ping-pong ball-sized robots called Droplets at the University of Colorado Boulder. These Droplets can perform lone singular or other complex collective swarm tasks such as containing oil spills or to self-assemble into hardware once separately launched into space [86].

Construction and Deconstruction Processes: modular robots have introduced Modular Construction (MC) as an alternative to conventional and traditional construction methods, due to their scaffolding and sandboxing abilities [40]. Thus, offering higher quality, efficiency, productivity, accuracy, safety, performance, and precision levels. This comes at the same time as ensuring a proper allocation of timing, resource requirements, planning, and collaborative managements [252] to achieve intelligent designs of future buildings [428]. For example, to avoid hazards of Modular Construction (MC) and Modular Integrated Construction (MiC), a Crane Safety Index (CSI) was developed to improve the Occupational Health and Safety (OHS) platforms [286]. After presenting the IoMRT-related applications where modular robots play a key part, it is important for us to highlight the main properties of modular robots compared to other types of robots.

2.1.7/ PROPERTIES: MODULAR ROBOTS VS. TRADITIONAL ROBOTS

In general, the introduction of modular robots and their flexibility in terms of size and shape, made them more reliable than other robots, due to their ability to overcome several key and main limitations of their predecessors. In traditional robots, the entire robot typically consists of a single unit with a fixed structure and functionality, whereas modular robots are composed of multiple interchangeable modules that can be reconfigured to adapt to different tasks or environments. This modularity allows modular robots to exhibit greater flexibility, scalability, and adaptability compared to their traditional counterparts. Additionally, modular robots often feature self-reconfigurability, enabling them to autonomously change their shape or configuration to better suit the task at hand. These differences result in modular robots being better equipped to handle diverse and dynamic operating conditions, making them particularly suitable for applications requiring versatility and robustness. In the following, we present these main specialties that characterize modular robots from other robots.

The first characteristic of modular robots, as their name suggests, is their modularity. As they are made up of separate components, communication between them is a key priority.

As a result, **Direct communication or Peer-to-Peer** is mainly used to avoid the need for a centralized point. It enables distributed systems to operate autonomously.

If we consider the network aspects, we can envisage several granularities. **Point-to-point** networks (BBs, M-Blocks, Kilobots, and FreeBOT) [230, 368, 372, 253] use a direct connection that densely links the modules making up the modular robot. In this case, the communication graph overlaps perfectly with the robot neighborhood graph. The **Communication Medium** can also differ, with electrical connection solutions such as those used by *Blinky Blocks*, or infrared connection solutions such as those used by *Kilobots* or *Droplets* [247]. The advantage of an electrical connection is that data can be transmitted quickly and reliably as long as the mechanical connection is good, whereas infrared communication can suffer from masking, reflections or light pollution.

Other systems use **wireless Communication**, enabling more global communications to broadcast the same information to all modules simultaneously. **Hybrid Systems** offer solutions with **Less Communication Density**. Consider point-to-point robot clusters where cluster leaders can communicate wirelessly.

The **data type** that can be exchanged can also vary enormously. *M-Blocks* use BlueTooth to communicate with neighbors and are able to detect their neighbor's ID reading a bar code, *Blinky Blocks* can send each other large, complex messages (e.g. up 227 B per message).

Now let's take a look at the power issues. There are many different solutions, each with its unique advantages and disadvantages. The first solution consists of **Embedding Energy** in the form of batteries in each robot (this is the case with Kilobots [372], M-Block [369] and FreeBots [253]). The advantage is that energy is fully available locally, without the need for a wired connection to the power supply but the autonomy is assured by regular replacement or recharging of the batteries, which can be tedious as the number of modules increases. Another drawback is the weight of the batteries, which can be a major constraint for mobile robots. The alternative solution is to build an **Electrical Network** inside the robot network and power each robot from this electrical network. This is the case with *Blinky Blocks*, for which the 6 connectors provide both energy and communication. Each *Blinky Block* is a node in the electrical network in which it participates, and this network is connected to the central electrical network by one or more power supply points. Linked to this second solution, **Power Sharing** is an important mechanism between the modules of modular robots since it allows the balance of power over the whole system. Energy must also be a resource that is conserved by the systems that drive the robots. Some activities, such as the use of certain actuators, are very energy-intensive and will unbalance the energy state of the whole set of modules. This consumption can

be rebalanced by dynamically reorganizing high-consumption activities. For example, in *Blinky Blocks*, the LEDs and the buzzer are the two most power-hungry actuators. Good energy management should limit the simultaneous use of these two actuators to maximum power.

In the case of self-reconfigurable modular robots. Additional features have just been added. **Docking Mechanism** allows any module to independently connect or/and disconnect from other modules while ensuring the module's configuration remains operational in case of any connection failure. **Fault Tolerance** is also adopted with a well-defined margin for accepted misalignment to ensure an accurate design of modular robots with an accurate performance to achieve the intended tasks. In fact, modular robots can have a **Fixed Movement Number** which helps them to ensure a better rotation movement, making them more suitable for adoption to enhance the orientation and accuracy of positioning. Modular robots can also have a **Repeatable Connection** to ensure an easy yet quick ability between modules to either connect or disconnect on a repeated basis to achieve the required task. They can also be **Tamper Resistant** to withstand any impact or heavy motion especially covering the area that surrounds the configuration's mechanical strength since it is the weakest.

In the case of Swarm Robotics. They include **Homogeneous/Heterogeneous Inter-connected Robots** that are capable of solving problems collectively using swarm algorithms [395], and to react and interact autonomously. Swarms also include three main key parts: **Controlled Swarms**, which are usually several (modular/non-modular) robots that perform tasks and orders based on orders given from a human operator, such as in the case of digital fireworks and light shows using drones. **Semi-Autonomous Swarms**, which are (modular/non-modular) robots that are semi-autonomous, which means they are semi-controlled, being capable of partially performing tasks without any human intervention, but also rely on human interaction in case of any faults or failures. Mostly modular robotic systems that are either traditional or undergoing training to "learn" how to perform the self-healing process. **Autonomous Swarms**, which are usually autonomous (modular/non-modular) robots that are either homogeneous or heterogeneous, and capable of communicating with each other and performing tasks based on their own "analysis", without any human interaction or intervention. Thus, being capable of autonomously performing self-healing and self-reconfiguration tasks.

Next, we present and discuss the importance of Autonomous Operations. Modular self-reconfigurable robots can also act and react independently in case of any incident(s), such as an accident or malicious physical event (disassembly, separation, etc). Modular

robots can include **AI-based Operations** to attach, detach, and/or reconfigure (i.e. resize or reshape) to avoid obstacles or to overcome them. This ensures that they have **Freedom of Movement** without the need to slide on each other. Offering them an extended degree of freedom without the need to be physically interconnected to move (i.e. swarm robots). This also includes the **Degree of Freedom (DoF)** due to their flexible nature and size and self-reconfiguration capabilities [422], modular robots can reconfigure to achieve docking, locomotion, powering, and DoF features [58]. **Self-Assembly** allows a robot to connect either physically or virtually to establish structures to achieve the defined shape. This depends on the **Connection Speed** as it is very important especially during connection and disconnection and vice versa, to achieve reconfiguration tasks, including **Intra-communication**, which allows the connected modules to directly communicate autonomously self-reconfigure, without the need for an external source [375]. Another key feature includes **Repeatable Reversibility** that allows modules to not only connect but also disconnect to maintain the right self-reconfiguration process, and **Flexible Size** which also allows several modules to be grouped together to increase the system configuration's number [416]. **Power Management** is also important as it is achieved by transferring power from low to high-functional modules to increase the modular robotic system's operation time. A **Trade-Off** is required between an accurately aligned modular robot for docking and a tolerant docking system for misalignment. This allows us to achieve **Energy Efficiency** by limiting the module's power consumption required for the docking and undocking process.

After highlighting the main key differences between both robots and modular robots, it is important for us to also present the importance of the self-reconfiguration concept that has become a widely adopted process, especially in modular robots.

To sum up our thesis work, robots are autonomous machines designed to perform tasks, whereas modular robots are a specific type of robot composed of individual modules that can be rearranged to adapt to different tasks or environments. To be more specific, robots are electro-mechanical devices programmed to autonomously execute tasks or under remote control, exhibiting various degrees of complexity and functionality. On the other hand, modular robots represent a specialized subset within the field of robotics, characterized by their modular architecture comprising individual units capable of independent movement and reconfiguration. This modular design offers unmatched flexibility and adaptability, enabling them to dynamically alter their physical structure to suit diverse tasks or environmental conditions. Such capabilities are achieved through sophisticated algorithms governing module coordination and reconfiguration processes, facilitated by onboard sensors, actuators, and computational resources. As a result, modular robots offer unprecedented flexibility and scalability, making them highly conducive to applications requiring rapid deployment, task versatility, and robust performance in dynamic and uncertain environments. We also did a quick comparison between four different modular

robots (i.e. BBs, M-Blocks, KiloBots, and FreeBot) and we presented their differences in Table 2.1.

2.2/ SELF-RECONFIGURATION PROCESS

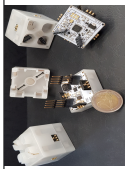



"Self-reconfiguration" means that a given IoMRT device can use its control system to change its overall shape. This term is often accompanied by the "modular" concept where a modular robot is made up of a set of identical modules (often in a lattice structure) that can be attached or detached from the main modular robotic system. Self-reconfiguration means changing the shape made by the global set of robots or changing the total space covered by the ensemble of robots.

In this thesis, we represent and discuss three types of self-reconfiguration: the lattice-based robots, the swarm or the clouds of drones that conduct self-reconfiguration tasks, to cover a different surface on the floor (like Kilobots do [272]), or a volume in the sky (aerial multi-robot systems [184]). The main challenge, in this case, is to localize the robots relative to each other in space and in real-time, to plan all the local movements of all the robots to reach the goal shape while avoiding collisions. A second situation is the self-reconfiguration of modular robots, where the robots are attached to create a dense heap of material (called programmable matter). Reconfiguring such a system is more complex due to the attachment constraints and the relative motion constraints that differ depending on the robot type. For this programmable matter, to realize a self-reconfiguration we consider three motion types: First, the internal motions or tunneling, made possible by meta-modules [325] which locally allows internal exchanges of robots. The second solution is to define motions around the external border of the set [176], and the third one consists in constructing porous internal structures where the robot can move inside the object [442, 347].

In other terms, lattice-based modular robots are structured systems where individual modules form a grid-like pattern, allowing for precise coordination and manipulation, making them suitable for tasks in constrained environments. Swarm robotics involves the coordination of a large number of relatively simple robots, known as "agents" or "swarm-bots," to achieve collective behaviors through decentralized control and local interactions. These swarms typically operate without a central controller, relying instead on local communication and simple rules governing individual behavior, and they also offer scalability and adaptability in unpredictable environments. Cloud of drones operate remotely and communicating with each other and a central control system. They often involve centralized control and predetermined missions assigned to individual drones

Self-Reconfiguration processes are very complex processes due to the number of motions being exponential compared to number of modules. This implies a large paralleliza-

Table 2.1: Comparison table between Blinky Blocks, M-Blocks, KiloBots, and FreeBot.

Figures	Name	Shape	Battery	Movement	Ability	Connection	Communication	Control	Equipment
	Blinky Blocks	Square-like	External	Stationary	Emit Sound and Change Color	Physical	Magnets or Electromagnets	Autonomous	N/A
	M-Blocks	Square-like	Internal	Mobile	Jump, Spin or Hop	Physical	Magnets or Electromagnets	Autonomous	Breaks and Internal Flywheel
	KiloBots	Round-like	Internal	Mobile	Unconventional Locomotion	Wireless	Reflected Infrared	Autonomous	Vibration Motors
	FreeBots	Round-like	Internal	Mobile	Locomotion or Hop	Physical	Magnets or Electromagnets	Autonomous	Rubber Wheels

tion of the motions to get an acceptable time to transform an object to another shape. Several authors presented their work which included computing sets of independent paths for robots to create simultaneous motions [246], or independent structures that allow lines to create simultaneous trains of mobile robots [349].

Moreover, self-reconfiguration is one of the unique characteristics of modular robotic systems in loRT domain, that allow them to change both size and shape. Therefore, it is essential to highlight the main advantages, criteria and characteristics of the self-reconfiguration process such as changing the shape, (i.e the topology of the associated graph).

2.2.1/ CRITERIA

The **self-reconfiguration criteria** of three-dimensional lattice-based modules rely on the number of essential properties. The complexity arises in three-dimensional (3D) lattice-based modules due to the additional spatial dimensions, which require more intricate coordination and control mechanisms compared to two-dimensional (2D) systems. In 3D configurations, modules have more degrees of freedom and possible configurations, leading to increased complexity in achieving self-reconfiguration while maintaining stability and functionality. The **fidelity** or quality of the realized shape by a set of robots is inherently dependent on the resolution of the robot's description of the object, which heavily depends on the number and size of these robots. Nonetheless, reconfiguration algorithms can propose an approximation of the goal shape which depends on the duration of the reconfiguration

The planning is the first step in the auto-reconfiguration process and involves determining the order and parallelism of movements, i.e. choosing which modules can move at the same time without interlocking. It can be done in an initial step (sometimes centralized) [340], or divided into sub-tasks processed during the self-reconfiguration process. **L'homogeneity** is achieved whenever the massive ensembles of mass-producible and modular robotic units are interchangeable in a Programmable Matter. Homogeneity is not necessary but it implies interchangeability that widely reduces the complexity of the self-reconfiguration process. A heterogeneous system involves moving module $\#i$ to a precise position P_i , whereas, in the case of a homogeneous system, any module can be chosen to fill position P_i . Another very important criterion for efficient self-reconfiguration is **parallelization**. Considering that programmable matter is made up of tens of thousands of modules, sequential movement of these modules causes reconfiguration times that are too long to be practically usable. Planning must maximize the number of modules simultaneously in motion. The **reliability** is achieved between parallelism and ease

of convergence of the intended system.

Once all the modules that need to move have an associated goal position, they must then make all the intermediate moves without interfering with each other's movements. This is the **reconfiguration** stage. The process is distributed by nature because the modules move independently and autonomously. However, since they all share the same environment, they must synchronise their actions. The **robustness** of the reconfiguration process can be achieved whenever faults are almost guaranteed to not occur. Fault detection during the self-reconfiguration process is still a very open and complex research area, for when a robot moves, many events can cause faults during the movement, which makes it inefficient. Thus, when taking into account any fault detection during the self-reconfiguration process, will widely increase the complexity of the process. In fact, this is currently hardly treated and mitigated in the literature [34].

In conclusion, the criteria for self-reconfiguration in three-dimensional lattice-based modules are defined by several essential properties that play a crucial role in facilitating the self-reconfiguration process, ensuring the effectiveness and efficiency of the modules' adaptive capabilities. By considering and optimizing these properties, the performance and versatility of lattice-based modular systems can be enhanced, paving the way for their widespread adoption in various applications requiring dynamic shape changes and reconfigurations.

2.2.2/ CHARACTERISTICS

Self-reconfigurable robotic systems have many characteristics [457], where one of them allows them to reconfigure their initial modules' connections (configuration A) into a goal configuration (configuration B) to adapt to a change in its working environment [440, 463]. Such characteristics define them from other reconfigurable systems. In this thesis, we present a list that includes other key characteristics:

Active and Inactive Elements: Can perform their intended tasks in an autonomous independent manner and without any needed human interference or assistance. **Compatible Elements:** Elements tend to be compatible in terms of logical, physical, mechanical and computational manner **Complete Motion:** Modular robots with complete motion as active elements offer unparalleled flexibility and adaptability, enabling them to dynamically reconfigure their shape and behavior to suit diverse tasks and environments. This allows active elements to freely complete motions in a given three-dimensional self-reconfigurable space. **Linking and Connectivity:** Links can connect to cubes and vice versa, where cubes receive the link connectors via an attachment point. In fact, each cube is also connected to one or more link and vice versa. In most cases, linking and connectivity are both done physically to maintain a physical grip between modular self-

reconfigurable robots [287], which can also be done via latching [414]. Moreover, connectivity is dual in distributed robots for programmable matter, since each connection is a physical connection between modules that attaches them. In fact, it can also be a network connection that allows them to exchange messages. **Lattice Form:** Which allows the cube's positions to fit a cubic lattice which guarantees the neighbouring elements' interlocking.

In conclusion, the defining characteristic of self-reconfigurable modular robots lies in their ability to autonomously alter the connections between modules, transitioning from an initial configuration to a goal configuration, thus enabling adaptation to changes in the working environment. This unique capability sets them apart from other reconfigurable systems and underscores their versatility in various applications.

2.2.3/ ADVANTAGES

Due to the uniqueness of the modular self-reconfigurable robots and robotic systems in IoRT and IoMRT domains, this offers a variety of advantages that saw their adoption for real-time and real-world applications in and outside the IoT domain. We enumerate below a list of robots' advantages of a set of self-reconfigurable modular robots versus a unique equivalent traditional robot. These main advantages are presented as follows:

Modularity: Using several identical modules makes it possible to transform a single complex modular system into a set of simple systems with the ability to auto-reconfigure into the intended goal shape to overcome or avoid a given obstacle. The fact that the modules are identical enables transformation by ensuring uniformity and compatibility across the modular robotic system. Identical modules can easily interchange positions and roles without requiring complex adjustments or reconfigurations while achieving both scalability and functionality properties. This uniformity simplifies the coordination and communication between modules, allowing them to seamlessly collaborate to achieve the desired transformation. **Reduced Cost:** The overall cost can be reduced especially for very large systems, where equivalent unique systems are very complex to produce. Modular systems can offer economic advantages through scalability and reduced downtime, but their cost-effectiveness depends on factors such as initial investment, production volume, and customization needs as they must be carefully evaluated on a case-by-case basis. In some cases, modular systems can be more economical than traditional systems since they offer flexibility, easier maintenance, and the ability to upgrade or replace individual components. They also do not necessarily require large numbers to be effective, but they often benefit from scalability, which allows them to expand. **Homogeneity:** This homogeneity would ensure that the overall robot cost is far less reduced through the mass production of complex robotic systems using one of the few adopted uniform modules.

Less Complexity: Designs are made to be far less complex to ensure that they maintain a compatible property that allows them to connect and interconnect with other robots and robotic systems without having their performance or tasks affected. This makes it suitable for both homogeneous and heterogeneous environments. In fact, configuring a small system is much easier than a larger one. However, one should visualize all the functioning contexts of the modular robot's global complexity including the modules' ensemble before trying to make another similar model.

Well-Defined Configuration: This ensures that systems are often well-configured with an easy-to-use and easy-to-deploy method to ensure that their adoption into real-world applications and devices will allow them to maintain higher performance. **Less Human Intervention:** Especially when specific tasks (i.e collision detection and obstacle avoidance) are assigned for modular robots, due to their "intelligence nature", modular robots are capable of achieving the intended goal shape autonomously, and without the need for any human interference. **Enhanced Compatibility:** The adoption of enhanced compatibility allows them to become more suitable for operational use by different types of resource-constrained devices. **Enhanced Trade-Off:** The adoption of versatile modular robots offers the potential chance to adopt a trade-off between their performance and both mechanical and computational complexities.

The auto-reconfiguration of modular robots has many characteristics and advantages that highlight the importance of their adoption. Here, we list them all and discuss them. **Increased Rigidity & Mobility:** Which is added for both fixed and static robotic structures, allowing the adoption of an enhanced force strength that is capable of raising, lowering, pushing, pulling, moving, and throwing other objects. **Self-Reconfiguring:** Offers the advantage of moving robotic systems and devices to be capable of handling unknown situations, which in return would require self-repairing or self-correction (i.e. CEBOT and Polybot) [140]. **Self-Repairing:** Allows machines to successfully replace faulty parts autonomously to correct a given task and to achieve higher accuracy. In the case of modular systems, addressing faults often entails the relatively straightforward task of removing malfunctioning modules and substituting them with functional ones (i.e. ATRON [78]). However, ensuring the safe extraction of defective modules can be a complex process, involving considerations such as system integrity, operational continuity, and safety protocols to mitigate potential risks during maintenance or replacement procedures [132, 260]. **Self-Healing:** These are properties that allow modular robots to repair themselves and to achieve higher flexibility and adaptability which allows them to be deployed in a variety of environments, while also making them suitable for deployment for various scenarios (e.g. self-healing soft pneumatic robots [436] and self-healing polymers [437]). The advantage over self-repairing is that one cannot extract the faulty element, but the intervention is much more complex, requiring sophisticated mechanisms or algorithms capable of iden-

tifying and addressing issues within the system without physical module replacement [43].

Self-Replication: Allows robots to autonomously reproduce themselves without human intervention and at a reduced cost and time. Despite it not being yet achieved, except that it remains a significant challenge due to the complexity involved in creating modules capable of autonomously assembling into a complete and functional replica of the original robot. Such a practical implementation is still in its early stages and has not yet reached the sophistication level seen in natural self-replication processes [289, 2].

New Morphologies and Shapes: The self-reconfiguration ability allows robots and modular robots to assemble and disassemble or vice-versa in an autonomous manner to form new morphologies and to achieve new complex 2D or 3D shapes. Before we conclude this, the unique characteristics of modular self-reconfigurable robots and robotic systems in the domains of loRT and loMRT present a multitude of advantages for real-world applications. These advantages include modularity, reduced cost through mass production, homogeneity, simplicity in design, well-defined configuration, reduced human intervention, enhanced compatibility, and the opportunity for enhanced trade-offs between performance and complexity. By harnessing these benefits, modular robotic systems offer a promising avenue for achieving versatile and efficient solutions across various domains and applications within the IoT ecosystem. Additionally, modular robotic systems offer a multitude of advantages, including increased rigidity and mobility, self-reconfiguring capabilities for handling unknown situations, self-repairing mechanisms for fault correction, self-healing properties enhancing adaptability, and the potential for self-replication, albeit with current challenges. These capabilities pave the way for innovative applications and advancements in various fields, driving the evolution of robotics towards greater autonomy and functionality.

2.2.4/ SELF-RECONFIGURATION TYPES

Self-reconfiguration also has three main types that characterize it from any other reconfiguration processes. Thus, offering more advantages that overcome the limitations that traditional reconfiguration robotic systems suffer from. This is done by relying either on assembly or disassembly processes which are conducted by the robotic systems' components or modules. These types are briefly described as follows:

- **Intra-Reconfigurability:** Includes single-entity systems of modular or non-modular robotic systems that are capable of changing their morphology without the need for any assembly or disassembly processes (i.e. n-Omino-Based Reconfigurable Robot and reconfigurable wall disinfection robot) [213, 388].
- **Inter-Reconfigurability:** Includes modular or non-modular robotic systems that change their morphology via either assembly or disassembly processes.
- **Nested-Reconfigurability:** Includes a hybrid modular or non-modular robotic system

that combines both inter and intra-reconfigurability properties.

Despite the advantages that the self-reconfiguration process offers especially in terms of criteria, characteristics advantages, and types, except that in terms of IoMRT, they are prone to various limitations, failures, and challenges, which we will be listing in detail in the next section.

2.3/ PROBLEMS IOMRT: LIMITATIONS, FAILURES & CHALLENGES

In IoMRT, robotics [479, 484] including self-reconfigurable modular robotics and robotic systems are often being adopted into a variety of Internet of Things (IoT) domains such as Medical and Industrial IoT [488, 489]. As a result of such adoption, they are left prone to a variety of attacks, limitations, and challenges.

2.3.1/ LIMITATIONS & CONSTRAINTS

Aside from the already presented possible attacks, failures, and challenges against IoT, the self-reconfigurable modular robots also suffer from several limitations and constraints, which we present as follows:

The network infrastructure faces unique challenges and limitations that must be addressed to ensure seamless communication and coordination among modular robotic systems. **Deadlocks:** Also prove to be a limitation especially when there's congestion regarding the number of the modules that are waiting to perform their intended task(s). They also refer to situations where multiple modules are unable to proceed since each module is waiting for another to release a resource or perform an action, resulting in a standstill in the robot's operation. These deadlocks can occur due to conflicts in resource allocation, communication errors, or improper coordination among modules, and they pose significant challenges for ensuring the smooth and efficient functioning of modular robotic systems. **Disconnections:** Also prove to be a limitation that causes modules to disconnect along with the modules' failure to establish a link. **Bottlenecks:** still prove to be a limitation, especially for hardware equipment, especially in large-scale applications with self-reconfiguring systems.

Power limitations emerge as a critical factor shaping the design and operation of modular robotic systems, necessitating innovative solutions to optimize energy consumption and extend operational autonomy. **Power Supplies:** Communications require a higher power usage especially when adopting the parallelism mechanism or during its short

movement. This proves to be a severe limitation for resource-constrained devices in modular robotic systems. Therefore, we propose a method to manage this power usage must be developed such as robots with high power covering a longer distance, while robots with short power cover the shortest paths. **Power Backups:** Since actuators require the largest amount of power in a given IoT system aside from the processors, in case of reaching the end of power consumption or a power outage, the whole system will go down. As a result, the absence of backups proves to be a limitation that affects modular robotic systems. **Optimisation:** Supplying modules with power is not an easy task, hence it proves to be a limitation. Supplying a module with a battery is also not easy since both weight and size increase. This makes it difficult to move them around. For this issue, we propose the adoption of capacitors that can offer a power boost to solve this problem.

The domains of motion, sensing, and structural design constitute pivotal facets influencing the capabilities and performance of modular robotic systems within the IoMRT, demanding sophisticated approaches to enhance maneuverability, perception, and adaptability in diverse environments. **Actuator Spacing:** Still proves to be a limitation, especially for spherical modules due to the placement of the actuators which is limited by complexity, weight, and engineering concerns [64]. **Limited Sensing:** Sensors in modular robotics such as swarm robotics have a limited sensing capability that also affects both vision and communication ranges [104]. **Motion Constraints:** Meta-modules for lattice-based modular robotic systems suffer non-holonomic motion constraints which impose motion constraints and add more planning complexity. This was sorted in [107], by grouping modules to act as a "unit". **Lack of Bonding Interfaces:** Which due to the limited precision and strength, along with the field's both mechanical and electrical robustness. **Lack of Power Efficiency:** Due to limited motion precision and energy efficiency especially for modular resource-constrained robotic devices. **Lack of Scalability:** Due to challenges that surround the communications in the IoT domain, low-level controls, and high-level plannings. **Lack of Robustness:** Due to challenges related to failure modes, optimal configuration, and misalignment. **Complex Construction:** Due to challenges surrounding the rapidity and the construction of large architectural modular robotic systems. **Lack of Proper Demonstration:** Due to challenges related to key algorithms, noise, and error issues. **Complex Self-Replication Process:** Due to hardware, software, module, and algorithmic challenges. This highly innovative domain currently only applies to specific situations such as cellular robots. Generalization faces many barriers to overcome. **Lack of Trivial Sensing:** Due to the limited sensing and detection capabilities of ongoing communications on robotic objects. These constraints will surely result in several serious failures, which we will be listing in the next subsection.

2.3.2/ FAILURES

Modular robots and robotic systems in various IoT domains including IoRT and IoMRT, are prone to several failures which may have serious effects on their main tasks including performance and accuracy. Writing a program for a distributed system is much more complex than writing code for a single system. It relies on the exchange of messages between modules, which implies the algorithm's scheduling with appointments between communicating modules to wait for certain messages before continuing the program. For example, if you look at the skeleton of any program in *Blinky Blocks* (cf. Listing 2.1), you'll see that a `BBinit()` function is executed once at startup, but must quickly hand over to the system, and the rest of the code is split between the `BBloop()` function and the event handler (`process_standard_packet(...)`).

Listing 2.1: skeleton of any *Blinky Block* program.

```
#include <BB.h>

// start up code
void BBinit() {
    setColor(getID() == 1?RED:BLUE);
}

// function recalled at infinity
void BBloop() {
}

// Event handler called when a message is received
uint8_t process_standard_packet(L3_packet *p) {
    return 0;
}
```

Interactions between systems may produce thousands of situations that must be treated by the code. Another point is that the complexity of distributed systems lies in the synchronization and collaboration of codes, rather than in the local codes themselves, which are often short and use limited memory. For this reason, the main limitations are presented and highlighted, in our thesis work, with general suggestions to overcome these failures.

Connectivity Failure. Connection between modules, nodes, and robots is prone to interference, not well-established, and not secure, may fail to connect and establish a strong and stable link between modular robots and robotic systems. Multiple network layers (i.e., point-to-point between modules in contact and wireless between specific modules scattered throughout the whole) can be used to strengthen the security of communications while also increasing the risk of cyber-attacks.

Coding Failure. The risk of coding errors is greater in distributed programming for modular robots, as it must be robust to variations in transmission times between modules, which can influence message arrival times and order. Debugging is complicated by the fact that a program may work in most situations, but may also produce errors in the case of particular configurations or faulty robots.

Implementation Failure. Depending on their compatibility and version, modular robotic systems may not operate nor work well over certain simulators, nor can connect to other modular robotic systems or users. Therefore, adopting a uniform standard may be a suitable technique in this case to avoid further implementation failures in IoT. In the case of *Blinky Blocks*, all modules must run the same application code and have the same system version to be able to participate in the same distributed application.

Performance Failure. For technical reasons, distributed systems made up of autonomous modules are even more difficult to monitor than centralized systems. It may result in errors and crashes some of which can be fatal to IoT robotic systems. Hence, the performance must be evaluated on a constant if not a daily basis.

Design Failure. Modular robots and robotic systems if not well-built and designed, may be prone to design failure, depending on their state of the art. As a result, designs must be set well ahead to avoid such problems which may affect all the failures mentioned above. Modular systems must first be evaluated at the level of the individual blocks (CPU, memory, sensor, actuator, etc), then at the level of the whole system, to check that communications are operational, regardless of the number of modules involved.

Software Failure. Software, if not constantly tested updated, batched, and patched, may affect the whole simulation process, as well as the implementation of modular robotic tasks, results, and experiments. Simulation serves as a powerful tool for accelerating innovation and optimizing the design, behavior, and deployment of modular robotic systems, enabling researchers and engineers to explore diverse configurations, control strategies, and environmental conditions without the need for physical prototypes.

Hardware Failure. Hardware equipment can be very fragile if not carefully used and taken care of and may be damaged partially, or beyond recovery. This is also a problem as it may have a huge effect on the modular robotic system's performance. Therefore, robotic materials and components must maintain some sort of tamper-resistant design to avoid being damaged.

Tasking Failure. The adopted algorithms may fail to carry out a given complicated task especially when dealing with larger modules [6]. Therefore, tasks must be well-defined before being assigned to a given modular robotic system or robot.

Personnel Failure. Inexperienced users may be the reason for one or many failures from the ones mentioned above. Hence, users must have some level of experience and training to avoid such unwanted and unnecessary failures and mistakes. For example, the configuration of feeder systems for *Blinky Blocks* sets is not supported by automatic wizards, and only the user's experience can help to place the right number of feeder points in the right position. These failures will surely cause some really hard challenges, which we will be naming and discussing in the next subsection.

2.3.3/ CHALLENGES

Modular robotic systems are also prone to several challenges that affect both IoMRT's performance and connectivity with other modules and links to achieve the required 3D-2D shape. As a result, these main challenges are presented as follows:

In terms of operational, equipment and material, we present the challenges below as follows. **Cost:** The cost can be challenging regarding the necessary equipment that needs to be used including simulators, programs, simulation environment, and testing results. Therefore, this challenge should be mitigated before proceeding further.

Battery Life: Also proves to be a serious challenge, especially for modular robots and IoRT systems that consume a lot of power. This is also a challenge for resource-constrained devices. Passive recharging systems using light energy (with photocells [261]) or electromagnetic systems [397] can significantly increase module autonomy. Implemented algorithms can also help to increase system autonomy, for example by prioritizing the most heavily loaded modules to give the others time to recharge.

Design Complexity: Designing modular robots and robotic systems can be challenging if the aim, goal, and tasks are not pre-defined well ahead before proceeding with its deployment in IoT. Secondly, this design may come up against technological hurdles due to the size of the systems or their self-reconfiguration capabilities, as in the case of the design of the *3D Catoms* [345], which must attach to neighbors and also be able to rotate around other modules.

Compatibility: The compatibility with other modular robotics must also be updated and uniform in order to be able to be linked to cooperate with other modular robotic systems. However, adopting a uniform version is not an easy task and the compatibility requires further testing before being adopted. Hence, it proves to be a challenge.

Functionality: Functionality with other robots is also challenging if tasks are not distributed over robots to avoid congestion and bottleneck issues. Functionality in modular systems is usually grouped under the notion of agent [118]. Each agent must handle one or more functions, and may or may not be activated on each module. It is therefore important that the number of agents activated is sufficient to meet requirements.

Software: Designing software that handles the performance of complex tasks by modular robots is still challenging [383]. Hence, more solutions are required to be studied and evaluated to overcome this issue. **Self-Reconfiguration Operation (SRO):** Proves to be a software-hardware challenge [422] as modular robotic systems are still challenged with the SRP concept to perform their necessarily required tasks. Hence, further work is still required. **Intelligent Behaviour:** Another challenge surrounding the mechanisms' design is the adoption of an "intelligent" system [491], capable of conducting "intelligent behaviors" due to several issues related to electro-mechanical, and material design. **Dimension and Size:** For dimension, mechatronic modules are constrained by the dimensional limitations of their batteries, actuators, and connection mechanisms [141]. As for the size, it proves to be a limitation, especially for micro-scale modules [283] due to technical issues related to micro-scale sensors, actuators, and motors. The size of the modules if reduced can affect their performance capabilities within the IoT domain. Thus, proving to be a challenging limitation. The best solution would be to find how to reduce the module's size without affecting its capabilities [294]. **Computational Power:** Computational power between modules still proves to be a limitation for modular robotic systems, since it severely affects active sensing, which affects the actuator's performance, as well as the whole decision-making process. **Communication Bandwidth & Range:** Aside from being affected by the computational power in terms of communication bandwidth, the communication range is also limited to the local connection due to technical constraints such as energy consumption and bandwidth constraints.

In terms of protection and prevention management, Security still poses a challenge since if not well-secure, modular robotic systems may be prone to cyber, physical, or cyber-physical IoT attacks [67]. **Safety:** Can also impose a challenge since its deployment in real-case and real-life scenarios can impose a safety risk to both humans and machines within the IoT domain [466]. **Privacy:** Is also challenging especially if working within the military, law enforcement intelligence, industrial or any other sensitive domain, since the modular robotic operators along the in-use robotic systems risk having their sensitive or (highly) classified information and details leaked and exposed [242]. **Accuracy:** Can be challenging depending on the false/true positive/negative rate, as well as the margin for failure. Hence, the adoption of a highly accurate modular robot is still challenging and not an easy task [42].

As a result of these IoMRT limitations, failures, and challenges, more security issues will

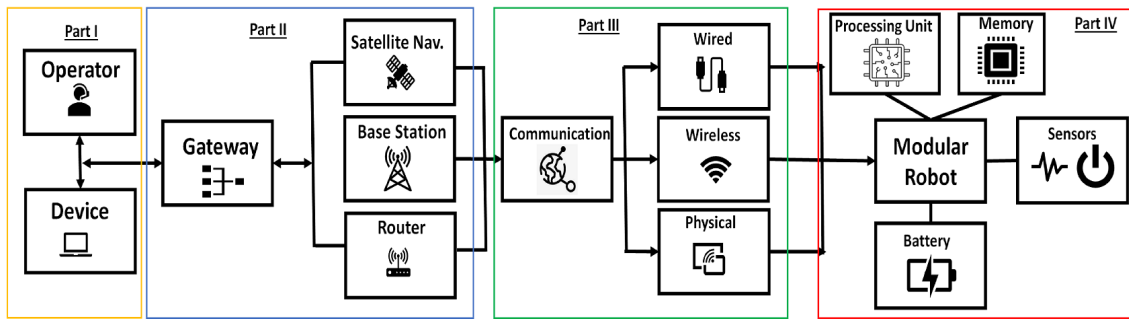


Figure 2.9: IoMRT System Mapping, further detailed in Figure 2.10 and Figure 2.11.

emerge.

2.4/ IOMRT SECURITY

Modular self-reconfigurable robots, like any other robotic system linked to the IoT domain, are also susceptible and prone to the same security issues, threats, vulnerabilities and attacks that target the IoRT. For this reason, this section has been introduced to highlight and discuss the main security issues and concerns that surround the IoMRT, in addition to the main threats and vulnerabilities, along with the main security attacks, which are presented as follows: However, before starting, it is always important to present a system mapping concept that illustrates the whole IoMRT architecture, which shows at every part which component is targeted and how, hence a clarification was illustrated in Figure 2.9.

2.4.1/ SECURITY ISSUES & SAFETY CONCERNS

Despite the great advantages that the Modular Self-Reconfigurable Robotic systems offer especially for IoT, except that the main risks, threats, limitations, attacks and challenges that surround this domain left them prone to a variety of security issues, as well as security and safety concerns. These main issues and concerns are presented as follows. In fact, a Defense-In-Depth (DID) Security Solution was proposed for Attack-In-Depth (AID) cases, mostly that target Modular Robots and their components as seen in Figure 2.10.

2.4.1.1/ SECURITY ISSUES

Modular robotic systems suffer from a variety of IoT-related security issues that affect their performance and accuracy when it comes to performing certain tasks, including IoMRT's security gaps that target modular and non-modular robotic systems and applica-

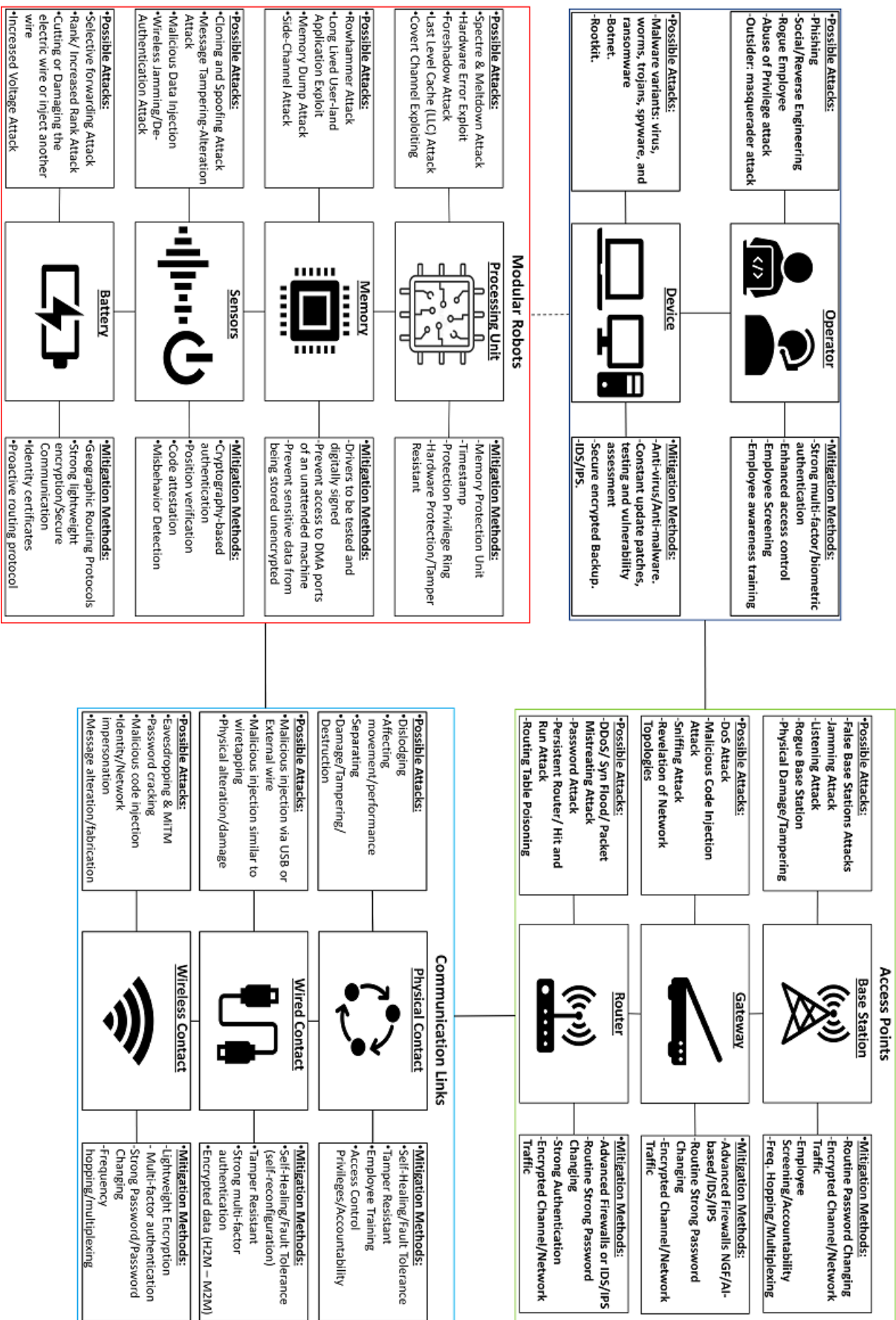


Figure 2.10: A Proposed Defense-in-Depth Security Solution for Attack-in-Depth Case against Modular Robots and their components.

tions alike [70]. As a result, these main issues are presented in this thesis as follows:

Lack of Programming Skills: Can affect both tasks and performance of software, firmware, and application programs that are running using the written code, aside from leaving them prone to further cyber-physical or IoT attacks. Hence, secure and excellent coding skills are required to overcome this ambiguity.

Lack of Tamper-Resistant Equipment: Especially in terms of IoT hardware domain, since any partial or total damage of a given hardware equipment would result in the loss of modular robots functional and operational capabilities [173].

Lack of Self-Healing: Where modular robots will be vulnerable to cascading attacks without the ability to self-heal or recover. Hence, this process is required to maintain both performance and avoid or mitigate any disruption [29].

Lack of Secure Connectivity: Leaves the communication between modular robots and human operators/users insecure and vulnerable to wireless IoT attacks [125]. Hence, the connection must be secure in a way that offers a trade-off between security and performance. These security issues will result in incidents that will lead to the rise of both security and safety concerns.

2.4.1.2/ SECURITY & SAFETY CONCERNS

Due to high IoT security concerns [70] surrounding the modular robotic domain, the impact on a variety of IoT domains from a security perspective has been drastically increasing, causing major security issues. We present these main concerns and highlight them as follows:

On National Safety. The adoption of modular robotic systems in the future will see them being adopted in a wide range of IoT systems including IoRTs and IoMRTs. However, the safety issue that lurks around is based on the design, task performance, and accuracy concept. Therefore, safe practices must be adopted ahead of their deployment so their authorized use will be deemed as a safe practice.

On National Security. The ongoing adoption of modular robotic systems in the IoT domain, might be exploited by both criminals and terrorists alike to conduct their malicious

operations in the physical and not only the cyber field. This security issue must be seriously adopted as it can be used as part of the "domestic terrorism" concept that seriously threatens the lives of local citizens, along with the cause of material damage [45]. Therefore, this concept should not only be adopted but also monitored by both police and law enforcement personnel.

On Military Operations. Modular armed robots are now being extensively deployed for combat purposes by various countries, where these unmanned modular robots are participating in combat tasks (i.e land, sea, surface/underwater, air and space (Satellite Navigation-SatNav)) [20], especially in the Middle Eastern [361] and African regions, as well as in conflicts in Eastern Europe [319]. This also includes the Explosive Ordnance Disposal (EOD), bomb diffusion, and demining operations [130] in many cases including but not limited to Northern Ireland, the Middle East (i.e Syria, Iraq), Africa (Mali, South Sudan, Somalia, Congo, etc) [3], Asia (i.e Afghanistan and Pakistan), North and South America [82, 394]. For this reason, it is important to adopt effective counter-measures to overcome this growing threat which has proven to be lethal against military targets including installations, material and personnel. For this reason, various robotics were deployed including smart sensors and detection systems to detect and prevent cross-border operations including but not limited to armed infiltration, tunnels, rocket firing, paramilitary operations, ambushes, etc. In fact, AI-based swarm robots saw deployment within the ranks of the British (i.e RAF base in Cumbria) [44] and US (i.e Utah desert exercises EDGE 22) [391] armies mainly as part of the "Interactive Drone Swarm" and as part of the Human Machine Teaming (HMT) project based on advanced command, control, communications, computers, information/intelligence, surveillance, and reconnaissance (A-C4ISR) and AI-supported intelligence, surveillance, and reconnaissance (AI-ISR) technologies. The same concept of swarm is said to be applied by France [120]. In fact, during the last Gaza conflict (May 2021), Israel launched the "first AI war" where swarm robotics (mainly drones) were deployed during a real combat scenario [214, 243], and the introduction of its most advance loitering munition "LANIUS" drone (as part of Elbit Systems / Legion-X). [324] Similar scenarios will soon be applied by both Russia and Ukraine against one another during the ongoing war (since 2014 [17], especially as the AI-based drone warfare is evolving between both sides in southern, northeastern and eastern parts of Ukraine (following the invasion since February 2022) [233, 214], with more kamikaze drones being used by Ukraine (RAM-II [241], PD-1, PD-2, Polish made WARMATE [47], US-made Switchblade-300/600 [264] and Aevex Phoenix Ghost loitering munition [427], as well as modified commercial UAVs including Parrot, DJI, ST-35 Silent Thunder and MatrixUAV Comandor [14, 229]), and Russia (mostly KUB-BLA and its enhanced ZALA Lancet variants, as well as Iranian-made Shahed-136 and Arash-2) [124, 378].

On Counter-Terrorism Operations. Modular armed robots were also widely used in counter-terrorism (i.e radicalism/extremism) and counter-insurgency operations mostly in the Middle East (i.e. Lebanon, Syria, Gaza strip, and Iraq), Arab Gulf (i.e. Yemen) and North African regions (i.e Somalia, Libya, Yemen). However, they were also adopted by various terror groups using explosive-laden robots including boats and drones to target military and civilian installations [407], as well as attempts to wirelessly intercept or hijack them [179, 237]. Meanwhile, different counter-measures are being adopted and developed to mitigate this new growing risk and threat. Similar techniques were also adopted by criminals and narco-gangs in Latin America. Therefore, this threat is also growing and requires having it addressed as soon as possible [273]. This may result in the introduction of the Robot-based IED, or the RBIED concept but in a swarm-like formation similar to loitering munition and [214] Boat-Borne IED (BBIED) [68].

On Industrial Operations. Rival industrial companies especially in terms of competition can be prone to a variety of IoT-related security and safety issues especially as both sides is leading innovative plans and tasks using robots and designing them such as stealing business trades and exposing business secrets. However, functional, technical and operational problems threaten the safety and privacy of the working personnel due to faulty operations either caused by a cyber/physical attack (i.e industrial espionage or sabotage) or due to modular system's failures. A safer practice must be adopted in a secure way to protect working personnel [70, 61].

On Law Enforcement Operations. Robots including ground robots and drones are being constantly used for a variety of domestic policing tasks including border patrol and control [308, 156], crowd control, rare species protection, border trespassing, protests and cross-border protests monitoring and control [401, 94], as well as to tackle domestic crimes and terrorism including with domestic EODs. However, terrorists are also known to carry out cross-border attacks using robots including UUVs, UAVs and Ground robots to conduct cross-border attacks (i.e bomblet dropping [83]), Lightweight Multi-role Missile (LMM) "Martlet" launching (RAF Jackal drone), surveillance or reconnaissance missions or to smuggle weapons [419]. Also, criminal gangs are also adopting this technique to perform the already mentioned attacks, in addition to smuggling drugs (i.e narco-drones [134] or drug-smuggling drone submarines [470]), and white weapons (i.e knives, blades, etc) into prisons [177, 59].

On Medical Operations. Robots including drones and ground robots [30] were extensively deployed to combat the ongoing COVID-19 pandemic including its variants via interaction with infected patients, raising awareness, cleaning/sanitizing infected areas, and

monitoring patients [488, 484, 212]. However, robots were also prone to a campaign of IoT-related cyber-attacks mainly ransomware attacks in order to extract money, especially in bitcoin. Thus, risking secret medical documents and files being leaked and exposing the privacy of thousands of patients. After highlighting and discussing both IoMRT main security and safety concerns, our work will focus on the main security threats and vulnerabilities.

2.4.2/ SECURITY THREATS & VULNERABILITIES

In this section, the aim is to highlight the main issues and vulnerabilities that surround and target the modular robotic systems specifically, and the robotic domain generally especially when being a key part of the IoT concept. In IoT, threats that surround the modular robotic domain are not limited to one aspect, but rather to a list that lurks around it and can seriously affect it and damage its performance, accuracy, tasks and connectivity. Below, we attempt to provide an inventory of these threats.

2.4.2.1/ THREAT SOURCE

In IoT, robotics threats are now growing, and they are not only limited to industrial rivals and competitors. It outgrew to include crimes, warfare (i.e. spying, sabotage, or espionage), and terrorism. Threats can come from different sources [24] and can be part of cyber-crimes, cyber-warfare, cyber-espionage, or even cyber-terrorism. This thesis lists the main ones as follows:

Whistle-blowers: Or insiders are bribed, or dissatisfied employees. This threat is classed as one of the most dangerous threats that surround the modular robotic domain due to their ability to steal or leak highly sensitive information regarding the modular robotic system either for spying, reconnaissance, or cyber-attacks.

Outsiders: Can either be hackers or attackers that aim to gain remote access to any IoT device or system connected to the modular robots to steal information, inject malicious payload, bring the robotic system down or gain access.

Rival Businesses: Can also be the main cause of any threat especially when both rival sides are very competitive. The aim is usually to leak or steal sensitive information and documents that can damage the other rival company's reputation by exposing customers and business deals [196].

Third Party Companies: Especially When these companies are not verified nor trusted, or they tend to have bad reputations or perform suspicious acts via their applications, software, and programs. Hence, it is essential to verify the legitimacy and reputation of each company.

Rival Governments: Are also competing in order to take on the lead on covering modular robots and modular robotic systems especially in terms of self-healing and self-reconfiguration in all IoRT domains, Hence the use of state-sponsored hackers, hacktivists, spies or a cyber-army divisions to perform defensive or offensive tasks such as stealing model designs, algorithms, databases, spying, etc, all as part of cyber-warfare, cyber-espionage, or industrial espionage/sabotage [126, 465, 384].

Extremists/Terrorists: Or radical insurgents have started to understand the concept of IoT and started to using and developing their own robotic designs (i.e Unmanned Aerial Vehicles (UAVs), Unmanned Ground Vehicles (UGVs), Unmanned Surface Vehicles (USVs) and Unmanned Underwater Vehicles (UUVs)) in their terror attacks as described in [479, 484] (i.e. case of ISIS in Mosul [333]). Modular (self-reconfigurable) robots are no different. Therefore, it is essential to prevent such technology from being further exploited by terror groups and organizations alike. That, is not to mention the threat coming from cyber-terrorism along with their cyber-attacks also targeting this domain. It is also important to identify the threat. Hence the threat type is discussed next.

2.4.2.2/ THREAT TYPE

IoT threats can be divided into a variety of types which affect the modular robotic systems differently. For this reason, they were classed and divided as follows:

Cyber-Physical Threats are threats that can be divided into two main types, which we present as follows: **Cyber Threat:** Due to the open wireless connection and communication between modular robots and their robotic systems including devices, electronics, sensors, actuators etc, all of this equipment are prone to cyber-attacks since the connection is open, non or ill-secure even if locally, would leave it prone to an imminent attack. **Physical Threat:** Takes place when an intruder or a rogue employee targets, destroys, or physically damages the robotic equipment or even steals it.

An Act of God or the loss arising from inevitable accident can also occur due to **Natural Threats** caused by a natural event (i.e. flood, earthquake, storm, lightning, etc) especially

since modular robots will be deployed to complex and challenging environments such as the ones mentioned above.

Duty-Related Incidents can also be reported due to many reasons. Here, we list the main reasons and direct causes that may be the reason behind. **Accidents:** Accidents can take many aspects which can be cascading, fatal or non-fatal accidents which would threaten the whole robotic system to collapse or not perform its intended causing harm or/and material damage. **Misconfiguration:** This type of threats can affect how a given algorithm or program is executed due to coding bug, or bad coding skills which would also affect the performance, accuracy, and tasks of modular robots and robotic systems alike. **Technical Threat:** Is related to technical communication and connectivity issues which may be prone to potential IoT attacks, or technical difficulties and limitations in terms of the distance covered, frequency, and the noise that may affect them. **Operational Threat:** Is based on a lack of monitoring of the ongoing performance of a given device linked or part of the modular robotic system or robot. This can cause a device to deviate, which would leave it prone to errors and affect its accuracy. **Power Threats:** Threats from power or electricity occur when the incoming voltage current is not regulated causing both IoT and modular robotic devices to be destroyed or damaged. Another case would be the cause of power cuts or blackouts without backup power supplies would bring both connections and devices down [480]. The type of threat is important to be known. However, the nature of this threat is equally as important. Hence, it is discussed next.

2.4.2.3/ THREAT NATURE

The nature of any threat especially those targeting the IoT domain can be divided into two main types which we describe them as follows:

- **Human:** Include accidents are done by human users or operators, as well as rogue, dissatisfied employees. Also, insiders and outsiders (i.e. hackers or attackers) can also be included and are further described below.
- **Non-Human:** Which can be natural or non-natural. Natural as mentioned above due to weather or other environmental conditions. Non-natural can be due to technical, operational or any other factors which will be further described below.

After identifying these threats, the work will focus next on the main security vulnerabilities.

2.4.2.4/ SECURITY VULNERABILITIES

Modular robots are also prone to a variety of IoT-related security vulnerabilities that can affect their whole processing along with their operational and functional task requirements;

which in turn would affect their accuracy, performance, and productivity [435, 417]. For this reason, we present these main vulnerabilities as follows:

Network vulnerabilities. The network vulnerabilities of modular self-reconfigurable robotic systems encompass susceptibility to unauthorized access, data interception, and manipulation, potentially compromising system integrity and functionality. More specifically, **Network Vulnerability:** Modular robotic systems are vulnerable to a variety of IoT-related communication and connections attacks which can either be passive (i.e eavesdropping) or active (i.e man-in-the-middle attack), along with other network attacks such as replay, sniffing, spoofing, etc.

Software, simulation, and application vulnerabilities. The vulnerabilities in software, simulation, and applications of modular self-reconfigurable robotic systems pose risks such as software bugs, inaccurate simulations, and compromised application functionality, undermining system performance and security. Here, we divide them and present them separately. **Software Vulnerability:** May result due to the lack of periodic updates of new design and security patches, leaving modular robotic systems in IoT vulnerable especially when using third-party software. **Simulator Vulnerability:** Simulators if not well tested also in terms of safe-to-use and secure-for-use concepts will make it easier for an attacker to simulate their attack ahead of conducting them on real-case scenarios. **Application Vulnerability:** If not tested or used from a trusted source can affect the performance as well as result in a possible privacy breach to the whole modular robotic system, especially in the IoRT and IoMRT domains.

Safety and security vulnerabilities. The safety and security vulnerabilities of modular self-reconfigurable robotic systems encompass risks such as physical hazards, unauthorized access to sensitive data, and potential manipulation of system behavior, posing threats to both human operators and the integrity of the system itself. Here we discuss each of them separately. **Safety Vulnerability:** Safety tasks and design if not well-established can result in ill-performance and bad accuracy with a high error rate. Therefore, the safety concept must also be adopted to prevent such vulnerabilities from occurring. **Security Vulnerability:** If modular robots and robotic systems are not secure with the adoption of the right security measures, their services might either be interrupted, intercepted, or halted temporarily or permanently. Therefore, security measures must also be considered especially in the built-in IoT design. After classifying and identifying these security vulnerabilities, it is important for us to identify and classify which security attacks, events, or incidents might take place. Hence, the IoMRT security attacks are presented next.

2.4.3/ SECURITY ATTACKS

In IoT, modular robotics like their peers in the robotic domain are prone to a variety of security attacks that target them of which they exploit similar security gaps, weaknesses and vulnerabilities in order to achieve their desired malicious goal [281, 297].

Attacks that target the modular robotic and the IoMRT domains are not limited to one but many aspects, especially since the integration of modular robotic domain will see it being adopted and applied in all IoT fields including but not limited to Industrial IoT, Medical IoT (also against the ongoing COVID-19 pandemic) [295, 499] and Battlefield IoT [148]. This leaves them prone to a variety of security attacks which can have serious implications on them. As a result, these main attacks along with the targeted components and their field are listed and presented below as follows including attack source, types, and classifications.

Hence, it is important to identify the attack source next.

2.4.3.1/ ATTACK SOURCE

The attack cause is not limited to one source. Instead, the modular robotic domain due to its adoption and integration in the IoT domain, has become the main attention of attackers from different sources mainly including:

Politics. Hacking modular robots could yield significant political gains by enabling adversaries to disrupt critical infrastructure, compromise national security, or gain strategic advantage through espionage or sabotage. Here we divide deeper and further explain them. **Hactivism:** May be another form of attack source, as a way to protest against any government modular robotic design such as protest against the use of autonomous vehicles and their combat lethal capabilities. **States-Sponsored:** The source can be the cause of a state-sponsored hacking such as the ongoing cyber-wars between countries, or state-sponsored terrorism as a way to destabilise other countries such as the stealing of valuable modular robotic information including secret designs. **Cyber-Terrorism:** This is another ongoing attack source where terrorists are gaining more sophisticated robotic capabilities which they were extensively used in the Middle East and African regions. This is not limited to cyber-attacks such as intercepting robotic information such as footage, or partially controlling systems, but rather manufacturing their own modular and non-modular robotic designs including UAVs, UUVs, and UGVs for a variety of tasks including armed, explosive-laden, and loitering robots. This is much more details in [479, 484], respectively. **Cyber-Warfare:** may well be the new attack source especially when deploying modular robots and robotic systems for combat tasks and operations, along with the reliance on cyber-attacks to target these robots and robotic systems. Modular robotics

may also be prone to (counter) espionage, sabotage, and surveillance operations.

Competition and Rivalry. For a rival company, hacking modular robots could provide valuable insights into proprietary technology, intellectual property, or strategic business operations, allowing them to gain a competitive edge in the market. **Company-Sponsored:** Is usually the cause of competition between rival companies when competing against each other within the modular robotic domain. This most notably includes industrial espionage, sabotage, or surveillance tasks and attacks.

Personal Gains. For personal gains, hacking modular robots could involve stealing sensitive information, accessing financial data, or even causing physical harm, all of which could be leveraged for blackmail, extortion, or other malicious purposes. We summarised this as part of **Cyber-Crimes:** Are the cause of cyber-criminal activities including hackers or attackers who aim to exploit the modular robotic domain mostly for personal gains. Hardware equipment damage or stealing is another part of criminal activities.

Since the attack source is identified, it is important to identify the attack type. This is presented in the next sub-subsection.

2.4.3.2/ ATTACK TYPES

Attacks against IoT including IoRT and IoMRT in general, and modular robotic systems in specific, can take many forms types, and shapes and can either target either the modular robot or its human operator [400, 10]. Therefore, it is essential to be familiar with the security concept [215] of which the modular robotic domain lacks, by identifying these main attacks in order to be later on capable of adopting the right security measures to either mitigate or prevent them.

In this thesis, IoMRT attacks were divided into four: network-based, gateway-based, operator-based and modular robot-based, where a set of possible security solutions is generally presented for every single attack classification which we present as follows:

Network-based Attacks. A variety of IoMRT network-related attacks is listed and presented below. However, it is always important to maintain continuous patching, perform penetration testing, and investigate previous attacks using forensics toolkits. Further solutions could also be adopted and added to secure the IoMRT's network communications such as Intrusion Detection Systems (IDS), bidirectional link checks, Geographic Routing Protocols (GRP), strong lightweight encryption, lightweight message authentication, encrypted channels, multi-factor authentication, strong and constant password changing policy, and proactive routing protocols [227]. **Distributed Denial of Service Attack:**

Where hundreds or thousands of computerized modular robots can exchange packets simultaneously to target the main gateway, or to infect the connected devices and turn them into zombies (botnets). **SYN Flood Attack:** Where a rogue modular robot sends a large number of TCP/SYN packets using a forged address to end all connections between modular robots and base stations. **Brute Force Attack:** Especially where an attacker targets IoMRT gateways especially routers to guess the password and gain access. This depends on the password's strength and the attacker's skills. **DMA Attack:** This is a side-channel attack type that targets an IoMRT device by exploiting high-speed expansion ports (mostly unused) to gain unauthorized direct memory access (DMA). **Hello Flooding Attack:** Targets edge nodes in IoMRT networks, and usually occur when a network node sends a Hello packet in high power, overpowering the parent node and having the other network node mistaking it as the parent node [121]. **Rank Attack:** Affects the network performance as a result of increased overhead control, a low delivery of packet ratio, and a high end-to-end delay [1]. **Increased Rank Attack:** Occurs when a malicious node chooses to increase its rank through the falsification of its DAG Information Object (DIO) messages to disrupt the routing topology [218]. **Selective Forwarding Attack:** Occurs when a malicious node discards network packets in a selective manner by intercepting sensitive data to manipulate it or prevent it from being sent, while forwarding non-critical data. **Wireless Attack:** Disrupts and interrupts the availability of IoT's robot-to-robot and robot-to-humans wireless communication either permanently via jamming or temporarily via de-authentication. **Channel Monitoring Attack:** Monitors the incoming and outgoing exchanged data messages between IoT's robot-to-robot and robot-to-humans either passively through eavesdropping, or actively through man-in-the-middle attacks to intercept, modify or falsely inject malicious data. In order to overcome this issue, channels must be made secure, and the communication must be encrypted in a real-time manner. **Password Attack:** Password cracking attacks target the authenticity of modular robotic systems to break the connection and communication parts to gain unauthorized access to the system and to monitor the ongoing exchange of data for further manipulation, modification, or denial. **Malicious Code Injection (MCI):** Or Remote Code Execution (RCE) attacks execute malicious codes, exploit coding vulnerabilities or inject malicious coding scripts covertly to gain control, or/and target both functional and operational tasks. Professional coders must be assigned to the task, especially those with a security background to prevent such attacks from taking place. **Illusion Attack:** Occurs when one or many compromised modular robots are connected to the IoMRT network to generate false or malicious data that will be spread across the whole network. **Obfuscation Attack:** Obfuscates the intended meaning of communication by making the message difficult to read and understand by often using ambiguous language. **Cloning And Spoofing Attack:** Occur when Cloning duplicates the spoofed data, whilst spoofing clone the intercepted data from IoMRT networks to gain unauthorized access to the

modular robotic system. **Wireless Jamming Attack:** Severely interrupts and disrupts any established wireless communication between modular robots in swarm formation and modular robots and their operators. This can be done in a selective or non-selective manner. It can also be temporary, periodic, and constant. In fact, De-authentication is one of its main attack types. **Delay Attack:** Causes serious delays for timely and high-priority message transmissions across IoMRT channels, possibly causing communication bottlenecks and deadlock. **Wormhole-like Attack:** Can target the IoMRT network by deploying one or many malicious modular robots mostly undetected, eavesdropping and recording wireless information.

Gateway-based Attacks: Also occur and can cause serious problems that would expose the whole IoMRT network and systems alike. Therefore, it is always important to maintain continuous patching, perform penetration testing, investigate previous attacks using forensics toolkits, set IDS/IPS, Firewalls, packet filtering, position verification, lightweight real-time encryption, identity-based security, anonymized networks, and communications, especially for P2P networks [207], packet misbehavior detection, header, and payload encryption (i.e. Transport Layer Security (TLS)) [477]. **Persistent Gateway Attack:** Applies the hit-and-run concept by constantly injecting frequent harmful data packages into the IoMRT gateway and network in order to gain unauthorized control. **Packet Mistreating Attack:** Injects IoMRT network packets with malicious codes that disrupt and confuse the gateway that treats them, bypassing its security and infecting the network. **Routing Table Poisoning Attack:** Manipulates the gateway's routing table which reveals all the transferred and received information. This is often done by editing the information packets and often targets both IoMRT networks and servers. **Hit-and-Run Attack:** also known as test hacks, where malicious data is injected into the gateway via coding. It is a very basic attack method but still remains in use. **Sniffing Attack:** Targets the Border Gateway Protocol (BGP) to detour traffic through a malicious network by retrieving the BGP's routing information to unmask the IoMRT's network topologies. **False Base Stations Attack:** Performs active/passive attacks that target connected modular robots to steal valuable information, geolocate operators, and jam their signal [217] either to end connection or urge operators to operate over higher less secure less prone to interference frequencies to listen and intercept incoming and outgoing conversations. **Version Number Attack:** Is initiated by the global repair mechanism which increases the network traffic control to affect its availability and performance.

Operator-based Attacks: Sometimes it is much easier to attack the operator in order to exploit any of the IoMRT components, especially when the system, server network, and devices are highly secure. Therefore, it is important for users to receive awareness training, especially against phishing, social engineering, reverse social engineering

and accountability must well be enforced. Attacks can also be based on insiders and whistle-blowers, therefore, it is also important to adopt an employee screening policy, privileges, and access control must be monitored, and employees must also be encouraged to work in a friendly and stress-free environment. Ethical hacking can play a key role, especially during a penetration testing simulated attack to see the readiness and reaction of operators during such an event. **Insider Attack:** Often happens when disgruntled employees with prior knowledge of the IoMRT network topology, gateway login, and password information can have unauthorized access and target the IoMRT domain. Gateways should have strong configurations and constantly updatable software, and access controls should be always supervised [142]. **Social Engineering Attack:** Aim to exploit either operators or users working within the modular robotic domain in order to retrieve or extract information mainly by exploiting human emotions or instincts [238, 381]. A proper way to overcome this issue is via employee training, screening, and maintaining accountability. **Reverse Social Engineering Attack:** Includes a person-to-person where a given attacker has direct contact with a modular robotic operator and compels them to divulge sensitive information either through soft means (seduction or personal fulfillment) or hard means (psychological damage or blackmail), or violent means (physical beating or torture). **Phishing Attack:** Target the employees working within the modular robotic domain by sending suspicious emails with malicious files or documents attached to them to gain unauthorized but privileged access to install a backdoor, or even add a malicious program running in the background to retrieve malicious data. This issue can be mitigated via employee training, screening, and maintaining accountability [400, 10]. **Abuse of Privilege Attack:** Takes place when a rogue employee exploits its access control privilege to perform malicious tasks such as leaking secrets, exposing systems, spreading malware infection, or installing spyware. **Stolen ID Attack:** Occurs when a given legitimate operator card is stolen from them without their notice, to gain access into a given organization to perform an insider task. **Fake ID Attack:** Occurs especially when security and safety measures are ill-applied in a given IoMRT-linked department, which grants any user access without Identification and verification of the authenticity of the user. **Forged ID Attack:** Is more advanced than fake ID attacks, especially when a forged ID matches the legitimate one and goes undetected by the Identification and verification process.

Modular Robot-based/Swarms Attacks: These attacks often target one or several modular robotic components including devices, applications, software, hardware, operating systems, batteries, etc which can seriously affect the IoMRT domain. There, it is always important to maintain continuous patching, perform penetration testing, and investigate previous attacks using forensics toolkits. Moreover, for physical security against potentially malicious devices, Input–Output Memory Management Unit (IOMMU) can be adopted against memory attacks, unwanted ports should be closed, data encryption can

be maintained, and sensitive data should not be stored unencrypted in Random Access Memory (RAM). Moreover, misbehavior detection, anti-virus/anti-malware, constant testing, and vulnerability assessment/checking, secure encrypted Backup, code attestation, constant update patches, signature-based detection, and battery voltage monitoring (if needed). Also, forensics and ethical hacking can play a part. **Environmental Attack:** Which is often achieved by exploiting stigmergy as a communication mean, or by introducing a physical barrier that prevents a ground/sea/air moving swarm from heading towards their desired direction [357]. **Pre-programmed Rogue Swarms:** Especially those that are based on Machine Learning (ML-based) to break into IoMRT devices and networks by conducting AI fuzzing to detect Zero-Day exploits [174] to expand the attack across the network, where in most cases they evade detection [112]. The main aim is either for reconnaissance to retrieve confidential information, or to hijack the pre-programmed swarms to lead to more harmful attacks including DDoS. **Re-introduction Attack:** Which can take many forms, such as removing and reintroducing individual swarm elements or conducting supply chain attacks [390]. **Misappropriation Attack:** Usually occurs when swarms are led and diverted to perform a task other than their intended objective [393]. **Spectre Exploitable Attack:** Aims to break down the isolation between the different IoMRT applications, allowing the tricking of error-free programs to leak sensitive information and exploit any update patches. **Meltdown Exploitable Attack:** Breaks down the fundamental isolation between the IoMRT's user applications and the operating system to gain access to the memory and retrieve sensitive information about both programs and operating systems. **CPU Attack:** Which targets the central processing unit (CPU) available resources to degrade its performance and causes it to loop, overload, livelock, or deadlock. This can also be achieved by exploiting a hardware error in CPU design requiring a hard reset to fix it [447]. **ZombieLoad Attack:** Steals sensitive data and keys whenever they are accessed by an IoMRT device, before running a malicious program that exploits internal CPU buffers to retrieve confidential and sensitive information, mainly passwords, user keys, and disk encryption keys. **Foreshadow Attack:** Is a speculative execution attack that steals sensitive information stored inside personal IoMRT devices or third-party clouds and extracts data from Intel processors, Virtual Machines (VMs), hypervisors (VMM), operating system (OS) kernel and System Management Mode (SMM) memories. **LLC Attack:** Occurs when a Last Level Cache (LLC) is exploited, having its shared resources across different cores on the same processors being intercepted. Thus, recovering sensitive information such as personal information, keys, stored data, etc. A similar approach was presented in [206], to recover information leakage caused by the data access time difference. **Physical Electrical Damage:** Which can be caused by cutting and damaging the electric wire or injecting another wire similarly to the wiretapping technique, to increase the voltage which damages and destroys the IoMRT systems (i.e Blinky Blocks). **Power Blackout Attack:** By targeting the main electricity/power source

to partially damage the IoMRT system which causes a total blackout or partial blackout for the system as a whole. **Boot-sector Virus:** Usually spread via IoT storage devices mainly Universal Serial Bus (USB) devices. The boot-sector aims to replace the original boot-sector by implementing its own malicious version to modify any modular robotic file or program. Anti-viruses and IDS systems can be useful to mitigate this attack. **Spyware Attack:** Collects small information pieces about the modular robotic system without the operator's knowledge, which means in a covert manner. Spyware attacks can monitor all the performed tasks on the system. It can also modify the system controls and can even modify, remove, or install new software and applications. **Snarf Attack:** Aims to copy masqueraded data or files over a given IoT network with the aim to gain a backdoor access to the system to gain higher access and permission privileges. **Ransomware Attack:** Encrypts all the data linked to IoT's modular robotic domains, while also locking backed up data to prevent both users and operators from recovering data. **Trojan Attack:** These are masquerade applications or files that look legitimate to bypass any security measures that are protecting modular robotic systems from gaining unauthorized access. Random Access Trojan (RAT) attacks are another type of attack that aim to gain a higher access privilege to modify a modular robotic system's performance, accuracy, functionality and operations. Anti-viruses with anti-Trojan specialties along with the adoption of IDS/IPS systems should be helpful to secure against Trojan attacks. **Worm Attack:** Are self-replication programs to modify, delete, and negatively impact the IoT's network traffic to delete modular robotic files or documents or install a backdoor, while self-replicating on other modular robotic systems. To overcome this attack, it is important to use anti-viruses and IDS/IPS systems. **Buffer Overflow Attack:** Exploits the modular robotic Operating Systems to hijack the control of modular robots. The attack targets the modular robot's memory causing an overflow. Other type of buffer overflow include the pointer overflow attack which is caused by poor buffer management [454]. To overcome this attack type the buffer size must be limited, while adopting strict coding standards. **Identity Attack:** Takes place when the modular robot's identity is retrieved by a given attacker, which can further expose additional information about the modular robot, connected system, network, and its operator. **Rogue Bot Attack:** Takes place when one or more modular robots are all exploited to perform malicious tasks, especially life-threatening tasks such as carrying criminal or terrorist operations, or even hijacked by opposing forces to be used under the pretext of "turning their weapons against them". In this case, most weaknesses are exploited in the modular robots or security gaps within the IoMRT network/gateway are exploited. **Reverse Engineering Attack:** Aims to find software/OS flaws within the modular robotic system, before reverse-engineering the code to scan and find more IoMRT-linked vulnerabilities. **Physical Attack:** Includes tampering with the modular robot or even damaging it or physically destroying it and destroying its sensitive components, which would result in partial damage, damage beyond recovery, or permanent loss. In fact, this thesis

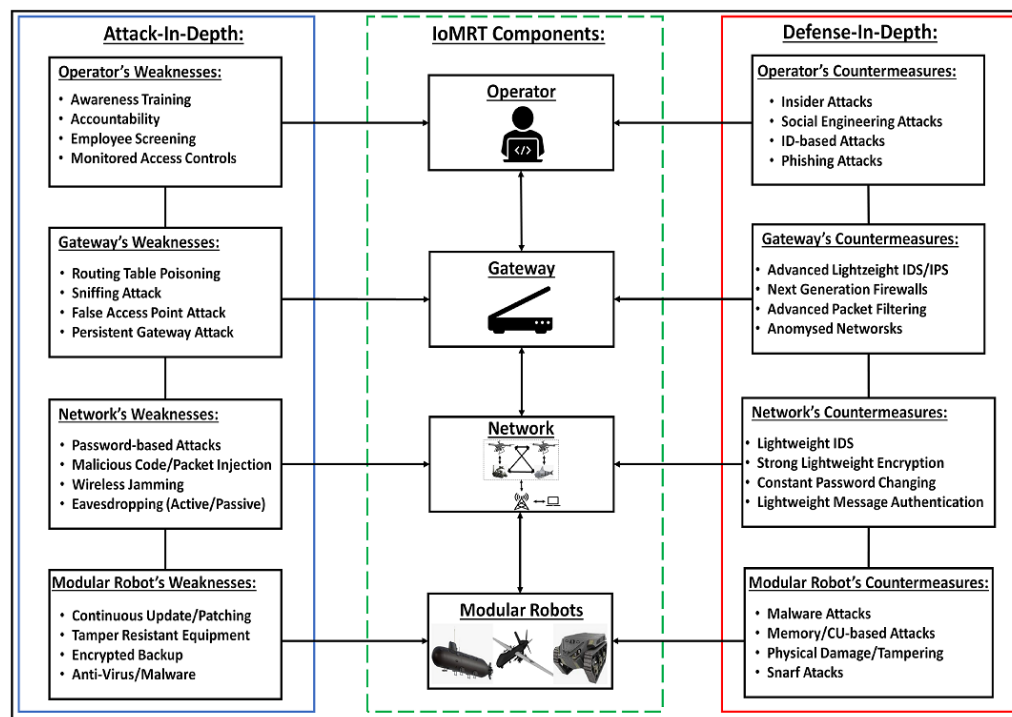


Figure 2.11: A proposed framework that classifies IoMRT's main components, along with their key weaknesses and countermeasures.

proposes the following framework (see Figure 2.11) that identifies the main IoMRT components highlights weaknesses, and suggests suitable security/safety solutions.

Now, it is important to classify these attacks.

2.4.3.3/ ATTACKS CLASSIFICATION

In IoT, modular robots and robotic systems are also prone to a variety of attacks that may have serious implications and effects on their performance, accuracy, tasks, and connectivity. Most of these robotic attacks are summarised in [479, 484]. These attacks which can be divided into cyber and/or physical attacks may be combined to form a cyber-physical attack that may have a much more devastating and cascading effect on both robotic systems and platforms.

As for Cyber Attacks. Modular robotic systems are wireless, and not wired in order to cover a wider range. Due to their wireless nature while communicating with each other, modular robots are prone to several cyber attack types which are also common in the IoT domain (i.e. IoRT and IoMRT). Here we list them as follows: **Malicious Injection:** Modular robotic systems may be prone to injection attacks such as file injection (i.e. viruses), payload injection (malicious third-party program) and code injection attacks (backdoor, buffer overflow, endless loops) which may affect its ability to perform tasks. **Interruption:**

The modular robotic system connection may be prone to interruption attacks either temporarily via de-authentication or permanently via jamming. **Interception:** The connection may also be prone to interception either passively (i.e. eavesdropping) or actively (i.e. man-in-the-middle attack). **Modification:** Modular robotic system updates or patches may be modified by a malicious third party which can either be an insider (rogue or unsatisfied employee) or an outsider (hackers and attackers). **Coding:** Badly written codes or/and algorithms can be prone to a variety of IoT attacks such as modification, or infinite loop attacks where codes keeps on being executed. Backdoors can also occur, where an attacker can gain access to the full coding/algorithm format.

As for Physical Attacks. Modular robots and robotics systems are also prone to physical attacks just like in IoT, making them prone to physical damage, alteration and/or destruction. Here we list them as follows: **Tampering:** Modular robots/robotic systems are also prone to physical damage and damage beyond recovery mostly by insiders. **Physical Damage:** Though this can happen by accident, it is still classed as an attack, where a user accidentally damages the modular robot or robotic system. **Theft:** Theft attacks occur when a modular robot or a modular robotic system is stolen either through a robber or smuggled by an insider. **Malicious Control:** This type of attack occurs when an insider user misuses the system to carry out rogue tasks depending on the access levels assigned to them.

2.4.3.4/ TARGETED COMPONENTS

In IoT and IoMRT, attacks usually target one of the modular robotic system components before spreading in a cascading manner to reach and target other components. Therefore, it is essential to highlight which components are usually targeted while briefly stating what security measures and countermeasures (i.e. cryptographic and non-cryptographic solutions) [418] can be adopted to sort this problem.

Firmware components: Are vital for modular robots as they dictate functionality, regulate communication between modules, and execute tasks, thereby defining the robot's behavior and capabilities. However, they are prone to exploits and attacks. **Firmware Attacks:** Usually exploit firmware codes usually stored on flash memories. This can be through modification, false injection, or deletion of the stored code. Therefore, authentication methods must be set along with a threshold line to monitor any deviation in the firmware's activities.

Operating system components serve as the backbone of modular robots, providing essential functionalities such as task scheduling, resource management, and communication protocols, which are crucial for their overall performance and functionality. However, they are prone to exploits and attacks. **Operating System Attacks:** Take place, especially against older OS versions, or new untested versions such as arbitrary code execution, and root-kit attacks. Therefore, constant updates must be made along with the update of new security patches to mitigate this growing risk and reduce this threat to an acceptable level.

Application components: Are essential for modular robots as they determine specific functionalities and tasks that the robots can perform, enabling them to adapt to various environments and applications, thus enhancing their versatility and utility. However, they are prone to exploits and attacks. **Application Attacks:** Especially untested, unverified, or third-party applications are prone or cause a variety of attacks mainly viruses, Trojans, and worms along with another type of attacks [81]. Therefore, applications must be tested and verified from a trusted source.

Coding components: Are crucial for modular robots as they provide the instructions and algorithms necessary for the robots to execute tasks, interact with their environment, and communicate with other modules, enabling them to achieve desired behaviors and functionalities. However, they are prone to exploits and attacks. **Coding Attacks:** Codes if ill-written or unchecked, are prone to a variety of security attacks such as malicious code injection attacks, malicious modification attacks, endless looping, and many others [215]. Therefore, especially in IoT, codes must be checked and tested to avoid this cascading fatal error.

Memory components: Are essential for modular robots as they store data, instructions, and configurations necessary for their operation, allowing them to retain information about their environment, past actions, and internal states, which enables them to adapt their behavior and perform tasks efficiently and effectively. However, they are prone to exploits and attacks. **Memory Attacks:** Also take place through privilege exploitation, or unauthorized access gained to cause buffer overflow memory attacks buffer errors or return-to-libc attacks [96]. This can be sorted by adopting a well-defined authentication and authorization process, as well as the adoption of memory protection methods such as anti-viruses and Intrusion-Detection/Prevention Systems (IDS and IPS).

Software components: Play a crucial role in modular robots as they encompass the algorithms, protocols, and control mechanisms that govern their behavior, enabling them to

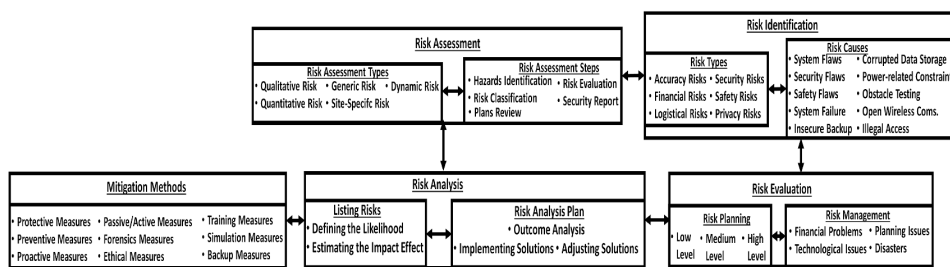


Figure 2.12: Proposed IoMRT Risk Life cycle.

perform various tasks, communicate with other robots, and coordinate their actions within a distributed system, thereby facilitating collaboration and achieving desired objectives. However, they are prone to exploits and attacks. **Software Attacks:** Usually take place by targeting the virtual part of the software via a variety of malware attacks [194]. Protective measures include constant update patches, secure and safe backups as well as security updates. Now, it is important to identify the main IoMRT-related risks to identify both the likelihood and impact of the possibility of these attacks against the main IoMRT components.

2.5/ IoMRT-RELATED RISKS

The aim of this part is to highlight the main IoT issues, vulnerabilities and attacks that surround and target the modular robotic systems (IoMRT) specifically, and the robotic domain (IoRT) generally. This adoption, despite having a huge positive impact with a greater contribution to the improvement of IoT sectors, except that it left the modular robotic systems prone to larger variety of risks and threats. In fact, an IoMRT proposed risk assessment work is summarised in Figure 2.12. Therefore, the aim is to identify the risk types and the main causes of these risks which are presented below. As a result, the main attack types, threat types and natures, along with the main risks are presented below. In fact, a qualitative risk assessment table Table 2.2 was also presented for modular robotic systems as part of IoMRT.

2.5.1/ RISK IDENTIFICATION

Similar to any other IoT-related field, modular robots are prone to a variety of risks that jeopardize their functionality and operational availability, while at the same time targeting one or many of its main aspects including safety, security, privacy, and accuracy. Therefore, it is important to understand each type of these risks, as well as their main causes to ensure that mitigation can be adopted to reduce risks to acceptable levels.

Table 2.2: Proposed Qualitative Risk Assessment Analysis for Modular Robotic Systems.

Attack Target	Targets			Risk			Security Measures						
	Type	Motive	Security	Safety	Privacy	Likelihood	Impact	Severity	Act	Protective	Preventive	Proactive	Awareness
Operator-based	Insider-based	High	Yes	Possible	Yes	High	Significant	Significant	Deliberate	✓	✓	✓	✓
	Social-Engineering	High	Yes	Possible	Yes	High	Significant	Significant	Deliberate/Accident	✓	✓	✓	✓
	Phishing-based	High	Yes	Possible	Yes	High	High	High	Accident	✓	✓	✓	✓
	Privilege Exploit	Significant	Yes	Yes	Possible	Moderate	High	High	Deliberate	✓	✓	✓	✓
	ID-based	Significant	Yes	Possible	Possible	Moderate	Significant	Significant	Deliberate	✓	✓	✓	✓
Gateway-based	Persistent Gateway	High	Yes	No	Possible	Significant	High	High	Deliberate	✓	✓	✓	N/A
	Packet Mistreating-based	High	Possible	No	Yes	High	High	High	Deliberate/Accident	✓	✓	✓	N/A
	Routing Table Poisoning	Significant	Yes	No	Yes	Significant	High	High	Deliberate	✓	✓	✓	N/A
	Packet Injection-based	High	Yes	No	Yes	High	High	High	Deliberate	✓	✓	✓	N/A
	False Base Stations	Significant	Yes	Possible	Yes	Moderate	High	High	Deliberate	✓	✓	✓	✓
Network-based	Eavesdrop/MiTM	High	Yes	No	Yes	High	High	High	Deliberate	✓	✓	✓	✓
	Jamming-based	High	High	High	No	No	Significant	High	high	✓	✓	✓	N/A
	Packet-based	High	Yes	No	Yes	Significant	Moderate	Moderate	Deliberate	✓	✓	✓	N/A
	Transmission Delay-based	Moderate	Yes	No	No	Significant	High	Significant	Deliberate	✓	✓	✓	N/A
	Coding-based	High	Yes	Yes	Yes	Possible	Significant	High	Deliberate/Accident	✓	✓	✓	✓
Modular Robot-based	Software-based	Moderate	Yes	No	Possible	Moderate	Significant	High	Deliberate	✓	✓	✓	N/A
	Memory-based	Moderate	Possible	Possible	Yes	Significant	Significant	High	Deliberate	✓	✓	✓	N/A
	CPU/PU-based	Moderate	Possible	Possible	Yes	Significant	Significant	High	Deliberate	✓	✓	✓	N/A
	Malware-based	High	Yes	Yes	Yes	High	High	High	Deliberate	✓	✓	✓	✓
	Hardware-based	Moderate	Possible	Possible	Yes	No	Moderate	High	Deliberate/Accident	✓	✓	X	✓

2.5.1.1/ RISK TYPES

There are many risks that surround both IoT and modular robotic domains. These risks can have serious impact and implications on the whole operational performance of modular self-reconfigurable robots and robotic systems. Therefore, it is essential to highlight these main risks to understand more how they affect the robotic systems in order to try and mitigate this threat, especially since security risks are not limited to the security concept, but also include safety and privacy concepts since modular self-reconfigurable and non-self-reconfigurable robotic systems and modular robots are now being adopted in the Internet-of-Things (IoT) domain.

Safety and Security. The integration of modular robots poses multifaceted risks encompassing privacy breaches, safety hazards, and security vulnerabilities, stemming from potential data exposure, physical harm, and susceptibility to cyber threats. More precisely, we discuss each risk. **Security Risks:** Include the possibility of hijack or controlling the modular robot to perform malicious tasks, or stealing the design to develop a similar robot but with bad intentions such as doing harm or material damage. **Safety Risks:** Lie in the possibility of having modular robots or robotic systems being the cause of fatal or non-fatal accidents that threaten the life of a human user, consumer, or operator. **Privacy Risks:** Risk having sensitive business-related and non-business-related information being leaked or stolen by external or internal attackers risking affecting the organization's reputation and business trade, especially in the modular self-reconfigurable robotic domain.

Precision and Accuracy. The precision and accuracy risks of modular robots entail the potential for misalignment or calibration errors, leading to inaccuracies in task execution and compromised performance. Here, we discuss them both separately. **Accuracy Risks:** Aim to ensure that the assigned task should be performed with high accuracy with low true/false-positives and true/false negatives and must be error-free. **Precision Risks:** Lie in potential alignment errors, module misplacements, or inaccuracies in shape transformations, which could compromise task performance or system functionality.

Financial and Economical. The financial, economical risks of modular robots, introduced the logistical risks, which encompass challenges related to initial investment costs, operational expenses, supply chain disruptions, and logistics complexities, which may impact overall efficiency and cost-effectiveness. Here, we discuss them both separately. **Financial Risks:** Including the possibility of suffering financial losses due to the high price of equipment, failure to meet the customers' needs, and failure to reach the intended goal

in the modular robotic domain. **Logistical Risks:** Include risking the possibility of having the logistics chain between developers and end-users being interrupted or halted. Leading to the stoppage of IoT materials, equipment, and devices from being moved towards their intended destination and vice versa. Moreover, it is also important to identify the cause(s) of these risks.

2.5.1.2/ RISK CAUSES

The rise of various robotic security and cyber-security issues, IoT threats, and vulnerabilities led to the robotic systems, applications, and devices alike prone to various (cyber) risks [18, 359]. Therefore, the main cyber-risks along their negative effects are presented as follows:

Modular Robotic Flaws. The flaws that risk affecting the modular robotic domain hinder the seamless integration and optimal performance of modular robotic systems across diverse applications. Our main focus is on system, safety, security flaws. **System Flaws:** running systems especially on simulators if not constantly updated and monitored may result in a security flaw that can be easily exploited by an attacker either in a cyber way in terms of software, or the physical way in terms of hardware equipment. **Security Flaws:** Mostly built modular robots or robotic systems have either no or weak protective security measures. This risks them being prone to a variety of cyber attacks either conducted internally or externally. Hence, security must be considered as early as the built-in design model. **Safety Flaws:** Are usually affiliated with the appliance of modular robots in hazardous environments along with unsafe practices where accuracy plays a key role. Therefore, safety in terms of preventive security measures is required to ensure the safe and secure use of modular robots and modular robotic systems respectively.

Bad Practices. The modular robotic domain is susceptible to various practices lacking basic proactive measures, which undermine its potential for seamless integration and optimal performance across diverse applications. From our point of view, this includes **Insecure Backup:** If modular robots are not safely and securely backed to ensure that they can maintain self-reconfiguration and self-healing capabilities, the whole process will risk being brought down to a halt. Backups must also be made safe from any ransomware attacks with the capabilities to recover. **Illegal Access:** Takes place when unauthorized users gain cyber or/and physical access, or authorized users abuse their privileges to conduct malicious acts mostly via backdoors [70]. This risks damaging the whole modular robotic system, as well as stealing valuable information. **Open Wireless Communication:** Since the communications between modular robots and robotic systems are open

wireless communications, they risk being intercepted, jammed, or interrupted. Hence, communications must be made secure in a real-time manner using a trade-off between security and performance.

Lack of Maintenance. The modular robotic domain faces challenges in terms of maintenance, with issues such as limited accessibility to modules, complex repair procedures, and the need for specialized expertise hindering efficient upkeep and prolonging downtime. Based on our own research, this includes: **Corrupted Data Storage:** If not properly stored in modular robots, processing units may be incapable of helping them in decision-making tasks, which would risk affecting both accuracy and performance. Therefore, data must be structured and well-defined to ensure proper data storage which is also inspected to avoid any ambiguity and redundancy. **Power-related Constraints:** Due to power-related issues, modular robots are prone to various resource-constrained issues including excessive power consumption, draining, and resource-exhaustion. **System Failure:** Especially in modular robotic domains can risk causing a major or cascading system failure or system bug which can target either operational or functional performances or even both. **Obstacle Testing:** The absence of obstacle testing can prove to be a serious risk since when deployed in real case scenarios and environments if the modular robots are not well tested, then surely trouble will take place. Therefore, obstacle testing must always be applied on an ongoing basis. This testing should include but not be limited to environmental obstacle testing, terrain obstacle testing, and hazard obstacle testing. **Battery Issues:** Such as overheating and explosion can be the cause of either intrusion or as a result of random sensor protocol failures. This can be mitigated through safety circuits and battery management systems [262].

As both risk causes and types are identified, it is important to know the life cycle of each risk.

2.5.2/ RISK LIFE-CYCLE

Based on the previously identified risks, it is important to improvise in order to advise a proper risk life cycle, which should be suitable for the newly introduced IoMRT domain. This should help with advising better protective and precaution plans with early detection of risks and evaluation of their severity levels to know what security measures should be adopted as part of mitigation/deterrence part. This life cycle is presented as follows:

2.5.2.1/ RISK PLANNING

Understanding IoT risks is not an easy task, since risks are not limited to one aspect nor limited to one source. However, it is essential to start with identifying these security and safety risks defining them, and evaluating them in accordance to their levels. Before adopting risk mitigation methods to eliminate any threat that offers a serious or potential attack. For this reason, it is essential to advise a well-defined plan in order to be capable of maintaining the right security and safety measures, that also take into consideration the privacy concept, by identifying risks, mitigating them, and reducing risks to an acceptable level, especially since risks are classified in according to three main levels. The risk's severity levels are presented and described as follows:

- **Low Level:** This means that risks are mitigated or do not represent a serious threat. In other terms, risks are within an acceptable level and do not need to be mitigated. However, they should be constantly assessed and monitored.
- **Medium Level:** This means that the possibility of risks occurring is possible, threats are present, and there's a likelihood for an attack to take place. Therefore, security measures should be adopted and taken into consideration as a precaution.
- **High Level:** This means that the threats are imposing a serious risk, and the likelihood of an attack happening is equally as high, with a serious damaging impact. Therefore, security measures must be adopted at once, and emergency and response plans must be adopted and advised.

2.5.2.2/ RISK MANAGEMENT

The adoption of a risk management plan is essential due to its ability to identify, assess, and control combined IoRT and IoMRT threats that surround the modular self-reconfigurable domains including modular robots and modular robotic systems. In terms of risk management it is essential to identify the main topics that risk management usually covers:

Financial Problems. Related to present and future issues and uncertainties related to mitigating risks associated by cyber-physical attacks and security, or by ongoing financial problems related to business losses or issues.

Issues. That are often related to technology and planning. **Technological Issues:** Technological issues may cause impose some serious threat to a given modular robotic organization, which may prove to be very risky. These technological issues may be related to functionality issues, operational issues, performance issues, and accuracy issues

which may hinder the performance of tasks. Hence, risk management plans are highly recommended and required. **Planning Issues:** Are related to poor planning requirements and advised plans such as planning management issues, strategic planning, emergency planning, backup planning, precaution planning, error and legal liability planning which are advised to set a strategy that mitigates risks and overcome threats. Therefore, planning management is essential to manage risks.

Disasters. Such as natural (i.e. environmental conditions, acts of God, etc) and non-natural accidents (i.e. accidents or deliberate acts) may occur and offer some serious risks that may affect the modular robotic organization's performance, especially its availability. Hence, a risk management plan is required including backup plans.

Therefore, the adoption of a successful risk management plan will ensure that the full range of risks that surround the modular robotic organization are addressed, especially the risks with cascading effects or impact. This means that the organization is capable of meeting its strategic goals via the established strategic plans which rely on enforcing the adoption of the right security management concept.

The adoption of a risk management strategy would also ensure that these organizations are capable of focusing on both internal and external threats which increases the chances of mitigating both internal and external risks by making smart risk decisions based on smart risk management strategies that also take into consideration the positive risks concept that offer a positive impact.

2.5.2.3/ RISK ASSESSMENT

Following the combined IoT safety and security planning based on the adoption of the risk management concept, the next step is to assess each risk. Therefore, a risk assessment method will be adopted to identify and classify risks between the above-mentioned levels so medium risks are sorted, while high risks are dealt with by taking real-time active security measures to reduce them to a much more acceptable level through mitigation. However, in order to assess risk, it is important to follow these main steps first before classifying risks according to their types.

Risk Assessment Types: Ahead of assessing risks, it is important to identify what is the risk assessment types [25] in order to have that knowledge that helps in adopting the next risk assessment steps that prove to be vital to mitigate threats and reduce the IoRT/IoMRT risk's likelihood and impact. **Qualitative Risk Assessment (QIRA):** Adopts a subjective and general approach that focuses on identifying both IoRT/IoMRT risks and threats [406] by measuring both their likelihood and impact on a given modular

robotic system. **Quantitative Risk Assessment (QnRA):** Relies on numerical values and uses verifiable data in order to identify both risks and threats while evaluating their likelihood and impact [510] in a much more accurate manner compared to the Qualitative Risk Assessment method. In Quantitative Risk Assessment, IoRT/IoMRT risks are assessed based on real-time data and information especially in terms of cost, delays, and resource/energy consumption to produce more real-time and accurate results. This risk assessment type improves the project risks and offers a better understanding of risks along with their trigger conditions, effect and impact against targeted modular robotic systems. Also, methods on how to mitigate these risks are also presented in accordance to the available budget. **Generic Risk Assessment (GRA):** Aims to identify and assess hazards that are common and hazards that are already known, covering general areas of operations that surround the modular robotic domain including locations, activities, and areas of operations [268]. GRA focuses on common hazards as well as their likelihood and impact in case of their occurrence, along with the security measures and precautions that can be generally adopted to mitigate them without diving deeper into details. **Site-Specific Risk Assessment (SSRA):** Or Site-Specific Hazard Assessment (SSHA) is a specialized risk assessment method that is constantly advised, planned, and adapted to a given robotic site while containing real-time details and information about this specific modular robotic project [408]. This includes the hazards that surround them in this given site in addition to the specialized security measures and risk-control methods that should be adopted to mitigate the likelihood and impact of this risk. Unlike the GRA method, SSRA includes a much more detailed approach that focuses in an ongoing manner on the safety, availability and security aspects of robotic systems and working personnel per each deployment site or performed activity. **Dynamic Risk Assessment (DRA):** Is useful in decision-making due to its ability to assess and analyze a modular robotic work environment in a real-time manner following an ongoing process to mitigate threats and reduce the occurrence and likelihood of a given risk from happening. This allows offering the best method to mitigate risks [170]. DRA is often adopted to assess IoRT/IoMRT safety and security risks in a first response form, such as the adoption of emergency response and incident response plans to mitigate risks and threats alike upon their detection and in a real-time manner.

Risk Assessment Steps: Following the identification of the main types of risk assessment, further steps are required in order to identify, classify, and evaluate risks, before setting a security report and reviewing the adopted plan for further improvements against both threats and risks alike. **Hazards Identification:** Risks cannot be assessed before identifying what are the main hazards that surround a given organisation especially those involved in modular robotics and robotic systems. These hazards can either be natural, technical, or human accidents or even deliberate acts such as physical tampering or cy-

ber attacks. **Risk Classification:** Upon the hazard identification phase, it is important to start classifying these risks depending on two factors which are the likelihood of the occurrence of this risk, and the impact of this risk in case of its occurrence. Upon this classification, risks are divided into three main levels: low, medium, and high, of which security measures will be adopted to ensure active real-time protection. **Risk Evaluation:** Once risks are classified, the next step would be to ensure that they are evaluated by answering the questions of "What will be harmed?" and "How will it be harmed?". Once these two key questions are answered, the action to prevent their occurrence will then be taken via a well-advised and combined security-safety plan. **Security Report:** Upon evaluating risks, security gaps and vulnerabilities are discovered and reported. As a result, the combined security-safety plan is set to review which are the most active and effective security measures that can be added and deployed to the modular robotic system to offer a higher level of security without affecting its performance. **Plan Review:** Once risks are identified, classified, evaluated, and mitigated based on the presented security report, the risk assessment plan is reviewed before being approved to ensure that the plan is ready, secure, and well-set, and if there are any further needs to adjust it before applying it.

2.5.2.4/ RISK ANALYSIS

The importance of the risk analysis phase is to be able to analyze the likelihood and impact of each risk following its identification and IoT classification (i.e. business, financial, security, safety risk, etc) depending on the risk's severity level as mentioned earlier.

Risk Analysis Phases. In this thesis, the risk analysis phases are presented and described as follows: **Listing Risks:** The first phase in the risk analysis domain would be to list all the potential risks that the modular robotic organization may encounter and take them into consideration, especially in terms of impact and likelihood. **Defining the Risks' Likelihood:** Following the listing of risks, the likelihood of the occurrence of each listed risk is evaluated and analyzed depending on their severity level(s). **Estimating the Impact Effect:** After defining the likelihood of each risk's occurrence, the impact is then evaluated in terms of financial, operational, and business losses. **Risk Analysis Plan:** Upon analyzing the assessed risks on the list, determining their likelihood, and estimating their impact, a risk analysis plan is then advised for the purpose of adopting as an IoRT/IoMRT mitigation method. This list is made specifically to adhere to the list of risks, therefore it is not a standard model and is prone to change depending on the changing nature of risks and threats. **Outcome Analysis:** Once the plan is advised and applied for real-life and real-time testing, the presented results which contain a full description of the adopted IoRT/IoMRT security and precaution measures are analyzed to see the plan's effect on the performance and the security performance including its strengths and

weaknesses (i.e security gaps). **Adjusting Solutions:** Depending on the result analysis, the solution is adjusted to ensure that a trade-off is set between the modular robotic systems and the adopted security measures. This will make sure that the performance is not affected and is still maintained with a high availability level while implementing the highest security, safety, and precaution measures. **Implementing Solutions:** Once the solution is adjusted and IoRT/IoMRT security/safety issues and bugs are mitigated. The solution that focuses on the risk analysis plan is then ready to be deployed and implemented in order to be actively operational. In other terms, the point of risk analysis is to identify and quantify any potential risk after their classification and prioritization including accidents based on technical issues, human errors, and infrastructure failures and attacks including physical tampering, viruses, and cyber-attacks. Once risks are analyzed, mitigation methods should be presented.

2.5.2.5/ MITIGATION METHODS

Mitigation methods include the adoption of the right IoRT/IoMRT security measures to mitigate the risk of both the likelihood and occurrence of a given incident whether it was an accident or whether it was an attack. For this reason, it is important to know which IoT security measure or the number of security measures need to be adopted and deployed in order to offer a higher level of security to mitigate any threat, overcome any attack, and reduce the risk of having modular self-reconfigurable robots and robotic systems being targeted to an acceptable level. As a result, these main security measures are presented as follows:

Defensive Measures. They can take the form of: **Protective Measures:** Include adopting precautionary actions and procedures to defend against any potential attack whether it is a cyber, physical, or cyber-physical type. **Preventive Measures:** Include establishing security measures in the form of steps that need to be taken in case of an incident to mitigate this threat and prevent it if possible. This is done by developing a security strategy with prevention plans to overcome any attack ahead of its occurrence. **Proactive Measures:** Include the preventive actions taken to mitigate the likelihood of an occurring incident by decreasing the likelihood of a given IoRT/IoMRT attack and reducing the risk impact to an acceptable level. Proactive measures also include methods to mitigate the damage caused by a given accident or malicious incident. **Backup Measures:** Include the adoption of secure backup methods to maintain both functional and operational availability in case of accidents or deliberate acts (i.e ransomware) to ensure that modular robotic systems are still capable of maintaining a level of availability that ensure that their operations remain ongoing.

Passive and Active Measures. Both measures are essential to secure communication links and channels to prevent any possible passive or/and active attack attempt. **Passive Measures:** Include the adoption of periodic monitoring methods such as the passive monitoring of modular robotic networks and communication channels without modifying any traffic, mainly through inspection and issuing alerts upon detecting malicious or suspicious traffic. **Active Measures:** Include adopting active monitoring actions that can be combined with passive measures to offer a higher level of IoRT/IoMRT security protection. Active measures are usually taken actions when passive measures alert the adopted security measures that a given modular robotic system is issuing malicious or suspicious traffic, or there's an ongoing suspicious traffic heading from an unknown source toward the modular robots or robotic systems. Active measures can mainly be protective, preventive, or proactive.

Ethical and Forensics Measures. These measures are essential to investigate an incident or simulate a realistic attack scenario to evaluate both defensive and attacking measures at the same time to enhance security and safety. **Forensics Measures:** Include adopting incident register and recovery methods that allow the building of a well-defined map that shows how a modular robotic vulnerability or gap was exploited and how a given attack took place. This is one of the main "lessons learned" mechanisms that can be adopted to improve all the security measures described above [483, 486]. **Ethical Hacking Measures:** Are often adopted to evaluate the level of security of IoMRT systems and devices, along with their response against potential events mostly malicious (i.e. attacks or accidents). This is often achieved through a set of scenarios that simulate real events to ensure much more accurate mitigating approaches. In this case, these measures can be split into two: attacks against IoMRT's MSRRs or attacks being carried out via these MSRRs.

Training and Simulation Measures. These measures are essential to ensure a higher level of readiness to mitigate and react to any given threat while evaluating its risks. **Training Measures:** include training working personnel depending on their profession within the modular robotic domain, since it is not enough to only rely on securing systems without training their operators in terms of security awareness to overcome social engineering and phishing attack types. **Simulation Measures:** Include the adoption of ethical hackers to detect any security gap, reveal how it can be exploited, and highlight the main security measures that should be adopted to mitigate this threat and reduce its risk for exposure. This also includes highlighting the main strong points and weaknesses of a given modular robotic company or organization to help them adopt the right security plan for IoT [485]. Since risks are now identified, we need to seek suitable solutions that can be adopted

and used to mitigate these security-safety-related threats, vulnerabilities, and attacks.

2.6/ PRESENTED MODULAR ROBOTIC SYSTEMS SOLUTIONS

Several existing solutions were presented to ensure that the proprieties of self-reconfiguration of Modular Robotic Systems (MRS) especially in IoT domains such as IoRT and IoMRT, respectively. For this reason, this section presents and discusses the different types of available solutions that adhere to different limitations, challenges, and drawbacks that surround the 3D modular robotics domain.

2.6.1/ THEORETICAL ROBOTIC SOLUTIONS

Various comprehensive reviews and surveys were also conducted by various authors to shed more light on the importance of modular robotic systems, as well as the progress made regarding the robotic domain, highlighting their characteristics, advantages, drawbacks, and challenges.

Angluin et al. explored the computational power of networks of small resource-limited mobile agents while defining two new models of computation based on pairwise interactions in [19]. Issues surrounding the computational complexity, cost, size, cheap hardware, and resource-constrained devices were also highlighted. Open problems and future directions were further discussed, such as the characterization of the power of stable computation, the access by uniform random sampling, the stable computation model, and the interaction graph were also highlighted. Fitch et al. described how a distributed planner in [138] ensures that self-reconfiguring robots that are made up of heterogeneous modules change their configuration without the need for any additional space. This new work is based on their two previous works [135, 137] including the reconfiguration planning for homogeneous robots and heterogeneous reconfiguration planning that requires temporary working space for execution. Future work includes the investigation of several relative position constraints to overcome them. Hou et al. presented a computational complexity analysis of optimal reconfiguration planning problem surrounding the chain-type modular robots in [195]. For this matter, a theoretical proof was provided along an efficient procedure to estimate both lower and upper bounds for the optimal solution. Future work was aimed towards the evaluation of the reconfiguration algorithms with the objective to their performance for modular and reconfigurable robots.

In [332], Patitz presented a survey that introduced the basic concepts of tile-based self-assembly with a special focus on the algorithmic nature of its newer theoretical models such as abstract Tile Assembly Model (aTAM), kinetic Tile Assembly Model (kTAM) and 2-Handed Assembly Model (2HAM) while providing an overview on their results to pre-

vent and correct errors. Future work aims to achieve the development of newer and more complex models, as well as the understanding of which of these models can be effectively and efficiently built while relying on both theoretical and experimental understanding of self-assembling systems. Ahmadzadeh et al. presented, discussed, and analyzed the challenges and gaps that tackle the enhancement of MRS capabilities in [6] including the hardware's robustness, and the development of the right software and algorithms. Therefore, 64 solution methods and algorithms developed in 125 research papers were reviewed and classified according to their applicability per operation while defining their capabilities. Their challenges, advantages, and weaknesses were also analyzed. Future work aims to investigate the research areas of MRS algorithms to explore promising research directions. Chalk et al. presented and discussed the problem of designing robust, fair coin-flipping systems in [71] which are essential for the implementation of randomized self-assembly algorithms by addressing the limited control issues over species concentrations. The presented work was extended to reach distributions of at least two outcomes, as well as non-uniform distributions. Future work is focused on applying randomization in self-assembly to computing functions.

In [105], Derakhshandeh et al. investigated the feasibility of solving fundamental problems relevant to programmable matter by using the already presented geometric amoebot model in [102] to ensure efficient local-control algorithms. Future work was aimed at identifying the minimum set of key geometric properties to achieve the fully functional work of the already presented algorithms. Boemo et al. have shown how the localized DNA computation circuits can be analyzed similarly to the distributed systems in [48]. The authors also discussed how the used software and theory can be improved through their combined use. Di et al. studied the shape of distributed systems of programmable particles and considered the shape recovery problem in [110]. As a result, a solution based on the non-faulty anonymous particles was presented.

Naz et al. focused on lattice-based modular robots that use neighbour-to-neighbour communications in [303]. Challenges that surround the complex distributed algorithms for programmable matter in massive-scale lattice-based networks were also presented especially in terms of latency and reliability. The approach is only suitable for small networks, hence suffering from scalability and packet collision issues. Future work included experimenting with the practical impact of the huge diameter and average distance of massive-scale lattice-based networks to ensure both the design's efficiency and effectiveness. Network properties of modular robotic systems which use hybrid communication models will also be studied. Daymude et al. presented a comprehensive review in [97] while discussing the use of distributed algorithms under the amoebot model and its variant the hybrid programmable matter for a variety of tasks such as shape formation, shape recognition, object coating, compression, shortcut bridging, and separation. Two distinct algorithm types (deterministic and fully stochastic) were presented for amoebot model

type. Future work considers several improvements such as generalizing the amoebot model to three-dimensional space, extending the amoebot model to incorporate energy costs, and developing a general framework for fault-tolerant algorithms along with other algorithms for multiple robots.

This can be summarized in Table 2.3.

Table 2.3: **The Adopted Existing of Theoretical Robotic Solutions.**

Information		Solution	
Date	Authors	Type	Description
2006	Angluin et al. [19]	Overview	Reviewing the networks' computational power
2007	Fitch et al. [138]	Overview	Reviewing the reconfiguration planning for robots
2010	Hou et al. [195]	Analysis	Analysing the reconfiguration planning problem
2014	Patitz [332]	Survey	introducing the basic concepts of tile-based self-assembly
2015	Ahmadzadeh et al. [6]	Analysis	Analysing the challenges and gaps that surround the MRS domain
2015	Chalk et al. [71]	Overview	Discussing the problem of designing coin flipping systems
2015	Derakhshandeh et al. [105]	Overview	Investigated the feasibility of solving programmable matter problems
2015	Boemo et al. [48]	Analysis	Analysing the localised DNA computation circuits
2018	Di et al. [110]	Study	Studied the shape of distributed systems of programmable particles
2018	Naz et al. [303]	Analysis	Focused on lattice-based modular robots
2019	Daymude et al. [97]	Review	Presenting algorithm types for amoebot model type

2.6.2/ MOBILE ROBOTIC SOLUTIONS

Mobile robotic solutions were also presented to ensure robotic mobile capabilities such as moving, carrying objects and changing shapes. The main solutions are presented as follows.

Yim et al. presented a class of distributed control algorithms based on a “goal-ordering” mechanism for 3D metamorphic modular robotic system reconfiguration, especially the Proteo model in [493]. Modules can move to one of their open neighbour sites following certain motion constraints. Performance results show that the algorithms are distributed and ready to implement. In [64], Campbell & Pillai described the Collective Actuation (CA) as a novel technique that coordinates the efforts of many tiny modules to achieve larger movements and forces. This technique also ensures that the actuator's range and

capacity can be partly fungible via the ensemble's topology and algorithm's design, with the ability to bend large-scale complex structures to achieve the realization of large-scale joints at low control complexity. Testing results based on Theoretical and Physical experiments and simulations show that CA techniques exert a higher force than a single pair of modules. Future work aimed to study the effect of the limited inter-module friction and the dynamic reconfiguration to enhance robotic systems. In [290], the architecture of a multimedia sensor network was presented by Mostefaoui and Piranda using a real-time 3D reconstruction. This architecture was dedicated to video surveillance. This proposal aims to address the system resources optimization by reducing network bandwidth and the consumption of video data fusion/exploitation. Real experiments indicate that the captured device can fulfill the target application requirements even when using low/medium resolution video.

Sproewitz et al. presented a decentralized approach as a reconfiguration strategy for self-reconfiguring modular robots and building blocks for moving furniture such as Roombots (RB) in [415]. Simulation tests show the effectiveness of the presented solution while investigating the seeding order's influence on the goal structure. The development of homogeneous, self-reconfiguring modular robot systems (i.e. Roombots) was also achieved with future work being focused on adopting advanced seeding recipes, runtime metamodule changes, and cyclic movements-based reconfiguration.

Hołobut et al. presented a way to arrange spherical modules into microstructures in [190], using fixed connections to build a skeleton within the Programmable Matter (PM). Two variants of linear-actuator microstructure were also presented and studied. Future work aims to study other actuator microstructures of different properties. Bourgeois et al. presented the progress made that surrounds the cyber-physical conjugation within the Claytronics project in [55]. Moreover, ways that transfer a cyber representation into the matter through the reliance on the programmable matter for cybermatics were also presented. Future work may include studies related to physical constraints transferring a cyber representation since it is a complex process closely linked to the physical capabilities per Catom.

Pescher et al. introduced and explained in detail a Generic Assembly Planner by Constrained Disassembly (GAPCoD) that operates on all modular robot kinds in [340]. The adaptability of the method was tested to different constraints and physical modules. Thalamy et al. presented a novel deterministic and distributed method to rapidly construct an object's scaffold from an organised modules reserve in [444]. This model is said to be parameterisable with a Face-Centered-Cubic (FCC) lattice structure to overcome deadlocks and avoid collisions. A framework for constructing scaffolded shapes was also presented in sub-linear time and with high parallelism.

A novel approach was presented by Lopez et al. in [263]. The approach is based on self-healing property of a synthesised hydrogel to attach and detach robotic modules via

water and without the need for external energy. Tensile, fatigue and adhesion tests were presented to demonstrate the mechanical performance and evaluate it.

These can be summarised in the following Table 2.4.

Table 2.4: **The Adoption of Existing Mobile Robotic Solutions.**

Information		Solution	
Date	Authors	Type	Description
2001	Yim et al. [493]	Overview	Presenting goal-ordering distributed control algorithms
2008	Campbell & Pillai [64]	Review	Coordinate the tiny modules efforts for larger movements
2009	Mostefaoui and Piranda [290]	Approach	Presenting a real-time 3D reconstruction
2010	Sproewitz et al. [415]	Approach	Presented a decentralised approach as a reconfiguration strategy
2014	Holobut et al. [190]	Approach	Presented a way to arrange spherical modules into microstructures
2016	Bourgeois et al. [55]	Approach	Presented the progress that surround the cyber-physical conjugation
2020	Pescher et al. [340]	Approach	Presented GAPCoD to operate on modular robots
2020	Thalamy et al. [444]	Approach	Presented a method to rapidly construct an object's scaffold
2022	Lopez et al. [263]	Approach	Presented a self-healing property to attach and detach robotic modules via water

STATE OF ART

3.1/ INTRODUCTION

This thesis presents an advanced up-to-date study on the self-reconfiguration of modular robots and robotic systems including swarm robots, which were mentioned, highlighted, and discussed by various papers [50, 422, 349, 440]. Moreover, the paper extends the presented work in [141, 6] and has worked on gathering the most related and recent structures and algorithms to the modular self-reconfigurable robotic domain. Further research was also conducted regarding the main modular robotic systems' characteristics and applications mentioned in [7], in addition to their limitations and challenges, discussed in [491], with more complementary information being added, and more details with examples being presented.

Both Claytronics and self-assembly concepts which were presented in [163] and [332], are further studied and detailed (section 3.2.3). Also, the idea of programmable matter, presented in [162, 171] is further explained. Other key concepts related to self-organization and self-adaptability which were mentioned in [294, 383] are further detailed and described in this thesis.

In addition to all of the mentioned above, this thesis further contributes by presenting its added work in the following list:

- **Security Concepts:** This thesis presents, highlights, and classifies the IoMRT, especially from a security point of view and safety concept, by discussing available solutions and proposing others.
- **Cyber-Physical Security Aspects:** This thesis generally takes into consideration the main threats and risks that affect the modular robotic domain, especially in terms of self-reconfiguration, as well as the most prominent type of attacks that may potentially target the modular robotic domain (i.e eavesdropping, packet interception/manipulation, misconfiguration, etc).
- **Modular System Mapping:** The whole modular robotic system mapping is also presented and divided into four main parts which are: Operator, Gateway, Communication

and Modular Robot, (see Figure 2.9) to present a clear illustration of the IoMRT domain.

- **Proposed IoMRT's Components Security Framework:** An IoMRT security framework is also presented to highlight the main attacks and weaknesses of every IoMRT component, as well as the most suitable and ideal security measures and countermeasures.
- **Safety & Privacy Aspects:** Both safety and security best practices of modular robotics systems for IoT are presented and discussed to achieve the needed trade-off without affecting their performance.
- **Proposed System Mapping Security Framework:** That takes the Defense-in-Depth (DID) the main possible security solutions that can be applied as active measures and countermeasures against Attack-in-Depth (AID) cases that actively target IoMRT and their Modular Robots components.
- **Drawbacks & Challenges:** limitations, drawbacks, and challenges that surround the current and future states of modular robotic systems are presented and discussed.
- **Self-Reconfiguration Aspect:** This thesis presents the main characteristics, components, and advantages that surround the self-reconfiguration domain in modular robots and robotic systems.
- **Risk Assessment:** This thesis presents a risk assessment methodology that can be adopted once the security aspect and concept are grasped by modular robotic systems and robots to mitigate threats and reduce both the likelihood and impact of any given risk.
- **Threat Identification:** This thesis identifies threat sources, and types and also classifies them in an easy-to-understand manner, making it easier to track them and mitigate them.
- **Presented Solutions:** This thesis presents, discusses, and analyses most of the modular reconfigurable and self-reconfigurable robots and robotic systems solutions including recent and up-to-date solutions.
- **Modular Robotic Classification:** Which is presented based on their movement and module types, structure, and communication nature.
- **IoSRT future concept:** The IoSRT concept is also highlighted as it will be a key part of the future of IoT alongside the IoMRT, where the full focus is on the IoMRT, before taking this concept and applying it to the IoSRT by merging swarm modular self-reconfigurable robotic systems into the IoT.

3.2/ DETAILED STATE-OF-ART

3.2.1/ FRAMEWORKS & MODULE ROBOTIC SOLUTIONS

Frameworks and modules were also presented to enhance the performance of modular robotic systems to allow them to ensure higher performance, efficiency, and accuracy at a reduced cost and in a timely manner. For this reason, various solutions were presented. Fukuda et al. presented an optimal structure decision method for fixed/mobile base type manipulators to determine the cell type, arrangement, degree of freedom, and link length in [145]. The solution was presented to overcome the issues that surround the self-reconfigurable Cellular Robotic Systems (CEBOT). Simulation results show the efficiency of the presented method in terms of optimal structure decision method and the adopted communication system to ensure CEBOT's ability to become a self-organizing universal manipulator. Yoshida et al. presented a distributed reconfiguration for a 3D re-configurable structure with many identical mechanical units in [494]. This adopted stochastic relaxation Markov Random Field (MRF) method enables a given structure to transform itself into the recommended structure in a way that allows each unit with identical software/hardware to play any role in a given system and apply self-repair upon damage. Simulation results show the effectiveness of this method in terms of self-assembly and self-repair. Bojinov et al. presented a new approach to self-configuration which is suitable for modular robots in [50]. Proteo's simulation experiments stated that the solution is suitable for the creation of "emergent" structures with the desired functionality. As for future work, the authors aim to use this approach as part of an overall hierarchical control scheme in conjunction with other self-reconfiguration or control methods to handle complex tasks. Unsal et al. presented a multi-layered planner for the motion of modules to reconfigure modular robotic systems in [456], along with the introduction of the idea of meta-cubes. Experimental results show that modules have high performance, on-board computation, and power with inter-module communication capabilities. Future work aims to add new modules to the physical implementation. Unsal et al. also presented a class of 3D modular robotic bipartite systems with a self-reconfiguration property in [457] that addresses the motion-planning problem of bipartite modular systems. Results show the feasibility, task orientation, and energy efficiency of the adopted system design. Future work includes refining the hardware implementation for small semi-autonomous modules.

Rus & Vona presented the Crystalline Atom module as a basis for homogeneous, unit-modular, self-reconfigurable robot systems in [377]. The Crystalline module concept and the basic motions that allow a Crystalline robot system to self-reconfigure were also described. Simulation results have shown its efficiency as a planner for automated shape metamorphosis. Roderich et al. presented a study on the utility of self-assembling

robots, before presenting a framework with results of two experimental sets of s-bots that physically connect to self-assemble to adapt to environmental conditions that overcome their ability to operate individually and enhance their decision making [453]. Dewey et al. presented and analyzed a meta-modules theory with an associated distributed asynchronous planner for lattice-based modular robotic systems to address the non-holonomic motion constraints in [107]. The presented approach allowed the shape transformation to be divided into a planning task and a resource allocation task. The effectiveness of the presented metamodule abstraction has shown how a simple but effective planner can serve as a guideline to maintain global connectivity and for the construction of metamodules on any modular robotic system. Future work aimed to enhance the performance of the presented solution to optimize the planning algorithm and implement a much more effective resource manager. Future work includes the adoption of dynamic reconfiguration and self-assembly on large-scale systems. Gorbunov et al. described the implementation of a metamorphic robotic system in [165] to overcome the NP-complete problem of using a minimal number of reconfiguration steps surrounding the self-reconfigurable modular robots. Experimental results show that the approach can be used as an efficient planner. In [232], Knaian et al. performed a shape reconfiguration experiment using a four-segment Millimeter-Scale Motorised Protein (Milli-Motein), which is the highest-resolution chain-type programmable matter system. Experimental results show that it can switch from a straight line to a prescribed shape in 5 seconds, consuming only 2.6 Watts of power during reconfiguration, while holding its shape even without power. Future work was aimed toward the development of improved technologies for three-dimensional free-form fabrication and assembly of robotic systems at a reduced cost. Pinciroli et al. presented ARGoS (Autonomous Robots Go Swarming) as an open source novel multi-robot simulator in [341]. ARGoS is both a flexible and efficient simulator that simulates complex experiments involving large robot swarms of different types. Experimental results show that ARGoS achieves higher performance, efficiency, and flexibility than other existing simulators. Future work aims to improve the ARGoS design to provide real-time performance for hundreds of thousands of robots.

Rubenstein et al. presented Kilobot in [372] as a low-cost robot designed to test collective algorithms on thousands of robots to solve the issues of cost, time, or complexity. Kilobot is capable of running Scalable Distributed Self-Assembly and Self-Healing (SDASH) along with other collective behaviours. Future work included making Kilobot's hardware and software design open for public use, along with the building of a 1024 Kilobot along with the implementation of SDASH on it. Fitch & McAllister presented a general framework for reconfiguration along with an example implementation for SuperBot-style modules, especially for lattice-based and hybrid robot types in [139]. The approach has two main characteristics such as not needing to hand-code meta-module motions and

ensuring the reconfiguration with a lesser number of modules. Future work included testing the implementation of this presented algorithm on other modules. Woods et al. presented an actively computational self-assembly model to study the complexity of self-assembled structures with active molecular components in [476]. This model is based on biology's fantastic ability to assemble bio-molecules that form systems with complex structures and dynamics. Results show the efficiency of this model against passive self-assembly models especially in terms of self-assembled shapes and patterns. Mathieson et al. revealed how properties of folding pathways over a 2-base strand differ from those over a 4-base alphabet in [276]. The paper also shows how to efficiently find min-barrier, pseudoknot-free pathways from initial to final Minimum of Free Energy (MFE) structures. An efficient algorithm for constructing such a pathway was also provided. Results show how folding pathways with temporary and/or repeated base pairs can have lower energy barriers than pathways without such base pairs. Future work aimed to study questions surrounding the use of available software tools for folding pathway and energy barrier prediction. In [146], Furcy et al. developed a "3D temperature 1" optimal encoding construction based on the "2D temperature 2" optimal encoding construction of Soloveichik and Winfree to answer questions presented by Cook, Fu and Schweller's paper. Results reveal the effectiveness of their optimal tile complexity in a three-dimensional variant of Winfree's abstract Tile Assembly Model.

Thachuk et al. identified the sources of leak pathways in existing DNA Strand Displacement Systems (DSD) schemes before presenting a simple, domain-level motif for the design of leak-resistant DSD systems in [438]. The presented schemes are capable of implementing combinatorial Boolean logic formulas and can be extended to implement arbitrary chemical reaction networks. In [7], Ahmadzadeh et al. presented Module-Information-Task-Environment (MITE) as a novel framework to define and characterise the Modular Robotic System's (MRS) applications and properties, while also providing a methodical review on MRSs along with an up-to-date and comprehensive list of hardware specifications of modular robots. An analysis was also provided covering trends, research gaps, challenges and open problems to establish a link between the MITE and MRS planning and control algorithms. Gmyr et al. investigated the problem of shape formation with robots on tiles to rearrange the set of movable tiles to achieve the right shape in [161]. The results show that the arbitrary number of tiles can achieve the formation of simple shapes when using a single operating robot. Future work was aimed at investigating how multiple robots can cooperate to speed up the simple or complex shape formation process. Holobut et al. presented a distributed procedure suitable for autonomous reconfiguration planning in [191] that allows a robot to predict whether the next planned reconfiguration step will overstress intermodular connections. Despite it suffering from several drawbacks related to simplicity and accuracy issues, except that

it achieves an acceptable efficiency level. Tucci et al. presented Constructive Solid Geometry for Programmable Matter (CSG4PM) as an efficient method to overcome memory and computational time issues per module in [451]. CSG4PM was compared with three other solutions, showing that CSG4PM also requires less decoding time. Future work aims to cut off the received CSG tree per catom before the sending of different parts to its neighbors.

Thalamy et al. presented an approach that uses scaffolding to overcome the motion constraints which affects the speed of the self-reconfiguration process in [439]. This novel approach was introduced to scaffold-based self-reconfiguration of large modular robotic ensembles. A parametric scaffolding model was also presented to increase parallelism, supports mechanical stability, and simplifies planning and coordination. Simulation results show that this approach achieves a high throughput with no congestion. Future work aims to replace the resource requests by continuously feeding 3D Catoms up every scaffold branch.

Romanishin et al. presented in [369], a decentralised control framework for lattice-based Modular Self-Reconfigurable Robots (MSRR) that facilitates the neighbour identification using a novel magnetic fiducial system. Results to ensure the efficiency and scalability of this framework were achieved by testing twelve 3D M-Block robotic modules. A new self-reconfiguration method that is compressible and expandable was presented by Bassil et al. in [32].

The method is based on organising modules in meta-modules that form a 3D porous structure while flowing in parallel to avoid path blocking. Results show how the structure can be self-reconfigured from its initial to its given goal shape.

These solutions can be further summarised in Table 3.1.

3.2.2/ ALGORITHMIC ROBOTIC SOLUTIONS

Algorithmic solutions were also presented to ensure smoother simulations in a realistic environment at a reduced cost and in a timely manner, covering a higher number of modules and robots that are being simultaneously tested. Such solutions were also presented to ensure that modular robots are capable of self-configuring to transform into a different shape with fewer errors and less time. For this reason, several solutions were presented.

Kotay & Rus presented algorithms for planning the robotic Molecules' motion on a substrate of other Molecules in [236]. A scaffold planning approach was also presented to increase parallelism and remove blocking and color constraints from the configuration planning, along with a new gripper-type connection mechanism for the Molecule which does not require any power to maintain a connection. Future work includes more planning toward achieving an enhanced version of the low-level offline movement, a

Table 3.1: The Adoption of Existing Frameworks & Module Robotic Solutions

Information		Solution	
Date	Authors	Type	Description
1990	Fukuda et al. [145]	Method	Determines the cell type, arrangement, degree of freedom and link length
1998	Yoshida et al. [494]	Method	Allows each unit to play any role and apply self-repair
2000	Bojinov et al. [50]	Approach	Suites the creation of “emergent” structures with the desired functionality
2001	Unsal et al. [456]	Plan	Has high performance, on-board computation, and power with inter-module
2001	Unsal et al. [457]	MRS Class	Achieves feasibility, task orientation, and energy efficiency
2001	Rus & Vona [377]	Module	Allows a Crystalline robot system to self-reconfigure
2001	Roderich et al. [453]	Framework	Allows s-bots to self-assemble and enhance their decision making
2008	Dewey et al. [107]	Approach	Allows shape transformation to be divided into planning and a resource allocation task
2011	Gorbenko et al. [165]	Implementation	Overcomes the NP-complete problem of using a minimal number of reconfiguration steps
2012	Knaian et al. [232]	Experiment	Switches from a straight line to a prescribed shape in 5 seconds, consuming only 2.6 Watts
2012	Pincirolini et al. [341]	Simulator	Simulates complex experiments involving large robot swarms of different types
2012	Rubenstein et al. [372]	Algorithm	Runs SDASH and other collective behaviours
2013	Fitch & McAllister [139]	Framework	Ensures the reconfiguration with a lesser number of modules
2013	Woods et al. [476]	Model	Achieves higher efficiency against passive self-assembly models
2015	Mathieson et al. [276]	Algorithm	Achieves lower energy barriers than pathways without such base pairs
2015	Furcy et al. [146]	Model	Achieves an effective optimal tile complexity
2015	Thachuk et al. [438]	Motif	Implements combinatorial Boolean logic formulas
2016	Ahmadzadeh et al. [7]	Framework	Characterises the MRS applications
2017	Gmyr et al. [161]	Solution	Achieves the formation of simple shapes when using a single operating robot
2017	Holobut et al. [191]	Procedure	Suitable for autonomous reconfiguration planning
2017	Tucci et al. [451]	Method	Overcomes memory and computational time issues per module
2019	Thalamy et al. [439]	Approach	Overcomes the motion constraint
2019	Romanishin et al. [369]	Framework	Facilitates the neighbour identification using novel magnetic fiducial system
2022	Bassil et al. [32]	Module	Overcomes path blocking

library for the scaffold planner, and ensuring that the gripper connection mechanism passes its prototype stage. Vassilvitskii et al. presented a distributed planning algorithm

for a modular robot system in [460], that creates any 3D shape including intelligent objects such as concavities and internal holes. The algorithm can create close-packed structures and also prevents blocking positions since it considers kinematic constraints. Future work wishes to extend the algorithm to have more parallel docking positions on a 3D lattice and this method with other algorithms to control the movements of each module. In [60], Butler and Rus presented the PacMan algorithm that is suitable for distributed actuation and system planning with 2D or 3D unit-compressible modular robot systems. A correctness analysis was made for the two given versions of the algorithm. The algorithm was tested on the Crystal robot, and the results show that the presented solution can avoid deadlocks and disconnection. Their implementation presented the scalability and efficiency of the algorithm via parallelism, while several encountered limitations were highlighted and discussed.

Fitch et al. presented an algorithmic basis for heterogeneous self-reconfiguring systems in [136]. Simulation results show a good algorithmic feasibility, which is suitable for the development of applications that rely on the use of heterogeneous self-reconfiguration and hardware systems while presenting both centralized and decentralized out-of-place solutions. Future work was aimed at enhancing the presented solution to achieve better results and overcome many obstacles. Stoy & Nagpal presented a scale-independent approach that relies on a two-step process of self-reconfiguration in [423]. This algorithm controls the self-reconfiguration process via a growth process. Experimental results show that the price of the self-repair capability is high, especially in terms of time steps, moves, and communication messages, which is not an ideal solution. Stoy also presented an approach to solving the issues that surround the self-reconfiguration process in [420]. This approach is said to be a systematic, scalable, and convergent solution. Simulation results show that this approach is more efficient in terms of overcoming self-reconfiguration process complexity issues. Future work aims to try and adopt this approach to a self-reconfigurable robot.

Fitch et al. defined a free space by an arbitrarily shaped bounding region in [137]. A novel heterogeneous reconfiguration algorithm was also presented to plan module trajectories via the structure's module which is split into two phases: shape-forming, and sorting the goal configuration. Also, Fitch & Butler presented a locomotion technique that performs both planning and actuation control in sub-linear time and memory in [135]. The algorithm is based on reinforcement learning and is known to use dynamic programming to plan module paths in parallel. A novel localized cooperation scheme was also presented along with other self-reconfiguration algorithms. Fitch et al. described how a distributed planner in [138] ensures that self-reconfiguring robots that are made up of heterogeneous modules change their configuration without the need for any additional space. This new work is based on their two previous works including the

reconfiguration planning for homogeneous robots and heterogeneous reconfiguration planning that requires temporary working space for execution. Future work includes the investigation of several relative position constraints to overcome them. Rubenstein & Nagpal presented in [374], Kilobot as a simple modular robot designed to work in a collective to self-assemble and self-heal the collective shape. This was achieved by relying on an algorithm that ensures the ability of the collective to self-assemble and self-heal while keeping the shape size proportional to the collective robot number.

In [284], the design, prototyping, and control of a modular 2D modular and self-reconfigurable robot for conveying microparts was presented by Mobes et al. using actuators, electronics and micro-controllers, as well as Electro-Permanent (EP) magnets which were used for both the linkage and the traveling system to avoid any power consumption and conserve energy during the linkage. An algorithm was also presented for all block units, allowing the reconfiguration of a set of blocks from one spatial configuration to another using real-time simulator software. In [123], El-Baz et al. presented a scalable distributed iterative algorithm to convey fragile and tiny micro-parts, as well as to control the block motion of a reconfigurable micro-electromechanical modular surface. The system is said to be suitable for Micro-Electro-Mechanical Systems (MEMS) such as semiconductors manufacturing, micro-mechanics, and the pharmaceutical industry due to its reconfigurability, flexibility, scalability, and optimality. In [301], Naz et al. presented an ABC-Centre as a scalable iterative algorithm for electing an approximate-center module in modular robots. The accuracy of the algorithm was tested using Blinky Blocks systems. Results show that the ABC-Centre is suitable for large lattice-based modular robot ensembles with low memory resources [35].

In [103], Derakhshandeh et al. presented a general algorithmic framework for shape formation problems in Self-Organising Particle Systems (SOPs) using the spanning forest primitive and the snake formation primitive algorithms. Kawano presented a reconfiguration algorithm for a robot composed of sliding cube modules with a limited motion primitive in [220]. Despite the limitation of motion primitive which complicates the development of the reconfiguration algorithm, except that it simplifies the design of module hardware, reduces the size of manufacturing costs, and ensures that the algorithm can be applied in an environment with obstacles. Kawano also presented a full-resolution reconfiguration algorithm for a heterogeneous modular robot composed of sliding cube modules with a limited motion primitive in [221]. The algorithm overcomes several restrictions related to the design of modular robots. Experimental results show the correctness and completeness of the algorithm, especially for three-dimensional connected structures where the reconfiguration is executed in quadratic operating time cost. Future work included the improvement of the algorithm's improvement for application in environments with obstacles. Cannon et al. presented a Markov chain-based algorithm to overcome the compression

issues surrounding the geometric amoebot model in [65]. This is done by exploiting the memoryless, stochastic nature of Markov chains to achieve particle compression in the amoebot model. Simulation results show how their algorithm offers provable guarantees of its behaviour. Kawano designed a permutation algorithm in [222] using Limited Sliding Cubes with full resolution, which can be executed in the used space by the tunnelling robot. The algorithm uses a three-dimensional meta-module to maintain both the robot structure's connectivity and the existence of mobile modules. A video was presented to show the correctness and completeness of this algorithm. However, the algorithm will not be merged with the algorithm presented in [221]. Zhu et al. presented a distributed, hybrid, and parallel mechanism for decentralized self-reconfiguration of Modular Self-Reconfigurable (MSR) robots in [512]. Simulation results show that this mechanism is scalable, efficient, convergent, and robust to modules' failure. Future work aims to tackle the reliability of local communication between the modules and the manual reproduction design rules of L-systems-driven self-reconfiguration of modular robots.

Tucci et al. presented a complete, local, and parallel reconfiguration algorithm for metamorphic robots with a loose quadratic upper bound on the total module movements' number in [452]. This algorithm can perform in-place parallel distributed reconfiguration in worst-case quadratic time, as well as local decision-making and completeness of reconfiguration. Pescher et al. presented a new concept to make the design phase of car development much easier and more interactive in [338]. Their solution consists of using a modular robot combined with a shape memory polymer to create an interactive car piece model. Hence, the Non-Uniform Rational Basis Spline (NURBS) dichotomy algorithm was used to evaluate the method's accuracy through the simulation of a polymer over the Catoms which were re-organized into the desired shape. Simulation results show a high accuracy level which increases once the Catoms in use are smaller. In [299], Naz et al. presented the k-BFS SumSweep algorithm suitable for modular robotic ensembles named LMRs, which are suitable for lattice-based and resource-constrained distributed modular robotics known as LMRs. The presented k-BFS SumSweep algorithm was used to elect an approximate center node in LMRs. Simulation results on hardware modular robots and a simulator for a large robots ensemble show that the algorithm offers a good trade-off in terms of accuracy which varies between 92% and 100% with high efficiency in terms of communication and time and limited memory footprint.

Pescher et al. introduced a new concept that aims to make the car development process more interactive and much easier in [339]. This presented global algorithm mixes the self-reconfigurable autonomous Modular Robots and Shape-Memory Polymer to manage the interaction with both users and the self-reconfiguration of programmable matter to mold the polymer surface over the 3D Catoms in whichever desired shape. This was done by adopting the Non-Uniform Rational Basis Splines (NURBS) using a dichotomy algorithm to evaluate the accuracy of this method. Thalamy et al. presented an improved and asyn-

chronous version of a previously presented distributed self-reconfiguration algorithm to build a parametric scaffolding structure using micro-robots in [443]. The algorithm in use has a local motion coordination algorithm and pipelining techniques to avoid collisions or deadlocks. Results indicate the scalability of these modules where the small motion time variations have a negligible impact on the whole reconfiguration time. Thus, making the algorithm robust to physical variations. Thalamy et al. addressed the self-configuration problem in large-scale modular robots before introducing the coating problem in [441]. An assembly method was provided to construct coating from a reserve of modules in a sandbox form using the Tucci algorithm and their Border Completion algorithm. The authors also stated that sandbox, scaffold construction, and coating can be used to speed up the construction of modular robotic objects without dramatically increasing the risk of collisions between modules or deadlocks.

Bassil et al. presented a robust clustering algorithm with linear complexity based on graph cut for large modular robot ensembles which is based on a distributed density-cut graph algorithm that divides the networks into a predefined number of clusters based on the final goal shape in [31]. Both implementation and demonstration were made on a real Blinky Blocks system for evaluation. Simulation results show that the performance is affected by the modules' number, the ensembles' shape, and the clusters' number.

In [421], Stoy presented a concept of non-random and non-fixed dynamic initial configurations that dynamically develop in response to the evolutionary process. As a result, a competitive co-evolutionary algorithm was implemented to show how both configurations and controllers evolve together to perform a mobile robot obstacle avoidance task. A distributed ID assignment algorithm was presented for modular robots by Assaker et al. in [23]. The algorithm ensures fast and efficient inter-module communications by maintaining easily calculated routes, and by ensuring the removal and the addition of system particles. A reconfiguration algorithm for shape-shifting modular robots with a triangular structure was presented by Gerbl et al. in [154]. This novel approach is based on extended binary trees and achieves the self-reconfiguration of Planar Adaptive Robot with Triangular Structure (PARTS) configurations, which are presented by unstructured triangular meshes. However, this algorithm still has shortcomings, especially in terms of collision avoidance constraints.

This is further summarised in Table 3.2.

3.2.3/ SIMULATION-BASED SOLUTIONS

VisibleSim [109] has proven itself to be a very reliable and usable simulator for realistic testing with simulative scenarios that can depict a challenging environment similar to the real-life one where testing can be made easily at a lesser time, lower computation cost, resource consumption and suitable for resource-constrained modular robotic de-

Table 3.2: The Adoption of Existing Algorithmic Robotic Solutions

Information		Solution	
Date	Authors	Type	Description
2000	Kotay & Rus [236]	Approach	A Scaffold planning approach to increase parallelism
2002	Vassilvitskii et al. [460]	Algorithm	Creates any 3D shape
2003	Butler & Rus [60]	Algorithm	suitable for distributed actuation
2003	Fitch et al. [136]	Algorithm	An algorithmic basis for heterogeneous self-reconfiguring systems
2004	Stoy & Nagpal [423]	Algorithm	Controls the self-reconfiguration process
2004	Stoy [420]	Approach	Solves the self-reconfiguration issues
2005	Fitch et al. [137]	Algorithm	Plans module trajectories
2007	Fitch & Butler [135]	Algorithm	Performs both planning and actuation control
2007	Fitch et al. [138]	Algorithm	Distributed planner for self-reconfiguring robots
2010	Rubenstein & Nagpal [374]	Algorithm	Ensures the ability of collective to self-assemble and self-heal
2012	Mobes et al. [284]	Algorithm	Allows the reconfiguration of a set of blocks
2014	El-Baz et al. [123]	Algorithm	Conveys and controls block motion
2015	Naz et al. [301]	Algorithm	Elects approximate-centre modules
2015	Derakhshandeh et al. [103]	Algorithm	Sorts shape formation problems in SOPs
2016	Kawano [221]	Algorithm	Overcomes several modular robot design restrictions
2016	Cannon et al. [65]	Algorithm	Achieves the particle compression in the amoebot model
2017	Kawano [222]	Algorithm	Maintains the robot structure's connectivity and the existence of mobile modules
2017	Zhu et al. [512]	Mechanism	Ensures scalability and robustness to modules' failure
2018	Tucci et al. [452]	Algorithm	Performs in-place parallel distributed reconfiguration
2018	Pescher et al. [338]	Concept	Make the design phase of car development much easier
2018	Naz et al. [299]	Algorithm	Suitable for lattice-based and resource constrained distributed modular robotics
2019	Pescher et al. [339]	Algorithm	Makes the car development process more interactive and easy
2019	Thalamy et al. [443]	Algorithm	Builds a parametric scaffolding structure using micro-robots
2020	Thalamy et al. [441]	Algorithm	Constructs coating from a reserve of modules in a sandbox
2020	Bassil et al. [31]	Algorithm	Divides the networks into a predefined number of clusters
2020	Stoy [421]	Algorithm	Dynamically develops in response to the evolutionary process
2022	Assaker et al. [23]	Algorithm	Ensures fast and efficient inter-module communications
2022	Gerbl et al. [154]	Algorithm	Achieves self-reconfiguration of PARTS configurations

vices. As a result, several solutions were conducted and presented including: in [109], where Dhoutaut et al. reported the progress achieved in the design of VisibleSim which efficiently mixes a discrete-event core simulator with discrete time functionalities and targets intelligent objects and/or robots such as Smart Blocks and Blinky Blocks. Experiments revealed that the VisibleSim can accurately simulate 2 millions of nodes at a 650k events/sec rate on a simple laptop. Piranda et al. presented a proof-of-concept of a self-reconfigurable robot based on sliding blocks in [349]. Experimental results show the effectiveness of this solution, where blocks can move along one another at an average speed of 16.4 mm/s with a holding force of 45 mN. Future work includes reducing the target map to local positions, as well as using this work as a basis to realize the Smart Blocks project. Another future improvement includes the use of an Electro-Permanent Magnet (EPM) to sense the rotor block position.

Bourgeois et al. presented CO2Dim, which stands for Coordination and Computation in Distributed Intelligent MEMS in [53] to overcome the faults and challenges of distributed intelligent Micro-Electro-Mechanical Systems (MEMS) such as scalability and faulty behaviours. Unlike other solutions, CO2Dim bridges the gap between simulation and real testbeds. The approach's efficiency still requires further demonstrations.

A distributed reinforcement learning strategy for morphology-independent lifelong gait learning for modular robots was presented by Stoy et al. in [79], where identical controllers are run on all modules. Experimental results tested on simulated and physical modular robots to show its effectiveness. Roderich et al. presented HiGen as a high-speed genderless mechanical connection mechanism for the docking of robotic modules and for self-reconfigurable modular robots, which allows its connectors to join with one another, which allows either side to disconnect in case of failure [328]. HiGen is classed as the fastest connection mechanism that reduces the needed time for modules to connect and disconnect when conducting complex self-reconfiguration tasks. In [302], Naz et al. presented a Modular Robot Time Protocol (MRTP) which is a network-wide time synchronization protocol for modular robots. The presented solution was evaluated using the Blinky Blocks hardware. Experimental results show a low-level time-stamping and clock skew compensation using linear regression. Future work include considering both centrality and clock stability in the time master election.

Naz et al. presented a Cylindrical-Catoms SelfReconfiguration (C2SR) as a parallel, asynchronous, and fully decentralized distributed algorithm to self-reconfigure lattice-based MSRs. C2SR was developed in the Claytronics project in [298]. The simulation was conducted on a simulated physical environment (VisibleSim). Results show the effectiveness of the used algorithm, especially in terms of communications, movements, and execution time. In [343], Piranda and Bourgeois presented a fully distributed rule-based algorithm for large modular self-reconfigurable robots, which are made from cubic modules (blocks). The use of motion rules was also presented to drastically simplify the

complexity of the sliding movements to fasten the reconfiguration process. The robustness of the algorithm was evaluated using “*VisibleSim*” simulator. Future work aims to work on a 3D reconfiguration with cubic Modular Self-reconfigurable Robots (MSR) to show the flexibility of the motion rules and to extend the work with the spherical robots of the Claytronics project.

Derakhshandeh et al. presented a universal shape formation algorithm that consists of systems of computationally limited but self-organised devices in [104]. However, the algorithm only relies on local information while requiring a constant-size memory per particle. Future work was aimed at investigating more complex descriptions of to-be-built shapes. In [344], Piranda et al. presented a Catom, as a quasi-spherical structure for micro-robots [55, 171] to address to the shape issues and other constraints that surround them, especially large-scale lattice-based modular robots. Moreover, rules that can be used in self-reconfiguration algorithms by programmers were also presented.

Piranda & Bourgeois conducted a study of different scenarios to validate the ability to move and propose suitable methods to manufacture self-reconfigurable spherical micro-robots such as the Claytronics project in [344]. A detailed model named “catom” was also presented for the realization of a quasi-spherical module for realizing programmable matter. Catom including the 3D-printed catoms shells, are the key elements that constitute any large-scale lattice-based modular robots in the Claytronics project. Piranda & Bourgeois presented a detailed quasi-spherical catom model that overcomes all the constraints to build programmable matter in [345]. The model is said to ease movements using a semi-curved shape and also provide sufficiently strong connectors.

Romanishin et al. presented a work that was built upon existing similar research and that outlined the specifications, designs, and algorithms for a new modular self-reconfigurable robotic system [366]. The use of Magnetic Lead Screws (MLS) actuators was also presented due to their high mechanical efficiency with the ability to separate and reattach.

These solutions are presented and summarised in Table 3.3.

3.2.4/ ROBOTIC SECURITY SOLUTIONS

Despite the beneficial advantages that modular and self-reconfigurable robots and robotic systems offer, they still remain prone to a variety of security issues including gaps and attacks. For this reason, several security solutions were presented to mitigate this threat and offer a safer and more secure adoption and usage of modular self-reconfigurable robotic systems.

Gonzalez et al. presented an architecture based on self-reconfiguration to overcome versatility solutions and allow the implementation of hardware-accelerated secure applications in FPGA-based embedded systems [164]. Cryptographic co-processors were also deployed to prove the feasibility of the presented solution with a Secure Shell (SSH) appli-

Table 3.3: The Adoption of Simulation-based Robotic Solutions

Information		Solution	
Dates	Authors	Type	Description
2013	Dhoutaut et al. [109]	Design	Efficient VisibleSim for intelligent objects/robots, and design with simulation results.
2013	Piranda et al. [349]	Concept	An effective self-reconfigurable robotic solution based on sliding blocks.
2013	Bourgeois et al. [53]	Approach	CO2Dim in Distributed Intelligent MEMS that overcomes scalability and faulty behaviours.
2013	Stoy et al. [79]	Strategy	Allows identical controllers to run on all modules.
2014	Roderich et al. [328]	Mechanism	HiGen allows connectors to join with one another, which allows either sides to disconnect in case of failure.
2016	Naz et al. [302]	Protocol	MRTP, a network-wide time synchronization protocol for modular robots.
2016	Naz et al. [298]	Algorithm	C2SR, a parallel, asynchronous and fully decentralised distributed algorithm to self-reconfigure lattice-based MSRs.
2016	Piranda & Bourgeois [343]	Algorithm	Fully distributed rule-based algorithm for large modular self-reconfigurable robots made from blocks.
2016	Derakhshandeh et al. [104]	Algorithm	Consists of systems of computationally limited but self-organised devices.
2016	Piranda et al. [55]	Structure	addresses to shape issues and other constraints related to large-scale lattice-based MSRs.
2018	Piranda & Bourgeois [344]	Study	Proposes suitable methods to manufacture self-reconfigurable spherical micro-robots such as Claytronics project.
2018	Piranda & Bourgeois [345]	Model	Overcomes all the constraints to build programmable matter.
2022	Romanishin et al. [366]	Presented Work	Uses Magnetic Lead Screws actuators due to their high mechanical efficiency and ability to separate and reattach.

cation being developed in a low-cost commercial device to run a standard operating system. Kepa et al. reviewed the hardware attacks against the Field Programmable Gate Array (FPGA) Reconfigurable Computing (RC) systems, and presented a method to secure system credentials generation and trusted self-reconfiguration, using a secure reconfiguration controller (SeReCon) and partial reconfiguration (PR) in [225]. SeReCon provides a root of trust (RoT) for RC systems to provide integrity-maintaining self-reconfiguration by analyzing each new Internet Protocol (IP) core structure before reconfiguration and was tested on a number of security attack scenarios. Hourany et al. presented a new se-

curity protocol that is both optimised and dedicated for IoT programmable matter in [197]. The protocol overcomes the distributed architecture's traditional security protocols and encryption algorithms by relying on the use of lightweight cryptography that uses the same encryption protocol as a hashing function. Results have shown the efficiency of the presented solution.

Wang et al. addressed the security threats at the design and implementation stage of an autonomous mobile robot and presented the RoboFuzz design as a directed fuzzing study of critical environmental parameters affecting the robot in [467]. Detection and mitigation algorithms were also developed to counteract the RoboFuzz's impact. Experimental results show that RoboFuzz has a 93.3% rate of imposing concrete threats with an overall loss of work efficacy is 4.1%. Ferrer et al. introduced a new method to secure cooperation between large groups of robots by encapsulating cooperative robotic missions in an authenticated data structure known as a Merkle tree in [133]. In this method, robots must prove their integrity by exchanging cryptographic proofs. Real-world and simulation results revealed the feasibility of using Merkle trees. Zarrouk et al. presented a hardware-based secure boot mechanism that is lightweight, keyless, and unique for each device in [498]. This solution is based on a Secret Unknown Hash (SUH) which is initiated in a post-manufacturing, unpredictable single-event process.

Parween et al. designed and deployed a self-reconfigurable robot in an actual drain environment in [329]. The platform has a fuzzy logic system for collision avoidance and an adaptive algorithm controller with inertial measuring unit (IMU)-based feedback. The solution was deployed in a lab setting and in a real-time drain environment to demonstrate its effectiveness, level-shifting capability, and auto-correction to maintain a safe distance from the platform's wall. Samarakoon et al. presented a novel method based on Fuzzy Logic Systems (FLSs) to maximize the coverage area by using the shape-changing ability to access narrow spaces with fewer cost requirements and a less explicit set of training data in [385]. The solution was introduced to overcome the limitation of conventional area coverage methods of tiling robots, hence three novel techniques were presented including the Fuzzy Logic System-Genetic Algorithm (FLS-GA), Fuzzy Logic System-Particle Swarm Optimisation (FLS-PSO), and Fuzzy Logic System-Simulated Annealing (FLS-SA).

In fact, this can be further summarised in the following Table 3.4.

3.3/ CONCLUSION

In conclusion, this thesis has made a significant advancing step in the direction of the field of modular robotics, simulation, and self-reconfigurable systems. The exploration of algorithmic solutions has resulted in improved efficiency and effectiveness in simulating

Table 3.4: The Adoption of Existing Robotic Security Solutions

Information		Solution	
Date	Authors	Type	Description
2008	Gonzalez et al. [164]	Application Security	Adoption of cryptographic co-processors to secure applications
2010	Kepa et al. [225]	System Security	Introduction of SeReCon to protect the RC systems' integrity
2021	Hourany et al. [197]	Security Protocol	Usage of lightweight cryptography and hashing
2021	Wang et al. [467]	Robotic Security	Introduction of RoboFuzz to study critical environmental parameters
2021	Ferrer et al. [133]	Robotic Security	Exchange of cryptographic proofs to confirm the integrity
2021	Zarrouk et al. [498]	Hardware Security	Adoption of a keyless lightweight boot mechanism
2021	Parween et al. [329]	Robotic Security	Adoption of fuzzy logic system for collision avoidance
2021	Samarakoon et al. [385]	Coverage Security	Increase of coverage methods of tiling robot

realistic environments for modular robotic systems. These solutions have effectively addressed the challenges associated with conducting simulations with a large number of modules and robots simultaneously while minimizing costs and time requirements.

The algorithms presented in this thesis offer strategies for easier simulations capable of simultaneously handling a higher volume of modules and robots, thereby enabling more comprehensive testing scenarios. Additionally, these algorithms have enhanced the self-configuring capabilities of modular robots, enabling them to transform into different shapes with greater accuracy and efficiency.

Through the implementation of these solutions, this thesis has contributed, in the field of programmable matter, to the advancement of modular robotic systems, making them more adaptable, error-resistant, and suitable for a wide range of applications. Furthermore, the thesis has shed light on the importance of addressing security vulnerabilities and potential attacks in modular and self-reconfigurable robotic systems. Various security solutions have been proposed to ensure the safe and secure adoption and utilization of such systems, representing significant advancements in addressing security challenges associated with modular and self-reconfigurable robotic systems. In summary, the contributions discussed in this thesis represent significant advancements in the field of modular robotics, simulation, and self-reconfigurable systems. The utilization of VisibleSim as a reliable and efficient simulator for realistic testing, along with the introduction of innovative algorithms and protocols, underscores the thesis's impact on advancing the capabilities and security of modular robotic systems, paving the way for safer and more secure deployment in various real-world applications.



CONTRIBUTION

CONTRIBUTION I - THE CONCEPT OF INTERNET OF MODULAR ROBOTIC THINGS

4.1/ INTRODUCTION

The integration of Modular Self-Reconfigurable Robots (MSRR), as part of programmable matter, and robotic systems into the Internet of Things (IoT), led us to the introduction of a new novel concept called the Internet of Modular Robotic Things (IoMRT) including the Internet of Swarm Robotic Things (IoSRT), which has spread into diverse solutions across various IoT-related domains, such as medical, military, law enforcement, space exploration, business, agricultural, and cyber-physical domains. These systems boast self-reconfiguration and self-healing capabilities, addressing the limitations of traditional robotics with minimal human intervention. Modular self-reconfigurable robots offer advantages in re-usability, cost-effectiveness, and adaptability within complex environments, promising complex 3D shapes with ease of operation.

The development of MSRRs remains a significant challenge despite the promising characteristics they offer. These challenges include but are not limited to achieving robustness in self-reconfiguration mechanisms, optimizing energy consumption, ensuring scalability, and addressing security concerns. This thesis provides an analytical view of MSRRs, emphasizing the importance of security considerations and addressing potential threats and risks. Additionally, various approaches by fellow researchers are examined, focusing on prototypes, modules, and algorithms. As the IoMRT domain transitions towards modular and swarm concepts, this thesis also predicts and introduces the emergence of a new novel future IoMRT-based era called the Internet of Modular Swarm Robotic Things (IoMSRT) which is imminent. These systems, capable of autonomous task execution, inter/intra-communication, self-(re)configuration, self-correction, and self-healing, are poised to redefine communication and task performances. The integration

of modular and swarm robotics not only enhances adaptability and efficiency but also enables collaborative problem-solving and autonomous task distribution in dynamic IoT environments (see Table 4.1).

Programmable matter, characterized by small autonomous building blocks, offers programmable capabilities to achieve diverse geometric structures. Communication among robots is crucial for distributed algorithms but is affected by the message size which creates a communication bottleneck. By optimizing data transmission through computational processing, the efficiency of these algorithms can be significantly enhanced. The paradigm shift towards modular robots within IoT has fueled interest in programmable matter. Nanorobots, such as *Blinky Blocks*, exemplify this trend, combining distributed programming with modular structures. While offering flexibility and adaptability, nanorobots pose challenges in security due to resource constraints. Lightweight security protocols are imperative to ensure real-time security without compromising performance. Moreover, advancements in nanotechnology and material science are expected to further enhance the capabilities of programmable matter, paving the way for innovative applications in modular robotics and IoT.

Table 4.1: **Abbreviation table with acronyms and their definitions.**

Acronyms	Definition
AI	Artificial Intelligence
BB(s)	Blinky Block(s)
IoT	Internet of Things
IoRT	Internet of Robotic Things
IoMRT	Internet of Modular Robotic Things
IoSRT	Internet of Swarm Robotic Things
IoMSRT	Internet of Modular Swarm Robotic Things
IoPMoT	Internet of Programmable Matter of Things
LCAPBB	Lightweight Cryptography and Authentication Protocol for Blinky Blocks
LCAPPM	Lightweight Cryptographic Algorithms and Protocols for Programmable Matter
ML	Machine Learning
MR(s)	Modular Robot(s)
MRS(s)	Modular Robotic System(s)
MSRR(s)	Modular Self-Reconfigurable Robot(s)
MSRRS(s)	Modular Self-Reconfigurable Robotic System(s)
SI	Swarm intelligence
UAV(s)	Unmanned Aerial Vehicle(s)
UUV(s)	Unmanned Underwater Vehicle(s)
USV(s)	Unmanned Surface Vehicle(s)
UGV(s)	Unmanned Ground Vehicle(s)
VS	VisibleSim

4.2/ DETAILS OF THE CONTRIBUTION

This contribution can be presented in the following bullet points:

- **State of Art:** Two new novel IoT-based concepts were introduced and studied in terms of safety, security, privacy, and performance, providing a mapping of the modular robotics domain in terms of IoT as part of a broader picture of IoT-based programmable matter. System mapping was also achieved to study the main weaknesses and strong points of modular robots, especially the case of Blinky Blocks as part of programmable matter.
- **Benchmarking:** Where a series of tests were conducted on varying sets of Blinky Blocks, with each set being tested using different message sizes, to identify the main reason behind the communication delay.
- **Enhanced Communication:** By providing compression algorithms capable of reducing the message size/length to reduce the communication time, overhead, and bottleneck.
- **Enhanced Security:** By providing message authentication and lightweight cryptography protocols to secure communications and verify the identities of the communicating parties.
- **Crypto-Compression:** Which will be extended beyond the thesis, where we will apply cryptography to the compressed message, to ensure a higher level of Communication Security (ComSec).

4.3/ COMMUNICATION NATURE & TECHNOLOGIES

Communication is an essential part that the modular robotic domains within the IoT field rely on to establish connections either physically or wirelessly to perform the necessary tasks. The proposed communication system can be divided into two main parts: IoMRT connection based on operator-to-modular robots, and Modular robots connection based on modular robot-to-modular robot (see Figure 2.11). In the following, we present and discuss the main communications technologies that are used or can be used for IoMRT and modular robot communication systems within the IoT domain.

Furthermore, all existing IoT connection types can be employed in the context of IoMRT systems. In the following, we will principally focus on describing all possible IoMRT communication systems. On the other hand, the communication nature of modular robots differs from one modular model to another, especially in the case of swarms and swarm formation.

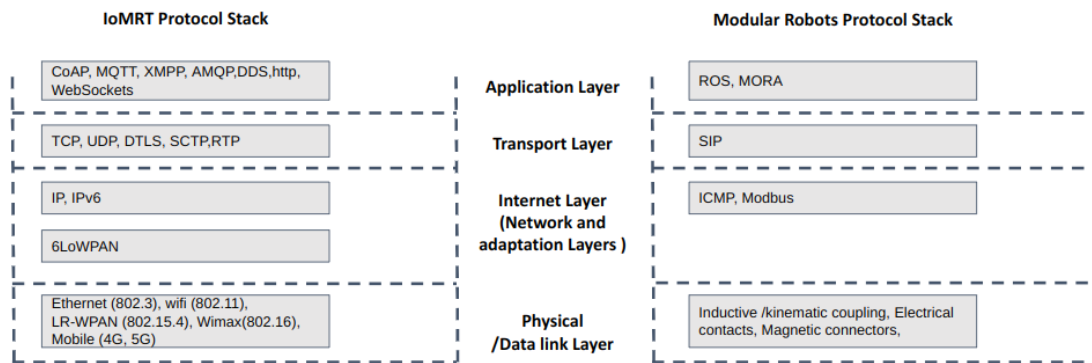


Figure 4.1: A proposed Communication Layer based on IoMRT/Modular Robotics Protocol Stacks

To understand the IoMRT's and modular robots' communication nature, it is important to further understand the main communication layers, their protocols (see Figure 4.1), and their communication algorithms, as well as its main networking topologies, which are presented and detailed below as follows. It is also important to note that not all the mentioned communications layers, protocols, and network topologies are being applied, as some of them are about to see adoption or are classed as theories that are undergoing further testing and simulations before being integrated into the modular robotic systems within the wireless IoT domain.

4.3.1/ IO MRT LAYER PROTOCOLS

IoMRT Layer Protocols are used, and can or may be used in the future for communication and data exchange to enable connectivity, data transmission, and interoperability within the IoMRT domain. Often serves as a set of rules and standards that govern how modular robots and their components communicate with each other by managing connections and defining the format, structure, and mechanisms for transmitting and receiving data.

4.3.1.1/ IO MRT PHYSICAL/DATA-LINK LAYER COMMUNICATIONS

The use of various communication types by modular robots for their Link Layer depends on their specific implementation, design goals, application requirements, and constraints. Their adoption into the IoMRT domain may not be limited to one technology but instead, it may require a combination of these communication types. Here we present a list of the most common communication types of the Link layer that can be used for the modular robotic domain:

Wired Communication: Is established using wires or cables to establish a direct communication link including Ethernet cables, serial cables, or custom wiring solutions. Thus, offering a reliable and high-bandwidth link between modules with fast data transfer and synchronization [168].

Wireless Communication: Allows modules to communicate with each other using wireless protocols such as Wi-Fi, Bluetooth, Zigbee, or proprietary wireless protocols [69] including 3G, 4G/LTE and 5G [365]. Allowing more flexibility and mobility within the modular robot system. Wireless communication includes two main key types for modular robotic systems: **Wireless Personal Area Network (WPAN):** (i.e. Bluetooth and Zigbee) Provides a low-power and low-data-rate communication solution and enables short-range connectivity where modular robots primarily rely on local communication for coordination and cooperation, while exchanging information, coordinating their actions, and synchronizing their behaviors without the need for physical connections [57]. **Low-Power Wide Area Networks (LPWANs):** (i.e. LoRaWAN and Sigfox) Are designed for long-range and low-power connectivity and are used in applications such as Internet of Things (IoT) involving modular robots with the ability to penetrate obstacles and communicate over larger distances [473].

Infrared (IR) Communication: Is a line-of-sight link communication method used by modules that are equipped with IR transceivers to carry data and commands between modules [178], covering short-range communication (e.g. Linbots) [278]. *KiloBots uses IR communication with bound on the ground.*

Radio Frequency (RF) Communication: This Is another wireless communication method used for longer distances, where modules that are equipped with RF transceivers can communicate using radio waves, covering a wider range and overcoming obstacles [8]. Each Kilobot is equipped with infrared transmitters and receivers that allow them to communicate with nearby Kilobots. This communication enables them to coordinate their actions and perform collective tasks, such as formation control, pattern formation, and swarm behavior. In fact, Kilobots use infrared communication due to its cost-effectiveness, low power consumption, localized nature, interference resistance, and suitability for facilitating coordination within a swarm of small robots.

Optical Communication: Uses light signals which carry data and commands to establish communication between modules including technologies such as Light-Emitting Diodes (LEDs) and Photodetectors, providing high-speed data transfer [392]. The key difference between IR communication and optical communication lies in the wavelength of

the electromagnetic spectrum they utilize. IR communication typically operates in the infrared portion of the spectrum, which includes wavelengths longer than those of visible light, while optical communication operates using visible light or near-infrared wavelengths. Therefore, while both IR and optical communication rely on light for transmission, they differ in the specific range of wavelengths they employ. Additionally, optical communication often involves more focused and precise beam directionality, making it suitable for longer-range and higher-speed communication applications compared to IR communication, which tends to be more diffuse and localized.

Acoustic Communication: Is an effective method where line-of-sight or physical connections are not feasible, since it uses sound waves for inter-module communication, where modules that are equipped with speakers and microphones can transceive acoustic signals to exchange data [119].

Bluetooth: Is a wireless communication protocol that enables short-range communication between devices and offers low power consumption. It can be used in modular robot systems where close-range communication and coordination are needed [402].

Wireless Fidelity (Wi-Fi): Provides high-speed data transfer rates and long-range communication, allowing modules to enable secure connectivity and data exchange [334]. and connect to a local area network (LAN) or the internet via a centralized control (i.e. central control unit: such as a base station or a dedicated controller), to achieve remote control and monitoring, seamless integration with other IoT systems and devices.

4.3.1.2/ IOMRT NETWORK LAYER PROTOCOLS

The choice of IoT network protocols for modular robots depends on a set of specific requirements including communication range, data rate, interoperability level, and power consumption. Therefore, the choice of one or a combination of network protocols for modular robots can vary depending on the design, requirements, and applications in use. In this thesis, we list the most commonly used or can-be-used IoT network protocols in modular robotics:

Zigbee: Is a low-power wireless communication protocol specifically designed for IoT applications, including modular robots as it provides secure and reliable short-range communication and coordination for modules [249].

Thread: Is a low-power, wireless networking protocol designed for IoT devices, including modular robots that rely on battery-powered modules, as it enables modular robot modules to form a mesh network, offers easy integration with other IoT devices and cloud-based services, and builds a trusted and secure modular robot network. Thus, allowing modular robot systems to integrate with a wider range of IoT systems and devices [504].

Long Range Wide Area Network (LoRaWAN): Is a Long-Range Low-Power, Wide-Area Network (LPWAN) protocol with low data rates and low power consumption that enables the communication between IoT devices and a LoRaWAN network, making it a suitable centralized control system or a base station for modular robotic systems, especially as a communication protocol between the individual modules, while ensuring tracking, localization and monitoring [389].

Narrowband IoT (NB-IoT): Is a cellular network technology specifically designed for IoT devices that offers wide coverage, deep indoor penetration, and low power consumption. NB-IoT can be used in modular robot systems, allowing modules to establish connectivity with NB-IoT networks and exchange data with other modules or a centralized control system [371]. Thus, ensuring that modular robot modules can communicate over large distances (indoors and outdoors) while operating for extended periods without frequent battery replacement or recharging. This also includes ensuring that the transmitted data is successfully delivered, making them suitable for modular robot systems.

Sigfox: Is a Low-Power, Wide-Area Network (LPWAN) protocol that enables long-range communication with low data rates with a network layer-like functionality, which can be used by modular robots to transmit sensor readings, status updates, or simple commands between robot modules [462]. However, it is not very suitable for modular robot applications that require higher bandwidth or real-time communication.

Z-Wave: Operates in the sub-GHz frequency range and provides secure and reliable communication. However, it is not a very suitable communication solution for modular robots unless they are designed to interact with home automation devices [318], where Z-Wave can be used as a communication medium to integrate Z-Wave-compatible sensors into modular robot systems, and remotely control and monitor Z-Wave-enabled actuators or modular devices.

4.3.1.3/ IOMRT TRANSPORT LAYER PROTOCOLS

The transport layer protocol is responsible for the reliable and efficient real-time data transfer between modules or components within the same modular robot system, to ensure that data packets are delivered accurately and correctly. There is not a specific session layer protocol that is specifically designed for modular robots, except that there are several general-purpose transport layer protocols that can be employed or may be employed in the future.

Transmission Control Protocol (TCP): Provides reliable, connection-oriented communication between modules [356] and can be used by modular robots to establish a reliable connection, ensuring that data is delivered in the correct order and without loss.

User Datagram Protocol (UDP): Is commonly used in robotic systems, despite it being connectionless and not providing reliable delivery or ordering of data, except that it offers lower latency and overhead, making it suitable for modular robots [458].

Datagram Transport Layer Security (DTLS) : Can be used to secure communication between individual robot modules or between the modular robot and a central control system, allowing modular robots to communicate securely even in challenging environments by adding security features to UDP [409]. The use of DTLS protocol in IoMRT is a topic for future research work.

Stream Control Transmission Protocol (SCTP): Combines the features of TCP and UDP, while providing reliable, message-oriented communication, making it suitable for modular robot systems to establish and manage multiple concurrent sessions between modules [267]. The use of SCTP protocol in IoT and IoMRT is a topic for future research work.

Real-Time Transport Protocol (RTP): Is designed to deliver audio and video streams over IP networks, which makes it suitable for use as a session layer protocol in modular robots to enable real-time data transfer between modules, which is essential for synchronization and data packets ordering [150].

4.3.1.4/ IOMRT APPLICATION/SESSION LAYER PROTOCOLS

The choice of application(/session) layer protocol which can be incorporated within the transport layer, depends on the specific requirements of the modular robot system, such

as the nature of the data being exchanged, the desired level of real-time communication, and the needed scalability, which can result in the developing of custom session layer protocols. Though there is not a specific application(/session) layer protocol dedicated solely to modular robots, there are several general-purpose session layer protocols that can be employed.

To facilitate specific functionalities and data exchange between modular robots, various application layer protocols can be employed, where the choice of protocol depends on the desired functionalities and communication patterns between modules. This thesis presents the main application layer protocols that are or can be commonly used in modular robot systems as follows:

Message Queuing Telemetry Transport (MQTT): Is a secure lightweight and efficient publish-subscribe messaging protocol designed for constrained devices and low-bandwidth, high-latency IoT networks [151]. Due to its decentralized and scalable nature, it can be used in resource-constrained modular robotic systems to facilitate efficient and reliable synchronous/asynchronous communication between the modules and other components within the IoT system, as well as cloud services and IoT platforms [144].

Constrained Application Protocol (CoAP): Is a specialized protocol designed for constrained devices and low-power networks, and enables modules in a modular robot system to communicate and interact with each other over constrained networks [274].

Hypertext Transfer Protocol (HTTP): Can also have applications in modular robot systems, particularly for data exchange and control types including establishing a client-server communication model, exchanging configuration data and parameters between modules, transmitting sensor data from individual modules to a central server or data processing unit and facilitating remote monitoring and logging of modular robot activities [224].

Data Distribution Service (DDS): Is often used in distributed systems, including IoT applications, and can be used to facilitate communication and data exchange among modules within a modular robot system. Thus, benefiting from its robustness, scalability, and flexibility to achieve reliable data exchange, coordination, and collaboration among modules [503].

Advanced Message Queuing Protocol (AMQP): Provides reliable, secure, and interoperable communication between modules, as it allows flexible peer-to-peer communica-

tion in a modular robot system [98].

WebSocket: Provides full-duplex communication channels over a single Transport Control Protocol (TCP) connection, enabling secure real-time communication and bidirectional data transfer between modules [192]. It also provides remote control and monitoring of modular robot systems, allows publishing events or updates to connected clients or systems, and enables bi-directional communication [353].

4.3.2/ MODULAR ROBOTIC COMMUNICATION LAYERS

In this part, we will describe the different possibilities of each communication layer of modular robots.

4.3.2.1/ MODULAR ROBOTS PHYSICAL/DATA-LINK LAYER COMMUNICATIONS

Modular robots use a variety of physical communication methods that establish communication between individual modules, depending on the functionalities and needs of the modular robotic domain in accordance with the IoT field. The most common physical communication methods include:

Inductive Coupling: Involves using magnetic fields to transmit data between modules [21], where each module is equipped with an inductive coil that establishes wireless communication by inducing electrical currents in nearby coils [167].

Electrical Contacts: Establish direct electrical communication between modules that physically connect in the form of pins [189], connectors, or conductive pads, allowing data transfer between the modules [282].

Magnetic Connectors: Are used to securely facilitate physical and electrical connections between modules [128], as they rely on magnets to align and hold the modules together, creating a secure connection [501], such as the case of Blinky Blocks [355].

Kinematic Couplings: Are mechanical connections that allow modules to physically connect while maintaining a precise alignment, with the ability to incorporate electrical contacts or optical interfaces for data exchange [351]. A reliable and repeatable connection can be established by using features such as ball joints, pins, or sockets [379].

Pneumatic or Hydraulic Tubes: Are used to establish physical communication [116] by the transfer of compressed air or fluid between modules, to enable communication for power transmission or control signal exchange [117].

Physical Docking Stations: Can be incorporated to provide a standardized interface and alignment mechanism where modules can align to physically connect and communicate [334].

Virtual Docking Stations: Refer to a method where modular robots establish connections without the need for physical contact or mechanical docking mechanisms since they rely on wireless communication protocols and algorithms to facilitate their modules' coordination and synchronization [502].

4.3.2.2/ MODULAR ROBOTS NETWORK LAYER

In this part, we will describe the different possible protocols of the network layer of modular robots.

Inter-Module Communication Protocol (IMCP): Is designed for modular robot systems to facilitate communication and data exchange between modules, enabling coordination, task allocation, and information sharing between modular robots [188].

Modbus: Allows for communication between modular robots, especially in industrial automation and control systems, enabling control and monitoring capabilities by providing a standardized way to read and write data registers [497].

4.3.2.3/ MODULAR ROBOTS SESSION LAYER

In this part, the Session Initiation Protocol (SIP) of the session layer of modular robots is described.

Session Initiation Protocol (SIP): Is used for session management in modular robots communication systems; making it suitable for modular robot systems to establish and control communication sessions between modules, exchange control messages, and coordinate their actions [429].

4.3.2.4/ MODULAR ROBOTS APPLICATION LAYER

In this part, we will describe these two different possible protocols of the application layer of modular robots.

Robot Operating System (ROS): Is a flexible framework for writing robot software that helps build modular robot systems using its own application layer protocol for communication between modules to enable data exchange and control of sensor/command data [364]. It is often used in various applications, including industrial automation, robotics research, autonomous vehicles, drone development, healthcare robotics, and smart home systems.

Modular Open Robotics Architecture (MORA): Is an open-source software framework specifically designed for modular robot systems by encompassing a set of application layer protocols that define communications between modules including information sharing, action coordination, and the execution of distributed tasks [168]. MORA is an innovative framework designed to facilitate the seamless integration and interoperability of modular robotic systems, offering a flexible and scalable platform for researchers and developers to collaborate on the creation of modular robotics solutions across diverse applications and industries.

4.3.2.5/ MODULAR ROBOTS SIMULATION FRAMEWORK

In this part, we will describe the different possible simulation frameworks of modular robots.

Representational State Transfer (REST) - RESTful API: Is an architectural style commonly used in web services that can be leveraged in modular robot systems to expose specific functionalities, enable easy integration with other modular systems, and allow interaction between external applications via Hyper Text Transfer Protocol (HTTP) standard [434].

Distributed Component Object Model (DCOM): Is a proprietary protocol developed by Microsoft for communication between software components across networked computers, and can be used in modular robot systems to facilitate the heterogeneous communication between modules that run on different devices or platforms [440].

Modular Open Robot Control Software Interface (MORSE): Is a simulation and robotics framework that supports the communication and coordination of modular robots by allowing them to interact and share information in (2D/3D) simulated or real-world environments.

C++ Development Library: such as the case of *VisibleSim*, which is designed for researchers that have computer programming experience as it consists in a **C++ framework** for building lattice modular robot simulators controlled by distributed programming. *VisibleSim* takes the form of an open-source project under AGPLv3 license and is available on Github. In *VisibleSim* lingo, the distributed program that is executed on each module during the simulation is named a *BlockCode*. It is effectively the controller of the modules and where users will describe the behavior of the robot in response to all kinds of events whether external (interactions with the world, reception of a message, etc.), or internal (interruption or timer, initialization, end of a motion, etc.). Unlike other simulators where each robot is fitted with a number of sensor and actuator components, this distinction is not materialized in *VisibleSim*. Modules from any type of robot are however fitted with a constant number of *interfaces*, depending on the geometry of their lattice, and which can both be used for sensing connected modules (by examining whether an interface is connected) and communicating with them. In the current state of the simulator communication between modules is only natively allowed in a peer-to-peer manner between connected neighbors.

Distributed Robot Operating System (D-ROS): Is a designed ROS extension for modular robots that enables distributed control and communication among modules so they can operate autonomously while collaborating [239].

Service-Oriented Architecture (SOA): Is an architectural approach that can be adopted in modular robot systems to enable flexible communication and integration, where modules expose their functionalities as invoked services by other modules [76].

4.4/ IOMRT ALGORITHMS

Modular robots need specific basic algorithms to perform more complex tasks. Leader election [198] is a very important task, breaking down the symmetry of robot assemblies (where all modules are identical) by electing a member to organize some of the processing before eventually handing over to another leader. Finding the centroid of a system, i.e. a module placed at the 'network' center of the whole that is a strategic position for

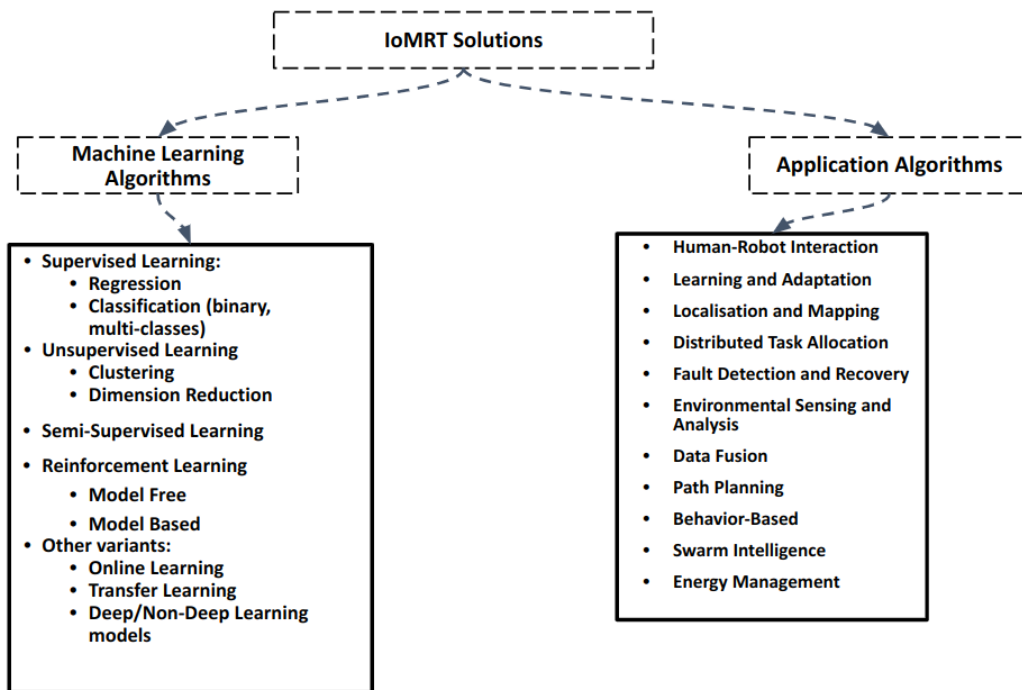


Figure 4.2: Possible Machine Learning Solutions and Application for IoMRT.

communicating with all the others, is an important issue. Naz et al. [299] propose a solution that elects a leader at the center of the network. Time synchronization algorithms are important when robots need to perform coordinated actions, such as activating actuators at the same time. As robot clocks are not always very regular, especially in very small systems, it is necessary to synchronize them regularly. Naz et al. propose a protocol in [300] to evaluate clock drift via message exchange.

Additionally, modular robots rely on a variety of algorithms depending on the task's complexity especially when integrated into the field of IoT, to ensure a much more effective IoMRT environment, including algorithms that may be integrated in the future. As a result, machine-learning-based, application layer-based, as well as other algorithms are presented and discussed (see Figure 4.2).

4.4.1/ MACHINE LEARNING ALGORITHMS

The selection of machine learning algorithms depends on the specific requirements of the modular robot application which depends on the complexity of the task [201]. In fact, hybrid approaches that combine multiple algorithms or use machine learning in conjunction with other control or planning techniques are common in the field of modular robotics. Either way, the choice of algorithms depends on the task and the available sensor data. Here are some commonly used machine learning algorithms in the context of modular robots:

Supervised Learning Algorithms: Can be applied for classification/ regression tasks like object recognition, gesture recognition, or terrain classification. Models can be built by using algorithms such as Support Vector Machines (SVM), Decision Trees, Random Forests, or Neural Networks [211].

Unsupervised Learning Algorithms: Allow models to discover patterns or structures in unlabelled data especially when dealing with large amounts of data. They also include algorithms such as Clustering (e.g. k-means clustering) or to select important information by using dimensionality Reduction (e.g. Principal Component Analysis) algorithms [468].

Semi-Supervised Learning Algorithms: Build a model based on limited labeled and unlimited unlabelled data by using unsupervised learning as a pre-processing step and followed by a supervised algorithm to form a big label dataset. It can be seen as supervised learning extended by unsupervised learning [455]. This type of ML model if employed in the case of data labelling or collection of new observations proves to be a hard task.

Reinforcement Learning Algorithms: enable modular robots to learn through trial and error and optimize their behaviour involving sequential decision-making, like path planning, task allocation, or resource optimization, and can include Q-Learning or Deep Q-Networks (DQN) algorithms [506].

Other Machine Learning variants: include the following list. **Deep Neural Networks:** Are widely used including Convolutional Neural Networks (CNNs) for Visual Perception Tasks, Recurrent Neural Networks (RNNs) for sequential data processing, and generative adversarial networks (GANs) for generating synthetic data or improving robot performance [250]. Models can be supervised, unsupervised, or semi-supervised in addition to be employed also in model-based reinforcement learning. **Transfer Learning:** Transfers learned representations or policies that allow modular robots to quickly adapt to new scenarios or tasks by leveraging knowledge and experience gained from an environment or a task [259]. **Online Learning Algorithms:** Are employed by modular robots to adapt and update their models in real-time whenever a new sensor data is collected [111] and include algorithms such as Online Gradient Descent or Online Random Forests. **Particle Swarm Optimization (PSO):** Can be employed by modular robots to optimize parameters or configurations for tasks like path planning or swarm coordination [95].

Recent work by the FEMTO-ST/OMNI team involves implementing a **distributed Neuronal network** [350]. For this purpose, a pre-trained neural network tasked with recognizing shapes in an image is distributed across a set of robots, with each robot assigned

to process one or more neurons. The input is determined by the sensors of the robots, in this case, an image whose pixels are distributed among the robots. Subsequently, the robots exchange information to facilitate data transfers between the layers of neurons, and eventually, some robots possess information about the shape of the initial input.

4.4.2/ APPLICATION ALGORITHMS

Modular robots can use various application layer algorithms to enable specific functionalities and tasks [49]. However, the choice of these algorithms depends on what the modular robot system needs to accomplish, which may require combining and customizing different algorithms to meet the application requirements to ensure efficient communication, coordination, decision-making, and task execution. Here are some application layer algorithms examples that modular robots can use:

Independence and Autonomy. First, we explore applications that try to give robots independence and self-sufficiency, enabling them to operate independently and execute tasks without constant human intervention and continual human supervision. **Swarm Intelligence Algorithms:** Are deployed to achieve collective behavior properties, by drawing inspiration from natural systems, such as ant colonies or bird flocks to allow modular robots to exhibit "similarly" coordinated and adaptive behaviors. Similar algorithms include particle Swarm Optimisation (PSO), Ant Colony Optimisation (ACO), or Artificial Bee Colony (ABC) algorithms [433]. **Behaviour-Based Algorithms:** Combine simple reactive rules so modular robots can exhibit complex behaviors. Thus, providing mechanisms for coordination and interaction between modules using finite state machines, subsumption architecture, or behavior trees [152]. **Decision-Making Algorithms:** Can range from simple rule-based systems to more complex algorithms like Markov Decision

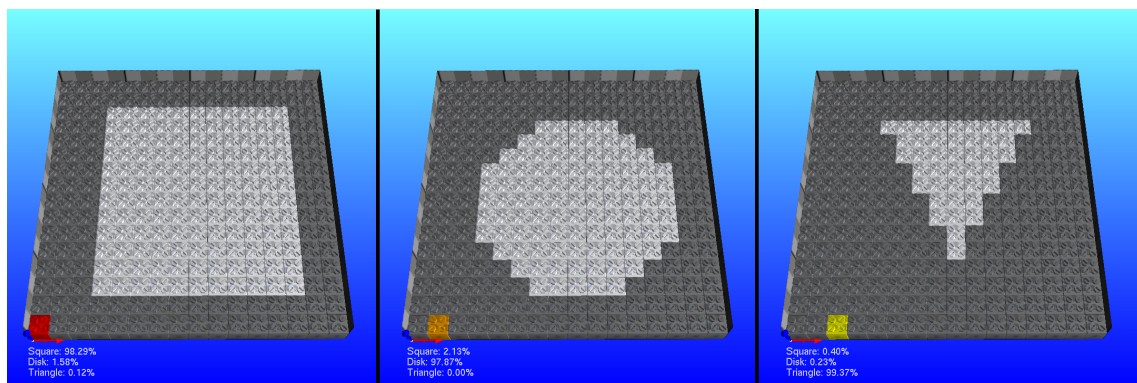


Figure 4.3: Neuronal Network Distributed over a grid of *Blinky Blocks*. Red *Blinky Blocks* at the bottom left corner detects squares, it's orange right neighbor detects disks, and finally the next one in green is for triangles.

Processes (MDP), reinforcement learning, or utility-based decision-making, and are used to enable modular robots to make intelligent choices based on their perception of the environment and their internal state [205]. **Path Planning Algorithms:** Help modular robots determine optimal paths or trajectories to navigate through the environment, by taking into consideration obstacle avoidance, shortest path calculation, or dynamic re-planning factors to ensure safe and secure movement [219]. **Distributed Task Allocation Algorithms:** Are used to allocate tasks among the modules efficiently, while considering factors such as module capabilities, task requirements, and communication constraints to assign tasks to these modules [75]. **Task Coordination Algorithms:** Ensure effective collaboration and synchronization between modular robots by enabling them to communicate, exchange information, and coordinate their actions to accomplish complex tasks, such as object manipulation or formation control [200]. **Human-Robot Interaction Algorithms:** Enable natural and intuitive communication between humans and robots, including reaction to human oral/physical commands, gesture recognition, voice recognition, or facial expression analysis [396].

Environmental Mapping: Detection and Mitigation. Initially, our focus centers on applications geared towards environmental mapping, detection, and mitigation, facilitating robots to perceive and navigate through their surroundings while identifying and addressing potential hazards or challenges. **Communication Protocols and Middleware:** Include the Robot Operating System (ROS), Message Passing Interface (MPI), or the publish-subscribe pattern to facilitate communication and data exchange between modules by defining the structure, rule, and format for data/commands transmission [495]. **Learning and Adaptation Algorithms:** Improve the modular robots' performance over time by enabling them to acquire new knowledge, adapt to changing environments, and refine their behavior via reinforcement/machine learning, or evolutionary algorithms [248]. **Fault Detection and Recovery Algorithms:** Incorporate fault detection and recovery algorithms to identify and handle system failures or malfunctions by monitoring the modular robot's performance, detecting anomalies or errors, and triggering appropriate recovery strategies, such as self-healing, reconfiguration, redundancy activation, or error correction [226]. **Localisation and Mapping Algorithms:** Determine the position of modular robots and create maps of their surroundings using sensor data, such as odometry or visual information, to estimate the robot's location and build a 2D/3D realistic representation of the environment [5]. **Environmental Sensing and Analysis Algorithms:** Gather information about their surroundings by processing sensor data, performing feature extraction, and enabling environmental understanding, such as object/obstacle detection, localization/identification, or mapping [157]. **Data Fusion Algorithms:** Combine and integrate the data collected from modular robots' multiple sensors or sources of information to obtain a more accurate and comprehensive understanding of the environment via sensor

fusion, and extraction of meaningful information for accurate and timely decision-making [13]. **Energy Management Algorithms:** Are applied in scenarios where resources are limited or modular robots are deployed for extended periods to optimize power consumption, battery usage, or energy harvesting strategies to maximize the modular robot's operational time and overall system performance [149].

The choice of algorithms depends on the capabilities of the modular robot system and the desired task to be accomplished. Aside from machine learning and application layer algorithms, modular robots can use various other algorithms to perform different tasks, which are listed below:

Communication and Coordination. Algorithms for communication and coordination in modular robots are essential for enabling seamless interaction and collaboration among individual modules, ensuring efficient task execution, and achieving collective behaviors in complex environments. **Communication and Coordination Algorithms:** Modular robots often require algorithms for communication and coordination among the individual modules to perform cooperative tasks. These algorithms can include consensus algorithms, distributed algorithms, or leader-election algorithms [461]. They perform cooperative tasks and can include consensus, distributed, or leader-election algorithms. **Control Algorithms:** Precisely control modular robots and can include Proportional-Integral-Derivative (PID) control, Model Predictive Control (MPC), or Fuzzy Logic Control (FLC) [158]. **Sensor Fusion Algorithms:** Integrate sensor measurements and provide accurate and reliable state estimation to improve the perception of modular robots and understanding of their environment. Such algorithms include Kalman or particle filters [13].

Mapping and Planning. Mapping and planning algorithms for modular robots are crucial for creating accurate representations of the environment, identifying obstacles and landmarks, and generating optimal paths or plans to navigate through space and accomplish various tasks effectively and autonomously. **Mapping Algorithms:** Include Simultaneous Localization And Mapping (SLAM i.e.: FastSLAM, GraphSLAM, and Extended Kalman Filter (EKF)) to allow modular robots to build maps of their environment and estimate their position within that map [425]. **Enhanced Path Planning Algorithms:** Determine optimal or feasible paths to navigate through modular robots' environment using algorithms such as A* (A-star), Dijkstra's algorithm, Rapidly-exploring Random Trees (RRT), and Probabilistic Roadmap (PRM) [431]. **Planning and Task Allocation Algorithms:** Are used for high-level planning and task allocation by determining which modules should perform specific tasks based on factors like module capabilities, task requirements, and system constraints [77]. Examples of such algorithms include but are not limited to include Task Allocation Graph (TAG) or Market-Based Task Allocation (MBTA)

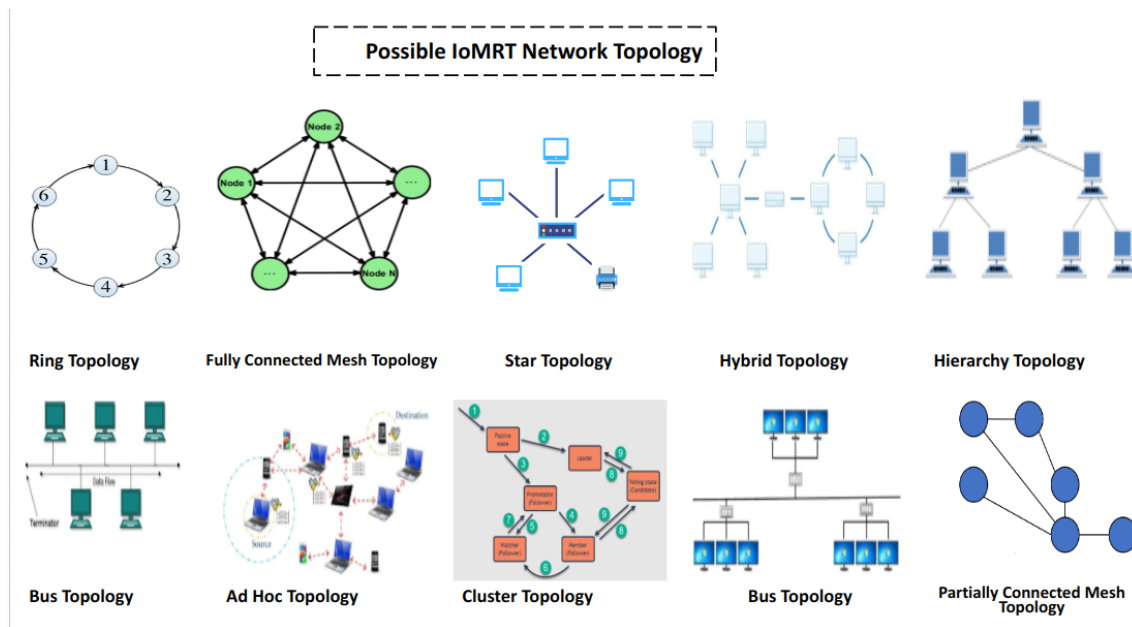


Figure 4.4: IoMRT Network Topologies.

algorithms.

4.4.3/ IOMRT NETWORKING TOPOLOGIES

The choice of IoT networking topologies for modular robots depends on several key factors such as the level of coordination required, scalability, fault tolerance, power constraints, and environmental conditions to determine the most appropriate topology for efficient and reliable communication among the modular robot's modules. Each topology has its advantages and trade-offs, and the selection should be based on the specific needs and characteristics of the modular robot system. This thesis presents the most common IoT networking topologies (see Figure 4.4) that are applied or can be applied to modular robots:

Star Topology: All the modules in the modular robot system communicate with a central hub or controller that acts as a gateway for data exchange and coordination, allowing centralized control and management [513]. The star topology in modular robotic systems is crucial as it facilitates centralized control and communication, enabling efficient coordination and management of individual modules while enhancing scalability and fault tolerance.

Mesh Topology: Involves direct communication between individual modules within the modular robot system, where each module can communicate with multiple neighboring

modules, enabling distributed communication, self-configuration, decentralized control, and fault tolerance [305]. The mesh topology in modular robotic systems is essential as it fosters decentralized communication and control, enabling robustness, adaptability, and fault tolerance through direct interconnectivity among modules without reliance on a central node.

Tree Topology: Includes the spanning tree concept, similar to a hierarchical structure, where modules are organized in a tree-like fashion with a root module at the top (central coordinator) and child modules branching out below. This facilitates efficient data dissemination and control [411]. The tree topology in modular robotic systems offers hierarchical communication and control, facilitating efficient data routing, scalability, and organization, while enabling streamlined coordination and management of complex tasks across the robot ensemble.

Hybrid Topology: Combines multiple networking topologies (i.e. combination of star and mesh topologies) to meet the specific needs of the modular robot system where modules communicate with a central hub while also establishing direct peer-to-peer communication with neighboring modules [183]. Thus, achieving a trade-off between centralized control and decentralized communication. The hybrid topology in modular robotic systems combines the advantages of different network structures, offering flexibility, fault tolerance, and adaptability to diverse operating environments, while enhancing robustness and resilience in task execution and communication.

Ad hoc Topology: Often adopted by modular robots operating in dynamic or unpredictable environments, where modules establish communication links on the fly without relying on pre-existing infrastructure [448]. The ad hoc topology in modular robotic systems enables dynamic and decentralized communication among modules without relying on fixed infrastructure, allowing for spontaneous collaboration and adaptability in rapidly changing environments, thereby enhancing flexibility and scalability. The ad hoc topology facilitates self-organization and self-configuration of modular robotic systems, enabling modules to establish communication links autonomously based on proximity and network conditions.

Ring Topology: Allows modules to be connected in a circular loop, where each module is connected to its adjacent modules, with data being transmitted in a sequential manner passing through each module until it reaches the intended recipient [404]. Despite it being an efficient communication structure, except that it is prone to a single point of failure. The ring topology in modular robotic systems fosters efficient communication and

data transmission by allowing each module to be directly connected to two neighboring modules, promoting fault tolerance and enabling seamless information flow circularly.

Bus Topology: Connects modules in a linear fashion, where each module is connected to a common communication bus, despite the data being received by all other modules, except that only the intended recipient can process it [257]. Despite it offering easy integration and communication between modules, except that it suffers from collision issues in case of simultaneous data transmissions. The bus topology in modular robotic systems facilitates centralized communication and control, enabling modules to connect to a shared communication channel, simplifying data exchange and coordination while ensuring scalability and flexibility in system design.

Cluster Topology: Groups modules into clusters or sub-networks based on their proximity or functional similarity, where each cluster has a leader module that manages communication within the cluster and serves as a gateway for inter-cluster communication [507]. Cluster topology in modular robotic systems enables the organization of modules into distinct groups, facilitating efficient communication and collaboration within clusters while allowing for decentralized control and scalability, thereby optimizing system performance and adaptability to varying tasks and environments.

Fully Connected Topology: Allows each module in the modular robot system to be directly connected to every other module, forming a complete communication network [459]. Despite offering direct and efficient communication, except that it can become complex and impractical in case of a higher module number. A Fully Connected topology in modular robotic systems ensures that each module is directly connected to every other module, facilitating robust communication, data sharing, and collaboration among modules, thereby maximizing system flexibility and fault tolerance while enabling complex collective behaviors and tasks.

Hierarchical Topology: Organizes modules in a multi-level hierarchy, where higher-level modules have control and coordination authority over lower-level modules, as communication flows vertically to ensure a scalable and structure communication [16]. The advantage of this topology is that it can be adopted when there's a need for different decision-making levels within the modular robot system. Hierarchical topology in modular robotic systems establishes a structured framework for organizing modules into multiple levels of interconnected subsystems, enabling efficient communication, task allocation, and coordination across different hierarchical levels, thereby enhancing system scalability, flexibility, and complexity management.

4.5/ SUGGESTION & RECOMMENDATION:

This thesis takes into consideration both IoRT and IoMRT's safety, privacy, and availability aspects from a security point-of-view regarding the modular robotic systems to ensure their safe and secure use ahead of their deployment in a real challenging and changing environment adopting complex scenarios. This concept is adopted to avoid ambiguity regarding the concept that safety and security mean the same thing, which is not true. We also adopted this form to start from the least important to the most important security points. Also, the main learned lessons are presented. The whole work is summarised in Figure 4.5.

4.5.1/ SECURITY ASPECT

The security aspect must be taken into consideration to present the right security measures and countermeasures to ensure a much more secure adoption and practices to protect the IoRT and IoMRT domains, aside from the consideration of the adoption of the security-by-design concept. This aspect is summarised in the following security suggestions and recommendations:

Ethical and Forensics Measures: Ethical measures such as the ones presented in [485] should also be adopted to mitigate the threat coming from internal and external attacks. Forensic measures are similar to the ones presented in [483] should also be taken into consideration to understand how an event took place to overcome further similar scenarios.

Security Measures: Should also be considered since the early design phase to avoid any attack or mitigate it by either protecting the users, wireless connection, communication [480], network [313], hardware (i.e. anti-theft and anti-tampering), software (i.e.

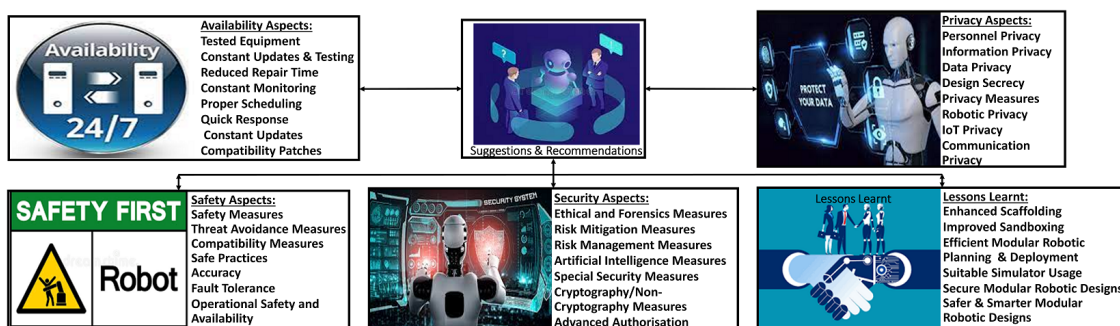


Figure 4.5: Suggestions, Recommendations and Lessons Learnt.

constant updates, patching, and coding verification) or any other equipment using cryptographic or non-cryptographic measures. Secure configuration can also be adopted to prevent any malicious modification or tampering and to secure configuration management. **Constant Risk Management Measures:** Should also be considered to identify, evaluate, assess, analyze, and mitigate both risks, threats, and vulnerabilities. **Advanced Authorisation:** Where users and operators are granted permanent or temporary access permissions depending on their Field-of-Knowledge (FoK) and Area-of-Operation (AoO) based on their assigned tasks, under full supervision. Both device and user biometric authentication methods can also be adopted to define access control and confirm the legitimacy of operators and devices in use. **Threat Identification and Monitoring:** Includes the deployment of monitoring and detection mechanisms to identify any potential security threats surrounding the modular robotics system.

Artificial Intelligence Measures: Must be further studied to ensure that modular robots are capable of "learning" in an artificially "intelligent" way to avoid any human interaction. This can be done by the adoption and integration of Machine Learning (ML) methods and techniques especially in cases of self-reconfiguration. Federated Learning (FL) might well be another method to train datasets to ensure more intelligent swarms [482]. This also covers ML-based cryptography/non-cryptography measures that should also be considered and taken into consideration even if the modular robotic system in the IoMRT domain is still new with limited or no knowledge about it including how to secure communications and monitor incoming/outgoing network traffic and packet exchange.

4.5.2/ SAFETY ASPECT

The safety aspect must also be addressed in hopes of reducing the likelihood of both risks and threats that may endanger human lives or the human way of life. Hence, several safety suggestions and recommendations are also presented.

Operator Training: Includes adhering to safety procedures, emergency protocols, and proper handling and management of modular robots. Moreover, an emergency stop mechanism can also be installed to halt operations instantly to prevent accidents and reduce the risk of hazards.

Compatibility Measures: Should also be adopted to be suitable to connect with a variety of modular robots or robotic systems along with their resource-constrained devices, along with the software/hardware in use to reduce the cost and power consumption.

Fault Tolerance: Modular robotic systems should maintain an ability to carry out their intended tasks (i.e self-assembly, self-healing, and self-reconstruction) uninterrupted and despite the occurrence of any possible failure in one or many of its modular robotic components.

Collision/Threat Avoidance Measures: Should also be considered especially in terms of collision/threat identification, threat source, type, management, and countermeasures.

Accuracy: Must also be taken into consideration to ensure that the accuracy of the performed and conducted tasks is high and less prone to errors especially false positives and false negatives.

Safety Measures: Should be adopted since design and during testing phases to avoid any event that may risk the injury or loss of human lives, or affect the whole industrial performance. This also includes: **Safe Practices:** Which take into consideration personnel safety by providing a safe working space, system safety by operating within a safe environment, and information safety by employing measures that ensure safe transmission of information and safe communication channels between modular robots. **Operational Safety & Availability:** Must also be maintained by deploying backup devices that can operate in case any modular robotic system is either affected or down. This will help ensure a safe operational mode and secure the operational availability of modular robots. **Safety Measures:** Includes safety signs and warnings around areas where modular robots are operating to avoid potential hazards around restricted areas. This also includes compliance with safety standards and regulations regarding the safe construction, design, and operation of modular robotic systems. **Regular Safety Checks:** And inspection to identify any potential safety risks that may affect the modular robot's behavior, as well as defining a reporting and investigation process for safety incidents to report any safety concerns and investigate incidents.

4.5.3/ PRIVACY ASPECT

Privacy should also be taken into consideration and not only the users' privacy but the whole business and work ethics (i.e. secret business trades, classified designs, etc) which play a key part in achieving a successful modular self-reconfigurable robot or robotic system. Hence, several privacy suggestions and recommendations are presented:

Design Secrecy: Modular robotic designs must also be protected and held privately especially when designing or working on sensitive topics mainly related to military, law

enforcement, police, etc.

Privacy Measures: Should be considered especially when dealing with modular robots with much advanced settings to prevent any leak of important information that risks any personnel or industrial exposure. This also includes: **Information Privacy:** Such as secret business deals and classified modular robotic information must be well-preserved and secure to avoid any possible leakage or stealing of information. **Data Privacy:** Especially when there's an established communication (short/close range or medium range communication) between modular robotic systems including swarms, which requires the transmitted data not only to be encrypted but also transmitted over a safe and secure communication channel.

Personnel Privacy: Including operators, designers, architects, and programmers must also be secretly held and well-preserved to risk leaking information about the users behind any working modular robotic project.

4.5.4/ AVAILABILITY ASPECT

Availability is one of the main key aspects that make modular robots and modular robotic systems operational, which in turn would also affect both accuracy and performance levels. Therefore, it is also important to take into consideration this key aspect before proceeding further and diving deeper into this domain. As a result, some of the key availability measures are presented as follows:

Equipment Testing and Repair Time. Algorithms for equipment testing and repair time estimation in modular robots are essential for diagnosing faults, determining the extent of damage, predicting repair durations, and scheduling maintenance activities efficiently to minimize downtime and ensure the continuous operation of robotic systems. **Tested Equipment:** Testing is recommended at different stages of the equipment's lifespan to ensure that they maintain the required level of functionality. **Reduced Repair Time:** In case of faults or malfunctioning, these equipment should have their repair time reduced, along with their inspection time, without causing modular robotic systems to go out of service. Hence, periodic maintenance can also be adopted to solve this issue.

Monitoring and Scheduling. Monitoring and scheduling algorithms for modular robots play a crucial role in overseeing system performance, collecting real-time data, detecting anomalies, optimizing task allocation, and dynamically adjusting schedules to ensure efficient resource utilization and timely completion of operations. **Constant Monitoring:** Is

also recommended to ensure that the modular robotic systems along with their equipment are functioning properly and under normal conditions, to avoid any obstacle that can delay or hinder their operational availability. **Proper Scheduling:** A proper timing scheduling should be advised to identify maintenance times, testing times, fixing times, and reaction times to any incident against any component or device connected to the modular robotic system.

Quick Response. In case of any emergency that affects the availability of modular robotic systems, specialist teams should react quickly to ensure a quick fix (i.e frequency and length of downtime) or to switch to secondary systems to maintain that operations will resume normally without interruption nor interference, especially in case of cyber-attacks against the modular robotic systems' availability.

4.5.5/ LESSONS LEARNT

Though the domain is still classed as "new", it is important to highlight the main lessons that are learned from previous IoRT-related robotic domains and which can also be added and adopted here as part of the present IoMRT and future IoSRT. Therefore, the main lessons learned and to be learned are classed as follows:

Efficient Planning and Deployment. Efficient planning and deployment strategies for modular robots involve advanced algorithms for task allocation, resource optimization, path planning, and deployment coordination, leveraging real-time data analytics, predictive modeling, and adaptive learning mechanisms to streamline operations, minimize idle time, optimize energy consumption, and enhance overall system performance and scalability in dynamic environments. **Enhanced Scaffolding:** To ensure that the targeted shape or structure is properly adopted and constructed with the ability to adopt self-repairing and self-correction capabilities. **Improved Sandboxing:** Should also be adopted to ensure that the running code is safely examined, observed, and analyzed to prevent any coding bugs or vulnerabilities, especially from codes that are untested or/and untrusted. **Efficient Modular Robotic Planning:** Should also be considered to ensure that the intended modular self-reconfigurable architecture is well-planned and well-defined, with no design flaws, and with less latency and resource consumption. **Efficient Modular Robotic Deployment:** Should also be considered following the adoption of a well-defined design to ensure its Readiness-to-Deploy (RtD) in the intended Operational Field (OF) and Areas of Operations (AoO).

Design Safety and Security. Design safety and security measures for modular robots encompass comprehensive risk assessments, robust hardware and software safeguards, encryption protocols, access controls, anomaly detection algorithms, and fail-safe mechanisms, aiming to mitigate vulnerabilities, prevent unauthorized access or tampering, ensure system integrity, and uphold user and environmental safety standards throughout the robot's lifecycle. **Secure Modular Robotic Designs:** Should be adopted especially in terms of channel communication, packet exchange, modular robotic systems, servers, and their inter/intra-communication. **Safer Modular Robotic Designs:** Designs must be done in a safe manner that ensures their adoption of a tamper-resistant property to maintain their operational functionality and availability without interruption.

Realistic Real-Time Simulation. Realistic real-time simulation is essential for modular robotics as it provides a virtual environment where the behavior and performance of modular robotic systems can be accurately modeled and tested in real-time, enabling researchers and engineers to evaluate algorithms, validate designs, and optimize performance before deployment in physical environments. **Smarter Modular Robotic Operations:** Should also be considered and adopted to reduce false positives and false negatives which may affect the accuracy and real-time performance of the modular robotic operation. Also, designs should start considering the adoption of self-healing (part of self-recovery), self-sustaining, self-reconfiguration, and self-replication processes that can help them achieve these tasks. **Suitable Simulator:** Simulators should also be invented and developed to help both researchers and operators within the modular robotic domain and IoT fields, such as the case of the VisibleSim [445]. Based on the learned lessons, it is important to present our future work.

4.5.6/ FUTURE WORK

Future work will shed more light and focus on the modular robotic domain and its integration into the IoT to form the new IoMRT concept. As such, future tasks will mainly include:

Lightweight Solutions. Lightweight solutions in modular robotics refer to the development and implementation of compact, resource-efficient algorithms, software, and hardware components tailored to modular robotic systems, aiming to minimize computational and memory requirements while maximizing performance, scalability, and energy efficiency, thereby enabling seamless integration, operation, and deployment of modular robots in resource-constrained environments and applications. **Lightweight Security Solutions:** Which aim to secure data transmissions either through lightweight crypto-

graphic algorithms such as encryption and/or message authentication algorithms, as well as maintaining entity authentication to avoid the already mentioned attacks by using a lightweight cryptographic protocol that can be based on hash or symmetric encryption algorithm. Moreover, a lightweight intrusion detection scheme is required and can be applied at different levels (depending on the constraints of computation/ transmission).

Lightweight Source and Channel Coding: A new efficient data compression/detection and or correction algorithm will be required for real-time application and to respond better to entity computation and resource constraints.

Suitable Solutions. Suitable solutions in the context of modular robotics entail the development and implementation of algorithms, software, and hardware components that effectively address specific challenges and requirements posed by modular robotic systems, ensuring optimal performance, scalability, reliability, and adaptability across various applications and environments. These solutions are tailored to meet the unique needs of modular robots, enabling them to achieve desired functionalities, tasks, and objectives efficiently and effectively. **Security Domain:** For the propagation of modular robotic systems that take into consideration three main IoT aspects which are privacy, safety, and security, and combine them into the security domain. **Energy Efficiency:** Future efforts should focus on improving modular robotic energy efficiency, which allows them to operate for a longer period. The focus can involve long-lasting or fast-charging batteries or the reliance on renewable energy sources such as solar panels to enable self-sustainability.

Obstacle Avoidance. Future solutions will focus on how to overcome obstacles in a series of scenarios in a simulated and realistic environment to ensure their deployment at a later stage in the real world. This concept will be based on two key interconnected ideas which are “**shape definition**“ based on obstacle detection, and “**self-reconfiguration**“ based on obstacle avoidance.

Human-Robot Interaction. Includes developing future intuitive and natural interfaces to improve the design of collaborative control strategies to ensure a safe and secure human-robot (modular) interaction.

Learning Capabilities. Learning capabilities in modular robotics encompass a wide range of techniques, including machine learning, reinforcement learning, and evolutionary algorithms, enabling robots to acquire knowledge, adapt to changing environments, improve performance, and autonomously optimize behavior through iterative experience-based learning processes, thereby enhancing their flexibility, adaptability, and efficiency across various tasks and applications. **Enhancement Testing:** To ensure the solutions’

operational success in terms of accuracy, performance, error margin (i.e. false negatives and false positives), power, and resource computation especially for constrained modular robotic devices in a constant and ongoing manner. **Improved Learning:** Future research will focus on how modular robots may adopt enhanced adaptive and learning capabilities to autonomously adapt to different environments and tasks using machine learning algorithms to improve their performance and accuracy over time.

IoMSRT. Future work will include further studying the integration of modular robots into/with the swarm concept and vice versa, to form the future of the robotic domain and its integration in the IoT as a whole. A near-future move that will surely see the introduction of the Internet of Modular Swarm Robotic Things (IoMSRT) concept, as well as the Internet of Programmable Matter Things (IoPMT) which will surely see not only a swarm of modular robots of the same type communicating with each other but also other different modular robotic types (see Figure 4.6).

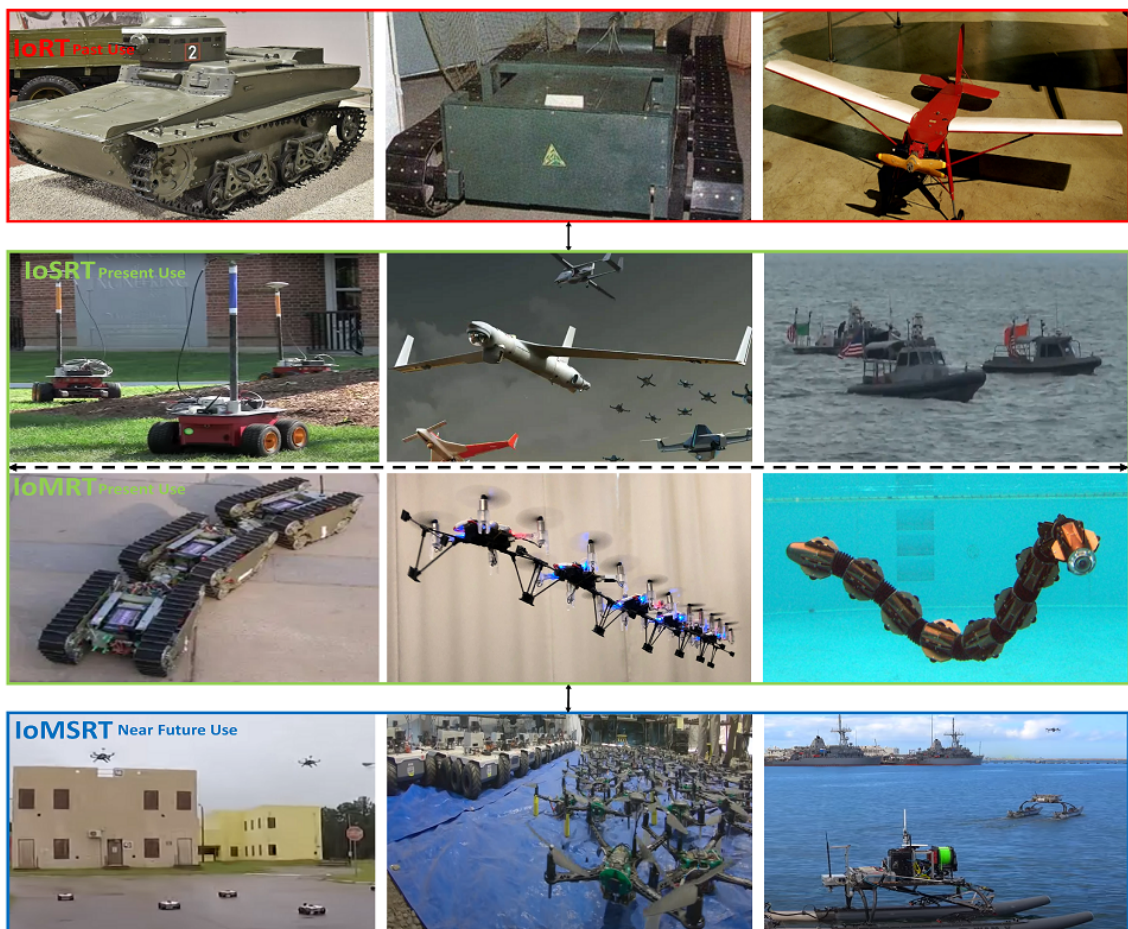


Figure 4.6: The evolving of the robotic domain within the IoT field including past, present, and near future use.

4.6/ CONCLUSION

As modular robots evolve into a swarm formation and start the adoption of the swarm concept, the robotic domain within the IoT field is being reshaped and re-innovated in a way that allows it to fulfill new gaps and accomplish new tasks. The modular concept, within the IoT, has since addressed new challenges and overcame the limitations of other non-modular robots with higher success rates, less time, and fewer resource requirements. As a result, this contribution has presented a survey that highlights in a detailed way the main characteristics, architecture, types, components, advantages, and drawbacks of MSRRs, in addition to the main IoT-related (i.e. IoMRT, IoSRT, and future IoMSRT) limitations and challenges that surround the modular reconfigurable and self-reconfigurable robotic domain including modular robotics and modular robotic systems. A comparison and analysis were presented to show the differences between both robots and modular robots, along with their relation to swarm robots. Moreover, the adoption of MSRRs in IoT domains including industry, medical, cyber-physical, military, law enforcement, and agricultural domains was also presented and discussed with many examples being highlighted.

Unlike previous theses, this thesis sheds light on the security and safety aspects by highlighting the main threats, risks, and attacks that lurk around, surround, and target this modular robotic domain. A much more detailed way was presented to offer an insight from a security background about the new perspective that takes into consideration the privacy, safety, availability, and security aspects upon the development of any security/safety measures and counter-measures that are to be taken into consideration to mitigate a threat and reduce risks to an acceptable level.

A framework was also proposed for the IoMRT taking into consideration the attack and defense-in-depth strategies after presenting the system mapping which is divided into four key parts (operator, gateway, communication link, and modular robot). Security attacks were also explained and discussed to highlight where the main vulnerability or security gap(s) could well be exploited, including the source and type of the attacks. Risks were also explained and detailed in a clear manner where the risk planning concept was introduced, in addition to the suitable security solutions types, as well as forensics and ethical hacking concepts being added and discussed to maintain a secure modular robotic environment which seemed to have lacked the security concept in its context. This thesis also presented key modular robotic solutions, which were mentioned, discussed, and analyzed, including the most recent ones, with many examples being added and highlighted. As modular robots evolve in a swarm-like formation and adopt the swarmanoid concept by becoming more and more AI-based with far lesser semi-supervised human intervention, they surely will evolve from the IoMRT to the IoMSRT. As a result, modular robot swarms will lead the future of the robotic domain within the IoT in both homogeneous and hetero-

geneous ways, not only to communicate with the same robot types, but rather with robots from different types, shapes, and robots operating on different terrains; even in some cases, being able to operate alone to adopt an advanced self-assembly/disassembly, collision avoidance or target acquisition strategy via "decision making".

CONTRIBUTION II - A NEW *Blinky Block* COMMUNICATION PROTOCOL

5.1/ INTRODUCTION

Programmable matter is made of small autonomous building blocks that can be programmed to achieve a wide range of geometric objects and structures with programmable capabilities to change their color or shape, which leads to the creation of programmable matter [54].

Most of the algorithms use the communication capabilities of robots to share local information to enlarge the global knowledge of the set. These communications are the weak point of distributed algorithms, as they represent the longest processing time. We will show that communication time is mainly due to the size of the data embedded in the messages. Even if the computational capabilities of the robots used in the programmable subject are quite small, the idea developed here is to use these computational capabilities to process the data received to reduce the size of the data transmitted.

The context used in this chapter is central to the problem of self-reconfiguration of programmable matter. Self-reconfiguration consists of programming modular robots so that modules move relatively to each other to change the overall shape of the assembly [373, 424, 33].

The preliminary step in any self-reconfiguration algorithm is to give the modules a way of knowing the final shape to be made. In [234], Tucci et al. presented a very efficient 3D scene encoding model for the self-reconfiguration process that describes a 3D model in the form of a Constructive Solid Geometry tree (CSG tree) combining the simple geometric objects placed in the leaves. Combination can be union, intersection, and difference of sub-trees. A string code may be generated from a depth search first traversing the tree.

Blinky Blocks are small cubic modular robots that make up the key component of the Claytronics project to create highly adaptable and reconfigurable objects and environ-

ments (cf. Figure 5.1). Each *Blinky Block* can be attached with its magnets to form complex geometric shapes, can exchange messages with directly connected neighbors, and react to noise by emitting sounds or/and changing colors. We use these real robots as a test bed to validate some parts of distributed algorithms for programmable matter. Even though they have no autonomous movement capability, their communication and computing capacity means that algorithms for programmable matter can be implemented on several hundred connected real robots.

The adoption of modular robots into the Internet of Things (IoT) [481] with the implementation of AI-empowered applications and services [482], can reshape the robotics concept [487, 479] into a new modular self-reconfigurable swarm, capable of operating in large numbers synchronously and simultaneously.

Communication between similar modular robotic systems as IoT components is essential to perform the intended task. However, this can be delayed due to the message size, which the length of the message can prove to be challenging and result in communication delays. For that, several solutions for Wireless Multimedia Sensor Networks (WMSN) were presented such as in [432] to reduce this redundancy by discarding a certain number of data packets while guaranteeing its integrity (quality). Other solutions include low-overhead data compression techniques [28], Compressed Sensing (CS) algorithms for data compression [73], and data compression and transmission scheme for power reduction in IoT-enabled wireless sensors [101].

However, they are prone to delays which can affect their ability to react in real-time which is often caused not only by the *Blinky Blocks* number but rather by the Message Length, which the higher the message, the higher the delay will become. As a result, several experimental results were tested on different compression/decompression algorithms to verify which one is more suitable to be applied to *Blinky Blocks* to mitigate the issue of delay and ensure a higher real-time reaction to users' orders and commands.

The following section presents preliminary work on the study of robots to evaluate their communication and computation capabilities. The next part proposes a study of classical

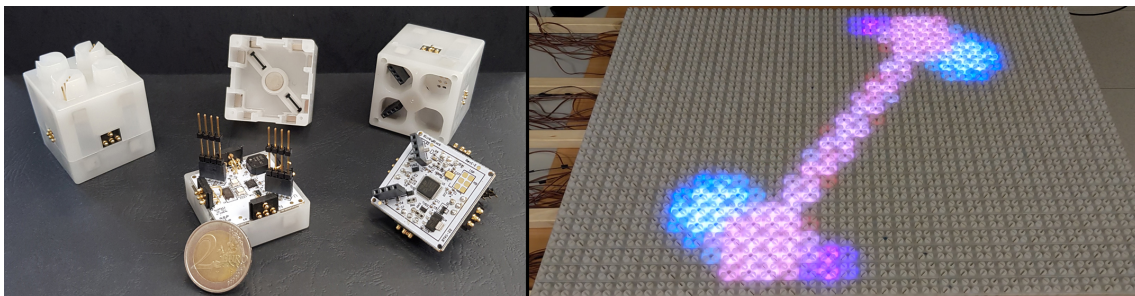


Figure 5.1: Left: BB Hardware. Right: a set of 768 BBs running the same program to visualise a cutting plane of 3D Objects.

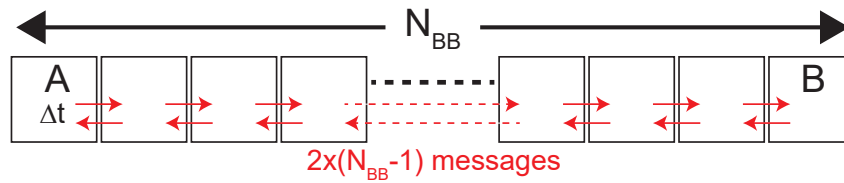


Figure 5.2: Experimental network diagram used to measure message propagation times.

compression models compared to Huffman’s method. Finally, our method is presented and completed by an experiment on a real problem applied to a large number of connected robots.

5.2/ Blinky Blocks BENCHMARK & COMPRESSION MODELS

A preliminary study of *Blinky Blocks* has enabled to assess their communication and pure computing capabilities. *Blinky Blocks* use very standard communication systems (6 UARTs, one on each side of the *Blinky Block*) and a processor very common in embedded systems (ARM Cortex M0 from STMicroelectronics, the STM32F091CB with 32 KB RAM and 128 KB flash memory), which allows us to generalize this study to most distributed multi-robot systems used in the context of programmable matter, such as the *3D Catom* [336].

First, to analyze the communication delay on *Blinky Blocks* in terms of the message length (M_l) and number of *Blinky Blocks* (N_{BB}), we place N_{BB} *Blinky Blocks* forming a simple line and we compute the communication time of several messages with different size of embedded data using the configuration presented Figure 5.2.

Measurement of the total time taken to transfer a message on all *Blinky Blocks* is carried out by a distributed program running on the robots. This program starts with the first extremity *A* sending a message to its only neighbour, at the local time t_0 stored in *A*. When the message reaches an internal module with two neighbours, the message received is sent back to the connected opposite port. When the message reaches the extremity *B* (which has only one neighbour connected), the message is sent back to the receiving port. When the back message reaches *A* at local time t_1 , the time $\Delta_t = t_1 - t_0$ gives the average duration of $2 \times (N_{BB} - 1)$ message transfers where N_{BB} varies from 4 to 52 respectively.

We repeat this operation 1000 times to deduce the average duration of the transmission of messages (T_{ML}). The benchmark tests were performed on a series of 52 *Blinky Blocks* with each set being tested for a message of N bytes, with N taking 7 values in [2..227]. Based on the obtained results (see Figure 5.3a and 5.3b), we found that only the message length affects on the communication time. Finally, we propose a linear approximation of

the duration of the message depending on its length:

$$t = 0.08935 \times M_l + 1.516 \quad (5.1)$$

In a second study, we carried out a number of calculations on each *Blinky Blocks* set, such as mathematical operations and decompression using Huffman's method. This study led us to the conclusion that all the computations required in distributed algorithms for programmable matter were negligible compared with communication time. Here, for example, decompressing a Huffman code of 1061 bytes takes 15 ms, which is comparable to sending 150 bytes from a *Blinky Blocks* to a neighbor.

Thirdly, we studied the various compression algorithms available and compared them with the Huffman method implemented on our *Blinky Blocks*. Data compression algorithms can be divided into two classes: Lossless Compression which allows the original data to be fully reconstructed from the compressed data and with no information loss, and Lossy Compression which is especially used for multimedia data such as images and audio. It allows the original data to be reconstructed with a certain loss of information, but it can achieve better data reduction compared to lossless compression as it allows more space to be freed up.

In our case, the type of compression is message (textual data) compression and consequently, the required compression time should also be lossless, due to its ability to prevent the loss of any data during the compression/decompression process to avoid any modification to the original sent message. On the other hand, Figure 5.4 represents a

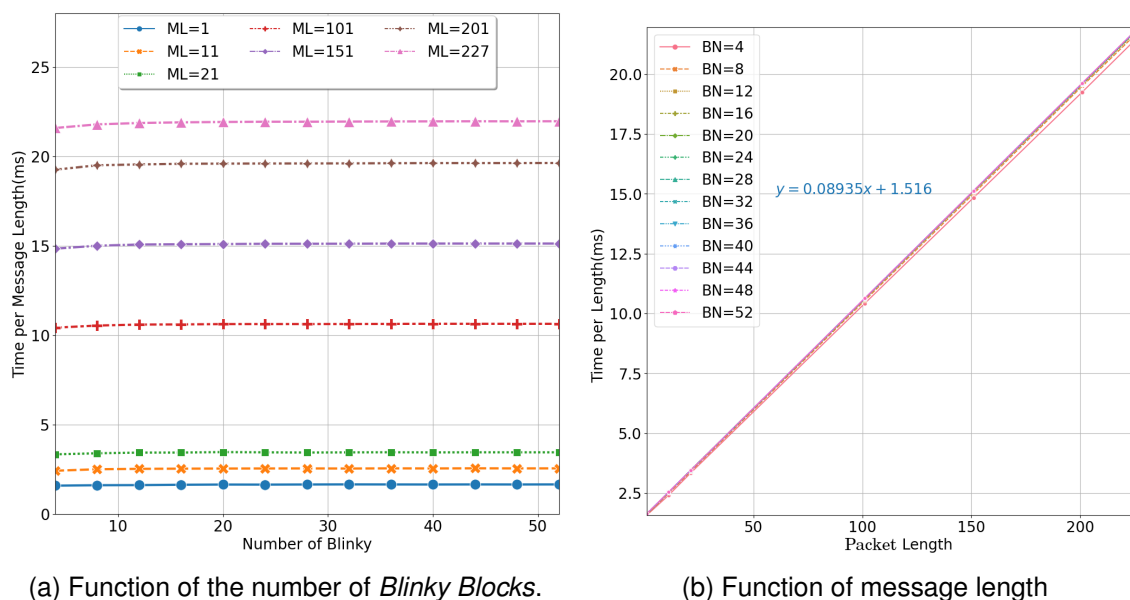


Figure 5.3: Variation of communication delay.

taxonomy of existing lossless data compression schemes. In this thesis, a lossless set of these compression schemes was tested to confirm whether they are suitable to be implemented with lattice-based modular robots or not.

A description of the most known and widely used lossless compression algorithms is presented in Figure 5.4. We tried different kinds of compression methods to give a brief description of each of the widely selected lossless data compression algorithms [12]:

- **DEFLATE**: is a lossless compression algorithm widely used in many popular compression utilities like gzip, zip, and PNG. It uses a combination of Huffman coding and LZ77 sliding window compression to compress text data [322].
- **LZ77**: is a lossless compression algorithm that uses a sliding window technique to compress textual data. It works by identifying repeated patterns in the input text and replaces them with references to previous occurrences of the same pattern [514].
- **LZW**: stands for Lempel-Ziv-Welch, is a dictionary-based lossless compression algorithm that is used in several popular file formats like GIF and TIFF. It works by building a dictionary of frequently occurring patterns in the input text and replaces them with shorter codes [108].
- **Brotli**: is a relatively new compression algorithm that was developed by Google. It uses a combination of a modern variant of the LZ77 algorithm, Huffman coding, and second-order context modeling to achieve higher compression ratios compared to other algorithms like DEFLATE [11].
- **Zstd**: stands short for Zstandard, and is a compression algorithm developed by Facebook. It uses a combination of Huffman coding, Finite State Entropy (FSE) compression, and a fast dictionary search algorithm to achieve high compression ratios and fast decompression speeds [84].

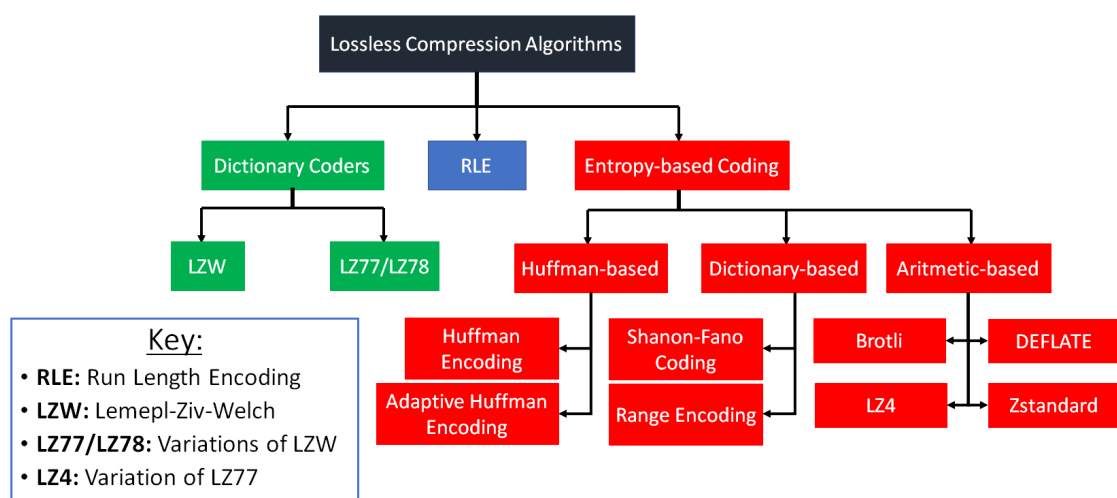


Figure 5.4: Taxonomy of Existing Lossless Compression Algorithm Types.

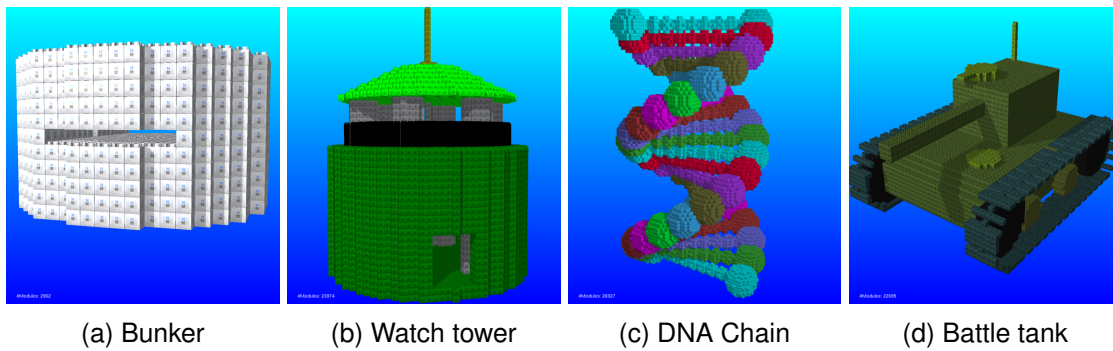


Figure 5.5: VisibleSim view of 3D models used for experiments.

To test the effectiveness of our proposed solution, we decided to create more or less complex shapes to get different message lengths. Therefore, we designed four 3D models on *OpenSCAD* [62] modeller: (a) a "Bunker", (b) a fortified "Watchtower", (c) an "ADN" and (d) a "Battle Tank" and integrated them on the VisibleSim simulator [446] to create a set of *Blinky Blocks* that fills the models as shown in Figure 5.5.

Our vectorial description language alphabet is made of 32 different characters, the description models are encoded into a list of 5-bit codes building the CSG tree. In fact, Table 5.1 shows the code size for each one of them. This code can be compressed before sending it in the graph of modules and is locally decoded inside each module (without storing the model) before being integrated into our simulator and applied using Huffman decomposition.

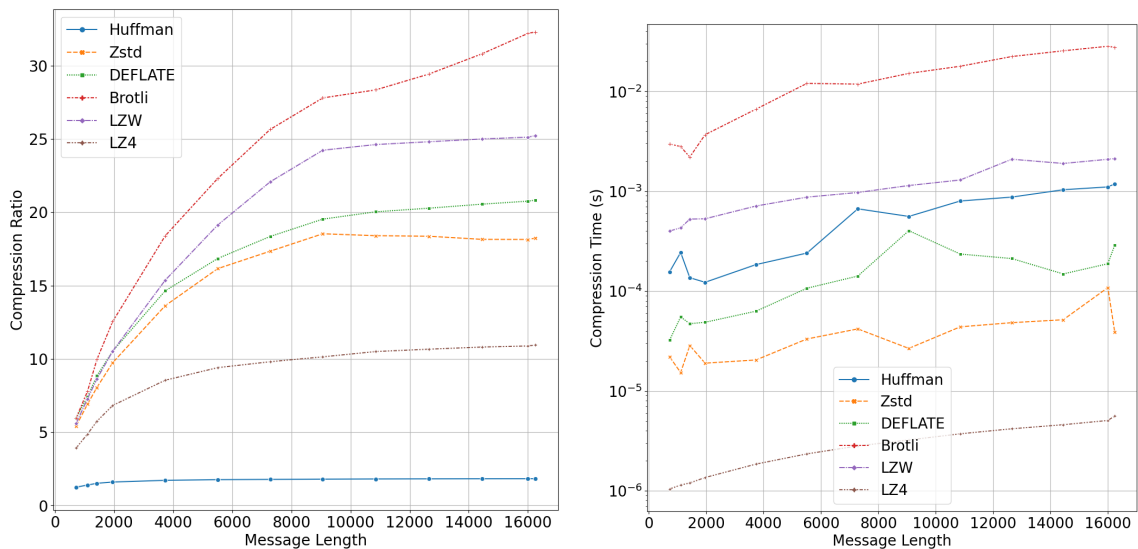
5.3/ METHOD AND EXPERIMENTS

To verify which compression algorithm is most suitable for both compression and decompression, a comparison was made in terms of the compression ratio of the data and the compression/decompression time.

The comparison was made between these lossless compression algorithms in terms of data compression ratio (see Figure 5.6a), data compression, and data decompression times (see Figure 5.6b). The testing was done on real messages that *Blinky Blocks*

Table 5.1: Size of the Designed Data Models.

3D model	Brut Size	5-Bit Coded	Huffman Header	Huffman Body
Bunker	116 Bytes	580 bits	139 bits	429 bits
Watchtower	397 Bytes	249 bits	167 bits	1422 bits
DNA chain	3722 Bytes	18610 bits	139 bits	12147 bits
Tank	3986 Bytes	19930 bits	188 bits	14432 bits



(a) Comparison of compression ratio versus message length. (b) Variation of the compression time versus message length.

Figure 5.6: Two efficiency comparisons for different lossless compression algorithms.

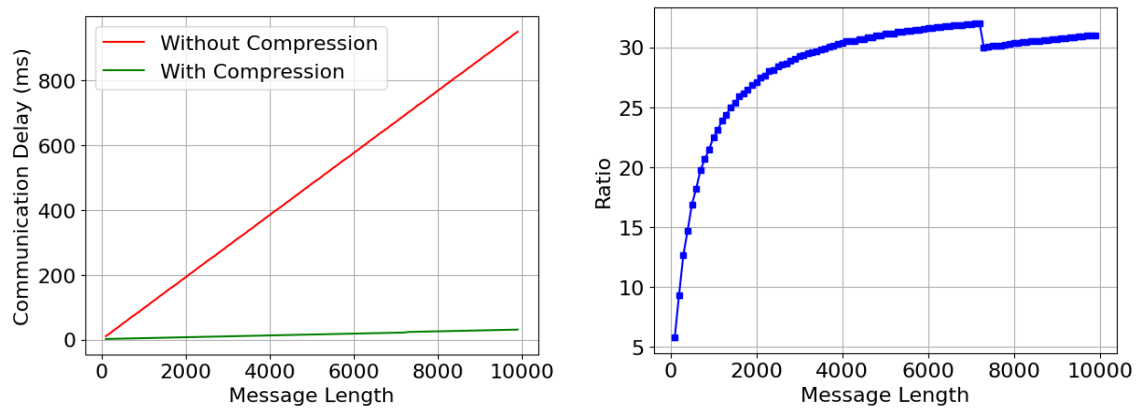


Figure 5.7: Variation of Communication Delay (a) and Time Ratio (b) in Terms of message length with/without Brotli Compression.

can use. Therefore, one can conclude that even though Brotli does not have the fastest compression and decompression times, except that it achieves the best result when it comes to message size compression by reducing the original message size as seen in Table 5.2, and data by 55%. Thus, it appears to be the best lossless compression algorithm for both data compression and decompression time, as seen in Figure 5.7, and it is a suitable candidate for implementation with *Blinky Blocks*.

To be more precise, the compression process was performed only once on the master side to prepare data to be flooded into the network, while the decompression process was done on each *Blinky Block* at each computation of its color (see Table 4.1).

Table 5.2: Numerical Example between different lossless compression algorithms using different message data sizes.

3D Model	Original Size (Bytes)	Huffman	Zstd	DEFLATE	Brotli	LZW	LZ4
Bunker	116	241	88	84	82	99	126
Tower	397	432	180	162	165	190	263
ADN Chain	3772	2252	288	266	229	253	443
Tank	3986	2577	569	491	444	507	1046

The data transmitted in our application is used to describe a 3D configuration (i.e. the shape to be occupied by the set of robots). These vector data are used to determine whether or not a grid position (occupied by a *Blinky Block*) is inside this shape. Compression is performed once by the external server, and then the compressed message (size N_{comp}) is sent to all robots via a module connected to the server. The decompression process, made in each *Blinky Block* in parallel, can be carried out by a single traversal of all received data. Then, data is 'decoded' on the fly, without storing a decompressed version of the message. This results in a complexity decomposition algorithm $O(N_{comp})$.

Despite Brotli having a high compression time, it also has the lowest decompression time. However, since we only need to compress the message once and decompress it every single time per *Blinky Block*, we found that Brotli seems to be the most ideal solution for this. Based on the obtained experimental results, we have shown how Brotli outperforms the other lossless compression algorithms in terms of compression ratio (as shown in Figure 5.6a).

As a result, one can deduce the effectiveness of the Brotli lossless compression algorithm in terms of both data compression and decompression and the reduction of message length. Thus, it proves to be a very effective method to mitigate the delay problem and effectively reduce its computation and execution time. Its appliance on *Blinky Blocks* comes as a novel solution for, to our knowledge, we are the first to propose applying lossless compression algorithms to a set of modular robots *Blinky Blocks* in terms of "Programmable Matter" and select the best one.

Regarding Figure 5.6a, the experimental validation of the given remarks was applied. On the left side, the communication delay of messages with different lengths is compared in both compressed (using Brotli) and original (non-compressed) versions. These graphs clearly show the gains made from the use of Brotli compression to transmit messages. The second graph (right), shows the link between the compressed and non-compressed message lengths for different message sizes. This experience confirms that the gain is very important whenever we have a higher message length.

Therefore, Brotli is a versatile compression algorithm that offers excellent compression ratios, especially for *Blinky Block* messages, while still maintaining reasonable compression and decompression speeds. To further confirm the accuracy of our presented work, we tested it on VisibleSim, which is a software tool for simulating and programming modular robots (BBs), and compared it with the already obtained results (see Figure 5.6a and Figure 5.6b) to show how close these results are and that the executed code remains the same wherever it is tested. Thus, it shows that the proposed algorithm has no compatibility or coding issues since it operates on the size of the message and not on the *Blinky Blocks* configuration.

After several tests on real data models and having it compared with other lossless compression algorithms in terms of compression/decompression time and compression ratio, Huffman will be replaced with the Brotli compression algorithm. Thus, offering the highest known compression ratio with far fewer compression and decompression times compared to Huffman.

Finally, we propose a more practical experiment to validate the complete process, consisting of compressing the 3D model, distributing the data code to a large set of connected *Blinky Blocks*, and decompressing many times the stored code in each *Blinky Block* to use the 3D data to set their color. Figure 5.8 shows a picture of the setup of this experiment, also used to produce the video¹.

The setup shown on the left side of Figure 5.8, includes a laptop connected to a grid of 768 *Blinky Blocks* (32×24). The laptop first sends the coordinates to each *Blinky Block* and then sends the compressed model to the *Blinky Blocks*. At launch, the *Blinky Blocks* create a common coordinate system to obtain a position (cx, cy, cz) relative to the module in the lower left corner, by applying the algorithm proposed in [348]. The spanning tree created for this purpose will be used to distribute the code to all the blocks.

In this application, we use a Huffman encoding algorithm which we ran on the laptop to create the code from the 3D model (the DNA model presented on the right side of Figure 5.8) and send it to one of the *Blinky Blocks*. To check that the data is well-received and uncompressed by each *Blinky Block*, after the reception we repeat 60 rounds that compute the color of a horizontal plane at level cz crossing the 3D scene.

At each stage, each *Blinky Block* analyses the encoding chain eight times to calculate the color at eight different positions of the space inside the block. This method allows the creation of anti-aliasing effects. Positions are $(cx \pm 0.25 \times l, cy \pm 0.25 \times l, cz \pm 0.25 \times l)$ where l is the width of the cubic *Blinky Block*. After half a second, each *Blinky Block* switches to

¹Video of Real-time decompression on *Blinky Blocks*: <https://youtu.be/xjAKxByAEII>

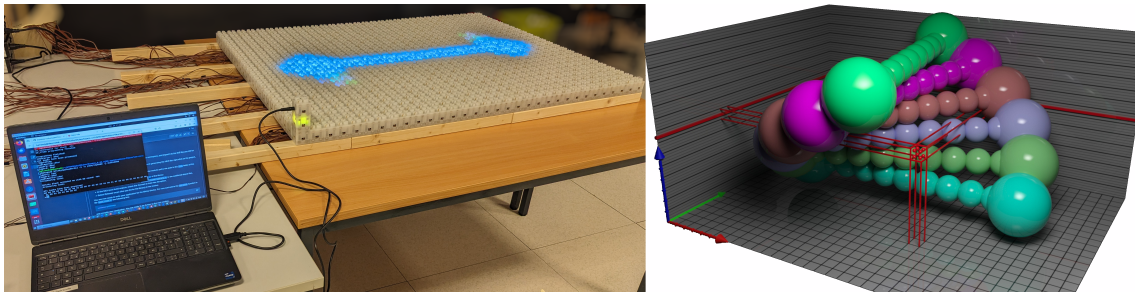


Figure 5.8: An example of *Blinky Blocks* application with transmission and decompression of a 3D description model (the short DNA chain presented in the right picture).

the next stage by increasing its c_z position by $0.25 \times l$ and recomputing a new color.

5.4/ CONCLUSION AND FUTURE WORKS

In this chapter, we propose a study of the efficiency of a set of *Blinky Blocks* robots in terms of communication delay and computation time.

Based on the obtained results, we show that the communication delay linearly depends on the size of the message, and presented Huffman as a lossless compression algorithm as a novel method, which was substituted by Brotli as an ideal solution.

To reduce the communication delay, we propose to add a lossless compression algorithm. We compare a set of recent efficient algorithms with the Huffman coding method. We express the compression ratio and compression/decompression execution time for each of them. Moreover, the obtained results show that the Brotli algorithm requires the minimum overhead in terms of execution time and can achieve the maximum compression ratio. Therefore, this work indicates that the Brotli algorithm should be introduced at *Blinky Blocks* to reach a minimum communication delay.

In the future, this work will further extend to cover three main points:

- First, the adoption of Huffman as the first compression mechanism that can perform compression on *Blinky Blocks* proved to be a success. However, it cannot compress large messages within the accepted range of *Blinky Blocks*' message length, which varies from 1 to 227 bytes. Therefore, based on the presented results above, Brotli will be introduced as a successor to replace Huffman's compression.
- The constant integration of *Blinky Blocks* into the IoT domain [488, 489] and its interaction with different IoT devices will surely require not only textual data to be exchanged, but also audio, video, and even images. Therefore, other compression algorithms will be tested, depending on the changing nature of *Blinky Blocks* and the structure of the integrated data to reduce the communication delays between *Blinky Blocks*.

- Compression is surely an important mechanism to reduce communication delays. However, it is important to ensure that this communication is not intercepted by a malicious/non-malicious party. Therefore, a very lightweight cryptographic solution that takes into consideration the resource-constrained nature of *Blinky Blocks* is required and will be integrated with the compression mechanism to ensure the first crypto-compression solution for *Blinky Blocks* that reduces delays and secures the communication by preventing the interception of the compressed messages.

CONTRIBUTION III - LCAPBB: LIGHTWEIGHT CRYPTOGRAPHIC ALGORITHMS AND PROTOCOLS FOR BLINKY BLOCKS AND PROGRAMMABLE MATTER

6.1/ INTRODUCTION

The remarkable switch from traditional fixed-design robots to modular robots in the field of IoT has surely boosted interest in programmable matter. This was achieved by mixing homogeneous nanorobots with distributed programming such as in the case of *Blinky Blocks* which are stationary, yet operate together within a modular robotic structure. A modular robot is composed of interchangeable and reconfigurable modules or components, including sensors and actuators, allowing it to self-configure into different shapes, and sizes and perform different functions and tasks (i.e. emitting sounds or changing color). Such modularity allows these robots, especially nanorobots to be flexible in design and adaptable to different tasks depending on the environment(s). Despite the advantages, nanorobots have no central control, are limited in terms of memory, and are resource-constrained. Therefore, applying traditional security protocols is challenging even if compatible with the IoT domain. As a result, lightweight security protocols and cryptographic solutions are required to achieve high real-time security while reducing message delay, network overhead, and memory usage.

6.1.1/ PROBLEM FORMULATION

Despite being the first security solution to be introduced to protect nanobots against hacking, PROLISEAN seems to suffer from several security weaknesses and flaws, which will be explained in Section 6.2. Such flaws are more specifically related to authentication and cryptography. This thesis will highlight them and then will propose LCAPBB as a mitigation solution using real-life datasets and experiments to prove it.

6.1.2/ RELATED WORK

Modular robots (i.e. nanorobots and *Blinky Blocks*) tend to offer several advantages that outweigh traditional robots [487, 479], especially in artificial intelligence [482], healthcare [488, 330] and industry [489]. This was presented with the introduction of modular robots, as part of programmable matter, into IoT, which led to the introduction of the Internet of Modular Robotic Things (IoMRT) concept, which can also be translated into the Internet of Programmable Matter of Things (IoPMoT) [481]. However, in terms of security, they tend to be vulnerable and susceptible to physical (i.e. tampering) and network attacks (i.e. eavesdropping, man-in-the-middle, packet injection, and packet interception). To mitigate this threat, in [197], Hourany et al. presented PROLISEAN with four versions, as the smallest footprint possible while providing a strong level of security against hacking. However, after a quick review, we realized that this work is vulnerable to various attacks and needs several key enhancements. As a result, we proposed LCAPBB. Our work comes as an enhanced version of the security solution already proposed by Hourany et al. to achieve greater security for programmable matter in general, and for nanobots and *Blinky Blocks* in specific [230].

6.1.3/ CONTRIBUTION

The main contribution of this thesis is to restudy the level of security of PROLISEAN, the previously presented security solution in [197], by presenting the main vulnerabilities and weaknesses in each of the four versions and proposing a mitigation solution for each version to achieve a higher level of confidentiality, integrity, and availability. In other terms, our work will include only one version in LCAPBB that summarises the other four versions, while ensuring that each scenario per version can be achieved, to ensure flexibility and security with less processing and resource consumption. Despite protecting nanobots against hacking, PROLISEAN seems to suffer from several security weaknesses and flaws related to authentication and cryptography, which will be explained in Section 6.2. This thesis will highlight them and then will propose LCAPBB as a mitigation solution using real-life datasets and experiments to prove it.

6.1.4/ ORGANISATION

This chapter is divided into eleven sections in addition to the introduction and is presented as follows: In section 6.2, *Blinky Blocks* attacks and countermeasures are presented, discussed, and analysed. In Section 6.3, the PROLISEAN Protocol is studied in each of its versions, where its weaknesses are presented while proposing suitable countermeasures per each version. In Section 6.4, a flexible and robust secure protocol is proposed while using the dynamic key approach, and evaluating the security requirements for each presented application type. In section 6.5, the proposed lightweight cryptographic algorithm is presented including the dynamic cryptographic primitives. In Section 6.6, the lightweight cipher schemes are presented including a substitution cipher, permutation cipher variant, and lightweight stream cipher, while a security analysis and a sensitivity test are conducted to verify the key sensitivity and evaluate the proposed update cryptographic primitives process and the randomness of generation algorithm key stream. In section 6.7, the experimental performance results are presented. In Section 6.8, we perform the cryptanalysis of the proposed cipher scheme to check and present its resistance against statistical, chosen/known plain-text/cipher-text, brute force, and more powerful attack types. In Section 6.9, a performance analysis is performed to study the effect of error propagation and computation delay. In Section 6.10, several key suggestions and recommendations are proposed and discussed. In Section 6.11, we conclude this work.

6.2/ *Blinky Blocks*: ATTACKS & COUNTERMEASURE

xx*Blinky Blocks* system is a modular distributed execution environment made up of centimeter-size cube blocks (i.e. roughly 40 mm) that are attached to each other via magnets (see Figure 6.2) and communicate through serial links on the block [231] using the neighbor-to-neighbor communication model (see Figure 6.1), with processing, storage, and communication (with up to 6 neighbors). This communication is based on a Transport Control Protocol (TCP) variant for time synchronization [53], which is the Modular Robot Time Protocol (MRTP), a network-wide time synchronization protocol for modular robots [300]. Only one block is needed to be connected to a power supply to power the whole model. This includes the latest version manufactured by Tech Power Electronics in collaboration with the FEMTO-ST research institute [53]. They can also be manually reconfigurable, allowing users to plug and unplug them during runtime to test their new changing behaviors since they are programmed in the same way [22]. However, despite their advantages, they are prone to a variety of attacks. Despite protecting nanobots against hacking, PROLISEAN seems to suffer from several security weaknesses and flaws related to authentication and cryptography, which will be explained in Section 6.2. This thesis will highlight them and then will propose LCAPBB as a mitigation solution

Table 6.1: Table of Notations

Notation	Definition
MK	Master Key
DK	Dynamic Key
SK	Secret Session Key
K_s	Substitution sub-key
n	Number of bytes in an input message
π	A dynamic produced permutation box
π^{-1}	The inverse corresponding permutation table
S	A dynamic produced substitution table
S^{-1}	The inverse corresponding substitution table
RK	A set of dynamic produced keystream
DK	Dynamic Key
Dk_1	The first 128 MSB set of DK and it is used to produce a dynamic Substitution table π .
Dk_2	The second 128 MSB set of DK and it is used to produce a dynamic Substitution table π .
Dk_3	The third 128 MSB set of DK and it is used to produce a dynamic update Substitution table S_{up}
Dk_4	The Fourth 128 MSB set of DK and it is used to produce a dynamic permutation table π_{up}
L	Number of rows of an image
C	Number of columns of an image
P	Number of planes (in gray-scale $P=1$)
len	Number of bytes in a message
h and w	Number of rows and columns in a sub-matrix, respectively
π	A dynamic produced permutation table
π^{-1}	The inverse corresponding permutation table
π_{up}	A dynamic update permutation table
S	A dynamic produced substitution table
S^{-1}	The inverse corresponding substitution table
S_{up}	A dynamic update substitution table
ns	Number of sub-matrices in one image
KS	The produced keystream obtained by using S and π
\oplus	XOR
\parallel	Concatenation
R	Pseudo-Random Sequence
x_i	Packet Sequence
r	Correlation Coefficient

using real-life datasets and experiments to prove it.

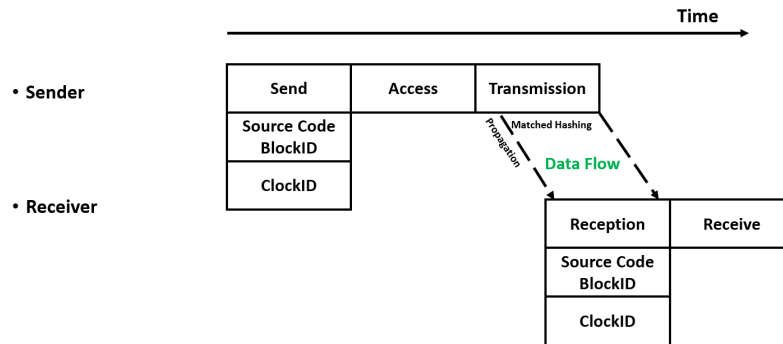


Figure 6.1: BBs Neighbour-to-Neighbour Authenticated Communication.

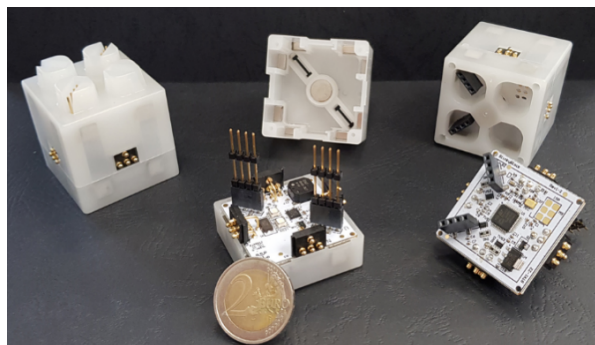


Figure 6.2: BBs Hardware.

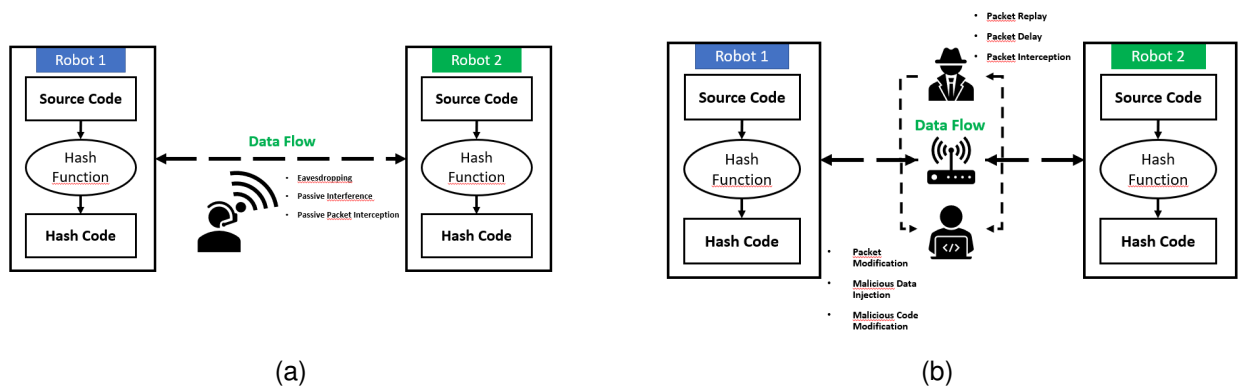


Figure 6.3: (a) Example of passive attacks and (b) active attacks against modular robots.

6.2.1/ BB POSSIBLE ATTACKS

As part of programmable matter, *Blinky Blocks* may be prone to several possible attack types (see Figure 6.3) such as:

- **Packet Delay:** Once the module receives the message from the Base Station (BS, which in this case presents the master *Blinky Block* that sends a broadcast request to identify its neighbors), it either does not reply or delays the reply to cause latency.
- **Packet Replay (Loop):** The packet does not acknowledge receiving a request, so

it urges the BS to keep sending requests that cause network congestion and traffic overhead, as well as emptying the BS pool of requests.

- **Rogue Connected Device:** Can either be exploited or a maliciously added module, which affects the operational performance and ongoing network activities.
- **Rogue Access Point (AP)/Base Station:** Rogue AP is placed to monitor incoming/outgoing network traffic (passive) or/and manipulate packets and data transmissions (passive).
- **Active/Passive Eavesdropping:** Listening to incoming and outgoing network traffic over one of its communication channels [313]. If they were encrypted, password cracking and counter-encryption methods would be applied.
- **Data Injection/Modification:** Once the communication link is breached, as part of an active eavesdropping attack, incoming and outgoing data can be manipulated.

6.2.2/ BB MITIGATION METHODS

Several mitigation methods can be suggested to mitigate these attacks such as:

- **Data Integrity With Source Authentication:** Can prove to be an ideal solution to protect the *Blinky Blocks*, especially in terms of security event(s).
- **Lightweight Encryption:** Can protect from passive attacks, including eavesdropping attacks.
- **Lightweight Message Authentication:** can protect the device from any integrity/authentication message attacks (modification/manipulation). A common operation is used to achieve lightweight cryptography due to the resource-constrained nature of the *Blinky Block* devices.
- **Lightweight Authentication Protocol:** Can protect the *Blinky Block* device from impersonation or unauthorized attacks.

6.3/ PROLISEAN PROTOCOL

PROLISEAN is a security protocol presented by Hourany et al. in [197] and stands for Protocol of Lightweight Security Embedded in an Architecture of Nanobots. According to the authors, PROLISEAN is designed to offer the smallest footprint while offering a strong security level against hacking and has four main versions. Each version is presented and briefly described in this section, while highlighting its vulnerabilities and how to mitigate them.

6.3.1/ FIRST VERSION: SIMPLE AUTHENTICATION

In this version, authentication achieves the first level of confidentiality, as each modular robot provides its source code to the other robot (i.e., nanobot) and compares it with its own. Since all nanobots have the same source code, the code comparison is efficiently achieved without the need to add unnecessary lines in their program (see Figure 6.4). However, this version is prone to integrity attacks, since the data flow remains prone to interception, leakage, or/and modification.

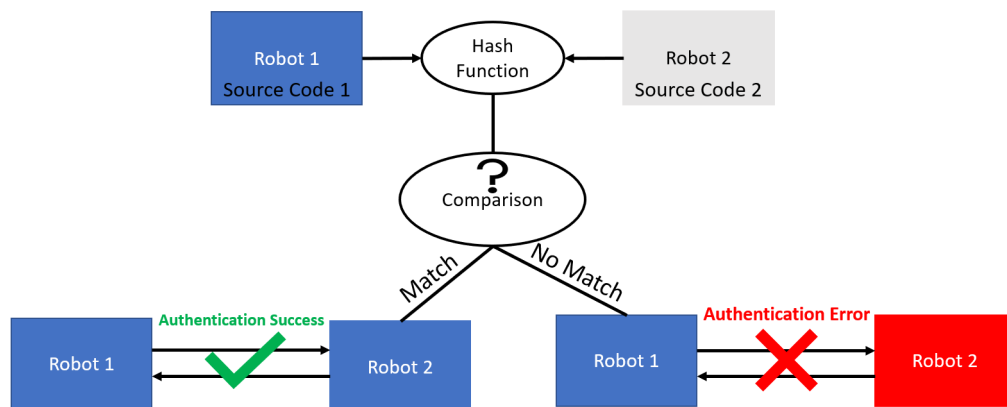


Figure 6.4: PROLISEAN Protocol: Version 1.

6.3.1.1/ VULNERABILITIES OF THE FIRST VERSION

The issue is that legal *Blinky Blocks* will transmit the correct hash code of the source code to illegal *Blinky Blocks* which can re-transmit it to legal *Blinky Blocks* and consequently can be considered as legal *Blinky Blocks* and communicate its received message. Therefore, the first authentication version is insecure and weak and does not consider this mechanism. In the following, we fix this issue by modifying the authentication mechanism.

6.3.2/ SECOND VERSION: AUTHENTICATION & CIPHERING

The second version includes two main phases:

1. **Phase I - Authentication:** The second version is the extension of the first version, where the first part of the source code is hashed and compared to check whether the authentication process is achieved or not (see Figure 6.5). The hash value is also said to be stored in the robot's memory due to its key role in phase II.
2. **Phase II - Encrypted Communications:** Upon the connection of nanobots, the

hash function is encrypted using the block symmetric cipher algorithm, where a common key is stored in each nanobot.

The aim of this version is not only to achieve confidentiality but also integrity, to prevent any leak of credentials during the authentication. Moreover, it also improves security by cutting the code into uneven parts, where the separation between the first part is used for authentication, while the second part is used for encryption with the key being kept secret by the user. However, it is not a very suitable solution for resource-constrained modular robots such as *Blinky Blocks* and nanobots.

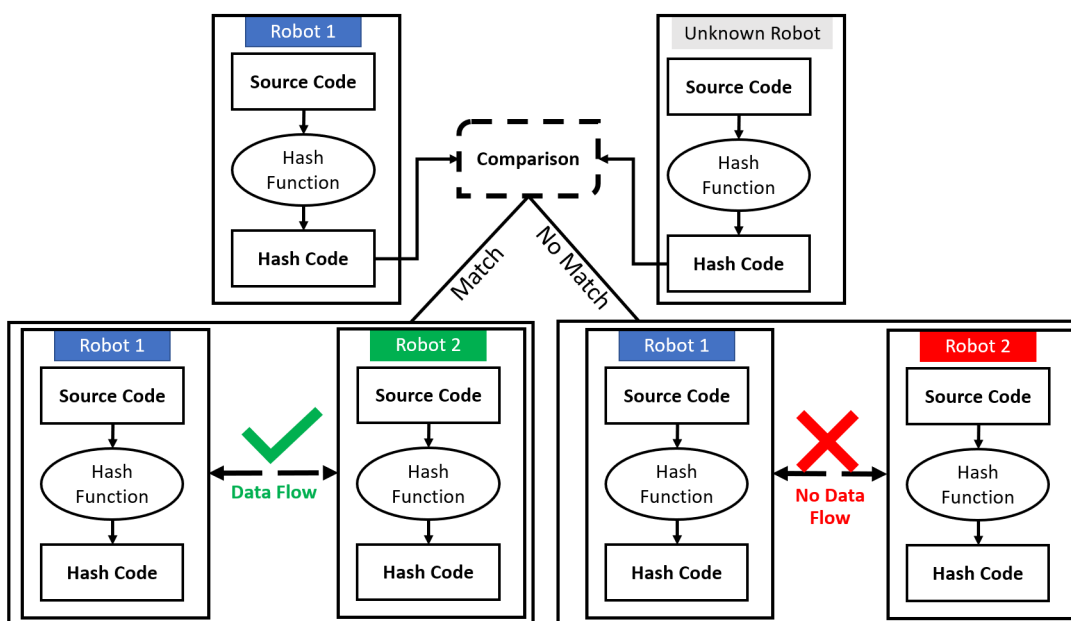


Figure 6.5: PROLISEAN Protocol: Version 2 - Phase I: Authentication.

6.3.2.1/ VULNERABILITIES OF THE SECOND VERSION

The same secret key is used for all communication and is independent of the *Blinky Block* entity ID, which means that the collected ciphertext is for only one key, which facilitates the task of the ciphertext-only cryptanalysis approach, and makes known plaintext/ciphertext cryptanalysis also possible. In addition to that, any *Blinky Block* can be comprised of physical and side-channel attacks, which can lead to the recovery of the secret key. Therefore, the weakness of this variant is that the secret key is fixed and should be dynamic to prevent the listed and other possible attacks, which helps to ensure the security of *Blinky Block* communication and functionality.

6.3.3/ THIRD VERSION: PROLISEAN PROTOCOL

The third version includes the PROLISEAN protocol and is based on two phases:

1. **Phase I - Refined Authentication Phase:** The code does not need to be hashed to provide high-security hashed blocks. Instead, only certain lines are required to be hashed (see Figure 6.6). Since all modules have the same seed, random numbers cannot be generated using a chaotic arithmetic sequence that defines which lines to hash.
2. **Phase II - Ciphred Communication with Multiple Keys:** The authors considered that the encryption key is unique per couple of nanobots, where they confirmed that the identity block was the same for both. Thus, a session key was used to encrypt data between them. Up to 4 or 6 session keys can be saved simultaneously before deleting them to create some memory space at the end of communication between them.

Although it offers a safer version, it is more resource-consuming. Thus, it is not suitable for resource-constrained *Blinky Blocks* or nanobots.

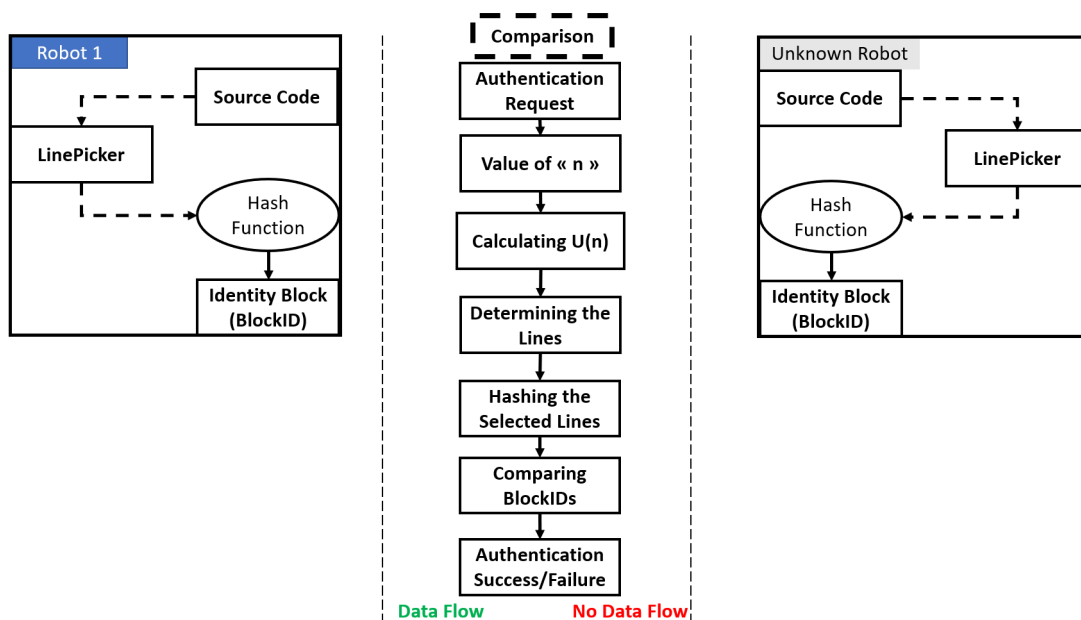


Figure 6.6: PROLISEAN Protocol: Version 3 - Phase II: Refined Authentication Phase.

6.3.3.1/ VULNERABILITIES OF THE THIRD VERSION

First of all, using a chaotic map requires floating computation with *Blinky Blocks*, in addition to converting it to integer representation. However, *Blinky Blocks* suffer from limited

memory in addition to having limited energy and requiring more delay, which can be a hard challenge, especially if they are used for real-time application. Here, it is a performance issue since an overhead in terms of delay and resources is introduced. It would be better if we could replace it with an invertible integer lightweight selection technique that can reduce the required memory, energy, and delay. In Table 6.2, we present several possibilities that can be used instead of the floating chaotic map and for a specific number of lines Nl .

Table 6.2: Invertible Polynomial functions.

$S = f(x)$	Conditions
$\text{mod}(ax + b, 2^w)$	a should be odd, and b can be odd or even
$\text{mod}(ax^2 + b \times x + c, 2^w)$	a should be even, while b should be odd, and c can be odd or even
$\text{mod}(a \times (x^3) + b \times x + c, 2^w)$	
$\text{mod}(a \times (x^4) + b \times x + c, 2^w)$	
$\text{mod}(a \times (x^5) + b \times x + c, 2^w)$	
$\text{mod}(a \times (x^6) + b \times x + c, 2^w)$	

Starting by initial vector X , where the value at index i in X is equal to i ($X[i] = i$, $i = 1, 2, \dots, Nl$) and w is equal $\lceil \log_2(w) \rceil$. To obtain the corresponding line, any selected polynomial function of Table 6.2 can be used to produce the NL pseudo-random selected lines. The difference between the different functions in Table 6.2 that increase the degree of the polynomial will introduce more integer multiplication operations but still lower to logistic map (2 degrees) that requires float multiplication compared to integer one. Therefore, *Blinky Blocks* can communicate secure parameters of the selected polynomial (e.g. a and b).

6.3.4/ FOURTH VERSION: IMPROVED PROLISEAN PROTOCOL

At the time of connection, both modules will be exchanging the ID given by the header, as well as their clock and a hashed part of their code, which will only be known by the second robot (see Figure 6.7). If both hashing results are the same, the modules will be flagged as authentic, and the next protocol phase, which is the encryption, is then triggered. The encryption is the same as presented in version three.

6.3.4.1/ VULNERABILITIES OF THE FOURTH VERSION

Aside from it being resource-constrained and unsuitable for resource-constrained modular robots such as *Blinky Blocks* and nanobots, this fourth version suffers in terms of encryption and authentication. As for possible attacks, this version is prone to the following list of threats:

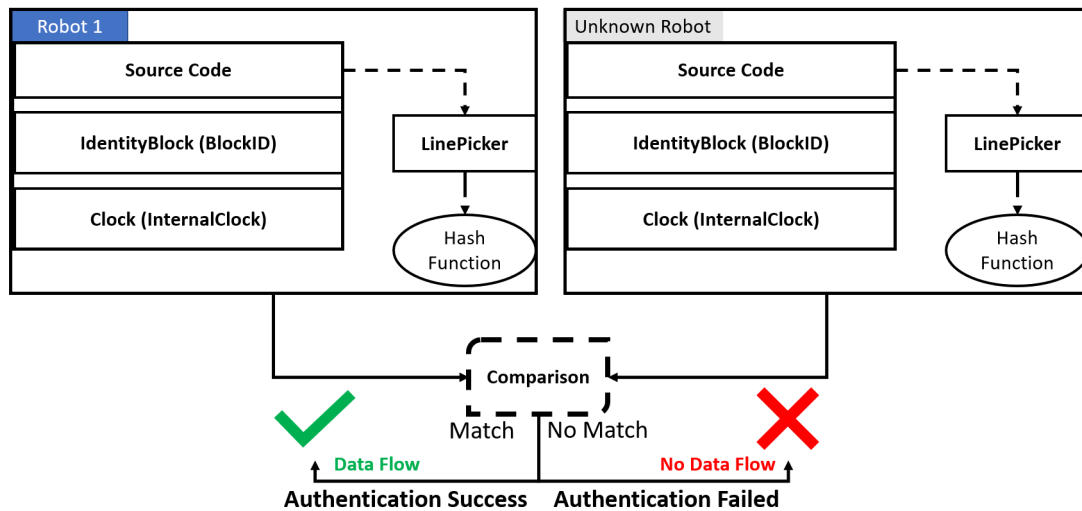


Figure 6.7: PROLISEAN Protocol: Version 4 - Authentication Phase.

- ID spoofing, which may lead to impersonation attacks.
- Hash collision, where an attacker can generate a different code that produces the same hash value.
- Shared key can be leaked since it is made up of the ID and clock values.
- No mutual authentication, since it does not ensure that the other module is also genuine.
- Lack of periodic key rotation of ID and clock values, which in the long-term may expose it to potential attacks.
- Code exposure, such as the exchanged hash parts of the code (e.g. Hash(CodeA)) and Hash(CodeB) provide information about the code's structure, which may lead to reverse engineering and identifying vulnerabilities.

6.4/ PROPOSED SECURE AND ROBUST PROTOCOL

6.4.0.1/ PROPOSITION FLEXIBLE & ROBUST SECURE TRANSMISSION PROTOCOL

The proposed authentication scheme is based on the concept that the hash code is the same for all legal *Blinky Blocks* as they have the same source code. Therefore, we propose to use it as a symmetric encryption key to encrypt the communicated message, which is the concatenation between ID, hash code, random number N_1 , and timestamp T_1 . Thus, the receiver can decrypt the received message, compare the hash code with the computed/stored one, and validate if the decrypted ID is the same as the requested *Blinky Block*. If both steps are validated, the request is authenticated for the receiver and now is the time for the receiver to authenticate for the requester. Similarly, the same message will be encrypted but the encryption key will mix the decrypted random

number and the hash code, in addition to producing a new random number N_2 and a new timestamp T_2 . This will prevent replay and man-in-the-middle attacks. The requested will decrypt this message as it has the hash code and N and validate if the same random number, the ID of the verified, and the hash code are correct.

After this, both entities will produce the same session key to establish secure communication between them. In this context, it is equal to the encryption of $N_1 \oplus N_2$ mixing with the hash code as input message and $N_1 \oplus N_2$ as secret key (see Table 6.1). Then, this key will be hashed and divided into two parts. If a message encryption and authentication algorithm are required, the first one is the session encryption key and the second one is the session message authentication key. Based on this and as N_1 and N_2 are random, different session keys will be produced that ensure the dynamicity property.

6.4.1/ DYNAMIC KEY APPROACH

This approach requires the use of the dynamic key and here we recommend the use of a pseudo-random line of source code to form it. Here, the session key is used as a seed with any lightweight stream cipher and it will be iterated for each time in a synchronized manner at both authenticated entities. The length of the required produced keystream is based on the selected degree of polynomial function that decides the required number of coefficients (at least two parameters (a , and b) for the first degree polynomial function, and 3 (a , b , and c) for a second-degree polynomial). This means that the produced keystream for each communicated message is used to form the coefficient of the selected polynomial degree.

Then, the selected polynomial function line will be iterated to produce NL unique indices that have values that vary between 1 and l , where l represents the number of lines in the source code. These lines will be hashed to produce the dynamic key that can be used for one or several input messages α (depending on configuration).

6.4.2/ SECURITY REQUIREMENTS VS APPLICATION TYPE

Moreover, we propose to divide the application case of *Blinky Blocks* into three cases:

1. **Public Application:** In this context, Message Confidentiality (MC) is not necessary, but to preserve the good functionality of *Blinky Block*, Message Authentication (MA) is necessary and can be ensured by using the recent lightweight MA approach presented in [314].
2. **Confidential Application:** In this context, message encryption and authentication

are required simultaneously. We recommend using the byte substitution/permutation process as an encryption scheme [312] with dynamic substitution/permutation tables, respectively. In addition, the proposed scheme of [314] can be employed as MAA to ensure source authentication and data integrity.

3. **Secret Application:** Here, robust message encryption and authentication are required simultaneously. The recent solution of [315] can be a good candidate for this task, as it was validated with fewer computations and resources compared to existing MAE schemes, in addition to the high level of security, as it is based on the dynamic approach. This approach requires a higher level of security compared to a confidential one, which means that a lower value of α and a high value of Nl is required.

Here, the assumption is that an illegal *Blinky Block* cannot duplicate a legal ID of another legal *Blinky Block*.

6.5/ PROPOSED LIGHTWEIGHT CRYPTOGRAPHIC ALGORITHMS

In this section, the dynamic cryptographic primitives are presented.

6.5.1/ DYNAMIC CRYPTOGRAPHIC PRIMITIVES

The dynamic key DK will be hashed by using a cryptographic hash function. This will allow us to reach a higher key sensitivity against any dynamic bit change(s). The choice of this secure hash function (SHA-512) is based on its possession of desirable cryptographic hash properties including a strong collision.

The produced hashed value has a length of 512 bits and is divided into four different sub-keys (bits each):

$$DK = \{DK_1, DK_2, DK_3, DK_4\}.$$

While, DK_1, DK_2 are required to generate cipher primitives. DK_3 and DK_4 are used to produce dynamic update cipher primitives (permutation table π_{up} and substitution table S_{up}) (ref. Table 6.1).

In this proposed approach, in case of any bit changes in the dynamic key, a different set of cipher primitives and updated cipher primitives is updated. Therefore, the cryptanalysis task becomes unfeasible.

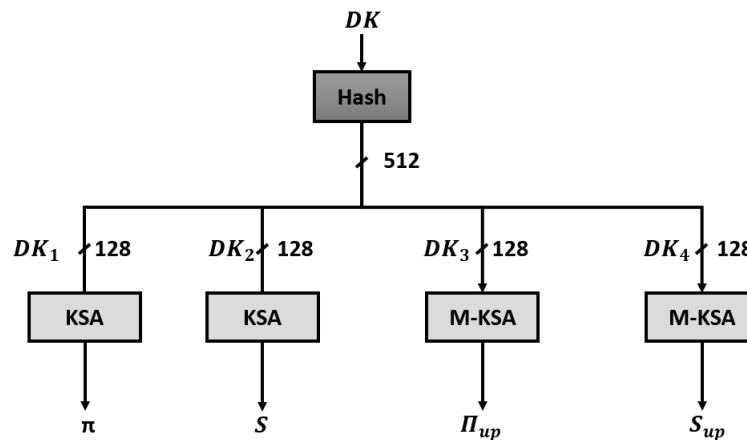


Figure 6.8: Proposed dynamic key generation steps DK , and the proposed techniques to construct the cryptographic and update cryptographic primitives.

6.6/ LIGHTWEIGHT CIPHER SCHEMES

In the following, several cipher schemes that require one round are presented and can be classified as **single operation or several operations**.

Examples of one-operation ciphers are substitution, permutation, or addition keystream (generating key stream requires only one operation) and they are described in the following.

6.6.1/ SUBSTITUTION CIPHER

In this step, the produced dynamic Substitution table S is used to substitute bytes of X (ref. Table 6.1)) as summarised in the following equation:

$$c_i = S[m_i] \text{ and } i = 1, 2, \dots, n \quad (6.1)$$

where c_i represents the ciphertext C , and m_i represents the plaintext M , respectively. In addition, c_i and m_i represents the i^{th} byte of encrypted and plaintext, respectively and n is the number of bytes per message.

On the other hand, the inverse byte substitution operation for each encrypted byte is done by using the inverse substitution table S^{-1} to recover the original message M as presented in the following equation.

$$m_i = S^{-1}[c_i] \quad (6.2)$$

On the other hand, after the encryption or decryption process, an update of the substitu-

tion table S is done by substituting its elements by using the update substitution table S_{up} (ref. Table 6.1).

6.6.2/ PERMUTATION CIPHER VARIANT

This cipher variant consists of permuting bytes of plaintext M by using a dynamic permutation table π (ref. Table 6.1)). This is expressed in the following equation:

$$C = M(\pi_i) \quad (6.3)$$

Where i represents the number of bytes and it varies between 1 and n . This implies that the ciphertext represents a permutation of the plaintext, guided by a dynamic permutation table. To recover the original message M , the inverse byte permutation operation is done by using the inverse permutation table π^{-1} . This process is expressed by the following equation:

$$M = C(\pi_i^{-1}) \quad (6.4)$$

The difference between permutation and substitution cipher variants is in the index of the table. In the case of substitution, bytes of plaintext are the index that select values of the substitution table as ciphertext. In the case of permutation cipher, the index is the element of the permutation table and is used to order the plaintext array according to the dynamic permutation table.

On the other hand, after the encryption or decryption process using this cipher variant, an update of the permutation table π is done by permuting its elements by using an update permutation table π_{up} (ref. Table 6.1)).

6.6.3/ LIGHTWEIGHT STREAM CIPHER: ADDITION KEYSTREAM CIPHER

Each byte of the message M (m_i , and $i = 1, 2, \dots, n$) is "exclusive or" (\oplus) with its corresponding generated byte keystream (i.e. $m_1 \oplus r_1, m_2 \oplus r_2, \dots, m_n \oplus r_n$). The contribution in this cipher variant is that the required keystream is produced in a lightweight manner and based on recursively substituting an initial random array and based on updating the substitution table. In addition, after each keystream generation iteration, the substitution is updated by using the update substitution table before swapping them. This is presented using the following algorithm:

The produced keystream KS benefits from the randomness and uniformity of the initial array (which contains all unique byte elements). In this proposition, the pseudo-random

Algorithm 1 The Proposed key-stream Generation Technique.

```

1: function KEY-STREAM-GENERATION( $S, S_{up}, n$ )
2:   for  $i \leftarrow 1$  to  $\lceil \frac{n}{256} \rceil$  do
3:      $S \leftarrow S_{up}[S]$ 
4:      $KS \leftarrow R||S$ 
5:      $swap(S, S_{UP})$ 
6:   return  $R = r_1, r_2, \dots, r_n$ 

```

substitution table is used as an initial table. Let us indicate that in this variant, the substitution and update substitution primitives are employed to produce the required keystream in contrast to the first proposition (single substitution cipher variant), which is used to substitute bytes of messages (ref. Table 6.1)).

The encryption is done by mixing keystream R with message M according to the following equation:

$$c_i = r_i \oplus m_i \text{ and } i = 1, 2, \dots, n \quad (6.5)$$

where r_i and m_i represents the i^{th} byte of the produced keystream R using the proposed lightweight keystream generation process as presented in Algorithm 1 and message, respectively.

Furthermore, to recover the original message M , the inverse byte keystream operation for each encrypted byte is done by producing the same keystream and mixing it with the received ciphertext as presented in the following equation:

$$m_i = r_i \oplus c_i \text{ and } i = 1, 2, \dots, n \quad (6.6)$$

Moreover, this scheme requires a simple lightweight keystream generation technique. In the following, an efficient and robust technique is presented and it is based on using the produced substitution S and update substitution S_{up} tables as presented in Algorithm 1. In this algorithm, the substitution table is substituted in a recursive manner by using the update substitution table to produce 256 bytes for each substitution time. The number of times of update substitution depends on the message length.

In the following, the next cipher variant is multi-operations that can achieve better security level compared to previous single cipher operation but it requires more computation and memory requirement and consequently energy consumption.

6.6.3.1/ MULTI-OPERATIONS CIPHER VARIANT

The proposed encryption scheme is summarised in Algorithm 2. It is mainly divided into two sub-functions (operations) that are Keystream – Generation, and RoundFunction

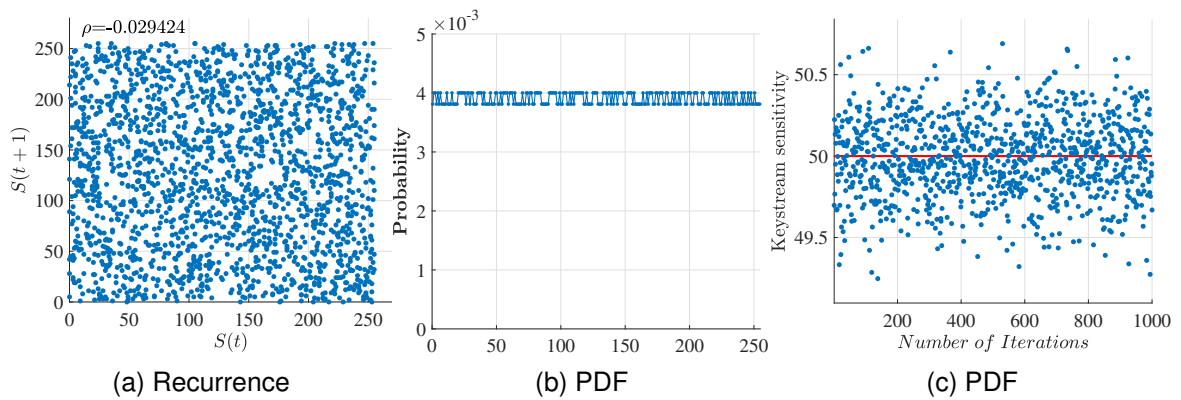


Figure 6.9: (a) Recurrence, (b) PDF and key sensitivity(c) of the produced key-stream for 1 000 random dynamic keys.

Algorithm 2 The proposed One Round Encryption Algorithm.

```

1: function ONE_ROUND_ENCRYPTION( $M, S, \pi$ )
2:    $KS \leftarrow$  Keystream – Generation( $S_1, S_{UP}$ )
3:    $temp \leftarrow M \oplus KS$ 
4:    $temp \leftarrow S[temp]$ 
5:    $C \leftarrow temp[\pi]$ 
6:   return  $C$ 

```

(f) that consist of all previous cipher variants (keystream mixing, substitution, and permutation, respectively). The RoundFunction consists of three operations, which are Keystream – mixingKey, ByteSubstitution, and Byte – Permutation. These operations were previously described and their corresponding inverse ones as they represent the previous single cipher variants:

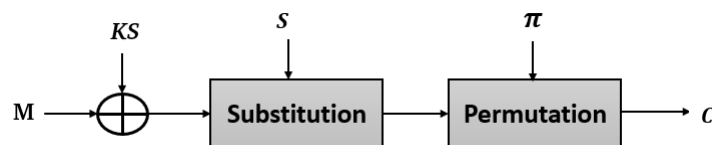


Figure 6.10: Block diagram illustrating the key steps of the proposed lightweight multi-operations cipher scheme.

6.6.3.2/ UPDATING CRYPTOGRAPHIC PRIMITIVE TECHNIQUE

The permutation tables π and π_{up} have the same length and also the substitution tables S have the same length of the update substitution table (S_{up}) (ref. Table 6.1). The proposed technique for updating the cryptographic permutation and substitution tables is described as follows in Algorithm 3. In this proposition, the permutation table π is updated by permuting its elements (updated) using the updated permutation table π_{up} , where the substitution table S is substituted (updated) by using the updated substitution table S_{up} . The

proposed update cryptographic process is designed to be very simple and to achieve the desired cryptographic properties. Furthermore, after each update process, a swap operation between permutation and update permutation table is done, in addition to swap between substitution and update substitution table. Swap operations are introduced to increase the periodicity and to reach the desired cryptographic properties.

Algorithm 3 The proposed Update Cipher Primitives Algorithm.

Input: Cipher primitives: Permutation table (π); Substitution table S ; Update permutation table (π_{up}); Update substitution table (S_{up})

Output: Updated permutation table π , S , π_{up} , S_{up}

```

1: function UPDATE_CIPHER_PRIMITIVES( $\pi$ ,  $S$ ,  $\pi_{up}$ ,  $S_{up}$ )
2:    $\pi \leftarrow \pi[\pi_{up}]$ 
3:    $S \leftarrow S_{up}[S]$ 
4:    $swap(S, S_{up})$ 
5:    $swap(\pi, \pi_{up})$ 
6:   Return  $\pi$ ,  $S$ ,  $\pi_{up}$ ,  $S_{up}$ 

```

6.6.4/ SECURITY ANALYSIS

To avoid statistical attacks effectively, a cipher scheme must satisfy both randomness and uniformity properties [317]. Therefore, several statistical tests were conducted to verify that the proposed scheme achieved the required properties. This is achieved by validating that the ciphertext reaches a random recurrence, a uniform distribution, and a low coefficient correlation between the original and the encrypted messages. These properties are analyzed as follows:

6.6.4.1/ UNIFORM DISTRIBUTION

To resist frequency attacks, ciphertexts must satisfy the uniformity property. This means that the frequency of all symbols in the encrypted message must remain very close to a uniform distribution. This means that each symbol has an occurrence probability close to $\frac{1}{n}$, with n representing the symbols' space being equal to 8 for byte messages. This can be visually and statistically justified. Visually, it can be proved by plotting the PDF of the encrypted message. The PDF of standard original messages and their corresponding encrypted ones are shown in Figure 6.13. These visual results indicate that the PDFs of encrypted messages follow the uniform distribution, where all symbols have an occurrence probability close to $\frac{1}{256} = 0.039$.

The original message distribution and its corresponding cipher one for a length of 1024 bytes are shown in Figure 6.13 a-d. This shows that the encrypted messages are spread

over the same space as the original one and have the same distribution in both cases.

Additionally, the entropy test at the block level (described in [311]), is used to validate this result. Therefore, the uniformity at the block level is satisfied if its corresponding entropy value is close to $\log_2(N)$, which according to [311] is the desired value. The numerical statistical results of the entropy tests are also presented in Table 6.4. This confirms that the uniformity property is achieved at the block level.

6.6.4.2/ RANDOM RECURRENCE

The recurrence plot measures the randomness level, which is reached by the obtained ciphertext, by estimating the correlations between the data sequences. By considering a sequence $X = x_0, x_1, \dots, x_{N-1}$, a vector with delay $t \geq 1$ can be constructed as $X_t = x_t, x_{t+1}, x_{t+2}, \dots, x_{N-t-1}$ (ref. Table 6.1)).

The variation between X_t and X_{t+1} for $t = 0$ for the produced cryptographic primitives (permutation or substitution) in Figure 6.11-a and for plaintext and ciphertext are shown in Figure 6.12 a-c. Based on this result, it is clear that the encryption process reduces the pattern.

6.6.4.3/ CORRELATION COEFFICIENT

Whereas, the correlation coefficient r_{xy} between two vectors x and y can be calculated using the following equation:

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x) \times D(y)}} \quad (6.7)$$

where:

$$cov(x, y) = \frac{1}{N} \times \sum_{i=1}^N (x_i - E(x))(y_i - E(y))$$

$$E(x) = \frac{1}{N} \times \sum_{i=1}^N x_i$$

$$D(x) = \frac{1}{N} \times \sum_{i=1}^N (x_i - E(x))^2$$

$E(x)$, and $D(x)$ represents the mean value and the mean deviation value of x sequence, respectively. In addition, $cov(x, y)$ represents the covariance operation between two sequences x and y .

The correlation coefficient between the original and encrypted messages versus 1 000 different keys is always close to 0. This shows that the correlation coefficient is always close to zero. This indicates that there is no detectable correlation between the original and its corresponding cipher image.

6.6.5/ SENSITIVITY TEST

Sensitivity tests are used to validate the message and key's avalanche effect. These tests are performed to quantify the percentages of differences between encrypted messages when one bit differs from the original message or from the secret (also dynamic) key, with the desired value being the difference of 50% at the bit level.

6.6.6/ KEY SENSITIVITY FOR THE DYNAMIC APPROACH

In response to a slight change in the keys K or IV , the sensitivity refers to a huge change in the cipher image, where the sensitivity of K and IV is analyzed for 1 000 random keys and IVs using the key sensitivity criteria. This criteria represents the mean of the SSIM metric tested before, and the percent Hamming distance used to show the positions' number where the corresponding pixels tend to differ between the plaintext and ciphertext images, and is calculated as follows:

$$\begin{aligned}
 KS_w &= \frac{\sum_{k=1}^{Tb} C_w \oplus C'_w}{Tb} \times 100 & (6.8) \\
 &= \frac{\sum_{k=1}^{Tb} E_{DK_w, IV}(P) \oplus E_{DK'_w, IV}(P)}{Tb} \times 100
 \end{aligned}$$

where C_w , C'_w are the corresponding cipher images using dynamic key DK_w and DK'_w respectively. All the elements of DK'_w are equal to those of DK_w , except for one element, which is the random Least Significant Bit (LSB). Indeed, the same processing is realized to measure the sensitivity of IV , and will give the same result, since K and IV are mixed to form the input of the key derivation function.

In Figure 6.14-b, the dynamic key sensitivity is shown versus 1 000 random keys, where only a random bit is changed in the dynamic key used DK_i . It is shown that the majority of samples are close to the optimal value in bit level ($KS_w = 50\%$), and behave as a normal distribution with standard deviation $std = 1.1113$.

Therefore, the proposed scheme is secure enough to overcome chosen/known plain-text attacks since the dynamic key approach is used.

In addition, Figure 6.14-b) shows the numerical statistical results for all cipher variants are close to the ideal one as a dynamic key approach is used. Based on these results, the difference between both encrypted messages is very close to the desired value. This indicates that the proposed cipher scheme satisfies the required key sensitivity level.

6.6.6.1/ MESSAGE SENSITIVITY TEST

The proposed cipher is based on the dynamic key-dependence approach, which means that the dynamic key changes for each input message. Moreover, the proposed cipher updates cipher primitives for each input block. Hence, the same message is encrypted under different dynamic keys, which leads to different cipher primitives for each block. Thus, different encrypted messages are obtained with a difference that is close to 50 % as seen in Figure 6.14-b. Therefore, the proposed cipher satisfies the message sensitivity (avalanche effect) by benefiting from the dynamic key approach. Finally, the proposed cipher achieves the required message and key sensitivity.

6.6.7/ EVALUATION OF THE PROPOSED UPDATE CRYPTOGRAPHIC PRIMITIVES PROCESS

Here, the performance of the proposed "update cryptographic primitive" technique is evaluated and analyzed to prove its secure deployment in the proposed cipher scheme. We first consider the cipher primitive of the permutation scheme before applying the proposed update technique on the produced permutation boxes (tables). A simulation test was conducted to test the recurrence and correlation (ρ) of updated permutation tables using 1 000 random dynamic keys, where the recurrence to produce the permutation box using a random dynamic sub-key DK_1 is shown in Figure 6.11a. Since the plot is widely scattered and randomly distributed, this shows that the primary or update permutation tables produced achieve the desired outcome (high degree of randomness). The correlation between multiple recurrence plots corresponding to the updated (renew) permutation tables for 1 000 iterations is shown in Figure 6.11b. The produced updated permutation tables achieve a high randomness degree. This is due to recurrence values being always close to the ideal value, 0.

Moreover, Figure 6.11c shows the correlation between the original permutation table and its updated one, where the value is always close to zero. This confirms that the updated permutation table is from the original primary permutation table. In Figure 6.11d, another correlated test was made and shown. It was conducted between two successive

(updated) permutation tables for 1 000 iterations, where its results show that it is independent of any two successive permutation tables.

Based on the obtained results, the proposed update technique tends to be highly and efficiently secure with no existing correlation in all of the presented cases. This offers a high degree of robustness against passive attacks and prevents any information from being leaked.

Table 6.3: Statistical results for 1 000 update permutation iterations

Coefficient Correlation Tests	Minimum	Mean	Maximum	Standard Deviation
ρ of the recurrence of produced dynamic permutation tables	-0.216	-0.003	0.235	0.065
ρ between the primary permutation table and its updated version (permuted version)	-0.175	0.001	0.218	0.064
ρ between two successive permutation tables	-0.181	0.001	0.263	0.06

The statistical analysis also confirms the high independence level between updated permutation tables, as seen in Table 6.3. Based on the obtained results, the standard deviation of the listed cases is very close to 0. This indicates that the correlation values are close to the desired mean value and that the primary permutation table and the updated ones are highly uncorrelated. This allows the permutation tables and proposed update substitution tables to attain a high degree of randomness and uniqueness to render them immune to cryptanalysis, overcome eavesdropping, and prevent unauthorized users from extracting any useful information from the encrypted color vectors.

6.6.8/ RANDOMNESS OF THE GENERATION ALGORITHM KEY-STREAM

The generated key-stream should be extremely random and uniform. Figure 6.9-a validates that the produced key-stream KS , is fully highly nonlinear, while Figure 6.9-b indicates that the PDF of the produced key-stream is very identical to the uniform one.

6.7/ EXPERIMENTAL PERFORMANCE RESULTS

The following experimental results are presented and shown in the following Table 6.4, and Figures 6.12- 6.13.

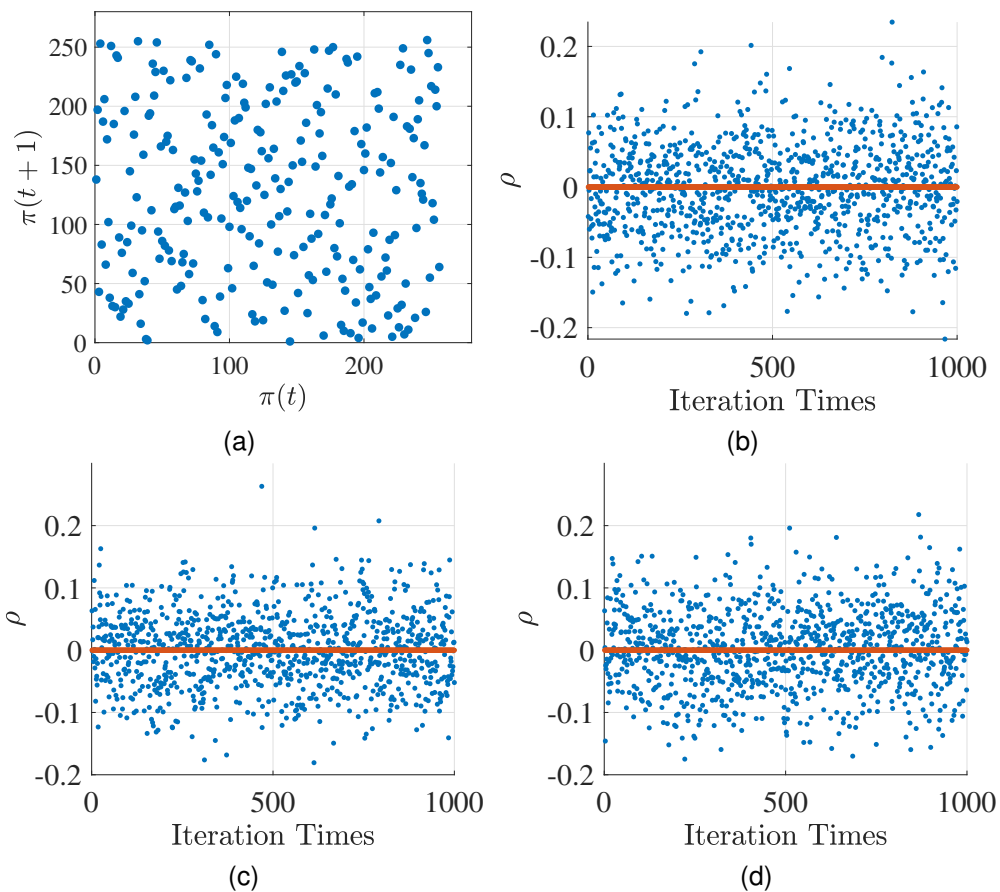


Figure 6.11: (a) The recurrence of a randomly-generated permutation table, (b) the correlation coefficient of the recurrence of 1 000 randomly-generated permutation tables, (c) the correlation coefficient between a randomly-generated permutation table and its updated version, and (d) the coefficient correlation between two subsequent permutation tables.

Table 6.4: Statistical Results

Statistical Results				
	Min	Mean	Max	Std
<i>Dif</i>	49.0112	49.9957	50.7629	0.2729
<i>KS</i>	49.0356	49.9902	51.0284	0.2850
<i>He</i>	7.9387	7.9547	7.9658	0.0040

6.8/ CRYPTANALYSIS OF THE PROPOSED CIPHER SCHEME

In this section, the proposed scheme is assessed and analyzed to see its robustness and ability to overcome multiple attack types including differential, statistical, brute-force, and chosen/known plain-text/cipher-text attacks. The proposed scheme is a public one, where an attacker can have complete knowledge about all the used cryptographic operations except for the information about the secret key, nonce, and the dynamic keys sequence. The choice of use of the dynamic-key scheme is to overcome the weak points that are

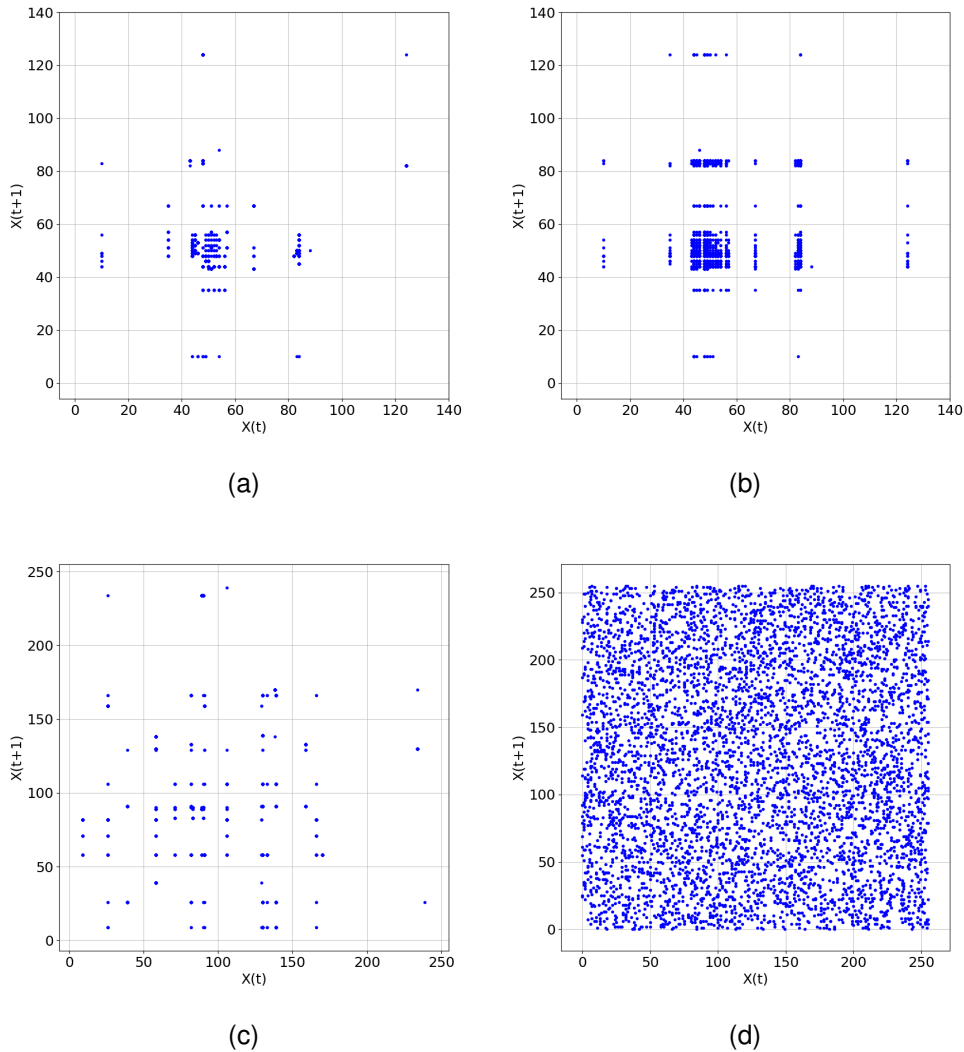


Figure 6.12: (a) The recurrence of a message, (b) permuted, (c) substituted, and mixed with keystream one (d) for a random session key.

found in the static key approach including accidental key disclosure, and/or single image failure. Next, we explain and prove how the proposed cipher variants can be immune and are capable of overcoming the already listed attack types.

6.8.1/ RESISTANCE AGAINST STATISTICAL ATTACKS (CIPHERTEXT ONLY ATTACKS)

The produced cipher-text must exhibit a high randomness, uniformity, and recurrence degree to resist all statistical attack types as dynamic cryptographic primitives are updated for each new input message.

Two additional tests were also conducted. These tests are the coefficient correlation

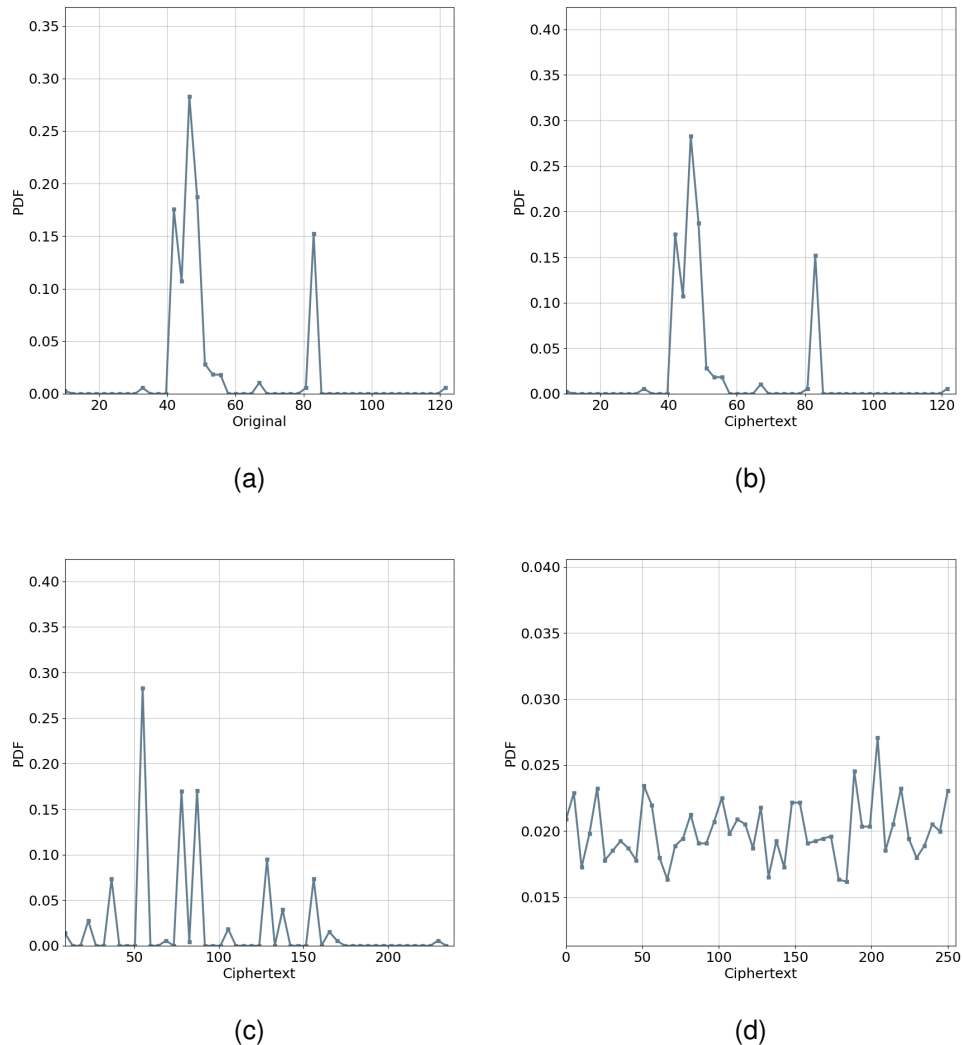


Figure 6.13: (a) The distribution of a message, (b) permuted, (c) substituted, and mixed with keystream one (d) for a random session key.

test and difference test. The difference test is the percentage difference. Both tests were conducted between the original plaintext and its corresponding encrypted version for 1 000 iterations.

It is proven in Figure 6.14-a, that both encrypted and unencrypted messages are completely independent of one another. Since the difference percentage is always close to 50%, the independence property is always validated.

Moreover, the statistical results of the percentage difference between plaintext and ciphertext are shown in Table 6.4. The mean value of the percentage difference is 50%, which is also the desired value. In all statistical tests, the standard deviation is too small. This means that results are always close to the mean value. A low standard deviation

value will surely lead to a small variation between the minimum and maximum values. Based on the conducted statistical tests, the obtained results tend to be always close to the ideal value. Thus, it confirms the robustness of the proposed cipher variants and its ability to overcome statistical attacks.

6.8.2/ RESISTANCE AGAINST CHOSEN/KNOWN PLAIN-TEXT/CIPHER-TEXT ATTACKS

The key sensitivity test managed to prove that a completely different ciphertext is obtained (50% variation at the bit level) whenever a secret key's single bit is changed or whenever the cryptographic primitives are updated after each new message. In our approach, the key sensitivity is achieved for each new input multimedia content. This is due to the dependence of cipher primitives and update cipher primitives, which constantly change, on the produced dynamic key. The key test sensitivity result is applied for 1 000 iterations. Any slight changes that affect the nonce or the secret key will result in a different dynamic, which would also lead to different cryptographic and update cryptographic primitives sets with a difference probability of at least 50%. In Table 6.4, the key sensitivity property is validated with an average that is always close to the ideal value, with a very low standard deviation. This shows that the values of key sensitivity tend to be close to 50%, with the plaintext sensitivity being achieved using the dynamic key-dependent structure. This approach is a variable cipher-primitive scheme with different cryptographic primitives (permutation and substitution tables in addition to keystream vector) being used to encrypt each input message. All to validate the ability of the proposed scheme to overcome chosen/known plain-text/cipher-text attacks and to prevent unauthorized users from obtaining valuable information. Thus, proving its robustness against both linear and differential attacks.

6.8.3/ RESISTANCE AGAINST KEY-RELATED ATTACKS

To achieve a higher resistance level against key-related attacks, cipher schemes must achieve the required sensitivity value of 50%, of which is the case of our proposed cipher scheme. This is primarily due to the use of variable cipher primitives which change per input image.

6.8.4/ WEAK KEYS

A set of dynamic keys is provided by the proposed dynamic key derivation approach, with a high degree of randomness. Each dynamic key is split into two dynamic sub-keys

sets, with both of them being used to produce the required cipher primitives and update cipher primitives, respectively. This shows that the used cryptographic and updated cryptographic primitives (permutation and substitution tables and keystream vector) are directly related to a dynamic key. Therefore, achieving the desirable cryptographic properties (message and key avalanche effect). In case of any existing weakness(es) in any of the input multimedia contents, both previously processed and to-be processed ones will not be affected. This is due to having cryptographic primitives being updated regularly via cryptographic primitives. As a result, ciphertexts are derived independently to protect them from any disclosure accidents or/and events. Thus, making the attacker's task more challenging. As a result, the maximum cryptographic strength is achieved, rendering our proposed cryptographic solution to be highly resistant against weak keys.

6.8.5/ RESISTANCE AGAINST BRUTE-FORCE ATTACKS

The key space of the session secret key can be 2^{128} , 2^{256} or 2^{512} . Moreover, the key space of nonce and the dynamic key are both 2^{512} , respectively. This makes them both large enough to overcome brute-force attacks. One should also note that the secret session key is padded with zeros since it has the same length as the nonce. A move that is crucial to execute the XOR operation.

6.8.6/ RESISTANCE AGAINST MORE POWERFUL ATTACKS

The dynamic key-dependent approach ensures a high level of resistance and robustness against powerful attacks. This is due to its employed dynamic cryptographic primitives. The choice of our approach is because all existing cryptanalysis techniques are built around the concept of a static secret key and static cipher primitives (the same cipher primitives are used for all communicated sessions).

Besides, using a dynamic cipher structure limits the ability of the attackers to break out the cipher. As a result, security is realized with a single simple process instead of using the AES scheme, with many iterations. As a result, a very low complexity scheme is achieved, while ensuring a high security level. Thus, preventing current cryptanalysis techniques from exploiting the proposed dynamic crypto-compression scheme. This would require new cryptanalysis techniques to break "dynamic cipher schemes". However, there are no schemes that are currently available of this sort, yet.

In this section, the discussion validates our proposed cipher scheme in terms of security and efficiency. The resiliency of our proposed scheme was proven especially against well-known attacks, such as statistical, brute force, chosen/known plaintext/ciphertext,

and linear and differential attacks [480]. It is extremely difficult for an attacker to know the produced cipher primitives or their updated versions, which are used for every message.

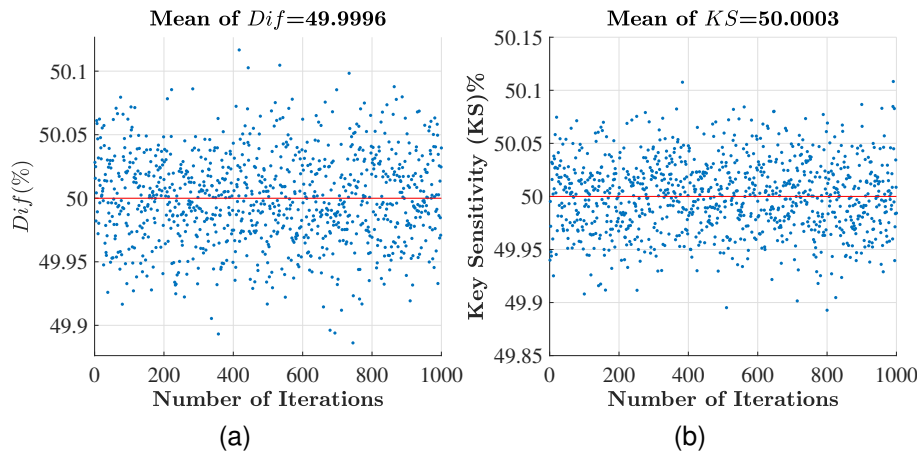


Figure 6.14: Difference between original and encrypted messages (a), key sensitivity (b) against 1 000 random dynamic keys for the proposed cipher with all operations.

6.9/ PERFORMANCE ANALYSIS

In this section, we analyze the performance of the proposed cipher scheme toward quantifying its effectiveness. Two important metrics are presented in detail, the effect of error propagation and the associated latency.

Here, we analyze our cipher's performance to quantify its effectiveness. Two important metrics are also presented and detailed. This includes the error propagation effect and the associated latency, respectively.

6.9.1/ EFFECT OF ERROR PROPAGATION

The proposed cipher can be considered as a stream cipher with dynamic cipher primitives. The effect of any bit error in the encrypted block c_i will only be constrained to the corresponding bytes in the decrypted blocks (m_i). Moreover, ECB limitations and the trade-off between the avalanche effect and local block error propagation are avoided in this scheme. This is done by using a dynamic key approach with different cipher primitives for each input block. Furthermore, identical blocks will be encrypted with different cipher primitives and different encrypted blocks will be produced. Moreover, the advantage of the proposed cipher is that it limits the error effect to a byte instead of the whole block as is the case of ECB, CBC, and CFB with the traditional block cipher [122], due to the required avalanche effect property.

The effect of errors at the bit level on the proposed cipher scheme is limited to its corresponding byte and not block(s), where errors are introduced with uniform random distri-

bution in the ciphertext. This means that the error impact of the proposed solution is low compared to traditional block ciphers with ECB, CBC, or CFB. These operation modes exhibit a higher error propagation rate (2% of random uniform errors are sufficient to destroy a message). The results confirm that the proposed cipher achieves lower error propagation, and consequently, efficient channel coding corrector algorithms can correct these errors.

In addition, when comparing recent dynamic key-dependent cipher solutions [131, 316], three blocks $\{m_{i-1}^{\hat{}}, \hat{m}_i, m_{i+1}^{\hat{}}\}$ are affected by the bit error in the decrypted message. Two of them $\{\hat{m}_i, m_{i+1}^{\hat{}}\}$ have random bit errors that occur independently in any bit position with an expected probability of $\frac{1}{2}$, whilst the third block $m_{i-1}^{\hat{}}$ has only one specific bit error in the same bit error position. However, for the proposed scheme, a bit error only introduces a byte error at the same corresponding byte position, which is also less (2 bytes for each bit error) compared to the presented solution of [311]. This indicates that the proposed cipher exhibits a lower error propagation compared to the recent dynamic key-dependent cipher schemes and existing cipher standards.

6.9.2/ COMPUTATIONAL DELAY

The main objective is to design an efficient cipher that reaches a higher security level with the minimum number of operations. Therefore, the objective is to reduce latency and resources/energy consumption, along with computational complexity. To assess the associated delay of the cipher, several delays are presented and quantified as follows:

1. T_S denotes the required execution time of byte substitution for a block of h bytes.
2. T_{xor} denotes the required "exclusive-or" execution time between two blocks of h bytes.
3. T_{π} represents the required execution time of permutation of byte for a block of h bytes.
4. T_{Sl} denotes the time required to permute the input blocks.

Therefore, the total Computational Delay (CD) of the proposed scheme to encrypt a mes-

sage M of n bytes is:

$$\begin{aligned} CD_{SubV} &= n \times T_S + 256 \times T_S & (6.9) \\ CD_{PermV} &= 2 \times n \times T_\pi \\ CD_{MixingV} &= n \times T_{xor} + 256 \times T_S \\ CD_{MultiOperV} &= n \times (T_{xor} + T_S + 2 \times T_\pi) + 256 \times T_S \end{aligned}$$

while the total computation delay of the standard AES in [93] to encrypt a message of n bytes is:

$$CD_{AES} = n \times (r \times T_S + (r + 1) \times T_{xor} + (r - 1) \times T_D + r \times T_{SR}) \quad (6.10)$$

where T_D represents the required delay for the AES Mix-column operations (for all 4 columns), which has a very high delay compared to other AES operations. T_{SR} represents the required delay for the AES "Shift-rows" operations and r represents the number of rounds. The minimum value of r is 10 for 128 bits secret key, and hence, the minimum AES computation delay is given by:

$$CD_{AES(r=10)} = n \times (10T_S + 11T_{xor} + 9T_D + 10T_{SR}) \quad (6.11)$$

Consequently, the AES computation delay seems to be larger once compared to the proposed one, especially since the proposed solution avoids diffusion operations (such as mix-columns of AES), which reduces the required delay. This proves that for a 128-bit secret key, our proposed scheme requires a lesser computational complexity when compared to the AES standard cipher. In fact, for AES 192 and 256-bit secret keys, r is respectively equal to 12 and 14. This shows that they require more execution time compared to a 128-bit secret key.

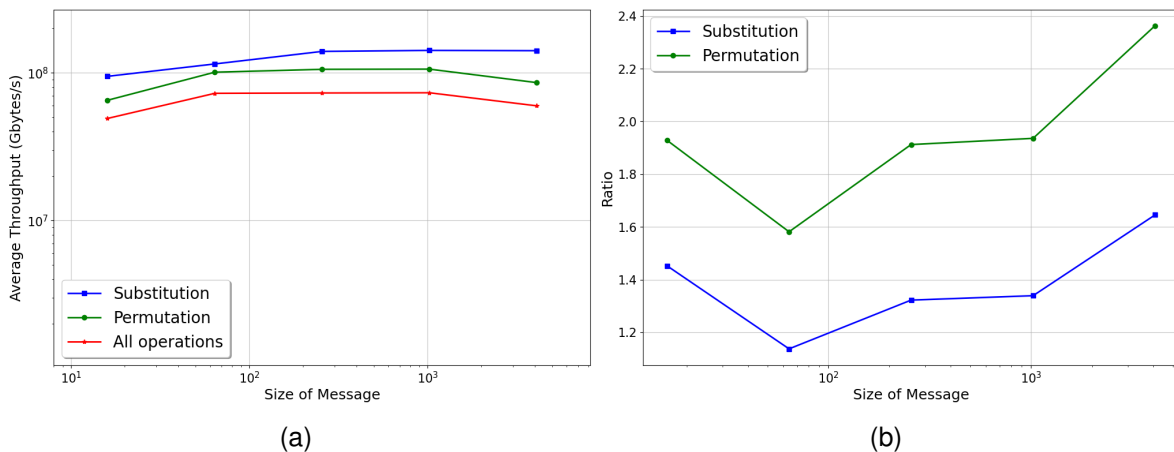


Figure 6.15: Throughput (a) of the proposed single operation variants and the corresponding ratio of the proposed cipher with all operations (b) and other single operation cipher variants.

Moreover, the computational delays of both the key derivation function and cipher primitives construction are described below. They are also quantified to assess the total associated delay as follows:

1. T_H denotes the required hash time for a block of h bytes.
2. T_{KSA} denotes the required RC4-KSA execution time.
3. $T_{MKSA}(x)$ denotes the required execution time of the modified KSA of RC4 for a table with h elements.

$$CD_{KDF} = T_H + 2 \times T_{KSA} + \times T_{MKSA}(n) \quad (6.12)$$

RC4 is a simple stream cipher that exhibits a low computational delay is used to construct the cipher primitives such as substitution/permutation tables. Moreover, round keys are generated in the function of the first substitution and permutation tables.

However, it still introduces a negative effect for small-sized messages. Hence, a different key derivation function was adopted for low-data rate applications. As a result, we updated the dynamic key and primary cipher primitives after δ small-sized messages, which also depends on τ threshold data bytes. Moreover, all cipher primitives are renewed, except for the second substitution table (S_2) which is updated after each encrypted message, While S_1 is being modified (permuted) for each input block.

Therefore, our solution is independent of the message size but is dependent on the configurable threshold data length. Therefore, decreasing τ leads to an increase in the security level and the required delay and resources and vice-versa. Finally, δ can be configured according to the required security level and the application context.

6.10/ SUGGESTIONS & RECOMMENDATIONS

Several suggestions and recommendations can be made to maintain a more secure *Blinky Block* environment including:

- **Very Lightweight Message Authentication Algorithm (VLMAA):** Can provide the integrity of data and the authentication source that relies on a dynamic key structure that is based on a single round and simple operations, while maintaining a high-security level.
- **Using a Single Round:** A single round and simple operations are used to achieve permutation, substitution, and "exclusive OR" operations.
- **Reduced Resource Consumption:** As the adoption of this solution would reduce the computational complexity, and resource overhead to achieve a simpler and less

complex implementation.

- **High-Security Level:** Where the uniformity, sensitivity, and randomness are achieved by using lightweight cryptographic algorithms and protocols.

6.11/ CONCLUSION

In conclusion, the convergence of nanorobots within the framework of modular robots in the context of IoT marks a transformative leap in modular technology due to its real-time automation, adaptability, and reconfigurability. However, they were still threatened by a variety of security vulnerabilities and attacks. Despite being the first security solution to be introduced to protect nanobots against hacking, PROLISEAN seems to suffer from several security weaknesses and flaws. As a result, in light of this paradigm shift, this thesis introduced LCAPBB which addresses the inherent limitations of its predecessor, PROLISEAN, by capitalizing on cryptographic methodologies and offering a higher level of security in terms of authentication and cryptography. In conclusion, the integration of nanorobots as components of modular robots within the Internet of Modular Robotic Things (IoMRT) domain introduces cutting-edge technologies that revolutionize various fields such as military, law enforcement, manufacturing, disaster response, industrial, and healthcare, leveraging their adaptable and reconfigurable nature to enable versatile applications in complex environments, fostering real-time data-driven insights and automation. Hence, this thesis proposes LCAPBB, a new Lightweight Cryptography and Authentication Protocol for *Blinky Blocks* (LCAPBB), as an enhancement of PROLISEAN within the realm of Lightweight Cryptographic Algorithms and Protocols for Programmable Matter (LCAPPM), addressing the main flaws and offering proper enhancements through cryptographic approaches (see Table 4.1).

IV

CONCLUSION

GENERAL CONCLUSION

As modular robots evolve into a swarm formation and start the adoption of the swarm concept, the robotic domain within the IoT field is being reshaped and re-innovated in a way that allows it to fulfill new gaps and accomplish new tasks. The modular concept, within the IoT, has since addressed new challenges and overcame the limitations of other non-modular robots with higher success rates, less time, and fewer resource requirements. Moreover, as modular robots evolve in a swarm-like formation and adopt the swarmanoid concept by becoming more and more (AI) Artificial Intelligence-based with far lesser semi-supervised human intervention, they surely will evolve from the IoMRT to the IoMSRT. As a result, modular robot swarms will lead the future of the robotic domain within the IoT in both homogeneous and heterogeneous ways, not only to communicate with the same robot types, but rather with robots from different types, shapes, and robots operating on different terrains; even in some cases, being able to operate alone to adopt an advanced self-assembly/disassembly, collision avoidance or target acquisition strategy via "decision making". As a result, this thesis has presented a survey that highlights in a detailed way the main characteristics, architecture, types, components, advantages, and drawbacks of MSRRs, in addition to the main IoT-related (i.e IoMRT, IoSRT, and future IoMSRT) limitations and challenges that surround the modular reconfigurable and self-reconfigurable robotic domain including modular robotics and modular robotic systems. A comparison and analysis were presented to show the differences between both robots and modular robots, along with their relation to swarm robots. Moreover, the adoption of MSRRs in IoT domains including industry, medical, cyber-physical, military, law enforcement, and agricultural domains was also presented and discussed with many examples being highlighted.

Unlike previous surveys, this thesis sheds light on the security and safety aspects, as well as performance by highlighting the main threats, risks, and attacks that lurk around, surround, and target this modular self-reconfigurable robotic domain. A much more detailed way was presented to offer an insight from a security background about the new perspective that takes into consideration the privacy, safety, availability, and security aspects upon the development of any security/safety measures and counter-measures that

are to be taken into consideration to mitigate a threat and reduce risks to an acceptable level. A framework was also proposed for the IoMRT taking into consideration the attack and defense-in-depth strategies after presenting the system mapping which is divided into four key parts (operator, gateway, communication link, and modular robot). Security attacks were also explained and discussed to highlight where the main vulnerability or security gap(s) could well be exploited, including the source and type of these attacks. Risks were also explained and detailed in a clear manner where the risk planning concept was introduced, in addition to the suitable security solutions types, as well as forensics and ethical hacking concepts being added and discussed to maintain a secure modular robotic environment which seemed to have lacked the security concept in its context. This thesis also presented key modular robotic solutions, which were mentioned, discussed, and analyzed, including the most recent ones, with many examples being added and highlighted.

To validate our contribution, in this thesis, we proposed a study of the efficiency of a set of BB robots in terms of communication delay and computation time. Based on the obtained results, we show that the communication delay linearly depends on the size of the message rather than the BB number, and initially presented Huffman as a lossless compression algorithm as a novel method, which was substituted by Brotli as an ideal solution that overcomes the limitations of Huffman when dealing with textual data. To reduce the communication delay, we proposed to add a lossless compression algorithm. We compared a set of recent efficient algorithms with the Huffman coding method. We expressed the compression ratio and compression/decompression execution time for each of them. The obtained results showed that the Brotli algorithm required the minimum overhead in terms of execution time and could achieve the maximum compression ratio. Therefore, this work indicated that the Brotli algorithm was introduced into the *Blinky Blocks* to reach the least communication delay, and as part of a novel compression solution within the programmable matter (i.e. IoPMoT).

Another contribution included the convergence of *Blinky Blocks* within the framework of modular robots in the context of IoT, marking a transformative leap in modular technology due to its real-time automation, adaptability, and reconfigurability. However, despite being the first security solution to be introduced to protect *Blinky Blocks* against hacking, PROLISEAN seems to suffer from several security weaknesses and flaws. As a result, in light of this paradigm shift, this thesis introduced a new Lightweight Cryptography and Authentication Protocol for *Blinky Blocks* (LCAPBB) which directly addresses the inherent limitations of its predecessor, PROLISEAN, by exploiting its cryptographic methodologies and offering a higher level of security in terms of authentication and cryptography. Hence, this thesis proposes LCAPBB as an enhancement of PROLISEAN within the realm of Lightweight Cryptographic Algorithms and Protocols for Programmable Matter (LCAPPM), addressing the main flaws and offering proper enhancements through cryp-

tographic approaches.

7.1/ SUMMARY OF THE PHD THESIS

The future of robotic systems lies within the core principles of self-reconfiguration, self-shaping, self-scaling, and self-healing processes, all aimed at achieving heightened levels of robustness, flexibility, and adaptability. In our thesis we endeavor to provide a comprehensive examination of the Modular Self-Reconfigurable Robots (MSRR) and Modular Self-Reconfigurable Robotic Systems (MSRRS) and their integration with the Internet of Things (IoT), emphasizing their crucial role in real-world and real-time IoT applications in the foreseeable future. By emphasizing the importance of their adoption, this study seeks to address unforeseen challenges and ensure the long-term sustainability of robotic systems capable of self-healing, self-reconfiguration, and self-replication tasks. Moreover, we also delve into the potential drawbacks, challenges, threats, and security vulnerabilities that may target modular robotic systems, aiming to pave the way for secure, accurate, and error-free domains such as the Internet of Robotic Things (IoRT) and the emerging Internet of Modular Robotic Things (IoMRT). The integration of IoMRT introduces a novel IoT concept that complements and transcends the limitations of the IoRT, facilitating enhanced connectivity, synchronization, and communication among autonomous robots and distributed systems. Additionally, we address challenges related to environmental conditions and terrain, offering greater adaptability and flexibility to varying circumstances. Through a comparative analysis with the IoRT domain, this thesis examines performance, safety, security, accuracy, and privacy aspects associated with IoMRT, providing insights into challenges and vulnerabilities. Furthermore, future directions for IoMRT are discussed, accompanied by lessons learned, suggestions, and recommendations to mitigate risks associated with its adoption. Ultimately, in this thesis, we seek to position IoMRT as an advanced iteration of IoRT, addressing its limitations through modular robotic concepts while acknowledging and addressing potential drawbacks and challenges to pave the way for the future of IoT. Additionally, we explore the integration of swarm robotics to form the Internet of Swarm Robotic Things (IoSRT) and the emergence of the Internet of Modular Swarm Robotic Things (IoMSRT), offering a glimpse into the future of modular robots and swarms combined, with potential applications in civilian and military domains.

In today's rapidly evolving technological landscape, the integration of robotics and the IoT presents boundless opportunities for innovation and advancement. As we witness the transformative potential of modular self-reconfigurable robots and their linkage to IoT,

it becomes imperative to delve deeper into their applications and implications. Motivated by the prospect of creating resilient, adaptable, and self-sustaining robotic systems, our interest in exploring the intersection of MSRR, MSRRS, and IoT has been ignited. By delving into the intricacies of self-reconfiguration, self-scaling, and self-healing processes, we aim to contribute to the development of next-generation robotic systems capable of overcoming unforeseen challenges and achieving unparalleled levels of efficiency and reliability. Through this research, we are driven to unravel the potential of the IoMRT and its transformative impact on real-world IoT applications. By addressing the inherent challenges and vulnerabilities while harnessing the collective power of swarm robotics, we are eager to pave the way for a future where modular robotic systems seamlessly integrate with all the IoT-related domains, revolutionizing industries and enhancing the quality of life.

The increasing interconnectedness of our world has propelled modular robotic systems into the forefront of technological innovation, playing integral roles within the IoT. These adaptable and self-configurable modular robots, whether lattice-based or in swarm formations, represent intelligent autonomous entities capable of flexible deployment in the IoRT and the emerging IoMRT. This study focuses on exploring the concept of IoMRT, particularly emphasizing lattice-based self-reconfigurable modular robots and their architectures, designs, criteria, and considerations regarding performance, safety, and security. By addressing current challenges and envisaging future advancements, particularly in enhancing security against physical and cyber threats, this research aims to harness the potential of integrating robotic systems and MSRR into the IoT under the IoMRT framework. These advancements hold promise across diverse industries, including law enforcement, military (counter-terrorism) operations, and healthcare, offering unparalleled improvements in power consumption, time efficiency, reusability, and adaptability. Furthermore, our research delves into optimizing communication efficiency within programmable matter systems, proposing a data compression scheme to reduce communication delays between modules significantly. Additionally, we introduce a Lightweight Cryptography and Authentication Protocol for modular robots, aiming to enhance security and prevent unauthorized access or modification of data. As we delve into future work, our focus remains on further enhancing communication protocols, exploring AI integration for improved performance, and researching lightweight authentication solutions to ensure the seamless integration of modular robotic systems into the IoT landscape. Through these endeavors, we aim to unlock the full potential of IoMRT, revolutionizing industries and shaping the future of robotics and IoT integration.

Our thesis contribution represents a significant advancement in the field of robotics and IoT through its exploration of the concept of the IoMRT. The adoption of modular nanorobots, as part of programmable matter, led us also to introduce a new novel concept called the Internet of Programmable Matter of Things (IoPMoT). By focusing on lattice-based self-reconfigurable modular robots and swarms, the study delves into their intricate architectures, designs, and criteria, emphasizing considerations related to safety, security, and privacy. Through thorough analysis and evaluation, our thesis identifies current challenges facing IoMRT implementation and proposes strategies for overcoming them, particularly in terms of bolstering security measures against both physical and cyber threats. Notably, we also introduce a novel Lightweight Cryptography and Authentication Protocol specifically tailored for modular robots, providing enhanced security measures to prevent unauthorized access or modification of data. Furthermore, we address the efficiency of communication within programmable matter systems, proposing a sophisticated data compression scheme aimed at significantly reducing communication delays between modules. This compression scheme, based on rigorous analysis and testing, demonstrates promising results in optimizing communication efficiency while minimizing computational overhead. Collectively, these contributions pave the way for the seamless integration of modular robotic systems into the IoT landscape, offering unprecedented advancements in various industries such as law enforcement, military (counter-terrorism) operations, and healthcare. Through these endeavors, our thesis contributes to shaping the future of robotics and IoT integration, revolutionizing industries and fostering technological innovation on a global scale.

In conclusion, our thesis surveys the characteristics and challenges of MSRRs within the IoT domain, emphasizing security considerations and proposing a framework for IoMRT security. Additionally, it explores the efficiency of *Blinky Blocks* robots in terms of communication delay and computation time, suggesting the use of the Brotli compression algorithm to minimize delay. Finally, it highlights the transformative potential of integrating nanorobots into modular robotics, proposing LCAPBB as a security enhancement and envisioning the evolution of modular robot swarms towards the Internet of Modular Swarm Robotic Things (IoMSRT). As a result, our thesis marks a significant step forward in the convergence of robotics and the Internet of Things, particularly through the exploration of the Internet of IoMRT with programmable matter to introduce the new IoPMoT concept. By addressing key challenges and proposing innovative solutions, including a novel Lightweight Cryptography and Authentication Protocol and an optimized data compression scheme, our thesis lays the groundwork for the seamless integration of modular robotic systems into the IoT landscape. Moving forward, our future work will focus on further enhancing communication protocols, exploring advanced AI integration for improved

performance and autonomy, and researching lightweight authentication solutions to ensure the secure and efficient operation of modular robotic systems within IoMRT. Through ongoing research and collaboration, we aim to unlock the full potential of IoMRT, revolutionizing industries and shaping the future of robotics and IoT integration. In factm we will also shed light on a list of future work that will extend beyond this thesis to cover another newly introduced concept called IoPMoT.

In other terms, our thesis offers a comprehensive exploration of Modular Self-Reconfigurable Robots (MSRRs) within the Internet of Things (IoT) domain, shedding light on their unique characteristics and the challenges they face. Though it is fascinating to see how modular robotics intersects with IoT, it is important to elaborate on the mentioned security considerations. Security is a critical aspect, especially when dealing with interconnected systems. In our thesis, we dive deeper into the potential vulnerabilities of MSRRs within IoT networks and propose a framework for IoMRT security. This framework addresses concerns such as authentication, encryption, and access control to ensure the integrity and confidentiality of data transmitted between modules and IoT devices. This sounds like a crucial aspect, especially considering the sensitive nature of data handled by IoT systems. However, it is also important to study both communication efficiency and computational performance. As a result, we also examined the efficiency of MSRRs in terms of communication delay and computation time. Given the distributed nature of modular robotic systems, minimizing communication delay is paramount for real-time applications. Our research suggests the adoption of the Brotli compression algorithm to reduce delay and optimize bandwidth usage, enhancing overall system performance. As part of our future enhancements or extensions to modular robotics, we propose an intriguing concept involving the integration of nanorobots into modular robotics. Nanorobots offer unique capabilities, such as precise manipulation at the nanoscale and targeted sensing within biological systems. We introduce the concept of Lightweight Cryptography-Assisted Protocol for Brotli-based Compression in Biorobots (LCAPBB) as a security enhancement for nanorobots, envisioning their integration into modular robot swarms. This allows us to foresee the evolution of modular robot swarms towards the Internet of Modular Swarm Robotic Things (IoMSRT), where the convergence of modular robotics, IoT, and nanotechnology holds immense potential for creating interconnected systems of unprecedented scale and versatility. We envision IoMSRT as a paradigm shift, where modular robot swarms collaborate seamlessly with IoT devices and nanorobots to tackle complex tasks and address emerging challenges in diverse domains.

As a result, our thesis represents a significant milestone in advancing the convergence of robotics and IoT, particularly through our exploration of IoMRT with programmable matter, introducing the innovative concept of the Internet of Programmable Matter-based Things (IoPMoT). IoPMoT fits into the broader landscape of robotics and IoT by leveraging

programmable matter technologies, such as shape-shifting materials or self-assembling structures, to create dynamic, adaptive systems that seamlessly integrate with IoT networks. This concept opens up new possibilities for creating self-reconfigurable, self-healing robotic systems that can autonomously adapt to changing environments and tasks. Our thesis tackles key challenges in realizing the vision of IoPMoT by proposing innovative solutions, including a novel Lightweight Cryptography and Authentication Protocol and an optimized data compression scheme. These contributions lay the groundwork for the seamless integration of modular robotic systems into the broader IoT landscape, enabling secure, efficient communication and operation. Moving forward, another part of our future work will focus on further enhancing communication protocols to accommodate the unique requirements of IoPMoT systems. Additionally, we aim to explore advanced AI integration techniques to improve performance and autonomy, enabling modular robots to adapt and learn in real time. Moreover, we'll investigate lightweight authentication solutions to ensure the secure operation of modular robotic systems within IoPMoT networks. Through ongoing research and collaboration, we aim to unlock the full potential of IoPMoT, revolutionizing industries and shaping the future of robotics and IoT integration. By enabling dynamic, adaptive systems that can self-organize and self-optimize, IoPMoT has the potential to drive innovation across sectors, from law enforcement, military, manufacturing, and logistics to healthcare and smart cities. Thus, introducing a new IoPMoT-based modular era of connectivity, efficiency, and innovation.

7.2/ PERSPECTIVES

As part of our perspectives for future work, further studies are going to be conducted to cover a variety of objectives that we will be working on. Before starting to list them, it is important to explain each term before listing them and present the proposed plan to integrate them as part of our future work, while showing, from our perspective, both advantages and expectations during the implementation phase while presenting the possible occurrence of difficulties that can take the form of disadvantages.

AI Integration. AI plays a crucial role in various IoMRT-based aspects. This covers the following modular robotic situations. **Self-Organization and Self-Assembly**, where AI algorithms can enable modules to autonomously organize and assemble themselves into desired configurations. **Adaptation and Learning**, where AI algorithms, such as reinforcement learning, can help modular robots learn and optimize their behaviors over time. **Cooperation and Coordination**, where AI facilitates cooperation and coordination in scenarios where multiple modular robots need to work together to accomplish tasks. **Fault Tolerance and Robustness**, where AI techniques like fault detection, diagnosis,

and recovery enhance the system's resilience by identifying issues and implementing appropriate responses.

Reinforcement Learning. RL allows modular robots with interchangeable modules to autonomously learn and adapt their behaviors to achieve various tasks and goals. This includes **Modular Agent Architecture**, where RL algorithms train individual modules or groups of modules to perform specific tasks or behaviors. **State Representation**, where state representations capture relevant aspects of the robot's configuration, surroundings, and task context, providing input for decision-making and learning algorithms. **Action Selection and Control**, where action selection mechanisms, such as policy gradients, Q-learning, or actor-critic methods, determine how agents explore and exploit the environment to learn optimal control strategies. **Reward Design and Feedback**, where the reward design influences the learning process by shaping agent preferences and encouraging adaptive behaviors, while feedback mechanisms provide timely rewards or penalties to RL agents, facilitating learning progress and convergence towards optimal policies.

Predefined Knowledge for Self-Reconfiguration. Self-reconfiguration is a key capability in modular robotics that allows modular robots composed of interchangeable modules to autonomously change their shape, structure, or functionality to adapt to different tasks or environments through coordinated actions, often guided by artificial intelligence algorithms. This covers: **Module Design**, where modular robots consist of individual modules with standardized connectors such as actuators, sensors, processors, and communication components or interfaces that enable them to mechanically and electronically connect. **Topology Representation**, which includes graphs or matrices to model the configuration space and relationships between modules, facilitating planning and control algorithms. **Motion Planning**, which includes algorithms that determine the sequence of actions required for modules to achieve a desired reconfiguration while avoiding collisions and constraints. **Sensing and Perception**, which enables modules to perceive their environment and detect relevant features or obstacles during self-reconfiguration.

Enhanced Fault-Tolerance. This allows modular robots to ensure robust performance and resilience against failures in individual modules or components. This allows **Redundancy and Duplication**, which provides backup in case of failure by replacing failed modules to maintain the modular system's functionality without interruption. **Dynamic Reconfiguration**, which enables modular robots to adapt their configurations in response to faults or changes by disconnecting and re-configuring themselves to bypass the faulty component or redistribute tasks. **Fault Detection and Diagnosis**, which monitors the

health and performance of individual modules and identifies anomalies or deviations from expected behavior to detect faults such as sensor failures, actuator malfunctions, or communication errors. **Adaptive Recovery Strategies**, which allows modular robots to dynamically adjust their behaviors and strategies in response to varying fault conditions.

Multi-Network Communications. This communication type enables modular robots to establish and maintain connectivity through multiple communication channels both autonomously and simultaneously. **Heterogeneous Communication Modalities**, where modular robots can integrate multiple communication modalities to establish diverse communication links including wired and wireless types, while acoustic and optical communication modalities can be used to enable communication in underwater or line-of-sight environments. **Adaptive Communication Protocols**, which dynamically select and switch between different communication modalities, while optimizing communication performance by balancing factors such as latency, throughput, and energy consumption. **Redundant Communication Paths**, which enhance reliability and fault-tolerance in modular robotic systems by providing alternative routes for data transmission, while establishing parallel or backup communication links to mitigate the impact of network failures, interference, or signal attenuation. **Distributed Network Management**, includes protocols that enable modular robots to autonomously configure, monitor, and manage communication networks without centralized control.

- **Integration of Reinforcement Learning:** The concept of reinforcement learning will be studied to cover modular robot tasks within the context of programmable matter (i.e. IoPMoT) by getting the optimal decision with the minimum cost in terms of delay and communication and to avoid errors. Such integration allows for autonomous adaptation, efficient task learning, and optimization of behaviors through interactive exploration and reward-driven feedback. However, disadvantages such as sample inefficiency, exploration-exploitation trade-offs, and difficulty in defining reward functions may persist. In terms of expectations, this solution requires careful consideration of training stability, safety, and generalization capabilities. This approach can be achieved by developing and designing algorithms that allow modules to autonomously learn and adapt behaviors based on environmental feedback and objectives' tasks, to facilitate improved decision-making and task execution via collaboration within the modular robotic system.
- **Study of Wireless Communication:** The application of wireless communication into *Blinky Blocks* will be added to allow remote monitoring and control remotely. It can be used as another way to communicate messages, which helps to ensure data availability without the need for physical interaction. This study facilitates flexible, scalable, and decentralized control architectures, enabling seamless coordination and communication among modules without the constraints of physical connections. However, limitations

such as limited bandwidth, potential interference, and vulnerability to cyber attacks will also arise. In terms of expectations, this solution surely requires careful design considerations and optimization strategies. In terms of expectations, this solution requires robust protocols, energy-efficient designs, and security measures to ensure reliable and secure communication. This approach can be achieved by conducting empirical experimentation and simulation to analyze signal propagation, interference mitigation, and network topology optimization, which allows us to adopt the right network protocol concerning its advantages and drawbacks. For example, Bluetooth connection has been explored using Blinky blocks. Our work will be based on ensuring the security of this short-ranged wireless connection, and the possibility to evolve it to cover medium and long range if possible.

- **Hybrid Communication:** This type of communication will also be studied to see how *Blinky Blocks* can initiate reciprocal communication with other modular robotic types. A key advantage is that hybrid communication in modular robots combines the benefits of both wired and wireless communication, offering improved reliability, flexibility, and energy efficiency by leveraging the strengths of each modality. A main disadvantage includes the complexity of network configuration, synchronization, and fault tolerance, which prove to be a serious integration issue for multiple communication modalities. This approach can be achieved via a communication system that involves both wired and wireless technologies that allow modular robots to use the most suitable communication mode based on distance, proximity, bandwidth requirements, and environmental conditions including obstacles. More specifically, this future work will be based on the Blinky Blocks' ability to switch from (Bluetooth) wireless communication to its initial physical communication form in case of interruption caused by a jamming/de-authentication attack or connection failure.
- **Enhanced Security:** By applying more sophisticated lightweight security solutions with a higher level of security and protection to ensure a higher security level for resource-constrained modular robotic systems. Authentication is also studied in terms of message and source authentication to secure the message and the sending/receiving entities. This enhanced security ensures protection against unauthorized access, data breaches, and cyber attacks, safeguarding sensitive information and maintaining the integrity and confidentiality of robotic systems. However, it will surely cause a challenge in terms of additional computational overhead, complexity in system design, and potential trade-offs with performance and resource constraints. In terms of expectations, this solution surely necessitates a careful balance between security requirements and operational efficiency. In this regard, this work will either include a strong authentication mechanism between entities, based on the biometric concept of each Blinky Block identity, or a pseudo-random lightweight cryptographic mechanism that is secure and suitable for these resource-constrained modular robots.

- **Selective Crypto-compression Approach:** Combines the previous contribution of compression with the proposed lightweight cipher scheme to ensure data confidentiality with minimum overhead in terms of resources and delay. In terms of advantages, the selective crypto-compression approach in modular robots offers a balance between security and resource efficiency, allowing for secure communication while minimizing computational overhead and communication bandwidth usage. However, such implementation may introduce complexity in encryption and compression algorithms, potentially impacting real-time performance. In terms of expectations, this solution requires careful consideration of trade-offs between security, efficiency, and latency. This involves selectively integrating compression and encryption techniques to data transmissions between modules to secure information exchange while minimizing computational resources and network overhead. More specifically, this selective crypto-compression approach will work on encrypting the header of each message sent, since the body of this message is already compressed and all its patterns were changed, leaving the header exposed in a plaintext format.
- **Monitoring Approach:** To check malicious traffic, it can help to detect attacks against connected modular robots such as DoS/DDoS. This approach can be based on machine learning or statistical algorithms. This approach can help to monitor the behavior of modular robots and communication endpoints to detect any abnormal or unauthorized activities including sudden increases in message traffic, unusual message destinations, or unauthorized access attempts. This will surely enable real-time assessment of system health, performance, and environmental conditions, facilitating proactive fault detection, diagnosis, and optimization for enhanced reliability and efficiency. However, this will require additional computational and energy overhead, which will impact the system's responsiveness and autonomy. In terms of expectations, this solution requires efficient sensor deployment and data processing techniques to mitigate resource constraints. This involves implementing a system that allows continuous observation and management of the individual modules comprising the robot. This approach can be developed to monitor and analyze the transmitted patterns between Blinky Blocks, and with the use of AI, it should be able to differentiate between the normal and abnormal behaviors of each Blinky Block, allowing us to detect the source of the issue/attack and its impact while offering suitable solutions to mitigate this threat.
- **Introducing Network Coding into Modular Robots:** It can be one of the possible ways to enhance modular robots' communication as it can have multiple paths. Introducing network coding can help to ensure data availability and to resist channel errors or packet loss. This will surely enhance data reliability, throughput, and fault tolerance by allowing modules to efficiently encode and decode information packets, enabling robust communication in dynamic and noisy environments. However, it will also introduce computational overhead, latency, and complexity in encoding and decoding

processes. In terms of expectations, this solution requires careful optimization and trade-offs between communication efficiency and processing resources. This involves designing communication protocols that allow modules to exchange and process data in a distributed manner to enhance reliability, adaptability, and efficiency. This is still a new concept that requires us to further study this whole concept before proposing a suitable solution in the near future.

BIBLIOGRAPHY

- [1] Zahrah A Almusaylim, NZ Jhanjhi, and Abdulaziz Alhumam. Detection and mitigation of rpl rank and version number attacks in the internet of things: Srpl-rp. *Sensors*, 20(21):5997, 2020.
- [2] Amira Abdel-Rahman, Christopher Cameron, Benjamin Jenett, Miana Smith, and Neil Gershenfeld. Self-replicating hierarchical modular robotic swarms. *Communications Engineering*, 1(1):35, 2022.
- [3] Temitope Francis Abiodun and Captain Raheem Taofeek. Unending war on boko haram terror in northeast nigeria and the need for deployment of military robots or autonomous weapons systems to complement military operations. *Journal DOI*, 6(6), 2020.
- [4] Evan Ackerman. Nasa training'swarmie'robots for space mining. *IEEE Spectrum Automaton*, 2014.
- [5] André Silva Aguiar, Filipe Neves dos Santos, José Boaventura Cunha, Héber Sobreira, and Armando Jorge Sousa. Localization and mapping for robots in agriculture and forestry: A survey. *Robotics*, 9(4):97, 2020.
- [6] Hossein Ahmadzadeh and Ellips Masehian. Modular robotic systems: Methods and algorithms for abstraction, planning, control, and synchronization. *Artificial Intelligence*, 223:27–64, 2015.
- [7] Hossein Ahmadzadeh, Ellips Masehian, and Masoud Asadpour. Modular robotic systems: Characteristics and applications. *Journal of Intelligent & Robotic Systems*, 81(3-4):317–357, 2016.
- [8] Tajim Md Niamat Ullah Akhund, Watry Biswas Jyoty, Md Abu Bakkar Siddik, Nishat Tasnim Newaz, SK Ayub Al Wahid, and M Mesbahuddin Sarker. lot based low-cost robotic agent design for disabled and covid-19 virus affected people. In *2020 fourth world conference on smart trends in systems, security and sustainability (WorldS4)*, pages 23–26. IEEE, 2020.
- [9] Ian F Akyildiz and Ahan Kak. The internet of space things/cubesats. *IEEE Network*, 33(5):212–218, 2019.

- [10] Rana Alabdan. Phishing attacks survey: Types, vectors, and technical approaches. *Future Internet*, 12(10):168, 2020.
- [11] Jyrki Alakuijala, Andrea Farruggia, Paolo Ferragina, Eugene Kliuchnikov, Robert Obryk, Zoltan Szabadka, and Lode Vandevenne. Brotli: A general-purpose data compressor. *ACM Transactions on Information Systems (TOIS)*, 37(1):1–30, 2018.
- [12] Jyrki Alakuijala, Evgenii Kliuchnikov, Zoltan Szabadka, and Lode Vandevenne. Comparison of brotli, deflate, zopfli, lzma, lzham and bzip2 compression algorithms. *Google Inc*, pages 1–6, 2015.
- [13] Mary B Alatisse and Gerhard P Hancke. A review on challenges of autonomous mobile robot and sensor fusion methods. *IEEE Access*, 8:39830–39846, 2020.
- [14] Jürgen Altmann and Dieter Suter. *Small and Very Small Armed Aircraft and Missiles: Trends in Technology and Preventive Arms Control*. Universitätsbibliothek Dortmund, 2023.
- [15] Hamzeh Alzu'bi, Iyad Mansour, and Osamah Rawashdeh. Loon copter: Implementation of a hybrid unmanned aquatic–aerial quadcopter with active buoyancy control. *Journal of field Robotics*, 35(5):764–778, 2018.
- [16] Victor P Andreev. Control system mobile robots with modular architecture as a multi-agent system with a hierarchical topology. *Annals of DAAAM & Proceedings*, 30, 2019.
- [17] MAJ Andrew, W Sanders, and F Leavenworth. *Drone swarms—a monograph by school of advanced military studies*, 2017.
- [18] Pushkar Aneja, Maanya Manocha, Shagun Verma, and Madhumita Kathuria. An overview of cyber risks in internet of things (iot) world. *Int. J. Intell. Syst. Technol. Appl.*, 8(6):235–267, 2020.
- [19] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed computing*, 18(4):235–253, 2006.
- [20] Emil Archambault and Yannick Veilleux-Lepage. Drone imagery in islamic state propaganda: flying like a state. *International Affairs*, 2020.
- [21] Farshad Arvin, Simon Watson, Ali Emre Turgut, Jose Espinosa, Tomáš Krajník, and Barry Lennox. Perpetual robot swarm: long-term autonomy of mobile robots using on-the-fly inductive charging. *Journal of Intelligent & Robotic Systems*, 92:395–412, 2018.

- [22] Michael P Ashley-Rollman, Seth Copen Goldstein, Peter Lee, Todd C Mowry, and Padmanabhan Pillai. Meld: A declarative approach to programming ensembles. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2794–2800. IEEE, 2007.
- [23] Joseph Assaker, Abdallah Makhoul, Julien Bourgeois, Benoît Piranda, and Jacques Demerjian. A dynamic id assignment approach for modular robots. In *International Conference on Advanced Information Networking and Applications*, pages 91–104. Springer, 2022.
- [24] Ahmad W Atamli and Andrew Martin. Threat-based security analysis for the internet of things. In *Secure Internet of Things (SloT), 2014 International Workshop on*, pages 35–43. IEEE, 2014.
- [25] Terje Aven. Risk assessment and risk management: Review of recent advances on their foundation. *European Journal of Operational Research*, 253(1):1–13, 2016.
- [26] Sharif Azem, Anam Tahir, and Heinz Koepl. Dynamic time slot allocation algorithm for quadcopter swarms. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2022.
- [27] Mohd Hafiz Baharuddin. Development of a stingray robot. 2012.
- [28] Rajib Banerjee and Sipra Das Bit. An energy efficient image compression scheme for wireless multimedia sensor network using curve fitting technique. *Wireless Networks*, 25:167–183, 2019.
- [29] Michael D Bartlett, Michael D Dickey, and Carmel Majidi. Self-healing materials for soft-matter machines and electronics. *NPG Asia Materials*, 11(1):1–4, 2019.
- [30] Marcin Bartosiak, Gianni Bonelli, Lorenzo Stefano Maffioli, Ugo Palaoro, Francesco Dentali, Giovanni Poggialini, Federica Pagliarin, Stefano Denicolai, and Pietro Previtali. Advanced robotics as a support in healthcare organizational response. a covid-19 pandemic case. In *Healthcare Management Forum*, page 08404704211042467. SAGE Publications Sage CA: Los Angeles, CA, 2021.
- [31] Jad Bassil, Mohamad Moussa, Abdallah Makhoul, Benoît Piranda, and Julien Bourgeois. Linear distributed clustering algorithm for modular robots based programmable matter. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3320–3325. IEEE, 2020.
- [32] Jad Bassil, Benoît Piranda, Abdallah Makhoul, and Julien Bourgeois. A new porous structure for modular robots. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 1539–1541, 2022.

- [33] Jad Bassil, Benoit Piranda, Abdallah Makhoul, and Julien Bourgeois. Repost: Distributed self-reconfiguration algorithm for modular robots based on porous structure. In *IEEE RSJ International Conference on Intelligent Robots and Systems (IROS 2022)*, Kyoto, Japan, oct 2022.
- [34] Jad Bassil, Perla Tannoury, Benoît Piranda, Abdallah Makhoul, and Julien Bourgeois. Fault-tolerance mechanism for self-reconfiguration of modular robots. In *2022 International Wireless Communications and Mobile Computing (IWCMC)*, pages 360–365. IEEE, 2022.
- [35] Jad Bassil, Jean-Paul A Yaacoub, Benoît Piranda, Abdallah Makhoul, and Julien Bourgeois. Distributed shape recognition algorithm for lattice-based modular robots. In *2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 85–91. IEEE, 2023.
- [36] Ranbir Singh Batth, Anand Nayyar, and Amandeep Nagpal. Internet of robotic things: driving intelligent robotics of future-concept, architecture, applications and technologies. In *2018 4th International Conference on Computing Sciences (ICCS)*, pages 151–160. IEEE, 2018.
- [37] Rebecca Belisle, Chih-Han Yu, and Radhika Nagpal. Mechanical design and locomotion of modular-expanding robots. 2010.
- [38] James G Bellingham. Platforms: Autonomous underwater vehicles. 2009.
- [39] Gerardo Beni. Swarm intelligence. *Complex Social and Behavioral Systems: Game Theory and Agent-Based Models*, pages 791–818, 2020.
- [40] Angat Pal Singh Bhatia, SangHyeok Han, and Osama Moselhi. A simulation-based statistical method for planning modular construction manufacturing. *Journal of Information Technology in Construction (ITcon)*, 27(7):130–144, 2022.
- [41] Giovanni Bianchi, Kavinda Pradeep Herath Herath Mudiyansele, and Simone Cinquemani. Design of a swimming snake robot. In *Bioinspiration, Biomimetics, and Bioreplication XII*, volume 12041, pages 75–81. SPIE, 2022.
- [42] Pietro Bilancia, Luca Monari, Roberto Raffaelli, Margherita Peruzzini, and Marcello Pellicciari. Accurate transmission performance evaluation of servo-mechanisms for robots. *Robotics and Computer-Integrated Manufacturing*, 78:102400, 2022.
- [43] R Adam Bilodeau and Rebecca K Kramer. Self-healing and damage resilience for soft robotics: A review. *Frontiers in Robotics and AI*, 4:48, 2017.
- [44] John Blaxland, Srinjoy Bose, and Paul Lushenko. The significance—and potential—of a fourth wave of drone warfare scholarship. In *Drones and Global Order*, pages 245–259. Routledge, 2022.

- [45] Joel Block. A laws of war review of contemporary land-based missile defence system 'iron dome'. *Scientia Militaria: South African Journal of Military Studies*, 45(2):105–128, 2017.
- [46] Brandon C Boatwright and Andrew S Pyle. “don't mess with ukrainian farmers”: An examination of ukraine and kyiv's official twitter accounts as crisis communication, public diplomacy, and nation building during russian invasion. *Public Relations Review*, 49(3):102338, 2023.
- [47] Ingvild Bode and Thomas Frank Arthur Watts. Loitering munitions and unpredictability: Autonomy in weapon systems and challenges to human control. 2023.
- [48] Michael A Boemo, Andrew J Turberfield, and Luca Cardelli. Automated design and verification of localized dna computation circuits. In *International Workshop on DNA-Based Computers*, pages 168–180. Springer, 2015.
- [49] Nicolas Boillot, Dominique Dhoutaut, and Julien Bourgeois. New applications for mems modular robots using wireless communications. *IEEE Systems Journal*, 11(2):1094–1106, 2015.
- [50] Hristo Bojinov, Arancha Casal, and Tad Hogg. Emergent structures in modular self-reconfigurable robots. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 1734–1741. IEEE, 2000.
- [51] Jack Bordes, Kyle Doody, John Feeney, Kyle Hamrock, Collin Riechman, and Benjamin Morales. Fires support next. 2022.
- [52] Amanda Bouman, Muhammad Fadhil Ginting, Nikhilesh Alatur, Matteo Palieri, David D Fan, Thomas Touma, Torkom Pailevanian, Sung-Kyun Kim, Kyohei Otsu, Joel Burdick, et al. Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2518–2525. IEEE, 2020.
- [53] Julien Bourgeois, Jiannong Cao, Michel Raynal, Dominique Dhoutaut, Benoît Piranda, Eugen Dedu, Ahmed Mostefaoui, and Hakim Mabed. Coordination and computation in distributed intelligent mems. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pages 129–136. IEEE, 2013.
- [54] Julien Bourgeois, Benoit Piranda, Andre Naz, Nicolas Boillot, Hakim Mabed, Dominique Dhoutaut, Thadeu Knychala Tucci, and Hicham Lakhlef. Programmable matter as a cyber-physical conjugation. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2016)*, pages 002942 – 002947, Budapest, Hungary, oct 2016. IEEE, IEEE.

- [55] Julien Bourgeois, Benoit Piranda, Andre Naz, Nicolas Boillot, Hakim Mabed, Dominique Dhoutaut, Thadeu Tucci, and Hicham Lakhlef. Programmable matter as a cyber-physical conjugation. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 002942–002947. IEEE, 2016.
- [56] Kenny Breuer. Flight of the robobee, 2019.
- [57] Alberto Brunete, Ernesto Gambao, Miguel Hernando, and Raquel Cedazo. Smart assistive architecture for the integration of iot devices, robotic systems, and multi-modal interfaces in healthcare environments. *Sensors*, 21(6):2212, 2021.
- [58] Alberto Brunete, Avinash Ranganath, Sergio Segovia, Javier Perez de Frutos, Miguel Hernando, and Ernesto Gambao. Current trends in reconfigurable modular robots design. *International Journal of Advanced Robotic Systems*, 14(3):1729881417710457, 2017.
- [59] Robert J Bunker and John P Sullivan. Cartel drone utilization combat trends. *Criminal Drone Evolution: Cartel Weaponization of Aerial Ieds*, 2021.
- [60] Zack Butler and Daniela Rus. Distributed planning and control for modular robots with unit-compressible modules. *The International Journal of Robotics Research*, 22(9):699–715, 2003.
- [61] Mark Button. economic and industrial espionage, 2020.
- [62] Sebastian Büttrich. 3d modeling with openscad-part 1. *LOW-COST 3D PRINTING*, page 83, 2018.
- [63] Cindy Calderón-Arce, Juan Carlos Brenes-Torres, and Rebeca Solis-Ortega. Swarm robotics: Simulators, platforms and applications review. *Computation*, 10(6):80, 2022.
- [64] Jason Campbell and Padmanabhan Pillai. Collective actuation. *The International Journal of Robotics Research*, 27(3-4):299–314, 2008.
- [65] Sarah Cannon, Joshua J Daymude, Dana Randall, and Andréa W Richa. A markov chain algorithm for compression in self-organizing particle systems. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, pages 279–288, 2016.
- [66] Stephen J Carlson, Prateek Arora, and Christos Papachristos. A multi-vtol modular aspect ratio reconfigurable aerial robot. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8–15. IEEE, 2022.

- [67] Luca Caruana and Emmanuel Francalanza. A safety 4.0 approach for collaborative robotics in the factories of the future. *Procedia Computer Science*, 217:1784–1793, 2023.
- [68] Mahmut Cengiz. Prevention of the procurement of arms and explosives by terrorist groups.
- [69] Diego Centelles, Antonio Soriano-Asensi, José Vicente Martí, Raúl Marín, and Pedro J Sanz. Underwater wireless communications for cooperative robotics with uwsim-net. *Applied Sciences*, 9(17):3526, 2019.
- [70] Cesar Cerrudo and Lucas Apa. Hacking robots before skynet. *Cybersecurity Insight, IOActive Report, Seattle, USA*, 2017.
- [71] Cameron T Chalk, Bin Fu, Alejandro Huerta, Mario A Maldonado, Eric Martinez, Robert T Schweller, and Tim Wylie. Flipping tiles: concentration independent coin flips in tile self-assembly. In *International Workshop on DNA-Based Computers*, pages 87–103. Springer, 2015.
- [72] Mohammadreza Chamanbaz, David Mateo, Brandon M Zoss, Grgur Tokić, Erik Wilhelm, Roland Bouffanais, and Dick KP Yue. Swarm-enabling technology for multi-robot systems. *Frontiers in Robotics and AI*, 4:12, 2017.
- [73] Fred Chen, Anantha P Chandrakasan, and Vladimir M Stojanovic. Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors. *IEEE journal of solid-state circuits*, 47(3):744–756, 2012.
- [74] Peng Chen, Yuanjie Dang, Ronghua Liang, Wei Zhu, and Xiaofei He. Real-time object tracking on a drone with multi-inertial sensing data. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):131–139, 2017.
- [75] Xinye Chen, Ping Zhang, Guanglong Du, and Fang Li. A distributed method for dynamic multi-robot task allocation problems with critical time constraints. *Robotics and Autonomous Systems*, 118:31–46, 2019.
- [76] Denis Chikurtev. Service-oriented architecture for control of modular robots. In *2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC)*, pages 304–309. IEEE, 2022.
- [77] Shushman Choudhury, Jayesh K Gupta, Mykel J Kochenderfer, Dorsa Sadigh, and Jeannette Bohg. Dynamic multi-robot task allocation under uncertainty and temporal constraints. *Autonomous Robots*, 46(1):231–247, 2022.
- [78] David Johan Christensen. Evolution of shape-changing and self-repairing control for the atron self-reconfigurable robot. In *Proceedings 2006 IEEE International*

- Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2539–2545. IEEE, 2006.
- [79] David Johan Christensen, Ulrik Pagh Schultz, and Kasper Stoy. A distributed and morphology-independent strategy for adaptive locomotion in self-reconfigurable modular robots. *Robotics and Autonomous Systems*, 61(9):1021–1035, 2013.
- [80] Timothy H Chung. Offensive swarm-enabled tactics (offset). DARPA, 2021.
- [81] George W Clark, Michael V Doran, and Todd R Andel. Cybersecurity issues in robotics. In *Cognitive and Computational Aspects of Situation Management (CogSIMA), 2017 IEEE Conference on*, pages 1–5. IEEE, 2017.
- [82] B Cochrane. The development of the british approach to improvised explosive device disposal in northern ireland. 2015.
- [83] Andrew Coco. Jury-rigged jihād: The impact of improvised weapons use by sub-state armed groups in iraq & syria. 2021.
- [84] Yann Collet and Murray S. Kucherawy. Zstandard compression and the application/zstd media type. *RFC*, 8878:1–45, 2018.
- [85] Tim Cooke. *A Timeline of Military Robots and Drones*. Capstone, 2017.
- [86] Nikolaus Correll. Ping pong ball-sized robot for various applications in industries. *Advanced Manufacturing Technology*, 34(10):1–3, 2013.
- [87] Nikolaus Correll, Chris Wailes, and Scott Slaby. A one-hour curriculum to engage middle school students in robotics and computer science using cubelets. In *Distributed Autonomous Robotic Systems*, pages 165–176. Springer, 2014.
- [88] Fernanda Coutinho, Marco Silva, and Jorge Barreiros. A study of microcontroller simulator tools for autonomous and online learning. In *2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE)*, pages 1–6. IEEE, 2022.
- [89] Alessandro Crespi and Auke Jan Ijspeert. Online optimization of swimming and crawling in an amphibious snake robot. *IEEE Transactions on robotics*, 24(1):75–87, 2008.
- [90] Alessandro Crespi and Auke Jan Ijspeert. Salamandra robotica: a biologically inspired amphibious robot that swims and walks. In *Artificial life models in hardware*, pages 35–64. Springer, 2009.
- [91] Alessandro Crespi, Konstantinos Karakasiliotis, Andre Guignard, and Auke Jan Ijspeert. Salamandra robotica ii: an amphibious robot to study salamander-like swimming and walking gaits. *IEEE Transactions on Robotics*, 29(2):308–320, 2013.

- [92] Tomasz Czapla and Józef Wrona. Technology development of military applications of unmanned ground vehicles. In *Vision Based Systems for UAV Applications*, pages 293–309. Springer, 2013.
- [93] Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.
- [94] Tariq Dana. A cruel innovation: Israeli experiments on Gaza's great march of return. *Sociology of Islam*, 8(2):175–198, 2020.
- [95] PK Das and Prabir Kumar Jena. Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators. *Applied Soft Computing*, 92:106312, 2020.
- [96] David J Day and Zheng-Xu Zhao. Protecting against address space layout randomisation (ASLR) compromises and return-to-libc attacks using network intrusion detection systems. *International Journal of Automation and Computing*, 8(4):472–483, 2011.
- [97] Joshua J Daymude, Kristian Hinnenthal, Andréa W Richa, and Christian Scheideler. Computing by programmable particles. In *Distributed Computing by Mobile Entities*, pages 615–681. Springer, 2019.
- [98] Edi Moreira M de Araujo, Augusto Loureiro da Costa, and Alejandro RG Ramirez. Interaction protocols for multi-robot systems in industry 4.0. *Robotics Software Design and Engineering*, page 155, 2021.
- [99] Iris De Falco, Giada Gerboni, Matteo Cianchetti, and Arianna Menciassi. Design and fabrication of an elastomeric unit for soft modular robots in minimally invasive surgery. *JoVE (Journal of Visualized Experiments)*, (105):e53118, 2015.
- [100] Satcha de Henning Michaëlis. Turkey's and Iran's drone supply in the war in Ukraine. *Policy*, (21), 2023.
- [101] Chacko John Deepu, Chun-Huat Heng, and Yong Lian. A hybrid data compression scheme for power reduction in wireless sensors for IoT. *IEEE transactions on biomedical circuits and systems*, 11(2):245–254, 2016.
- [102] Erik D Demaine, Matthew J Patitz, Robert T Schweller, and Scott M Summers. Self-assembly of arbitrary shapes using RNAse enzymes: Meeting the Kolmogorov bound with small scale factor. *arXiv preprint arXiv:1004.4383*, 2010.
- [103] Zahra Derakhshandeh, Robert Gmyr, Andréa W Richa, Christian Scheideler, and Thim Strothmann. An algorithmic framework for shape formation problems in self-organizing particle systems. In *Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication*, pages 1–2, 2015.

- [104] Zahra Derakhshandeh, Robert Gmyr, Andréa W Richa, Christian Scheideler, and Thim Strothmann. Universal shape formation for programmable matter. In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 289–299, 2016.
- [105] Zahra Derakhshandeh, Robert Gmyr, Thim Strothmann, Rida Bazzi, Andréa W Richa, and Christian Scheideler. Leader election and shape formation with self-organizing programmable matter. In *International Workshop on DNA-Based Computers*, pages 117–132. Springer, 2015.
- [106] Saddam Hocine Derrouaoui, Yasser Bouzid, Mohamed Guiatni, and Islam Dib. A comprehensive review on reconfigurable drones: Classification, characteristics, design and control technologies. *Unmanned Systems*, 10(01):3–29, 2022.
- [107] Daniel J Dewey, Michael P Ashley-Rollman, Michael De Rosa, Seth Copen Goldstein, Todd C Mowry, Siddhartha S Srinivasa, Padmanabhan Pillai, and Jason Campbell. Generalizing metamodules to simplify planning in modular robotic systems. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1338–1345. IEEE, 2008.
- [108] HN Dheemanth. Lzw data compression. *American Journal of Engineering Research*, 3(2):22–26, 2014.
- [109] Dominique Dhoutaut, Benoît Piranda, and Julien Bourgeois. Efficient simulation of distributed sensing and control environments. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pages 452–459. IEEE, 2013.
- [110] Giuseppe Antonio Di Luna, Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Giovanni Viglietta. Line recovery by programmable particles. In *Proceedings of the 19th International Conference on Distributed Computing and Networking*, pages 1–10, 2018.
- [111] Rishabh Dixit, Amrit Singh Bedi, Ruchi Tripathi, and Ketan Rajawat. Online learning with inexact proximal online gradient descent algorithms. *IEEE Transactions on Signal Processing*, 67(5):1338–1352, 2019.
- [112] Michael Donevski and Tanveer Zia. Cyber diversity index for sustainable self-control of machines. *Cybernetics and Systems*, pages 1–27, 2022.
- [113] M Dorigo, L Gambardella, F Mondada, D Floreano, and S Nolfi. Swarmanoid: Towards humanoid robotic swarms.

- [114] Marco Dorigo. The swarm-bots and swarmanoid experiments in swarm robotics. In *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 1–1. IEEE, 2014.
- [115] Marco Dorigo, Dario Floreano, Luca Maria Gambardella, Francesco Mondada, Stefano Nolfi, Tarek Baaboura, Mauro Birattari, Michael Bonani, Manuele Brambilla, Arne Brutschy, et al. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4):60–71, 2013.
- [116] Matthew Doyle. The propulsion of reconfigurable modular robots in fluidic environments. 2019.
- [117] Matthew J Doyle, João V Amorim Marques, Isaac Vandermeulen, Christopher Parrott, Yue Gu, Xinyu Xu, Andreas Kolling, and Roderich Groß. Modular fluidic propulsion robots. *IEEE transactions on robotics*, 37(2):532–549, 2020.
- [118] Daniel S Drew. Multi-agent systems for search and rescue applications. *Current Robotics Reports*, 2:189–200, 2021.
- [119] Daniel S Drew, Matthew Devlin, Elliot Hawkes, and Sean Follmer. Acoustic communication and sensing for inflatable modular soft robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11827–11833. IEEE, 2021.
- [120] Tyago Driemeyer, Victoria Ellwanger, and Vinícius Nardin. Lethal autonomous weapons. *FACULDADE DE CIÊNCIAS ECONÔMICAS*, page 94.
- [121] Akhil Dubey, Deepak Meena, and Shaili Gaur. A survey in hello flood attack in wireless sensor networks. *Int. J. Eng. Res. Technol*, 3, 2014.
- [122] Morris Dworkin, NATIONAL INST OF STANDARDS, and TECHNOLOGY GAITHERSBURG MD COMPUTER SECURITY DIV. Recommendation for block cipher modes of operation. methods and techniques. 2001.
- [123] Didier El-Baz, Benoît Piranda, and Julien Bourgeois. A distributed algorithm for a reconfigurable modular surface. In *2014 IEEE International Parallel & Distributed Processing Symposium Workshops*, pages 1591–1598. IEEE, 2014.
- [124] Mohammad Eslami. Iran’s drone supply to russia and changing dynamics of the ukraine war. *Journal for Peace and Nuclear Disarmament*, 5(2):507–518, 2022.
- [125] Christian Esposito and Mario Ciampi. On security in publish/subscribe services: A survey. *IEEE Communications Surveys & Tutorials*, 17(2):966–997, 2015.
- [126] Bernard Everett. Optically transparent: The rise of industrial espionage and state-sponsored hacking. *Computer Fraud & Security*, 2013(10):13–16, 2013.

- [127] HR Everett and Michael Toscano. Unmanned ground vehicles. 2015.
- [128] Andres Faiña. Evolving modular robots: Challenges and opportunities. In *ALIFE 2021: The 2021 Conference on Artificial Life*. MIT Press, 2021.
- [129] Andrés Faiña, Francisco Bellas, Daniel Souto, and Richard J Duro. Towards an evolutionary design of modular robots for industry. In *International Work-Conference on the Interplay Between Natural and Artificial Computation*, pages 50–59. Springer, 2011.
- [130] Jiwei Fan, Ruitao Lu, Xiaogang Yang, Fan Gao, Qingge Li, and Jun Zeng. Design and implementation of intelligent eod system based on six-rotor uav. *Drones*, 5(4):146, 2021.
- [131] Zeinab Fawaz, Hassan Noura, and Ahmed Mostefaoui. An efficient and secure cipher scheme for images confidentiality preservation. *Signal Processing: Image Communication*, 42:90–108, 2016.
- [132] Yanqiong Fei and Chengyuan Wang. Self-repairing algorithm of lattice-type self-reconfigurable modular robots. *Journal of Intelligent & Robotic Systems*, 75:193–203, 2014.
- [133] Eduardo Castelló Ferrer, Thomas Hardjono, Alex Pentland, and Marco Dorigo. Secure and secret cooperation in robot swarms. *Science Robotics*, 6(56):eabf1538, 2021.
- [134] Brenda Fiegel, Geoffrey Demarest, John P Sullivan, Robert Bunker, Alexandra Phelan, Thomas E Ayres, Juan-Camilo Castillo, Byron Ramirez, Max G Manwaring, Peter F Schaefer, et al. Narco-drones: a new way to transport drugs. *Criminal Drone Evolution: Cartel Weaponization of Aerial leds*, page 7, 2021.
- [135] Robert Fitch and Zack Butler. Scalable locomotion for large self-reconfiguring robots. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2248–2253. IEEE, 2007.
- [136] Robert Fitch, Zack Butler, and Daniela Rus. Reconfiguration planning for heterogeneous self-reconfiguring robots. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2460–2467. IEEE, 2003.
- [137] Robert Fitch, Zack Butler, and Daniela Rus. Reconfiguration planning among obstacles for heterogeneous self-reconfiguring robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 117–124. IEEE, 2005.

- [138] Robert Fitch, Zack Butler, and Daniela Rus. In-place distributed heterogeneous reconfiguration planning. In *Distributed Autonomous Robotic Systems 6*, pages 159–168. Springer, 2007.
- [139] Robert Fitch and Rowan McAllister. Hierarchical planning for self-reconfiguring robots using module kinematics. In *Distributed Autonomous Robotic Systems*, pages 477–490. Springer, 2013.
- [140] Robert Fitch and Daniela Rus. Self-reconfiguring robots in the usa. *Journal of the Robotics Society of Japan*, 21(8):832–838, 2003.
- [141] Robert Fitch, Kasper Stoy, Serge Kernbach, Radhika Nagpal, and Wei-Min Shen. Reconfigurable modular robotics. *Robotics and Autonomous Systems*, 7(62):943–944, 2014.
- [142] Warwick Ford and Michael S Baum. *Secure electronic commerce: building the infrastructure for digital signatures and encryption*. Prentice-Hall, Inc., 1997.
- [143] Jan Fras, Yohan Noh, Mateusz Macias, Helge Wurdemann, and Kaspar Althoefer. Bio-inspired octopus robot based on novel soft fluidic actuator. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1583–1588. IEEE, 2018.
- [144] Matas Führer, Roland Heinrich, Abdelwadoud Mabrouk, Tobias Christian Piller, Abdelmajid Khelil, and Kubilay Yildiz. Digital lab for basic and advanced features of message queuing telemetry transport (mqtt). In *Proceedings of the 8th International Conference on e-Society, e-Learning and e-Technologies*, pages 7–13, 2022.
- [145] Toshio Fukuda and Yoshio Kawauchi. Cellular robotic system (cebot) as one of the realization of self-organizing intelligent universal manipulator. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 662–667. IEEE, 1990.
- [146] David Furcy, Samuel Micka, and Scott M Summers. Optimal program-size complexity for self-assembly at temperature 1 in 3d. In *International Workshop on DNA-Based Computers*, pages 71–86. Springer, 2015.
- [147] Bruno Gabrich, Guanrui Li, and Mark Yim. Modquad-dof: A novel yaw actuation for modular quadrotors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8267–8273. IEEE, 2020.
- [148] Nikhil B Gaikwad, Hrishikesh Ugale, Avinash Keskar, and NC Shivaprakash. The internet of battlefield things (iobt) based enemy localization using soldiers location and gunshot direction. *IEEE Internet of Things Journal*, 2020.

- [149] Akhil Hannegudda Ganesh and Bin Xu. A review of reinforcement learning based energy management systems for electrified powertrains: Progress, challenge, and potential solution. *Renewable and Sustainable Energy Reviews*, 154:111833, 2022.
- [150] Qing Gao, Jinguo Liu, and Zhaojie Ju. Robust real-time hand detection and localization for space human–robot interaction based on deep learning. *Neurocomputing*, 390:198–206, 2020.
- [151] Carlos A Garcia, William Montalvo-Lopez, and Marcelo V Garcia. Human-robot collaboration based on cyber-physical production system and mqtt. *Procedia manufacturing*, 42:315–321, 2020.
- [152] Pablo Garcia-Aunon, Jaime Del Cerro, and Antonio Barrientos. Behavior-based control for an aerial robotic swarm in surveillance missions. *Sensors*, 19(20):4584, 2019.
- [153] A René Geist, Jonathan Fiene, Naomi Tashiro, Zheng Jia, and Sebastian Trimpe. The wheelbot: A jumping reaction wheel unicycle. *IEEE Robotics and Automation Letters*, 7(4):9683–9690, 2022.
- [154] Michael Gerbl and Johannes Gerstmayr. Self-reconfiguration of shape-shifting modular robots with triangular structure. *Robotics and Autonomous Systems*, 147:103930, 2022.
- [155] Neil Gershenfeld, Raffi Krikorian, and Danny Cohen. The internet of things. *Scientific American*, 291(4):76–81, 2004.
- [156] Dan Gettinger and Arthur Holland Michel. Law enforcement robots datasheet. *Center for the Study of the Drone, Bard College*, pages 55–56, 2016.
- [157] Maani Ghaffari Jadidi, Jaime Valls Miro, and Gamini Dissanayake. Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring. *The International Journal of Robotics Research*, 38(6):658–685, 2019.
- [158] David M Giles, Alexander Sinyuk, Mikhail G Sorokin, Joel S Schafer, Alexander Smirnov, Ilya Slutsker, Thomas F Eck, Brent N Holben, Jasper R Lewis, James R Campbell, et al. Advancements in the aerosol robotic network (aeronet) version 3 database—automated near-real-time quality control algorithm with improved cloud screening for sun photometer aerosol optical depth (aod) measurements. *Atmospheric Measurement Techniques*, 12(1):169–209, 2019.
- [159] Kyle Gilpin, Ara Knaian, and Daniela Rus. Robot pebbles: One centimeter modules for programmable matter through self-disassembly. In *2010 IEEE International Conference on Robotics and Automation*, pages 2485–2492. IEEE, 2010.

- [160] Kyle Gilpin, Keith Kotay, Daniela Rus, and Iuliu Vasilescu. Mische: Modular shape formation by self-disassembly. *The International Journal of Robotics Research*, 27(3-4):345–372, 2008.
- [161] R Gmyr, I Kostitsyna, F Kuhn, C Scheideler, and T Strothmann. Forming tile shapes with a single robot. In *Abstr. European Workshop on Computational Geometry (EuroCG)*, pages 9–12, 2017.
- [162] Seth Copen Goldstein, Jason D Campbell, and Todd C Mowry. Programmable matter. *Computer*, 38(6):99–101, 2005.
- [163] Seth Copen Goldstein and Todd Mowry. Claytronics: A scalable basis for future robots. *Robosphere*, Nov, 2004.
- [164] Ivan Gonzalez, Sergio Lopez-Buedo, and Francisco J Gomez-Arribas. Implementation of secure applications in self-reconfigurable systems. *Microprocessors and Microsystems*, 32(1):23–32, 2008.
- [165] Anna Gorbenko and Vladimir Popov. Programming for modular reconfigurable robots. In *Proceedings of the Spring/Summer Young Researchers' Colloquium on Software Engineering*, number 5, 2011.
- [166] Robert Grabowski, Luis E Navarro-Serment, Christiaan JJ Paredis, and Pradeep K Khosla. Heterogeneous teams of modular robots for mapping and exploration. *Autonomous Robots*, 8:293–308, 2000.
- [167] Carl Greene, John Naglak, Casey Majhor, Jeremy P Bos, and Wayne W Weaver. Near field wireless power transfer via robotic feedback control. In *2020 IEEE Aerospace Conference*, pages 1–7. IEEE, 2020.
- [168] Felix Grimminger, Avadesh Meduri, Majid Khadiv, Julian Viereck, Manuel Wüthrich, Maximilien Naveau, Vincent Berenz, Steve Heim, Felix Widmaier, Thomas Flayols, et al. An open torque-controlled modular robot architecture for legged locomotion research. *IEEE Robotics and Automation Letters*, 5(2):3650–3657, 2020.
- [169] Lars Grimstad and PI Johan From. Thorvald ii-a modular and re-configurable agricultural robot. *IFAC-PapersOnLine*, 50(1):4588–4593, 2017.
- [170] Marta Grobelna. Behavior prediction of cyber-physical systems for dynamic risk assessment. In *European Dependable Computing Conference*, pages 30–38. Springer, 2021.
- [171] Agneev Guin. Programmable matter-claytronics. In *presented at the 58th international instrumentation symposium, San Diego, California*, pages 4–8, 2012.

- [172] Erico Guizzo. Hibot demos new amphibious snake robot. *IEEE Spectrum: Technology, Engineering, and Science News*, 2013.
- [173] Sarah Haas, Thomas Ulz, and Christian Steger. Secured action authorization for industrial mobile robots. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pages 806–811. IEEE, 2018.
- [174] Mohammed Sayeeduddin Habeeb and T Ranga Babu. Network intrusion detection system: A survey on artificial intelligence-based techniques. *Expert Systems*, page e13066, 2022.
- [175] Matthew F Hale, Mike Angus, Edgar Buchanan, Wei Li, Robert Woolley, Léni K Le Goff, Matteo De Carlo, Jon Timmis, Alan F Winfield, Emma Hart, et al. Hardware design for autonomous robot evolution. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2140–2147. IEEE, 2020.
- [176] Matthew D Hall, Anil Özdemir, and Roderich Groß. Self-reconfiguration in two-dimensions via active subtraction with modular robots. In *Robotics: Science and Systems*, 2020.
- [177] TX Hammes. Technology converges; non-state actors benefit. *American Defense Policy*, page 143, 2021.
- [178] Muhammad Haziq Hasbulah, Fairul Azni Jafar, and Mohd Hisham Nordin. Comprehensive review on modular self-reconfigurable robot architecture. *Int. Res. J. Eng. Technol*, 6(4):1317–1331, 2019.
- [179] Hvard Haugstvedt. A flying threat coming to sahel and east africa? a brief review. *Journal of Strategic Security*, 14(1):92–105, 2020.
- [180] Hvard Haugstvedt. A flying reign of terror? the who, where, when, what, and how of non-state actors and armed drones. *Journal of Human Security*, 19(1):1–7, 2023.
- [181] Simon Hauser, Mehmet Mutlu, and Auke J Ijspeert. Kubits: Solid-state self-reconfiguration with programmable magnets. *IEEE Robotics and Automation Letters*, 5(4):6443–6450, 2020.
- [182] Elliot W Hawkes, Laura H Blumenschein, Joseph D Greer, and Allison M Okamura. A soft robot that navigates its environment through growth. *Science Robotics*, 2(8):eaan3028, 2017.
- [183] Jun He and Feng Gao. Mechanism, actuation, perception, and control of highly dynamic multilegged robots: A review. *Chinese Journal of Mechanical Engineering*, 33(1):1–30, 2020.

- [184] Daniel Hert, Tomas Baca, Pavel Petracek, Vit Kratky, Robert Penicka, Vojtech Spurny, Matej Petrlik, Matous Vrba, David Zaitlik, Pavel Stoudek, et al. Mrs drone: A modular platform for real-world deployment of aerial multi-robot systems. *Journal of Intelligent & Robotic Systems*, 108(4):64, 2023.
- [185] Hanne Heszlein-Lossius, Yahya Al-Borno, Samar Shaqqoura, Nashwa Skaik, Lasse Melvaer Giil, and Mads F Gilbert. Traumatic amputations caused by drone attacks in the local population in gaza: a retrospective cross-sectional study. *The Lancet Planetary Health*, 3(1):e40–e47, 2019.
- [186] Daniel C Hinck, Jonas J Schöttler, Maria Krantz, Katharina-Sophie Isleif, and Oliver Niggemann. A cross-frequency protective emblem: Protective options for medical units and wounded soldiers in the context of (fully) autonomous warfare. *arXiv preprint arXiv:2305.05459*, 2023.
- [187] Shigeo Hirose and Yoji Umetani. The development of soft gripper for the versatile robot hand. *Mechanism and machine theory*, 13(3):351–359, 1978.
- [188] Kevin Holdcroft, Anastasia Bolotnikova, Christoph Belke, and Jamie Paik. Modular robot networking: a novel schema and its performance assessment. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12698–12705. IEEE, 2022.
- [189] Kevin A Holdcroft, Christoph H Belke, Samir Bennani, and Jamie Paik. 3pac: A plug-and-play system for distributed power sharing and communication in modular robots. *IEEE/ASME Transactions on Mechatronics*, 27(2):858–867, 2021.
- [190] Paweł Hołobut, Michał Kurska, and Jakub Lengiewicz. A class of microstructures for scalable collective actuation of programmable matter. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3919–3925. IEEE, 2014.
- [191] Paweł Hołobut and Jakub Lengiewicz. Distributed computation of forces in modular-robotic ensembles as part of reconfiguration planning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2103–2109. IEEE, 2017.
- [192] Hansi Hong, Zifan Wen, Sheng Bi, Yue Zhang, and Wenxing Yang. Roveros: Linking ros with websocket for mobile robot. In *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 626–630. IEEE, 2019.
- [193] Roberto Horowitz, W-W Kao, Michael Boals, and Nader Sadegh. Digital implementation of repetitive controllers for robotic manipulators. In *1989 IEEE International Conference on Robotics and Automation*, pages 1497–1498. IEEE Computer Society, 1989.

- [194] Doreen Horschig. Cyber-weapons in nuclear counter-proliferation. *Defense & Security Analysis*, 36(3):352–371, 2020.
- [195] Feili Hou and Wei-Min Shen. On the complexity of optimal reconfiguration planning for modular reconfigurable robots. In *2010 IEEE International Conference on Robotics and Automation*, pages 2791–2796. IEEE, 2010.
- [196] Tie Hou and Victoria Wang. Industrial espionage—a systematic literature review (slr). *Computers & Security*, 98:102019, 2020.
- [197] Edy Hourany, Bachir Habib, Camille Fontaine, Abdallah Makhoul, Benoit Piranda, and Julien Bourgeois. Prolisean: A new security protocol for programmable matter. *ACM Transactions on Internet Technology (TOIT)*, 21(1):1–29, 2021.
- [198] Edy Hourany, Bachir Habib, Abdallah Makhoul, Benoit Piranda, Julien Bourgeois, and Pierre-Cyrille Heam. Elector: Deterministic leader election algorithm for modular robots. In *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/Private/Comp/Meta)*, pages 1551–1559. IEEE, 2022.
- [199] Jen-Hsuan Hsiao, Jen-Yuan Chang, and Chao-Min Cheng. Soft medical robotics: clinical and biomedical applications, challenges, and future directions. *Advanced Robotics*, 33(21):1099–1111, 2019.
- [200] Junyan Hu, Ali Emre Turgut, Tomáš Krajník, Barry Lennox, and Farshad Arvin. Occlusion-based coordination protocol design for autonomous robotic shepherding tasks. *IEEE Transactions on Cognitive and Developmental Systems*, 14(1):126–135, 2020.
- [201] Jiang Hua, Liangcai Zeng, Gongfa Li, and Zhaojie Ju. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. *Sensors*, 21(4):1278, 2021.
- [202] H-W Huang, Fazil Emre Uslu, Panayiota Katsamba, Eric Lauga, Mahmut S Sakar, and Bradley J Nelson. Adaptive locomotion of artificial microswimmers. *Science advances*, 5(1):eaau1532, 2019.
- [203] Weiguang Huo, Samer Mohammed, Juan C Moreno, and Yacine Amirat. Lower limb wearable robots for assistance and rehabilitation: A state of the art. *IEEE systems Journal*, 10(3):1068–1081, 2014.
- [204] Jules Hurst. Robotic swarms in offensive maneuver. *Joint Force Quarterly*, 87(4):105–11, 2017.

- [205] Ron Iphofen and Mihalís Kritikos. Regulating artificial intelligence and robotics: ethics by design in a digital society. *Contemporary Social Science*, 16(2):170–184, 2021.
- [206] Gorka Irazoqui, Thomas Eisenbarth, and Berk Sunar. Cross processor cache attacks. In *Proceedings of the 11th ACM on Asia conference on computer and communications security*, pages 353–364, 2016.
- [207] Liaoliang Jiang, Tong Li, Xuan Li, Mohammed Atiquzzaman, Haseeb Ahmad, and Xianmin Wang. Anonymous communication via anonymous identity-based encryption and its application in iot. *Wireless Communications and Mobile Computing*, 2018, 2018.
- [208] Hu Jin, Erbao Dong, Gursel Alici, Shixin Mao, Xu Min, Chunshan Liu, KH Low, and Jie Yang. A starfish robot based on soft and smart modular structure (sms) actuated by sma wires. *Bioinspiration & biomimetics*, 11(5):056012, 2016.
- [209] Craig Jones. Asteroid mining and the enclosure of outer space: New space economy discourses and ethnofuturist critique. 2022.
- [210] Michael Jones and Matthew Joordens. Design of an angular radial robotic stingray. In *2014 World Automation Congress (WAC)*, pages 234–239. IEEE, 2014.
- [211] Gregory Kahn, Pieter Abbeel, and Sergey Levine. Badgr: An autonomous self-supervised learning-based navigation system. *IEEE Robotics and Automation Letters*, 6(2):1312–1319, 2021.
- [212] M Shamim Kaiser, Shamim Al Mamun, Mufti Mahmud, and Marzia Hoque Tania. Healthcare robots to combat covid-19. In *COVID-19: Prediction, Decision-Making, and Its Impacts*, pages 83–97. Springer, 2021.
- [213] Manivannan Kalimuthu, Thejus Pathmakumar, Abdullah Aamir Hayat, Prabakaran Veerajagadheswar, Mohan Rajesh Elara, and Kristin Lee Wood. Optimal morphologies of n-omino-based reconfigurable robot for area coverage task using meta-heuristic optimization. *Mathematics*, 11(4):948, 2023.
- [214] Zachary Kallenborn. Infoswarms: Drone swarms and information warfare. *The US Army War College Quarterly: Parameters*, 52(2):87–102, 2022.
- [215] Mariusz A Kamiński. Operation “olympic games.” cyber-sabotage as a tool of american intelligence aimed at counteracting the development of iran’s nuclear programme. *Security and Defence Quarterly*, 29(2):63–71, 2020.
- [216] Vemema Kangunde, Rodrigo S Jamisola, and Emmanuel K Theophilus. A review on drones controlled in real-time. *International journal of dynamics and control*, 9(4):1832–1846, 2021.

- [217] Leyli Karaçay, Zeki Bilgin, Ayşe Bilge Gündüz, Pinar Çomak, Emrah Tomur, Elif Usundag Soykan, Utku Gülen, and Ferhat Karakoç. A network-based positioning method to locate false base stations. *IEEE Access*, 9:111368–111382, 2021.
- [218] Somnath Karmakar, Jayasree Sengupta, and Sipra Das Bit. Leader: low overhead rank attack detection for securing rpl based iot. In *2021 International Conference on COMMunication Systems & NETWORKS (COMSNETS)*, pages 429–437. IEEE, 2021.
- [219] Karthik Karur, Nitin Sharma, Chinmay Dharmatti, and Joshua E Siegel. A survey of path planning algorithms for mobile robots. *Vehicles*, 3(3):448–468, 2021.
- [220] Hiroshi Kawano. Complete reconfiguration algorithm for sliding cube-shaped modular robots with only sliding motion primitive. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3276–3283. IEEE, 2015.
- [221] Hiroshi Kawano. Full-resolution reconfiguration planning for heterogeneous cube-shaped modular robots with only sliding motion primitive. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5222–5229. IEEE, 2016.
- [222] Hiroshi Kawano. Tunneling-based self-reconfiguration of heterogeneous sliding cube-shaped modular robots in environments with obstacles. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 825–832. IEEE, 2017.
- [223] Hiroshi Kawano. Distributed tunneling reconfiguration of sliding cubic modular robots in severe space requirements. In *Distributed Autonomous Robotic Systems*, pages 1–15. Springer, 2019.
- [224] Ben Kehoe, Sachin Patil, Pieter Abbeel, and Ken Goldberg. A survey of research on cloud robotics and automation. *IEEE Transactions on automation science and engineering*, 12(2):398–409, 2015.
- [225] Krzysztof Kepa, Fearghal Morgan, Krzysztof Kosciuszkiewicz, and Tomasz Surmacz. Serecon: a secure reconfiguration controller for self-reconfigurable systems. *International Journal of Critical Computer-Based Systems*, 1(1-3):86–103, 2010.
- [226] Eliahu Khalastchi and Meir Kalech. Fault detection and diagnosis in multi-robot systems: A survey. *Sensors*, 19(18):4019, 2019.
- [227] Sampada A Khorgade and D Ghuse Namrata. Attacks and preventions in wireless sensor network. *International Journal of Engineering Research and General Science*, 3(2 Part 2), 2015.

- [228] Isabelle Khurshudyan and Kamila Hrabchuk. Ukrainians, crossing dnier river, test russian lines on southern front. *The Washington Post*, pages NA–NA, 2023.
- [229] Daryl G Kimball. The nuclear taboo remains strong for now. *Arms Control Today*, 53(1):3–3, 2023.
- [230] Brian T Kirby, Michael Ashley-Rollman, and Seth Copen Goldstein. Blinky blocks: a physical ensemble programming platform. In *CHI'11 extended abstracts on human factors in computing systems*, pages 1111–1116. 2011.
- [231] Brian T. Kirby, Michael Ashley-Rollman, and Seth Copen Goldstein. Blinky blocks: a physical ensemble programming platform. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pages 1111–1116, New York, NY, USA, 2011. ACM.
- [232] Ara N Knaian, Kenneth C Cheung, Maxim B Lobovsky, Asa J Oines, Peter Schmidt-Neilsen, and Neil A Gershenfeld. The milli-motein: A self-folding chain of programmable matter with a one centimeter module pitch. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1447–1453. IEEE, 2012.
- [233] W Knight. Russia's killer drone in ukraine raises fears about ai in warfare the maker of the lethal drone claims that it can identify targets using artificial intelligence.', 2022.
- [234] Thadeu Knychala Tucci, Benoit Piranda, and Julien Bourgeois. Efficient scene encoding for programmable matter self-reconfiguration algorithms. In *32nd Annual Symposium on Applied Computing (SAC 2017)*, volume Morocco, Marrakesh of *ACM International Conference Proceedings*, pages 256 – 261, Marrakech, Morocco, apr 2017.
- [235] C Korpela and Jeffrey L Caton. Swarms in the third offset. ed. *White, SR, Closer than you think: The implications of the Third Offset Strategy for the US Army, US Army Command and General Staff College, Fort Leavenworth, KS, US. Liang, Q & Xiangsui, W*, 2015.
- [236] Keith D Kotay and Daniela L Rus. Algorithms for self-reconfiguring molecule motion planning. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, volume 3, pages 2184–2193. IEEE, 2000.
- [237] Lucy van der Kroft. Yemen's houthis and the terrorist designation system. 2021.

- [238] Katharina Krombholz, Heidelinde Hobel, Markus Huber, and Edgar Weippl. Advanced social engineering attacks. *Journal of Information Security and applications*, 22:113–122, 2015.
- [239] Shivesh Kumar, Hendrik Wöhrle, José de Gea Fernández, Andreas Müller, and Frank Kirchner. A survey on modularity and distributivity in series-parallel hybrid robots. *Mechatronics*, 68:102367, 2020.
- [240] Dominika Kunertova. The ukraine drone effect on european militaries. 2022.
- [241] Dominika Kunertova. The war in ukraine shows the game-changing effect of drones depends on the game. *Bulletin of the Atomic Scientists*, 79(2):95–102, 2023.
- [242] Giovanni Lacava, Angelica Marotta, Fabio Martinelli, Andrea Saracino, Antonio La Marra, Endika Gil-Uriarte, and Victor Mayoral Vilches. Cybsersecurity issues in robotics. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, 12(3):1–28, 2021.
- [243] Henning Lahmann. The future digital battlefield and challenges for humanitarian protection: A primer. *Available at SSRN 4088521*, 2022.
- [244] Cecilia Laschi, Matteo Cianchetti, Barbara Mazzolai, Laura Margheri, Maurizio Follador, and Paolo Dario. Soft robot arm inspired by the octopus. *Advanced robotics*, 26(7):709–727, 2012.
- [245] Jun-Young Lee, Jaemin Eom, Woo-Young Choi, and Kyu-Jin Cho. Soft lego: bottom-up design platform for soft robotics. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7513–7520. IEEE, 2018.
- [246] Jakub Lengiewicz and Paweł Hołobut. Efficient collective shape shifting and locomotion of massively-modular robotic structures. *Autonomous Robots*, 43(1):97–122, 2019.
- [247] Samuel Lensgraf, Amy Sniffen, Zachary Zitzewitz, Evan Honnold, Jennifer Jain, Weifu Wang, Alberto Li, and Devin Balkcom. Droplet: Towards autonomous underwater assembly of modular structures. In *Proceedings of Robotics: Science and Systems*, 2021.
- [248] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 58:52–68, 2020.
- [249] Haiyuan Li, Haoyu Wang, Linlin Cui, Jiake Li, Qi Wei, and Jiqiang Xia. Design and experiments of a compact self-assembling mobile modular robot with joint actuation and onboard visual-based perception. *Applied Sciences*, 12(6):3050, 2022.

- [250] Qingbiao Li, Fernando Gama, Alejandro Ribeiro, and Amanda Prorok. Graph neural networks for decentralized multi-robot path planning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11785–11792. IEEE, 2020.
- [251] Shuguang Li and Daniela Rus. Jellocube: a continuously jumping robot with soft body. *IEEE/ASME Transactions on Mechatronics*, 24(2):447–458, 2019.
- [252] Xiao Li, Chengke Wu, Fan Xue, Zhile Yang, Jinfeng Lou, and Weisheng Lu. Ontology-based mapping approach for automatic work packaging in modular construction. *Automation in Construction*, 134:104083, 2022.
- [253] Guanqi Liang, Haobo Luo, Ming Li, Huihuan Qian, and Tin Lun Lam. Freebot: A freeform modular self-reconfigurable robot with arbitrary connection point-design and implementation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6506–6513. IEEE, 2020.
- [254] JieJunYi Liang, QinHao Zhang, Yang Liu, Tao Wang, and GuangFu Wan. A review of the design of load-carrying exoskeletons. *Science China Technological Sciences*, pages 1–17, 2022.
- [255] Chao Liu, Qian Lin, Hyun Kim, and Mark Yim. Smores-ep, a modular robot with parallel self-assembly. *arXiv preprint arXiv:2104.00800*, 2021.
- [256] Dan Liu, Bifeng Song, Wenqing Yang, Xiaojun Yang, Dong Xue, and Xinyu Lang. A brief review on aerodynamic performance of wingtip slots and research prospect. *Journal of Bionic Engineering*, pages 1–25, 2021.
- [257] Shaoshan Liu. *Engineering autonomous vehicles and robots: the DragonFly modular-based approach*. John Wiley & Sons, 2020.
- [258] Xinzhong Liu, Cong Xie, Wenwu Xie, Peng Zhu, and Zhihe Yang. Security performance analysis of ris-assisted uav wireless communication in industrial iot. *The Journal of Supercomputing*, 78(4):5957–5973, 2022.
- [259] Yueyue Liu, Zhijun Li, Huaping Liu, and Zhen Kan. Skill transfer learning for autonomous robots and human–robot cooperation: A survey. *Robotics and Autonomous Systems*, 128:103515, 2020.
- [260] Zhe Liu, Weidong Chen, Hesheng Wang, Yun-Hui Liu, Yi Shen, and Xiangyu Fu. A self-repairing algorithm with optimal repair path for maintaining motion synchronization of mobile robot network. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(3):815–828, 2017.

- [261] Bei Long, Xiaojie Liang, Yong Pei, Xiongwei Wu, Xianyou Wang, and Man-Kay Law. Free-standing bioi@ mwcnts photoelectrodes for photo-rechargeable zinc-ion batteries. *Journal of Materials Science & Technology*, 2024.
- [262] Anthony Bahadir Lopez, Korosh Vatanparvar, Atul Prasad Deb Nath, Shuo Yang, Swarup Bhunia, and Mohammad Abdullah Al Faruque. A security perspective on battery systems of the internet of things. *Journal of Hardware and Systems Security*, 1(2):188–199, 2017.
- [263] Antonio López-Díaz, Jesús De La Morena, Francisco Ramos, Ester Vázquez, and Andrés S Vázquez. A novel hydrogel-based connection mechanism for soft modular robots. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7124–7130. IEEE, 2022.
- [264] AdAm Lowther and MAHBUBE K SIDDIKI. Combat drones in ukraine. *Air & Space Operations Review*, 1(4), 2022.
- [265] Chengyi Lu, Guang Pan, Zhaoyong Mao, Liangwei Shi, Qiaogao Huang, Wenlong Tian, Yuli Hu, Haitao Wu, Zhenhua Wang, and Kening Sun. Multiradical-stabilized hollow carbon spheres as a pressure-resistant cathode for fast lithium/sodium storage with excellent performance. *Journal of Materials Chemistry A*, 8(18):8875–8882, 2020.
- [266] Ming Luo, Erik H Skorina, Weijia Tao, Fuchen Chen, Selim Ozel, Yinan Sun, and Cagdas D Onal. Toward modular soft robotics: Proprioceptive curvature sensing and sliding-mode control of soft bidirectional bending modules. *Soft robotics*, 4(2):117–125, 2017.
- [267] Emanuele Magrini, Federica Ferraguti, Andrea Jacopo Ronga, Fabio Pini, Alessandro De Luca, and Francesco Leali. Human-robot coexistence and interaction in open industrial cells. *Robotics and Computer-Integrated Manufacturing*, 61:101846, 2020.
- [268] Mahapara Mahak and Yashwant Singh. Threat modelling and risk assessment in internet of things: A review. In *Proceedings of Second International Conference on Computing, Communications, and Cyber-Security*, pages 293–305. Springer, 2021.
- [269] Marco M Maia, Diego A Mercado, and F Javier Diez. Design and implementation of multicopter aerial-underwater vehicles with experimental results. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 961–966. IEEE, 2017.
- [270] Aliah Majed, Hassan Harb, Abbass Nasser, and Benoit Clement. Run: a robust cluster-based planning for fast self-reconfigurable modular robotic systems. *Intelligent Service Robotics*, pages 1–11, 2023.

- [271] Nader A Mansour, Taesoo Jang, Hangeol Baek, Buhyun Shin, Bongjo Ryu, and Youngshik Kim. Compliant closed-chain rolling robot using modular unidirectional sma actuators. *Sensors and Actuators A: Physical*, 310:112024, 2020.
- [272] David March, Julia Múgica, Ezequiel E Ferrero, and M Carmen Miguel. Honeybee-like collective decision making in a kilobot swarm. *arXiv preprint arXiv:2310.15592*, 2023.
- [273] Garik Markarian and Andrew Staniforth. *Countermeasures for Aerial Drones*. Artech House, 2020.
- [274] Mónica Martí, Carlos Garcia-Rubio, and Celeste Campo. Performance evaluation of coap and mqtt.sn in an iot environment. *Multidisciplinary Digital Publishing Institute Proceedings*, 31(1):49, 2019.
- [275] Nithin Mathews, Anders Lyhne Christensen, Rehan O’Grady, and Marco Dorigo. Spatially targeted communication and self-assembly. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2678–2679. IEEE, 2012.
- [276] Leigh-Anne Mathieson and Anne Condon. On low energy barrier folding pathways for nucleic acid sequences. In *International Workshop on DNA-Based Computers*, pages 181–193. Springer, 2015.
- [277] Constantinos Mavroidis, Charles Pfeiffer, and Michael Mosley. 5.1 conventional actuators, shape memory alloys, and electrorheological fluids. *Automation, miniature robotics, and sensors for nondestructive testing and evaluation*, 4(3):189, 2000.
- [278] Ross M McKenzie, Mohammed E Sayed, Markus P Nemitz, Brian W Flynn, and Adam A Stokes. Linbots: Soft modular robots utilizing voice coils. *Soft robotics*, 6(2):195–205, 2019.
- [279] Nir Meiri and David Zarrouk. Flying star, a hybrid crawling and flying sprawl tuned robot. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5302–5308. IEEE, 2019.
- [280] D Mercado, M Maia, and F Javier Diez. Aerial-underwater systems, a new paradigm in unmanned vehicles. *Journal of Intelligent & Robotic Systems*, 95(1):229–238, 2019.
- [281] Justin Miller, Andrew B Williams, and Debbie Perouli. A case study on the cybersecurity of social robots. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 195–196. ACM, 2018.

- [282] Kei Misumi, Gwenn Ulliac, Naoto Usami, Benoit Piranda, Yoshio Mita, Akio Higo, and Julien Bourgeois. Micro-scale electrostatic attach-detach device for self-reconfigurable modular robotic system. In *2020 Symposium on Design, Test, Integration & Packaging of MEMS and MOEMS (DTIP)*, pages 1–4. IEEE, 2020.
- [283] Kei Misumi, Naoto Usami, Akio Higo, Gwenn Ulliac, Benoit Piranda, Julien Bourgeois, and Yoshio Mita. Integration of a cmos lsi chiplet into micro flexible devices for remote electrostatic actuation. In *2022 Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS (DTIP)*, pages 1–4, 2022.
- [284] Sebastian Mobes, Guillaume J Laurent, Cedric Clevy, Nadine Le Fort-Piat, Benoit Piranda, and Julien Bourgeois. Toward a 2d modular and self-reconfigurable robot for conveying microparts. In *2012 Second Workshop on Design, Control and Software Implementation for Distributed MEMS*, pages 7–13. IEEE, 2012.
- [285] Negar Mofakham Naser Eslami, Ahmad Momeni Rad, and Seyed Ahmad Tabatabai. Military use and arms race in space from the perspective of international law. *Journal of Space Science and Technology*, 2022.
- [286] Saeed Reza Mohandes, Sherif Abdelmageed, Sakda Hem, Joo Sang Yoo, Tharindu Abhayajeewa, and Tarek Zayed. Occupational health and safety in modular integrated construction projects: The case of crane operations. *Journal of Cleaner Production*, page 130950, 2022.
- [287] Francesco Mondada, Michael Bonani, Stéphane Magnenat, André Guignard, Dario Floreano, Frans Groen, Nancy Amato, Andrea Bonari, Eiichi Yoshida, and Ben Kröse. Physical connections and cooperation in swarm robotics. In *8th Conference on Intelligent Autonomous Systems (IAS8)*, number CONF, pages 53–60, 2004.
- [288] Rodrigo Moreno and Andres Faiña. Emerge modular robot: a tool for fast deployment of evolved robots. *Frontiers in Robotics and AI*, page 198, 2021.
- [289] Matthew S Moses and Gregory S Chirikjian. Robotic self-replication. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:1–24, 2020.
- [290] Ahmed Mostefaoui and Benoît Piranda. Multimedia sensor networks: an approach based on 3d real-time reconstruction. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, pages 188–195, 2009.
- [291] Ali Mousaei, Enayatollah Yazdani, and Mohammadali Basiri. Russia’s balancing acts in the 2014 ukraine crisis. *Research Letter of International Relations*, 16(1), 2023.

- [292] Yash Mulgaonkar, Anurag Makineni, Luis Guerrero-Bonilla, and Vijay Kumar. Robust aerial robot swarms without collision avoidance. *IEEE Robotics and Automation Letters*, 3(1):596–603, 2017.
- [293] M Muralidharan and IA Palani. Development of subcarangiform bionic robotic fish propelled by shape memory alloy actuators. *Defence Science Journal*, 71(1), 2021.
- [294] Satoshi Murata and Haruhisa Kurokawa. *Self-organizing robots*, volume 77. Springer, 2012.
- [295] Robin R Murphy, Vignesh Babu Manjunath Gandudi, and Justin Adams. Applications of robots for covid-19 response. *arXiv preprint arXiv:2008.06976*, 2020.
- [296] Kohei Nakajima, Helmut Hauser, Rongjie Kang, Emanuele Guglielmino, Darwin G Caldwell, and Rolf Pfeifer. A soft body as a reservoir: case studies in a dynamic model of octopus-inspired soft robotic arm. *Frontiers in computational neuroscience*, 7:91, 2013.
- [297] Renzo E Navas, Hélène Le Boudier, Nora Cuppens, Frédéric Cuppens, and Georgios Z Papadopoulos. Do not trust your neighbors! a small iot platform illustrating a man-in-the-middle attack. In *International Conference on Ad-Hoc Networks and Wireless*, pages 120–125. Springer, 2018.
- [298] André Naz, Benoît Piranda, Julien Bourgeois, and Seth Copen Goldstein. A distributed self-reconfiguration algorithm for cylindrical lattice-based modular robots. In *2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*, pages 254–263. IEEE, 2016.
- [299] André Naz, Benoit Piranda, Julien Bourgeois, and Seth Copen Goldstein. Electing an approximate center in a huge modular robot with the k-bfs sumsweep algorithm. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4825–4832. IEEE, 2018.
- [300] Andre Naz, Benoit Piranda, Julien Bourgeois, and Seth Copen Goldstein. A time synchronization protocol for large-scale distributed embedded systems with low-precision clocks and neighbor-to-neighbor communications. *Journal of Network and Computer Applications*, 105:123–142, 2018.
- [301] André Naz, Benoit Piranda, Seth Copen Goldstein, and Julien Bourgeois. Abc-center: Approximate-center election in modular robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2951–2957. IEEE, 2015.
- [302] André Naz, Benoît Piranda, Seth Copen Goldstein, and Julien Bourgeois. A time synchronization protocol for modular robots. In *2016 24th Euromicro International*

- Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, pages 109–118. IEEE, 2016.
- [303] André Naz, Benoît Piranda, Thadeu Tucci, Seth Copen Goldstein, and Julien Bourgeois. Network characterization of lattice-based modular robots with neighbor-to-neighbor communications. In *Distributed Autonomous Robotic Systems*, pages 415–429. Springer, 2018.
- [304] PJ Neal and Second Place Winner. From unique needs to modular platforms: The future of military robotics. US Naval Institute, 2010.
- [305] Nadia Nedjah, Luigi Maciel Ribeiro, and Luiza de Macedo Mourelle. Communication optimization for efficient dynamic task allocation in swarm robotics. *Applied Soft Computing*, 105:107297, 2021.
- [306] Gabe Nelson, Aaron Saunders, Neil Neville, Ben Swilling, Joe Bondaryk, Devin Billings, Chris Lee, Robert Playter, and Marc Raibert. Petman: A humanoid robot for testing chemical protective clothing. *Journal of the Robotics Society of Japan*, 30(4):372–377, 2012.
- [307] Gabe Nelson, Aaron Saunders, and Robert Playter. The petman and atlas robots at boston dynamics. *Humanoid Robotics: A Reference*, 169:186, 2019.
- [308] Hoa G Nguyen and John P Bott. Robotics for law enforcement: Applications beyond explosive ordnance disposal. In *Enabling Technologies for Law Enforcement and Security*, volume 4232, pages 433–454. International Society for Optics and Photonics, 2001.
- [309] Luong A Nguyen, Thomas L Harman, and Carol Fairchild. Swarmathon: a swarm robotics experiment for future space exploration. In *2019 IEEE International Symposium on Measurement and Control in Robotics (ISMCR)*, pages B1–3. IEEE, 2019.
- [310] Martin Nisser, Dario Izzo, and Andreas Borggraefe. An electromagnetically actuated, self-reconfigurable space structure. *Transactions of the Japan Society for aeronautical and space sciences*, 14:1–9, 2017.
- [311] Hassan Noura, Ali Chehab, Lama Sleem, Mohamad Noura, Raphaël Couturier, and Mohammad M Mansour. One round cipher algorithm for multimedia iot devices. *Multimedia tools and applications*, 77:18383–18413, 2018.
- [312] Hassan Noura and Damien Couroussé. Lightweight, dynamic, and flexible cipher scheme for wireless and mobile networks. In *Ad Hoc Networks: 7th International Conference, AdHocHets 2015, San Remo, Italy, September 1-2, 2015. Proceedings 7*, pages 225–236. Springer, 2015.

- [313] Hassan Noura, Tarif Hatoum, Ola Salman, Jean-Paul Yaacoub, and Ali Chehab. Lo-rawan security survey: Issues, threats and possible mitigation techniques. *Internet of Things*, page 100303, 2020.
- [314] Hassan N Noura, Ola Salman, Raphaël Couturier, and Ali Chehab. Novel one round message authentication scheme for constrained iot devices. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–17.
- [315] Hassan N Noura, Ola Salman, Raphaël Couturier, and Ali Chehab. A single-pass and one-round message authentication encryption for limited iot devices. *IEEE Internet of Things Journal*, 9(18):17885–17900, 2022.
- [316] Hassan N Noura, Lama Sleem, Mohamad Noura, Mohammad M. Mansour, Ali Chehab, and Raphaël Couturier. A new efficient lightweight and secure image cipher scheme. *Multimedia Tools and Applications*, Sep 2017.
- [317] Mohammad Noura, Hassan Noura, Ali Chehab, Mohammad M Mansour, Lama Sleem, and Raphaël Couturier. A dynamic approach for a lightweight and secure cipher for medical images. *Multimedia Tools and Applications*, 77:31397–31426, 2018.
- [318] JC Odirichukwu, PO Asagba, and FE Onuodu. Interoperable protocols of the internet of things and internet of robotic things: A review. *International Journal of Computing, Intelligence and Security Research*, 1(1):101–123, 2021.
- [319] Piotr Olszewski. The threat z enia k a related to the use of improvised l adunk 'o in explosives during the attack ó in terrorist-do 's benefits and recommendations. *Weapon Technique Problems*, 49, 2020.
- [320] Laurean-Georgel Oprean. Artillery and drone action issues in the war in ukraine. *Scientific Bulletin*, 28(1):73–78, 2023.
- [321] Esben Hallundbæk Østergaard, Kristian Kassow, Richard Beck, and Henrik Hautop Lund. Design of the atron lattice-based self-reconfigurable robot. *Autonomous robots*, 21:165–183, 2006.
- [322] Savan Oswal, Anjali Singh, and Kirthi Kumari. Deflate compression algorithm. *International Journal of Engineering Research and General Science*, 4(1):430–436, 2016.
- [323] Yasemin Ozkan-Aydin and Daniel I Goldman. Self-reconfigurable multilegged robot swarms collectively accomplish challenging terradynamic tasks. *Science Robotics*, 6(56):eabf1628, 2021.

- [324] Anita Brigitta Pál. A hadviselés, terrorizmus és drónhasználat komplex kapcsolata. *Biztonságtudományi Szemle*, 5(2):21–32, 2023.
- [325] Irene Parada, Vera Sacristán, and Rodrigo I Silveira. A new meta-module design for efficient reconfiguration of modular robots. *Autonomous robots*, 45(4):457–472, 2021.
- [326] Sukho Park, Kyoungrae Cha, and Jongoh Park. Development of biomedical micro-robot for intravascular therapy. *International Journal of Advanced Robotic Systems*, 7(1):1, 2010.
- [327] Lynne E Parker, Balajee Kannan, Xiaoquan Fu, and Yifan Tang. Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2488–2493. IEEE, 2003.
- [328] Christopher Parrott, Tony J Dodd, and Roderich Groß. Higen: A high-speed genderless mechanical connection mechanism with single-sided disconnect for self-reconfigurable modular robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3926–3932. IEEE, 2014.
- [329] Rizuwana Parween, MA Muthugala, Manuel V Heredia, Karthikeyan Elangovan, and Mohan Rajesh Elara. Collision avoidance and stability study of a self-reconfigurable drainage robot. *Sensors*, 21(11):3744, 2021.
- [330] Sanjay S Patel and Poojan N Patel. A brief review on nanorobotics applications in medicine and future prospects. *Asian Journal of Research in Pharmaceutical Science*, 13(1), 2023.
- [331] Madhav Patil, Tamer Abukhalil, and Tarek Sobh. Hardware architecture review of swarm robotics system: Self-reconfigurability, self-reassembly, and self-replication. *International Scholarly Research Notices*, 2013, 2013.
- [332] Matthew J Patitz. An introduction to tile-based self-assembly and a survey of recent results. *Natural Computing*, 13(2):195–224, 2014.
- [333] Henrik Paulsson. *Soldiers, militants, and small drones*. 2018.
- [334] Robert H Peck, Jon Timmis, and Andy M Tyrrell. Self-assembly and self-repair during motion with modular robots. *Electronics*, 11(10):1595, 2022.
- [335] Raul Pete Pedrozo. The black sea grain initiative: Russia’s strategic blunder or diplomatic coup? *International Law Studies*, 100(1):12, 2023.

- [336] Yimai Peng, Gordy Carichner, Yejoong Kim, Li-Yu Chen, Rémy Tribhout, Benoît Piranda, Julien Bourgeois, David Blaauw, and Dennis Sylvester. A high-voltage generator and multiplexer for electrostatic actuation in programmable matter. *IEEE Journal of Solid-State Circuits*, 58(4):915–928, 2023.
- [337] Eduardo R Perez-Guagnelli, Sarunas Nejus, Jian Yu, Shuhei Miyashita, YanQiang Liu, and Dana D Damian. Axially and radially expandable modular helical soft actuator for robotic implantables. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4297–4304. IEEE, 2018.
- [338] Florian Pescher, Benoit Piranda, Stéphane Delalande, and Julien Bourgeois. Surface approximation by molding a shape-memory polymer on a modular robot. In *International Symposium on Distributed Autonomous Robotic Systems*, 2018.
- [339] Florian Pescher, Benoît Piranda, Stephane Delalande, and Julien Bourgeois. Molding a shape-memory polymer with programmable matter. In *Distributed Autonomous Robotic Systems*, pages 65–78. Springer, 2019.
- [340] Florian Pescher, Benoit Piranda, Nils Napp, and Julien Bourgeois. Gapcod: A generic assembly planner by constrained disassembly. In *International Conference on Autonomous Agents and Multiagent Systems*, 2020.
- [341] Carlo Pinciroli, Vito Trianni, Rehan O’Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, et al. Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm intelligence*, 6(4):271–295, 2012.
- [342] Benoit Piranda. *Designing and Programming Lattice-Based Modular Robots for Creating Programmable Matter*. PhD thesis, nov 2020.
- [343] Benoit Piranda and Julien Bourgeois. A distributed algorithm for reconfiguration of lattice-based modular self-reconfigurable robots. In *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, pages 1–9. IEEE, 2016.
- [344] Benoit Piranda and Julien Bourgeois. Designing a quasi-spherical module for a huge modular robot to create programmable matter. *Autonomous Robots*, 42(8):1619–1633, 2018.
- [345] Benoit Piranda and Julien Bourgeois. Geometrical study of a quasi-spherical module for building programmable matter. In *Distributed Autonomous Robotic Systems*, pages 387–400. Springer, 2018.

- [346] Benoit Piranda and Julien Bourgeois. Datom: A deformable modular robot for building self-reconfigurable programmable matter. *arXiv preprint arXiv:2005.03402*, 2020.
- [347] Benoît Piranda, Paweł Chodkiewicz, Paweł Hołobut, Stéphane PA Bordas, Julien Bourgeois, and Jakub Lengiewicz. Distributed prediction of unsafe reconfiguration scenarios of modular robotic programmable matter. *IEEE Transactions on Robotics*, 37(6):2226–2233, 2021.
- [348] Benoit Piranda, Frederic Lassabe, and Julien Bourgeois. Disco: A multiagent 3d coordinate system for lattice based modular self-reconfigurable robots. In *IEEE International Conference on Robotics and Automation (ICRA 2023)*, London, England, may 2023.
- [349] Benoît Piranda, Guillaume J Laurent, Julien Bourgeois, Cédric Clévy, Sebastian Möbes, and Nadine Le Fort-Piat. A new concept of planar self-reconfigurable modular robot for conveying microparts. *Mechatronics*, 23(7):906–915, 2013.
- [350] Benoit Piranda, Mohammad Ali Nemer, Abdallah Makhoul, and Julien Bourgeois. Ai4pm: Distributed and intelligent programmable matter. In *revision at IEEE RSJ International Conference on Intelligent Robots and Systems (IROS 2024)*, NY, USA, oct 2024.
- [351] Doina Pisla, Iulia Andras, Alexandru Pusca, Corina Radu, Bogdan Gherman, Paul Tucan, Nicolae Crisan, Calin Vaida, and Nadim Al Hajjar. Design and functional analysis of a new parallel modular robotic system for single incision laparoscopic surgery. In *International Workshop on Medical and Service Robots*, pages 32–41. Springer, 2023.
- [352] Michail Ploumis. Ai weapon systems in future war operations; strategy, operations and tactics. *Comparative Strategy*, 41(1):1–18, 2022.
- [353] Andre Potenza, Andrey Kiselev, Alessandro Saffiotti, and Amy Loutfi. An open-source modular robotic system for telepresence and remote disinfection. *arXiv preprint arXiv:2102.01551*, 2021.
- [354] Ahalya Prabhakar. Communicating and modeling information through motion. 2020.
- [355] Laura Pruszko, Hongri Gu, Julien Bourgeois, Yann Laurillau, and Céline Coutrix. Modular tangible user interfaces: Impact of module shape and bonding strength on interaction. In *Proceedings of the Seventeenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 1–15, 2023.

- [356] Lennart Puck, Philip Keller, Tristan Schnell, Carsten Plasberg, Atanas Tanev, Georg Heppner, Arne Roennau, and Rüdiger Dillmann. Performance evaluation of real-time ros2 robotic control in a time-synchronized distributed network. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 1670–1676. IEEE, 2021.
- [357] Anies Hannawati Purnamadajaja and R Andrew Russell. Pheromone communication: implementation of necrophoric bee behaviour in a robot swarm. In *IEEE Conference on Robotics, Automation and Mechatronics, 2004.*, volume 2, pages 638–643. IEEE, 2004.
- [358] John Racette, Simon Lotero, Jeffrey Gordon, Chris Dinelli, Arvin Ebrahimkhanlou, Sihua Shao, Pedram Roghanchi, and Mostafa Hassanalian. Hybrid ugv and drone system for mine rescue assistance. In *AIAA AVIATION 2022 Forum*, page 3287, 2022.
- [359] Petar Radanliev, David Charles De Roure, Carsten Maple, Jason RC Nurse, Razvan Nicolescu, and Uchenna Ani. Cyber risk in iot systems. 2019.
- [360] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, and Rob Playter. Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes*, 41(2):10822–10825, 2008.
- [361] Don Rassler. Remotely piloted innovation: Terrorism, drones and supportive technology. Technical report, US Military Academy-Combating Terrorism Center West Point United States, 2016.
- [362] Uneeb Yaqub Rathore. Flexibility, scalability, and efficiency in next-generation digital signal processors. 2022.
- [363] V Lakshmi Rebbapragada. Distributed battle management for command and control. In *Proceedings of MILCOM'94*, pages 542–545. IEEE, 1994.
- [364] Sean Rivera and Radu State. Securing robots: An integrated approach for security challenges and monitoring for the robotic operating system (ros). In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 754–759. IEEE, 2021.
- [365] Ignacio Rodriguez, Rasmus S Mogensen, Allan Schjørring, Mohammad Razzaghpour, Roberto Maldonado, Gilberto Berardinelli, Ramoni Adeogun, Per H Christensen, Preben Mogensen, Ole Madsen, et al. 5g swarm production: Advanced industrial manufacturing concepts enabled by wireless automation. *IEEE Communications Magazine*, 59(1):48–54, 2021.

- [366] John Romanishin, James M Bern, and Daniela Rus. Self-reconfiguring robotic gantries powered by modular magnetic lead screws. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4225–4231. IEEE, 2022.
- [367] John W Romanishin, Kyle Gilpin, Sebastian Claici, and Daniela 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)*, pages 1925–1932. IEEE, 2015.
- [368] John W Romanishin, Kyle Gilpin, and Daniela Rus. M-blocks: Momentum-driven, magnetic modular robots. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4288–4295. IEEE, 2013.
- [369] John W Romanishin, John Mamish, and Daniela Rus. Decentralized control for 3d m-blocks for path following, line formation, and light gradient aggregation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4862–4868. IEEE, 2019.
- [370] Laura Romeo, Antonio Petitti, Roberto Marani, and Annalisa Milella. Internet of robotic things in smart domains: applications and challenges. *Sensors*, 20(12):3355, 2020.
- [371] Sudhir K Routray, KP Sharmila, Abhishek Javali, Aritri D Ghosh, and Sushanta Sarangi. An outlook of narrowband iot for industry 4.0. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 923–926. IEEE, 2020.
- [372] Michael Rubenstein, Christian Ahler, and Radhika Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE international conference on robotics and automation*, pages 3293–3298. IEEE, 2012.
- [373] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.
- [374] Michael Rubenstein and Radhika Nagpal. Kilobot: a robotic module for demonstrating behaviors in a large scale (units) collective. In *Proceedings of the IEEE 2010 international conference on robotics and automation workshop, modular robotics: state of the art*. Institute of Electrical and Electronics Engineers, 2010.
- [375] Daniela Rus, Zack Butler, Keith Kotay, and Margette Vona. Self-reconfiguring robots. *Communications of the ACM*, 45(3):39–45, 2002.
- [376] Daniela Rus and Margette Vona. A physical implementation of the self-reconfiguring crystalline robot. In *Proceedings 2000 ICRA. Millennium Conference. IEEE Inter-*

- national Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 1726–1733. IEEE, 2000.
- [377] Daniela Rus and Marsette Vona. Crystalline robots: Self-reconfiguration with compressible unit modules. *Autonomous Robots*, 10(1):107–124, 2001.
- [378] Stuart Russell. Ai weapons: Russia’s war in ukraine shows why the world must enact a ban. *Nature*, 614(7949):620–623, 2023.
- [379] Matteo Russo, Jorge Barrientos-Diez, and Dragos Axinte. A kinematic coupling mechanism with binary electromagnetic actuators for high-precision positioning. *IEEE/ASME Transactions on Mechatronics*, 27(2):892–903, 2021.
- [380] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics*, pages 10–20. Springer, 2004.
- [381] Fatima Salahdine and Naima Kaabouch. Social engineering attacks: A survey. *Future Internet*, 11(4):89, 2019.
- [382] David Saldana, Bruno Gabrich, Guanrui Li, Mark Yim, and Vijay Kumar. Modquad: The flying modular structure that self-assembles in midair. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 691–698. IEEE, 2018.
- [383] Mazeiar Salehie and Ladan Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM transactions on autonomous and adaptive systems (TAAS)*, 4(2):1–42, 2009.
- [384] Hadeel Salman. Examining defences to state-sponsored cyber-operations. 2021.
- [385] SM Bhagya P Samarakoon, MA Viraj J Muthugala, and Mohan Rajesh Elara. Toward obstacle-specific morphology for a reconfigurable tiling robot. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–13, 2021.
- [386] Javier Felipe Moncada Sánchez, Orlando García Hurtado, and Roberto Manuel Poveda Chaves. Economic drones in education. *Ilkogretim Online*, 20(6):1291–1298, 2021.
- [387] Andrew W Sanders. Drone swarms. Technical report, US Army School for Advanced Military Studies Fort Leavenworth United States, 2017.
- [388] Ash Wan Yaw Sang, Chee Gen Moo, SM Bhagya P. Samarakoon, MA Viraj J Muthugala, and Mohan Rajesh Elara. Design of a reconfigurable wall disinfection robot. *Sensors*, 21(18):6096, 2021.

- [389] M Sangeetha and K Srinivasan. Swarm robotics: a new framework of military robots. In *Journal of Physics: Conference Series*, volume 1717, page 012017. IOP Publishing, 2021.
- [390] Ian Sargeant and Allan Tomlinson. Review of potential attacks on robotic swarms. In *Proceedings of SAI Intelligent Systems Conference*, pages 628–646. Springer, 2016.
- [391] Sayani Sarkar and Nathan Johnson. A deep-learning, vision-based framework for testing swarm algorithms using inexpensive mini drones. In *Unmanned Systems Technology XXIV*, volume 12124, pages 93–100. SPIE, 2022.
- [392] Mohammed E Sayed, Jamie O Roberts, Ross M McKenzie, Simona Aracri, Anthony Buchoux, and Adam A Stokes. Limpet ii: A modular, untethered soft robot. *Soft Robotics*, 8(3):319–339, 2021.
- [393] Karen Scarfone, Derrick Dicoi, Matthew Sexton, Cyrus Tibbs, et al. Guide to securing legacy ieee 802.11 wireless networks. *NIST Special Publication*, 800:48, 2008.
- [394] Wolfram Schoor, Martin Förster, and Arne Radetzky. Realistic training simulations of explosive ordnance disposal & improvised explosive device disposal robots. In *IEEE 10th International Conference on Industrial Informatics*, pages 875–880. IEEE, 2012.
- [395] Melanie Schranz, Martina Umlauf, Micha Sende, and Wilfried Elmenreich. Swarm robotic behaviors and current applications. *Frontiers in Robotics and AI*, 7:36, 2020.
- [396] Francesco Semeraro, Alexander Griffiths, and Angelo Cangelosi. Human–robot collaboration and machine learning: A systematic review of recent research. *Robotics and Computer-Integrated Manufacturing*, 79:102432, 2023.
- [397] Alex Sendrós, Mahjoub Himi, Raúl Lovera, Luis Rivero, Ruben Garcia-Artigas, Aritz Urruela, and Albert Casas. Electrical resistivity tomography monitoring of two managed aquifer recharge ponds in the alluvial aquifer of the llobregat river (barcelona, spain). *Near Surface Geophysics*, 18(Geoelectrical Monitoring):353–368, 2020.
- [398] Sangok Seok, Albert Wang, Meng Yee Chuah, Dong Jin Hyun, Jongwoo Lee, David M Otten, Jeffrey H Lang, and Sangbae Kim. Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot. *Ieee/asme transactions on mechatronics*, 20(3):1117–1129, 2014.
- [399] M Sfakiotakis, A Kazakidi, and DP Tsakiris. Octopus-inspired multi-arm robotic swimming. *Bioinspiration & biomimetics*, 10(3):035005, 2015.

- [400] Hamidreza Shahbaznezhad, Farzan Kolini, and Mona Rashidirad. Employees' behavior in phishing attacks: What individual, organizational, and technological factors matter? *Journal of Computer Information Systems*, pages 1–12, 2020.
- [401] Noel Sharkey. Why robots should not be delegated with the decision to kill. *Connection Science*, 29(2):177–186, 2017.
- [402] K Gnana Sheela, Parvathy J Menon, S Swetha, CM Vandana, and Riya Mendez. Review on bio-inspired modular robotic system. *Materials Today: Proceedings*, 24:1918–1923, 2020.
- [403] Fan Shi, Moju Zhao, Masaki Murooka, Kei Okada, and Masayuki Inaba. Aerial regrasping: Pivoting with transformable multilink aerial robot. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 200–207. IEEE, 2020.
- [404] Masahiro Shibata and Sébastien Tixeuil. Partial gathering of mobile robots from multiplicity-allowed configurations in rings. In *International Symposium on Stabilizing, Safety, and Security of Distributed Systems*, pages 264–279. Springer, 2020.
- [405] Jun Shintake, Vito Cacucciolo, Dario Floreano, and Herbert Shea. Soft robotic grippers. *Advanced materials*, 30(29):1707035, 2018.
- [406] Amanda Siebert-Evenstone, Joseph E Michaelis, David Williamson Shaffer, and Bilge Mutlu. Safety first: Developing a model of expertise in collaborative robotics. In *International Conference on Quantitative Ethnography*, pages 304–318. Springer, 2021.
- [407] Alyssa Sims. The rising drone threat from terrorists. *Geo. J. Int'l Aff.*, 19:97, 2018.
- [408] JP Sinha. Aerial robot for smart farming and enhancing farmers' net benefit. *INDIAN JOURNAL OF AGRICULTURAL SCIENCES*, 90(2):258–267, 2020.
- [409] Mrs SM et al. Dtls for iot: Securing communications in a constrained environment. *IJAPR, UGC Care*, 7(3):166–173, 2023.
- [410] Shubhdildeep Singh Sohal, Bijo Sebastian, and Pinhas Ben-Tzvi. Autonomous docking of hybrid-wheeled modular robots with an integrated active genderless docking mechanism. *Journal of Mechanisms and Robotics*, 14(1), 2022.
- [411] Karthik Soma, Koresh Khateri, Mahdi Pourgholi, Mohsen Montazeri, Lorenzo Sabattini, and Giovanni Beltrame. A complete set of connectivity-aware local topology manipulation operations for robot swarms. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5522–5529. IEEE, 2023.

- [412] Svitlana Sotnik and Vyacheslav Lyashenko. Overview of innovative walking robots. 2022.
- [413] Paul J Springer. *Military robots and drones: a reference handbook*. ABC-CLIO, 2013.
- [414] Alexander Sproewitz, Masoud Asadpour, Yvan Bourquin, and Auke Jan Ijspeert. An active connection mechanism for modular self-reconfigurable robotic systems based on physical latching. In *2008 IEEE International Conference on Robotics and Automation*, pages 3508–3513. IEEE, 2008.
- [415] Alexander Sproewitz, Philippe Laprade, Stéphane Bonardi, Mikaël Mayer, Rico Moeckel, Pierre-André Mudry, and Auke Jan Ijspeert. Roombots—towards decentralized reconfiguration with self-reconfiguring modular robotic metamodules. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1126–1132. IEEE, 2010.
- [416] Alexander Spröwitz, Rico Moeckel, Massimo Vespignani, Stéphane Bonardi, and Auke Jan Ijspeert. Roombots: A hardware perspective on 3d self-reconfiguration and locomotion with a homogeneous modular robot. *Robotics and Autonomous Systems*, 62(7):1016–1033, 2014.
- [417] Astha Srivastava, Shashank Gupta, Megha Quamara, Pooja Chaudhary, and Vidyadhar Jinnappa Aski. Future iot-enabled threats and vulnerabilities: State of the art, challenges, and future prospects. *International Journal of Communication Systems*, 33(12):e4443, 2020.
- [418] William Stallings. *Cryptography and network security: principles and practice*. Pearson Upper Saddle River, 2017.
- [419] Pietro Stefanini. Militant kites and balloons: Anti-colonial resistance in palestine’s great march of return. *Partecipazione e conflitto*, 14(2):663–680, 2021.
- [420] Kasper Støy. Controlling self-reconfiguration using cellular automata and gradients. In *Proceedings of the 8th international conference on intelligent autonomous systems (IAS-8)*, pages 693–702. Citeseer, 2004.
- [421] Kasper Støy. Co-evolution of initial configuration and control in evolutionary robotics. *ALIFE 2021*, 2021.
- [422] Kasper Stoy, David Brandt, David J Christensen, and David Brandt. Self-reconfigurable robots: an introduction. 2010.
- [423] Kasper Stoy and Radhika Nagpal. Self-repair through scale independent self-reconfiguration. In *2004 IEEE/RSJ International Conference on Intelligent Robots*

- and Systems (IROS)*(IEEE Cat. No. 04CH37566), volume 2, pages 2062–2067. IEEE, 2004.
- [424] Kasper Støy and Radhika Nagpal. Self-Reconfiguration Using Directed Growth. In *Distributed Autonomous Robotic Systems 6*, pages 3–12, 2007.
- [425] Muhammad Sualeh and Gon-Woo Kim. Simultaneous localization and mapping in the epoch of semantics: a survey. *International Journal of Control, Automation and Systems*, 17:729–742, 2019.
- [426] John W Suh, Samuel B Homans, and Mark Yim. Telecubes: Mechanical design of a module for self-reconfigurable robotics. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 4, pages 4095–4101. IEEE, 2002.
- [427] Ning Sun. Autonomous mobile sonobuoy and its combat application prospect. In *2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, volume 3, pages 268–272. IEEE, 2023.
- [428] Zexin Sun, Hongyuan Mei, Wenten Pan, Zhengwei Zhang, and Jie Shan. A robotic arm based design method for modular building in cold region. *Sustainability*, 14(3):1452, 2022.
- [429] Jan Švec, Petr Neduchal, and Marek Hruží. Multi-modal communication system for mobile robot. *IFAC-PapersOnLine*, 55(4):133–138, 2022.
- [430] Petras Swissler and Michael Rubenstein. Fireant3d: a 3d self-climbing robot towards non-latticed robotic self-assembly. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3340–3347. IEEE, 2020.
- [431] Rafal Szczepanski and Tomasz Tarczewski. Global path planning for mobile robot based on artificial bee colony and dijkstra’s algorithms. In *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, pages 724–730. IEEE, 2021.
- [432] Elie Fute Tagne, Hugues Marie Kamdjou, Adnen El Amraoui, and Armand Nzeukou. A lossless distributed data compression and aggregation methods for low resources wireless sensors platforms. *Wireless Personal Communications*, 128(1):621–643, 2023.
- [433] Jun Tang, Gang Liu, and Qingtao Pan. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. *IEEE/CAA Journal of Automatica Sinica*, 8(10):1627–1643, 2021.

- [434] Ajay Kumar Tanwani, Raghav Anand, Joseph E Gonzalez, and Ken Goldberg. Ri-laas: Robot inference and learning as a service. *IEEE Robotics and Automation Letters*, 5(3):4423–4430, 2020.
- [435] Lo'ai Tawalbeh, Fadi Muheidat, Mais Tawalbeh, Muhannad Quwaider, et al. Iot privacy and security: Challenges and solutions. *Applied Sciences*, 10(12):4102, 2020.
- [436] Seppe Terryn, Joost Brancart, Dirk Lefeber, Guy Van Assche, and Bram Vanderborght. Self-healing soft pneumatic robots. *Science Robotics*, 2(9):eaan4268, 2017.
- [437] Seppe Terryn, Jakob Langenbach, Ellen Roels, Joost Brancart, Camille Bakkali-Hassani, Quentin-Arthur Poutrel, Antonia Georgopoulou, Thomas George Thuru-thel, Ali Safaei, Pasquale Ferrentino, et al. A review on self-healing polymers for soft robotics. *Materials Today*, 47:187–205, 2021.
- [438] Chris Thachuk, Erik Winfree, and David Soloveichik. Leakless dna strand displacement systems. In *International Workshop on DNA-Based Computers*, pages 133–153. Springer, 2015.
- [439] Pierre Thalamy, Benoit Piranda, and Julien Bourgeois. Distributed self-reconfiguration using a deterministic autonomous scaffolding structure. 2019.
- [440] Pierre Thalamy, Benoît Piranda, and Julien Bourgeois. A survey of autonomous self-reconfiguration methods for robot-based programmable matter. *Robotics and Autonomous Systems*, 120:103242, 2019.
- [441] Pierre Thalamy, Benoit Piranda, and Julien Bourgeois. 3d coating self-assembly for modular robotic scaffolds. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11688–11695. IEEE, 2020.
- [442] Pierre Thalamy, Benoît Piranda, and Julien Bourgeois. Engineering efficient and massively parallel 3d self-reconfiguration using sandboxing, scaffolding and coating. *Robotics and Autonomous Systems*, 146:103875, 2021.
- [443] Pierre Thalamy, Benoit Piranda, Frédéric Lassabe, and Julien Bourgeois. Scaffold-based asynchronous distributed self-reconfiguration by continuous module flow. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4840–4846. IEEE, 2019.
- [444] Pierre Thalamy, Benoît Piranda, Frédéric Lassabe, and Julien Bourgeois. Deterministic scaffold assembly by self-reconfiguring micro-robotic swarms. *Swarm and Evolutionary Computation*, 58:100722, 2020.

- [445] Pierre Thalamy, Benoît Piranda, André Naz, and Julien Bourgeois. Visiblesim: A behavioral simulation framework for lattice modular robots. *Robotics and Autonomous Systems*, 147:103913, 2022.
- [446] Pierre Thalamy, Benoît Piranda, André Naz, and Julien Bourgeois. Visiblesim: A behavioral simulation framework for lattice modular robots. *Robotics and Autonomous Systems*, page 103913, 2021.
- [447] Henk CA van Tilborg and Sushil Jajodia. *Encyclopedia of cryptography and security*, 2011.
- [448] Stefan M Trenkwalder, Iñaki Esnaola, Yuri Kaszubowski Lopes, Andreas Kolling, and Roderich Groß. Swarmcom: an infra-red-based mobile ad-hoc network for severely constrained robots. *Autonomous Robots*, 44(1):93–114, 2020.
- [449] Ashis Tripathy, Md Julker Nine, Dusan Losic, and Filipe Samuel Silva. Nature inspired emerging sensing technology: Recent progress and perspectives. *Materials Science and Engineering: R: Reports*, 146:100647, 2021.
- [450] Yuxiao Tu, Guanqi Liang, and Tin Lun Lam. Freesn: A freeform strut-node structured modular self-reconfigurable robot -design and implementation. 01 2022.
- [451] Thadeu Tucci, Benoît Piranda, and Julien Bourgeois. Efficient scene encoding for programmable matter self-reconfiguration algorithms. In *Proceedings of the Symposium on Applied Computing*, pages 256–261, 2017.
- [452] Thadeu Knychala Tucci, Benoit Piranda, and Julien Bourgeois. A distributed self-assembly planning algorithm for modular robots. In *International Conference on Autonomous Agents and Multiagent Systems*, 2018.
- [453] Elio Tuci, Roderich Groß, Vito Trianni, Francesco Mondada, Michael Bonani, and Marco Dorigo. Cooperation through self-assembly in multi-robot systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 1(2):115–150, 2006.
- [454] Nathan Tuck, Brad Calder, and George Varghese. Hardware and binary modification support for code pointer protection from buffer overflow. In *37th International Symposium on Microarchitecture (MICRO-37'04)*, pages 209–220. IEEE, 2004.
- [455] Vaibhav V Unhelkar, Shen Li, and Julie A Shah. Semi-supervised learning of decision-making models for human-robot collaboration. In *Conference on Robot Learning*, pages 192–203. PMLR, 2020.
- [456] Cem Unsal and Pradeep K Khosla. A multi-layered planner for self-reconfiguration of a uniform group of i-cube modules. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of*

- Robotics in the the Next Millennium (Cat. No. 01CH37180)*, volume 1, pages 598–605. IEEE, 2001.
- [457] Cem Ünsal, Han Kiliççöte, and Pradeep K Khosla. A modular self-reconfigurable bipartite robotic system: Implementation and motion planning. *Autonomous Robots*, 10(1):23–40, 2001.
- [458] Claudio Urrea and John Kern. Design and implementation of a wireless control system applied to a 3-dof redundant robot using raspberry pi interface and user datagram protocol. *Computers and Electrical Engineering*, 95:107424, 2021.
- [459] Giovanni Valecce, Gianfranco Micoli, Pietro Boccadoro, Antonio Petitti, Roberto Colella, Annalisa Milella, and Luigi Alfredo Grieco. Robotic-aided iot: automated deployment of a 6tisch network using an ugv. *IET Wireless Sensor Systems*, 9(6):438–446, 2019.
- [460] Serguei Vassilvitskii, Mark Yim, and John Suh. A complete, local and parallel re-configuration algorithm for cube style modular robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 1, pages 117–122. IEEE, 2002.
- [461] Janardan Kumar Verma and Virender Ranga. Multi-robot coordination analysis, taxonomy, challenges and future scope. *Journal of intelligent & robotic systems*, 102:1–36, 2021.
- [462] Davide Villa, Xinchao Song, Matthew Heim, and Liangshe Li. Internet of robotic things: Current technologies, applications, challenges and future directions. *arXiv preprint arXiv:2101.06256*, 2021.
- [463] Lu Anh Tu Vu, Zhuming Bi, Donald Mueller, and Nashwan Younis. Modular self-configurable robots—the state of the art. In *Actuators*, volume 12, page 361. MDPI, 2023.
- [464] J Walker. Search and rescue robots—current applications on land, sea, and air, 2019.
- [465] David A Wallace, Amy H McCarthy, and Mark Visger. Peeling back the onion of cyber espionage after tallinn 2.0. *Md. L. Rev.*, 78:205, 2018.
- [466] Christoph Walter, Simone Bexten, Torsten Felsch, Myroslav Shysh, and Norbert Elkmann. Safety considerations for autonomous, modular robotics in aerospace manufacturing. *Frontiers in Robotics and AI*, 9:1024594, 2022.
- [467] Chundong Wang, Yee Ching Tok, Rohini Poolat, Sudipta Chattopadhyay, and Mohan Rajesh Elara. How to secure autonomous mobile robots? an approach with

- fuzzing, detection and mitigation. *Journal of Systems Architecture*, 112:101838, 2021.
- [468] Jiankun Wang, Tianyi Zhang, Nachuan Ma, Zhaoting Li, Han Ma, Fei Meng, and Max Q-H Meng. A survey of learning-based robot motion planning. *IET Cyber-Systems and Robotics*, 3(4):302–314, 2021.
- [469] Xiaolu Wang, Hongzhe Jin, Yanhe Zhu, Bangxiang Chen, Dongyang Bie, Yu Zhang, and Jie Zhao. Serpenoid polygonal rolling for chain-type modular robots: A study of modeling, pattern switching and application. *Robotics and Computer-Integrated Manufacturing*, 39:56–67, 2016.
- [470] Carsten Weerth et al. Cocaine smuggling by help of narco-submarines from south america to europe and africa: a proven case—a last wake-up call for customs services around the world. *Customs Scientific Journal CUSTOMS*, (1/2020):37–42, 2020.
- [471] Wuchen Wei, Xiaofu He, Xiaoguang Wang, and Mingxiang Wang. Research on swarm munitions cooperative warfare. In *International Conference on Autonomous Unmanned Systems*, pages 717–727. Springer, 2021.
- [472] Xiangzhi Wei, Yaobin Tian, and Shanshan Wen. Design and locomotion analysis of a novel modular rolling robot. *Mechanism and Machine Theory*, 133:23–43, 2019.
- [473] Mochammad Haldi Widiyanto, Ardiles Sinaga, and Maria Artanta Ginting. A systematic review of lpwan and short-range network using ai to enhance internet of things. *Journal of Robotics and Control (JRC)*, 3(4):505–518, 2022.
- [474] Jackson Wirekoh and Yong-Lae Park. Design of flat pneumatic artificial muscles. *Smart Materials and Structures*, 26(3):035009, 2017.
- [475] Robert Wood, Radhika Nagpal, and Gu-Yeon Wei. Flight of the robobees. *Scientific American*, 308(3):60–65, 2013.
- [476] Damien Woods, Ho-Lin Chen, Scott Goodfriend, Nadine Dabby, Erik Winfree, and Peng Yin. Active self-assembly of algorithmic shapes and patterns in polylogarithmic time. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 353–354, 2013.
- [477] Kai Xing, Shyaam Sundhar Rajamadam Srinivasan, Major Jose Rivera, Jiang Li, Xiuzhen Cheng, et al. Attacks and countermeasures in sensor networks: a survey. In *Network security*, pages 251–272. Springer, 2010.
- [478] Menglei Xiu, Lihua Li, Shimin Feng, Wenda Hou, and Longfei Wang. Analysis of uuv whip antenna radiated power and optimal working frequency in seawater environment. *Progress In Electromagnetics Research C*, 118:61–70, 2022.

- [479] Jean-Paul Yaacoub, Hassan Noura, Ola Salman, and Ali Chehab. Security analysis of drones systems: Attacks, limitations, and recommendations. *Internet of Things*, 11:100218, 2020.
- [480] Jean Paul A Yaacoub, Javier Hernandez Fernandez, Hassan N Noura, and Ali Chehab. Security of power line communication systems: issues, limitations and existing solutions. *Computer Science Review*, 39:100331, 2021.
- [481] Jean-Paul A Yaacoub, Hassan N Noura, and Benoit Piranda. The internet of modular robotic things: Issues, limitations, challenges, & solutions. *Internet of Things*, page 100886, 2023.
- [482] Jean-Paul A Yaacoub, Hassan N Noura, and Ola Salman. Security of federated learning with iot systems: Issues, limitations, challenges, and solutions. *Internet of Things and Cyber-Physical Systems*, 2023.
- [483] Jean-Paul A Yaacoub, Hassan N Noura, Ola Salman, and Ali Chehab. Digital forensics vs. anti-digital forensics: Techniques, limitations and recommendations. *arXiv preprint arXiv:2103.17028*, 2021.
- [484] Jean-Paul A Yaacoub, Hassan N Noura, Ola Salman, and Ali Chehab. Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations. *International Journal of Information Security*, pages 1–44, 2021.
- [485] Jean-Paul A Yaacoub, Hassan N Noura, Ola Salman, and Ali Chehab. A survey on ethical hacking: Issues and challenges. *arXiv preprint arXiv:2103.15072*, 2021.
- [486] Jean-Paul A Yaacoub, Hassan N Noura, Ola Salman, and Ali Chehab. Advanced digital forensics and anti-digital forensics for iot systems: Techniques, limitations and recommendations. *Internet of Things*, page 100544, 2022.
- [487] Jean-Paul A Yaacoub, Hassan N Noura, Ola Salman, and Ali Chehab. Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations. *International Journal of Information Security*, pages 1–44, 2022.
- [488] Jean-Paul A Yaacoub, Mohamad Noura, Hassan N Noura, Ola Salman, Elias Yaacoub, Raphaël Couturier, and Ali Chehab. Securing internet of medical things systems: limitations, issues and recommendations. *Future Generation Computer Systems*, 105:581–606, 2020.
- [489] Jean-Paul A Yaacoub, Ola Salman, Hassan N Noura, Nesrine Kaaniche, Ali Chehab, and Mohamad Malli. Cyber-physical systems security: Limitations, issues and future trends. *Microprocessors and Microsystems*, 77:103201, 2020.

- [490] Brian M Yamauchi. Packbot: a versatile platform for military robotics. In *Unmanned ground vehicle technology VI*, volume 5422, pages 228–237. International Society for Optics and Photonics, 2004.
- [491] Mark Yim, Wei-Min Shen, Behnam Salemi, Daniela Rus, Mark Moll, Hod Lipson, Eric Klavins, and Gregory S Chirikjian. Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):43–52, 2007.
- [492] Mark Yim, Ying Zhang, and David Duff. Modular robots. *IEEE Spectrum*, 39(2):30–34, 2002.
- [493] Mark Yim, Ying Zhang, John Lamping, and Eric Mao. Distributed control for 3d metamorphosis. *Autonomous Robots*, 10(1):41–56, 2001.
- [494] Echi Yoshida, Satoshi Murata, Haruhisa Kurokawa, Kohji Tomita, and Shigeru Kokaji. A distributed method for reconfiguration of a three-dimensional homogeneous structure. *Advanced Robotics*, 13(4):363–379, 1998.
- [495] Daishi Yoshino, Yutaka Watanobe, and Keitaro Naruse. A highly reliable communication system for internet of robotic things and implementation in rt-middleware with amqp communication interfaces. *IEEE Access*, 9:167229–167241, 2021.
- [496] Tomonori Yoshizaki and Mélanie Sadozaï. Avantage compétitif et offset strategies américaines en asie-pacifique. *Revue Defense Nationale*, 812(7):69–73, 2018.
- [497] Zhang Youchun and Zhang Gongyong. Design of multimodal neural network control system for mechanically driven reconfigurable robot. *Computational Intelligence and Neuroscience*, 2022, 2022.
- [498] Randa Zarrouk, Saleh Mulhem, Weal Adi, and Mladen Berekovic. Clone-resistant secured booting based on unknown hashing created in self-reconfigurable platform. In *International Symposium on Applied Reconfigurable Computing*, pages 203–217. Springer, 2021.
- [499] Ajmal Zemmar, Andres M Lozano, and Bradley J Nelson. The rise of robots in surgical environments during covid-19. *Nature Machine Intelligence*, 2(10):566–572, 2020.
- [500] Zhenishbek Zhakypov, Kazuaki Mori, Koh Hosoda, and Jamie Paik. Designing minimal and scalable insect-inspired multi-locomotion millirobots. *Nature*, 571(7765):381–386, 2019.
- [501] Chao Zhang, Pingan Zhu, Yangqiao Lin, Zhongdong Jiao, and Jun Zou. Modular soft robotics: Modular units, connection mechanisms, and applications. *Advanced Intelligent Systems*, 2(6):1900166, 2020.

- [502] Lianxin Zhang, Zhang-Hua Fu, Hengli Liu, Qingquan Liu, Xiaoqiang Ji, and Huihuan Qian. An efficient parallel self-assembly planning algorithm for modular robots in environments with obstacles. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10038–10044. IEEE, 2021.
- [503] Yongzhou Zhang, Christian Wurll, and Björn Hein. Kuberos: A unified platform for automated and scalable deployment of ros2-based multi-robot applications. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9097–9103. IEEE, 2023.
- [504] Yu Zhang, Yubin Liu, Xin Sui, Tianjiao Zheng, Dongyang Bie, Yulin Wang, Jie Zhao, and Yanhe Zhu. A mechatronics-embedded pneumatic soft modular robot powered via single air tube. *Applied Sciences*, 9(11):2260, 2019.
- [505] Moju Zhao, Tomoki Anzai, Fan Shi, Xiangyu Chen, Kei Okada, and Masayuki Inaba. Design, modeling, and control of an aerial robot dragon: A dual-rotor-embedded multilink robot with the ability of multi-degree-of-freedom aerial transformation. *IEEE Robotics and Automation Letters*, 3(2):1176–1183, 2018.
- [506] Wenshuai Zhao, Jorge Peña Queraltá, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pages 737–744. IEEE, 2020.
- [507] Xingwei Zhao, Bo Tao, and Han Ding. Multimobile robot cluster system for robot machining of large-scale workpieces. *IEEE/ASME Transactions on Mechatronics*, 27(1):561–571, 2021.
- [508] Xin Zhou, Xiangyong Wen, Zhepei Wang, Yuman Gao, Haojia Li, Qianhao Wang, Tiankai Yang, Haojian Lu, Yanjun Cao, Chao Xu, et al. Swarm of micro flying robots in the wild. *Science Robotics*, 7(66):eabm5954, 2022.
- [509] Yueheng Zhou, Ming Liu, Chaoyang Song, and Jianwen Luo. Kirin: A quadruped robot with high payload carrying capability. *arXiv preprint arXiv:2202.08620*, 2022.
- [510] Quanyan Zhu, Stefan Rass, Bernhard Dieber, and Victor Mayoral Vilches. Cyber-security in robotics: Challenges, quantitative modeling, and practice. *arXiv preprint arXiv:2103.05789*, 2021.
- [511] Xiaopan Zhu, Chunjiang Bian, Yu Chen, and Shi Chen. A low latency clustering method for large-scale drone swarms. *IEEE Access*, 7:186260–186267, 2019.
- [512] Yanhe Zhu, Dongyang Bie, Xiaolu Wang, Yu Zhang, Hongzhe Jin, and Jie Zhao. A distributed and parallel control mechanism for self-reconfiguration of modular robots using l-systems and cellular automata. *Journal of Parallel and Distributed Computing*, 102:80–90, 2017.

- [513] Yanliang Zhu, Dongchun Ren, Deheng Qian, Mingyu Fan, Xin Li, and Huaxia Xia. Star topology based interaction for robust trajectory forecasting in dynamic scene. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3255–3261. IEEE, 2021.
- [514] Jacob Ziv. The universal lz77 compression algorithm is essentially optimal for individual finite-length n -blocks. *IEEE transactions on information theory*, 55(5):1941–1944, 2009.

LIST OF FIGURES

1.1	IoPMRT: Present Application and Future Use.	6
2.1	Classification of Modular Robots in terms of IoMRT.	14
2.2	Different Modular Module-to-Module (Swarmanoids) and Human-to-Module Communication links.	16
2.3	Communication Between Operators and Modular Robots.	16
2.4	Some examples of Modular Unmanned Ground Vehicles.	18
2.5	Some examples of Hybrid Multi-Terrain Unmanned Modular Vehicles.	20
2.6	Some examples of Modular Unmanned Aerial Vehicles.	21
2.7	Some examples of Modular Unmanned Maritime, Aquatic, and Underwater Vehicles.	22
2.8	Modular Robots in main IoT-related domains and fields.	24
2.9	IoMRT System Mapping, further detailed in Figure 2.10 and Figure 2.11.	45
2.10	A Proposed Defense-in-Depth Security Solution for Attack-in-Depth Case against Modular Robots and their components.	46
2.11	A proposed framework that classifies IoMRT's main components, along with their key weaknesses and countermeasures.	61
2.12	Proposed IoMRT Risk Life cycle.	64
4.1	A proposed Communication Layer based on IoMRT/Modular Robotics Protocol Stacks	104
4.2	Possible Machine Learning Solutions and Application for IoMRT.	114
4.3	Neuronal Network Distributed over a grid of <i>Blinky Blocks</i> . Red <i>Blinky Blocks</i> at the bottom left corner detects squares, it's orange right neighbor detects disks, and finally the next one in green is for triangles.	116
4.4	IoMRT Network Topologies.	119
4.5	Suggestions, Recommendations and Lessons Learnt.	122

4.6	The evolving of the robotic domain within the IoT field including past, present, and near future use.	129
5.1	Left: BB Hardware. Right: a set of 768 BBs running the same program to visualise a cutting plane of 3D Objects.	134
5.2	Experimental network diagram used to measure message propagation times.	135
5.3	Variation of communication delay.	136
5.4	Taxonomy of Existing Lossless Compression Algorithm Types.	137
5.5	VisibleSim view of 3D models used for experiments.	138
5.6	Two efficiency comparisons for different lossless compression algorithms. .	139
5.7	Variation of Communication Delay (a) and Time Ratio (b) in Terms of message length with/without Brotli Compression.	139
5.8	An example of <i>Blinky Blocks</i> application with transmission and decompression of a 3D description model (the short DNA chain presented in the right picture).	142
6.1	BBs Neighbour-to-Neighbour Authenticated Communication.	149
6.2	BBs Hardware.	149
6.3	(a) Example of passive attacks and (b) active attacks against modular robots.	149
6.4	PROLISEAN Protocol: Version 1.	151
6.5	PROLISEAN Protocol: Version 2 - Phase I: Authentication.	152
6.6	PROLISEAN Protocol: Version 3 - Phase II: Refined Authentication Phase.	153
6.7	PROLISEAN Protocol: Version 4 - Authentication Phase.	155
6.8	Proposed dynamic key generation steps DK , and the proposed techniques to construct the cryptographic and update cryptographic primitives.	158
6.9	(a) Recurrence, (b) PDF and key sensitivity(c) of the produced key-stream for 1 000 random dynamic keys.	161
6.10	Block diagram illustrating the key steps of the proposed lightweight multi-operations cipher scheme.	161
6.11	(a) The recurrence of a randomly-generated permutation table, (b) the correlation coefficient of the recurrence of 1 000 randomly-generated permutation tables, (c) the correlation coefficient between a randomly-generated permutation table and its updated version, and (d) the coefficient correlation between two subsequent permutation tables.	167

6.12 (a) The recurrence of a message, (b) permuted, (c) substituted, and mixed with keystream one (d) for a random session key. 168

6.13 (a) The distribution of a message, (b) permuted, (c) substituted, and mixed with keystream one (d) for a random session key. 169

6.14 Difference between original and encrypted messages (a), key sensitivity (b) against 1 000 random dynamic keys for the proposed cipher with all operations. 172

6.15 Throughput (a) of the proposed single operation variants and the corresponding ratio of the proposed cipher with all operations (b) and other single operation cipher variants. 174

LIST OF TABLES

2.1	Comparison table between Blinky Blocks, M-Blocks, KiloBots, and FreeBot.	33
2.2	Proposed Qualitative Risk Assessment Analysis for Modular Robotic Systems.	65
2.3	The Adopted Existing of Theoretical Robotic Solutions.	77
2.4	The Adoption of Existing Mobile Robotic Solutions.	79
3.1	The Adoption of Existing Frameworks & Module Robotic Solutions . .	87
3.2	The Adoption of Existing Algorithmic Robotic Solutions	92
3.3	The Adoption of Simulation-based Robotic Solutions	95
3.4	The Adoption of Existing Robotic Security Solutions	97
4.1	Abbreviation table with acronyms and their definitions.	102
5.1	Size of the Designed Data Models.	138
5.2	Numerical Example between different lossless compression algorithms using different message data sizes.	140
6.1	Table of Notations	148
6.2	Invertible Polynomial functions.	154
6.3	Statistical results for 1 000 update permutation iterations	166
6.4	Statistical Results	167

LIST OF DEFINITIONS

Title: Modular Robotics Meets Internet of Things: Safety, Security and Performance Challenges and Countermeasures

Keywords: Autonomous Modular Robots; Modular Self-Reconfigurable Robotics; Soft Robotics; Self-Reconfiguration; Swarmanoids; Internet of Modular Robotic Things (IoMRT); Internet of Modular Robotic Things (IoMRT); Internet of Swarm Robotic Things (IoSMRT); Swarm Robotics; Modular Autonomous Warfare; Counter-Terrorism.

Abstract:

This thesis presents the concept of Internet of Modular Robotic Things (IoMRT) and examines the essential role of modular robotic systems within the expanding Internet of Things (IoT) ecosystem. These systems enable programmable matter and adaptive environments, fundamentally transforming intelligent automation. The thesis identifies key requirements for the integration of modular robotic systems, emphasizing modularity, scalability, and real-time adaptability for effective communication with IoT infrastructure. It introduces a communication optimization method using Brotli compression to enhance data transfer speed and reliability in lattice-based modular robots. Additionally, this thesis proposes the Lightweight

Cryptography and Authentication Protocol for Blinky Blocks (LCAPBB) to safeguard against cyber-physical attacks while addressing IoT's computational constraints. It explores how modular robotic systems can support autonomous environments and distributed computing, offering innovative solutions for smart cities, logistics, and healthcare. By addressing current challenges and anticipating future advancements, this research lays the foundation for the secure and scalable integration of modular robotic systems into the IoT, guiding researchers, engineers, and policymakers towards smarter, more adaptive environments that will shape the future of automation.

Titre : La Robotique Modulaire Rencontre l'Internet des Objets: Défis et Contre-Mesures en Matière de Sécurité, de Sûreté et de Performances

Mots-clés : Robots Modulaires Autonomes; Robotique Modulaire Auto-Reconfigurable; Robotique Souple; Auto-Reconfiguration; Swarmanoïdes; Internet des Choses Robotiques Modulaires (IdMRR); Internet des Choses Robotiques Modulaires (IdMRR); Internet des Robots en Essaim (IdRER); Robotique en Essaim; Guerre Autonome Modulaire; Lutte Contre Le Terrorisme.

Résumé :

Cette thèse présente l'Internet des Objets Robotiques Modulaires (IoMRT) et examine le rôle essentiel des systèmes robotiques modulaires dans l'écosystème en expansion de l'Internet des Objets (IoT). Ces systèmes permettent la matière programmable et des environnements adaptatifs, transformant ainsi fondamentalement l'automatisation intelligente. La thèse identifie les exigences clés pour l'intégration des systèmes robotiques modulaires, en mettant l'accent sur la modularité, l'évolutivité et l'adaptation en temps réel pour une communication efficace avec l'infrastructure IoT. Elle introduit une méthode d'optimisation des communications utilisant la compression Brotli pour améliorer la vitesse et la fiabilité du transfert de données dans les robots modulaires en treillis. De plus, la thèse

propose le Protocole Léger de Cryptographie et d'Authentification pour les Blinky Blocks (LCAPBB) afin de protéger contre les attaques cyber-physiques tout en répondant aux contraintes computationnelles de l'IoT. Elle explore comment les systèmes robotiques modulaires peuvent soutenir les environnements autonomes et l'informatique distribuée, offrant des solutions innovantes pour les villes intelligentes, la logistique et la santé. En abordant les défis actuels et en anticipant les avancées futures, cette recherche établit les bases de l'intégration sécurisée et évolutive des systèmes robotiques modulaires dans l'IoT, guidant chercheurs, ingénieurs et décideurs vers des environnements plus intelligents et adaptatifs qui façonneront l'avenir de l'automatisation.