

# Theory of Self-maintaining Robots



**Chenjie SHI**

170117971

Supervised by

Professor Sandor M Veres

Department of Automatic Control and Systems Engineering

University of Sheffield

5<sup>th</sup> May 2021

**A thesis Submitted for the degree of MPhil**

# Contents

	Page
<b>List of Figures</b>	<b>6</b>
<b>List of Tables</b>	<b>7</b>
<b>Symbol and notation</b>	<b>8</b>
<b>Abstract</b>	<b>10</b>
<b>Acknowledgement</b>	<b>11</b>
<b>1. Introduction</b>	<b>12</b>
1.1. Motivation . . . . .	14
1.2. Problem Statement . . . . .	15
1.3. Aims and Objectives . . . . .	15
1.4. Contributions . . . . .	16
1.5. Thesis outline . . . . .	16
<b>2. Review of Related Literature</b>	<b>19</b>
2.1. Robots for maintenance . . . . .	19
2.2. Self-configuring robots . . . . .	22
2.2.1 Modular robotic system . . . . .	23
2.2.1.1 Class . . . . .	23
2.2.1.2 Architecture . . . . .	24
2.2.1.3 Connector . . . . .	24
2.2.2 Modular configurable robot . . . . .	25
2.2.2.1 SambotI and SambotII . . . . .	25
2.2.2.2 S-bot . . . . .	26

---

2.2.2.3	SMC Rover . . . . .	26
2.2.2.4	CKBot . . . . .	26
2.2.2.5	ATRON . . . . .	27
2.2.2.6	Stochastic reconfiguration . . . . .	28
2.2.2.7	Conclusions on modular and reconfigurable robots	28
2.2.3	Soft Robot . . . . .	28
2.3.	Reliability theory . . . . .	29
2.3.1	Fundamental Development . . . . .	29
2.3.2	Methods . . . . .	30
2.3.3	Redundancy allocation problem (RAP) . . . . .	31
2.4.	Multi-robot task allocation . . . . .	33
2.5.	Conclusions . . . . .	35
<b>3.</b>	<b>Principles of Self-maintaining Robot Design</b>	<b>36</b>
3.1.	Application areas . . . . .	36
3.1.1	Dangerous places . . . . .	36
3.1.2	Long working time without human support . . . . .	37
3.1.3	Industrial manufacturing processes . . . . .	37
3.2.	Requirements of self maintenance . . . . .	37
3.3.	Solution ideas . . . . .	38
3.3.1	SMR grouping principles . . . . .	38
3.3.1.1	Single robot self maintenance . . . . .	38
3.3.1.2	Self maintenance in robot teams . . . . .	39
3.3.2	Modules and components . . . . .	39
3.3.3	Robots and Modules . . . . .	40
3.4.	Qualitative design choices . . . . .	41
3.4.1	Platform/locomotion modules . . . . .	41
3.4.2	Power modules and components . . . . .	42
3.4.3	Processor modules . . . . .	42
3.4.4	Communication modules and components . . . . .	42
3.4.5	Manipulator modules and parts . . . . .	43
3.4.6	Components for active radiation protection . . . . .	43
3.5.	Qualitative measures of design choices . . . . .	43

---

---

3.5.1	Platform/locomotion modules . . . . .	44
3.5.2	Power modules and components . . . . .	45
3.5.3	Processor modules . . . . .	45
3.5.4	Communication modules and components . . . . .	45
3.5.5	Manipulator modules and parts . . . . .	46
3.5.6	Components for active radiation protection . . . . .	46
3.6.	Conclusions . . . . .	47
<b>4.</b>	<b>System Reliability Models</b>	<b>48</b>
4.1.	Problem description . . . . .	48
4.2.	Structural models . . . . .	49
4.2.1	Robot and module redundancies . . . . .	49
4.2.2	Reliability deficiency/failure detection . . . . .	50
4.2.3	Capacity of robots . . . . .	50
4.2.4	Cold standby redundancy and active redundancy . . . . .	51
4.3.	Cost of maintenance . . . . .	51
4.3.1	Corrective maintenance . . . . .	52
4.3.2	Preventative maintenance . . . . .	53
4.4.	Component value and importance . . . . .	53
4.4.1	Component value evaluation . . . . .	53
4.4.2	Importance value evaluation . . . . .	54
4.5.	Quantification of reliability . . . . .	54
4.5.1	Full functional requirement . . . . .	55
4.5.2	Minimal functional requirement . . . . .	55
4.5.3	Partial functional requirement . . . . .	56
4.5.4	Reliability with cold standby redundancy . . . . .	56
4.6.	Types of module switching . . . . .	57
4.7.	Conclusions . . . . .	59
<b>5.</b>	<b>Redundancy Allocation Problem</b>	<b>61</b>
5.1.	Problem description . . . . .	61
5.2.	Complexity issues . . . . .	64
5.3.	Optimisation methods . . . . .	64

---

---

5.3.1	Alternative optimisation methods . . . . .	64
5.3.2	Use of evolutionary optimisation . . . . .	65
5.4.	Calculation and results . . . . .	65
5.4.1	Illustrated example . . . . .	66
5.4.2	Comparison between fixed configuration and self maintaining robots . . . . .	67
5.4.2.1	Use of the hypervolume indicator . . . . .	69
5.4.2.2	Working time . . . . .	71
5.4.2.3	Quantity of redundancy (Capacity) . . . . .	72
5.4.3	Ratio of switch reliability . . . . .	72
5.4.3.1	Fixed configuration . . . . .	72
5.4.3.2	Uncertain configuration . . . . .	73
5.4.4	Conclusions . . . . .	74
<b>6.</b>	<b>Task Allocations for Module Replacements</b>	<b>75</b>
6.1.	Principles of replacement . . . . .	75
6.2.	Deciding to replace . . . . .	76
6.3.	Procedure of replacements . . . . .	77
6.3.1	Preparation stage . . . . .	77
6.3.2	Operation stage . . . . .	78
6.3.3	Completion stage . . . . .	78
6.4.	Task description . . . . .	78
6.5.	Mathematical model of the allocation process . . . . .	80
6.5.1	Capacity and resources . . . . .	80
6.5.2	Objective function and constrains . . . . .	82
6.6.	Evolutionary optimisation example . . . . .	84
6.6.1	Pre-processing of data . . . . .	84
6.6.2	Chromosome representation . . . . .	85
6.6.3	Crossover . . . . .	86
6.6.4	Mutation . . . . .	86
6.6.5	Fitness function and penalty function . . . . .	87
6.7.	Results and comparisons . . . . .	89

---

---

<b>7. Conclusions and Future work</b>	<b>90</b>
7.1. Achievements . . . . .	90
7.2. Limitations . . . . .	91
7.3. Future work . . . . .	92
<b>References</b>	<b>105</b>

# List of Figures

2.1	Robots for maintenance . . . . .	20
2.2	Modular configuration system . . . . .	27
4.1	Robotic group system $R_{gr}$ in serial-parallel structure . . . . .	49
4.2	Single robot $R$ in serial-parallel structure . . . . .	49
4.3	Tree-graph of self-maintenance. F1-F6 are activities for self-maintenance, F7-F12 are capabilities needed and F13-F20 are modules supporting the capabilities. . . . .	58
5.1	Multi-optimization figures of test problem using NGSA-II . . . . .	67
5.2	Comparison between RSS and RSR at 1300 time unit from multi objective optimisation . . . . .	69
5.3	Hypervolume indicator (grey region) . . . . .	70
5.4	HV results (mean) of RSS and RSR from time 100 to time 2000 . .	71
5.5	HV results (mean) related with ratio of switch reliability (fixed configuration) . . . . .	73
5.6	HV results (mean) related with ratio of switch reliability (uncertain configuration) . . . . .	74
6.1	Mission area and robots . . . . .	81
6.2	Chromosome representation with four missions and four robots . .	85
6.3	Graph representation for a chromosome derived from Fig 6.2 . . . .	85
6.4	Crossover of 5 robots with four missions . . . . .	87
6.5	Convergence of Random, Tournament and Roulette Wheel selection (cost refers to the travel time) . . . . .	88

# List of Tables

5.1	Data used in the modules . . . . .	66
5.2	Robot Configuration without redundancy . . . . .	66
5.3	HV results (mean) of RSS and RSR by time . . . . .	71
5.4	The HV results (means) of the quantity of redundancy . . . . .	72
5.5	The HV results (mean) derived from Ratio of switch reliability (fixed configuration) . . . . .	73
5.6	The HV results (mean) derived from Ratio of switch reliability (uncertain configuration) . . . . .	74
6.1	MRTA specifications . . . . .	88



---

## Symbols and Notations

$t$	time
$\lambda$	rate parameter
$i$	number of robot/module/sub-module
$j$	number of robot/failed robot/mission
$v$	type of robots
$V$	total number of robots' types
$av$	average(mean)
$\gamma$	cost of one maintenance
$m$	number of module/module type
$n$	number of cold standby redundancy
$r$	reliability of module
$R$	reliability of robot/robots
$f$	probability density function
$k$	$k^{th}$ failure arrival
$p$	switch reliability
$\mu$	minimal functional requirement
$K$	at least $K$ robots/modules needs to survive
$l$	number of robots for one type
$L$	total number of robots
$c$	cost of module
$w$	storage of robot
$limit$	limitation of storage
$h_j$	module class number for module $j$
$v_j$	component value for module $j$
$im_j$	importance for module $j$
$z_i$	number of self-maintaining mission the robot $i$ attended
$t_z^i$	the mission $z$ which robot $i$ participated
$TD$	travel distance
$TT$	travel time
$D$	travel distance between missions/robots

---

$Q_m^i$       the quantity of viable manipulators for mission  $i$   
 $sp_i$       speed of robot  $i$

---

## Abstract

This thesis proposes a theory for robotic systems that can be fully self-maintaining. The presented design principles focus on functional survival of the robots over long periods of time without human maintenance. Self-maintaining semi-autonomous mobile robots are in great demand in nuclear disposal sites from where their removal for maintenance is undesirable due to their radioactive contamination. Similar are requirements for robots in various defence tasks or space missions. For optimal design, modular solutions are balanced against capabilities to replace smaller components in a robot by itself or by help from another robot. Modules are proposed for the basic platform, which enable self-maintenance within a team of robots helping each other. The primary method of self-maintenance is replacement of malfunctioning modules or components by the robots themselves. Replacement necessitates a robot team's ability to diagnose and replace malfunctioning modules as needed. Due to their design, these robots still remain manually re-configurable if opportunity arises for human intervention. A system reliability model is developed to describe the new theory. Depending on the system reliability model, the redundancy allocation problem is presented and solved by a multi objective algorithm. Finally, the thesis introduces the self-maintaining process and transfers it to a multi robot task allocation problem with a solution by genetic algorithm.

---

## Acknowledgement

I would like to acknowledge the support from my supervisor Professor Sandor M Veres, who provided me guidance and deeper understanding in my research.

I would also like to thank for advice from Dr Dana Damian, who offered me a lot of information about soft robots and guidance at the beginning of my study.

Finally, thanks to my parents, who gave me an opportunity to study at the University of Sheffield.

# Chapter 1

## Introduction

In the past decades, the development and application of robots have served an increasing range of areas from room cleaning to space exploration [1] [2] [3] [4]. Robotic researchers and engineers developed more functionality, flexibility, adaptability and reliability of new robotic systems. This meant that the complexity of robots was increased, while multi-robot systems and swarm robots were developed to satisfy practical needs.

Robots started to replace humans in dangerous missions such as repair and rescue in nuclear plants. For example, in 2011, a massive earthquake and tsunami hit the Tohoku area in Japan, where the Fukushima Daiichi Nuclear Plant also suffered a damage that lead to a meltdown accident with the release of radioactive materials [5] . To inspect the damage at the nuclear disaster, Quince robots were deployed in the nuclear leak's area [6]. However, the robots rapidly malfunctioned when the mission started as the thermal cooling system produced issues for the robots during the extreme nuclear conditions. This happened despite that earlier the Quince robots performed perfectly well over tough terrains in non-nuclear environments.

Most robots working in nuclear plants would remain in the working environment specific missions, no matter how they fulfilled their job. This is so because maintenance and repair of these robots is dangerous to humans in most cases. Robots working in an extreme environment would be normally be abandoned after their radioactive missions. This practice is uneconomic and leads to significant waste of resources.

---

Apart from the dangerous environment, following with wide-range application of robots, the maintenance of robots also become a problem, which means it should consume more money, time and even best-trained engineer in detection and repair missions. The reason for that is the high cost, poor adaptation and low stability compared with other complex systems in both hardware and software aspects. Of course, the robotic system lacks a self-maintaining ability is another crucial factor.

In the past decades, multi robots or swarms has also been processed [7] [8] [9] [10] [11]. Compared with a single robot, multi robots system, or swarm robots has more potential for fault tolerance. One robot malfunction maybe could avoid the breakdown of the whole robotic system, if there are enough robots.

So we propose a theory for robotic systems that can be fully self-maintaining to resolve the maintenance problem. The approach uses heterogeneous and modular architectures, which can also be manually-reconfigured if opportunity arises for human intervention. The new design principles focus on functional survival of the robots over long periods without human maintenance. Self-maintaining semi-autonomous mobile robots are in great demand in nuclear disposal sites from where their removal for maintenance is undesirable due to their radioactive contamination. Similar are requirements for robots in various space missions. For optimal design, modular solutions are balanced against intelligent capabilities to replace smaller components in a robot by itself or another robot .

Modules are introduced for the basic platform and the payload, both of which enable self-maintenance within a team of robots helping each other [12] [13]. Furthermore, depending on the structure of the new theory, The primary method of self-maintenance is replacement of malfunctioning modules by the robots themselves. Replacement necessitates a robot team's ability to diagnose and replace malfunctioning modules as needed.

Furthermore a multi robot task allocation problem (MRTA) is introduced, and developed to represent and direct the self-maintaining process of robotic systems by SMR theory [14] [15] [16].

---

## 1.1 Motivation

Robotic systems are applied everywhere, including safe places and dangerous places. Moreover with effects from different aspects, the robot could meet various issues, then resulting in faulty condition before its life. Then there are two choices for users - to repair or to replace. Both of these two choices would waste money and time, maybe influence the efficiency of related missions. So in a safe circumstance also easy to reach, the engineer could repair or replace it. However, apart from the safe places, in some dangerous surroundings such as a nuclear plant (near the reactor), the malfunction of the robot signifies that replacement action is the only choice rather than the repair. It means that a new robot should be deployed to substitute the failed one to continue the mission with the extended budget.

The multi robotic system can be credited against single-robot systems, when it follows  $k$  out of  $n$  principle or similar idea. But the deployment of robots could also increase the cost and complexity of control, when a robot is regarded as an individual or fundamental unit. Because in that case, the robot becomes consumed and could be run out of.

Some researchers and engineers separate the robotic system or single robot into many pieces called modules, which have fixed independent autonomy. They can exchange the module and to reconfigure itself to increase the functionality. However, the self-repair or self-maintaining of modular robots and reconfigurable robots is not the main purpose of these design, which means the research tool and method is not appropriate for self-maintaining ability.

A new theory needs to be developed to focus on the self-maintaining ability with a reliable research approach to enhance the survivor chance of the whole robotic system. The self-maintaining ability could endow the whole robotic system great progress to deepen the application of various areas.

---

## 1.2 Problem Statement

Self maintenance by a team of robots has so far been prevented by one or more from a number of key factors; listed as follows.

- **Functionality:** Robot design fell short of essential functionality to maintain itself or others such as sufficient locomotion ability or manipulator. [17] [18].
- **Cooperation:** Many robotic systems only consider one individual for fault tolerance in missions without cooperation. Most multi-robot systems are short of cooperation ability such as communications and also lack enough numbers of effective individuals to meet self-maintenance needs [19] [20].
- **Structure:** The structure of robots is complex and hard to analyse and repair by the robots themselves. These types of robots therefore must be repaired by qualified engineers [12].
- **Practicability:** Many designs can have good performance in the laboratory and in an ideal environment, but can lack practicability in real application in extreme environments[6] [12].
- **Reliability:** Many published works don't have or have less than needed reliability theory analysis, which lack mathematical models to describe reliability in design. For example, quantity and types of modules could influence the reliability of modular robotic systems[12].

## 1.3 Aims and Objectives

The overall aim of this thesis is to introduce a reliable and relatively complete self-maintaining theory for robots. This theory must support robotic systems with an ability to maintain themselves to extend their life time in service and also meet the functional requirements of robot operators.

To support the new theory, a few objectives have been set:

- **Objective 1:** Develop a theory for the self-maintaining robotic system (SMR theory), which can be widely used and conveniently applied. The theory should have practical significance.



- 
- Objective 2: In the theory in Objective 1, a common probabilistic model is to be developed. The model should support the validation and upgrade of self-maintaining robots.
  - Objective 3: Models and algorithm should be found and presented for the execution of self maintenance by a team of robots.

## 1.4 Contributions

The contributions of the thesis are as follows:

- A new self-maintenance theory called SMR theory is developed to focus on increasing the likelihood of survival by robot teams during long periods of missions. The theory can be applied to a single robot and to a team of robots.
- A group of requirements and hardware considerations are summarised with demonstration of some examples to help users to optimise their design for the applicability of the SMR theory.
- Reliability theory is developed for modular robotic system and a complete mathematical model is constructed to describe structure and redundancy. The model is time dependent rather than a static model, which can provide the chance of survival for the whole system at a specific time.
- A redundancy allocation problem is derived from the SMR theory presented and introduced in the thesis to optimise the robotic system's configuration before a mission starts. By the support of an evolutionary optimisation algorithm, the robotic system can balance reliability and cost.
- An algorithm is introduced to describe the self-maintenance situation and state of the process at any time, and how the robotic system can organise the resources and tasks to repair failed robots with high efficiency.

## 1.5 Thesis outline

The structure of the thesis is as followed:

- 
- Chapter 2 explores the related work in the past. As no papers were found on self-maintaining robots at all, Section 2.1 surveys the history of robots for maintenance. Section 2.2 presents self-configurable robots with different attributes and contribution to these areas. We note that self-configuration is not a substitute for self-maintenance. Section 2.3 reviews reliability theory of general machinery. Section 2.4 reviews different approaches and solutions for multi-robot task allocations. Finally, section 2.5 draws the conclusion of the chapter.
  - Chapter 3 presents the fundamental principles of self-maintaining robots and hardware optimisation. Section 3.1 examines the existing application areas, and section 3.2 decides on the requirements of SMR theory. Section 3.3 represents approaches to fulfil the requirements of SMR theory. Moreover, Section 3.4 lists the qualitative design choices. Section 3.5 illustrates the quantitative measures of design. Section 3.6 represents the conclusion of the chapter.
  - Chapter 4 presents how robot engineers can optimise their design and can adopt the SMR theory for reliability assessment. Section 4.2 lists the options for qualitative designs. Section 4.3 illustrates different solutions for cost of maintenance. Section 4.4 lists two important parameters - *componentvalue* and *importance*. Section 4.5 presents the quantification of reliability for SMR theory. Section 4.6 discusses the replacement process for modules. And section 4.7 introduces the summary of chapter 4.
  - Chapter 5 introduces a redundancy allocation problem and its solution. Section 5.1 presents the description of the problem derived from the previous chapters. Complexity issues are discussed in Section 5.2. Furthermore, Section 5.3 introduces optimisation and an evolutionary algorithm for the redundancy allocation problem. Finally, Section 5.4 illustrates results and makes various comparisons.
  - Chapter 6 describes replacement procedures for modules and solution for task allocations. Section 6.1 lists the principles of replacement, Section 6.2 introduces decisions for replacement actions. Section 6.3 puts forward the

---

steps of replacement during missions. Section 6.4 provides task descriptions. Then in Section 6.5 a mathematical model is developed. Section 6.6 introduces an evolutionary solution for the problem. In the end, section 6.7 presents the results and some comparisons.

- Chapter 7 draws conclusions and intimates future work. Section 7.1 summarises the achievements of the thesis, some imitation are listed in Section 7.2. Finally, section 7.3 introduces ideas for future work.

# Chapter 2

## Review of Related Literature

In this chapter some of the related work is reviewed, including maintenance robots and reconfigurable, modular robotic systems. Furthermore, general reliability theory of machinery is also reviewed.

Recent progress of multi-robot task allocation is also reviewed, which supports the thesis' work on the construction of a self-maintaining algorithms for resource allocation during missions.

### 2.1 Robots for maintenance

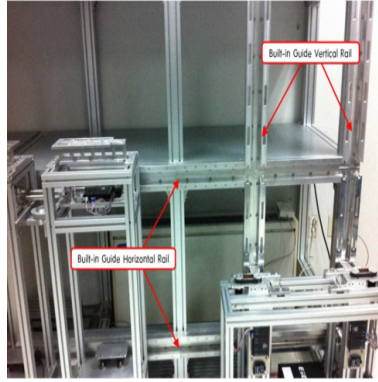
In the past there were many robotic systems developed and deployed in the maintenance area. However, most of them are trained to maintain other devices, which did not include robotic systems.

These robotic solutions enhance some specific functionality to fulfil maintenance tasks. The following is a list of some examples of robots developed for maintenance.

Luo et al. [21] developed a cable maintenance robot for cable-stayed bridges. With the wider application of cable-stayed bridges, maintenance also rapidly becomes a vital problem, when the bridges are always exposed to the natural environment including air, rain and sunshine. It is necessary to design a robot to maintain bridges with low cost, good safety and high efficiency, which is the robot for cable maintenance. By the modular method, the robot is consists of two modules- a climbing module and a maintenance module. The climbing



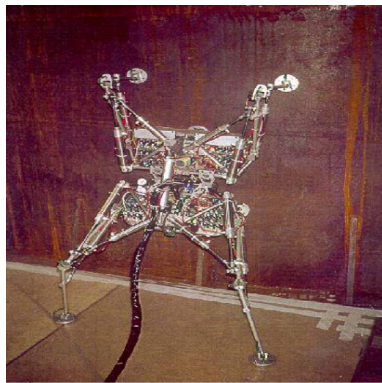
(a) Cable maintenance robot (CMR) [21]



(b) Building wall maintenance robot (BWMR) [22]



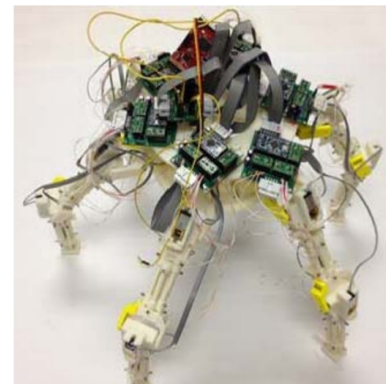
(c) Boiler water-cooling tubes maintenance robot [23]



(d) Robug IIs [24]



(e) Maintenance robot for wind power blade cleaning [25]



(f) Scalable hexapod robot SHeRo [19]



(g) An novel autonomous in-pipe robot to maintain the long-distance offshore oil pipelines [26]

**Figure 2.1** Robots for maintenance

modules have two different locomotion methods, an electrical climbing mechanism with all-driven wheels and a pneumatic worming climbing mechanism. As the author stated, displacement of maintenance modules can impose the robot deployed in different missions as well. The main work of the robot is painting to refresh the protection of cable-stayed bridges.

Moon et al. [22] designed a maintenance robot for facades (curtain-wall) of

---

high-rise buildings. Manual maintenance of facades on high-rise buildings is dangerous, attracts high cost and it is low in efficiency. The robotic system called 'building wall maintenance robot' (BWMR) tends to depend on the guide rail installed in a curtained wall. The system uses two types of robots - vertical climbing robots and horizontally moving robots for movements in two principle directions. A material transportation system can also be used to support the vertical robot to transfer material to a horizontal robot.

Maintenance robots for high-voltage transmission lines have been developed in the past [27]. Disturbance in the operational environment, including high altitude, high voltage and substantial electromagnetic interference, can result in contamination and damage to high-voltage transmission lines (HVTL). The maintenance of HVTL needs a live voltage operation, making it more challenging and dangerous. Many robots [28] [29] [30] have been designed for monitoring and maintenance, usually executing one operation rather than a complete a mission. They are also large and heavy, both of which cause difficulty in movements along the lines. Furthermore, the efficiency of a the manipulator used is questionable as it is done by remote control, which is lacking sufficient automation.

Gao et al. [23] developed a multi-functional climbing robot with two magnetic absorption tracks to maintain a boiler water-cooling system. Boilers play a crucial role in thermal power stations, where many factors such as strong steam and ash can lead to corrosion of the whole system. The robot is designed to protect against severe damage from the water-cooling system by regular maintenance. These robots also adopt a modular principle, which includes a double-tracked moving mechanism, an ash cleaning device, a slag purging device, a tubes' thickness measurement device, a marking device, and a control system. The robot has a strong locomotion ability, and can carry heavy payloads.

The Intelligent legged climbing service robot called Robug IIs is designed and fabricated to inspect and maintain large buildings and tall structures [24]. The design used adopted an insect-like structure and articulated limbs to increase the mechanical capability. The robot has shown good performance in experiments. But the robot lacks functionality such as a manipulator to assist in maintenance, as it is only an experimental robotic system for unstructured terrain.

---

Jeon et al. [25] developed a robot for wind turbine blade maintenance. As the application of wind power systems spreads, it increasingly needs robots to maintain their blades at high altitudes and in intense gusts of wind. The blade cleaning robot is equipped with 19 motors, 6 brush rotating motors, 3 brush vertical movement motors, 3 water jet turning angle regulating motors, 4 hoist motion motors to support the vertical motion along the blade.

Wang et al. [26] designed a novel autonomous in-pipe robot to maintain long-distance offshore oil pipelines. The merits of using pipelines to transport oil from sea to the land are energy-efficiency, high safety, low impact on the environment. Nevertheless, flaws occurring in the pipes are inevitable due to long-term erosion and mechanical effects of water. The robot is used to detect leakage and carry out pipeline maintenance. The mechanical structure of the robot is composed of an electric crawler and nine cylindrical sealed cabins, which also accept the modular principle to fulfil the purpose. In the software, the reactive self-rescue control technique is deployed in the robotic system to avoid its failure.

The scalable hexapod robot SHeRo has been designed for maintenance, repair, and operations [19]. The robot was equipped with 6 legs to mimic a spider-like structure, which enhances the locomotion ability in irregular terrain and supports the span of traverse larger gaps and obstacles and pass through narrow openings. However, designers and researchers focused on the development of their structure and locomotion control instead of their maintenance ability.

We can conclude that there are two different approaches to maintenance robots at present - inspection and maintenance. Robots with both inspection and maintenance ability are infrequent and are always applied to a specific area. Furthermore, only one maintenance robot [26] considers self-repair as a means to avoid failure of missions.

## 2.2 Self-configuring robots

The section lists self-maintaining work from robotic systems including rigid robots and soft robots.

---

## 2.2.1 Modular robotic system

Modular and re-configurable robotic systems have received a lot of attention in the development of robots in the past years. Modular robotic systems (MRS) have been derived from the dynamically re-configurable robotic system (DRRS), or cellular robotic systems by Fukuda [13]. The principles of these have been transferred to modular robotic systems. DRRS took its inspiration from observing living creatures, where cells form novel configurations to create new functionality when they assemble, although one cell itself may only have a simple structure and functionality. The paper [12] has shown that, compared with modular robotic systems, fixed-shapes or fixed functionality of robots limit the performance in unpredictable environments and in a variety of tasks. From the inspiration gained from flexibility, adaptability and self-organising properties of multi-cellular biological systems, a number of roboticists created self-organising machines that adapt themselves to unexpected conditions.

### 2.2.1.1 Class

Classes of MRS depend on the method of organisation including fixed-configuration, manually-reconfigurable, self-reconfigurable, and self-replicable [31].

- Fixed configuration: The shape of the robotic system is fixed, and does not have any ability to assemble.
- Manually-reconfigurable: The robots can be assembled to different configurations and structures by a human operator.
- Self-reconfigurable: The robotic system can change its own configuration and structure by itself
- Self-replicable: The robot can reproduce a similar configuration to its own.

A modular robotic system with self-reconfigurability is valuable for self-maintenance.



---

### 2.2.1.2 Architecture

Hardware architecture can be mainly classified into chain, lattice, mobile and hybrid [32] and [12].

- Chain architecture: Modules with few degrees of freedom (DOFs) are connected with each other and constitute a complex and flexible structure such as strings, loops and trees of many DOFs [33] [17] [34].
- Lattice architecture: Shape of modules are regular 3D or 2D hexagonal or rectangular patterns [35] [36] [37].
- Mobile architecture: Modules have locomotion ability to move independently in the environment [38] [39] [40].
- Hybrid architecture: It is the combination of various architectures.

The lattice architecture is mostly used in modular robotic systems [12]. The detail of this part will be discussed in reconfigurable robots' section.

### 2.2.1.3 Connector

Connector or docking systems play a vital role in the modular robotic system, affecting the functionality of the whole system.

Yim [41] [42] designed a connector for Polybot, which is a T-shaped hook structure with four bolts retained by a cross-shaped spring-loaded blade. Robots would have difficulty to use this on their own.

Jorgensen [43] proposed a hook based bonding mechanism for the Atron modular robotic systems.

Hossain [44] designed a rotary-plate genderless single-sided docking mechanism, which performs robustly and efficiently in unstructured terrains. The connector is deployed in ModRED robots [45].

It is clear that most of the existing modular robotic systems utilize mechanical methods, including hooks and bolts to form locked connections between two modules.

Murata[46] builds a permanent magnetic docking mechanism to connect M-TRAN modules. The spring system is also developed to loose the connection

---

between modules. Each module has three male and three female permanent magnets. Compared with the permanent magnetic, White [47] put forward an electromagnet connector.

Neubert [48] developed a strong, lightweight, and solid-state connection approach that depended on the low melting point of some alloys to form a reversible soldered connections without any moving components. This connector is appropriate for autonomous actions, which can connect and disconnect with the target without external manipulation. These connectors cannot be deployed in dangerous environments such as nuclear plants as their working temperature can be higher than that of the alloys used in the connections.

## **2.2.2 Modular configurable robot**

The units called 'modules' derive from modular robotic system that are always equipped with their own sensing, actuating and computing functionality. The docking system of modules allows the configuration of a larger structures. Hence, many modular robotic systems can be regarded as reconfigurable robotic systems due to their architectures. Furthermore, many robots would require self-maintenance ability from modular reconfigurable robotic systems by self-assembly.

### **2.2.2.1 SambotI and SambotII**

Wei et al. [49] put forward a new design and structure for a self-assembling robot called Sambot, which received inspiration from swarm robots. This robotic system consists of independent units including mobility and docking system to configure various structures. The single unit includes an active docking and an autonomous mobile body, which allows the connection with other units through various interfaces. With the application docking and locomotion functionality, the robotic system has become a mix of swarm robots and self-reconfigurable robots.

Then Tan et al. [50] continued the development of SambotI, and designed an upgraded a new version called Sambot II, which is enabled by more powerful calculation and accurate docking with a more stable system compared to

---

SambotI. Note however that SambotI and SambotII are homogeneous modular robotic systems with limited interaction ability with their environment to address practical problems such as inspection and rescue.

#### **2.2.2.2 S-bot**

Groß et al. [51] [52] developed a modular and swarm robotic system called S-Bot. The docking system of S-bot is supported by a gripper on-board each unit and it can achieve reconfiguration via semi-flexible connections. Three independent motors drive the connection of modules, which can create a three-dimensional formation. With a high quality combination of swarm and modular robotic systems, a robotic teamwork can fulfil arduous tasks including pulling heavy objects and overcoming gaps. Apart from creating connections, the rigid grippers can also provide many other capabilities to the robotic system for different missions such as interaction with other devices.

#### **2.2.2.3 SMC Rover**

Kawakami et al. [53] [54] designed a planetary rover system called "SMC rover", which is a modular robotic system that consists of one main body and detachable units named "uni-rover" derived from modular robotic systems. The main body and single-units are able to work independently without any interaction and connection. A single unit includes a manipulator and a wheel to move and build a connection with the main body. As a heterogeneous modular robotic system, the main body can connect with multiple units, but a single-unit can only attach itself to one other body. It is apparent that the SMC Rover can increase reliability and fault tolerance of robotic systems in various missions.

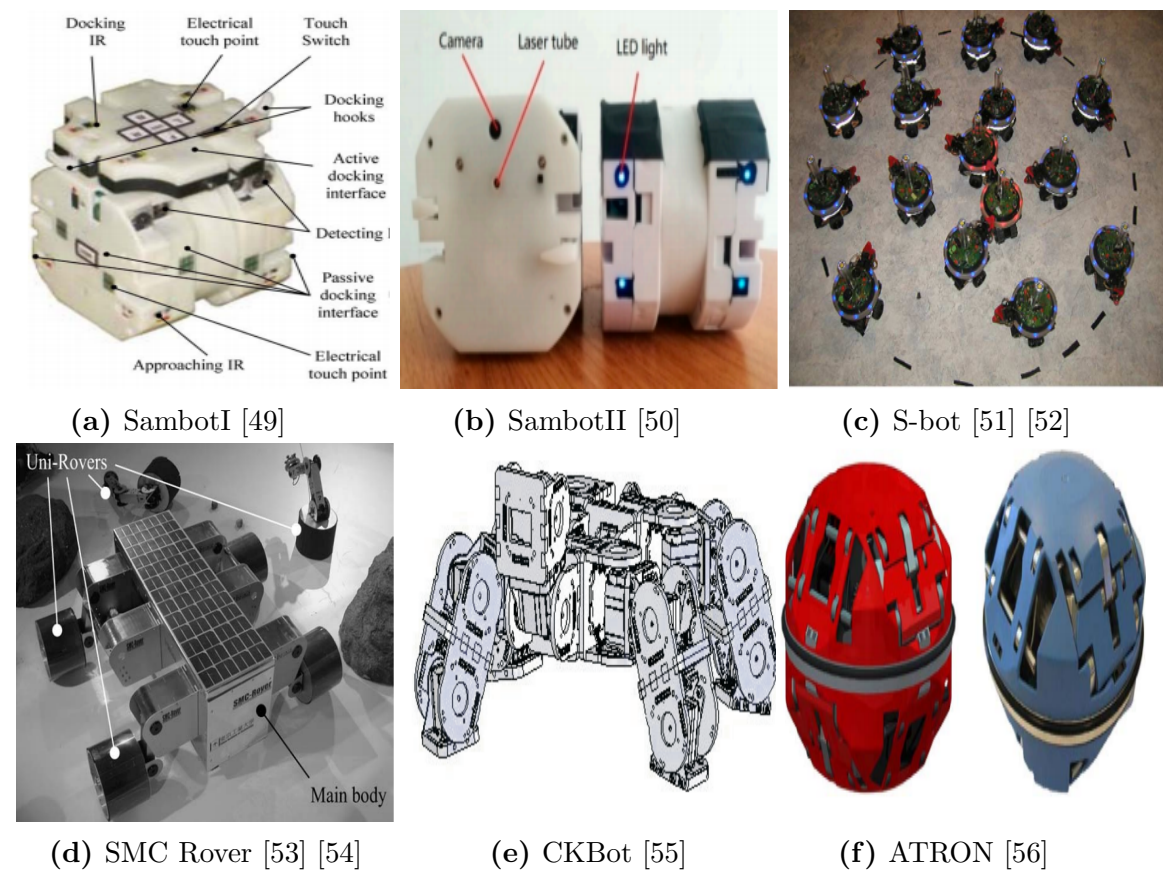
#### **2.2.2.4 CKBot**

Park et al. [55] developed a new modular configurable robot called "CKBot" as an experimental platform for validation algorithms. The CKBot (connector kinetic robot) can be regarded as a cube with four connectors on some facets. The robotic system can rely on attachment of units to transform itself into various configurations for new locomotion abilities such as walking and rolling. The merit

of the design is the demonstration of low level self-repair ability. After unexpected disconnection of units, the modules can locate each other and rebuild connections to reconfigure to the previous structure. The experimental environment assumes that any problem is only disconnection rather than damage to some modules and loss functionality.

### 2.2.2.5 ATRON

Østergaard [56] has put forward a new lattice-based self-reconfigurable robot named "ATRON". Each single unit consists of two hemispheres connected by a single revolute joint. ATRON illustrates good ability to self-reconfigure and form three-dimensional shapes. Apart from the demonstration of configuration, a method that relies on evolutionary computation is presented for self-repair ability at a limited level [57].



**Figure 2.2** Modular configuration system

---

### 2.2.2.6 Stochastic reconfiguration

The authors in [58] introduce and develop the first physical, three-dimensional stochastic robotic systems with self-assembly and configuration functionality. These are different from other designs of modular robotic system as single-units do not have locomotion ability before connections. This approach demonstrates that some stochastic processes can implement self-repairs by replacement of failed modules.

### 2.2.2.7 Conclusions on modular and reconfigurable robots

In this literature review some representative cases have been reviewed to illustrate recent progress and the importance of modular and reconfigurable robotic systems. It is no doubt that the results on modular and reconfigurable robotic system can contribute to self-repair or self-maintaining ideas in self-maintaining robots. Most modular robotic systems have a more or less potential for self-maintaining ability in ideal conditions. However, they mostly avoid the discussion of self-repair and pay no attention to it. On the other hand, many modular and reconfigurable robotic system would not be able carry enough functionality to fulfil some complex missions such as rescue in the nuclear plant, where self maintenance is needed.

## 2.2.3 Soft Robot

Apart from the traditional rigid robotic system, soft robots have been developed to have self-healing ability or damage resilience, which can be regarded as self-maintaining ability [59] to some degree. Soft robots are derived from biological systems of creatures including high degrees of freedom, dexterity, environment adaptability and power output [60] [61] [62]. The self-healing functionality and damage resilience of soft robots are achieved by the use of self-healing materials such as polymeric and elastomeric materials. As being different from traditional rigid robots, components with multifunctional materials can fulfil more than one functionality at one time. The main application area of such soft robots has been focusing on actuation, electronics and structure. The solutions for self-healing in soft robots mostly rely on the following materials:

- 
- Actuation: Dielectric Elastomer Actuators (DEAs) [63], inflatable actuators [64], hydrogel actuators [65].
  - Electronics: Room temperature liquid metal-based electronics [66] and composite polymer electronics [67].
  - Structure: Low melting temperature materials [68] and shape memory materials [69].

However, such soft robots still have many problems to overcome in the area of self-maintaining. Firstly, soft robots still lack proper development and application of new materials, which so far tend to be unstable long term. Secondly, the performance of these robots mainly depends on the nature of interfaces between robots. Finally, compared with rigid robots, soft robots have an advantage when interacting with humans. They are however still short of the level of environmental adaptation that is needed in dangerous places such as nuclear plants and space. At present soft robots contribute little to design idea and principle in the thesis.

## 2.3 Reliability theory

Here we review work related to our reliability theory for complex system, robotic system and redundancy allocation problem since reliability plays a crucial role in the self-maintaining robot's (SMR's) theory.

### 2.3.1 Fundamental Development

Despite the published papers and books on reliability of machines in general, there is limited research on robotic system reliability [70], while most related work focus on repeatability and accuracy [71].

Khodabandehloo et al [72] begin to apply failure mode and effect analysis (FMEA), event-tree analysis (ETA) and fault-tree analysis (FTA) on the estimation of robots. Jin et al. [73] evaluates probabilistic behaviour and performed reliability analysis of a multi-robot system with Petri net and Markov renewal process theory. Dashui et al. [74] introduces a reliability model and a

---

design method for robotic assembly operation, which can predict the assembly reliability of robots during a time period.

Khodabandehloo et al [75] present many effects related to safety and reliability of robotic systems, including the integrity of robots' hardware and software, communication of different devices, and environmental influences.

Dhillon and Yang [76] assessed the possibility of a system consisting of a robot and its safety system, which utilize supplementary variables and Markov techniques.

Monteverde and Tosunoglu [77] develops the analysis of an approach to create fault-tolerance of a robotic system via kinematic redundancy and dual actuation in robotic manipulators . They [78] promote and deploy their fault tolerance methods on the serial and parallel robotic systems.

Michaelson and Jiang [79] apply the redundancy systems on multiple robotic systems to evaluate the degree of fault-tolerance.

### 2.3.2 Methods

The section provides an overview of some papers that model, analyse and predict robotic system failures.

Kumar et al. [80] developed a hybrid technique to estimate the reliability of robotic systems. Various reliability values, including system failure rate, repair time, mean time between failure (MTBF), expected number of failure (ENOF), availability, and reliability are described by fuzzy membership functions. The aim is to prevent unexpected failures and enhance the performance of robotic systems. Fuzzy set theory has been deployed to quantify uncertainties, while model construction of the system depends on the fault tree. Furthermore, using so called lambda-tau methods, mathematical formulae for repair or failure rate are constructed. Moreover, genetic algorithms are deployed to deal with nonlinear programming problems.

Sharma et al. [81] presented a reliability analysis of multi-robot system, which includes Real Coded Genetic Algorithms (RCGA) and the fuzzy Lambda-Tau Methodology (FLTM). The optimal parameters mean time between failures (MTTF) and mean time to repair (MTTR) is proposed to be calculated

---

by genetic algorithms. Application of Petri nets (PNs) has been adopted to describe interactions among the active components of multi-robots systems. FLTM is used to acquire many parameters, including failure rate, repair time, the expected number of failure (ENOF) and reliability. The reliability framework of this paper optimises existing probabilistic approaches using graphical representations.

Ferguson and Lu [82] presented a fault tree analysis for a coolant outlet pipe snake-arm inspection robot deployed in a nuclear plant. Fault tree analysis can help users to gain a qualitative assessment for system reliability. Via a case study, the paper investigates the reliability of a nuclear robot and builds up the logical framework, which analyses the main reasons related to fault tolerance to support the engineers in order to upgrade their design.

Fazlollahtabar and Niaki [83] also develop a comprehensive fault tree analysis (FTA) for the main components of some industrial robotic systems. This approach integrates a reliability block diagram (RBD) to aid in evaluating the reliability of robots. Their case study considers an autonomous guided vehicle (AGV). For systems with many components, a decision tree-based hazard function is developed to acquire the failure rate of each component and the whole system. Two value parameters, cumulative hazard rate and average failure rate, are computed as well to evaluate overall system reliability. The merit of this method are to construct a complete system structure clearly and provide an efficient and reliable evaluation of result for robot users.

Coit [84] introduces a method to represent system reliability with cold-standby redundancy, and the system must be a non-repairable system. A system reliability model for variable components with imperfect component switching reliability is constructed and validated in a case study. The merit of the paper is that it provides standard solutions to quantify the reliability of machine systems.

### **2.3.3 Redundancy allocation problem (RAP)**

Using spare modules by robots can be regarded as relying on redundancy, which will be introduced in detail later. It is important to decide the allocation of redundancy or spare modules to optimise the robotic system before the missions



---

starts, which would affect the performance, hence here we review the topic of redundancy allocation for parts of machinery in general.

Liang and Smith [85] presented an ant colony meta-heuristic optimisation method to solve the redundancy allocation problem, which appears to be the first application of ant colony optimisation in reliability design. The performance of this method is more effective compared with genetic algorithms, and the computational burden is acceptable as well, while dealing with the high volume of input.

Kulturel-Konak et al. [86] have introduced the "tabu search" (TS) method into the redundancy allocation problem. TS is a competing meta-heuristic approach to solve large and complex combinatorial optimisation problems. It is a simple and convenient solution that moves through successive iterations by considering neighbouring shifts. The paper utilizes a penalty function to modify the TS method for RAP. To deal with various reliability optimization problems with unsuccessful programming methods, TSRAP has superior performance compared with the genetic algorithm approach.

Coit and Konak [87] developed a new heuristic solution for the redundancy allocation problem with multiple weighted objectives (MWOs). The idea is to transfer a multiobjective problem into a group of single-objective problems, which increases the reliability of each independent subsystem. There are a few methods that permit linear programming algorithms and software applied on redundancy allocation. The calculation and comparison show that the MWO is an efficient method to solve redundancy allocation problems.

Govindan et al. [88] developed a novel multi-objective method called MOHNS (Multiobjective Hybrid Metaheuristic) to solve the component redundancy allocation problem, and the approach is a hybridization of NSGA-II (Non-dominated Sorting Genetic Algorithm II) and an adaptive population-based simulated annealing method. The paper discusses the redundancy allocation problem for serial-parallel structures, including two scenarios - continuous monitoring and detection, and detection and switching only at the time of failure. Moreover, different subsystems and components have various allocation rules and strategies compared with other models in order to

---

enhance the practicability and universality. The purpose of the paper is to optimise the system configuration to achieve maximum system reliability and minimum total cost, while bringing these two values into a trade-off.

Chambari et al. [89] has put forward an efficient method for the redundancy allocation problem via a new redundancy strategy obtained by a simulated annealing algorithm. The redundancy strategy includes cold-standby and active status, which represent different reliability and activation models for a subsystem. The choices to be made by optimisation are more complex but the increase is tolerable. For comparison, the quality of SA (simulated annealing) for RAP is evaluated.

## 2.4 Multi-robot task allocation

In our approach to self maintaining robots, there are lots of robots involved in a missions. If one or more robots malfunction, the robotic system should put forward the flocking strategy to decide how many and which robots need to offer the support for the self-maintaining operation. In the self-maintaining process, energy cost and maintaining efficiency directly influence the stability of the robotic system. So the multiple robot task allocation (MRTA) is introduced to solve the flocking problem, which could enhance the advantage of self maintaining robot (SMR) theory. MRTA is a problem that decides which robots and when they should execute tasks to optimise the efficiency of the whole system to achieve a coordinated team behaviour.

Dong-Hyun et al [90] develop a resource-oriented, decentralised auction algorithm (RODAA) for the multi robot task allocation problem with multiple resources of robots and limited robot communications range. The paper utilizes a solar panel cleaning mission as an application and validation environment.

Kai et al [91] have put forward a solution that relies on a stochastic clustering auction (SCA), which uses a Markov chain search process in simulated annealing. The method is applied to heterogeneous robots teams. The advantage of this algorithm is that, by tuning the annealing suite and turning the upward movements on and off, the global robots team performance will move into the region between

---

the optimal global performance and the performance associated with a random allocation after the algorithm converges.

Shi et al [92] proposed a dynamic auction approach for differentiated tasks under cost rigidities (DAACR) to solve real-time, dynamic, complex and confrontational working environment MRTA problems, which is a merging algorithm to solve both distributed systems and those with a centralised structure. In case of robot failures, rescue missions can continue to damage the whole system without any positive return for the rescue. So the purpose of the paper is to decrease the system damage caused by disaster during missions.

Yuan et al. [93] introduced a CNP (Contract Net Protocol) combined with a neutral network to solve the multi-robot allocation problem. A heterogeneous multi-robot system named UMRS-1 is used as an experimental platform for a robot patrolling or used for intrusion detection. UMR-1 is a distributed system both in its logic and physically, in which robots can dynamically join in or quit via auto-configuration. Compared with robots in other bidding situations, each robot can provide more than one bid price instead of one price, so that each bidder price represents an ability in one aspect.

Zitouni et al. [94] have recommended a distributed approach using the consensus-based bundle algorithm and ant colony system to solve multi robotic allocation problems. The problem addressed is the application of UAVs in search and rescue missions for survivors. The objective is to save the maximum number of survivors within minimum time and overall travel distance. The problem is divided into two phases of inclusion and consensus, also utilising ant colony principles and adequate coordination mechanisms.

Arif et al. [95] presented an evolutionary algorithm to calculate the allocation and scheduling for a multi-robot system, which is applied to the ST-MR-TE (Single Task, Multi-Robot and Time Extended assignment ) problem. To increase the efficiency, the two chromosome representation is utilised to solve the problem, which is capable of a variety of MRTA distributional stages with good quality.

---

## 2.5 Conclusions

Derived from literature review, it is clearly that territory of robotic systems lack of a fundamental theory and its application for stable self-maintaining robots with practical multiple functions. The modular robotic systems offer a solid function to solve the self-maintaining problem and its functions. Furthermore, the multiple robots are introduced to enhance its survivability and practicability. Apart from that, system reliability theory introduces efficient methods to analysis and validate the fundamental idea in the thesis, which redundancy allocation problem provides a lot of useful advice to the users. Finally, MRTA supports the construction of self-maintaining process in missions.

# Chapter 3

## Principles of Self-maintaining Robot Design

In this chapter concepts of self-maintaining robots have been discussed, some of them first introduced here. The definition of self-maintaining robots, their architecture and functionalities are new here.

### 3.1 Application areas

As for the wider application of robots, there are still many areas where robots have poor performance but high potential. For example, the robots deployed in a nuclear power plants can maintain functionality for a short time, maybe 2 hours, then lose control and remain where they broke down, where they suffered from high radiation doses and possibly temperatures. Similar problems arise in space and planetary robotics. Self-maintaining robots can decrease the long term cost of robot use by making robots capable to renew themselves by self-maintenance. So self-maintenance is of critical importance in enhancing the stable use and adaptation of robots with improved average costs in some specific areas.

#### 3.1.1 Dangerous places

Dangerous place refers to an environment that has a high chance to damage or affect the normal function of a robot, leading to failure of missions. Robots such as the Quince deployed at the Fukushima Daiichi Nuclear Plant [6] had cooling

---

systems that were destroyed after the mission started. This resulted in a malfunctions of robots due to the high temperatures. Even though engineers and researchers developed components and materials with high tolerance to nuclear environments, the working time of robots was still very limited with high failure rates. When robots malfunctioned in the nuclear plant, maintenance of robots was impossible.

Also outer space and deep-sea operations can be classified as dangerous places, where the main affect on the robots is high pressure, poor visibility and narrow communication bandwidth.

### **3.1.2 Long working time without human support**

These situations present the robots with the challenge of executing tasks in a remote location without any maintenance support from humans for an extended period of time. The periods often stretch to months or years. The remote places can include desert, caves and deep see, where humans have difficulty to have access and support the robots. Failed robots at remote locations are abandoned, which can negatively affect the environment and the mission itself too.

### **3.1.3 Industrial manufacturing processes**

In industrial manufacturing based on robotic systems, where no human is present in the factory, failure of robots can lead to suspension of production, and can also increase costs of operating the robots as maintenance of robots would need human resources to inspect and resolve issues. Compared with other potential application areas, the main purpose of industrial applications is low cost and high efficiency.

## **3.2 Requirements of self maintenance**

To satisfy self-maintaining ability, robotic systems need to adhere various requirements, such as:

- Easy inspection: The robots can inspect faults easily and as soon as possible before serious effects result.

- 
- Convenient maintenance: Maintenance of robots should be easy and convenient, which can be carried out by robots.
  - High tolerance: Robotic systems need high tolerance for meeting the self-maintaining process.
  - Necessary functions: The robotic system should have the necessary functionalities to support the self-maintenance process.
  - Decision maker: The robotic system should have a calculation ability to organise the resources for self-maintenance.
  - Cooperation: When the robotic system consists of multiple individuals, ability of cooperation of robots is needed.

### **3.3 Solution ideas**

This section examines options for robot groupings and their modules for the development of self-maintaining robots.

#### **3.3.1 SMR grouping principles**

The problem of self maintenance by robots can be considered for

- (a) a single robot;
- (b) a homogeneous set of robots with similar architectures;
- (c) a heterogeneous set of robots with varying architectures.

##### **3.3.1.1 Single robot self maintenance**

In case of a single robot needing to self-maintain itself, it needs technical solutions, both in mechatronics and software, which keep its performance high for long periods of time. These solutions can include a structural organisation, which optimises the probability of the robot being capable of replacing failing components and able to reconfigure its software. Such a structure would inevitably include distributed computing of actuators, which handle components combined with some redundancy of the actuators in case they need to be

---

replaced by the robots themselves. Distributed software needs to recover from any failure except if all processors stop working at the same time. The probability of failure needs to be kept low. There is a lot to learn from aviation safety solutions developed for many years for passenger aircraft [article] flying today, which achieved remarkably high-reliability levels. The difference is that our robots are able to self-maintaining for very long time periods, unlike aircraft maintained by technical staff. Similarity occurs during a flight when robust operations performance is required. Some ideas of this thesis, in the use of both software and hardware redundancy, originate from aviation systems.

### **3.3.1.2 Self maintenance in robot teams**

The use of a team of robots, who can help each other, can significantly increase the probability that the team's performance can be maintained. The total failure of any single robot may be recoverable to full functionality.

Sets of robots with homogeneous architecture, where each robot has the same architecture, can simplify the overall design and assessment of reliability. It also simplifies the skills set the robots to need for repairs.

A heterogeneous set of robots, which have different architecture, likely to face more challenging problems when aiding each other in case of faults. On the other hand, Heterogeneous robots may be needed for practical work in some applications. Hence they remain an essential case for reliability assessment.

### **3.3.2 Modules and components**

A component in robotic systems is a small part that can be less expensive to replace than a module when it fails. For easy handling and replacement by the robots themselves, components should be plug-and-play (PAP). For instance, sensors and gripper actuators can be made self-testing and PAP replaceable, and so can be mobility components such as wheels and motors.

Modules represent combinations of components that together serve a well-defined functionality and as such are simple to replace, by PAP, for mechanical, computing hardware and software modules, and also for connectivity and communication modules.



---

Design and maintenance can be simplified if robots are based on a modular-architecture rather than on a components-based architecture. Here we assume that a module is an interconnection of a set of components. On the other hand, component replacement can be less expensive and wasteful, although the development of robot abilities to discover faults and replace components can come with higher overall system complexity.

It is often difficult to decide whether a module should be called a component or a component to be called a module. For this reason, we unify these concepts, and we refer to all PAP replaceable parts of a robot as 'modules'. Calculations of economic structural design and costs can decide which parts of a robot design should then become modules.

Based on the principles outlined above, the design requirements of self-maintaining robots (SMRs) can be introduced as follows.

*Definition 3.1* A single robot or a team of robots is called *self maintaining at robustness level  $k$*  if it satisfies the following conditions. If, during full functionality,  $k$  components malfunction, then

- (a) they are able to identify all failing modules;
- (b) they are able to replace all failing modules.

If, in addition, the time averaged costs of module failures and replacements is minimised by design, then they are called  *$k$ -robust optimised*.

It is much needed in practical robotic applications that *long term average cost* of self-maintenance is assessed and quantified. The theory presented here aims to help this assessment.

### 3.3.3 Robots and Modules

In the theory of self maintaining robots we need to address the problem of how heterogeneous robotic systems can acquire more ability and capacity for various missions with limited costs. Even though most modules have a relatively autonomous functionality compared with those in non-self-maintaining robots, we still call them as modules rather than robots, unlike in re-configurable robotics. A robot is represented by a configuration of several modules. The basic

---

unit is modules, components are handled like modules. Replaceable components with plug and play qualities are called modules.

## 3.4 Qualitative design choices

By nature of robotic development, the theory of self maintenance needs to address two main aspects of maintenance: (1) hardware self-maintenance (2) software self-maintenance. In later sections, the thesis focuses on the hardware solution to support the self-maintaining robots.

Concerning the hardware, which includes mechanical components, actuators and the electronics of digital control and computing, This section proposes six classes of modules to start from.

A most fundamental requirement for hardware self-maintenance is that the modules should have attributes such as easy assembly and facilitate the ability of assembly and replacement by the robots themselves. These principles can be aided by providing suitable mechanical designs of connectors for each hardware module, so that the manipulator can easily handle fitting and assembly in any self-maintaining process. Consequently, the *manipulator module* is a fundamental part of any SMR system.

Apart from manipulators, a locomotion module and a processor module are also vital in a team of robots for their planning, changing their locations and to provide control signals. The same modules can serve the execution of module extraction and fit for replacement. To illustrate a possible SMR theory and its mathematical analysis, here we discuss six classes of modules.

### 3.4.1 Platform/locomotion modules

A platform module class is a crucial one in that it can deliver mobility for itself and other payload/functional modules, which rely on it. Apart from its power subsystem with batteries, the platform module needs to have a docking system to connect with other modules. Docking can rely on electronic connectors, which attach various payload modules to it. Moreover, it can be a design choice to provide distributed computing by making a widely needed robot navigation system a part

---

of the platform module class.

### **3.4.2 Power modules and components**

The power module class includes a powerful battery for energy supply and some sensors, which help processors and manipulators to detect the available power and energy level remaining. The latter is especially relevant in reassembly and self-reconfiguration operations.

### **3.4.3 Processor modules**

The processor module class can be regarded as a computational unit that processes sensory signals and uses them in the execution of its movements and its manipulation tasks. This involves the computation of the variable complexity local model, which contains as much detail as needed in a given task. This module can also compute plans of an action series for a robot to achieve various goals. For hardware reliability of a robot, outside the processor module, some computation can be distributed, such as navigation in platform modules or visual feedback computation in manipulation modules. Prediction and detection of component and module failure can also be distributed to other modules. Planning for self maintenance and aiding the functional recovery of other robots, is however to be retained in the processor module class. The highest level decision making by hierarchical planning is also practical to be retained in the processor module to play a coordinating role for the operation of a robot individual.

### **3.4.4 Communication modules and components**

The communication module class can include two kinds of activities. The first is to provide a network of communication between modules of a single robot. And second is to provide the means of communication of the processor module with other robots. For safety, there can be the redundant set of communication channels both inside a robot as well as among robots. Among robots, wifi communication can be physically disturbed, and alternative communications by sound, light, visual

---

signalling and vibration are a practical alternative to provide reliability.

Under some extreme conditions, such as in nuclear waste silos, the alternatives of communication technology may be necessary to deploy. Similarly, communication with remote human supervisors of the robots necessitates the communication module to host computational means of alternative communications.

### **3.4.5 Manipulator modules and parts**

The manipulator module class plays a pivotal role in the functional life of self-maintaining robots. Most of the payload/functional modules, which serve the purpose rather than the mere survival of a robot. Also rely on the manipulator module. In most missions of a robot team, if there is no manipulator module left functioning, then the robotic system has lost its self-maintaining ability.

### **3.4.6 Components for active radiation protection**

Practical areas where self-maintaining robots are needed are nuclear waste processing, deep underwater work and space missions on space stations or moons, rocks and planets of the solar system. Other economic areas where self-maintenance is useful but less vital are agriculture, food production and manufacturing of goods. In two of the first three crucial applications the detection and modelling of radiation levels in the robot's environment are of vital importance.

The radiation protection module class has the purpose of sensing and computing special radiation models to inform the planning of the processor module and thereby protect the robot from avoidable harm. The module can also include a radiation shielding controller.

## **3.5 Qualitative measures of design choices**

Within the same class, modules can also have a variety of featured as needed in applications. For example, the locomotion module can include diverse methods mobility such as legged modules and wheeled modules. It is obvious that modules,

---

which rely on different methods, can influence attributes of modules such as their cost. Apart from methods, the choices of components and their assembly can lead to various production standards. Modules can acquire various attributes to meet requirements, while still belonging to the same class. The diversity of types within a module class can enhance the efficiency of robotic systems and can increase their chance of survival with limited cost through optimised designs based through optimised designs based on simulated missions.

### 3.5.1 Platform/locomotion modules

Via the variety of mechanisms for locomotion, the platform module can provide various methods such as ground motion, underwater, surface water and aerial motion. For ground motion or aerial motion, there are further diverse methods to fit with various missions. We list here some cases of robots with diverse locomotion method:

- Wheeled motion: wheeled mobility can achieve high efficiency and have a relatively simple mechanical implementation. [96].
- Legged motion: The legged robots can fit with the real-world rough environment, especially rescue missions after earthquakes or explosions including unstructured terrain and obstacles [97].
- Caterpillar motion: It is another motion method to overcome unwanted terrains and maintain an ideal speed in soft terrain such as desert, which is not efficient for large obstacles compared with legged robots [98].
- Snake-like motion: provides motion for flexible robots with a small cross-section to length ratio, which allows them to enter and operate in confined spaces [99].
- Fixed-wing motion: It is appropriate for covering large areas [100] [101].
- Multi-rotor motion: It is always deployed in narrow places.

---

### 3.5.2 Power modules and components

Battery modules can be classified by their battery type, weight and battery capacity (the total energy it can store), which fit with different missions according to energy consumption needs. Batteries with high capacity can offer more working time for robotic systems, but their price per weight can be higher.

- Fuel cells: A fuel cell is an electrochemical cell that converts chemical energy from a fuel into electricity through an electrochemical reaction.
- Generator systems: Compared with other power sources, such as the batteries, the gasoline offers high energy densities. It means that a generator system can be developed for the power source of robots to transfer the energy of gasoline to electrical power.
- Batteries: These tend to be the standard power sources for mobile robots. Lithium-ion Batteries, lead-acid and alkaline batteries are commonly used as power sources to supply energy.

### 3.5.3 Processor modules

In the processor modules computation can be carried out by a micro-controller board, which is powerful enough for most calculations in robot modules. Differences among processor types are their capacity and speed of computation, connectivity to sensors, to communication components and to power amplifiers.

### 3.5.4 Communication modules and components

The application area to large degree determines the type of devices and the way of communication between modules.

- Bluetooth: Bluetooth is a radio frequency cable with a short distance to replace the unlicensed technology with 2.4GHz bandwidth in the scientific industry. Compared with other communication equipment, Bluetooth have low cost and low power consumption. However, because of the 10 meters communication range limit and 1MB speed, Bluetooth is not an

---

appropriate method for some intelligent robots. On the other hand, the NXT robot with Bluetooth communication method has been reported to have good performance [74], which means Bluetooth communication of MRS is feasible in the line (chain) topology, ring topology and tree topology of some components.

- WLAN: wireless local area network belongs to the unlicensed radio frequency, mostly operated at 2.4GHz or 5.1GHz . The cabled 802.11n LAN network has powerful download or upload speed (50mbps to 1000mbps and now above). By the antennas, the WLAN can exceed a range of 30km.
- 4G with GPS: Fourth generation (4G) mobile devices and services can transform wireless communication into online, real time connectivity. The remote control system can utilize the hybrid communication method (4G and GPS). The system contains the terminal, the monitor system and the network for data transfer. Dependency on 3G networks restricts the working area, which is not reliable in some long-range missions.

### **3.5.5 Manipulator modules and parts**

Manipulator modules are complex combinations of components, which have different efficiency and accuracy such 2 DOF manipulators and 7 DOF manipulators. Moreover, the manipulator class can be classified into different types depending on their standards and costs. Their efficiency is an important parameter too.

### **3.5.6 Components for active radiation protection**

Active radiation protection has different types and solutions to meet different requirements derived from the level of danger in extreme environments. For example, if the working location is near reactors, the quality and protection of modules must be updated and ensured to support normal operation of robots.

Even though most modules have a relatively complete functionality compared with those in non-self-maintaining robot, we still call them as modules rather than robots, unlike in re-configurable robotics. A robot is represented by a configuration

---

of several modules. The basic unit is modules instead of components. Replaceable components with plug and play qualities are named modules too.

## **3.6 Conclusions**

In the chapter the problems of self-maintaining robots have been formulated and solved by a mixture of methods. Compared with old designs, such as modular robotics systems, the SMR theory, which was introduced here, has the potential to find a wide range of applications with high reliability, including those in space, in nuclear waste disposal and in planetary missions. Furthermore, our theory offers a better platform to deploy multiple functionalities to cope with requirements in harsh environments. To demonstrate the theory, this chapter builds a simple model described both qualitatively and quantitatively. This model will be applied and extended in the rest of the thesis.



# Chapter 4

## System Reliability Models

Based on prior system reliability theory and methods [84] [87], this thesis proposes its own methods and models to estimate the likelihood of systems malfunctioning during a given time period in Sections 3, 5 and 6. Derived from the calculation of mathematical models, lifetime, failure rate and even management of resources can be used to enhance robotic performance. This offers a new meaningful development approach for robotic systems through heterogeneous modular robotic systems. Apart from their use in design, these models can also be applied in an SMR system's decision making in the interest of self-maintenance. Here a probabilistic model is presented, which permits the optimisation of redundancy allocation in an SMR system.

### 4.1 Problem description

The problem of SMR system reliability needs to address two relevant aspects of self-maintenance: the first is structural reliability in terms of hardware redundancy. The second is functional reliability in terms of ability to reconfigure.

- Structural Reliability
- Functional Reliability

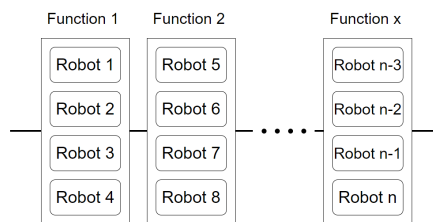
Most of the mathematical models in this chapter contribute to structural reliability for redundancy allocation. Based on these, designer can choose a module's reliability from their available range.

## 4.2 Structural models

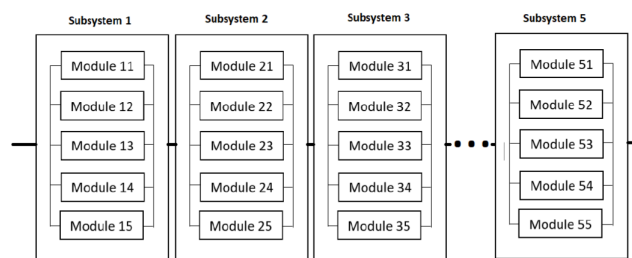
Robotic systems can include group levels and single robot levels. Every robot in a robotic group is a configuration of modules from a common range, while the robotic team can include of different robots. This arrangement can be regarded as a series-parallel system from reliability theory. The two-layer series-parallel system is considered for structural redundancy for its simplicity:

- Robot team: consists of a group of robot subgroups  $R_{gr}$  as in Fig. 4.1 .
- Single robot: composed of modular structure with functional redundancies  $R$  as in Fig. 4.2 .

**Figure 4.1** Robotic group system  $R_{gr}$  in serial-parallel structure



**Figure 4.2** Single robot  $R$  in serial-parallel structure



At the robotic team level, a series of robot subgroups, with similar capabilities, can provide redundancy as in Fig. 4.1 . For a single robot, groups of similar modules can provide redundancy for a series of functional capabilities as in Fig. 4.2 .

### 4.2.1 Robot and module redundancies

To increase reliability, most complex systems can adopt redundancy technology [102] . Redundancy techniques involve the application of both unit redundancy

---

and component redundancy. In our case unit-redundancy is robot-redundancy and component redundancy refers to redundancy of functional modules within a robot.

The robotic system  $R_{all}$  is composed of robots  $R_j$ , with  $m_i$  number different modules.

### 4.2.2 Reliability deficiency/failure detection

In most cases switching means that after due to an internal request, a cold standby component replaces a formerly active component in the event of the failure of the active component. However, the possible failure of switching to a cold standby component also needs to be analyzed for the likelihood of detection, which introduces the concept of switching reliability. In SMR theory switching reliability also covers the detection likelihood, which means the replacement reliability can be regarded as an integrated parameter of replacement reliability and detection reliability. Furthermore, switching has some inherent complexity due to possible module replacements across a group of robots, not only within a single robot.

The reliability of switching of a component with index  $i$  will be denoted by probability  $p_i$ . The associated prior probability of the detection of component failure will be denoted by  $p_i^d$ . It is however a practical simplification to limit detection probability to the ability of the module to self diagnose, which clearly is an under-estimate of the true value but easier to determine. Complete module failure is not an obstacle of detection if the approach taken is to use liveness signals for all modules, collected by all other modules onboard a robot. Detection of a whole robot's failure can be detected by other robots due to dropping its broadcasted liveness signals.

The switch reliability of a component with index  $i$  will be denoted by probability  $p_i^s$  and  $p_i^o$ , which represent detection/switch reliability of the same robot and detection/switch reliability of the whole robotic system.

### 4.2.3 Capacity of robots

Each robot has an upper-limit  $w$  for the number of associated modules for cold standby. These cold-standby modules, however, do not need to be physically

---

carried by each robot. It is more efficient if they can pick them up from a shared module station, which could also be regarded as a special type of "robot" with some particular configuration in its system reliability.

#### 4.2.4 Cold standby redundancy and active redundancy

In reliability studies, we find two mainstream strategies: cold standby and active redundancy [103]. In the application of the active redundancy method, all modules would start operating simultaneously and failed modules are replaced in realtime by the others. As they would need to be physically carried by robots this arrangement would be likely to reduce power efficiency of robot operations.

In cold standby modules are only started to be used when they are needed. This way modules are protected from operational stress so that no redundant module can fail before it is used. In many robotic application for extreme environments, there is no need for realtime, fast replacement, unlike in aviation on flying aircraft. It is overall likely that cold standby redundancy can increase the survivor chance of SMRs over a long time periods, compared with active redundancy strategies.

### 4.3 Cost of maintenance

In this section maintenance tasks are classified into three different types as in general system reliability theory [102].

1. *Corrective maintenance (CM)* is executed after a module malfunctions.
2. *Preventive maintenance (PM)* is a planned maintenance strategy when an item is activated and replaced regularly to prevent future failure.
3. *Failure-finding maintenance (FFM)* is a special type of preventative maintenance that covers functional and operational diagnosis to search for the next module to be replaced.

PM and CM will be adopted for self maintaining robots to reduce the failure rate of the whole system and extend the survival time for a team of robots. The cost of maintenance is introduced to evaluate the cost of replacement of modules according to PM and CM respectively. FFM is outside the scope of this work.

---

The SMR systems can mainly rely on corrective maintenance. It is effective where cost is more important than no interruption of work. For robotic systems deployed in a dangerous situation such as nuclear plants, the interruption of replacements, by gathering help from other robots, can potentially produce instability of operations, which can be more costly than robot maintenance. The likelihood of this happening is needed to be assessed before deciding on maintenance strategies.

### 4.3.1 Corrective maintenance

The purpose of corrective maintenance is to restore a module back to a functioning state as soon as possible by substituting the failed module/sub-module by a cold standby module. This is also called breakdown maintenance or run-to-failure maintenance. For simplicity, we assume that for a single module  $i$ , the probability of failure up to time  $t$  follows an exponential probability distribution  $F_i(t) = 1 - e^{-\lambda_i t}$ . Hence the mean time to failure (MTTF) for module  $i$  is:

$$MTTF(i) = \int_0^{\infty} e^{-\lambda_i t} dt = \frac{1}{\lambda_i} \quad (4.1)$$

$\beta_i(t) = \lambda_i t$  is the mean value of the number of failures during an interval  $t$ . The average (mean) cost of maintaining module  $i$  in a type  $v$  robot by replacement over a long time period is

$$Ac_{iv}^{av}(t) = \frac{\beta_i^v(t)\gamma_{iv}}{t} = \lambda_i \gamma_{iv} \quad (4.2)$$

where  $\gamma_{iv}$  is the cost of one maintenance.

The cost  $Ac_{iv}$  can be applied in most cases in SMR applications, when the robotic system is working in time-relaxed applications or the time of self-maintenance is negligible.

---

### 4.3.2 Preventative maintenance

Preventative maintenance aims to reduce the probability of failure of a module. Inspection, adjustments, lubrication, parts replacement, calibration and repair at times are applied to follow a maintenance policy. An active module of type  $i$ , or its parts, are replaced by cold-standby modules periodically at times  $t_0^i, 2t_0^i, 3t_0^i, \dots$ . Planned cost and unplanned cost play a crucial role in the cost of maintenance. Compared with planned cost, unplanned cost is more disruptive and complicated, which is related with factors such as problems of investigation after robot malfunction and its rescue solution.

## 4.4 Component value and importance

To define SMR theory's self-maintaining strategy, a parameter called 'component value' needs to be introduced. 'Importance' refers to an important parameter to analyse the efficiency of modules for missions, which only depend on the design of the module regardless of the environment.

Both parameters play a vital role to support the self-maintaining process and algorithm used.

### 4.4.1 Component value evaluation

- The component value evaluation and comparison are only used for the same class's modules. Different types of modules cannot be compared for their value.
- If one functioning module is more important than another module of the same class for the present mission, it has a higher value.
- A malfunctioning module acquires the lowest value during the range of component values.
- A *decisive module* in a robotic system is one that if lost, would lead to the breakdown of a whole robotic system. The modules have the highest value.

- 
- Component values of modules can be updated by a robot team by their AI-based computing ability (planning and inference of consequences of possible events) and dependent on variable mission goals.

#### 4.4.2 Importance value evaluation

- The importance value only depends on the design of the component or module itself, which is a constant value after the mission start. It represents a functioning module's significance for the robot's intended missions.
- If working the efficiency of one module is higher than of another one, the importance of this module is also higher than the other one's.

### 4.5 Quantification of reliability

System reliability with cold standby strategy, and perfect switching of modules, has been derived by Coit [84] for series-parallel systems. When applied to the  $j$ -th robot, it gives

$$R_j(t) = \prod_{i=1}^m (r_i(t) + \sum_{k=1}^{n_i-1} \int_0^t f_i^k(u) r_i(t-u) du) \quad (4.3)$$

where  $r_i(t)$  refers to the reliability of module  $i$  by time  $t$ ,  $f$  represents the probability density function,  $m$  is the number of module types indexed by  $i$  on a robot and  $n_i$  is the number of cold standby modules of type  $i$  available to the robot. The above formula only applies to perfect switching and needs to be modified to

$$R_j(t) = \prod_{i=1}^m (r_i(t) + p_i(t) \sum_{k=1}^{n_i-1} \int_0^t f_i^k(u) r_i(t-u) du) \quad (4.4)$$

where  $p_i(t)$  represents the detection/replacement reliability for module type  $i$  on a single robot, assuming detection is always made if a module fails.

A group of robots, as opposed to a single robot, has one great advantage when cold standby redundancy is applied: if they are structured similarly, they can share

---

the same pool of modules to pick their replacements from others. The shared structure of each robot can then be optimised with consideration to the reliability of robots helping each other in module replacements when one breaks down. Of course, it is also allowed that robots are able to utilise the cold redundancy from other robots without the same structure to maintain themselves in some emergency cases.

A group of robots does not need to be homogeneous as in practical applications various types of robots may need to work together. We assume in our reliability calculations that there are  $V$  kinds of robots, with type indices  $v = 1, 2, \dots, V$ . In a team of robots, the number of type  $v$  robots will be denoted by  $v_l$ .

#### 4.5.1 Full functional requirement

For some robot deployments it can be a requirement that all members of a group of  $v_l$  robots of type  $v$  needs to remain operational to fulfil a mission. It means that even if one robot fails, the whole robotic system would malfunction.

Therefore, the system reliability of robots with type  $v$  without cold-standby redundancy is

$$R^v(t) = (R_j(t))^{v_l} \quad (4.5)$$

#### 4.5.2 Minimal functional requirement

This is the case when the robot group  $v_l$  remains still usefully functioning until all robots of type  $v$  fail. In this case, the failure of a robotic system can be defined by all robots malfunctioning. It means that the structure of all robots in type  $v$  is parallel. The probability that at least one robot of type  $v$  does not malfunction, out of  $v_l$ , is:

$$R_\mu^v(t) = 1 - (1 - R_j(t))^{v_l} \quad (4.6)$$

If the operational condition is that at least one robot needs to function from each type, then the probability of this happening is

$$R_\mu(t) = \prod_{v=1}^V [1 - (1 - R_j(t))^{v_l}] \quad (4.7)$$



---

### 4.5.3 Partial functional requirement

A special case is when at least  $K$  robots need to survive out of total  $v_l$  type  $v$  robots for them to remain functionally useful. It represents that the robotic system can suffer a limited loss of robots less than or equal to  $v_l - K + 1$ . When the  $K$  is equal to 1, the robotic system represents a fully functional structure.

The probability of this can be calculated by

$$R_K^v(t) = \sum_{k=K}^{v_l} \binom{v_l}{k} (R_j(t))^k (1 - R_j(t))^{v_l - k} \quad (4.8)$$

### 4.5.4 Reliability with cold standby redundancy

Using modules kept on cold standby, a serial-parallel structure emerges, where the survival probability  $r_i^v(t)$  can be updated to a higher probability. If  $m_i$  modules are available from module type  $i = 1, 2, \dots, m$  on cold standby, then the formula

$$R_j^v(t) = \prod_{i=1}^{m^v} (r_i^v(t) + p_i(t) \sum_{k=1}^{m_i^v - 1} \int_0^t f_i^k(u) r_i(t - u) du) \quad (4.9)$$

is applicable to the requirement that robot  $j$  remains functional up to time  $t$ , where  $p_i(t)$  is the probability of detection/switching success for module type  $i$ , which we also call the reliability of combined detection and switching. These two possibilities can however be separated as shown later.

Using the above formulae for each subgroups of robots of the same kind, the full functional reliability with spare modules can be worked out as:

$$R(t) = \prod_{v=1}^V \prod_{j=1}^{v_l} R_j^v(t) \quad (4.10)$$

for the requirement that all robots need to remain fully functional.

The probability for minimal reliability, which requires that at least one robot functionally survives from each type  $v$ , is:

$$R_\mu(t) = \prod_{v=1}^V (1 - (1 - R_j^v(t))^{v_l}) \quad (4.11)$$

---

The reliability of partial functional survival for each type of robot can be computed by products of (4.8) for each robot type.

Coit [84] presented reliability formula for cold-standby redundancy strategy with the use of exponential and Erlang distributions (cf. Ardakan [104]). Underlying equations (4.4)-(4.11) are homogeneous Poisson processes, hence the  $i$ -th factor of (4.9) becomes:

$$R_{ji}(t) = r_i(t) + p_i(t) \sum_{k=1}^{n_i-1} \frac{e^{-\lambda_i t} (\lambda_i t)^k}{k!} \quad (4.12)$$

where the  $R_{ji}$  represents the reliability of module  $i$  in robot  $j$  and  $\lambda$  refers to the fault rate in the exponential distribution.

Based on (4.12), if  $m_i$  denotes the total number of spare modules of type  $i$  and  $p_i(t)$  is the detection and replacement probability, then the *reliability of robot  $j$*  is defined by

$$\tilde{R}_j(t) = \prod_{i=1}^m \left( r_i(t) + p_i(t) \sum_{k=1}^{m_i-1} \frac{e^{-\lambda_i t} (\lambda_i t)^k}{k!} \right) \quad (4.13)$$

which is the probability of the event that no more than  $m$  modules have been replaced from module type with index  $i$  in the robot group.

The reliability of module  $i$  at time  $t$  following exponential distribution :

$$r_i(t) = e^{-\lambda_i t} \quad (4.14)$$

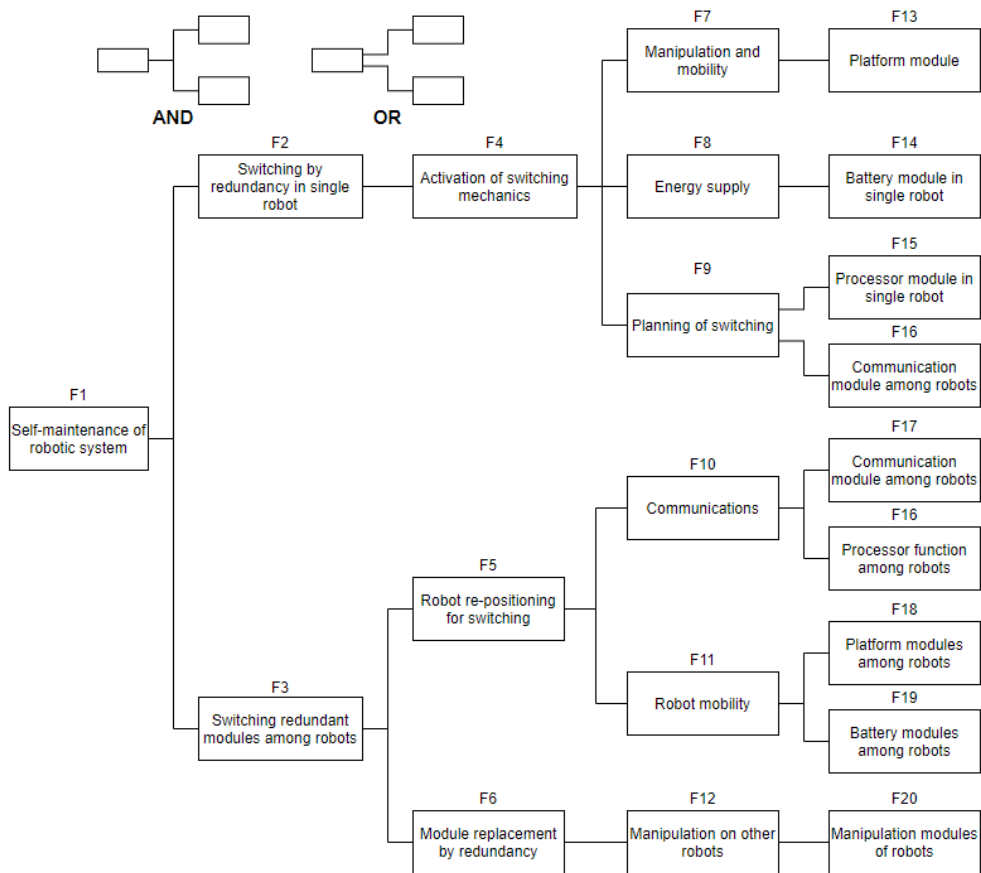
Where  $\lambda_i$  is the average rate of failure for module  $i$  over a long period : for low  $\lambda_i$  the module is likely to survive for long time.

## 4.6 Types of module switching

A robot can possibly replace some of its modules by itself and some only by help of others. The reliability of the execution of switching module  $i$  on its own, or by another robot, will be denoted by  $p_i^s(t)$  and  $p_i^o(t)$ , respectively.

There are a number of factors affecting switching reliability. A possible hierarchical decomposition of basic functionalities affecting switching are presented in Fig. 4.3. For the assessment some assumptions are made.

**Figure 4.3** Tree-graph of self-maintenance. F1-F6 are activities for self-maintenance, F7-F12 are capabilities needed and F13-F20 are modules supporting the capabilities.



**Assumptions:**

- Redundancy is activated by cooperation of the platform module (locomotion module) and the processor module.
- Switch reliability is defined by the reliability of switching to redundant module(s).
- The switch reliability is only dependent on the set of active modules.

In this SMR theory, self-maintenance splits into two main problems: switching redundancy on one robot and switching redundancy among a set of robots. Reliability of switching can be used to derive a solution for overall reliability assessment.

Switch reliability on a single robot is primarily dependent on module redundancy on a single robot, while reliability of a team of robots depends on the overall availability of redundant modules in the team.

---

The main modules, for which reliability is a key issue, are: platform modules (locomotion modules), manipulator modules, battery modules, communication modules, processor modules and active self-protection modules (e.g. radiation protection modules). Their reliability functions will be respectively characterized by  $r_l(t), r_m(t), r_b(t), r_c(t), r_p(t), r_a(t)$ . The reliability of a robot for its primary functions is

$$\tilde{R}_j^v(t) = R_{jl}^v(t)R_{jm}^v(t)R_{jb}^v(t)R_{jc}^v(t)R_{jp}^v(t)R_{ja}^v(t) \quad (4.15)$$

Communication activation includes the functioning of at least one of the processor and communication modules for diagnosis, reliability of the robot groups remaining capable of reporting about its conditions is

$$\tilde{R}_{jca}^v = R_{jc}^v(t)R_{jp}^v(t) \quad (4.16)$$

Reliability of a robot remaining able to move and ready for switching is

$$\tilde{R}_{jlbpc}^v = R_{jl}^v(t)R_{jb}^v(t)R_{jp}^v(t)R_{jc}^v(t) \quad (4.17)$$

as the locomotion, battery and processor modules are needed for planning and for control of any motion execution.

Reliability of remaining manipulation capability can be obtained by

$$\tilde{R}_{jbm}^v = R_{jb}^v(t)R_{jm}^v(t) \quad (4.18)$$

## 4.7 Conclusions

In this chapter structural reliability and functional reliability have been derived in terms of mathematical models. First the cost of maintenance over a long period has been addressed. Following the fundamental theory and architecture introduced for SMR theory, probabilistic models have been derived to evaluate subsystem reliability of the robotic system. Finally, using the scheme outlined block diagrammatically, a mathematical model has been developed to assess overall system reliability. These derivations contribute to the next chapter by enabling a proper statement of the redundancy allocation problem and thereby

---

allowing optimisation of the configuration of the robotic system before missions.

# Chapter 5

## Redundancy Allocation Problem

Before the missions start, the user or engineer must decide the robot types and redundancy allocation to ensure the chance of survival by the robotic system during a long time period without human intervention. System reliability with cost is introduced to predict the reliability against cost for the whole system. Optimization method is necessary to solve an NP hard multi-objective optimisation problem and obtain the best redundancy allocations before the deployment starts.

### 5.1 Problem description

In the previous sections, cold standby redundancy was used to enhance reliability. Other factors affecting the designer choice are price and quality of modules. The problem of optimising design of self maintaining robots has at least two main objectives: enhancing reliability and keeping running costs to a minimum. Reliability does however affect long term average maintenance costs through the need of replacements.

From the above descriptions and calculations, the system reliability of single robot by various switching reliabilities can be updated:

$$R_j^{vx}(t) = \prod_{i=1}^{m^v} (r_i^v(t) + p_{ji}^x(t) \sum_{k=1}^{m_i^v-1} \int_0^t f_i^k(u) r_i(t-u) du) \quad (5.1)$$

where the  $p_i(t)^x$  can refer to either the case that the robotic module of type  $i$  is

---

replaced by other robots, for instance on a robot  $j$ ; or that robot  $j$  replaces the module on its own. These are denoted by  $p_i^o$  and  $p_i^s$ , respectively.

Then the reliability of single robot is dependent on two different parts: (1) redundancy in own storage and (2) redundancy amongst robots. So the complete function of single robot reliability can be represented as:

$$\tilde{R}_j^v(t) = \prod_{i=1}^m (1 - (1 - R_{ji}^{vs})(1 - R_{ji}^{vo})) \quad (5.2)$$

As derived above, the reliability of the group of  $L$  robots of  $V$  types is

$$R_\mu(t) = \prod_{v=1}^V (1 - \prod_{j=1}^{v_i} (1 - \tilde{R}_j^v(t))) \quad (5.3)$$

is to be maximised at a time  $t$  of intended functional life of the robots.

The total cost (price of all hardware) for the whole system is

$$C_\mu = \sum_i^m c_i m_i \quad (5.4)$$

where  $c_i$  is the cost of a component (or module)  $i$  and  $m_i$  refers to the number of modules of type  $i$ , including active modules and redundant ones, in the whole system.

As mentioned above, it may not be efficient for the robots to use a single repository of spare modules. If their working area spreads over large distances, then it is uneconomical and can cause prolonged interruption of work to travel far to pick up modules on cold standby. Spare modules are also impractical to be carried around by robots. A midway solution is that each robot is allocated its own storage shed for spare parts, which is nearby to where the robot works. To be economical while providing the highest level of reliability, some constraints

$$w_{ji} \leq w_{ji}^{limit}, \quad j \leq L \quad (5.5)$$

are set, where  $w_{ji}$  is the number of modules of type  $i$  for robot  $j$  in its own storage and  $w_{ji}^{limit}$  refers to the maximum number of robot  $j$  is allowed to keep in storage. An alternative is to be non-specific about the upper limits for each robot and to

---

use the general constraints

$$w_{ji} \leq w_i^{limit}, \quad j \leq L, i \leq m \quad (5.6)$$

On the other hand, each robot have a limited storage for active modules and redundancy, it also defined by:

$$\sum_i^m w_{ji} \leq w_j^{limit}, \quad j \leq L, i \leq m \quad (5.7)$$

Note that

$$m_i = \sum_{j=1}^L w_{ji} \quad (5.8)$$

and the distribution of  $w_{ji}$  within  $m_i$  does not affect the reliability formula (5.3). According to equation (4.13), if robots are limited to use their own storage only, then their reliability needs to be modified to

$$R_{ji}^w(t) = r_i(t) + p_i^s(t) \sum_{k=1}^{w_{ji}-1} \frac{e^{-\lambda_i t} (\lambda_i t)^k}{k!} \quad (5.9)$$

which then provides a more specific

$$R_{all}^w(t) = \prod_{j=1}^L (1 - \prod_{i=1}^{w_{ji}} (1 - R_{ji}^w(t))) \quad (5.10)$$

that can be a much-reduced reliability level but with reduced costs of switching to spare modules.

In calculation and simulation, the choice of constraints and reliability function is according to the situation. But in later examples, it is assumed that the robots are working together without any flocking problem, which different modules, including active and cold-standby, utilize the same redundancy pool from equation (5.7) .



---

## 5.2 Complexity issues

Without using the local storage quota  $w_{ji}^{limit}$ , the number of evaluations for cost-reliability pairs is

$$X = \prod_{i=1}^m m_i^{limit} \quad (5.11)$$

where  $m_i^{limit}$  is the maximum practical number possible for each module type  $i$ . Alternatively, with the use  $w_{ji}$ , the complexity is

$$X^w = \prod_{i=1}^m \prod_{j=1}^L w_{ji}^{limit} \quad (5.12)$$

which can be significantly higher than (5.11).

In a practical implementation of self-maintaining robots working in an isolated area, it is expected that spare modules are regularly provided to robots by their human supervisors. If the supervisors maintain the optimised redundancy level, then they will keep up the associated reliability level. Under such assumptions, reliability and cost of modules can be jointly applied in the computation of the time-averaged continuous running cost of the robot team, which then becomes

$$C_{cont} = L \sum_{i=1}^m c_i / \lambda_i \quad (5.13)$$

As in many future applications, the robots are deployed for a fixed period with the number of cold standby modules to be determined, the optimisation problem is a multi-objective one, balancing reliability against overall costs. The complexity of this problem is NP-hard in terms of the number of robots as the variations of cases to be evaluated are as in (5.11) or (5.12) .

## 5.3 Optimisation methods

### 5.3.1 Alternative optimisation methods

The NP-hard optimisation for self-maintenance can be solved for a low number of robots by discrete evaluation of all cases. For the large number of robots and

---

modules, however, alternative optimisation methods are needed. As an example, next we present the application of an evolutionary optimisation algorithm.

### 5.3.2 Use of evolutionary optimisation

Deb [105] developed a non-dominated sorting-based multi-objective EA called NSGA-II. Their aim was to replace the original NSGA algorithm, which had a number of shortcomings such as computational complexity and lack of elitism. We found NSGA-II particularly effective for optimisation that involves two objectives.

In NSGA-II, the population is initialized first. The population is ranked depending on non-domination in each front by 1, 2 and so on. The first front is a set with completely non-domination individuals, while the second one is dominated by the individuals in the first front only, and this domination relationship carries on recursively. Each front is assigned a fitness measure to rank all individuals, which is called the 'crowding distance'. The crowding distance is used to evaluate distances between individuals and their neighbours. A larger average crowding distance can represent a better diversity in the population. A binary tournament selection process is run, which is based on the rank and crowding distance to pick up parents. An individual is selected for its lower rank than others or for its crowding distance being better than that of others. Finally, the offspring and current population is sorted again by dependence on non-domination and only the population of fixed size  $N$  are selected to the next generation. The algorithm fits well the two-objectives problem, which strikes a 'trade-off' mechanism between cost and system reliability. This is the main reason for us using NSGA-II with its 'trade-off' property rather than the MOEA/D with weight vector in [106].

## 5.4 Calculation and results

Derived from chapter 4, there are two groups of data that need to be solved and optimized. Each case has its own classes and types but utilise the same structure - serial-parallel structure. The space complexity and time complexity

---

**Table 5.1:** Data used in the modules

$m_i$	1	2	3	4	5	6	7	8	9
$c_i$ (pounds)	2000	2300	2400	200	230	280	400	420	450
$\lambda_i$ (minutes)	0.0031	0.0032	0.0034	0.0050	0.0052	0.0057	0.0034	0.0036	0.0037
$m_i$	10	11	12	13	14	15	16	17	18
$c_i$ (pounds)	300	310	360	1600	1700	1900	800	820	870
$\lambda_i$ (minutes)	0.0021	0.0022	0.0023	0.0012	0.0013	0.0016	0.0076	0.0077	0.0079

**Table 5.2:** Robot Configuration without redundancy

Robot number	Total slot	Free Slot	Configuration Code
1	6	2	1,4,7,10
2	6	2	1,3,8,16
3	6	3	1,5,8
4	6	3	1,4,7
5	6	5	2
6	6	4	3,9

are different among these cases. These cases reflect various working environments for application of the SMR theory in practice.

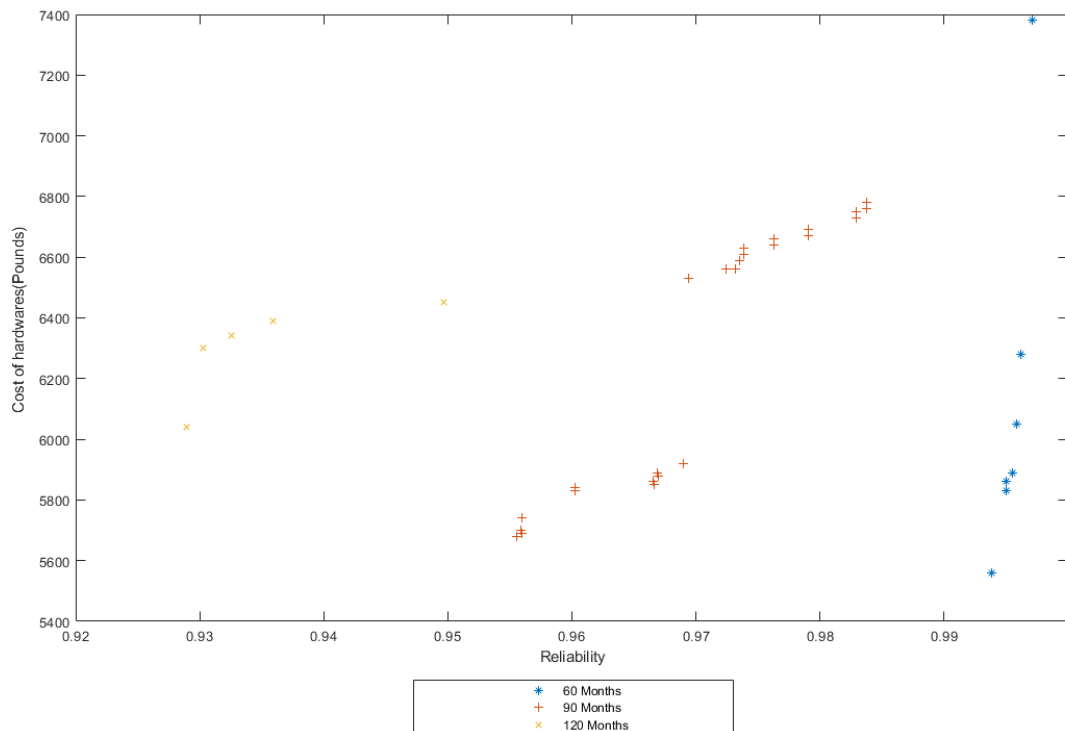
#### 5.4.1 Illustrated example

This example is a fundamental one and simple with limited number of classes and types. It means that space and time complexity are constrained. Assume that a robotic system has 6 robots of types  $v \in \{1, 2, \dots, 6\}$  and that each individual has 6 modules with redundancy limits  $w_j^{max} = 6$ . We also assume that only one robot is used from each type, so that  $v_l = 1, l = 1, \dots, 6$ . The replaceable core modules are: the platform module, battery module, processor module, manipulator modules, telecommunication module and active radiation prevention module. Each module has three quality types with varying system reliability and costs. A simple exponential function is used for the distribution of failure times, and known  $\lambda$  rates of failure are used to define the system reliability functions for each module.

Based on the assumptions above, a group of results have been calculated in MATLAB as in Fig. 5.1. The  $X$  coordinate displays the system reliability beyond time  $t$ , whereas the  $Y$  represents the cost of the whole system (cost of hardware). The points (star, plus and cross) in Fig. 5.1 show the non-dominated area in three different working times (60,90,120 minutes). In Fig. 5.1, the blue stars indicate

---

**Figure 5.1** Multi-optimization figures of test problem using NGS-II



the system reliability of a robotic system for 60 minutes, close to 1 and with a low cost of reconfiguration. Then, as the working time is increasing (to 90 minutes and 120 minutes), though the system reliability of the robotic system is weakening, it is still higher than the 0.9, which means that the whole system is working well for 120 minutes after the missions started. It is illustrated here that with a cold-standby strategy the robotic system always maintains a reasonably good working state for a long term period.

### 5.4.2 Comparison between fixed configuration and self maintaining robots

In this section, the robotic system with SMR theory (*RSS*) is compared with the robotic system with normal redundancy design (*RSR*) to extend the experiments. Both designs have the same pool of classes and types with the same parameters including cost and system reliability. Moreover, all redundancy relies on cold redundancy. The only difference is in the application of the SMR approach taken. Only the *RSS* switches cold redundancy modules from different robots to maintain the failure, while the *RSR* utilises the cold redundancy

---

located at the same robots to repair failed modules. Therefore, the switching reliability and redundancy specifications are different. Furthermore, for convenient representation, the cost and time would not adopt units such as months or days.

The shared characteristics of RSS and RSR:

- Pool of classes and types.
- Classes and types with the same parameters, reliability and cost.
- Reliability functions and mathematical models.
- Identical environmental and external influence.
- Full functional structure.

The differences between RSS and RSR:

- Switching reliability: RSS is able to use  $p_o$  and  $p_s$ . However, the RSR only uses  $p_s$ .
- Redundancy allocation.

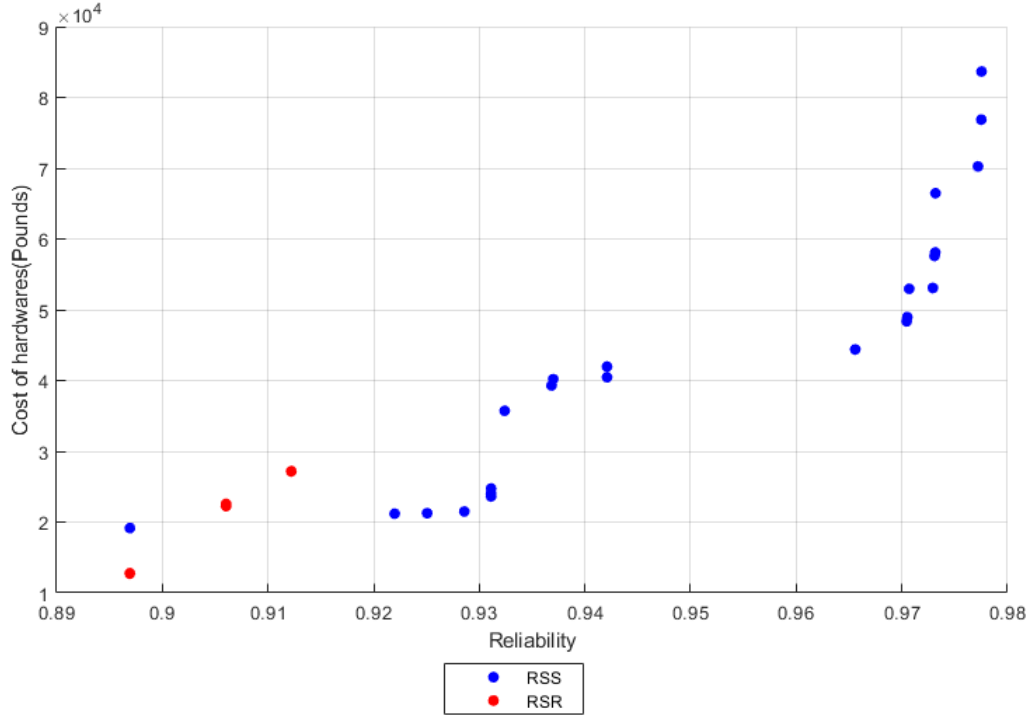
For multi objective optimisation, some assumptions are made:

- The switch time is neglected no matter whether switching is from the same robot or from other robots.
- The parameter change is neglected when switching is applied in different robotic systems.
- The execution difference is neglected within the same missions.
- Both RSS and RSR are assumed to operate within any mission.
- For the chance of survival, all spare capacity of robots would be used up by keeping redundancy levels high.

The information and configuration of modules is derived from Case 2 in the Section 4.4.2. Fig 5.2 illustrates an example at time unit 1300. It indicates the advantage of *RSS* at time 1300 compared with *RSR*, which is visible on the right hand side

of the figure as *RSR* does not appear there due to higher cost of hardware and low reliability. Because the comparison of results has only been derived from figures such as Fig 5.2, it is necessary to adopt a new tool to evaluate the outcome of calculations more accurately.

**Figure 5.2** Comparison between RSS and RSR at 1300 time unit from multi objective optimisation



#### 5.4.2.1 Use of the hypervolume indicator

The hypervolume indicator is a widely used set-quality indicators for multi-objective optimisation, which is to measure the area occupied by the set and bounded above by a given reference point or point set. It is regarded as 'size of the space covered' [107] [108] and 'size of the dominated space' [109].

*Definition of hypervolume* [110]: Given a point set  $S \subset \mathbb{R}^d$  and a reference point  $r \in \mathbb{R}^d$ , the hypervolume indicator of  $S$  is the measure of the area "over"  $S$  and bounded from above by  $r$ :

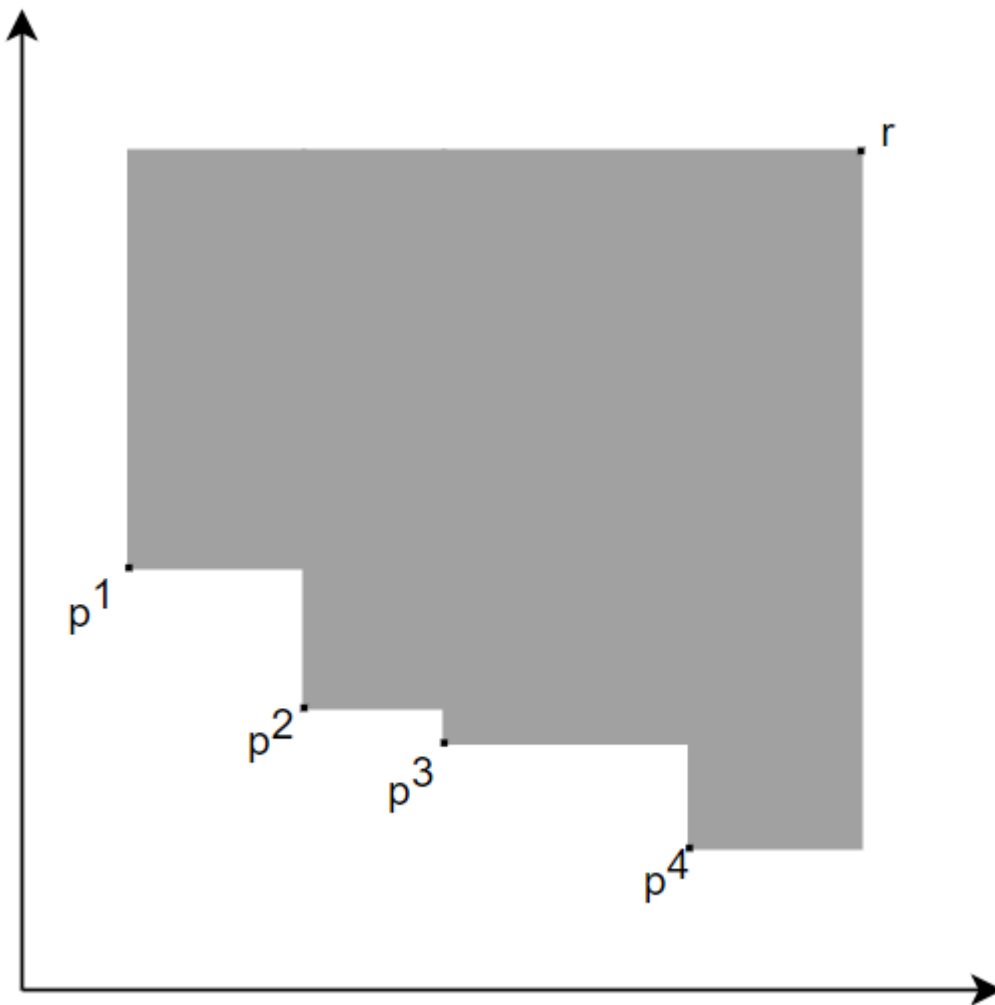
$$HV(S, r) = \Lambda(q \subset \mathbb{R}^d \mid \exists p \in S : p \leq q \text{ and } q \leq r) \quad (5.14)$$

where  $\Lambda$  denotes the Lebesgue measure and  $p \leq q$  means  $p_i \leq q_i$  for all  $i$ . The

---

Fig. 5.3 demonstrates the calculation process. The point set  $S$  represents the results derived from calculations by the redundancy allocation algorithms, in which each point  $p$  have two parameters, including cost and system reliability with  $X$  and  $Y$  coordinates, in the hypervolume graph 5.3. Furthermore, the hypervolume indicator is an approach for assessing multi-objective optimization algorithms, which evaluate the optimizer outcome by simultaneously taking into account the proximity of the points to the Pareto front, diversity, and spread.

**Figure 5.3** Hypervolume indicator (grey region)



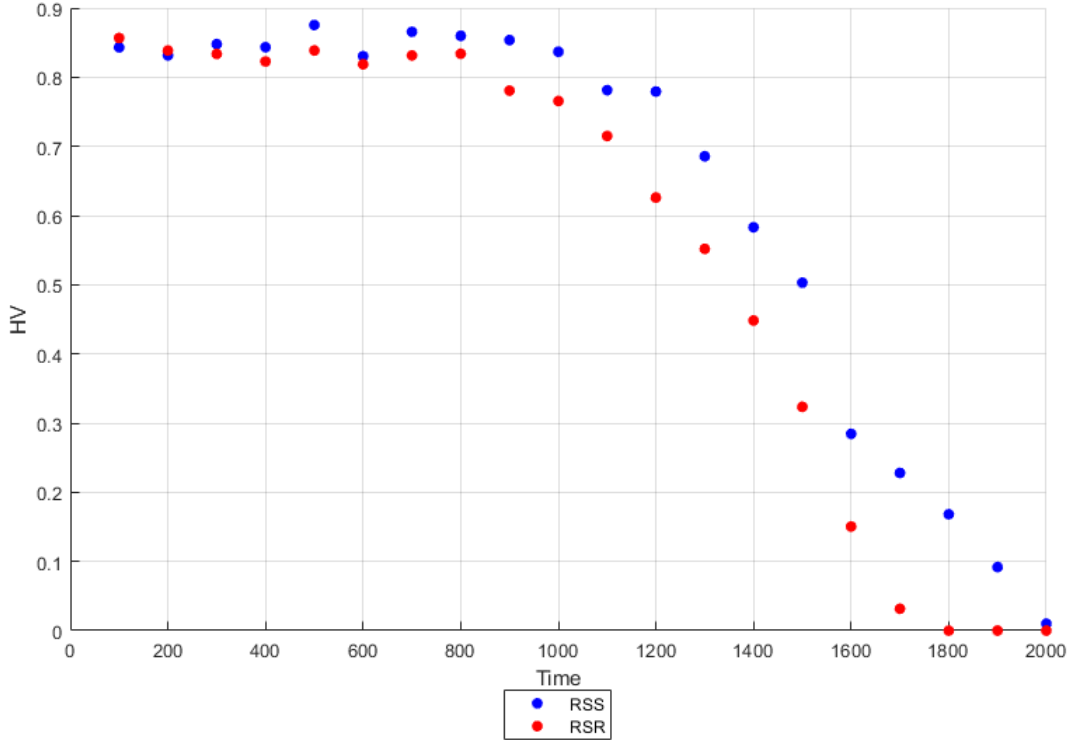
The advantage of the indicator is convenient to recognise, whose main drawback is computational cost. Depending on the HV, a statistical comparison between  $RSS$  and  $RSR$  on a particular test problem can be designed as follows.

**Table 5.3:** HV results (mean) of RSS and RSR by time

Time		100	200	300	400	500	600	700	800	900
HV	RSS	0.8432	0.8315	0.8478	0.8433	0.8754	0.8303	0.8656	0.8598	0.8537
	RSR	0.8567	0.8386	0.8336	0.8227	0.8387	0.8186	0.8314	0.8339	0.7806

1000	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
0.8368	0.7812	0.7792	0.6856	0.5830	0.5030	0.2844	0.2279	0.1682	0.0918	0.0100
0.7655	0.7150	0.6260	0.5518	0.4483	0.3235	0.1505	0.0316	0	0	0

**Figure 5.4** HV results (mean) of RSS and RSR from time 100 to time 2000



#### 5.4.2.2 Working time

In this case, the maximum and minimum range for the HV indicator are decided and maintained in a pair of fixed values for all calculations including *RSS* and *RSR* because of the fixed quantity of active modules and redundancy. From Fig 5.4, it is obvious that the advantage of SMR theory is widened by time increasing. Firstly, when a robotic system just starts or the time of missions is low, the old structure's robot *RSR* has a little superiority for SMR theory. Moreover, the SMR theory' robots (*RSS*) get back the control of HV, which maintains the preponderance compared with *RSR* over a long period. At the same time, the decline of *RSR* is accelerated. Finally, the HV becomes zero at the 1800 time unit, which means the system lost function at a high change. It illustrates that the SMR theory



---

**Table 5.4:** The HV results (means) of the quantity of redundancy

Number of Additive Redundancy		50	100	150	200	250
HV	RSS(mean)	0.8453	0.8871	0.8326	0.9866	0.9942
	RSR(mean)	0.7132	0.7839	0.7999	0.7662	0.6009
	Ratio of HV	0.8436	0.8836	0.9607	0.7766	0.6044

is powerful with a long working time, moreover, the gap is deepened if time is increased.

#### 5.4.2.3 Quantity of redundancy (Capacity)

Her another experiment is presented for the quantity of redundancy. Different from last section, due to the rise of the amount of redundancy, the maximum and minimum values for HV, and especially the cost, would not be kept in fixed values for all tests. This is so because the change of redundancy influences the configuration of robots. In that case, we pick up only 5 groups of conditions to calculate. The Ggap for each group of redundancy is 50. The ratio of HV is the value that HV of *RSS* divided *RSR*. From table 5.4, it is clear that, as the number of redundancy increases, results of *RSS* become better and produce a larger advantage compared with *RSR*.

### 5.4.3 Ratio of switch reliability

In this section, the thesis will discuss the influence of switching reliability. A ratio between  $p_o$  and  $p_s$  are introduced to discover the effect of switch reliability.

#### 5.4.3.1 Fixed configuration

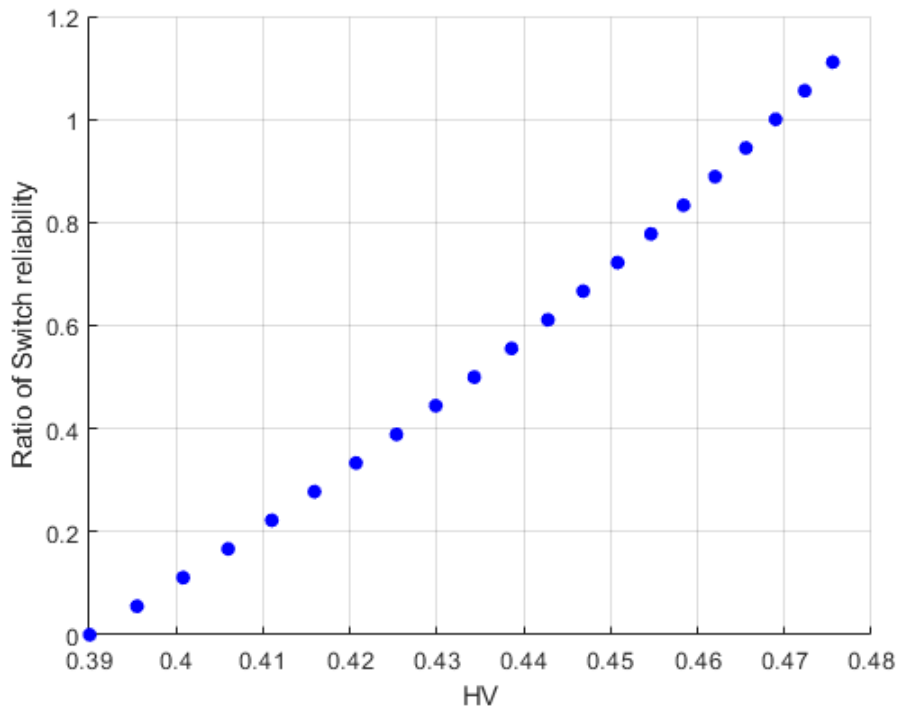
To control the variables of the experiments, the fixed configuration is utilized so that the only variable is the ratio of switch reliability of other robots, so that all calculations only depend on one group of configuration or chromosome. By Table 5.5 and Fig. 5.5, it is obvious that the ratio of switch reliability - ratio of  $p_o$  and  $p_s$  - can influence the performance of robotic systems. Moreover, the last ratio of switching reliability 0, in the table, refers to the *RSR* according to the fundamental theory. Higher ratio of switch reliability always represents better performance and more choice for robotic systems.

**Table 5.5:** The HV results (mean) derived from Ratio of switch reliability (fixed configuration)

Ratio of switch reliability	1.1111	1.0556	1.0000	0.9444	0.8889	0.8333	0.7778	0.7222	0.6667
HV(mean)	0.4756	0.4723	0.4690	0.4656	0.4620	0.4584	0.4550	0.4508	0.4469
HV (Normalization)	1	0.8723	0.7550	0.5698	0.7786	0.5370	0.3363	0.2310	0.4525

0.6111	0.5556	0.5000	0.4444	0.3888	0.3333	0.2778	0.2222	0.1667	0.1111	0.0556	0
0.4428	0.4386	0.4343	0.4299	0.4253	0.4207	0.4159	0.4110	0.4060	0.4008	0.3955	0.3901
0.4162	0.4094	0.5506	0.4369	0.2363	0.1405	0.5973	0	0.3874	0.3441	0.0609	0.1913

**Figure 5.5** HV results (mean) related with ratio of switch reliability (fixed configuration)



### 5.4.3.2 Uncertain configuration

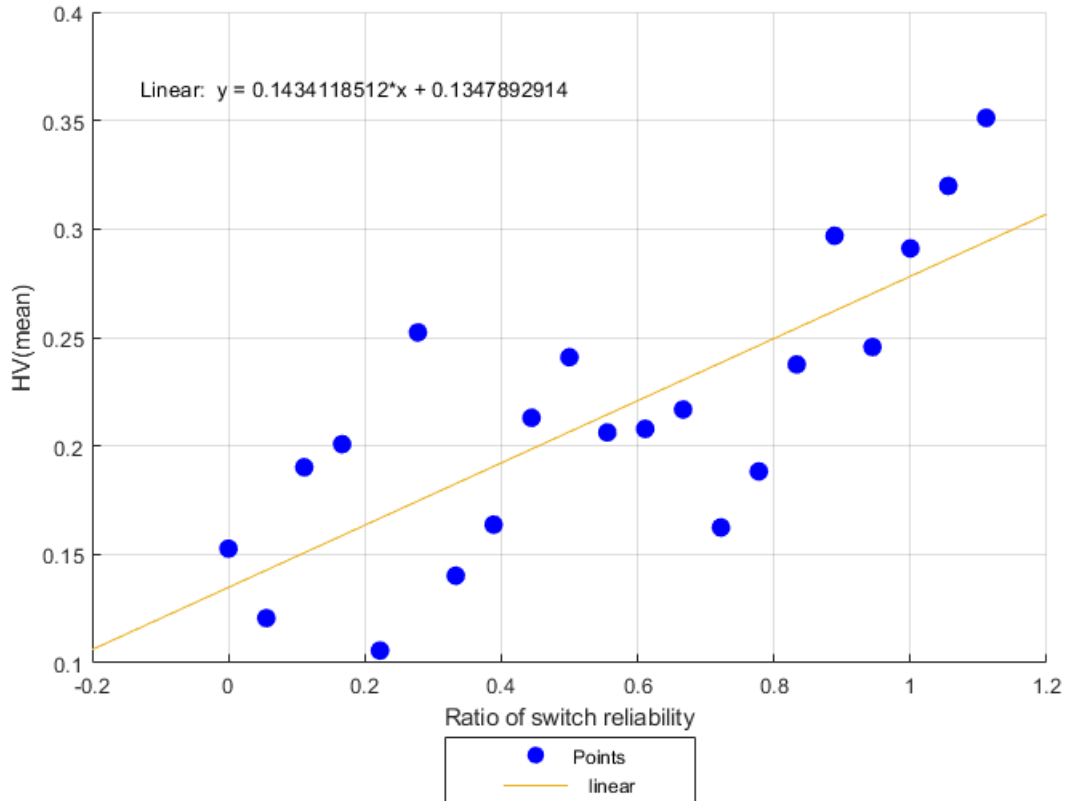
In this part, the input variable is the ratio of switching reliability without fixed configuration. It means that the robotic system can reorganise its redundancy to adapt to the new ratio of switch reliability. From Table 5.6 and Fig. 5.6, the fitting line visualises the main trend of HV by ratio of reliability, which show that the higher ratio refers to better performance in most cases. Unsurprisingly, the results scatter irregularly compared with fixed configuration. Scattering of points is related to the factor that the algorithm re-allocates the redundancy to deal with the low ratio of switch reliability problem.

**Table 5.6:** The HV results (mean) derived from Ratio of switch reliability (uncertain configuration)

Ratio of switch reliability	1.1111	1.0556	1.0000	0.9444	0.8889	0.8333	0.7778	0.7222	0.6667
HV(mean)	0.3512	0.3198	0.2910	0.2456	0.2969	0.2375	0.1883	0.1624	0.2168
HV (Normalization)	1	0.9621	0.9231	0.8830	0.8417	0.7991	0.7554	0.7104	0.6641

0.6111	0.5556	0.5000	0.4444	0.3888	0.3333	0.2778	0.2222	0.1667	0.1111	0.0556	0
0.2079	0.2062	0.2409	0.2130	0.1637	0.1402	0.2523	0.1057	0.2008	0.1902	0.1206	0.1527
0.6165	0.5676	0.5174	0.4657	0.4127	0.3583	0.3024	0.2450	0.1861	0.1256	0.0636	0

**Figure 5.6** HV results (mean) related with ratio of switch reliability (uncertain configuration)



#### 5.4.4 Conclusions

It can be seen from the experiments that the time, redundancy and the switching reliability all play vital roles in SMR theory. The results can illustrate the merits of the theory presented for long working times and various redundancy structures, which means robots with more hypervolume or capacity can perform better. Moreover, if the robotic system acquires a higher ratio of switch reliability or a good switch reliability in the same robots, the theory has a high chance to work more perfectly and offer more configuration choices for robots, which increases the reliability with lower cost.

# Chapter 6

## Task Allocations for Module Replacements

In this chapter we will focus on the self-maintaining process, especially on the module replacements. Apart from system reliability, module replacement is another crucial part to illustrate the merits of SMR theory. Firstly, the principle and stipulation of module replacement would be decided. The thesis divides the module replacement process into three different steps. Then, according to the principle of module replacement, the multi robot task allocation can be developed and solved.

### 6.1 Principles of replacement

This section will introduce the principles of replacement. Even though the thesis has discussed the replacement in the above sections, the SMR theory also needs a complete and systematic principle to describe the allocations for replacements.

- Component value: the redundant module, which is utilized to replace the malfunctioning module, must have a lower component value.
- Importance value: The module ready to replace the old one must have the same or higher importance value to continue the job after module replacement.
- Consumption: The energy consumption of the robotic system in the self-

---

maintaining process should be limited.

- Effect: The robotic systems should eliminate the effect and damage caused by the self-maintaining process and have the ability to continue the mission.
- Priority: At some extreme and emergency conditions such as explosives, the robotic system would repair the self-maintaining related modules and robots first.
- Supply: When the robotic systems do not have enough redundancy to maintain the full function status and continue missions, the robotic system would only maintain the necessary self-maintaining ability to wait for the new supply.

## 6.2 Deciding to replace

It is a vital step to analyse the necessity of replacement before the self-maintaining process, so we introduce it as a separate discussion before the procedure of replacements. Following checking the necessity, the robotic system would analyse whether or not to start the replacement process. The decision is associated with the robotic system and missions.

- Fault tolerance (effect): The robotic system should analyse the effect of the failed modules. If a failed module does not influence the accomplishment of missions with limited and acceptable results such as high energy consumption and low efficiency, the robotic system can neglect the failure or postpone the self-maintaining operation, then continue to work. But if the failed module or self-maintaining process can lead to an unacceptable degradation or lost ability for jobs, the robotic system must start the self-maintaining process immediately.
- Energy: The robotic system must analyse the energy consumption for self-maintaining processes and missions. If residue energy is not enough for module replacement or the rest of the jobs, the robotic system must neglect the failure or suspend it.

- 
- Redundancy: The robotic system must have enough redundancy to replace the failed modules.
  - Component value: The redundancy or active module ready to replace the failed one would have lower component value.
  - Importance: The predicted new configuration after the self-maintaining process must have enough 'importance' to execute the rest of the jobs without a timeout.
  - Compatibility: The module can be compatible with old configuration and related modules such as connections between modules with stable energy, information and mechanical load transmission.

## 6.3 Procedure of replacements

This section introduces the procedure of replacements to organise the behaviour of robotic systems after the decision of self-maintenance to be carried out. It covers the preparation stage, operation stage and finishing stage. However, the procedure excludes some extreme conditions.

### 6.3.1 Preparation stage

This stage is to inspect the fault and evaluate the characteristics of failure, then offer a plan to solve it. It is a final step to determine the quality of planned self-maintenance in a mission.

- Fault inspection and analysis: The robotic system inspects the fault and characteristics of the fault, assessing the quantity and type of unusual modules with evaluation.
- Deciding to replace or not: Depending on the situation and resources, determine a time in the calendar for replacement.
- Isolation: The failed modules would be isolated from the present robotic system to protect degradation.

- 
- Planning (multi-robot task allocation): The robotic system calculates and disseminates the replacement strategy for the robotic system.
  - Synchronisation: The robotic system publishes a strategy to the robots involved in each step.

### **6.3.2 Operation stage**

In the operation stage, robots would fulfil the replacement in as determined.

- Accepting order: All related robots must accept the orders from the central server.
- Flocking: Robots would move and flock at a specific position depending on the calendar.
- Replacement: The robots would replace the old module with redundancy to solve the problem.
- Check: After replacement, the robotic system must analyse the outcome of the repair. If the repaired robots still malfunction, the robotic system would return to the preparation stage for remedy or abandon.

### **6.3.3 Completion stage**

If the outcome of self-maintenance is a success, the robots return to their original positions and continue their jobs. If not, the robotic system would restart the self-maintaining process from the stage or require the remote support from engineers.

## **6.4 Task description**

To the module replacement task allocation (MRTA) problem, there are two important points that should be introduced for designing the algorithm. The first point refers to realise the characteristic of the tasks with the environment. Moreover, another point refers to an understanding of the robots' capability, including robotic mobility and configuration [111].

---

The overall goal (task) is to maintain the robotic system with as little cost as possible, which can be a compound task as well. A compound task presents a task that can be decomposed into a group of simple or compound subtasks, which only have one fixed full decomposition method. Following our theory, the decomposition of each objective is to be fixed. Because of the quantity of failed robots, the overall goal is decomposable, represented by a group of a single tasks (subgoal).

Here a single task represents one robots' problem, whatever how many failed modules found in it. Similarly, with the overall goal, in general, the single task can also be regarded as a complex task, that is, each task can be divided into elemental tasks, including moving to the target position and manipulating modules. Furthermore, here the self-maintaining problem belongs to the cross-schedule dependencies (XD), which presents task allocation problems for which the agent-task utilities have inter-schedule dependencies for complex tasks. It means that the calendar of one robot influences not only its own efficiency but also other robots' schedules.

The decomposition of overall goal and task should be:

- Overall goal: maintain the robotic system to a status required from missions.
- Task: repair all failures of one robot
- Elemental task 1: arrive at the objective position.
- Elemental task 2: offer the redundancy.
- Elemental task 3: manipulate the module and redundancy.

The capability of robots always represents the status and function of robots. The status of the robot can cover the position, energy remaining, speed and energy cost, while the robotic function refers to the ability to support the self-maintenance. The importance value and component value are also two crucial parameters to influence the capability.

For a single robot, the status and function include

- Original position: presents the coordinate of the robot before the self-maintenance operation.



- 
- Energy remains: how much energy remains in the battery.
  - Power: energy consumption .
  - Locomotion speed: the speed of robots on one or more terrains.
  - Configuration: characteristics of modules installed on the robot.
  - Function: represents the function of the robot and its importance value.

Apart from the two main points, the constraints of robots always play a vital role in the problem. For example, the redundancy must have a lower component value than the failed module. And the component value of the module must meet the requirement of missions and self-maintaining operation

- Component value: decide legitimacy of one replacement by comparison of component value derived from redundancy and the failed module.
- Importance value: determines whether or not a new module can take the place of a failed one in work efficiency for the rest of the mission.

Figure 6.1 illustrates an example of a self-maintenance process. In the mission area there are 6 robots, including 4 functional robots and 2 failed robots. The goal is to repair these failed robots depending on other robots. The task or sub-goal is to maintain one failed robot so that there are two tasks, also can be represented by atomic tasks. The Figure 6.1 shows the calendar that the robot *A1* and robot *A4* attend the maintenance of robot *B1*, then the *A4* continues to repair *B2* with *A2*, when the robot *A3* has not been involved.

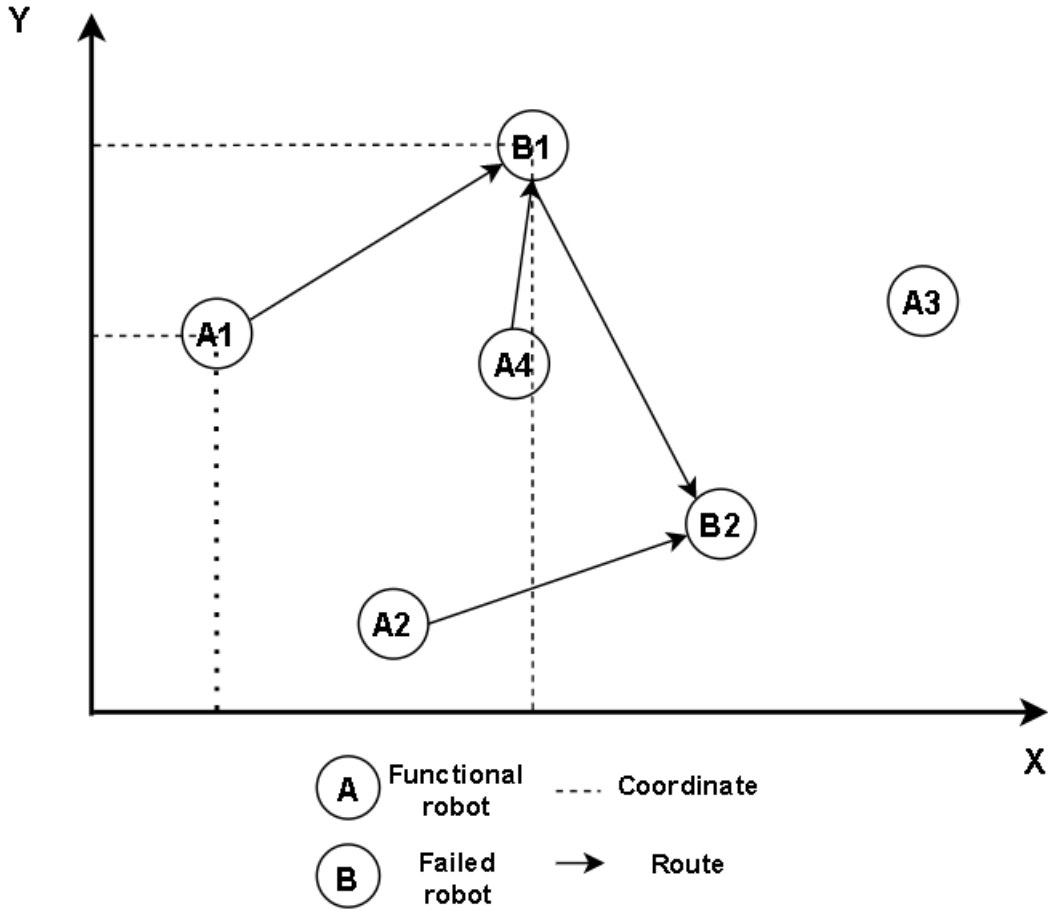
The purpose of MRTA is to find out the optimised calendar for flocking problems due to task, capacity and constraints.

## 6.5 Mathematical model of the allocation process

### 6.5.1 Capacity and resources

Depending on the SMR theory, a related multi-robot mission allocation environment is introduced, which also presents a flocking process. The

Figure 6.1 Mission area and robots



innovation of the algorithm and model description are appropriate for the SMR theory, especially for modular solutions with SMR theory. In the ST-MR-TE , the SMR assignment consists of a set of robots  $SR$  and a set of tasks  $ST$ , which is denoted as  $SA = \{SR, ST\}$ . Because the ST-MR-IA can be described as fragments of ST-MR-TE [112]. It is said that  $SA_n = (SR, ST_n)$  where  $SA_n \in SA$  and  $ST_n \in ST$  represents an assignment  $SA_n$  of any task  $ST_n$ . In ST-MR-IA, a assignment solution for any task  $ST_i$   $f : P_1 \rightarrow ST_i$  and objective function is  $TT = \max(TT_{A_i})$ .

Furthermore, in algorithm coding, the SMR robotic system called  $RS$  has  $n$  heterogeneous non-failure robots (units or combination of modules) such that  $u_1, u_2, u_3, \dots, u_n$ . So the robotic system is written:

$$RS = \{u_1, u_2, u_3, \dots, u_n\} \tag{6.1}$$

---

where  $u_i$  represent the robot/unit  $i$  and  $n$  refer to the quantity of robots/units. Moreover for one robot  $u_i$ , there are four values implemented to illustrate the status of robots including the 2-D coordinates  $(x_i, y_i)$ , energy remains  $D_i$  and the modules configuration status  $m_i$ . Then the relation can be expressed:  $u_i = \{(x_i, y_i, D_i \text{ and } m_i) : x_i, y_i \text{ and } D_i \in N\}$ . In the robot  $u_i$ , the modules situation and assembly is represented by the array  $m_i$ . In the array, it represents the modules class number  $h_j$ , component value (weight)  $v_j$  and importance  $im_j$  in the whole robotic system, where the  $j$  refers to the number of modules in this modules' array. Then it is illustrated like:

$$m_i = \{(h_1, v_1, im_1), (h_j, v_j, im_j), \dots, (h_{n_j}, v_{n_j}, im_{n_j}), \quad (6.2)$$

$$h_j \in N, v_j \in (0, 10) \text{ and } im_j \in (0, 10)\}$$

where  $n_j$  refers to the quantity of modules in robot  $u_i$ .

To add, task or malfunction robots illustrated by  $ST$  with  $n$  heterogeneous failure robots such as  $t_1, t_2, t_3, \dots, t_n$  which almost have the same demonstration with non-failure robots. Only difference is that in  $ST$ ,  $m$  refers to a broken module instead of a functional module. It is assumed that the malfunction robots would be isolated and not support the self-maintaining.

### 6.5.2 Objective function and constrains

This problem falls under the ST-MR-TE type of distribution with a centralised architecture. It is assumed any robot can only operate one self-maintenance job at a time, but a self-maintaining mission may need more than one robot at a time. The algorithm is to acquire the most efficient allocation and scheduling for the problem with the lowest sum of travel time for all robots. To resolve the problem, the objective function and constraints of multi-robots is developed for the allocation and scheduling of tasks for each robot to guarantee the efficiency of self-maintenance.

In self-maintenance of robot  $u_i$  assumes that a schedule  $S$  is denoted as  $S = \{s_1, s_2, s_3, s_4, \dots, s_n\}$ , which presents the array of schedules for the whole system,

---

where  $s_i = \{t_1^i, t_2^i, t_3^i, \dots, t_n^i\}$ . The  $s_i$  performs the schedule of robot  $u_i$ , where  $t_z^i$  represent mission  $z$  which robot  $i$  participated. The  $z_i$  refers to the quantity of missions the robot  $i$  involved.

For example, if one robot is in a non-negative and symmetric function  $D(u_i, t_j^i)$  denotes the travel distance between the original position (robot  $u_i$ ) and destination (mission  $t_j^i$ ), when  $D(t_{j-1}^i, t_j^i)$  illustrate the travel distance between two missions for robot  $i$ . So after self-maintaining missions, the robot will also come back or stay at original position to continue the rest of job before the self-maintaining process. For a robot who has been decided to support self-maintaining missions from the schedule  $s_i$  and the travel distance  $TD_i$  for robot  $u_i$  is supposed by

$$\begin{aligned}
TD_i &= 0 \text{ subject to } z_i = 0 \\
&= D(u_i, t_1^i) + D(t_1^i, u_i) \text{ subject to } z_i = 1 \\
&= D(u_i, t_1^i) + D(t_{z_i}^i, u_i) + \sum_{j=2}^{j=z_i} D(t_{j-1}^i, t_j^i) \text{ subject to } z_i \geq 2
\end{aligned} \tag{6.3}$$

If the robot  $i$  have not acquired any allocation for self-maintenance, the travel distance  $TD_i$  is equal to 0. Moreover the travel time denoted by  $TT_i$  is described by:

$$TT_i = \frac{TD_i}{sp_i} + \sum_{j=1}^{j=z_i} Tw(t_j) \text{ subject to } z_i \geq 1 \tag{6.4}$$

The  $sp_i$  represents the speed of robot  $u_i$  and the  $Tw$  represent the total waiting time for task  $z$ , which means the time between the robot reaching the  $z$  position and starting the self-maintaining process. If the robot  $u_i$  doesn't participate in any self-maintaining task ( $z_i=0$ ), the waiting time would be zero. The purpose of the multi-robot allocation problem is to find the appropriate allocation and scheduling solution for the whole system. Hence the travel time of the robotic system  $TT_{total}$  is given by:

$$TT = \max_{i \in \{1, \dots, n\}} TT_i \tag{6.5}$$

In addition, to form a feasible calendar, executability constraints must be satisfied. It is assumed that the redundancy is named by  $Re$  and quantity of suitable redundancy for missions  $i$  is  $Q_{Re}^i$  with failed module  $m_i$ . Furthermore,

---

$Q_m^i$  presents the quantity of viable manipulators for mission  $i$ . So executability constraints should be:

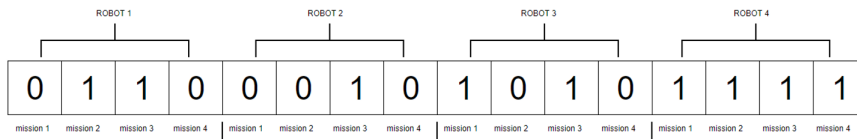
- (EC1)  $h_{m_i} = h_{Re}$ , the class of redundancy and failed module must be same
- (EC2)  $Q_{Re}^i \geq 1$ , there must be enough quantity of redundancy for replacement.
- (EC3)  $v_{m_i} \geq v_{Re_i}$  following the rule of *component value/weight* from the last chapter.
- (EC4)  $im_{m_i} \leq im_{Re_i}$  the replacement of modules must abide by the rule of *importance*.
- (EC5)  $Q_m^i \geq 1$  It means that any replacement must need a manipulator to support and any type of manipulator can help the module replacement.
- (EC6) The schedule of tasks is feasible for execution.

## 6.6 Evolutionary optimisation example

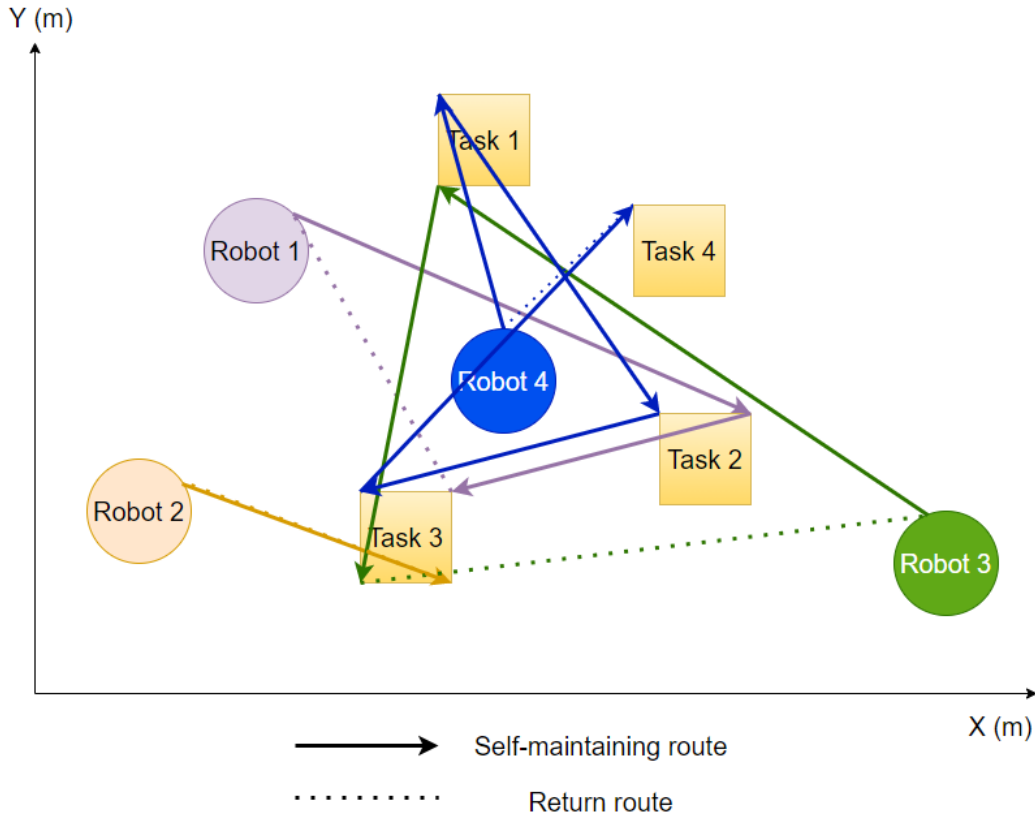
### 6.6.1 Pre-processing of data

In this problem setup the quantity of failed modules in one robot may be variable, which means that malfunctioning robots can have more than one of failed modules at the same time. Before setting up the genetic algorithm, a process should be developed and introduced to reduce the calculation burden. Firstly, maintenance of failed robots should be ordered by the 'component value' of failed modules  $v_j$ . This will decide the sequence of the self-maintaining process. If one robot with multiple failed modules, the highest 'component value' of the failed module will be regarded as the 'component value' of the failed robots:  $v_i^{failed} = \max(\sum_{j=1}^{j=n} v_j^i)$ , where the  $v_j^i$  refers to the component value of the failed module in robot  $u_i$ . The pre-processing of data decides the sequence of self-maintenance in missions.

**Figure 6.2** Chromosome representation with four missions and four robots



**Figure 6.3** Graph representation for a chromosome derived from Fig 6.2



### 6.6.2 Chromosome representation

The genetic algorithm is supposed to solve the problem in the ST-MR-IA, which the ST-MR-TE can be regarded as a combination of ST-MR-IA in taxonomy for multi-robot task allocation. Every robot covers the same length of the gene with a quantity of missions. The first four genes in the chromosome are made to represent the schedule for one robot  $u_1$ , where Boolean values refer to whether the robot  $u_1$  attends to this mission. It means that the quantity of genes is equal to the number of robots multiplied by the number of missions (failed robot) by Fig. 6.2. Furthermore, the Fig. 6.3 has shown the graph description of the schedule derived from the chromosome in Fig. 6.2. The variation in the chromosome indicates the set of robots which should take part in mission configurations. In Fig. 6.2, the

---

chromosome shows a configuration that the *robot 1* should take part in *mission 2* and *mission 3*, while *robot 2* should only operate in *mission 2*. Furthermore, *robot 3* is to join the self-maintaining job in *mission 1* and in *mission 3*. Differently from other robots, the *robot 4* is to attend all missions. In another approach the chromosome could be used to describe allocations in missions. For example, the *robot 1*, *robot 3* and *robot 4* are allocated to *mission 1*. Because the sequence of self-maintaining missions can be decided as in subsection 6.6.1, the algorithm could calculate travel time for each mission and sum them up (the total travel time of the chromosome). For example, the total travel time/working time of *mission 1* is the time that the last robot reaches the destination. Then the *mission 2* calculates the travel time based on their position in *mission 1*. In the *mission 3*, position of robots depends on the *mission 2*, in which the travel time of the last mission should also cover the return time to their original position before the mission starts. Furthermore, the travel time of the chromosome is the sum of travel times in all missions. Alternatively, depending on the chromosome and constraints, the algorithm could output the travel time for the fitness function . Fig. 6.3 shows the schedule derived from the chromosome in Fig 6.2 .

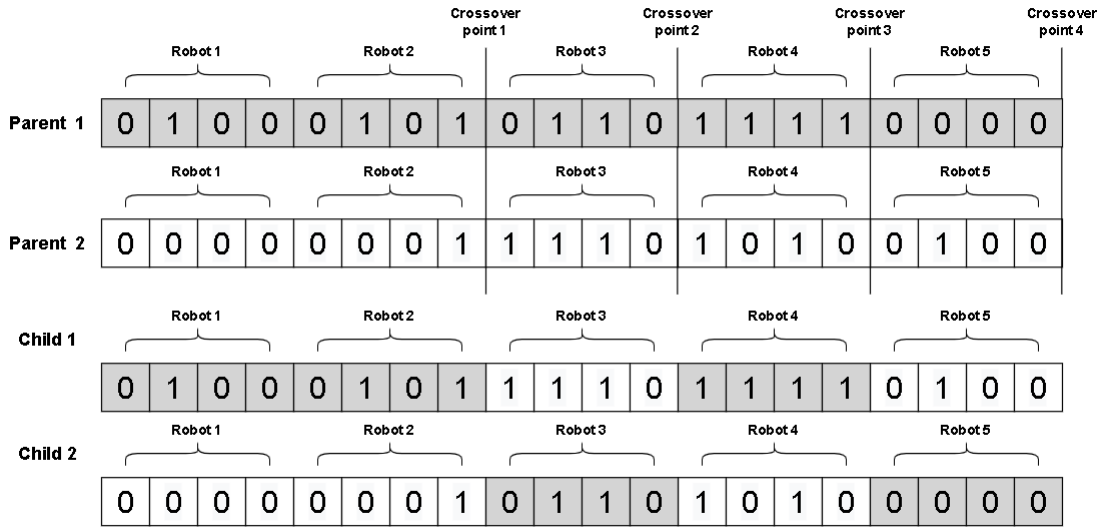
### 6.6.3 Crossover

In GA, a crossover operator enables the algorithm to produce better offspring by swapping of genes among two parents in Fig 6.4. Because of the construction of chromosome, the choice of swapping point can be located at any position in the chromosome.

### 6.6.4 Mutation

The crossover operator only reforms the construction of chromosomes but does not change any individual segment of the chromosome. So the mutation operator is used to change the specific genes. As different from the crossover operator, the mutation can happen at any position and any length without any limitation.

Figure 6.4 Crossover of 5 robots with four missions



### 6.6.5 Fitness function and penalty function

Travel time fitness function evaluates the whole system according to the allocation of tasks for each robot under constraints, which take into consideration each robot's travel distance and speed to evaluate the self-maintenance time:

$$Fitness\ function = TT \quad (6.6)$$

When the schedule violates the constraints, the penalty function is introduced:

$$Penalty\ function = TT + MAX \quad (6.7)$$

MAX is a maximum value for this problem, which could eliminate the chance of crossover for this chromosome.

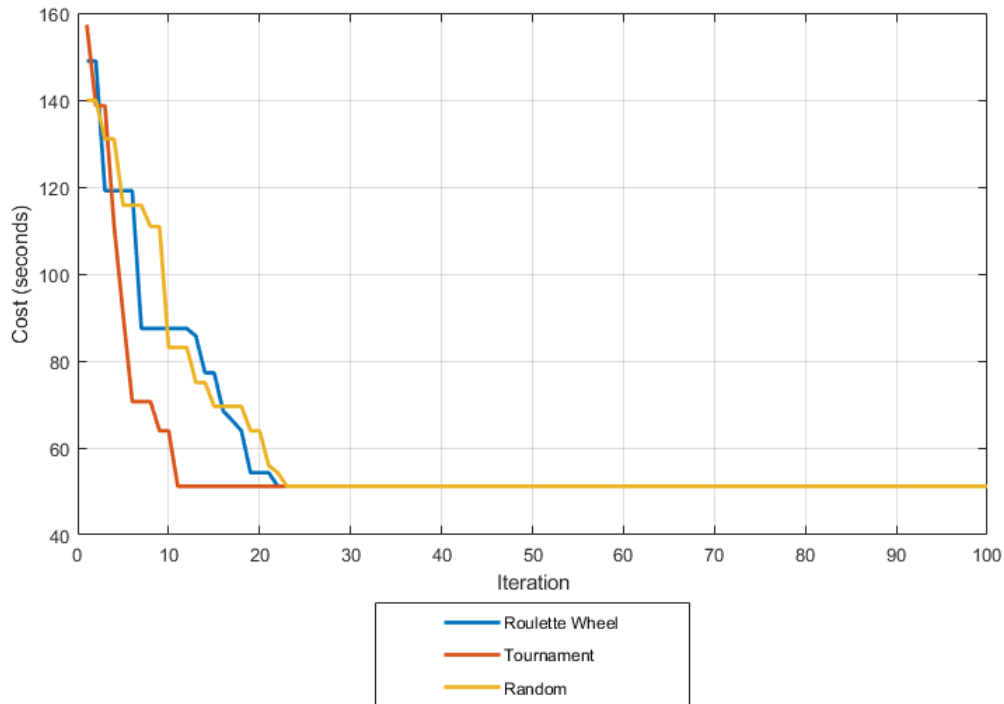
In other words, the fitness function is used to evaluate the travel time of each chromosome. For one chromosome or configuration of robots with missions, the algorithm would analyse whether the chromosome is legal, which can decide to apply some penalty accordingly. Then, if it is legal, continues with the decided self-maintaining sequence, and the fitness function would output the travel time or working time of the chromosome.



**Table 6.1:** MRTA specifications

Self-maintaining robotic system (ST-MR-TE)	
Number of functional robots	10
Number of failed robots	4
Number of missions	4
Generations	100
Population size	100
Selection method	Tournament
Crossover rate	0.8
Mutation rate	0.02
Map size	5m*5m
Number of modules' class	6
Number of modules' type	18

**Figure 6.5** Convergence of Random, Tournament and Roulette Wheel selection (cost refers to the travel time)



---

## 6.7 Results and comparisons

To validate and enhance the self-maintaining algorithm, a lot of examples have been generated. In the illustrated example, there are 10 robots which have four failed robots within a 5 meter square area. Furthermore, each failed robot only has one failed module that need to be replaced. In this example, we utilised different selection methods - *Roulette Wheel*, *Tournament* and *Random* to illustrate their efficiency for self-maintenance robots by time of convergence. The final results of three different selection methods are same (cost of time). From Fig. 6.5, it is obvious that the *Tournament* selection method has better performance , as this method converges in only 12 iterations. The time of convergence only takes half a minute by tournament selection, which is suitable for the self-maintenance operation. From the comparison, it is apparent that robot users and researchers should apply the tournament selection method in their future calculations. Apart from advice given on the selection methods to use, Chapter 6 offers a task allocation solution for multiple robots, that is especially applicable to SM robots. With the solution, developers and researcher can quickly choose the appropriate self-maintaining strategy.

# Chapter 7

## Conclusions and Future work

### 7.1 Achievements

In this thesis, we have developed some of the theoretical foundations of self-maintaining robots with the aim of assisting future robotic developments for applications where autonomous or remotely supervised robots need to work on their own, without physical contact with humans.

The best candidates for replaceable modules and components have been discussed. System reliability has been addressed from the viewpoints of structural reliability and functional reliability. The reliability of detection has been accounted for. Maintenance types have been identified as preventative (PM), corrective (CM) and fault finding (FFM) and their cost functions have been established.

Formulae have been provided for reliability over a finite time horizon for minimal and partial functional requirements. Computations for reliability under cold standby of components and modules has been presented, inclusive replacement/switching reliability for teams of robots with homogeneous and heterogeneous architectures.

Computations have been provided for long term operational capability of a team of robots over infinite time horizons. Complexity issues of design optimisation of self-maintaining robots have been addressed and an evolutionary computation example provided. One of the most important electro-mechanical components, universal connectors, for both mechanical strength and electrical reliability, have

---

been reviewed.

The basic theory presented can be refined further in individual designs of future robots in the nuclear, space, nature preservation areas, and also in dangerous environments of industrial laboratories.

Moreover, the module replacement task allocation (MRTA) has been developed to support and enhance the self-maintenance theory, which offers a new solution for resource allocations during missions.

With the efficient algorithm provided, the robotic system can avoid degradation and finish the module replacement at a lower level of consumption of resources. The main contributions of the algorithm is a chromosome representation and MRTA applications with matched condition and replacement policy.

## 7.2 Limitations

Here we present a few critical remarks regarding the work completed.

- In the redundancy allocation problem, the data for calculation is produced by simulation rather than collected from real robotic systems.
- Switch reliability only depends on the configuration of robots, without thinking of missions and status. It is obvious that the redundancy allocation problem in the Section 5.4.3.2 is affected. For example, the travel distance of robots also affects the switch reliability.
- In the MRTA problem, the solution does not support the 3D map and distribution of different terrains, which is far away from complex situations.
- Most of the results in the work rely on theoretical study, which is short of practical evidence from the realistic experiment.
- HV indicator is not effective enough for some special cases, which affect the comparison between SMR theory and old design.

---

## 7.3 Future work

As discussed in the last section, the system reliability part also needs to be updated to adapt to problems to the real world. Firstly, the robotic system is supposed to acquire a more reliable model of switching reliability, which directly affects the efficiency of SMR theory.

The new switching reliability should be evaluated as dependent on the resource allocation and missions, which cover the robot's track, the difficulty of jobs and other parameters. Then, for long periods' work, degradation is an inevitable aspect. The model should regard degradation as an important factor. Therefore, the new model could have a wider applicability not only for the SMR theory of robots but also for modular robotic systems.

Realistic case study is needed to support the redundancy allocation problem instead of the data produced from simulation. An SMR robot team needs to be built and used to validate and update the findings of the theoretical study in this thesis. The MRTA algorithm needs to be applied for their vital role in acquiring more extensions.

Finally, new simulated evaluation tools could be developed in the future to study the practical effectiveness of the SMR design methods to be applied. Simulations will also be able to provide intuition for further development of the SMR theory itself.

# References

1. Kochan, A. Robot cleans glass roof of Louvre pyramid. eng. *Industrial robot* **32**, 380–382. ISSN: 0143-991X (2005).
2. Martínez, D., Alenyà, G. & Torras, C. Planning robot manipulation to clean planar surfaces. eng. *Engineering applications of artificial intelligence* **39**, 23–32. ISSN: 0952-1976 (2015).
3. Launius, R. D. *Robots in Space Technology, Evolution, and Interplanetary Travel* eng. ISBN: 0-8018-9844-7 (Johns Hopkins University Press, Baltimore, 2008).
4. Xie, Y. *et al.* Adaptive optimal tracking control for multi-joint manipulator on space robot. eng. *Optimal control applications and methods* **41**, 1995–2007. ISSN: 0143-2087 (2020).
5. Kanai, Y. Monitoring of aerosols in Tsukuba after Fukushima Nuclear Power Plant incident in 2011. eng. *Journal of environmental radioactivity* **111**, 33–37. ISSN: 0265-931X (2012).
6. Nagatani, K. *et al.* *Redesign of rescue mobile robot Quince* eng. in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics* (IEEE, 2011), 13–18. ISBN: 9781612847702.
7. Rizk, Y., Awad, M. & Tunstel, E. Cooperative Heterogeneous Multi-Robot Systems: A Survey. eng. *ACM computing surveys* **52**, 1–31. ISSN: 0360-0300 (2019).
8. Saeedi, S., Trentini, M., Seto, M. & Li, H. Multiple-Robot Simultaneous Localization and Mapping: A Review. eng. *Journal of field robotics* **33**, 3–46. ISSN: 1556-4959 (2016).

- 
9. Behnke, S., Veloso, M., Visser, A. & Xiong, R. *RoboCup 2013: Robot World Cup XVII* eng (Springer, 2014).
  10. Zhao, X. H., Xu, J. & Zhao, L. Review of Groups-Robot Cooperation System Research Basing on Swarm Intelligence. eng. *Applied mechanics and materials* **80-81**, 1350–1354. ISSN: 1660-9336 (2011).
  11. Sharkey, A. J. C. Robots, insects and swarm intelligence. eng. *The Artificial intelligence review* **26**, 255–268. ISSN: 0269-2821 (2006).
  12. Ahmadzadeh, H., Masehian, E. & Asadpour, M. Modular Robotic Systems: Characteristics and Applications. eng. *Journal of Intelligent and Robotic Systems* **81**, 317–357. ISSN: 0921-0296 (2016).
  13. Fukuda, T. & Nakagawa, S. Approach to the dynamically reconfigurable robotic system. eng. *Journal of Intelligent and Robotic Systems* **1**, 55–72. ISSN: 0921-0296 (1988).
  14. Korsah, G. A., Stentz, A. & Dias, M. B. A comprehensive taxonomy for multi-robot task allocation. eng. *The International journal of robotics research* **32**, 1495–1512. ISSN: 0278-3649 (2013).
  15. Öztürk, S. & Kuzucuoğlu, A. E. Optimal bid valuation using path finding for multi-robot task allocation. eng. *Journal of intelligent manufacturing* **26**, 1049–1062. ISSN: 1572-8145 (2014).
  16. Wei, C., Ji, Z. & Cai, B. Particle Swarm Optimization for Cooperative Multi-Robot Task Allocation: A Multi-Objective Approach. eng. *IEEE robotics and automation letters* **5**, 2529–2536. ISSN: 2377-3766 (2020).
  17. Baca, J., Hossain, S., Dasgupta, P., Nelson, C. A. & Dutta, A. ModRED: Hardware design and reconfiguration planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration. eng. *Robotics and autonomous systems* **62**, 1002–1015. ISSN: 0921-8890 (2014).
  18. Kurokawa, H. *et al.* Distributed Self-Reconfiguration of M-TRAN III Modular Robotic System. eng. *The International journal of robotics research* **27**, 373–386. ISSN: 0278-3649 (2008).

- 
19. Agheli, M., Qu, L. & Nestinger, S. S. SHeRo: Scalable hexapod robot for maintenance, repair, and operations. eng. *Robotics and computer-integrated manufacturing* **30**, 478–488. ISSN: 0736-5845 (2014).
  20. Son, H., Hwang, N., Kim, C. & Kim, C. Rapid and automated determination of rusted surface areas of a steel bridge for robotic maintenance systems. eng. *Automation in construction* **42**, 13–24. ISSN: 0926-5805 (2014).
  21. Luo, J., Xie, S., Gong, Z. & Lu, T. Development of cable maintenance robot for cable-stayed bridges. eng. *Industrial robot* **34**, 303–309. ISSN: 0143-991X (2007).
  22. Moon, S.-M., Hong, D., Kim, S.-W. & Park, S. *Building wall maintenance robot based on built-in guide rail* eng. in *2012 IEEE International Conference on Industrial Technology* (IEEE, 2012), 498–503. ISBN: 9781467303408.
  23. Gao, X., Xu, D., Wang, Y., Pan, H. & Shen, W. Multifunctional robot to maintain boiler water-cooling tubes. *Robotica* **27**, 941–948 (2009).
  24. Luk, B. L., Cooke, D. S., Galt, S., Collie, A. A. & Chen, S. Intelligent legged climbing service robot for remote maintenance applications in hazardous environments. eng. *Robotics and autonomous systems* **53**, 142–152. ISSN: 0921-8890 (2005).
  25. Jeon, M., Kim, B., Park, S. & Hong, D. *Maintenance robot for wind power blade cleaning* eng. in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction* **29** (IAARC Publications, Waterloo, 2012), 1.
  26. Wang, Z., Cao, Q., Luan, N. & Zhang, L. Development of an autonomous in-pipe robot for offshore pipeline maintenance. eng. *Industrial robot* **37**, 177–184. ISSN: 0143-991X (2010).
  27. Jiang, W. *et al.* Live maintenance robot for high-voltage transmission lines. eng. *Industrial robot* **46**, 711–718. ISSN: 0143-991X (2019).
  28. Liu, L.-l. *et al.* Development of the insulator-replacing live working robot on transmission line. Chinese. *Journal of Electric Power, Science, and Technology* **31**. - 2017-01-01; - 2017-01-05, 121–127.



- 
- <https://search-proquest-com.sheffield.idm.oclc.org/scholarly-journals / development - insulator - replacing - live - working/docview/1855377204/se-2?accountid=13828> ( 2016).
29. Song, Y., Wang, H., Li, Z. & Jiang, Y. Vision Based Transmission Line Broken Strand Detection and Robot Behaviour Planning. Chinese. *Jiqiren (Robot)* **37**. - 2016-08-01; - 2016-08-03, 204–223. <https://search-proquest-com.sheffield.idm.oclc.org/scholarly-journals / vision - based - transmission - line - broken - strand/docview/1808100220/se-2?accountid=13828> (Mar. 2015).
  30. Wu, G. *et al.* Development of a mobile inspection robot for high voltage power transmission line. Chinese. *Dianli Xitong Zidonghua (Automation of Electric Power Systems)* **30**. - 2006-11-01; - 8; - 2011-11-12; SubjectsTermNotLitGenreText - Robots; Inspection; Power transmission lines; Management; Obstacles; Malfunctions; Transmission lines; High voltages; Electric power systems; Transformers; Infrared imaging; Machine vision; Automation, 90–93, 107. <https://search-proquest-com.sheffield.idm.oclc.org/scholarly-journals / development - mobile - inspection - robot - high - voltage/docview/28877188/se-2?accountid=13828> (2006).
  31. Murata, S. & Kurokawa, H. in *Self-Organizing Robots* 105–130 (Springer Tokyo, Tokyo, 2012). ISBN: 978-4-431-54055-7. [https://doi.org/10.1007/978-4-431-54055-7\\_6](https://doi.org/10.1007/978-4-431-54055-7_6).
  32. Seo, J., Paik, J. & Yim, M. Modular Reconfigurable Robotics. eng. *Annual review of control, robotics, and autonomous systems* **2**, 63–88. ISSN: 2573-5144 (2019).
  33. Baca, J., Ferre, M. & Aracil, R. A heterogeneous modular robotic design for fast response to a diversity of tasks. eng. *Robotics and autonomous systems* **60**, 522–531. ISSN: 0921-8890 (2012).
  34. Qiao, G., Song, G., Wang, W., Zhang, Y. & Wang, Y. Design and Implementation of a Modular Self-Reconfigurable Robot. *International Journal of Advanced Robotic Systems* **11**, 47. eprint:

- 
- <https://doi.org/10.5772/58379>. <https://doi.org/10.5772/58379> (2014).
35. Yang, Z., Fu, Z., Yu, G., Fei, J. & Zheng, H. A self-repairing approach for the M-Lattice modular robotic system using digital hormone model. eng. *Robotics and autonomous systems* **97**, 1–15. ISSN: 0921-8890 (2017).
  36. Mobes, S. *et al.* *Toward a 2D Modular and Self-Reconfigurable Robot for Conveying Microparts* eng. in *2012 Second Workshop on Design, Control and Software Implementation for Distributed MEMS* (IEEE, 2012), 7–13. ISBN: 1467312037.
  37. Guan, E., Fu, Z., Yan, W., Jiang, D. & Zhao, Y. *Self-reconfiguration Path Planning Design for M-Lattice Robot Based on Genetic Algorithm* in *Intelligent Robotics and Applications* (eds Jeschke, S., Liu, H. & Schilberg, D.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011), 505–514. ISBN: 978-3-642-25489-5.
  38. Fukuda, T., Buss, M., Hosokai, H. & Kawauchi, Y. Cell Structured robotic system CEBOT: Control, planning and communication methods. eng. *Robotics and autonomous systems* **7**, 239–248. ISSN: 0921-8890 (1991).
  39. Hirose, S., Shirasu, T. & Fukushima, E. F. Proposal for cooperative robot “Gunryu” composed of autonomous segments. eng. *Robotics and autonomous systems* **17**, 107–118. ISSN: 0921-8890 (1996).
  40. Wolfe, K., Moses, M., Kutzer, M. & Chirikjian, G. *M3Express: A low-cost independently-mobile reconfigurable modular robot* in (May 2012), 2704–2710. ISBN: 978-1-4673-1403-9.
  41. Yim, M., Duff, D. & Roufas, K. *PolyBot: a modular reconfigurable robot* eng. in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)* **1** (IEEE, 2000), 514–520 vol.1. ISBN: 9780780358867.
  42. Yim, M., Zhang, Y., Roufas, K., Duff, D. & Eldershaw, C. Connecting and disconnecting for chain self-reconfiguration with PolyBot. eng. *IEEE/ASME transactions on mechatronics* **7**, 442–451. ISSN: 1083-4435 (2002).

- 
43. Jorgensen, M., Ostergaard, E. & Lund, H. *Modular ATRON: modules for a self-reconfigurable robot* eng. in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)* **2** (IEEE, 2004), 2068–2073 vol.2. ISBN: 9780780384637.
  44. *RoGenSiD: A Rotary Plate Genderless Single-Sided Docking Mechanism for Modular Self-Reconfigurable Robots* **Volume 6B: 37th Mechanisms and Robotics Conference**. V06BT07A011 (Aug. 2013). eprint: <https://asmedigitalcollection.asme.org/IDETC-CIE/proceedings-pdf/IDETC-CIE2013/55942/V06BT07A011/4254889/v06bt07a011-detc2013-12938.pdf>. <https://doi.org/10.1115/DETC2013-12938>.
  45. Baca, J., Hossain, S., Dasgupta, P., Nelson, C. A. & Dutta, A. ModRED: Hardware design and reconfiguration planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration. eng. *Robotics and Autonomous Systems* **62**, 1002–1015. ISSN: 0921-8890 (2014).
  46. Murata, S. *et al.* M-TRAN: self-reconfigurable modular robotic system. eng. *IEEE/ASME transactions on mechatronics* **7**, 431–441. ISSN: 1083-4435 (2002).
  47. White, P., Kopanski, K. & Lipson, H. *Stochastic self-reconfigurable cellular robotics* eng. in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004* **3** (IEEE, 2004), 2888–2893 Vol.3. ISBN: 9780780382329.
  48. Neubert, J., Rost, A. & Lipson, H. Self-Soldering Connectors for Modular Robots. eng. *IEEE transactions on robotics* **30**, 1344–1357. ISSN: 1941-0468 (2014).
  49. Wei, H., Chen, Y., Tan, J. & Wang, T. Sambot: A Self-Assembly Modular Robot System. eng. *IEEE/ASME transactions on mechatronics* **16**, 745–757. ISSN: 1083-4435 (2011).
  50. Tan, W., Wei, H. & Yang, B. SambotII: A New Self-Assembly Modular Robot Platform Based on Sambot. eng. *Applied sciences* **8**, 1719. ISSN: 2076-3417 (2018).

- 
51. Gross, R., Bonani, M., Mondada, F. & Dorigo, M. Autonomous Self-Assembly in Swarm-Bots. eng. *IEEE transactions on robotics* **22**, 1115–1130. ISSN: 1552-3098 (2006).
  52. Gross, R., Tuci, E., Dorigo, M., Bonani, M. & Mondada, F. *Object transport by modular robots that self-assemble* eng. in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006* **2006** (IEEE, 2006), 2558–2564. ISBN: 0780395050.
  53. Kawakami, A., Torii, A., Motomura, K. & Hirose, S. *SMC rover: planetary rover with transformable wheels* eng. in *Proceedings of the 41st SICE Annual Conference. SICE 2002* **1** (IEEE, 2002), 157–162 vol.1. ISBN: 9780780376311.
  54. Motomura, K., Kawakami, A. & Hirose, S. Development of Arm Equipped Single Wheel Rover: Effective Arm-Posture-Based Steering Method. eng. *Autonomous robots* **18**, 215–229. ISSN: 0929-5593 (2005).
  55. Park, M., Chitta, S., Teichman, A. & Yim, M. Automatic Configuration Recognition Methods in Modular Robots. eng. *The International journal of robotics research* **27**, 403–421. ISSN: 0278-3649 (2008).
  56. Østergaard, E. H., Kassow, K., Beck, R. & Lund, H. H. Design of the ATRON lattice-based self-reconfigurable robot. eng. *Autonomous robots* **21**, 165–183. ISSN: 0929-5593 (2006).
  57. Christensen, D. *Evolution of shape-changing and self-repairing control for the ATRON self-reconfigurable robot* eng. in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006* **2006** (IEEE, 2006), 2539–2545. ISBN: 0780395050.
  58. White, P., Zykov, V., Bongard, J. & Lipson, H. Three-Dimensional Stochastic Reconfiguration of Modular Robots. eng. *Robotics today* **19**, 1. ISSN: 1544-8290 (2006).
  59. Adam Bilodeau, R. & Kramer, R. K. Self-healing and damage resilience for soft robotics: A review. eng. *Frontiers in robotics and AI* **4**. ISSN: 2296-9144 (2017).

- 
60. Kim, S., Laschi, C. & Trimmer, B. Soft robotics: a bioinspired evolution in robotics. eng. *Trends in biotechnology (Regular ed.)* **31**, 287–294. ISSN: 0167-7799 (2013).
  61. Rus, D. & Tolley, M. T. Design, fabrication and control of soft robots. eng. *Nature (London)* **521**, 467–475. ISSN: 1476-4687 (2015).
  62. Bilodeau, R. A. & Kramer, R. K. Self-Healing and Damage Resilience for Soft Robotics: A Review. *Frontiers in Robotics and AI* **4**, 48. ISSN: 2296-9144. <https://www.frontiersin.org/article/10.3389/frobt.2017.00048> (2017).
  63. Madsen, F. B., Yu, L. & Skov, A. L. Self-Healing, High-Permittivity Silicone Dielectric Elastomer. eng. *ACS macro letters* **5**, 1196–1200. ISSN: 2161-1653 (2016).
  64. Hughes, J. *et al.* Soft manipulators and grippers: A review. eng. *Frontiers in robotics and AI* **3**. ISSN: 2296-9144 (2016).
  65. Shintake, J., Schubert, B., Rosset, S., Shea, H. & Floreano, D. *Variable stiffness actuator for soft robotics using dielectric elastomer and low-melting-point alloy* in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015), 1097–1102.
  66. Dickey, M. D. Stretchable and Soft Electronics using Liquid Metals. eng. *Advanced materials (Weinheim)* **29**, 1606425–n/a. ISSN: 0935-9648 (2017).
  67. Tee, B. C.-K., Wang, C., Allen, R. & Bao, Z. An electrically and mechanically self-healing composite with pressure- and flexion-sensitive properties for electronic skin applications. eng. *Nature nanotechnology* **7**, 825–832. ISSN: 1748-3387 (2012).
  68. Lu, N. & Kim, D.-H. Flexible and Stretchable Electronics Paving the Way for Soft Robotics. eng. *Soft robotics* **1**, 53–62. ISSN: 2169-5180 (2014).
  69. Rodrigue, H., Wang, W., Han, M.-W., Kim, T. J. & Ahn, S.-H. An Overview of Shape Memory Alloy-Coupled Actuators and Robots. eng. *Soft robotics* **4**, 3–15. ISSN: 2169-5180 (2017).

- 
70. Dhillon, B., Fashandi, A. & Liu, K. Robot systems reliability and safety: A review. eng. *Journal of quality in maintenance engineering* **8**, 170–212. ISSN: 1355-2511 (2002).
  71. Brady, K. J., Anderson, R. J., Ning Xi & Tzyh-Jong Tarn. *Modular controller architecture for multi-arm telerobotic systems* in *Proceedings of IEEE International Conference on Robotics and Automation* **2** (1996), 1024–1029 vol.2.
  72. Khodabandehloo, K., Duggan, F. & Husband, T. *Reliability assessment of industrial robots* in *Proceedings of the 14th International Symposium on Industrial Robots* (1984), 209–220.
  73. Jin, Q., Sugawara, Y. & Seya, K. Probabilistic behavior and reliability analysis for a multi-robot system by applying Petri net and Markov renewal process theory. eng. *Microelectronics and reliability* **29**, 993–1001. ISSN: 0026-2714 (1989).
  74. Gao, D. & Wells, D. Robotic assembly operation reliability. *American Society of Mechanical Engineers* **28**, 65–70 (1990).
  75. Khodabandehloo, K. Robot safety and reliability. *Human-robot Interaction, Taylor & Francis, London*, 121–60 (1992).
  76. Dhillon, B. & Yang, N. Availability analysis of a robot with safety system. *Microelectronics Reliability* **36**, 169–177 (1996).
  77. Monteverde, V. & Tosunoglu, S. Development and application of a fault tolerance measure for serial and parallel robotic structures. *International Journal of Modelling and Simulation* **19**, 45–51 (1999).
  78. Monteverde, V. & Tosunoglu, S. *Effect of kinematic structure and dual actuation on fault tolerance of robot manipulators* in *Proceedings of International Conference on Robotics and Automation* **4** (1997), 2902–2907.
  79. Michaelson, D. G. & Jiang, J. *Modelling of redundancy in multiple mobile robots* in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)* **2** (2000), 1083–1087.

- 
80. Kumar, N. *et al.* Reliability analysis of a robotic system using hybridized technique. eng. *Journal of industrial engineering international* **14**, 443–453. ISSN: 1735-5702 (2018).
  81. Sharma, S., Kumar, D. & Kumar, A. Reliability analysis of complex multi-robotic system using GA and fuzzy methodology. *Applied Soft Computing* **12**, 405–415. ISSN: 1568-4946. <http://www.sciencedirect.com/science/article/pii/S1568494611003140> (2012).
  82. Ferguson, T. A. & Lu, L. *Fault Tree Analysis for an Inspection Robot in a Nuclear Power Plant* eng. in *IOP Conference Series: Materials Science and Engineering* **235** (2017), 12003.
  83. Fazlollahtabar, H. & Akhavan Niaki, S. T. Integration of fault tree analysis, reliability block diagram and hazard decision tree for industrial robot reliability evaluation. eng. *Industrial robot* **44**, 754–764. ISSN: 0143-991X (2017).
  84. Coit, D. Cold-standby redundancy optimization for nonrepairable systems. eng. *IIE transactions* **33**, 471–478. ISSN: 0740-817X (2001).
  85. Liang, Y.-C. & Smith, A. An ant colony optimization algorithm for the redundancy allocation problem (RAP). eng. *IEEE transactions on reliability* **53**, 417–423. ISSN: 0018-9529 (2004).
  86. Kulturel-Konak, S., Smith, A. E. & Coit, D. W. Efficiently Solving the Redundancy Allocation Problem Using Tabu Search. *IIE Transactions* **35**, 515–526. eprint: <https://doi.org/10.1080/07408170304422>. <https://doi.org/10.1080/07408170304422> (2003).
  87. Coit, D. & Konak, A. Multiple Weighted Objectives Heuristic for the Redundancy Allocation Problem. eng. *IEEE transactions on reliability* **55**, 551–558. ISSN: 0018-9529 (2006).
  88. Govindan, K., Jafarian, A., Azbari, M. E. & Choi, T.-M. Optimal Bi-Objective Redundancy Allocation for Systems Reliability and Risk Management. eng. *IEEE transactions on cybernetics* **46**, 1735–1748. ISSN: 2168-2267 (2016).

- 
89. Chambari, A., Najafi, A. A., Rahmati, S. H. A. & Karimi, A. An efficient simulated annealing algorithm for the redundancy allocation problem with a choice of redundancy strategies. *Reliability Engineering and System Safety* **119**, 158–164. ISSN: 0951-8320. [http : //www.sciencedirect.com/science/article/pii/S0951832013001476](http://www.sciencedirect.com/science/article/pii/S0951832013001476) (2013).
  90. Lee, D.-H., Zaheer, S. A. & Kim, J.-H. A Resource-Oriented, Decentralized Auction Algorithm for Multirobot Task Allocation. eng. *IEEE Transactions on Automation Science and Engineering* **12**, 1469–1481. ISSN: 1545-5955 (2015).
  91. Zhang, K., Collins Jr, E. & Shi, D. Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction. eng. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* **7**, 1–22. ISSN: 15564665 (2012).
  92. Shi, J., Yang, Z. & Zhu, J. An auction-based rescue task allocation approach for heterogeneous multi-robot system. English. *Multimedia Tools and Applications*. ISSN: 13807501 (2018).
  93. Yuan, Q., Guan, Y., Hong, B. & Meng, X. Multi-robot task allocation using CNP combines with neural network. eng. *Neural Computing and Applications* **23**, 1909–1914. ISSN: 0941-0643 (2013).
  94. Zitouni, F., Harous, S. & Maamri, R. A Distributed Approach to the Multi-Robot Task Allocation Problem Using the Consensus-Based Bundle Algorithm and Ant Colony System. eng. *IEEE Access* **8**, 27479–27494. ISSN: 2169-3536 (2020).
  95. Arif, M. & Haider, S. *A flexible evolutionary algorithm for task allocation in multi-robot team* in. **11056** (Springer Verlag, 2018), 89–99. ISBN: 9783319984452.
  96. Nagendran, A., Crowther, W., Turner, M., Lanzon, A. & Richardson, R. Design, control, and performance of the 'weed' 6 wheel robot in the UK MOD grand challenge. eng. *Advanced robotics* **28**, 203–218. ISSN: 0169-1864 (2014).



- 
97. Bellicoso, C. D. *et al.* Advances in real-world applications for legged robots. eng. *Journal of field robotics* **35**, 1311–1326. ISSN: 1556-4959 (2018).
  98. Lee, G. *et al.* Series of Multilinked Caterpillar Track-type Climbing Robots. eng. *Journal of field robotics* **33**, 737–750. ISSN: 1556-4959 (2016).
  99. Bogue, R. Snake robots: A review of research, products and applications. eng. *Industrial robot* **41**, 253–258. ISSN: 0143-991X (2014).
  100. Hassanalian, M. & Abdelkefi, A. Classifications, applications, and design challenges of drones: A review. eng. *Progress in aerospace sciences* **91**, 99–131. ISSN: 0376-0421 (2017).
  101. Hristov, G. V., Zahariev, P. Z. & Beloev, I. H. A Review of the Characteristics of Modern Unmanned Aerial Vehicles. eng. *Acta technologica agriculturae* **19**, 33–38. ISSN: 1335-2555 (2016).
  102. Rausand, M. & Høyland, A. *System reliability theory: models, statistical methods, and applications* (John Wiley & Sons, 2003).
  103. Kim, H. & Kim, P. Reliability–redundancy allocation problem considering optimal redundancy strategy using parallel genetic algorithm. eng. *Reliability Engineering and System Safety* **159**, 153–160. ISSN: 0951-8320 (2017).
  104. Ardakan, M. A. & Hamadani, A. Z. Reliability–redundancy allocation problem with cold-standby redundancy strategy. *Simulation Modelling Practice and Theory* **42**, 107–118. ISSN: 1569-190X. [http : //www.sciencedirect.com/science/article/pii/S1569190X13001937](http://www.sciencedirect.com/science/article/pii/S1569190X13001937) (2014).
  105. Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. eng. *IEEE Transactions on Evolutionary Computation* **6**, 182–197. ISSN: 1089-778X (2002).
  106. Zhang, Q. & Li, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. eng. *IEEE Transactions on Evolutionary Computation* **11**, 712–731. ISSN: 1089-778X (2007).

- 
107. Zitzler, E. & Thiele, L. *Multiobjective optimization using evolutionary algorithms : A comparative case study* eng. in *Lecture notes in computer science* (Springer, Berlin, 1998), 292–301. ISBN: 9783540650782.
  108. Zitzler, E. & Thiele, L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. eng. *IEEE transactions on evolutionary computation* **3**, 257–271. ISSN: 1089-778X (1999).
  109. Zitzler, E. *Evolutionary algorithms for multiobjective optimization: Methods and applications* (Citeseer, 1999).
  110. Guerreiro, A. P., Fonseca, C. M. & Paquete, L. *The Hypervolume Indicator: Problems and Algorithms* 2020. arXiv: 2005.00515 [cs.DS].
  111. Liu, C., Liu, C., Kroll, A. & Kroll, A. Memetic algorithms for optimal task allocation in multi-robot systems for inspection problems with cooperative tasks. eng. *Soft computing (Berlin, Germany)* **19**, 567–584. ISSN: 1432-7643 (2015).
  112. Gerkey, B. P. & Matarić, M. J. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. eng. *The International Journal of Robotics Research* **23**, 939–954. ISSN: 0278-3649 (2004).