TOWARDS A NOVEL RESILIENT ROBOTIC SYSTEM

A Thesis Submitted to the College of
Graduate Studies and Research
In Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy
In the Division of Biomedical Engineering
University of Saskatchewan
Saskatoon, Canada

By
Tan Zhang

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Division of Biomedical Engineering
Engineering Building
57 Campus Drive
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5A9

ABSTRACT

Resilient robotic systems are a kind of robotic system that is able to recover their original function after partial damage of the system. This is achieved by making changes on the partially damaged robot. In this dissertation study, a general robot, which makes sense by including active joints, passive joints, passive links, and passive adjustable links, was proposed in order to explore its resilience. Note that such a robot is also called an under-actuated robot. This dissertation presents the following studies.

First, a novel architecture of robots was proposed, which is characterized as under-actuated robot. The architecture enables three types of recovery strategy, namely (1) change of the robot behavior, (2) change of the robot state, and (3) change of the robot configuration. Second, a novel docking system was developed, which allows for the realization of real-time assembly and disassembly and passive joint and adjustable passive link, and this thus enables the realization of the proposed architecture. Third, an example prototype system was built to experiment the effectiveness of the proposed architecture and to demonstrate the resilient behavior of the robot. Fourth, a novel method for robot configuration synthesis was developed, which is based on the genetic algorithm (GA), to determine the goal configuration of a partially damaged robot, at which the robot can still perform its original function. The novelty of the method lies in the integration of both discrete variables such as the number of modules, type of modules, and assembly patterns between modules and the continuous variables such as the length of modules and initial location of the robot. Fifth, a GA-based method for robot reconfiguration planning and scheduling was developed to actually change the robot from its initial configuration to the goal

configuration with a minimum effort (time and energy).

Two conclusions can be drawn from the above studies. First, the under-actuated robotic architecture can build a cost effective robot that can achieve the highest degree of resilience. Second, the design of the under-actuated resilient robot with the proposed docking system not only reduces the cost but also overcomes the two common actuator failures: (i) an active joint is unlocked (thus becoming a passive joint) and (ii) an active joint is locked (thus becoming an adjustable link).

There are several contributions made by this dissertation to the field of robotics. The first is the finding that an under-actuated robot can be made more resilient. In the field of robotics, the concept of the under-actuated robot is available, but it has not been considered for reconfiguration (in literature, the reconfiguration is mostly about fully actuated robots). The second is the elaboration on the concept of reconfiguration planning, scheduling, and manipulation/control. In the literature of robotics, only the concept of reconfiguration planning is precisely given but not for reconfiguration scheduling. The third is the development of the model along with its algorithm for synthesis of the goal reconfiguration, reconfiguration planning, and scheduling.

The application of the proposed under-actuated resilient robot lies in the operations in unknown or dangerous environments, for example, in rescue missions and space explorations. In these applications, replacement or repair of a damaged robot is impossible or cost-prohibited.

ACKNOWLEDEGMENTS

I would like to express my sincere gratitude to my supervisor, Prof. Chris W.J. Zhang, who brought me to the area of robotics through his excellent guidance, insightful ideas and unsurpassed knowledge. His infectious enthusiasm and encouragement, patience and support have been major driving forces throughout my PhD study period.

I would like to thank my co-supervisors Prof. Dan Zhang and Prof. Madan M. Gupta for giving their guidance and help over the years. I would like to thank Prof. Denial X.B. Chen, Prof. Joel Lanovaz, Prof. Francis Bui, and Prof. Fengfeng Xi for being on the committee and providing valuable suggestions and ideas which have greatly improved my work.

I would like to thank Prof. Changli Liu and Prof. Zhiqin Qian at East China University of Science and Technology for providing an excellent environment for me to do the experimental validation for part of my work. I would like to thank Jincan Huang who made a big contribution to the fabrication of the robot modules.

I would like to thank all of the colleagues in the Advanced Engineering Design Laboratory (AEDL) for their friendship and help during my studies. In particular, Mengya Cai, Kirk D. Backstrom, Bin Han, Yihuan Dai, Jingya Li, Fan Zhang, Dong He, Lei Lei, and Bing Zhang for friendship, encouragement and help.

I would like to thank my parents and my husband (Zhengshou Yang) for their unconditional love and support in all aspects.

I would also like to thank the financial support during this research from the Natural Sciences and Engineering Research Council (NSERC) and from East China University of Science and Technology (ECUST) and Scholarship from China Scholarship Council (CSC).

DEDICATED TO

My Parents in China

My Husband Zhengshou

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

ADT	Axiomatic Design Theory
D-H	Denavite and Hartenberg method
DOF	Degree of Freedom
DP	Design Parameter
DSM	Design Structure Matrix
FCBPSS	F: function, C: context, B: behavior, P: principle, S: state, S: structure.
FR	Functional Requirement
GA	Genetic Algorithm
OPM	Object–Process Methodology
SDM	Shape Deposition Manufacturing

CHAPTER 1
INTRODUCTION

This dissertation presents the design methodologies for under-actuated resilient robots. This chapter introduces the background, motivation of the dissertation research, objectives, and finally, a concise description of the organization of the dissertation.

1.1 Background and Motivation

Robots are one of the most important dynamic systems today because of their practical applications and their ability to interact with humans. More and more robotic systems are required to achieve tasks in challenging operating conditions. This, however, increases the chance of system failures. Failure is defined as “a state of inability to perform a normal function” in Merriam Webster [2014].

The demand of sustainable operations requires robots with an ability to autonomously accommodate different types of failures, including unanticipated failures. “The traditional approach to failure accommodation consists of prediction of probable hazards and failure modes, followed by preventive over-design of physical machinery and control algorithms” [Zykov 2008]. This approach can only deal with limited types of failures. Robots, the concern of this dissertation, may work in situations where both their environments and task demands are uncertain and replacement and repair of the robot with partial damage is not possible. Therefore, a robot with an ability to recover from an unforeseen failure is imperative. Inspired from by biological systems that have a self-healing ability, the concept of the resilient robot has emerged in recent years.

According to Merriam Webster [2014], resilience is “an ability to recover from or adjust easily to misfortune or change.” The resilient robot is a robot that can recover its original function after a partial failure [adapted from Zhang and Lin 2010]. In this dissertation, a partial failure refers to a failure after which the robot may not perform the original function unless some changes are made. For example, a partial failure to a robot could be such that the robot has at least one active joint left.

A robot’s resilience can be achieved through three means: (1) behavior or principle change, (2) configuration change (also called reconfiguration), and (3) component state change [Zhang et al. 2014]. To the proponent’s best understanding, the resilient robot with these three means has not received attention in literature, though a few studies on resilient robot have appeared in literature. The resilient machine proposed by Bongard et al. [2006] might be the only so-called resilient robot prototype so far in literature. Indeed, one of the means for recovery in the resilient robot may be similar to the self-reconfiguration ability in robots [Yim et al. 2007]. However, the resilient robot goes beyond the self-reconfigurable robot as self-reconfiguration is the only means for recovery (a detailed discussion of this point is presented in Chapter 2). Therefore, a systematic study of resilient robots is needed, which becomes the major motivation of this study.

1.2 Research Objectives

The overall objective of the research in this dissertation was to improve the understanding of the resilience of robotic systems and to develop the technology for designing and realizing the resilient robotic system. To achieve this overall objective, two questions should be answered. Question 1: What would be the general architecture of resilient robots? Question 2: How may an original function be recovered after partial damage?

Regarding the first question, the general architecture of a robot for high resilience needs to be studied. Regarding the second question, the technology for the realization of resilient robots needs to be developed. The second question is closely related to the architecture defined in the first question.

To achieve the overall objective of the research and provide a rational answer to the aforementioned questions, the following specific objectives were defined for this dissertation.

Objective 1: To develop a general architecture of resilient robotic systems, of which a robot with the highest degree of resilience as opposed to the robots in the current literature can be constructed.

Architecture is to an engineering system as DNA is to a biological system [Zhang 2014a]. To engineer any new system, the first thing is to define the system's architecture.

Objective 2: To design a docking system of an under-actuated resilient robotic system based on the general architecture.

It is apparent from the preceding discussion that a resilient robot requires the robot to assemble two components or disassemble two connected components in real time. This operation can be generalized into the concept of docking (thus “docking system”), where two moving systems attach or detach dynamically.

Objective 3: To develop a methodology for finding the best goal configuration with which a failed robot or a robot with partial damage can perform the original function.

Objective 4: To develop a methodology to determine a reconfiguration “path” from a damaged configuration to a goal configuration so that the robot can perform the original function.

1.3 Dissertation Organization

The remainder of the dissertation is organized as follows:

Chapter 2 presents a comprehensive literature review on the concept of resilience and its relevance to robotics. The goal of the review is to elaborate on the identity of the resilient robot and to examine the problem dimensions of designing and constructing a resilient robot. This chapter also contains a literature review on the technology on self-reconfigurable robots, as the technology is similar to that of the resilient robot discussed in this dissertation.

Chapter 3 presents the development of a general architecture of resilient robots, corresponding to objective 1. This architecture is derived based on the general recovery strategies, with consideration of cost and effort. The architecture lays down a foundation to discuss the design of resilient robotic systems, configuration synthesis, and reconfiguration realization.

Chapter 4 presents the development of the docking system of an under-actuated resilient robotic system. The docking system is an essential system in an under-actuated resilient robot, as derived from the architecture proposed in Chapter 3. The docking system was designed as a unified interface for all modules. It also physically realized adjustable “passive” links (the third means for changing the robot, as mentioned before) and “passive” joints.

Chapter 5 presents a method for configuration synthesis of the under-actuated resilient robot. The configuration synthesis is formulated as an optimization problem and a case study is given to illustrate the method.

Chapter 6 presents a method for the reconfiguration process with a sequence of moves to reconfigure a damaged robot into a goal configuration.

Chapter 7 presents the conclusions and future work.

CHAPTER 2
LITERATURE REVIEW

This chapter gives an overview of some of the important previous work on “resilient” robots: (i) the concept of the resilient robotic system; (ii) development of the resilient robot architecture; (iii) the docking system; (iv) robot configuration synthesis; and (v) robot reconfiguration. The purpose of this review is to provide a justification of the needs and the scope of the research objectives defined in Chapter 1. The use of double quote for resilient means that the concept of resilient robot has not been formerly defined yet at this point but at the end of this chapter, a formal definition of resilient robot will be given.

2.1 The Concepts of Dynamic Systems

A robot is a kind of dynamic system. A dynamic system is “one where the state of the system changes with respect to time, space, and/or event” [Zhang 2014b]. This thesis only considered the time-based dynamics. In this section, definitions of several basic concepts about dynamic systems are given in order to facilitate the subsequent discussion in this dissertation.

Definition 2.1 *Architecture of a system*.

Crawley et al. [2004] defined architecture as “an abstract description of the entities of a system and the relationships between those entities.” According to Merriam-Webster [2014], an entity is “something that has real and independent existence” and an element is “a fundamental, essential, or irreducible constituent of a composite entity.” Lockemann [2003] pointed out that entities can be defined or categorized based on their functions. Crawley [2007] defined the function as “the activities, operations and transformations that cause, create or contribute to performance (i.e., changing voltage), which is performed at the interfaces and effects acted on elements.” Also, he defined interface as “a set of points of contact between parts.”

Definition 2.2 *Configuration of a system*.

Zhang [2014a] defined configuration as “a property of a system that describes what and how components are distributed in space.” As such, the configuration is a spatial property of a system. Since a system is viewed at three different levels—conceptual, embodiment, and detail—the configuration of a system should have three views: conceptual view, embodiment view, and detail view. According to the definition of conceptual, embodiment, and detail views of a system [Zhang 2013], only the conceptual and embodiment views make more sense for spatial properties. Therefore, this dissertation only considered the conceptual view and embodiment views of configuration. Figure 2.1 further illustrates the configuration at the conceptual view and embodiment view. Figure 2.1A shows the conceptual view of a five-component assembly. Figures 2.1B, 2.1C, and 2.1D show three different embodiment views of the same five-component assembly.

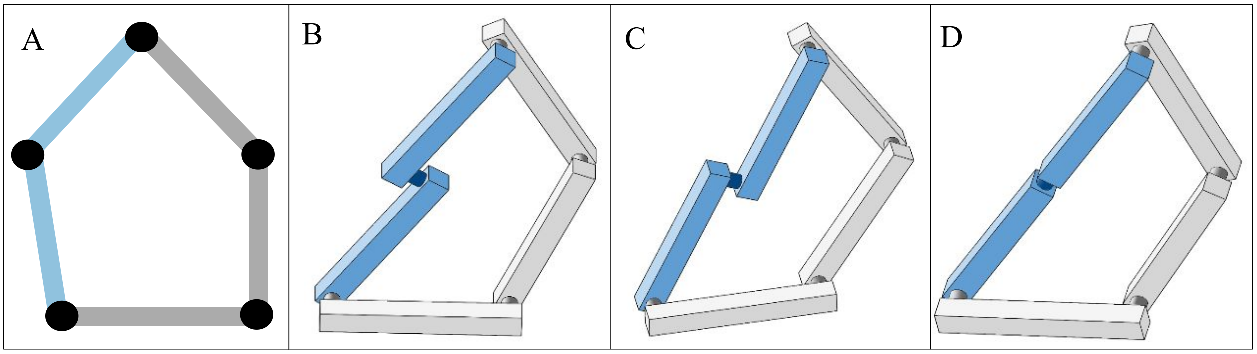


Figure 2.1. Configuration at the conceptual view and embodiment view. A five-component assembly from the conceptual view (A), and the same five-component assembly from embodiment views where the assembly between the two dark-colored components is different (B, C, D).

Definition 2.3 *Self-reconfiguration of a system*.

The system uses its own resources and effort to re-arrange the connections between components.

Definition 2.4 *Reconfigurable*.

Reconfigurable refers to whether a system can be reconfigured. A system is first reconfigurable, and then reconfiguration is the process of changing connections.

Definition 2.5 *Flexible, adaptable and adaptive*.

Flexible means that a system is able to accommodate different tasks by changing the system behavior but not changing its configuration. The definition of flexibility in literature may be different in different contexts (e.g., manufacturing systems). The definition of flexibility here may only be limited to the scope of this dissertation. Adaptable means that the structure of a system can be changed, which includes both reconfiguration and change of the geometry of a component in the system. Adaptive is similar to flexible but the means required to make a system adaptive are greater than those required to make a system flexible. In this dissertation, the terms reconfigurable and adaptable are used interchangeably unless otherwise stated.

Definition 2.6 *Reconfiguration or adaptation.*

It is a process by which a system's configuration or the geometry of its components is changed.

Definition 2.7 *Planning in reconfigurable or adaptable robots.*

Finding a sequence of moves from an initial configuration to a goal configuration in reconfigurable or adaptable robots.

Definition 2.8 *Scheduling in reconfigurable or adaptable robots.*

The scheduling in reconfigurable or adaptable robots is a sequence of operations, complete with time and efforts required. Scheduling occurs closer in time to the realization than planning does.

Definition 2.9 *Synthesis of a system*.

Synthesis is the combination of components to form a new system to satisfy a task.

Definition 2.10 *Modular robot.*

A modular robot consists of a number of modules with standard interfaces. The configuration of the robot can thus be changed by assembling and disassembling these modules. A module is always viewed as a package in that its interior is not exposed to the outside world.

2.2 The Concept of Resilient Robotic Systems

Resilience is a relatively new concept in the field of robotics, even in engineering, although it has been developed in the field of sociology and ecology. This section discusses the identity of the resilient robot. The discussion is carried out by distinguishing resilience from other similar concept such as robustness, reliability, and so on, as well as distinguishing resilient robots from other related types of robots, such as self-reconfigurable robots, self-repairing robots, and so on.

2.2.1 The Concept of Resilience

The word “resilience” originates from the Latin word “resilire,” which means “leaping back or rebounding” [McAslan 2010]. In different disciplines, resilience may be defined differently. The work in Wang [2013], Van Breda [2001], and Bhamra et al. [2011] have given a more comprehensive classification of resilience definitions. The common feature of the resilience concept is that resilience is the ability of an object or a system to recover from changes to a normal situation. This dissertation focused on the resilience definition in the operation context of engineering. Some efforts have been done on the concepts of resilience engineering [Hawks and Reed 2006, Hollnagel et al. 2006, Patternson et al. 2007, Bursztein and Goubault-Larrecq 2007, Zhang 2007]. These indicate that resilience engineering does not focus on how accidents happen and how failures result from accidents, but focus on how the system recovers from a failure. For example, Hollnagel [2006] defined resilience as "the ability of an organization (system) to keep, or recover quickly to, a stable state, allowing it to continue operations during and after a major mishap or in the presence of continuous significant stresses." Zhang and Lin [2010] stated, "Hollnagel's definition lacks a distinction of resilience from robustness" and defined resilience as "the ability of the system on how the system can still function to a desired level when the system suffers from a partial damage."

The identity of a resilient robot depends on whether there is a unique challenge or problem with the concept of resilience. There are several concepts that are closely related to the concept of resilience, such as "self-healing," "fault tolerance," "self-repairing," "sustainability," "reliability," "dependability," "survivability," and "robustness." Self-healing is well known in biological systems [Kumar et al. 1992]. Normal cells can handle normal demands, maintaining a steady or stable state. When a biological cell encounters excessive stresses or stimuli, it may undergo adaptations to shift to a new state in order to maintain its original function. When there is no possible adaptive response or a cell's adaptive capability is exceeded, cell injuries may develop. Upon suffering from a severe injury, the injured cells may die; however, upon suffering from a mild injury, an injured cell may "recover" through a complicated chemical change. For example, amino acids enable muscles to build up, repair, or regenerate. Even though Madden [2007] pointed out “our contemporary technology is not ready to interface with such a complex biological system,” i.e., regenerate an identical part, it is believed that the recovery always involves reconfiguration and re-adjustment of some smaller elements in the cell [Zhang and Lin 2010]. The concept of the resilient robot is inspired by biological self-healing, from both macro and micro viewpoints. However, most self-healing robots seem to refer to chemical reactions, which does not consider the reconfiguration and re-adjustment of robot components. For example, a self-healing robot proposed in the work of Geekologie [2013] can grow back skin back without a catalyst. A self-repairing robot is more restricted to the repairing of damaged components, that is, to repair the damaged component with external resources. Compared with self-healing and self-repairing, the scope of the recovery in resilience is more general, which includes more recovery strategies and methodologies than these two.

Fault tolerance is a classic notion in software engineering and is defined as “the ability to deliver service in the presence of faults” in Avizienis et al. [2004]. Zhang and Lin [2010] pointed out “fault in fault tolerance refers to errors made at the phase of software development and/or errors in the system input at the phase of software operation.” Fault tolerance also includes component damage. However, the recovery strategies in fault tolerance are usually based on the software development. Compared with the concept of fault tolerance, the solutions generated in a resilient system could be to change software and/or to change hardware no matter whether the faults are caused by software or hardware.

Sustainability is defined as “a system’s ability to sustain or to maintain itself” in Zhang et al. [2010]. A sustainable robot may have redundant parts that are used to replace failed parts.

Survivability is defined as “the ability of a system or an object to live or exist, especially in spite of difficult conditions” in Merriam Webster [2014]. Zykov et al. [2007] pointed out that “long-term physical survivability of most robotic systems today is achieved through durable hardware.”

Robustness is defined in Kitano [2004] as “an ability that allows a system to maintain its functions against internal and external perturbations or noises,” that is, how a system remains insensitive to noises.

Reliability is defined as “the ability of a system or component to perform its required functions under stated conditions for a specified period of time” in Verma et al. [2010], that is, how a system is sensitive to random failures. Reliable systems are similar to survival systems, and they are focused on how the system still functions given external disturbance. They focus on the strategy of prevention (reliability) and absorption during the event (robustness).

Dependability is defined as “the ability of a system to deliver a service that can justifiably be trusted and to avoid failures” in Bischoff and Graefe [2003]. In a dependable system, the user has faith that the robot will fulfill its functions under specified conditions. The reliable and robust systems can add value to the faith of the user. The resilient robot creates more faith in the mind of user, that is, a robot that has a degree of reliability, robustness, and resilience is a highly dependable robot.

2.2.2 The need for resilient robotic systems

Based on the above discussion, a resilient robotic system is able to recover its original function after a partial damage to the system.

The concept of the resilient robotic system provides a promising method for reducing the loss of robots, especially for those that work in dangerous or remote environments. Resilient robotic systems have several merits:

- Cost-effectiveness: reuse of the remaining systems reduces costs by extending system life.
- Repairability: redundancy may be brought in to deal with the faults caused by an internal/external environment.
- Durability: “a component for one function can be trained to do another function of another component against the system malfunction,” as defined by Zhang and Van Luttervelt [2011].
- Interconnectability: ease of replacement of the damaged components.

2.2.3 The concept of resilient robotic robots

A resilient robot is able to recover its original function after a partial damage. The first thing to develop is the strategy of recovery. This dissertation proposed three recovery strategies [Zhang et al. 2014]: strategy I: training a remaining system to perform its function based on a new principle, e.g., change of a control system with the relevant input resources to the robot; strategy II: changing the configuration of a system by re-arranging its components (see reconfigurable robot

[Bi 2002]); strategy III: changing the states of a component, e.g., changing the length of a bar component (see the so-called adjusting mechanism [Bi 2002]). The derivation of the three recovery strategies will be discussed in Chapter 3. This dissertation gives the definition of resilient robotic systems as follows.

Definition 2.11 *Resilient robotic systems*.

A kind of robotic systems that are able to recover their original function after a partial damage through at least one of the three recovery strategies.

2.2.4 Dimension of the problems in designing and constructing a resilient robot

In resilient robotics, recovery must be performed on a self-operation basis. Therefore, a resilient robot must equip itself with a cognition ability to plan and operate the system in a new situation where the required function of the system can be retained. Based on this and the proposed recovery strategies, the problem dimension for resilient robots will be formulated as follows:

> Problem 1: The general architecture of resilient robots, addressing the principles of recovery when a robot has partial damage. The term "principle of recovery" is used interchangeably with the term "strategy of recovery" throughout this dissertation.
>
> Problem 2: The ability to change the robot on its own to perform tasks as desired on the robot, including the original task.
>
> Problem 3: The ability to determine the goal configuration or goal behavior expected of the robot.
>
> Problem 4: The ability to make a plan and a schedule for a robot to change from an

initial configuration to a goal configuration.

Problem 5: The ability to manipulate and control the robot to perform the change schedule.

Put together, the above problems are called the problem dimension of resilient robotics; a full resilient robot must have solutions to the above problems. In this dissertation, the last problem, Problem 5, has not been studied systematically, though an ad hoc treatment of this problem in the prototype development was attempted. The next four sections will present a review and an analysis of the existing work on the first four problems.

2.3 Existing Resilient Robots

There are few published robot prototypes that include the term resilience in the robot name. However, other existing robots may have addressed one of the recovery strategies as proposed and discussed earlier in this dissertation. For example, self-reconfigurable robotics is a research field that is closely related to resilient robotics; self-reconfigurable robots can change their topology, which is coincident with recovery strategy II. As well, soft robots can easily change the state of components, which corresponds to recovery strategy III. It is noted that these "potential" resilient robots are not considered resilient but rather "flexible to different tasks."

2.3.1 Resilient machine

To the best of the writer's knowledge, the black starfish robot designed by Bongard et al. [2006] is the only published prototype called a resilient robot. This robot has four identical legs and a main body, as shown in Figure 2.2. All joints are actuated by servo-motors. Rotation to different angles can cause the body part to lie flat/downward/upward. The orientation of the main body is measured once the robot performs an action. After one lower limb gets damaged (the part indicated by an arrow in Figure 2.2), the robot starts to identify the failure and explore the best behavior to achieve the original function, e.g., moving forward. Thus, this robot has the recovery ability under recovery strategy I as defined in the previous section. There are other robot prototypes that are not called resilient robots but their damage-recovery algorithm, i.e., their ability to autonomously find compensatory behaviors in unanticipated situations, can potentially be used for damage recovery. Bongard et al. [2006] used an evolutionary algorithm-based (EA) self-modeling approach on the above resilient machine to search new behaviors. The algorithm synthesized candidate models and found the best one to match the collected sensor data as well as the best model to match the original task. Similarly, an evolutionary algorithm has been applied to a snake robot with a damaged body [Mahdavi and Bentley 2006] and to a four-legged robot with a broken leg [Berenson et al. 2005]. Christensen et al. [2014] proposed a central pattern generator-based (CPG) strategy to find efficient locomotion gaits after failures of several actuators. The work in Mostafa [2010] pre-defined different gait sequences for possible failures. The work in Erden and Leblebicioğlu [2008] incorporated the reinforcement learning scheme into the free gait generation to choose the more stable state after losing one leg. Koos et al. [2013] pointed out, “Bongard’s approach performs differently in the self-model and in reality.” They proposed a T-Resilience algorithm and implemented the algorithm on a hexapod robot, which lead to more gaits, faster and more efficiently.

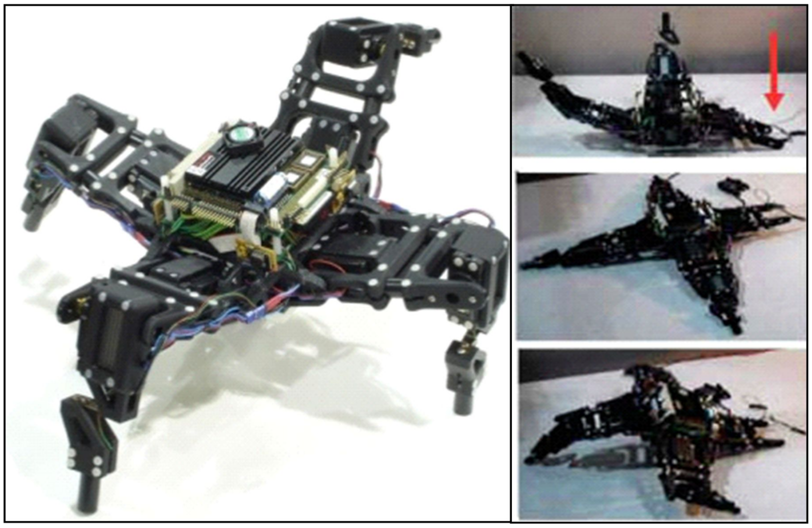


Figure 2.2. Resilient machine [Zykov 2008].

2.3.2 Self-reconfigurable robots

Self-reconfigurable robots are “a class of robots that are composed of many homogeneous or heterogeneous modules that can change the way they are connected on their own, thus changing the overall shape of the robot” [Stoy and Nagpal 2004]. They have great potential to become resilient, as they might be able to discard the damaged modules and replace them with other modules, i.e., recovery strategy II.

According to the functionality and arrangement of modules, the self-reconfigurable robots are divided into two types: lattice type and chain type. In lattice-type robot, modules are arranged in a regular pattern. Modules move to neighboring modules in a limited number of steps and patterns of assembly. Some of the lattice-type prototypes are shown in Figure 2.3. In the chain-type robot, modules have built-in joints. Modules can form a more variety of types of locomotion configurations. Some of the chain-type prototypes are shown in Figure 2.4. The two types may be combined to create hybrids of self-reconfigurable robots.

Figure 2.3. Examples of the lattice-type self-reconfigurable modular robots. (A) Molecule [Kotay et al. 1998]. (B) Telecubes [Suh et al. 2002]. (C) ICubes [Unsal and Khosla 2001]. (D) ATRON [Christensen 2006]. (E) The Programmable Parts [Bishop et al. 2005]. (F) Stochastic-3D [Lipson et al. 2005]. (G) A bipartite robot [Terada and Murata 2008]. (H) Odin [Garcia 2008]. [I] Miche [Gilpin et al. 2008].

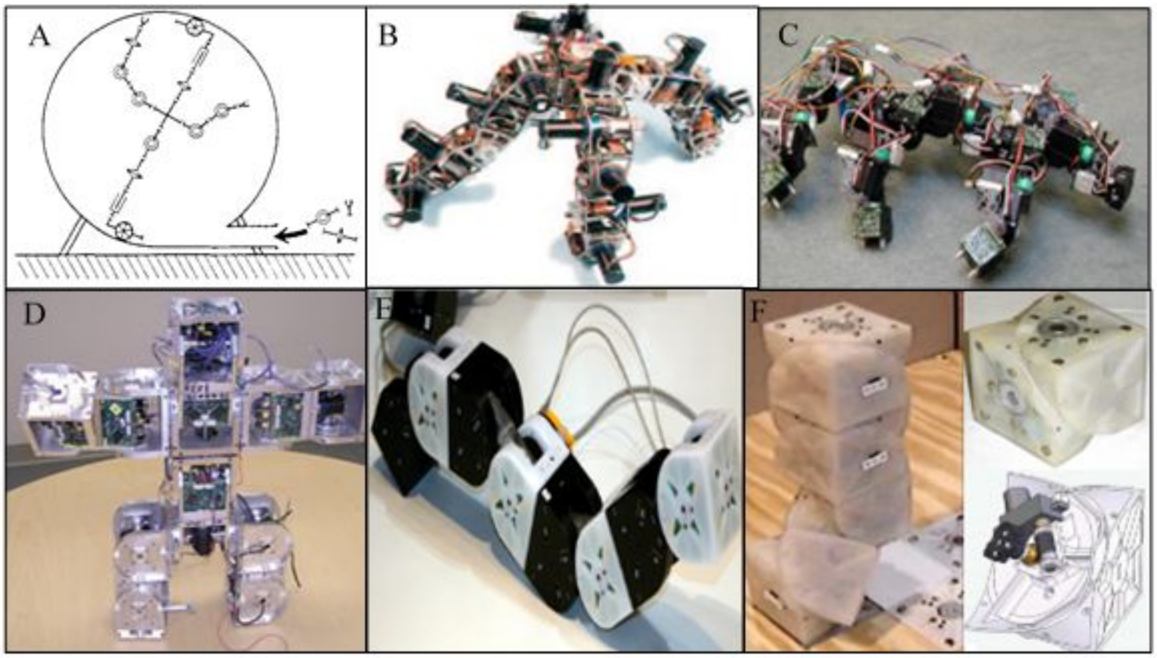


Figure 2.4. Examples of the chain-type self-reconfigurable modular robots. (A) CEBOT [Fukuda and Nakagawa 1987], (B) PolyBot [Yim et al. 2002], (C) CONRO [Shen et al. 2002], (D) SuperBot [Shen et al. 2006], (E) M-TRAN III [Kurokawa et al. 2008], (F) Molecubes and dissected view of the module [Zykov et al. 2005] and [Zykov 2007].

Most self-reconfigurable robots are homogeneous systems that consist of a single module type, which encapsulates actuators, connection systems, sensors, controllers, and effectors. The examples shown in Figure 2.3 and Figure 2.4 are homogeneous ones except Figures 2.2C, 2.3G, 2.3H, and 2.4A. The module of the Molecube robot is shown in Figure 2.4F.

Different actuators have different functions. They allow one module to move around, or to move with respect to other modules to generate locomotion, or to attach or detach neighbouring modules to reconfigure the whole system. The connection system is crucial for self-reconfigurable robots [Shen and Will 2001]. The connection system may also be called a docking system in that one module attaches to another that temporarily stands still. A detailed review of the existing docking system for self-reconfigurable robots will be given in the next section.

CEBOT [Fukuda and Nakagawa 1987], shown in Figure 2.4A, is the most prominent heterogeneous self-reconfigurable robot. It is a chain-type robot with three types of modules: actuation modules, structure modules, and modules with tools. Polybot, shown in Figure 2.4B and developed by Yim [1993], is also a heterogeneous chain-type robot. Similar to the homogenous robot, each module encapsulates the subsystems of structure, actuation, sensing, and connection. The source of heterogeneity comes from various types of sensors on different modules. Most heterogeneous self-reconfigurable robots consist of two types of modules: links and joints. For example, the Odin robot [Garcia 2008] in Figure 2.3H consists of active links and active joints: the active links mainly provide power and structure functionalities and the active joints transfer the power and information between any two neighboring links. ICubes [Unsal et al. 2001], shown in Figure 2.3C, uses passive cubes as structure modules and a manipulator that moves the passive cubes around. The manipulator is composed of rigid links and active joints. A similar robot was developed by Terada and Murata [2008], as shown in Figure 2.3G.

Morpho robot [Yu et al. 2008] is a heterogeneous self-deformable modular robot that can change the overall shape of the robot. It consists of active links and passive links. The active links function as a linear joint that can contract and expand when the motor drives a mounted rack, as shown in Figure 2.5A. The active links force the passive links to expand or contract to form new geometries, as shown in Figure 2.5B. In this dissertation, such passive links are called adjustable passive links.

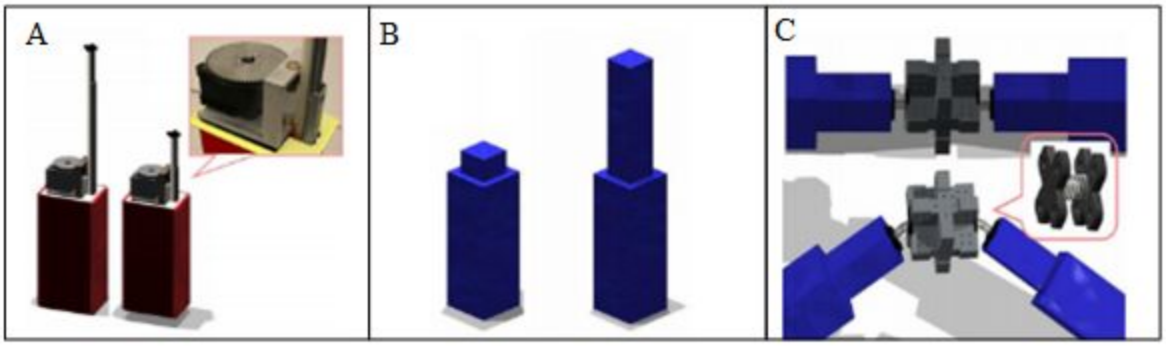


Figure 2.5. The three basic types of modules of the Morpho robot [Yu et al. 2008]. (A) Active links: the expanded (left) and contracted (right) state, (B) Passive links: the contracted (left) and expanded (right) state, (C) The connector that is used to connect to links and other modules.

Compared with the homogeneous robots, the self-reconfiguration capability of heterogeneous ones allows the system to perform locomotion tasks over difficult terrain. As well, robots with passive modules are more cost-effective.

2.3.3 Soft Robots

Trivedi et al. [2008] defines a “soft robot” as a robot that is inherently compliant and exhibits large strains in the normal operation.” Soft robots have an infinite number of degrees of freedom, which leads to dexterous mobility in a compliant way without causing injury. They can be resilient, as per recovery strategy III (i.e., changing their shape and form). Recent work on soft robots includes Chembot [Gizmag 2009], Shape Deposition Manufacturing (SDM) Hand [Dollar and Howe 2010], silicone-based soft machine [Morin et al. 2012], and universal gripper [Brown et al. 2011] (shown in Figures 2.6A to 2.6D). The Chembot has a hyper-elastic skin composed of multiple cellular compartments that are filled with air and loosely packed particles. There is an incompressible fluid and an actuator that can change its volume, which causes Chembot's skin to stretch to change the robot shape and roll around. The SDM Hand is featured with passive compliant joints to conform uncertain objects. The soft device has a stiff link and a soft wrapper backbone. Further, the joint is flexible. The soft device is fabricated using the polymer-based SDM technology. By pumping fluids or air through a narrow channel, the silicone-based soft robot changes its shape to achieve actuation or movement with a high flexibility to contact work and environment objects. The universal robotic gripper is a fingerless hand that uses a rubber membrane filled with ground coffee. Through a combination of suction, friction, and geometric interlocking, the hand is able to hold unfamiliar delicate or heavy objects of varying shapes without the need for sensing and/or feedback.

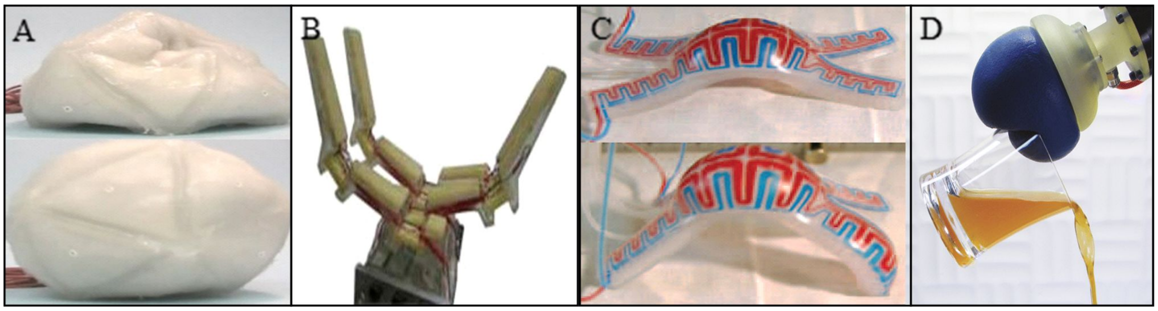


Figure 2.6. Examples of soft robots. (A) Chembot [Gizmag 2009], (B) Shape Deposition Manufacturing (SDM) Hand [Dollar and Howe 2010], (C) Silicone-based soft machine [Morin et al. 2012], (D) Universal robotic gripper [Brown et al. 2011].

It may be clear that soft robots have a high potential to be reconfigured on their own in that their shapes can be readily changed to conform to the load as well as to the environment in which robots are interacting [Trivedi et al. 2008]. Therefore, soft robots can be highly resilient, if the solutions to all the five fundamental problems previously proposed in Section 2.1 are provided.

2.3.4 Concluding remarks

Architecture is the backbone of a system. It defines the entities and the relationships between the entities in the system under development. The entities can be physical and non-physical. Specifically, entities are classified based on their behaviors and properties. Relationships can be physical and non-physical. When a relationship is physical, it may be called a connection. This dissertation is concerned only with the physical relationship. Any two physical entities may connect in the following ways: (1) via a separate physical connector, or (2) via a built-in connector.

Note that for robots, a configuration refers to a particular assembly of a particular set of entities and a particular set of connections. The configuration is thus based on the architecture. If a task is specified, a configuration of the robot system can be determined. Determination of the configuration for the tasks will be discussed in Section 2.4.

This section reviewed the architecture of existing robots that have a certain level of resilience in terms of components and relationships between modules. It has shown that the existing architectures were developed in an ad-hoc manner and a systematic study of the architecture for resilient robots is lacking. It also found that the robot or machine explicitly identified as “resilient” only used recovery strategy I. Although self-reconfigurable robots addressing recovery strategy II have been developed extensively, most of them are homogeneous robots with identical active modules. Most of the heterogeneous robots are specified for a certain task. For example, a manipulator with active (or passive) links and active joints picks up modules (or cubes) and places them in a location. Fewer heterogeneous robots include passive joints. The missing of passive joints in contemporary self-reconfigurable robots is responsible for less intelligent behavior and higher costs. Morpho [Yu et al. 2008] included passive modules and active modules and connectors. However, the passive modules function as linear joints and they are specified for contraction and extension only. Plus, their special module design cannot guarantee an efficient self-reconfiguration. Although soft robots may address recovery strategy III, particularly by changing the shape of materials (as the materials are soft or plastic), it is not convenient to control the material accurately with rigid links or joints.

Another missing aspect in the existing resilient robot architecture is the methodology that could be used to develop a system architecture based on the requirements of robots with partial failure. In this situation, a system design theory such as Axiomatic Design Theory [Suh 1990] could possibly be applied to the design of resilient robot architecture. This has been attempted in this dissertation and will be discussed later.

2.4 Connection System

Everything that crosses a boundary is facilitated by an interface. Interface is defined as “a set of points of contact between parts” [Crawley 2007]. Physically, an interface is the so-called connector in a system or the connection system of a component. To a high degree, the connection system decides the quality of a system, especially for modular systems that emphasize the minimum types of interfaces between components. This section is a literature review on the connection systems that can achieve connection and disconnection between modules, which are mainly used for the resilient robots and the self-reconfiguration robots mentioned in Section 2.2.

2.4.1 Realization of connection systems

There are three principles in literature available for designing the connection systems: magnetic, mechanical, and electrostatic.

The common connection system in modular or reconfigurable robotics is a mechanical latching system, which enables a strong connection. In the CONRO robot [Castano et al. 2000] and the PolyBot robot [Yim et al. 2002], a peg-and-latch connection system was used. When the peg male part inserts into the female part, a latch falls into place to lock the connection, as shown in Figure 2.7A. The connection will be released when the shape memory actuators (SMAs) are heated. A pin-and-hole-based connector has been applied in the I-Cubes robot [Unsal et al. 2001], as shown in Figure 2.7B. The pin is inserted into a hole and rotates, and then the pin is locked into the hole. This design is simpler but requirement high alignment precision to achieve attachment. Another design is to change the peg into a hook. The ATRON robot [Stoy et al. 2010] used the hook mechanism to achieve connections and disconnections, as shown in Figure 2.7C. A similar hook mechanism was used for the M-TRAN III robot [Kurokawa et al. 2008]. However, the hook-based design is more complex and the weight is increased. It is worth noting that Velcro can be viewed as a hook-based connection system. However, it still needs a detachment mechanism, which increases the complexity of the system.

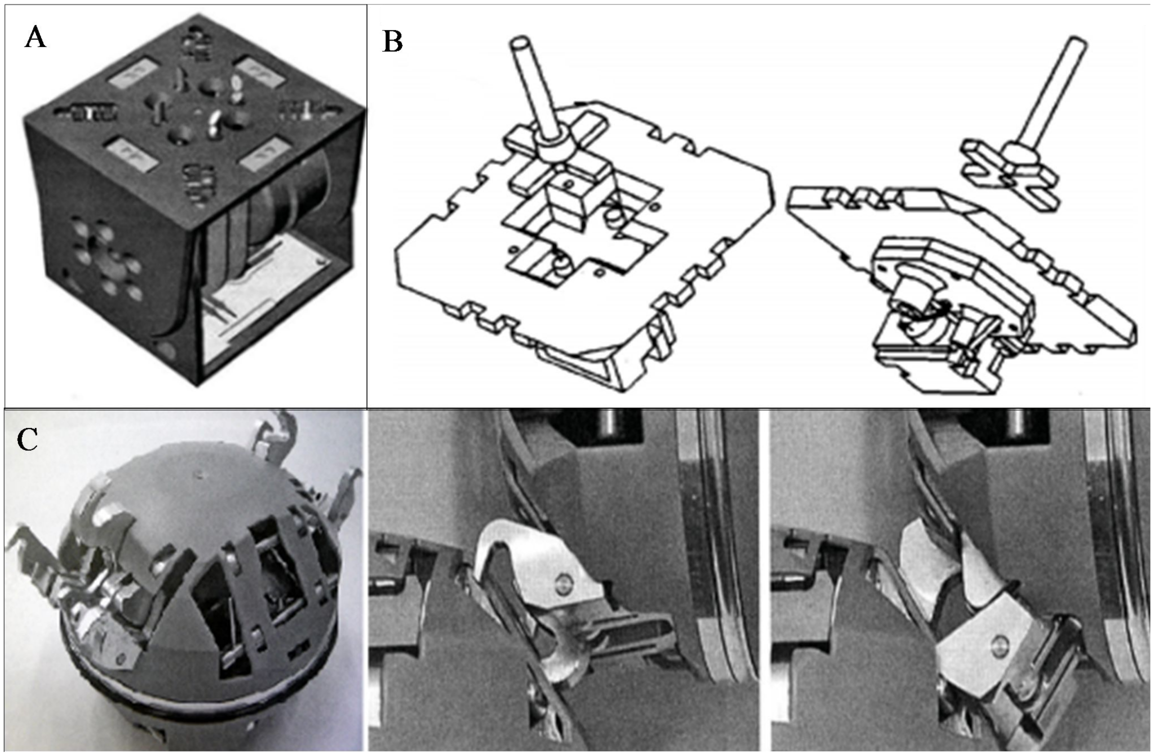


Figure 2.7. Examples of mechanical connectors. (A) A peg-and-latch connection system for PolyBot G3 [Yim et al. 2002], (B) A pin-and-hole-based connector for I-CUBES robot [Unsal et al. 2001], (C) A hook-based connector for ATRON robot [Stoy et al. 2010].

The magnet-based connection is another common approach in modular or reconfigurable robotics, which guarantees a convenient attachment. In the M-TRAN II self-reconfigurable robot [Murata et al. 2002], the connection system includes shape memory alloy (SMA) coils, springs, and magnets fixed on a connecting plat. As shown in Figure 2.8A, the adjacent surfaces are attached when the SMAs are not heated and pulled apart when the SMAs are heated. The connector of the Telecube self-reconfigurable robot [Suh et al. 2002] is made using switching permanent magnet arrays, as shown in Figure 2.8B. Two magnets are attracted when they are aligned in an opposite polarity and disconnected when the magnet arrays are switched by heating the SMAs. The magnetic connection system is efficient and accurate. However, the connection is not strong enough and it may be disconnected by accident. As well, it needs an actuation mechanism for detachment. Electromagnets can be used as actuation instead of SMA coils. However, the actuation mechanism is usually more complex and it increases the weight and power consumption of the robot.

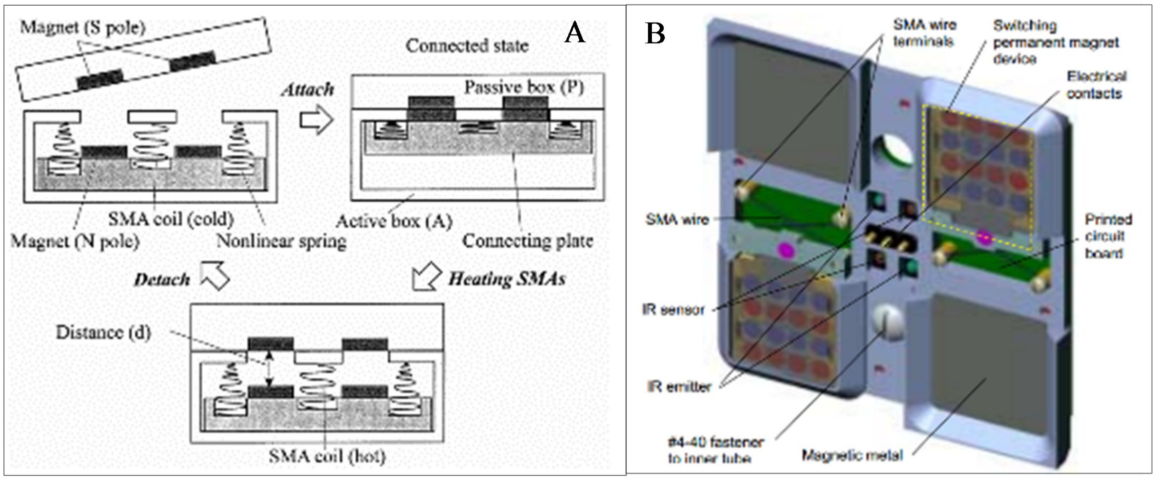


Figure 2.8. Examples of magnetic connectors. (A) A magnetic connection system for the M-TRAN II [Murata et al. 2002], (B) A magnetic connection system for the Telecube robot [Suh et al. 2002].

There are some other connection approaches as well. The vacuum-based connection [Garcia et al. 2010] serves the same purpose as the magnets but it requires precise alignment for attachment. The electrostatic connection [Karagozler et al. 2007] enables reliable connection strengths. Two faces are attracted when they are charged with different polarity. However, the size of the connector is big and the robustness seems poor.

2.4.2 Concluding remarks

The connection system plays an important role in a system. This review presents the connection systems that could be used for resilient robots. It was observed that there is a trade-off between simplicity and disconnection ability. The mechanical connection has strong strength but it is complex and is big in size. The permanent magnet connection has a simple attachment but the detachment needs an actuation system at a cost of complexity. The electromagnet connection has higher energy consumption but the structure of the electrostatic connection is complex. Other approaches such as Velcro and vacuum cannot guarantee the high connection strengths for the rigid robot but they are more suitable for soft robots.

Another issue is that most connection systems have an independent power source and they are limited to connection and disconnection. The independent connection system is designed at a cost of size, weight, and complexity, such as the above mechanical connection systems and magnetic connection systems. At this point, it may be possible to integrate the connection function and the operation function so that the connection system will not take too much space, a possibility investigated in this dissertation.

2.5 Configuration Synthesis of Robots

2.5.1 The relationship between architecture and configuration

By revisiting the definition of architecture and configuration (see Definition 2.1 and Definition 2.2 in Section 2.1), it may be clear that configuration is based on architecture. It is also clear that a particular configuration is supposed to perform a particular task. Configuration synthesis finds the "best" configuration to satisfy a task. The meaning of the "best" depends on the perspective of the designer. Certainly, any "best" must perform a desired task plus other desires as outlined by the designer. As such, configuration synthesis can be mathematically defined as an optimization problem.

A mathematical problem is nothing but variables. An optimization problem has three parts pertinent to the variable: variables that describe a configuration, constraints on the variables, and objectives that describe the property of the "best" configuration. As such, the domain of variables comes from the architecture, and variables and their relations per se come from configurations that can perform a desired task. The desired task is a source to derive or define the objective or goal. Both the architecture and task are sources to derive or define the constraint. The configuration synthesis problem can thus become the selection of variables and determination of the values of the variables under the constraints to make the (objective) function minimum or maximum. Figure 2.9 depicts the foregoing discussion diagrammatically.

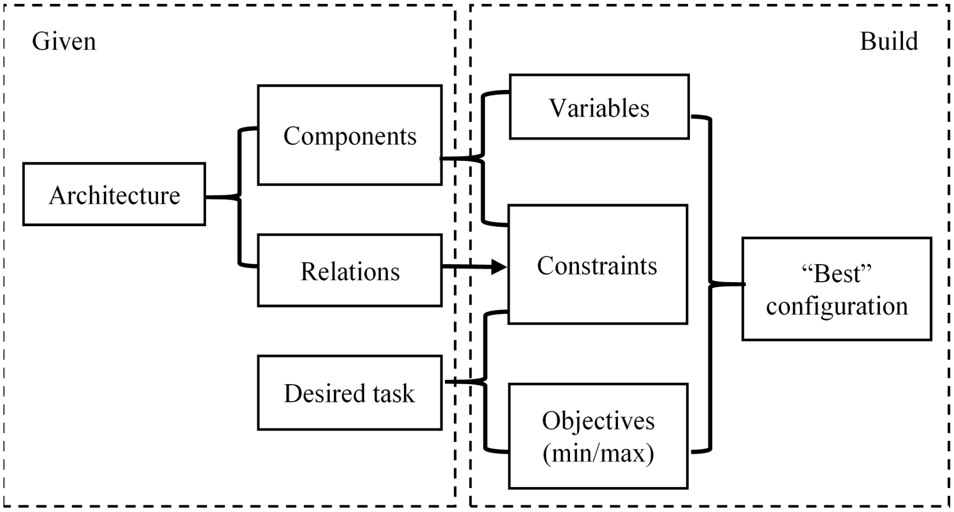


Figure 2.9. The relationship between architecture and configuration.

2.5.2 Literature pertinent to configuration synthesis

Configuration synthesis is typically used for modular robots for task specification in terms of workspace reachability, i.e., “a collection of working points to be followed by the end-effector in the workspace,” as defined in [Chen 1994, Chen and Burdick 1998], such as the tasks for obstacle avoidance, position accuracy, and loading capability at certain points. That makes sense, as any reconfigurable robot has some sort of standard interface to facilitate assembly and disassembly and a standard interface is the key element in any modular system [Zhang 2013].

Clearly, defining the variables is the first step toward solving the configuration synthesis problem. Matsumaru [1995] used a sequential design procedure based on a traditional kinematic model for configuration design of their modular manipulator. Similarly, Paredis and Khosla [1991] used D-H parameters as design variables with joint limitation as design constraints to design their reconfigurable modular manipulator. The traditional kinematic model is not applicable to configuration synthesis of modular robots when the number of modules is big (e.g., more than 10). Chen and Burdick [1995] defined design variables including number of joints, type of joints, number of links, type of links, and the assembly patterns of links, which are all discrete variables. In their work, workspace reachability, joint range availability, and manipulability were defined as design constraints. The minimal degree of freedom (DOF) was considered as a design objective, which expresses a desire for fewer active modules if possible. Bi and Zhang [2001, 2002] proposed a concurrent optimal design method to explore the entire configuration space rooted from type level, number level, and dimension level. Continuous variables (i.e., length of links and the posture of the origin of the robot) were also included in their optimization problem, along with discrete variables that represent the topology of the configuration. The constraints are similar to those in Chen and Burdick [1995]. The minimum energy consumption was considered as a design objective. A two-level genetic algorithm (GA) was proposed by Sakka and Chocron [2001] to search the configuration. The GA was also used in Bi's work [2002].

Another type of task specification is to achieve the fault tolerance of a robot (i.e., to synthesize the configuration that can continue the task after faults). Faults can be classified into actuator faults, sensor faults, and component faults, as discussed by Thavamani [2006]. This section focuses on the actuator faults and joint faults only. Specifically, the active joints may be locked or unlocked due to loss of control actions. Passive joints may be locked due to being stuck in the interface. Note that modular robots more easily suffer from these faults than other robots. Regarding the fault tolerance-based configuration, the key point is the evaluation of fault tolerance, which is the quantification of fault tolerance.

Identifying the fault tolerant configuration is important not only for avoiding precarious configurations but also for identifying the new workspace boundaries resulting from a joint failure. The workspace boundaries of the revolute manipulators correspond to singularities. Yoshikawa [1985] introduced manipulability as a measure of fault tolerance. Roberts et al. [1996] proposed an approach to measure the fault tolerance for kinematically redundant serial manipulators. Further, Roberts et al. [1996, 2001, 2008] proposed the measure of the fault tolerance of a robot. This work examined how joint failures affect the kinematics and dynamics and performance of a manipulator. Paredis [1991] proposed an agent-based approach to achieve "fault tolerance" by increasing the number of joint modules and changing the assembly patterns.

However, the current studies on configuration synthesis have not provided a general method due to their particular architecture. A configuration synthesis for under-actuated robots has not been studied.

2.6 Reconfiguration of Robots

Reconfiguration typically applies to self-reconfigurable robots. The self-reconfiguration problem includes reconfiguration planning and scheduling. Reconfiguration planning identifies a sequence of moves from an initial configuration to a goal configuration. Most works use metrics and heuristics to guide the search. The study in Prevas et al. [2002] proposed a hierarchical planner for self-reconfigurable modules. The highest level decides which cube to move, the lower level generates the position where module will move, and the third level generates the commands to move modules. The studies in Rus and Vona [1999, 2001] developed the Melt-Grow algorithm. An initial configuration is melted into an intermediate configuration, and a goal configuration is grown out of the intermediate configuration. The study in Yim et al. [2001] introduced “goal-ordering” methods (i.e., the locations of the modules of the final configuration are in a pre-defined order). The study in Butler and Rus [2003] presented the PacMan algorithm consisting of distributed planner and actuation protocols. The distributed planner develops paths with a set of pallets for individual modules using an iterative deepening search. The actuation protocol executes the modules in parallel. In these lattice-type robots, each module’s position is specified by unique 2D or 3D coordinates, and the configuration space is small. However, the configuration space becomes much larger in chain-type robots with arbitrary topologies. Thus, there is less work for chain-type reconfiguration [Casal and Yim 1999, Nelson 2005, Shen et al.

2002, Hou et al. 2011]. The study in Shen et al. [2002] proposed a hormone-inspired approach to communication in self-reconfiguration and control on CONRO. The study in Hou et al. [2011] proved that the reconfiguration planning problem from one configuration to another is NP-complete. However, they give the lower and upper bounds for the minimum number of move steps for any reconfiguration planning problem.

Reconfiguration scheduling is for the purpose of realization of a successful reconfiguration plan and decides (1) the components or subsystems, (2) the time, and (3) the effort. Reconfiguration manipulation with control means physically picking up components and subsystems to realize the change of configuration from the initial configuration to the goal configuration. The easiest way to select among possible moves is to pick up one randomly, as proposed in Murata et al. [1994], and Jones and Mataric [2003]. This method eventually moves the module to the goal position. However, their approach is not applicable to the robot architecture which includes passive joints and links. The study in Butler et al. [2001], and Ostergaard and Lund [2004] presented local rules to control the module moves. Different configurations correspond to different conditions. However, it is tedious to define the conditions when there are many possible configurations. By assuming that modules know their global coordinates in the configuration, the study in Ostergaard [2005] proposed the coordinate attractor methods that move modules by calculating the direction to the goal position. However, their approaches on self-reconfiguration may not be applicable to the architecture, other than the existing ones that are designed with nearly all modules being active modules, which includes passive joints.

2.7 Discussion and Conclusions

Based on the review of the existing robots that have a certain level of resilience, three design principles for resilient robotic systems can be concluded with the following axioms:

Axiom 1: A robot should be designed with function redundancy. Redundancy means “when one physical part or subsystem (say A) does not run out of its full capability, part of A can be trained to fulfill the role or partial role of another physical part or subsystem” [Zhang and Luttervelt 2011]. Function redundancy means that a system can perform one function with many configurations or states. The design of the starfish resilient machine in Bongard et al. [2006] is an example of the application of this axiom.

Axiom 2: The structure of a resilient robot should follow modular architecture [Zhang et al. 2011]. In the modular architecture, all components have standard interfaces to interact with each other; thus, a modular system can easily be changed in terms of configuration. A modular architecture can be further viewed as having two types: (1) both components and their interfaces are standard, and (2) components are not standard but their interfaces are. Note that in the literature, modular systems refer to type 1 only. Type 2 modular systems allow for a change of the shape of components, which is the case in soft robots. Further, most self-reconfigurable robots are based on Axiom 2, that is to say, they are modular systems.

Axiom 3: The structure of a resilient robot should follow the so-called adjustable architecture [Zhang et al. 2011]. In fact, the structure of soft robots follows this axiom; see the Chembot

[Gizmag 2009], the SDM Hand [Dollar and Howe, 2010], and the universal gripper [Brown et al. 2011].

It is worth mentioning that the three axioms may be applied in an integrated manner to increase the degree of resilience of a robotic system. A couple of conclusions from the existing studies, with respect to the problem dimension for resilient robots previously defined in Section 2.1.4, can be drawn as follows.

First, the existing resilient robots were developed in an ad-hoc manner and these robots may have a certain level of resilience. Specifically, the current literature has great confusions regarding the notion of resilient robots, the name "resilient robot" sometimes used does not have the full spectrum of features expected of resilient robots, and resilient robots are neither self-reconfigurable robots nor soft robots but principles that underpin these can be well employed in resilient robots.

Second, there are three classes of strategies seen in robots, which enable them certain ability for recovery.

Third, all the robots that may have a certain degree of resilience seem to include active joints only. Passive joints and components have not been considered in the literature at all for enhancing resilience.

Fourth, most of the connection systems have independent actuation and they are limited to connection and disconnection only.

Fifth, configuration synthesis was mostly applied to the fully under-actuated robot, without consideration of the research on the under-actuated robot. The reconfiguration problem mainly exists for the existing self-reconfigurable robots that have all active modules.

CHAPTER 3
ARCHITECTURE OF THE RESILIENT ROBOTIC SYSTEM

Architecture is the backbone of a system. It specifies the types of modules and the types of relationships between the modules (also see the definition of architecture in Chapter 2). In this chapter, a study on the architecture of resilient robots is presented, particularly with the help of Axiomatic Design Theory (ADT) in order to examine the architecture more rationally. This chapter is organized as follows. In Section 3.1, resilient robots are defined based on the robot architecture. In Section 3.2, the recovery strategies are presented, which are derived based on the function-behavior-structure (FBS) theory [Zhang 1994, Zhang et al. 2005]. In Section 3.3, the resilient robot is analyzed using ADT and a new architecture of resilient robots is presented. In Section 3.4, the proposed resilient robot is justified based on several criteria. A summary is given in the final section.

3.1 The Concept of Resilient Robotic Systems

Robots are defined as "physical systems that can perceive its environment through sensors and act upon that environment through actuators" [Russell and Norvig 2010]. This dissertation particularly considered this physical system as a dynamic system (see the discussion in Chapter 2).

Based on the definition above, there are four components, "action, plan, sensor, and environment," for a robot [adapted from Coste-Maniè and Simmons 2000]. A typical architecture of a robot is shown in Figure 3.1.

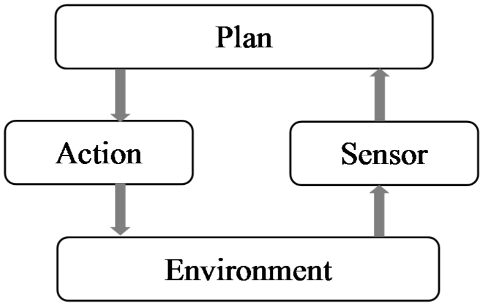


Figure 3.1. Architecture of intelligent robots [adapted from Russell and Norvig 2010].

A resilient robot is a robot that can recover its original function after a partial damage. The idea of the resilient system is associated with biological systems known to be self-healing. First, a biological system senses a wound. Second, the brain diagnoses the wound and makes a decision on how the wound may be healed. Third, the body implements the decision from the brain. Mapping such a self-healing process in robots, the three components (i.e., sense, plan, and act) of a resilient robot are described as follows:

Sense: the robot should have a monitoring system to detect failures.

Plan: the robot should have a system that can plan and schedule a change to a new state where the required function of the system can be retained from the failure state. This system may include a controller that performs the feedback control of change processes.

Act: the robot should have an actuation system to take a change action via the mechanical system based on the plan and schedule.

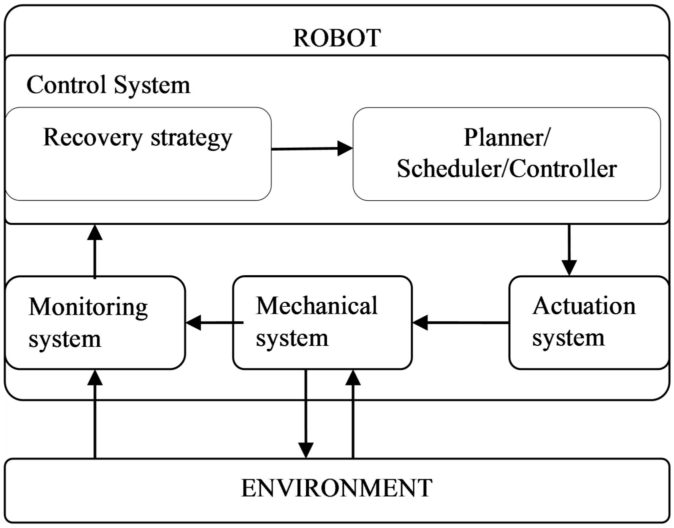


Figure 3.2. Architecture of resilient robotic systems from a functional viewpoint.

The general architecture of resilient robotic systems from a functional viewpoint is shown in Figure 3.2. A resilient robot has several components: (1) a monitoring system that serves to detect and diagnose the failed state; (2) a control system that deals with the feedback from the motoring system and sends commands to the actuation system; (3) an actuation system that implements the commands from the control system to change the (failed) robot to a new state; and (4) a mechanical system that realizes the recovery strategies. In (1), the analyzer processes sensory information and figures out the state of the remaining system. It can identify failures caused by both internal and external environments. In (2), there is a system to determine the recovery strategy, a planner/scheduler, and a controller to control the execution of the recovery process.

Note that failure identification is not in the scope of this dissertation. In this dissertation, the configuration of a damaged robot is assumed to be known. Recovery strategies will be further discussed in the following section. Later, the architecture of the resilient robot from the viewpoint of the mechanical system will be presented.

3.2. Recovery Strategies of Resilient Robots

The first and utmost task for recovery is to determine a so-called recovery strategy, the principle that governs a particular recovery process. To the writer's best knowledge, no one has studied the recovery strategies of a dynamic system. In the following, the general knowledge architecture of a system called FCBPSS [Zhang 1994, Zhang et al. 2005] is employed to derive the recovery strategy of resilient robots.

In FCBPSS, F: function, C: context, B: behavior, P: principle, S: state, and S: structure. Their relationships are shown in Figure 3.3. According to Zhang [1994] and Zhang et al. [2005], structure refers to components and connections among the components. The connections could be those topological or geometrical relations between components. State refers to attributes on the structure, e.g., mechanical property and location, etc. Principle refers to the fundamental laws with which one can develop a quantitative relation among the state variables. Behavior refers to the relationship among state variables, governed by the principle, and dynamic behavior refers to a sequence of states and transitions between them, which is governed by the principle, dynamics (stiffness, damping, inertia), and external controller to certain state variables. Context refers to the environments that surround a particular system. Function refers to utilities of the structure owing to its behavior in a context. A function only makes sense in a context. In a dynamic system, behavior is generally a function of time, event, and/or space [Zhang 2014b].

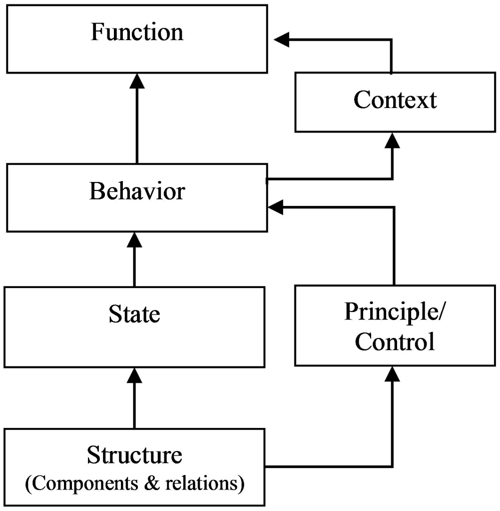


Figure 3.3. Function-context-behavior-principle-state-structure (FCBPSS) model of dynamic systems [adapted from Zhang and Lin 2010].

From FCBPSS, one can see that there may be many means (structure, state, controller, behavior, context, and principle) to change a system to attain a specific function. If the means to change is the structure, the recovery strategy reduces to the reconfiguration, and a resilient robot reduces to a self-reconfigurable robot. However, a resilient robot includes all the means, which separates the resilient robot concept from the self-reconfigurable robot. Along this line of thinking, one can derive the following strategies for recovery:

Strategy I: Training a failed system to have a new behavior, e.g., re-generation of the software controller in Figure 3.3.

Strategy II: Changing the configuration of a failed system to have a new configuration, which is the reconfigurable robot [Bi 2002].

Strategy III: Changing the states of components, e.g., changing the length of a bar component (see the so-called adjusting mechanism [Bi 2002, Ouyang et al. 2004]).

A combination of the above strategies is also possible, which may be called a hybrid strategy. Note that the above strategies are in fact derived from FCBPSS to change the function of a failed system, particularly via behavior, state, and/or structure (Figure 3.3). Strategy I refers to the change of a function via the behavioral change (i.e., change of the relationship among states). Further, the change of a behavior may be due to the change of the principle, and therefore this strategy may also refer to the change of principle. Strategy II refers to the structural change via the change of connectivity among components [Bi 2002]. Strategy III refers to the change of a function via the change of a component in itself. The change of a context to change the function of a robot is out of the scope of this thesis; however, this situation occurs in service systems, and the interested reader may refer to resilient service systems in Wang [2013].

In Figure 3.4, the first three recovery strategies are demonstrated in an example where a robot is assumed to be partially damaged. Originally, the robot moves by walking (A). After one leg is broken (B), the robot recovers its function (i.e., moving) by crawling (C_1 via strategy I), or re-arranging one of the remaining components (C_2 via strategy II), or changing the shape of one component (C_3 via strategy III).

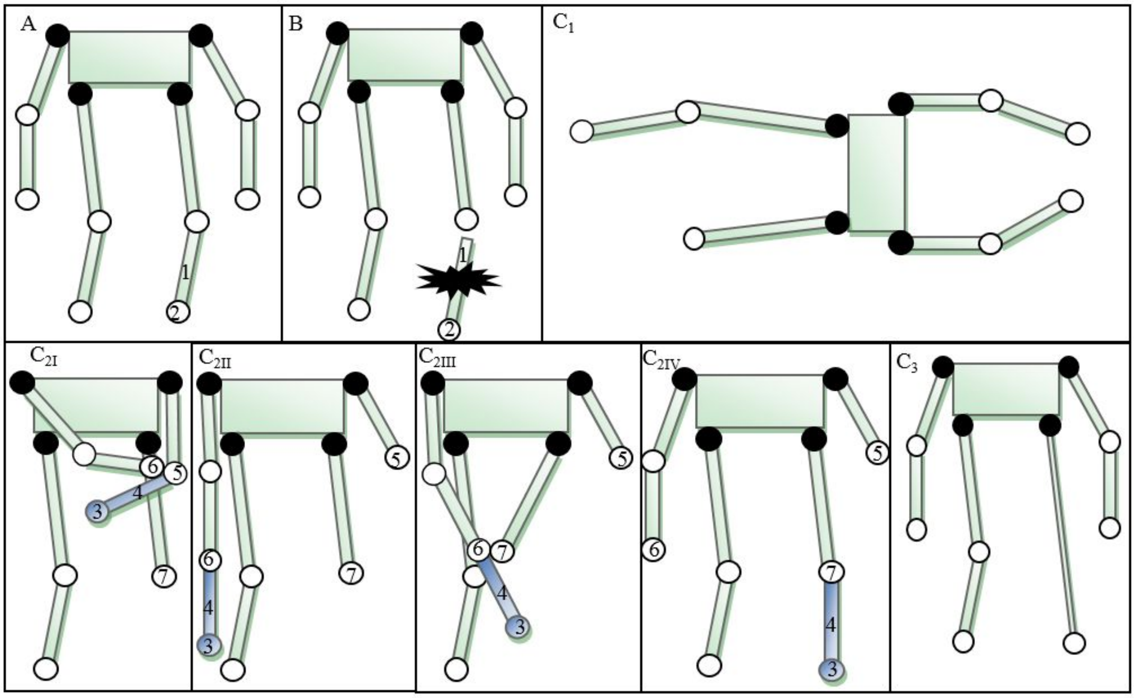


Figure 3.4. Robot recovers its original function through three strategies, denoted by C_1, C_2 and C_3, respectively. (A) Original state of a robot. (B) Part 1 damaged. (C_1) The first recovery strategy: the remaining system is trained to perform a new capability. (C_2) The second recovery strategy: the robot rearranges parts 3 and 4 (dark-colored) to where parts 1 and 2 used to be by reconfiguring the remaining system. This strategy is as follows: (C_{2I}) Part 6 connects with part 4; (C_{2II}) Part 4 disconnects from part 5; (C_{2III}) Part 7 connects with part 4; (C_{2IV}) Part 4 disconnects with part 6. (C_3) The third recovery strategy: the state of one part changes (i.e., extending part).

Recovery strategies II and III are in fact associated with the configuration changes of a robotic system. Therefore, at this point, one can see some similarities between reconfigurable robotics and resilient robotics. However, the concept of the resilient robot is wider than the concept of the reconfigurable robot. The reconfigurable robot refers to the means a robot recovers its function if the reconfiguration action is applied to the particular damaged robot.

3.3 The Design of the Resilient Robot Architecture

To define the resilient robot architecture, a systematic design theory called Axiomatic Design Theory (ADT) [Suh 1990] is applied. It is worth mentioning that the architecture design in this section is from a structural viewpoint, mainly about the mechanical system shown in Figure 3.2.

3.3.1 Axiomatic design theory (ADT)

Axiomatic design theory (ADT) is a system design methodology to systematically analyze the customers' needs and the design solutions [Suh 1990]. ADT defines the customers' needs in functional requirements (FRs), and defines the design solutions as design parameters (DPs) that are supposed to fulfill FRs. The design process can be viewed as a mapping from the functional requirements domain to the design parameters domain. The mapping can be represented as {FRs} = [A]{DPs}, where [A] is the design matrix based on the FRs and DPs and the design process is guided by [A]. ADT maintains the independence of the FRs, which suggests that all FRs should be uncoupled or decoupled. Note that some constraints could be imposed as part of the customers' needs, which provide bounds on the acceptable design solutions, and the constraints differ from the FRs in that the constraints, such as cost, do not have to be independent [Hommes 2010].

3.3.2 Functional Requirements

The architecture of resilient robot systems should recover the original function through the recovery strategies and meet other specifications. In this dissertation, the cost of resilient robots is also considered, which is easy to understand as the resilient robot is meant to reduce the loss of the robot after failure. Thus, cost can be viewed as a constraint for the design process.

Functional requirements (FRs) of resilient robots are classified based on the three recovery strategies. Behavior changing (Strategy I) requires that a system performs one function with many different configurations or states [Zhang et al. 2010], which means that the robot has function redundancy. Redundant machines are designed to have function redundancy. Typically, a component or a part of the component of the robot system for function - A should be trained to do function B of another component [Zhang and Lin 2010]. It is worth to note that the actuation redundancy and structural redundancy is mainly considered in this dissertation. Configuration changing (Strategy II) requires that the adjacent components can be disconnected and different components can be connected so that the robot can change into different configurations. Component changing (Strategy III) requires that the length of a component can be adjusted. Note that this dissertation only considers components made of rigid materials and not those soft materials with plasticity properties. Thus, the component may be assembled with several parts so that the component length changing can result from changing the connection between parts.

Based on the above discussion, the FRs to derive the general architecture of a resilient robot are given as follows:

- FR0: To generate different strategies of three types of robots in terms of the corresponding three recovery strategies.
- FR1: To enable variations of components of the three types of robots.
- FR2: To enable variations of assemblies for the three types of robots.

To derive the architecture of a resilient robot system, the above three FRs are decomposed and illustrated in Figure 3.5. Note that the cost is considered for the whole design process.

3.3.3 Design parameters

According to the Axiomatic Design Theory (ADT), a set of design parameters (DPs) will be identified to satisfy the FRs. Figure 3.5 shows the DPs of the resilient robot architecture, which corresponds to the FRs. The meanings of these DPs are illustrated in Figure 3.5. Note that passive joints, rigid links, and passive connectors do not have actuation systems and thus they have lower energy consumption, which satisfies the constraint, i.e., low cost.

According to the ADT, the proposed design satisfies the Independence Axiom. Here, an example of decomposition of FR0 and DP0 is given to illustrate this point. FR0 can be decomposed into three sub-FRs as follows.

- FR01: Design a robot that can change the behavior after a partial damage.
- FR02: Design a robot that can change the relationships between modules after a partial damage.
- FR03: Design a robot that can change the component state after a partial damage.

Robot	Component	Assembly
FR01: To change behavior **FR02**: To change configuration **FR03**: To change component	**FR11**: To provide motions with power source **FR12**: To provide motions without power source **FR13**: To provide different structures	**FR21**: To disconnect with other components **FR22**: To connect with other components **FR23**: To unlock a fixed connection **FR24**: To lock an unfixed connection
DP01: Robot with function redundancy **DP02**: Modular robot with reconfigurability **DP03**: Modular robot with adjustability	**DP11**: Active joints **DP12**: Passive joints **DP13**: Rigid links	**DP21**: Passive connectors that can be disconnected **DP22**: Passive connectors that can be connected **DP23**: Passive connectors that can be unlocked **DP24**: Passive connectors that can be locked

Figure 3.5. Functional requirements and design parameters for resilient robot architecture.

The DP0 is decomposed as follows.

- DP01: Build a robot with function redundancy.
- DP02: Build a modular robot that has reconfiguration capability.
- DP03: Build a modular robot that has adjustable modules.

Thus, according to the ADT, the relationship between FRs and DPs can be represented by the following design equation:

(3.1)

Similarly, the mapping between FR1 and DP1, FR2 and DP2 can be written by Eq. 3.2 and Eq.

3.3, respectively.

(3.2)

(3.3)

From the above design equation, the design matrices are triangular or diagonal matrices, which indicates that the design is decoupled and the independence of the FRs is satisfied.

3.3.4 The General Architecture of Resilient Robots

Based on the DPs, the definition of the general architecture of resilient robots is given with the highlight of the following features.

(1) A general resilient robotic system consists of two types of modules: joints and links. Further, joints are divided into two types, active joints and passive joints, and links are divided into two types, passive rigid links and passive adjustable links. Devices that do not have a power source are considered “passive,” and “their behavior is fully determined based on their passive dynamics,” as pointed out by Collins et al. [2005].

(2) The active joint module and rigid link module are the fundamental parts that have independent functions. Neither of them can be disassembled. Note that the link module is the same as the rigid link module in this dissertation unless otherwise stated.

(3) All modules have identical ports that are used to connect other modules in a configuration. A connection system is formed with a pair of ports. A pair of ports of two

modules could be connected, disconnected, locked, or unlocked. These actions on two modules could be achieved by the actuation of the neighboring joints of the two modules.

(4) A passive joint is formed with two link modules. There is a relative motion between them when the connection system is unlocked. In this dissertation, the passive joints could be disassembled.

(5) A passive adjustable link is a fixed assembly of two link modules. The length of the assembly can be adjusted by unlocking the connection system from one status and moving to another locked status. Passive adjustable links could be disassembled.

As can be seen from the above features, the robot includes both active and passive joints. The resilient robots in this dissertation are under-actuated. Under-actuated robots are “the robots with both active and passive joints,” as defined in Bergerman and Xu, [1996]. Later in this section, there will be more discussion on Features (4) and (5).

To understand and study the architecture and system, representations using words, code, drawing, etc. are used. There are several approaches to represent the architecture, such as hierarchical components structure, function structure [Pahl and Beitz 1999], design structure matrix (DSM) [Kruchten 1995], and Object–Process Methodology (OPM) [Dori 1998]. This dissertation uses OPM to represent the system architecture as it models functions and forms and their interrelationships in one graph, which makes the architecture easily understood. Here, the function consists of the action and the operand which is acted on [Crawly 2007] and form consists of the elements and their structure. From a structural viewpoint, the architecture of the resilient robot in this thesis is represented by OPM, as shown in Figure 3.6.

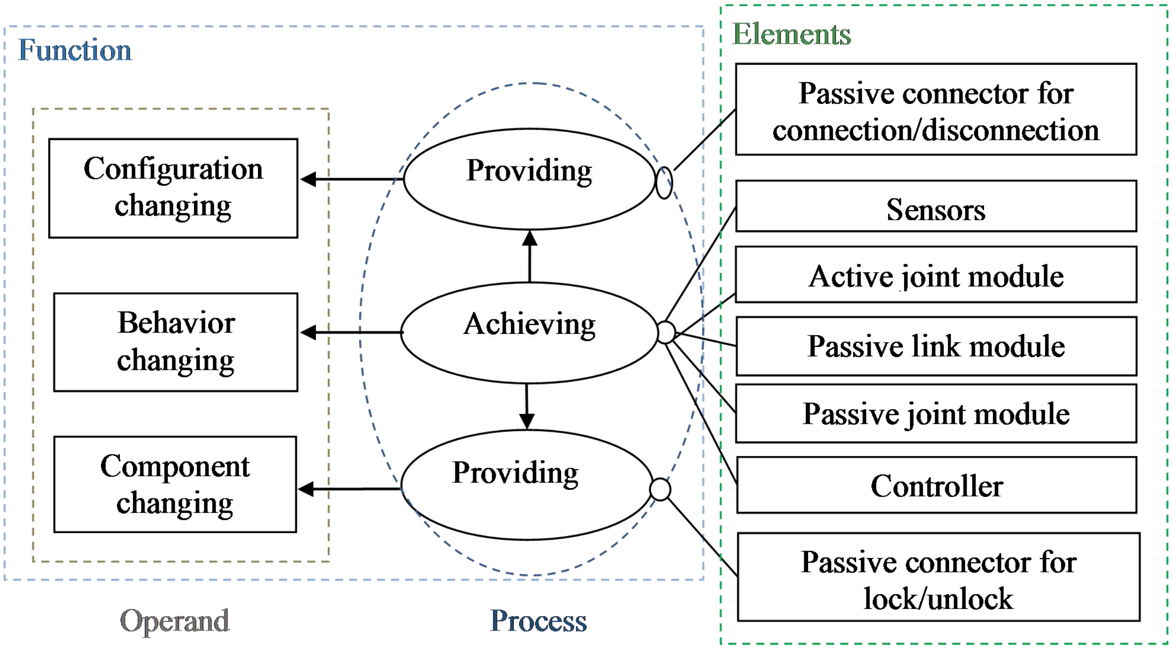


Figure 3.6. Architecture of the resilient robot from a structural viewpoint.

The challenge of realizing a resilient robot that follows the proposed architecture is the connection system, which could be realized by the key-lock-based connector, as shown in Figure 3.7A. Both active joint modules and link modules have a key-lock-based connection system (Figures 3.7B and 3.7C). A passive joint is formed with two rigid link modules when the key and lock on each module are connected and located within some range (Figure 3.7D). There is a relative motion between the two links, called passive connection in this thesis. When the key and lock of the two modules are connected and locked, the assembly with the two modules is fixed and it is called an adjustable link (Figure 3.7E). This link can be adjusted when the key is rotated to another range and then locked to another range. A schematic diagram for the adjustable link is shown in Figure 3.7F.

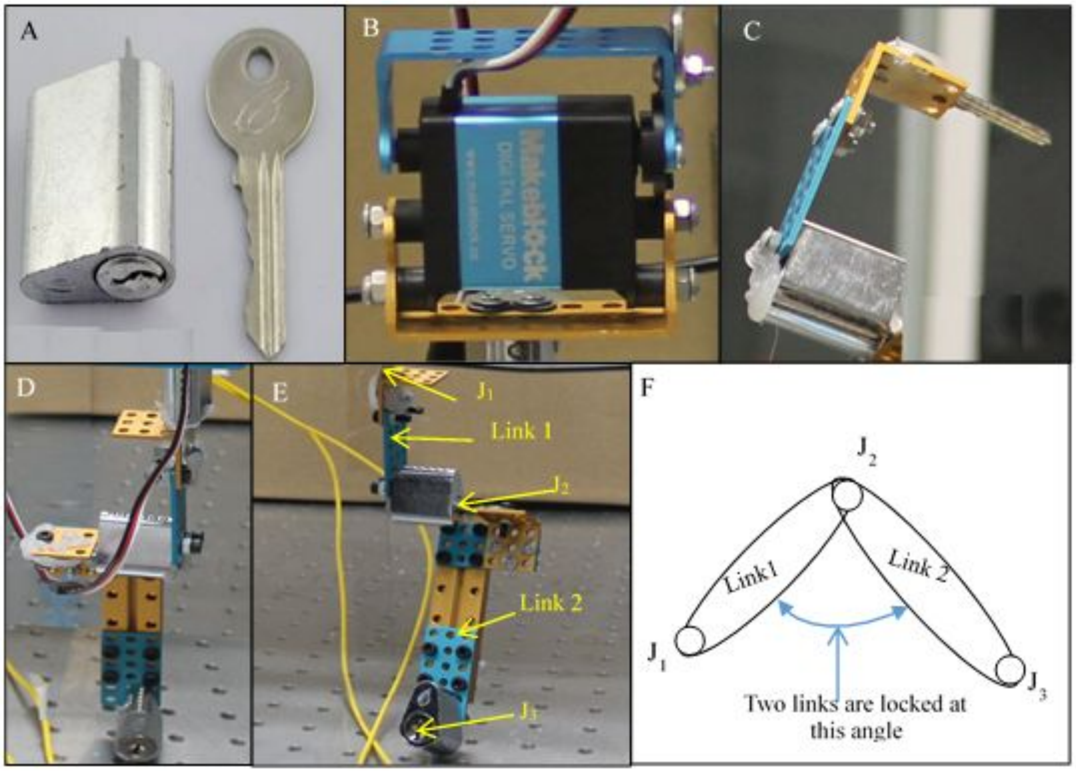


Figure 3.7. Architecture of the resilient robot from a structural viewpoint [Zhang et al. 2014a]. (A) a pair of ports: a lock and a key, (B) an active joint, (C) a link module that has a female port (a lock) and male port (a key), (D) a passive joint with two link modules, and (E) an adjustable link with two link modules. (F) A schematic diagram for the adjustable link in (E).

It is worth noting that the inclusion of passive joints is a good way to decrease the energy consumption. Besides, it is obvious that the robots can easily overcome the actuator failures, especially the following two failures: an active joint is unlocked and becomes a passive joint and an active joint is locked and becomes an adjustable link. The video can be found at https://www.youtube.com/watch?v=I_QRMryR_xQ.

Even though the key-lock-based connection system satisfies the architecture of the resilient robot in this dissertation, the performance of the robot is not good. This is because of the existing heavy key and lock and the large size. More importantly, there is a slight gap between key and lock, which leads to inaccuracy of the assembly. In Chapter 4, a better connection system based on the key-lock concept will be presented.

3.4 Justification for the Proposed Architecture

The architecture proposed in this dissertation meets the following criteria from different viewpoints [Crawley 2007, Kruchten 1995] to prove that it is correct.

First, from a logical viewpoint, this architecture meets all the functional requirements of the end user. As shown in Figure 3.6, the architecture satisfies all the functional requirements set out by the recovery strategies. However, most of the existing architectures of reconfigurable robots such as self-reconfigurable robots and soft robots address only one or two of the three strategies. The proposed architecture has the most coverage of the recovery strategies. As well, there are only two types of fundamental modules for a robot, i.e., active joint modules and link modules, which leads to the conclusion that the failure of active modules may simply change the role of the modules instead of rendering the system completely useless. Indeed, the under-actuated robot architecture is more general than the fully-actuated robot architecture in that if a fully-actuated robot is partially damaged, such as failure in one or more active joints, the robot may still be changed to play its original function. In short, the under-actuated robotic architecture is a novel feature with robotic systems and it will certainly increase the degree of the resilience of a robotic system.

Degree of resilience is the key point in this dissertation, which distinguishes the proposed under-actuated resilient robot from the existing resilient robots. Under-actuated robots have a low mobility actuation system but they achieve full mobility. It is known that if both under-actuated resilient robot and fully-actuated resilient robot have the same degree of resilience, the under-actuated one should be preferred (due to low cost).

Second, the under-actuated resilient robot considers the requirements related to the ease of developments that support the cost evaluation. The inclusion of passive joints can reduce manufacturing cost compared with a robot with identical active joints. The adjustable link is designed based on the rigid link and the connection system. There is no need to design links with a different length.

Third, the proposed architecture takes into account a system's non-functional requirements, "such as performance and system availability" [Kruchten 1995]. The under-actuated robot with passive joints has more diversified behaviors as the passive joints can interact with environment.

3.5 Summary and Conclusions

This chapter proposed a general architecture for the resilient robot. To make the derivation of the architecture more rational, defining this architecture was viewed as designing a "system." Owing to its rationality, therefore, the design theory ADT was applied. To apply ADT, the basic notions of FRs and DPs must be associated with an application. In this connection, the three recovery strategies were defined into the FRs, and the architecture was characterized by the DPs. In addition, this chapter also provided an account for the underlying reasons for the three recovery strategies concerned in this dissertation by employing the general knowledge architecture called FCBPSS with a slight extension to the dynamic system.

The architecture was represented diagrammatically with the so-called Object-Process Model (OPM). Indeed, the robot out of the proposed architecture is an under-actuated resilient robot which is the most general robot in terms of the architecture (because of the inclusion of passive joints and links along with active joints, and passive chains along with active chains). The proposed architecture was justified as possessing the highest degree of resilience as opposed to the existing robot architecture.

CHAPTER 4
A NOVEL DOCKING SYSTEM

This chapter discusses the challenges of incorporating the proposed architecture in the design of the mechanical system of the resilient robot. The main challenge here is the connection system, which is also called the docking system (see the discussion in Chapter 2). This is because the realization of both reconfigurability and adjustability of the resilient robot much depends on the docking system. A docking system is a pair of interfaces on two components that provides a connection between the two components. This chapter is organized as follows: Section 4.1 introduces the docking system design; Section 4.2 gives an overall design of the under-actuated resilient robot; Section 4.3 discusses the docking process and demonstrates the recovery process of a resilient robot; and Section 4.4 provides a summary.

4.1 Docking System Design

In Chapter 3, the key and lock principle was used to demonstrate the docking system (Figure 3.6). In the following, an improved physical docking system (as opposed to the one introduced in Chapter 3) is presented. The Axiomatic Design Theory (ADT) will be employed to examine the docking system to improve the quality of design.

4.1.1 Functional requirements for the docking system

Based on the resilient robot architecture proposed in Chapter 3, it can be seen that the introduction of the docking system to each module will eliminate the need for a separate connector. Further, with key-lock concept docking system, the passiveness, adjustability, and reconfigurability are in fact realized by the docking system. Note that the passiveness is not only a functional requirement (FR) (i.e., to form a passive joint) but also a constraint for the whole resilient robot (i.e., to reduce the cost). According to the key-lock principle, the docking system has a male interface and a female interface. Noted that both key and lock interfaces are identical in this dissertation unless otherwise stated. Figure 4.1A presents the simplicity of the male interface. Therefore, in this dissertation, only the female interface will be considered. The functional requirements for the docking system in the context of the physical system are as follows.

- FR01: Passiveness.
    - FR11: To generate a geometric boundary that allows the male interface to rotate freely.
- FR02: Reconfigurability.
    - FR21: To generate a geometric boundary to fix the male interface at a certain position.
    - FR22: To generate a space for the male interface to release from the female interface.
- FR03: Adjustability.
    - FR31: To generate a geometric boundary to fix the male interface at different locations.
    - FR32: To provide a space to move the male interface to another location.
    - FR33: To generate a space for the male to release from the location.

4.1.2 Design parameters for the docking system

Based on the foregoing functional requirements, the docking system is designed with the following design parameters (DPs).

- DP01: Passive joint.
- DP02: Connection and disconnection.
- DP03: Lock and unlock.

To achieve good modularity with the guidance of ADT, the docking system (i.e., female interface) is designed with several layers, as shown in Figure 4.1B. Therefore, the above DPs can be decomposed as follows.

- DP11: Passive connection layer (Layer 1 and Layer 2).
- DP12: Fixation layer (Layer 3).
- DP13: Adjustment layer (Layer 4).
- DP14: Release and unlock layer (Layer 5).
- DP15: Elastic force device (e.g., spring).
- DP16: Insertion and release area.

Thus, according to the ADT, the relationship between FRs and DPs can be represented by the following design equation:

(4.1)

From the above design equation, the design matrix is a non-triangular matrix, which means that the design mapping is coupled. To decouple the design mappings, the matrix needs to be rearranged or reordered into a triangular one. As can be seen from Eq. 4.1, the second row and the fourth row are identical, which means the corresponding FRs (i.e., FR21 and FR31) are the same. In this case, one of the two rows can be removed and replaced with a row of zeros, see Eq. 4.2. Note that the DPs could be identical if these DPs are at different levels, for example, when the DPs are actuators.

(4.2)

Then, the matrix in Eq. 4.2 can be reordered into a triangular matrix, see Eq. 4.3. It can be seen that the independence of the FRs is satisfied.

(4.3)

4.1.3 Docking system structure

Based on the design parameters, the structure of the docking system is designed in Figure 4.1. The female interface has several layers and it is spring-loaded at each layer to form different types of connections. Each module has two male interfaces and two female interfaces on each of its four sides. The reconfigurability and adjustability can be realized by the docking system as follows. By inserting the male interface to the female interface along the "inserting and releasing area," rotating the male interface in Layer 2, and then releasing it, the two interfaces are then connected and have relative motion, as shown in Figure 4.1E; in this situation, a passive joint is formed. Rotating the male interface to the inserting and releasing area and releasing the male interface, the two interfaces will be disconnected. When the male interface is inserted into Layer 5 and then rotated, the male interface will be bounced and fixed at the position between two teeth of Layer 4, as shown in Figure 4.1F; in this situation, the two interfaces are fixed and a passive link is formed with the two modules. The length of the new link can be adjusted by unlocking the connection (i.e., pushing the male interface to Layer 5) and locking the connection (i.e., releasing the male interface when it is rotated to another place between another two teeth); in this situation, the so-called adjustable link is achieved. The adjustable link can be disassembled by unlocking the connection and rotating to the insertion section and releasing it. The adjustable link can also be switched to a rigid link by unlocking the connection and rotating to the insertion section and then rotating the male interface when it goes to Layer 2.

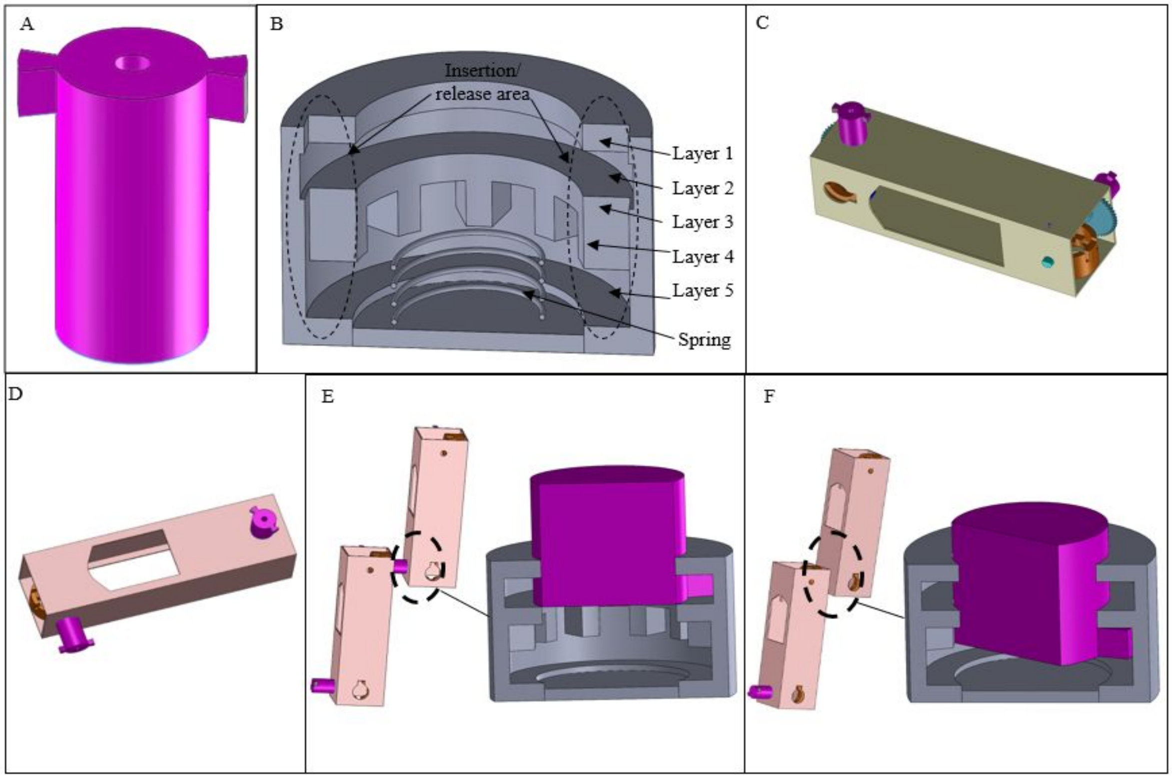


Figure 4.1. The modules of an under-actuated self-reconfigurable robot. (A) Male interface. (B)

Female interface. (C) A joint module with two female interfaces and two male interfaces. (D) A link module with two female interfaces and two male interfaces. (E) A passive joint and a cross-section view of the interface connection with the male interface located in Layer 2 of the female interface. (F) An adjustable link with two link modules and a cross-section view of the interface connection with the male interface located in Layer 4 of the female interface.

4.2 Hardware of the Resilient Robot

There are two types of modules: active joint modules and link modules. Each can be connected to each other through the docking system on all four sides of each module. Each module has two female interfaces and two male interfaces, as shown in Figure 4.2, and the physical prototypes of the link modules and active joints are shown in Figure 4.3. An active joint module has two actuators, which are the power supply of all movements, such as locomotion, connection/disconnection, and lock/unlock (this will be discussed later in this dissertation). The motor of each active joint is connected to a larger gear and a smaller gear for increased speed. The motor is transmitted to the two output axes at which the module connected with them is able to move. Figure 4.2 shows the mechanism of an active joint module, which consists of two motors, two gears, two transmission shafts, two female interfaces, and two male interfaces. Table 4.1 shows the attributes of active modules and link modules.

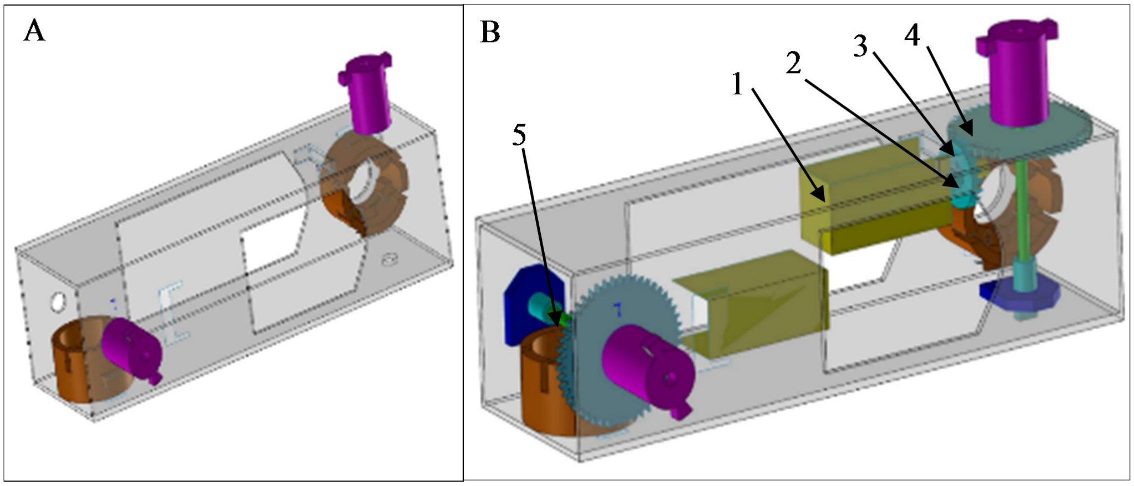


Figure 4.2. The structure of link modules and joint modules. (A) The link module with two female interfaces and male interfaces. (B) The mechanism of an active joint module: servomotor (1), transmission shaft (2), small gear (3), big gear (4), and output axis (5).

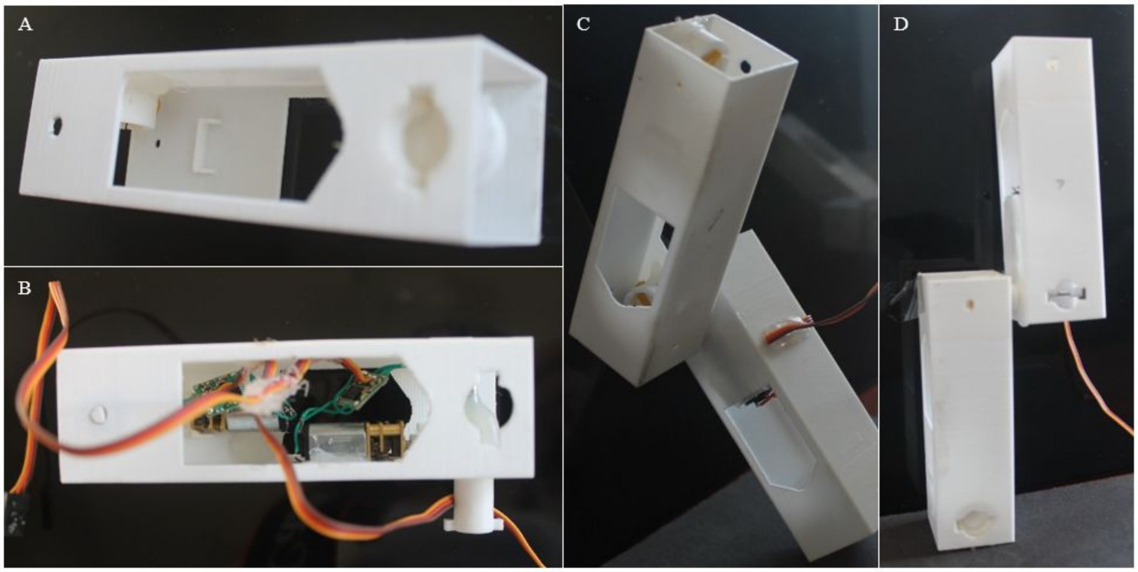


Figure 4.3. The hardware. (A) A link module. (B) A joint module. (C) An adjustable link formed by a link module and an active joint. (D) A passive joint formed by an active joint and a link module.

The communication structure and protocol on the motors were built. The control algorithm can be run on a laptop computer. The controller takes the sensor input from the physical robot and then calculates the corresponding parameters for each active joint. The information of the parameters is then sent to the motors.

4.3 Docking Process and Experiment

According to the architecture of the resilient robot, basic locomotion and reconfiguration are all based on the active joints and passive joints. The reconfiguration process is based on the docking system, which is actuated by active joints. Passive joints are not directly controlled. They are connected with active joints, so they are “looked after” by active joints. During the docking process, the active joints contribute to all of the motions. The docking process has three stages. First, two modules to be connected move so that two interfaces are positioned close to each other. For example, for a snake root to become a circle, the active joints will move to bend the robot to bring the head close to the tail. Second, the male interface will be guided to the female interface to satisfy different constraints. Depending on the position of the male interface, different connections (passive connection or locked connection) are formed. Third, once the male interface is pushed to a desired position, the connections must be rotated to establish the final connection. The second and third stages are complex and need higher accuracy.

Table 4.1. The attributes of active modules and link modules.

Module type	Description	Value
Link module	Weight, g	100
	Overall dimension (L/W/H), mm	110/32/32

	Length between two interfaces	90
Joint module	Weight, g	200
	Overall dimension (L/W/H), mm	110/32/32
	Length between two interfaces, mm	90
Potentiometer	Voltage (V)	5
	Weight, g	0.36
Motor (N20)	Voltage (V)	1.5-6
	Weight, g	9.5
	Speed (rmp/min)	100
	Torque (kg.cm)	0.15

In the experiment, the robot consists of 21 modules, as shown in Figure 4.4. The aim is to connect Module 1 with Module 10. Five runs were made to demonstrate the reconfiguration process. The average speed is approximately two minutes for docking and one and a half minutes for undocking. The experiments are autonomous and modules are powered by an external power supply and controlled by the program through a PC. The whole reconfiguration took about ten minutes.

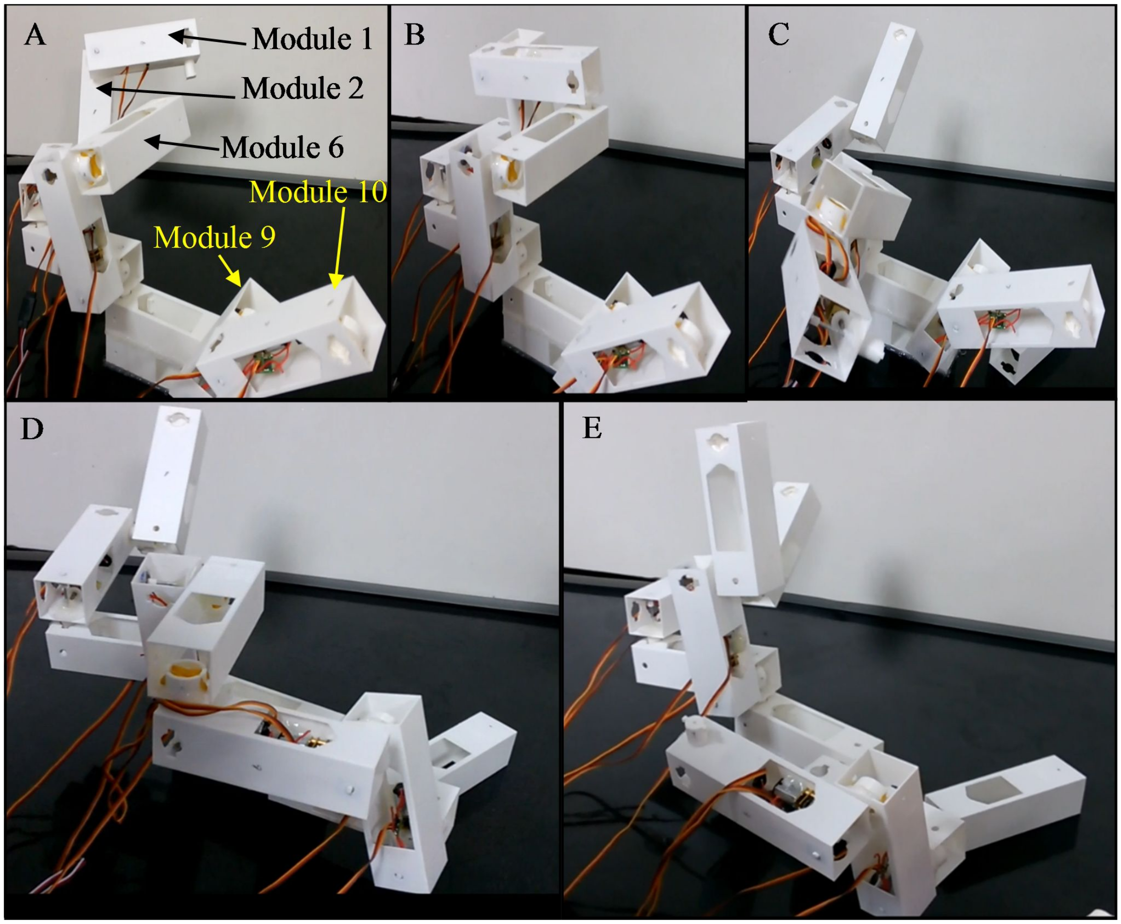


Figure 4.4. Self-reconfiguration experiment. (A) Original configuration, (B) Module 6 connects module 1, (C) Module 1 disconnects from Module 2, (D) Module 1 connects Module 10, (E) Module 1 disconnects from Module 6.

The experimented demonstrated that the performance of the robot is good, even if the robot has adjustable links and passive joints. It is worth mentioning that the two types of common failures for robots are that joints are locked or unlocked. For the robot in the experiment, these two types of failures can be viewed positively as the adjustable link (for the failure of a locked active joint) and the passive joint (for the failure of the unlocked active joint), respectively. In this case, it can be understood that the robot as developed can easily deal with failures of the active joints.

4.4 Summary

This chapter presented a docking system based on the architecture of a resilient robot. With the introduction of the docking system, the most distinctive feature of the under-actuated resilient robotic systems developed in this dissertation is that there are only two types of components, i.e., link modules and active joint modules. Passive joints and adjustable links can be formed with two modules by locating the dock at different positions. Docking and undocking actions do not need a separate actuation but by the actuators that are the power supply of locomotion. Therefore, the energy consumption for locomotion and reconfiguration is greatly reduced compared with existing self-reconfigurable robots. Besides, the design of the under-actuated resilient robot not only reduces the cost but also overcomes the two common actuator failures: an active joint is unlocked (thus becoming a passive joint) or an active joint is locked (thus becoming an adjustable link). This feature (i.e., the failure of active modules may simply change the role of the modules instead of rendering the system completely useless) is a novel feature with the robotic system, and it will certainly increase the degree of the resilience of the robot. In addition, the reusability of a damaged component or module is exhibited, which suggests a new dimension of green products and systems. This new dimension differs from the existing green products and design in that a product or module has multiple uses depending on its configuration and/or state.

CHAPTER 5
CONFIGURATION SYNTHESIS

As discussed in Chapter 1, an important research question is: how is original function recovered after a robot suffers partial damage? To answer this question, the first issue is the finding of a configuration of the damaged robot, upon which the damaged robot can still function. This configuration is called target or goal configuration. Note that there may be more than one goal configuration. Therefore, the “best” one makes sense; in this dissertation, finding the best goal configuration is called configuration synthesis. This chapter presents an approach to goal configuration synthesis.

This chapter is organized as follows. In Section 5.1, the representation of the configuration of a resilient robot is given. In Section 5.2, the optimization model of configuration synthesis is explained. In Section 5.3, a computational model is described to implement the model. In Section 5.4, a design case with discussion is given. Section 5.5 provides a summary.

5.1 Configuration Representation

The representation of the configuration of a robot becomes the first step for configuration synthesis. The applicable definition of configuration refers to Definition 2.3 in Chapter 2. The representation of a configuration includes the representation of modules and their connections in space. The following matrix is proposed for the representation of the configuration.

Each row and each column represents modules. If there are *m* modules in an assembly, the matrix is $m \times m$. Each element represents the detailed information of how two modules or objects are connected, namely, the information of the connection ports and the connection type. Note that there are two types of connections: fixed connection or relative motion.

Therefore, each element in the matrix has five numbers that give the detailed information of the ports of the two respective modules in connection. For instance, for row p and column k, the element has the following format $<i_1, i_2, i_3, i_4, i_5>$, where i_1 represents the type of Module p. i_2 represents the port of Module p, which will be connected with Module k. Similarly, i_3 and i_4 represent the type of Module k and the port of Module k that will be connected with Module p. Note that i_1 and i_3 is "1" or "2" where "1" means link and "2" means active joint. Both i_2 and i_4 are "1," "2," "3," or "4," which represents the port ID on one module. The last number, i_5, denotes the connection type of Module k and Module p. Particularly, i_5 is "1" or "2" where "1" means passive connection or joint, and "2" means fixed connection or joint. Note that the element (i, j) takes 0 if Module i and Module j are not connected.

Based on the above discussion, the matrix representation for the robot shown in Figure 5.2 is given below:

$$\begin{array}{cccccc}
 & <\mathit{Module\ 1}> & <\mathit{Module\ 2}> & <\mathit{Module\ 3}> & <\mathit{Module\ 4}> & <\mathit{Module\ 5}> \\
<\mathit{Module}\ 1> & 0 & <2,4,1,2,2> & 0 & 0 & 0 \\
<\mathit{Module}\ 2> & <1,2,2,4,2> & 0 & <1,4,2,2,2> & 0 & 0 \\
<\mathit{Module}\ 3> & 0 & <2,2,1,4,2> & 0 & <2,4,1,2,2> & 0 \\
<\mathit{Module}\ 4> & 0 & 0 & <1,2,2,4,2> & 0 & <1,3,1,1,1> \\
<\mathit{Module}\ 5> & 0 & 0 & 0 & <1,1,1,3,1> & 0
\end{array}$$

Figure 5.1. The representation of the configuration in Figure 5.2.

In the above matrix, the element in row 1 and column 2 has the following meanings. The first number, 2, means that Module 1 is an active joint module. The third number, 1, means that Module 2 is a link module. The second number, 4, and the fourth number, 2, mean that Port 4 of Module 1 connects Port 2 of Module 2. The fifth number, 2, means that this connection is a fixed connection. Clearly, the complete information of a particular configuration of the robot can be determined by this matrix.

5.2 Optimization Model for Configuration Synthesis

5.2.1 Problem definition

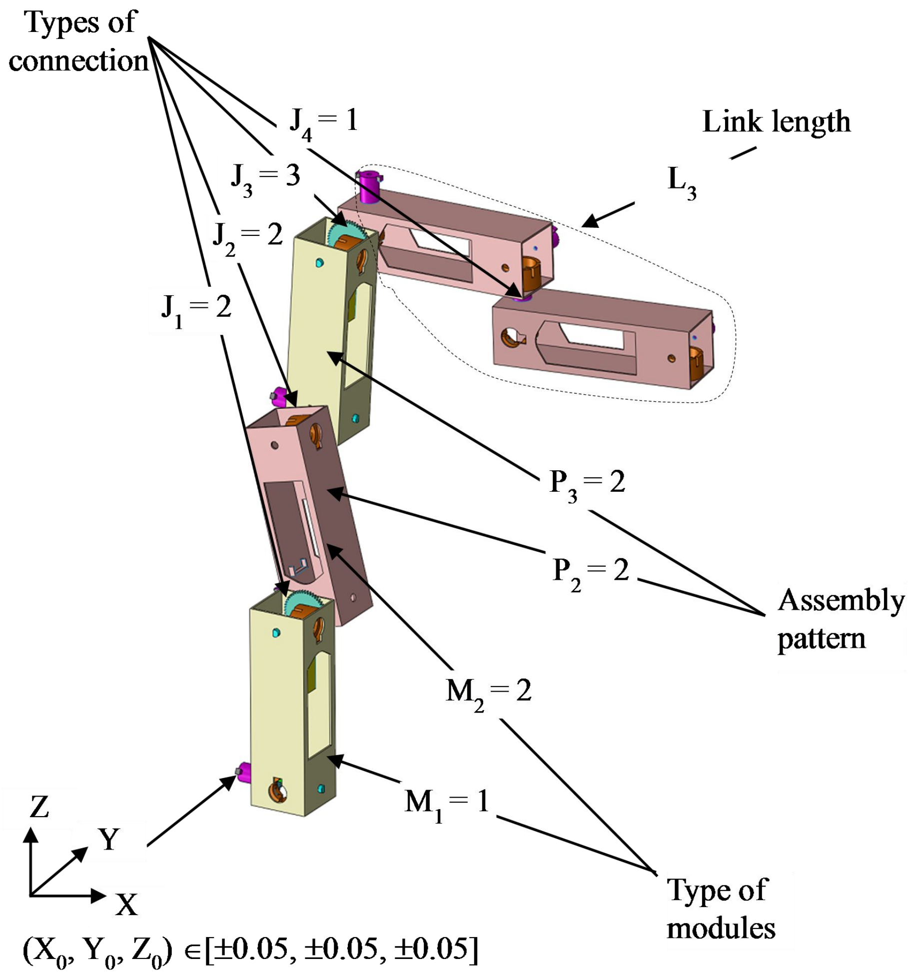
Configuration synthesis involves formulating an optimal configuration to achieve the given task, including design variables, design constraints, and design objectives. This dissertation focused on configuration synthesis for task-oriented under-actuated resilient robots. The task is specifically defined as a set of working points that are followed by the end-effector in the workspace without loss of the generality of discussions. The definition of configuration synthesis refers to Definition 2.9 in Chapter 2. In the context of optimization, it is viewed as an optimization problem including design variables, design constraints, and design objectives, which was discussed in Section 2.5.1.

5.2.2 Variables

The configuration is determined by the variables, i.e., the type of links, the type of joints, and the type of assemblies. These variables are coded sequentially, and the code thus represents the configuration of a robot. It is noted that they are convertible to the kinematic and dynamic variables or parameters of the robot. The variables are detailed as follows: 1) the type of joints: J; 2) the assembly pattern between modules: P; 3) the lengths of links between two adjacent joints: L; and 4) the initial location of the base: I. Note that variables J and P are discrete variables, while L and I are continuous variables. When these variables are instantiated, a robot configuration is defined uniquely. Thus, we consider the set of variables V as:

$$V= \{J, P, L, I\} \tag{5.1}$$

Figure 5.2 shows an example of a configuration determined by the above variables. The robot is a three-degree-of-freedom (DOF) robotic manipulator with two active joint modules and three link modules. The variables are marked in Figure 5.2.



J_i(i= 1,2,3)	Types of connections between modules, where i identifies a connection in an assembly. 1, 2, and 3 denote fixed connection, active joint connection, and passive joint connection, respectively. It is worth noting that the default connection is a fixed connection if it is not pointed out. Therefore, J refers to types of joints.
M_i(i=1,2,…,6)	Type of modules. Note that the information about the type of a module is included in J, so M is not included in the variables.
L_i(i=1,2,3)	Length of a link, i.e., the length between adjacent joints.
P_i(i=1,2,…,6)	Assembly pattern of each module. Each module has four ports, so there are six assembly patterns for each module.

Figure 5.2. Example of a 3-DOF configuration.

5.2.3 Constraints

Constraints ensure the feasibility of a robot configuration to perform a given task. Therefore, constraints can be defined corresponding to the task specification. In this dissertation, both kinematic and dynamic requirements will be considered. Kinematic constraints geometrically restrict the direction of mobility and dynamic constraints ensure the dynamic balance [Arai 1996]. Different configurations have different kinematic parameters and dynamic parameters. The method of the automatic generation of D-H kinematic parameters and dynamic parameters was studied, which can be referred to the previous work [Bi and Zhang 2001, Zhang et al. 2014b].

For an under-actuated robot, the kinematic relationship among θ_a and θ_p and x is expressed using the Jacobian matrix [Yoshida 1997]:

(5.2)

(5.3)

where x is the position of the end-effector, $x=[x_1, x_2 \ldots x_m]^T$, and is the given task (i.e., points). J_a and J_p are the Jacobian matrix for the active joints and passive joints.

The dynamic model for under-actuated robots takes on a form similar to fully actuated robots [Roberts 2001]. The dynamic equation of motion is generally expressed as follows:

(5.4)

where θ_a is the generalized coordinate for active joints, while θ_p is for passive joints. H_{aa}, H_{ap}, and H_{pp} are inertia matrices. b_a and b_p are Coriolis, centrifugal and gravity forces on active and passive joints, respectively. τ_a is the vector of torques applied at the active joints.

The kinematic requirement of joint *k* at task point *i* can be modeled as:

(5.5)

where and are the two boundary displacements.

Given the payload at the task space for a point P_i, say F_i, one can obtain the corresponding forces/torque on the joint modules, say f_{ik}. The design task to subject the robot end-effector to the described payload, F_i, can be modeled by the following constraints:

(5.6)

where is the maximum deriving force/torque that a joint module can generate.

5.2.4 Objectives

Two types of criteria are used to evaluate the configurations of the robots: manipulability and energy consumption. Manipulability is defined as “the ability to position and orient the end-effector” in Yoshikawa [1985]. Here, the dynamic manipulability is used to quantify the manipulator’s dexterity [Roberts 2001].

(5.7)

(5.8)

(5.9)

where denotes the dexterity of robot whose passive joints are . Note that the passive joints are considered to be free-swing joints. denotes the matrix H^{-1} with its columns removed. J and H are the Jacobian matrix and inertia matrix of the fully-actuated robot.

In this dissertation, three types of failures are considered for a resilient robot: i) one or more joints are locked, ii) one or more active joints are unlocked, and iii) the above two types of failures occurred simultaneously. The dynamic manipulability of a robot with all three types of failures [Roberts 2001] can be expressed as:

(5.10)

The superscript "u" and "l" denotes that the active joints are unlocked and joints are locked, respectively. is the dexterity of the robot in which joints is unlocked and joints is locked. denotes the matrix J with its columns removed. denotes the matrix H^{-1} with its rows and columns removed, and columns removed. Then the criterion of manipulability to be minimized is as:

(5.11)

The energy consumption is similar to the index of power consumption [Paredis 1996]. Note that the inclusion of passive joints is a good way to decrease the energy consumption. During the motion, the main cost controlled by configuration design is its energy cost in operation; this index is calculated as follows:

(5.12)

where *n* is the number of the joint axes of a configuration; *E* is total energy consumption along the trajectory of task; *Tra* is the space consisting of all working points along the trajectory; and $\tau_i(\theta)$ is the torque executed on the motion axis *i*.

The complete resulting objective function to evaluate the configuration is as follows:

(5.13)

where F_G is the global measure for a configuration candidate based on a task specification. k_1 and k_2 are the weights for the criteria of manipulability and energy consumption.

5.3 Implementation by Genetic Algorithm (GA)

The design problem has both discrete variables and continuous variables along with higher order derivatives of the objective function, and this makes the configuration difficult. In order to solve the configuration synthesis problem efficiently, the genetic algorithm (GA) is adopted to create the optimal configuration. In GA, a fixed-length binary string is used to represent a design variable, which is encoded as the chromosome as shown in Figure 5.2. This string is a set of values of each variable.

Genetic Code

In the optimization model, the variable is represented as an n dimensional vector $X=(x_1, x_2, \ldots, x_n)^T$. The gene string can be encoded into a chromosome with a string in Figure 5.3.

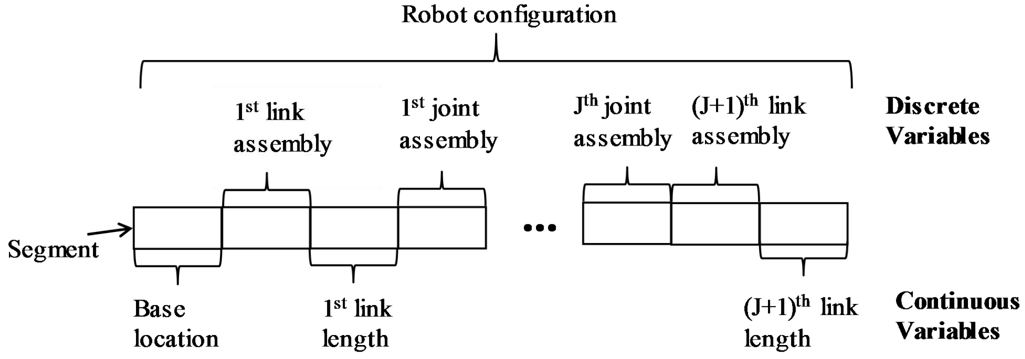


Figure 5.3. Encoding a robot configuration into a chromosome with a string.

Design of the fitness function

With GA, an initial population of random chromosomes is generated. The fitness of each configuration (i.e., each chromosome) is as follows.

(5.14)

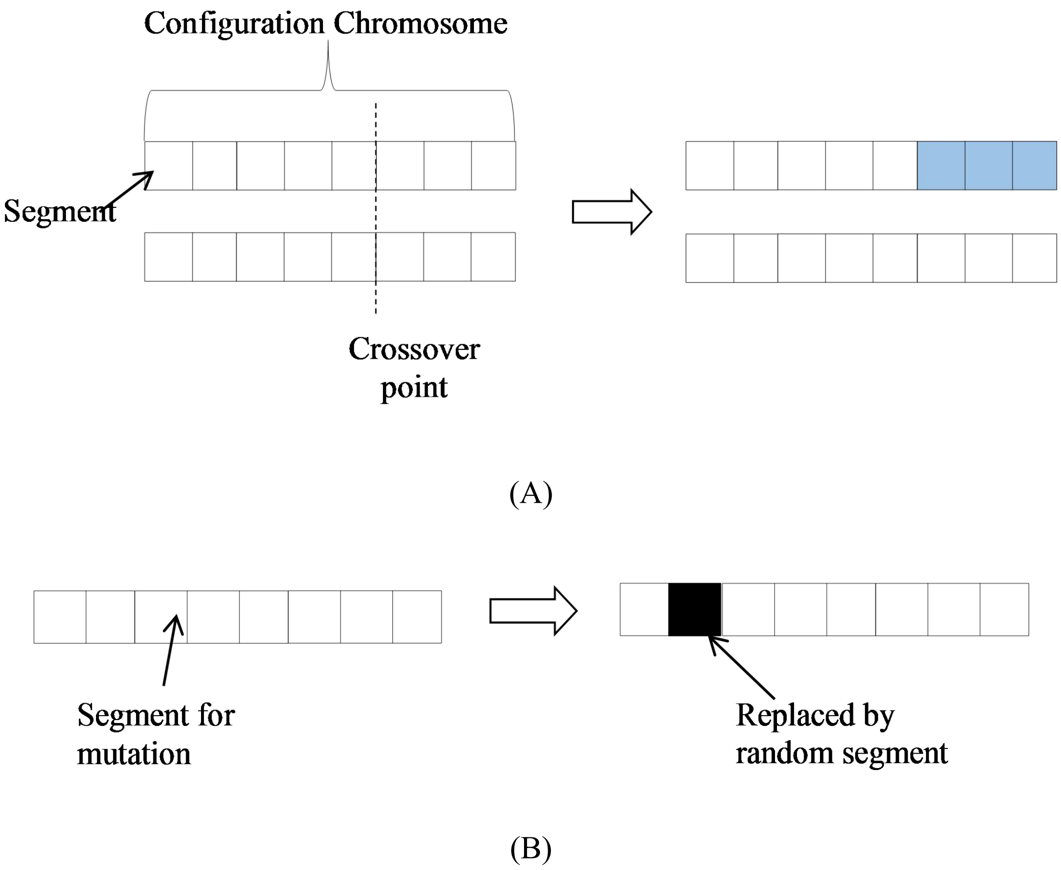


Figure 5.4. Crossover operation (A) and mutation operation (B).

Design of Operators

The roulette wheel method is used as a selection operator to pick individuals. The configuration with good fitness is chosen for crossover based on a random crossover point, as shown in Figure 5.4A. Then the new configurations are created. The mutation operation is applied to a string from which random segments are selected to be mutated and replaced by other random segments, as shown in Figure 5.4B. The ratios of the crossover and mutation operations to the total population are represented by r_a and r_c, respectively.

5.4 Case Study and Discussion

In this section, a three degree-of-freedom (DOF) manipulator will be used to illustrate the configuration synthesis for an under-actuated robot. The task specification is that the manipulator moves through five points, as shown in Table 5.1. The velocity and acceleration are zero at each point.

Table 5.1. Task specification.

Point	Position(mm)
1	(465, -65, 140)
2	(340, 220, -40)
3	(100, 200, -135)
4	(140, 65, 140)
5	(60, 120, -130)

The user requires a 3-DOF robot to achieve the task so that there are three joints that could be passive or active. Therefore, in the model of configuration synthesis, the total design variables include: i) three variables for types of joints; ii) three variables for the location of the base; iii) three variables for length of links; and iv) 3 – (m+3) variables for assembly pattern. m is determined by the length of links. For example, if m can be formed by two link modules, m is equal to 2. It is worth to note that the number of modules could be different in different configurations. For example, one single link module is enough when the length of a link is the same as that of a single link module.

For implementation, the parameters of the genetic algorithm are set as follows. The initial population of the individuals is 100, and the population number in each generation is also 100. The termination condition for the GA program is 100 generations.

The evolution process of the solution is shown in Figure 5.5. After thirty generations, the solution converged. The evolution process of the design variables is shown in Figure 5.6 and it can be seen that the design variables also converged after thirty generations.

The optimal configuration is shown in Figure 5.7. In order to illustrate the configuration synthesis algorithm, two individuals with lower fitness compared to the optimal configuration, Configuration I and Configuration II, are shown at random. The two configurations are feasible to achieve the task, as shown in figures 5.8 and 5.9. Each of them consists of more link modules and consume more energy, as compared with the optimal configuration.

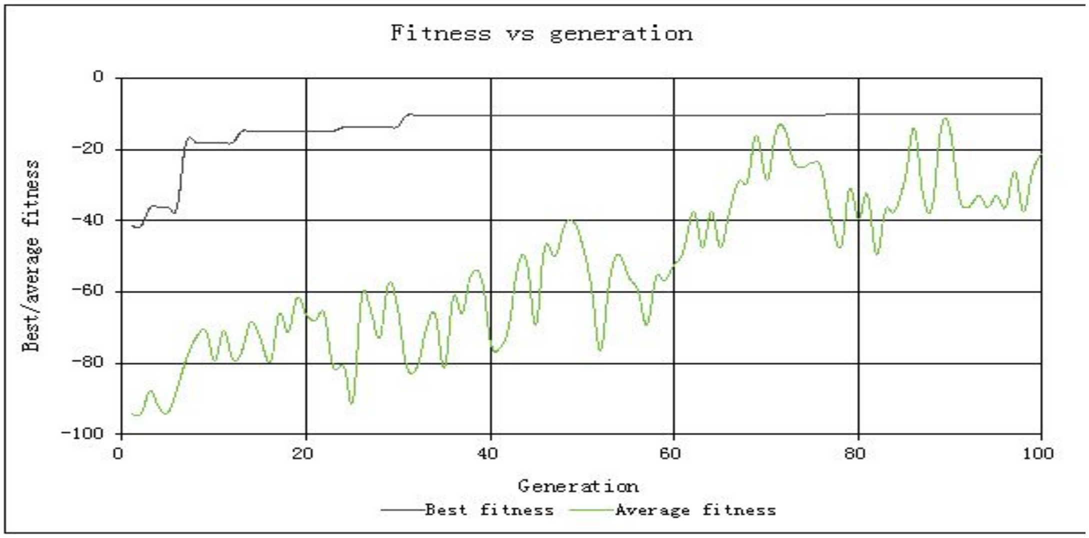


Figure 5.5. Best/average fitness

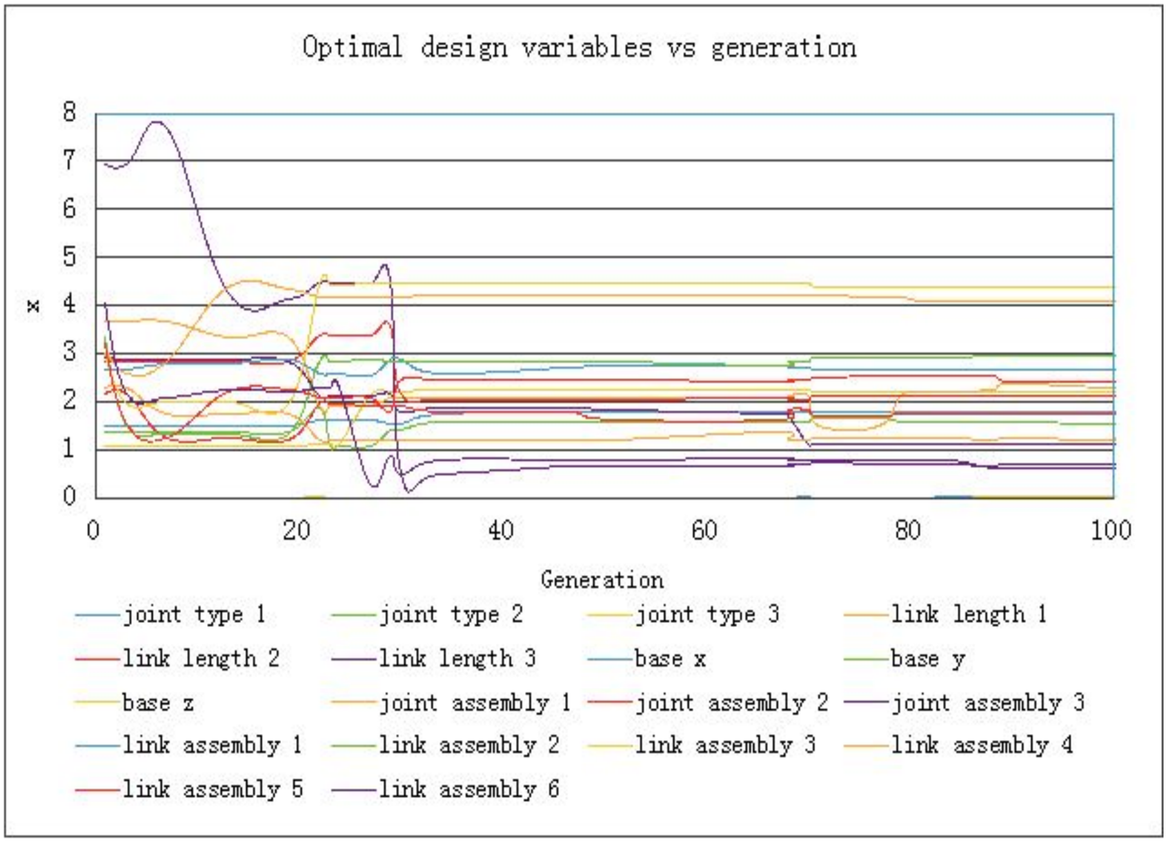


Figure 5.6. Optimal design variables.

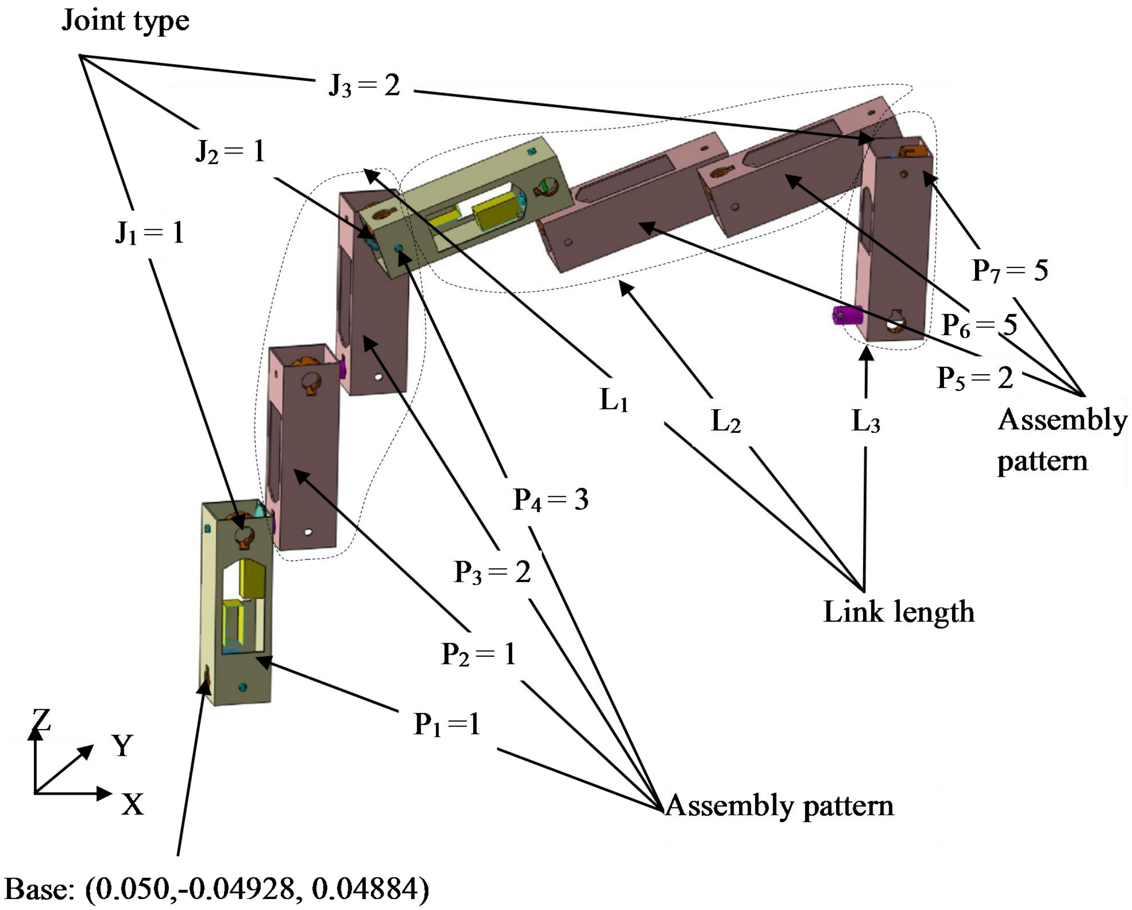


Figure 5.7. Optimal configuration consisting of two active joints, three links formed by five link modules, and one passive joint formed by two link modules (fitness= - 9.7911).

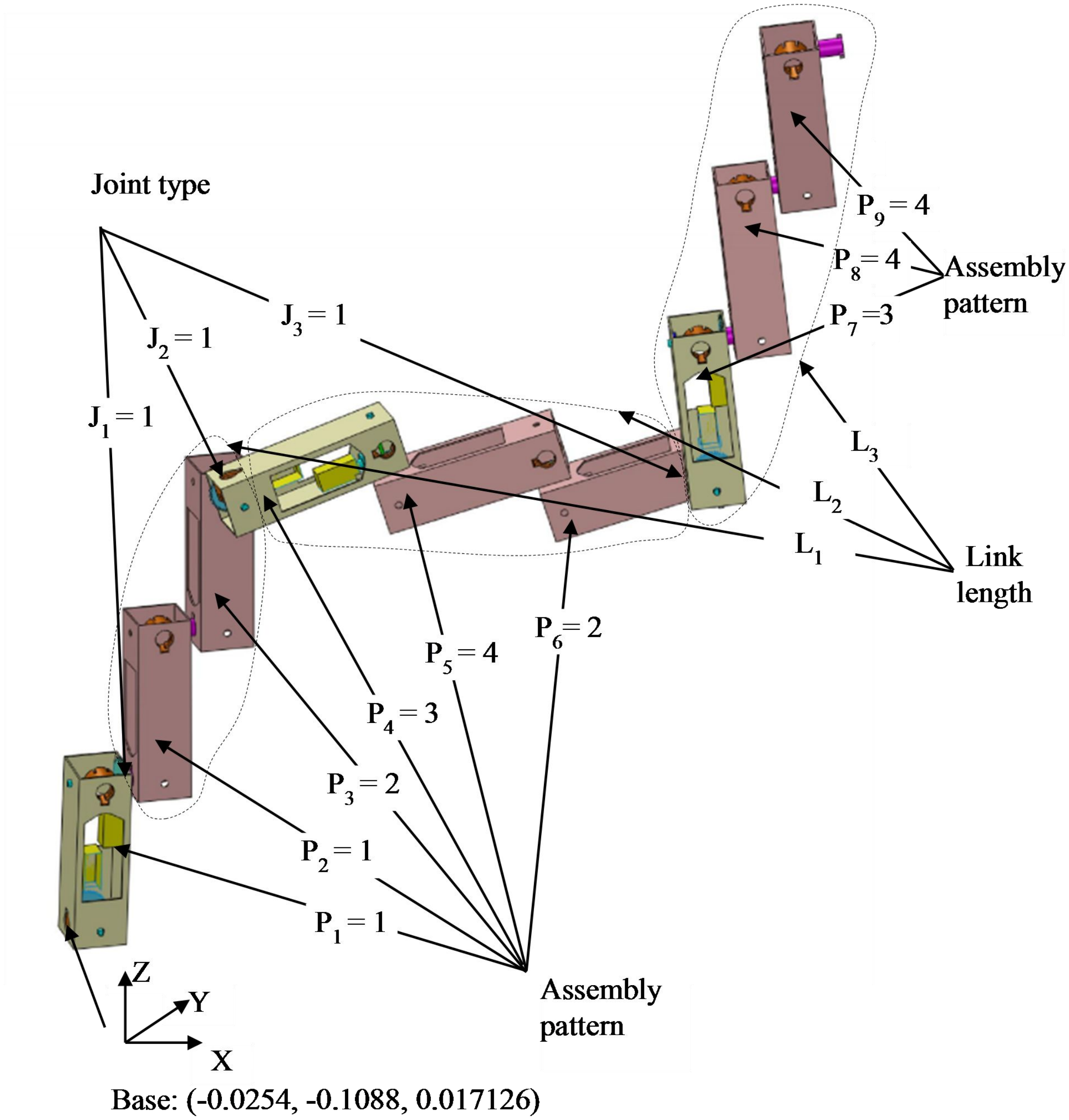


Figure 5.8. Feasible Configuration I consisting of three active joints and three links formed by six link modules (fitness= - 36.1318).

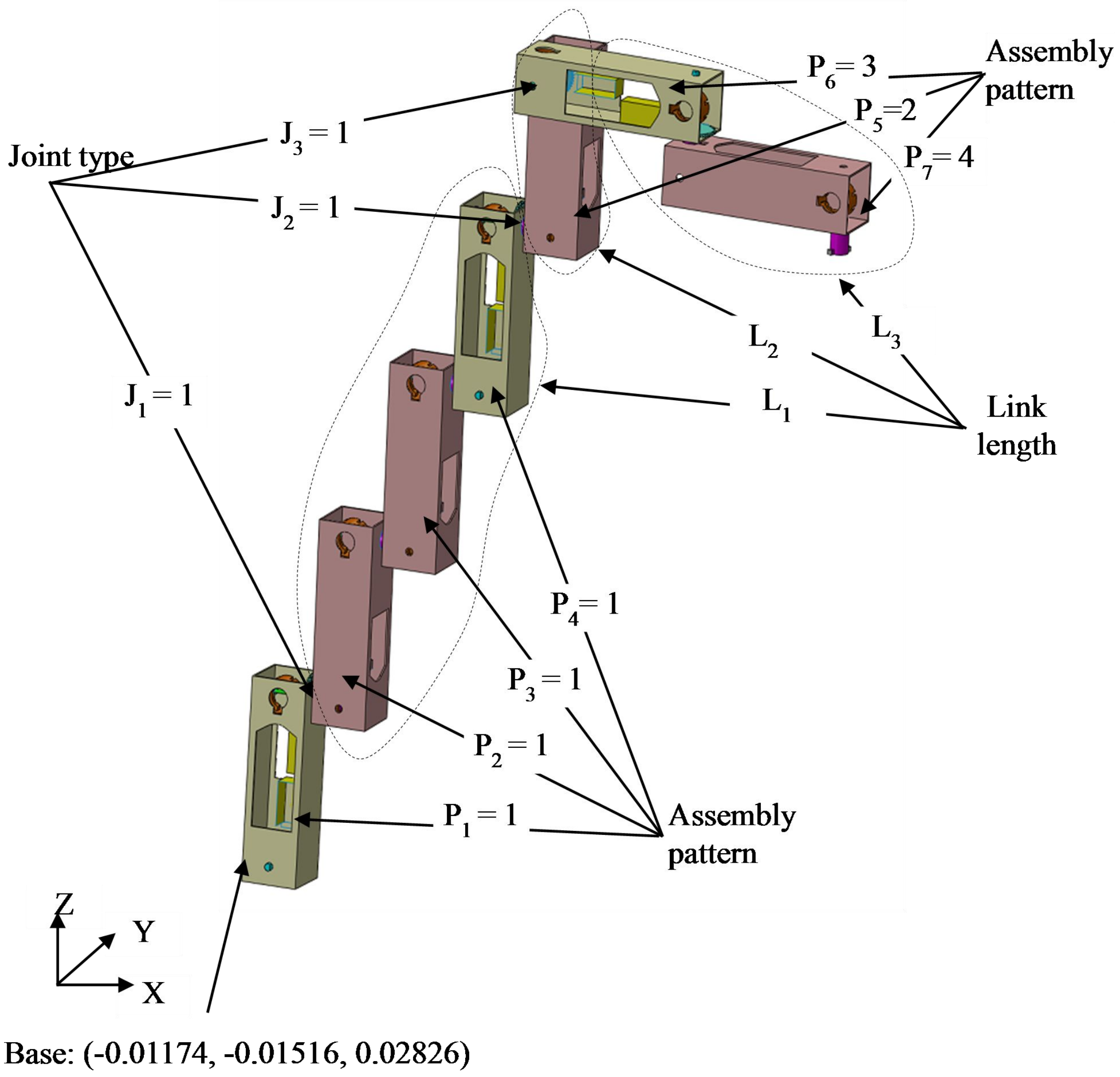


Figure 5.9. Feasible Configuration II consisting of three active joints and three links formed by six link modules (fitness= - 20.2094).

5.5 Summary

This chapter introduced an approach to synthesize the configuration of under-actuated robotic systems. This approach was validated by a 3-DOF manipulator in simulation (e.g., the GA search method). In this dissertation, the configuration synthesis was used to find the optimal configuration for strategy II and strategy III, respectively. For recovery strategy II (i.e., changing the configuration of a system by re-arranging its components), the approach could be applied to synthesize the optimal goal configuration a failed robot may possibly reach. Thus, the goal configuration can complete the original task with less energy and higher manipulability. For recovery strategy III (i.e., changing the length of a link), the model could be applied to find an optimal goal configuration whose links are different from that of the original robot. After the damaged robot reconfigures into a new configuration with different links, the robot can continue the task with less energy consumption and higher manipulability. It can be concluded that the configuration that includes passive joints is a good way to reduce the cost.

CHAPTER 6
RECONFIGURATION PLANNING AND SCHEDULING

Configuration synthesis is to find the best goal configuration at which the robot can still fulfill its original function. A series of changes toward a goal configuration on a damaged robot is needed instead of a one-step change. This series of changes is called a change plan or plan for brevity. Planning is simply a process to generate a list of actions in a sequential order (i.e., the move sequences of modules). Scheduling refers to a series of changes with time and effort concerns (force and torque out of the active joint) considered on the top of the plan. The present chapter presents a methodology for planning and scheduling of an under-actuated robot. It is worth to note that in this chapter the goal configuration is assumed to be known.

The chapter is organized into five sections. Section 6.1 defines the reconfiguration rules of the under-actuated robot. Section 6.2 proposes the computational model of reconfiguration planning and scheduling. In Section 6.3, a computational model is developed for the conceptual model proposed in the last section. In Section 6.4, a design case is given to demonstrate how the proposed models work. Section 6.5 provides a conclusion.

6.1 Reconfiguration Rules

Most of the existing studies on reconfiguration planning are limited to the architecture of robots, which have all active modules. This kind of architecture is not applicable to under-actuated robots. In under-actuated robots, the passive joints have to be manipulated by their neighboring active joints or active sub-systems or chains. Therefore, the reconfiguration rules for the under-actuated robot is this: when two parts are to be connected, each part should have a feasible sub-system. A feasible sub-system refers to an assembly of parts in which a passive joint's neighboring joint is not a passive joint. As shown in Figure 6.1B, the passive joint, Part B, is actuated by Sub-system 1. In this case, Sub-system 1 is a feasible sub-system.

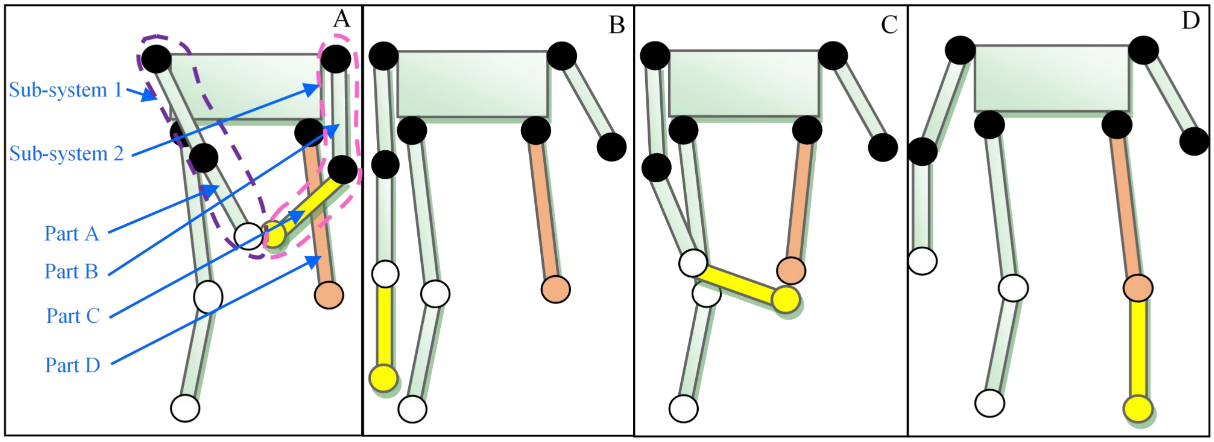


Figure 6.1. An example of the reconfiguration enhancement: Sub-system 1 (left arm) picks up Part C and assembles Part C with Part D. (A) Sub-system 1 and Sub-system 2 move to each other to connect Part A and Part C; (B) Part C disconnects from Part B; (C) Part C connects with Part D; (D) Part C disconnects from Part A. Note that black and white circles represent active and passive joints, respectively.

However, two sub-systems may not be connected when neither is feasible to move to the other due to geometry or passiveness of the joint. For example, as shown in Figure 6.1A, it is not feasible for Sub-system 2 to assemble Part B with Part C, due to geometry. In this case, a third feasible sub-system needs to be used so that the sub-system can pick up one part and place this part in the desired location of another sub-system. This can be viewed as an enhancement of the reconfigurability. Figure 6.1 shows a reconfigurability enhancement: a third sub-system (i.e., Sub-system 1) is used to pick up Part B and assemble it with Part C. Besides using a third sub-system to enhance the reconfigurability, another enhancement is to change the passive joint into a link (i.e., switch the passive joint into a fixed connection). This enhancement is mainly based on the passive joint design presented in Chapter 4. It is worth mentioning that the reconfiguration rules and reconfiguration enhancement will be used to guide the reconfiguration planning, which will be discussed later.

6.2 Algorithm for Reconfiguration Planning and Scheduling

The goal of reconfiguration planning is to achieve a minimum number of changes or moves or reconfigurations. The goal of scheduling is to achieve minimum effort and the shortest time based upon a reconfiguration plan. Though planning and scheduling may involve couplings, this dissertation did not consider any coupling of planning and scheduling.

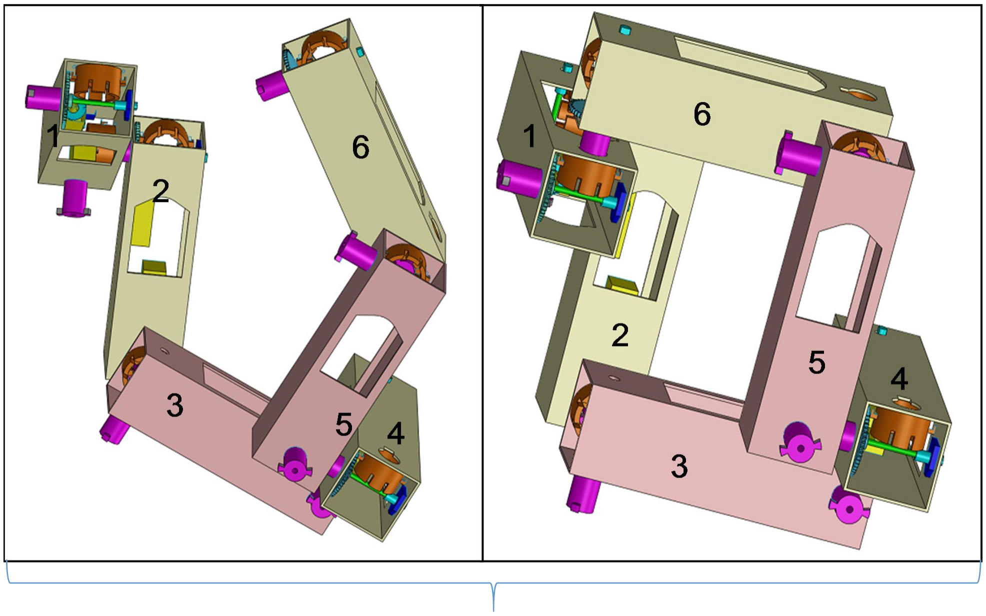
Naturally, an optimization model was employed for reconfiguration planning and reconfiguration scheduling. An optimization model consists of variables, constraints, and objectives, which are presented in detail in the following sections.

6.2.1 Variables

Reconfiguration planning and reconfiguration scheduling can be viewed as a series of connection/disconnection operations and actuation operations, respectively.

Connection operation is the connection between two modules with information of types of connections and types of modules. A 1×6 matrix is used to represent one connection operation. Particularly, the i^{th} connection operation is represented as g_i=[c, ID_1, ID_2, ID_3, ID_4, q], where ID_1 denotes the modules with its name of ID_1; ID_2 denotes the port (of the module ID_1) with its name of ID_2; ID_3 denotes the modules with its name of ID_3; and ID_4 denotes the port (of the modules ID_3) with its name of ID_4. Note that for the system which was taken as an example, each module has two male ports and two female ports, denoted as F_1, F_2, M_1, and M_2, respectively. Further, c denotes the connective status with 0 for connection and 1 for disconnection. If the two modules are to connect, q denotes the type of connection they form with 1 for passive joint and 2 for rigid link and 0 for no connection. For the system as shown in Figure 6.2, the code of the connection operation has the following meanings: the first "1" means that Port 2 (marked as "2") of Module 1 (marked as the second "1") will be connected with Port 3 (marked as "3") of Module 14 (marked as "14"). The last "1" means that the two modules will form a passive joint.

Actuation operation describes the actions of active joints for the connection/disconnection operations generated in the planning phase. To achieve the connection operation, each of the two modules will be actuated by the active modules on their own branch; then, the two branches form a loop. For the disconnection operation, the two modules and active modules are in a loop. A 3×m matrix is used to represent one actuation operation between two modules. Particularly, m is the number of active modules in the two branches when the two modules are to be connected, and m is the number of active joints included in the loop when the two modules are to be disconnected. In the matrix, each column represents the ID of each active module on the two branches or a loop, and the first, second, and third rows represent the actuations of the corresponding active joints, specified by angles. The rotation angles in Row 1 are multiples of 20º between -180º and 180º, which is the motion of the active joints before the two ports contact. The rotation angles in row 2 and row 3 are multiples of 3º between -20º and 20º. The second row represents the motion of the active joints when inserting the male interface to a layer (i.e., Layer 2 or Layer 5) of the female interface. Further, the third row represents the motion of the active joints when rotating in the corresponding layer of the male interface. For the system in Figure 6.2, the actuation operation respectively denotes the motions of the joint modules 1, 2, 4, and 6 so that the connection operation (i.e., Port 2 of module 1 connects with Port 3 of module 14 to form a passive joint) can be realized.



Planning sequence

1	1	2	6	3	1

Figure 6.2. A sample of connection operation: Port 2 of Module 1 connects with Port 3 of Module 14 to form a passive joint.

$$\begin{array}{cc} & \begin{array}{cccc} <Joint, 1 & Joint, 2 & Joint, 4 & Joint, 6> \end{array} \\ M_a = & \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \end{bmatrix} \end{array}$$

Figure 6.3. A sample of actuation operations for the connection operation in Figure 6.2.

The code of the actuation operation is shown in Figure 6.3 with the following meanings. The code in the first row, [$\alpha_1, \alpha_2, \alpha_3, \alpha_4$] respectively denotes the motions of joint modules 1, 2, 4, and 6 sequentially to move Port 3 of Module 6 close to Port *2* of Module 1 contacted. The code in the second row, [$\beta_1, \beta_2, \beta_3, \beta_4$] respectively denotes the sequential motions of joint modules 1, 2, 4, and 6 to insert Port 3 of Module 6 to Layer 2 of Port 2 of Module 1. The code in the third row, [$\gamma_1, \gamma_2, \gamma_3, \gamma_4$] respectively denotes the motions of joint modules 1, 2, 4, and 6 to rotate Port 3 of Module 6 in Layer 2 of Port 2 of Module 1.

6.2.2 Constraints

In the planning phase, each individual should match the goal configuration, i.e., no error should exist between this configuration and the goal configuration.

(6.1)

C_g is the goal configuration and C_j is the configuration generated in jth generation. Individuals that do not meet the reconfiguration rules and the reconfiguration enhancement will be rejected.

In the scheduling phase, the corresponding ports of two modules should be connected, i.e., no error should exist between two points of the connected ports:

(6.2)

where, (x_{k1}, y_{k1}) and (x_{k2}, y_{k2}) are the coordinate systems of the two points on the two interfaces that will be connected during the k^{th} reconfiguration (i.e., the k^{th} gene), as shown in Figure 6.4.

In the scheduling phase, the motion of the joints should satisfy the kinematic and dynamic requirements. The kinematic relationship among θ_a and θ_p and x is expressed using the Jacobian matrix, as follows:

(6.3)

(6.4)

where x is the end-effector coordinate, $x=[x_1, x_2 \ldots x_m]^T$. It is the given task points. J_a and J_p are the Jacobian matrix for active joints and passive joints.

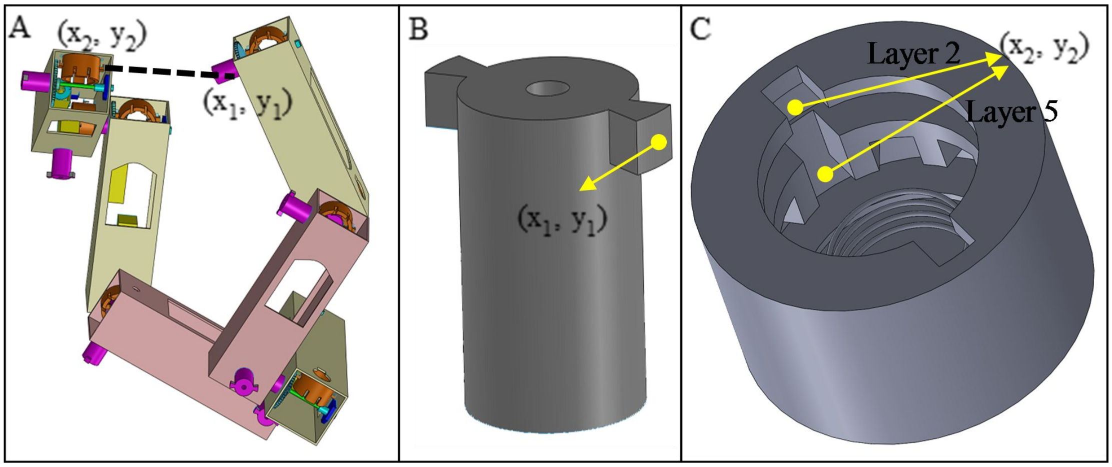


Figure 6.4. The distance between two interfaces of two modules. (A) The distance between two points on the male and female interfaces. (B) The coordinate system of the male interface, point (x_1, y_1) is located at the geometric center of either of the two concave surfaces. (C) The coordinate system of the female interface, point (x_2, y_2) is located at either of the two corresponding positions in the inserting area at different layers (Layer 2 for passive joint connection, Layer 5 for fixed connection).

The dynamic equation of motion is generally expressed as:

(6.5)

where θ_a is the generalized coordinate for active joints and θ_p represents passive joints. H_{aa}, H_{ap}, and H_{pp} are inertia matrices. b_a and b_p are Coriolis and centrifugal and gravity forces on the active and passive joints, respectively. τ_a is the vector of torques applied at the active joints.

Different configurations have different kinematic parameters and dynamic parameters. The method of the automatic generation of D-H kinematic parameters and dynamic parameters was studied, which can be referred to the previous work [Bi and Zhang 2001, Zhang et al. 2014b].

6.2.3 Objectives

The main objective is to find a set of solutions that meet all design constraints. To improve the planning solutions, the reconfiguration steps, N, will be employed as the objective function. Planning solutions with fewer steps will be considered in the scheduling phase. In the scheduling phase, an index energy consumption is introduced, which is similar to that proposed by Paredis [1994]. The index of energy consumption is calculated as follows:

(6.6)

where e is the energy consumption, Tra is the workspace that consists of all working points along the trajectory, m is the number of the active joints, and $T_i(\theta)$ is the torque of the i^{th} actuator.

6.3 Implementation by Genetic Algorithm (GA)

A genetic algorithm (GA) was employed for reconfiguration planning and scheduling. The procedure is detailed in a flowchart in Figure 6.5. The main strategy in the reconfiguration process is as follows: (i) in the planning stage, the solutions that do not satisfy the reconfiguration rules and the reconfiguration enhancement will be rejected; (ii) M_2 best solutions will be used in the scheduling phase; and (iii) the optimal scheduling solution is found until all these planning solutions are examined.

Design of Fitness Function

A variable-length genome is used to specify the planning sequence and scheduling sequence, which are encoded by the design variables, i.e., the connection and actuation operations. In the reconfiguration process, the planning steps (N) and the energy consumption are used as the evaluation functions for reconfiguration planning and reconfiguration scheduling, respectively. Since the GA evaluates the solution that achieves the maximum function as the best solution, the fitness function in planning phase is 1/N, and a minus sign is placed before the objective function. For a candidate solution that does not satisfy the constraints, its fitness will be given a larger negative value in order to reject the candidate.

Design of Genetic Operators

For selection, crossover, and mutation operation, the new generation group uses the following operators to proceed the evolution.

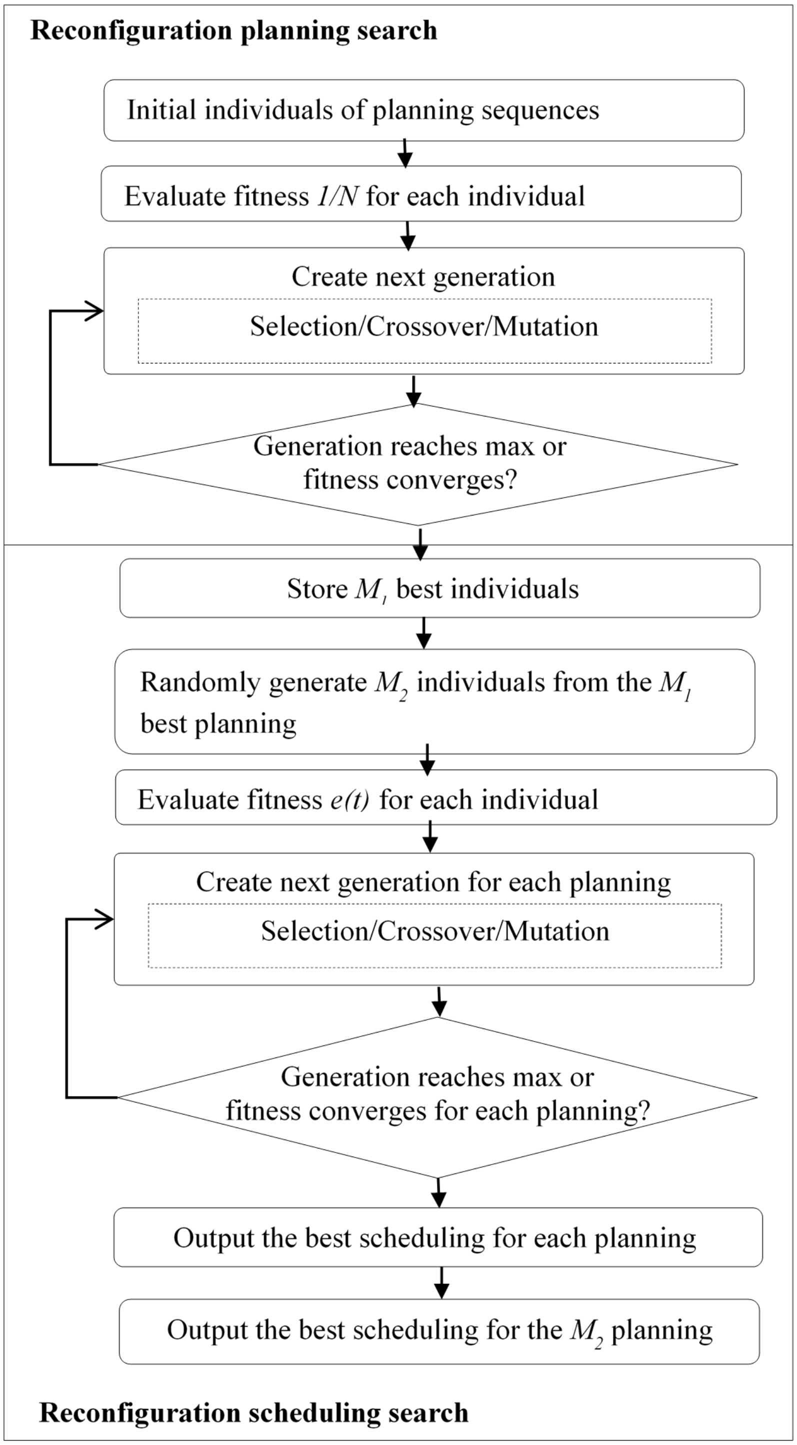


Figure 6.5. The flowchart of the method of reconfiguration planning and reconfiguration scheduling.

The roulette wheel method is used as the selection operator to pick individuals. The crossover operation for planning is preceded among different variable-length genomes. A "cut and splice" crossover operator is used (Figure 6.6A). The cut operator breaks a genome into two parts with a cut probability, p_{c1}. The cut position is chosen randomly. The splice operator joins two genomes with a splice probability, p_{c2}.

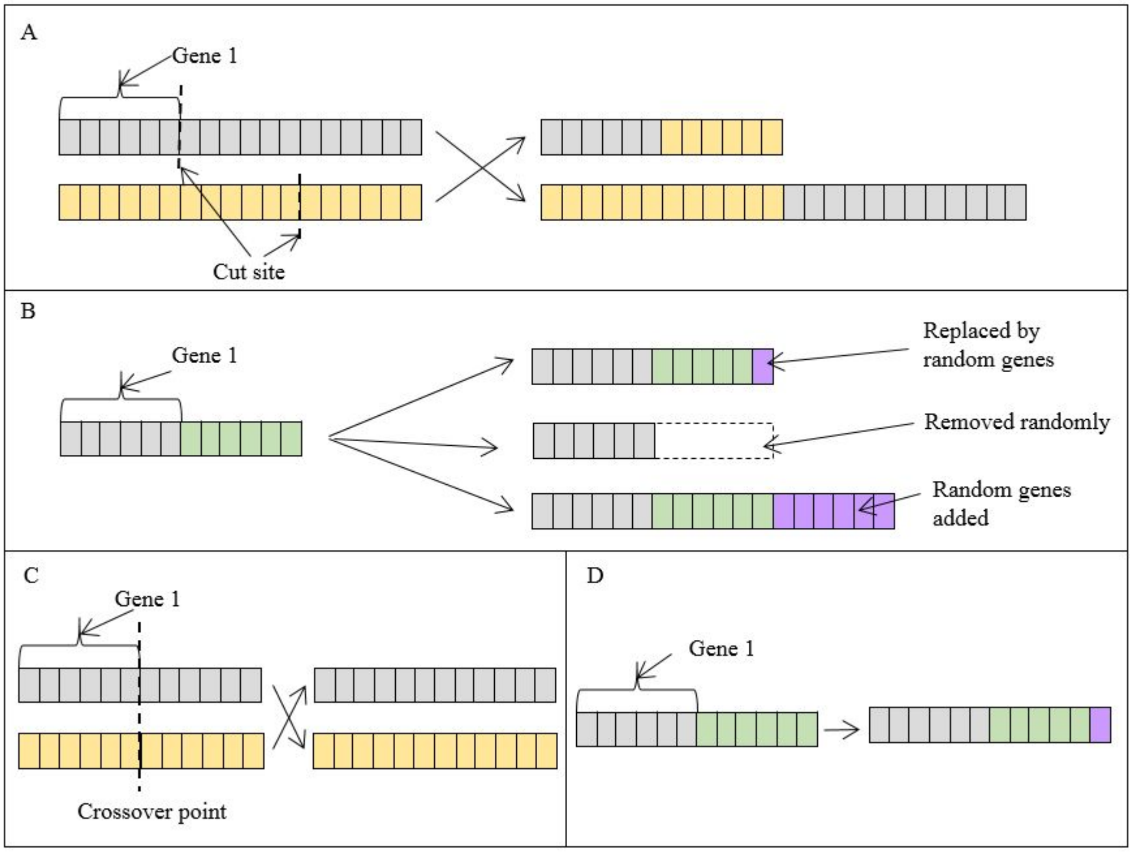


Figure 6.6. Crossover operators and mutation operators. (A) Cut and splice crossover operator for reconfiguration planning. (B) Mutation operators for reconfiguration planning: replacement, addition, and random. (C) One point crossover operator for reconfiguration scheduling. (D) Mutation operator for reconfiguration scheduling.

There are three types of mutation in planning: change, addition, and removal of genes (Figure 6.6B). Their probabilities are denoted as p_{m1}, p_{m2}, and p_{m3}, respectively. The crossover operation for scheduling is shown in Figure 6.6C. The crossover operation is a one cross-point crossover with the probability denoted as p_{c3}. Genome length genome is the same for each planning solution. There is only replacement of random genes in mutation, which is p_{m4}, (Figure 6.6D).

6.4 Case Study

The robot is required to change from the initial damaged configuration (Figure 6.7B) to the goal configuration (Figure 6.7F). In the damaged configuration, the robot consists of six active joint modules (light-colored) and six link modules (dark-colored). The passive joints exist in Module 11 and Module 12, and Module 1 and Module 6 (Figure 6.7F). The fixed connections exist in Module 3 and Module 7, Module 7 and Module 8, Module 8 and Module 10, and Module 9 and Module 10.

The algorithms were implemented in Matlab. The Matlab Robotic Toolbox [Corke 2013] and the GA Toolbox [Chipperfield et al. 1994] were used. The experiment was conducted on a PC running Windows Vista with a 2.40 GHz CPU and 3.00 GB of RAM. In the simulation, the parameters of the genetic algorithm are empirically set as follows.

- The initial population of individuals for planning is 100.
- Crossover ratio of planning: cut (r_{c1} =0.9); splice (r_{c2} =0.9).
- Mutation ratio of planning: replacement (r_{m1}=0.2), addition (r_{m2}=0.5), and removal

(r_{m3}=0.5).

- Thirty individuals stored from planning solutions.
- Crossover ratio of scheduling (r_{c3} =0.9).
- Mutation ratio of scheduling: replacement (r_{m4}=0.5).
- The termination condition for the GA program is 100 generations.

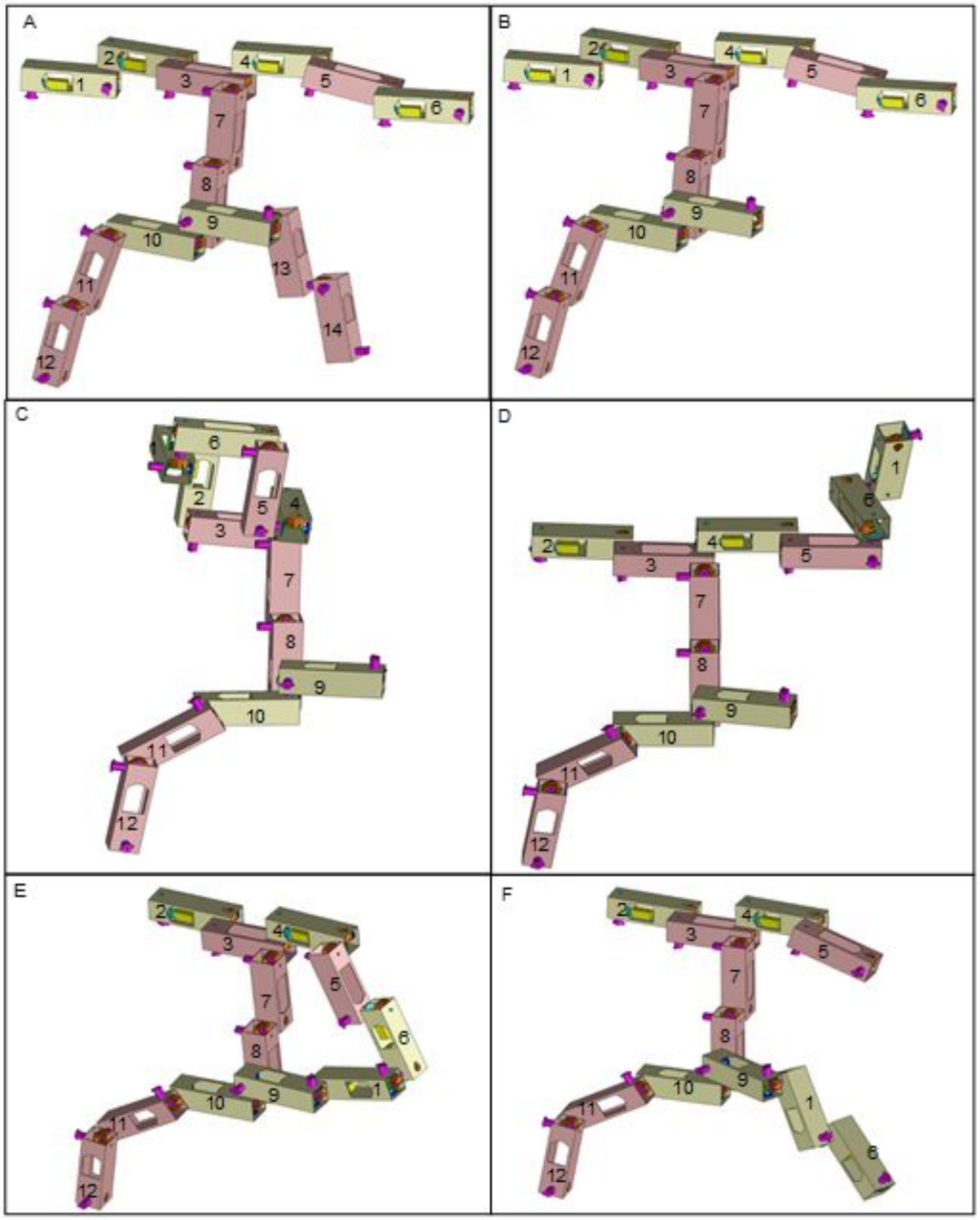


Figure 6.7. Self-reconfiguration process. (A) The robot state before failure. (B) The state of the damaged robot: Module 13 and Module 14 dropped out. (C) Module 1 connects with Module 6. (D) Module 2 disconnects from Module 1. (E) Module 1 connects with Module 9. (F) Module 6 disconnects from Module 5.

Figure 6.8 demonstrates the average and maximum value of fitness function with respect to each generation. It shows that the solution converged after twenty generations. The optimal solution that consumes minimum energy is the reconfiguration process shown in Figure 6.7. The process is: Module 6 picks up Module 1 and assembles it with Module 9.

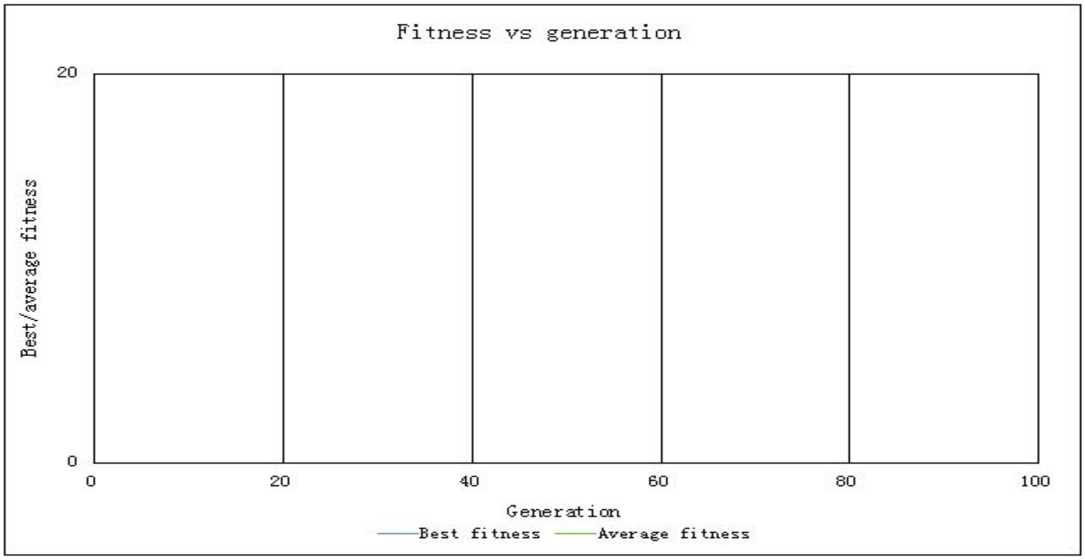


Figure 6.8. Best/average fitness.

In order to demonstrate physical feasibility, a physical robot capable of implementing the above reconfiguration process was built, as shown in Figure 6.9. The video can be found at https://www.youtube.com/watch?v=I_QRMryR_xQ.

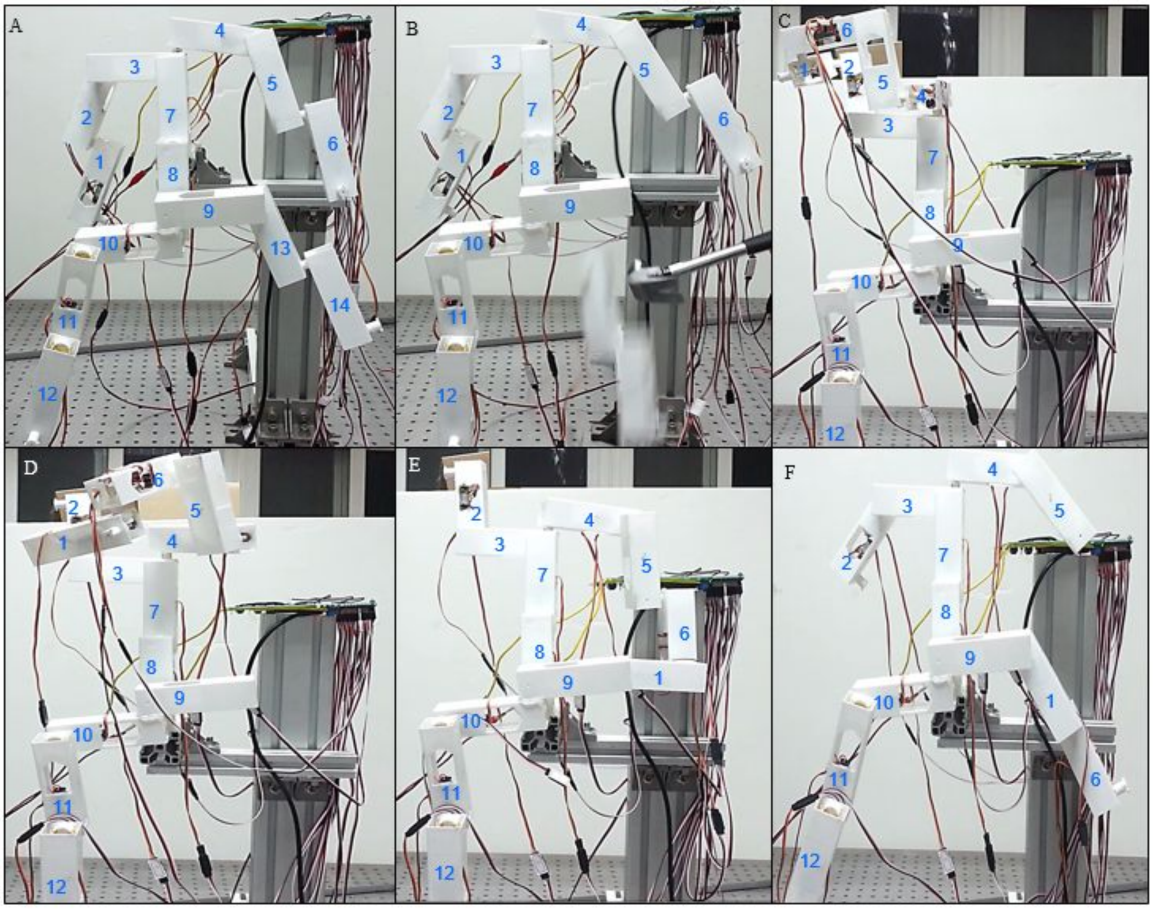


Figure 6.9. Snapshots for a self-reconfiguration trial with the same reconfiguration process as in Figure 6.7: Module 6 picks up Module 1 and assembles it with Module 9.

6.5 Summary and Discussion

This chapter introduced an algorithmic process that solves the problem to transform a configuration into another configuration to achieve a specific functional goal. Our algorithm searched the optimal solution through two phases, i.e., reconfiguration planning and reconfiguration scheduling. The first phase rejects the solutions that do not satisfy the reconfiguration rules, which narrows the search range when considering hardware compatibility of under-actuated, self-reconfigurable robots. The second phase tests the feasibility of planning and generates the optimal scheduling solution. The two-phase approach, as well as the natural way of genome encoding, greatly reduced the search time. The algorithm was validated in a simulation (e.g., the GA search method) and it was illustrated with an experiment.

CHAPTER 7
CONCLUSIONS AND FUTURE WORK

7.1 Overview and Summary

The motivation of this dissertation was based on the understanding that resilience is one of the most important features for modern intelligent robots used in situations where repair and replacement are not possible. Yet a systematic approach in both theory and methodology to achieve adequate resilience of robots is lacking. In particular, this dissertation took a general approach of bringing passive joints into robots to lead to an under-actuated resilient robot.

This dissertation addressed the following issues: (i) general architecture of resilient robot systems, which covers all the existing recovery strategies, (ii) the design of a novel docking system, (iii) configuration synthesis, and (iv) reconfiguration planning and scheduling.

A literature review and analysis of existing resilient robots and self-reconfigurable robots was conducted to show the significance of addressing the above issues. This has led to the development of a general architecture of resilient robots, in which the docking system design is noted as an important issue. The docking system can form passive joints and adjustable links, which achieve connection/disconnection and locking/unlocking. The general architecture is a foundation for the resilient robot configurations. Configuration synthesis for resilient robots was modeled as an optimization problem. The optimization problem consists of both discrete variables and continuous variables. The Genetic Algorithm (GA) was applied to solve the optimization problem. The optimal configuration obtained was used as the goal configuration into which the robot will change. Finally, the reconfiguration approach was presented to find a sequence of moves of active joints from a damaged state to a goal state through a reconfiguration planning phase and a reconfiguration scheduling phase.

7.2 Conclusions

In general, the research documented in this dissertation has demonstrated that the research objectives set out in Chapter 1 can be achieved. The following conclusions can be drawn:

(i) Regarding Objective 1, a general architecture of a resilient robot with passive joints and adjustable links was given. This architecture has the most extensive coverage of the recovery strategies that have not been fully addressed by the published architectures. The significance of the general architecture is: (1) it provides a benchmark for evaluating the resilience of robotic systems, and (2) the inclusion of passive joints and adjustable links not only reduces the cost but also provides an efficient approach to deal with the following failures: an active joint is unlocked (thus becoming a passive joint), and/or an active joint is locked (thus becoming an adjustable link). As well, a robot developed with the proposed architecture is an under-actuated resilient robot, which is the most general robot in terms of the architecture (because of the inclusion of passive joints and links, active joints, and adjustable links).

(ii) Regarding Objective 2, a docking system was developed based on the resilient robot architecture. The introduction of the docking system results in there being only two types of components, i.e., a link module and active joint module for the robot, which are the fundamental components. This further leads to the highest degree of resilience for the robot as opposed to the existing architecture of robots, as each module is not disassembled and does not affect others when it gets damaged. Passive joints and adjustable links can be formed by two modules when the docking is located at different positions. Docking and undocking actions do not have independent actuation, and they are actuated by the active joints. Therefore, the energy consumption for locomotion and reconfiguration is greatly reduced when compared with that of existing self-reconfigurable robots.

(iii) Regarding Objective 3, an approach to synthesize the configuration of under-actuated resilient robot was introduced. It has been observed that the architecture that adds passive joints and adjustable links can greatly reduce the energy consumption and further improve the robot's performance.

(iv) Regarding Objective 4, the reconfiguration process was divided into two phases: reconfiguration planning and reconfiguration scheduling. The first phase narrows the search range with consideration of hardware compatibility of under-actuated, self-reconfigurable robots. The second phase tests the feasibility of planning and generates the optimal scheduling solution.

7.3 Limitations and Future Work

The work presented in this dissertation has some limitations. First, the dynamic control of reconfiguration planning and scheduling for the resilient robot has not been addressed. This issue includes incorporation for control behavior into configuration synthesis and development of control approaches to eliminate position errors during the reconfiguration process. Second, the implementation of the optimization model for configuration synthesis and reconfiguration process lacks generality. Future studies are needed to overcome these shortcomings and to extend the present work as follows.

(1) Distributed control algorithms could be employed for reconfiguration planning and reconfiguration scheduling so that complex adaptive and dynamic shapes can be created. As well, the design and control should be dealt with simultaneously in the configuration synthesis. Other design issues, such as force/moment balancing and vibration control, need to be considered along with real-time control, a so-called Design For Control methodology [Zhang et al. 1999, Li et al. 2001].

(2) Research on the failure identification of under-actuated robots is a promising work. The failures could be caused by the external environment and/or the internal environment. Accompanying the feedback from sensors, a robot is then able to infer the faults based on the dynamics of an under-actuated robot. The roots of the failure and the state after failure can then be identified.

(3) It is significant to develop knowledge and technology for the measurement of resilience, and the relationship with other system properties such as reliability and cost needs to be further studied. This topic is important when a resilient robot is to be tailored for a particular application. Resilience may conflict with other system properties and may incur a high cost.

(4) Research on the so-called soft resilient robot is promising when smart materials with plasticity properties are included. In the soft resilient robot, the concept of the active

joint, passive joint, passive link, and adjustable link proposed in this dissertation can be borrowed. Thus, the recovery process could be based on the three recovery strategies in Chapter 3. It is worth mentioning that the adjustable link could be an active link, which may be contracted or expanded to change its shape, not just its length.

REFERENCES

1. Ammonius, H., Cohen, S. M., Matthews, G. B., 1991. “On Aristotle’s Categories”, Ithaca, NY: Cornell University Press.

2. Arai, H., 1996. “Controllability of a 3-DOF manipulator with a passive joint under a nonholonomic constraint”, *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 3707-3713, Minneapolis, Minnesota, USA.

3. Avižienis, A., Laprie, J.-C., Randell, B., and Landwehr, C., 2004. “Basic Concepts and Taxonomy of Dependable and Secure Computing”, *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, No. 1, pp.11-33.

4. Bergerman, M., and Xu, Y., 1996. “Robust joint and Cartesian control of under-actuated manipulator systems”, *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 118, No. 3, pp. 557-565.

5. Berenson, D., Estevez, N., and Lipson, H., 2005. “Hardware evolution of analog circuits for in-situ robotic fault-recovery”, *IEEE Proceedings of 2005 NASA/DoD Conference on Evolvable Hardware*, pp. 12-19.

6. Bhamra, R., Dani, S., and Burnard, K., 2011. “Resilience: the concept, a literature review and future directions”, *International Journal of Production Research*, Vol. 49, No. 18, pp. 5375–5393.

7. Bi, Z.M., and Zhang, W.J. (Supervisor), 2002. “On adaptive robot systems for manufacturing applications”, Ph.D dissertation, Department of Mechanical Engineering,

University of Saskatchewan, Saskatoon, Canada.

8. Bi, Z.M., and Zhang, W.J., 2001. "Concurrent Optimal Design of Modular Robotic Configuration", *Journal of Robotic Systems*, Vol. 18, No. 2, pp. 77-87.

9. Bischoff, R. and Graefe, V., 2003. "HERMES—an Intelligent Humanoid Robot Designed and Tested for Dependability", *Experimental Robotics VIII*. Springer Berlin Heidelberg, pp. 64-74.

10. Bishop, J., Burden, S., Klavins, E., Kreisberg, R., Malone, W., Napp, N., and Nguyen, T., 2005. "Self-Organizing Programmable Parts", *Proceedings of the 2005 IEEE International Conference on Intelligent Robots and Systems*, pp. 3684-3691, Edmonton, Alberta, Canada.

11. Bongard, J., Zykov, V., and Lipson, H., 2006. "Resilient machines through continuous self-modeling", *Science*, Vol. 314, pp. 1118-1121.

12. Brown, E., Rodenberg, N., Amend, J., Mozeika, A., Steltz, E., Zakin, M., Lipson, H., and Jaeger, H., 2011. "Universal robotic gripper based on the jamming of granular material“, *Proceedings of the National Academy of Sciences*, Vol. 107, No. 44, pp. 18809-18814.

13. Bursztein, E., and Goubault-Larrecq, J., 2007. "A logical framework for evaluating network resilience against fault and attacks", Lecture notes in computer science, Vol. 4846, pp. 212-227.

14. Butler, Z., Kotay, K., Rus, D., and Tomita, K., 2001. "Cellular automata for decentralized control of self-reconfigurable robots, *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pp. 721-726, Seoul, South Korea.

15. Butler, Z., and Rus, D., 2003. "Distributed planning and control for modular robots with

unit-compressible modules", *The International Journal of Robotics Research*, Vol. 22, No. 9, pp. 699-716.

16. Casal, A., and Yim, M., 1999. "Self-Reconfiguration planning for a class of modular robots", *SPIE Sensor Fusion and Decentralized Control in Robotic Systems II*, pp. 246-257, Boston, Massachusetts, USA.

17. Castano, A., Shen, W.M. and Will, P., 2000. "CONRO: Towards Deployable Robots with Inter-Robots Metamorphic Capabilities", *Autonomous Robots*, Vol. 8, No. 3, pp. 309–324.

18. Chen, I-Ming, 1994. "Theory and Applications of Modular Reconfigurable Robotic Systems", Ph.D. Thesis, Division of Engineering and Applied Science, California Institute of Technology, Pasadena, California, USA.

19. Chen, I-Ming and Burdick, J. W., 1995. "Determining Task Optimal Modular Robot Assembly Configurations", *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, pp.132-137, Nagoya, Aichi, Japan.

20. Chen, I.-M., and Burdick, J. W., 1998. "Enumerating Non-Isomorphic Assembly Configurations of a Modular Robotic System", *International Journal of Robotics Research*, Vol. 17, No. 7, pp. 702-719.

21. Chipperfield, A., Fleming, P. J., Pohlheim, H., and Fonseca, C. M., 1994. "Genetic Algorithm Toolbox for use with Matlab", Technical Report No. 512, University of Sheffield, Sheffield, UK.

22. Christensen, D. J., 2006. "Evolution of shape-changing and self-repairing control for the ATRON self-reconfigurable robot", *Proceedings of the 2006 IEEE Int. Conference on*

Robotics and Automation, May 15-19, pp. 2539-2545.

23. Christensen, D. J., Larsen, J. C., and Stoy, K., 2014. “Fault-tolerant gait learning and morphology optimization of a polymorphic walking robot”, *Evolving Systems*, Vol. 5, No. 1, pp. 21-32.

24. Collins, S., Ruina, A., Tedrake, R., and Wisse, M., 2005. “Efficient bipedal robots based on passive-dynamic walkers”, *Science*, Vol. 307, No. 5712, pp. 1082-1085.

25. Cooke, H. P., (transl.), 1983. “Aristotle Categories”, Vol. 1, Loeb Classical Library, Harvard University Press, USA.

26. Corke, P. I., 1996. “A Robotics Toolbox for MATLAB”, *IEEE Robotics and Automation Magazine*, Vol. 3, No.1, pp. 24-32.

27. Coste-Maniè, E., and Simmons, R., 2000. “Architecture, the Backbone of Robotic Systems”, *Proceedings of the 2000 IEE International Conference on Robotics and Automation*, pp. 67-72, San Francisco, California, USA.

28. Crawley, E., 2007. “System Architecture”, Lecture Notes, Engineering Systems Division, MIT, Cambridge, Massachusetts, USA.

29. Dollar, A. M., and R. D. Howe, 2010. “The highly adaptive SDM hand: design and performance evaluation”, *The International Journal of Robotics Research*, Vol. 29, No. 5, pp. 585-597.

30. Dori, D., 1998. “Object-Process Methodology: A Holistic Systems Paradigm”, Springer, Berlin.

31. Erden, M. S., and Leblebicioğlu, K., 2008. "Free gait generation with reinforcement learning for a six-legged robot", *Robotics and Autonomous Systems*, Vol. 56, No. 3, pp. 199-212.

32. Fukuda T., and Nakagawa, S., 1987. "A dynamically reconfigurable robotic system (concept of a system and optimal configurations)", *Proceedings of the 1987 International Conference on Industrial Electronics, Control, and Instrumentation*, pp. 588-595, Cambridge, Massachusetts, USA.

33. Garcia, R., Stoy, K., 2008. "The Odin Modular Robot: Electronics and Communication", MSc Thesis, the Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense, Denmark.

34. Garcia, R., Hiller, J., and Lipson, H., 2010. "A vacuum-based bonding mechanism for modular robotics", *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp. 57-62, Anchorage, Alaska, USA.

35. Geekologie. "Self-Healing Robot Skin Grows Back Together When Cut", Retrieved June 25, 2014, from http://geekologie.com/2013/09/self-healing-robot-skin-grows-back-toget.php

36. Gilpin, K., Kotay, K., Rus, D., and Vasilescu, I., 2008. "Miche: modular shape formation by self-disassembly", *The International Journal of Robotics Research*, Vol. 27, pp. 345-372.

37. Gizmag. "ChemBot: the shape-shifting robot that is the stuff of nightmares", Retrieved November 12, 2011, from http://www.gizmag.com/chembot-robot-blob/13119/

38. Hawks, C., and Reed, C., 2006. “Theoretical steps towards modeling resilience in complex system”, Lecture notes in computer science, Vol. 3980, pp. 644-655.

39. He, Y., and Zhang, W.J. (Supervisor), 2008. “A FBS approach to emergency management”, MS Thesis, Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Saskatchewan, Canada.

40. Holling C. S., 1973. “Resilience and stability of ecological systems”, *Annual Review of Ecology and Systematics*, Vol. 4, pp. 1-23.

41. Hollnagel, E., Woods, D. D., and Leveson, N., 2006. “Resilience engineering concepts and precepts”, Aldershot, UK: Ashgate.

42. Hommes, H. van E., 2007. “System Architecture”, Lecture Notes, Engineering Systems Division, MIT, Cambridge, Massachusetts, USA.

43. Hou, F.L., 2011. “Self-reconfiguration planning for modular robots”, Ph.D thesis, University of Southern California, Los Angeles, California, USA.

44. Jones, C., and Mataric, M.J., 2003. “From local to global behavior in intelligent self-assembly”, *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp. 721-726, Taipei, Taiwan.

45. Karagozler, M. E., Campbell, J. D., Fedder, G. K., Goldstein, S. C., Weller, M. P., and Yoon, B. W., 2007. “Electrostatic Latching for Inter-module Adhesion, Power Transfer, and Communication in Modular Robots”, *Proceedings of the 2007 IEEE International Conference on Intelligent Robots and Systems*, pp. 2779-2786, San Diego, California, USA.

46. Kitano H., 2004. “Biological robustness”, *Nat. Rev. Genet*, Vol. 5, pp. 826–837.

47. Koos, S., Cully, A., and Mouret, J. B., 2013. “Fast damage recovery in robotics with the T-resilience algorithm,” *The International Journal of Robotics Research*, Vol. 32, No. 14, pp. 1700-1723.

48. Kotay, K., Rus D., Vona M. and McGray C., 1998. “Self-reconfiguring robotic molecule”, *Proceedings of the 1998 IEEE/RSJ International Conference on Robots and Systems*, pp. 424-431, Victoria, British Columbia, Canada.

49. Kruchten, P.B., 1995. “The 4+ 1 view model of architecture”, *IEEE Software*, Vol. 12, No. 6, pp. 42-50.

50. Kumar. V., Cotran R.S. and Robbins S.L., 1992. Basic Pathology, Philadelphia, Pennsylvania: Saunders.

51. Kurokawa, H., Tomita, K., Kamimura, A., Kokaji, S., Hasuo, T., Murata, S., 2008. “Distributed self-reconfiguration of M-TRAN III modular robotic system”, *The International Journal of Robotics Research*, Vol. 27, pp. 373–386.

52. Lin, Y., and Zhang, W.J., 2004. “Towards a novel interface design framework: function-behavior-state paradigm”, *International Journal of Human Computer Studies*, Vol. 61, No. 3, pp. 259–297.

53. Lipson, H., White, P., Zykov, V., and Bongard, J., 2005. “3D Stochastic Reconfiguration of Modular Robots”, *Proceedings of the Workshop on Self-reconfigurable Robotics at the Robotics Science and Systems Conference*, MIT, Cambridge, Massachusetts, USA.

54. Madden J.D., 2007. “Mobile Robots: motor challenges and material solutions”, *Science*,

Vol. 318, pp. 1094-1097.

55. Mahdavi, S. H., and Bentley, P. J., 2006. “Innately adaptive robotics through embodied evolution”, *Autonomous Robots*, Vol. 20, No. 2, pp. 149-163.

56. Matsumaru, T., 1995. “Design and control of the modular robot system: TOMMS”, *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp. 2125-2131, Nagoya, Aichi, Japan.

57. McAslan, A., 2010. “The concept of resilience: understanding its origins, meaning and utility”, from http://www.torrensresilience.org/images/pdfs/resilience%20origins%20and%20utility.pdf.

58. Merriam-Webster Inc. *Merriam-Webster's collegiate dictionary*. Merriam-Webster, 2004.

59. Morin, S. A., Shepherd, R. F., Kwok, S. W., Stokes, A. A., Nemiroski, A., and Whitesides, G. M., 2012, “Camouflage and display for soft machines,” *Science* , Vol. 337, No. 6096, pp. 828–832, August 2012.

60. Mostafa, K., Tsai, C. S., and Her, I., 2010. “Alternative gaits for multiped robots with leg failures to retain maneuverability”, *International Journal of Advanced Robotic Systems*, Vol. 7, No. 4, pp. 31.

61. Murata, S., Kurokawa, H., and Kokaji, S., 1994. “Self-assembling machine”, *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 441-448, San Diego, California, USA.

62. Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., and Kokaji, S., 2002. “M-TRAN: Self-reconfigurable modular robotic system”, *IEEE/ASME Transactions on*

Mechatronics, Vol. 7, No. 4, pp. 431–441.

63. Nelson, C. A., 2005. “A framework for self-reconfiguration planning for unit-modular robots”, Ph.D thesis, West Lafayette (Indiana): Purdue University, USA.

64. Ostergaard, E. H., and Lund, H. H., 2004, “Distributed cluster walk for the ATRON self-reconfigurable robot”, *The 8th Conference on Intelligent Autonomous Systems*, pp. 291–298, Amsterdam, The Netherlands.

65. Ostergaard, E. H., 2005. “Efficient distributed hormone graph gradients”, *The 9th International Joint Conference on Artificial Intelligence*, pp. 1489-1494, Edinburgh, Scotland.

66. Ostergaard, E. H., Kassow, K., Beck, R., and Lund, H. H., 2006. “Design of the ATRON lattice-based self-reconfigurable robot”, *Autonomous Robots*, Vol. 21, No. 2, pp. 165-183.

67. Pahl, G., and Beitz, W., 1999. “Engineering Design”, 2nd ed. Springer-Verlag, London Ltd.

68. Paredis, C. J. J., and Khosla, P. K., 1991. “An Approach for Mapping Kinematic Task Specifications into a Manipulator Design”, *Proceedings of the 5th International Conference on Advanced Robotics*, Vol. 1, pp. 556-561, Pisa, Italy.

69. Patternson, E. S., Woods, D. D., Cook, R. I., and Render, M. L., 2007. “Collaborative cross-checking to enhance resilience”, *Cogn. Tech. Work*, Vol. 9, pp. 155-162.

70. Prevas, K.C., Ünsal, C., Efe, M.Ö., Khosla, P.K., 2002. “A Hierarchical Motion Planning Strategy for a Uniform Self-Reconfigurable Modular Robotic System”, *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp. 787-792, Washington, District of Columbia, USA.

71. Roberts, R.G., and Maciejewski, A.A., 1996. “A local measure of fault tolerance for kinematically redundant manipulators”, *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, pp. 543-552.

72. Roberts, R. G., 2001. “The dexterity and singularities of an underactuated robot”, *Journal of Robotic Systems*, Vol. 18, pp. 159-169.

73. Roberts, R.G., Yu, H.G., and Maciejewski, A.A., 2008. “Fundamental limitations on designing optimally fault-tolerant redundant manipulators”, *IEEE Transactions on Robotics*, Vol. 24, No. 5, pp. 1224-1237.

74. Rus, D., and Vona, M., 1999. “Self-reconfiguration planning with compressible unit modules”, *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pp. 2513-2520, Detroit, Michigan, USA.

75. Rus, D., and Vona, M., 2001. “Crystalline robots: self-reconfiguration with compressible unit modules”, *Autonomous Robots*, Vol. 10, No. 1, pp. 107-124.

76. Russell, S. J., and Norvig, P., 2010. “Artificial Intelligence: A Modern Approach”, Upper Saddle River, New Jersey: Prentice Hall.

77. Sakka, S., and Chocron, O., 2001. “Optimal design and configurations of a mobile manipulator using genetic algorithms”, *Proceedings of 10th IEEE International Workshop on Robot and Human Interactive Communication*, pp. 268-273, Bordeaux, Paris, France.

78. Shen, W.M., and Will, P., 2001. “Docking in self-reconfigurable robots”, *Proceedings of the 2001 IEEE International Conference on Intelligent Robots and Systems*, pp. 1049–1054, Maui, Hawaii, USA.

79. Shen, W.M., Salemi, B., and Will, P., 2002. "Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots", *IEEE Transactions on Robotics and Automation*. Vol. 18, No. 5, pp. 700–712.

80. Shen, W.M., Krivokon, M., Chiu, H., Everist, J., Rubenstein, M. and Venkatesh, J., 2006. "Multimode locomotion via Superbot robots", *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, pp. 2552–2557, Orlando, Florida, USA.

81. Støy, K., and Nagpal, R., 2004. "Self-repair through scale independent self-reconfiguration", *Proceedings of 2004 IEEE/RSJ International Conference on Robots and Systems*, pp. 2062–2067, New Orleans, Louisiana, USA.

82. Stoy, K., Brandt, D., and Christensen, D. J., 2010. "Self-reconfigurable Robots: an Introduction", The MIT Press, Cambridge, USA.

83. Suh, N. P., 1990. "The Principles of Design", Oxford, New York: Oxford University Press.

84. Suh, J.W., Homans S.B., and Yim M., 2002. "Telecubes: Mechanical design of a module for self-reconfigurable robotics", *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp. 4095-4101, Washington, District of Columbia, USA.

85. Terada, Y., and Murata, S., 2008. "Automatic Modular Assembly System and its Distributed Control", *The International Journal of Robotics Research*, Vol. 27, pp. 445-462.

86. Thavamani S., 2006. "Fault Tolerant Control of a Ship Propulsion System", Ph.D thesis, Department of Electrical Engineering, State University of New York, Binghamton, USA.

87. Trivedi, D., Rahn, C., Kier, W., and Walker, I., 2008. "Soft robotics: biological inspiration,

state of the art, and future research", *Applied Bionics and Biomechanics*, Vol. 5, No. 3, pp. 99-117.

88. Unsal, C., and Khosla, P. K., 2001. "A multi-layers planner for self-reconfiguration of a uniform group of I-Cubes Modules", *Proceedings of the 2001 IEEE International Conference on intelligent Robots and Systems*, pp. 598-605, Maui, HI, USA.

89. Ünsal, C., Kiliççöte, H., and Khosla, P. K., 2001. "A modular self-reconfigurable bipartite robotic system: Implementation and motion planning", *Autonomous Robots*, Vol. 10, pp. 23-40.

90. Verma, A. K., Ajit, S., and Karanki, D. R., 2010. "Reliability and safety engineering", Springer-Verlag London Ltd.

91. Van Breda, A. D., 2001. "Resilience theory: a literature review", Technical report for South African Military Health Service, Military Psychological Institute, Social Work Research & Development, from http://vanbreda.org/adrian/resilience/resilience_theory_review.pdf.

92. Wang, J.W., and Zhang, W.J. (Supervisor), 2013. "Towards a resilient networked service system", Ph.D dissertation, Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Canada.

93. Yim, M., 1993. "A reconfigurable modular robot with many modes of locomotion", Proc. JSME Int. Conf. on Advanced Mechatronics, Japan, pp. 283-288.

94. Yim, M., Zhang, Y., Lamping J. O., and Mao, E.W., 2001. "Distributed control for 3D shape metamorphosis", *Autonomous Robots*, Vol. 10, No. 1, pp. 41-56.

95. Yim, M., Zhang, Y., Roufas, K., Duff, D., and Eldershaw, C., 2002. “Connecting and disconnecting for chain self-reconfiguration with PolyBot”, *IEEE/ASME Transactions on Mechatronics, special issue on Information Technology in Mechatronics*, Vol. 7, Issue 4, pp. 431-441.

96. Yim, M., Shen, W. M., Salemi, B., Rus, D., Moll, M., and Lipson, H., 2007. “Modular Self-Reconfigurable Robot Systems - Challenges and Opportunities for the Future”, *IEEE Robotics & Automation Magazine*, Vol. 14, pp. 43-52.

97. Yoshida, K., 1997. “A general formulation for under-actuated manipulators”, *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1651-1957, Grenoble, France.

98. Yoshikawa, T., 1985. “Manipulability of Robot Mechanisms”, *International Journal of Robotics Research*, Vol. 4, pp. 3-9.

99. Yu, C.H., Haller, K., Ingber, D., and Nagpal, R., 2008. “Morpho: A self-deformable modular robot inspired by cellular structure”, *Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems*, pp. 3571–3578, Nice, France.

100. Zhang, T., Liu, C.L., Qian, Z.Q., Zhang, D., Gupta, M. M., and Zhang, W.J., 2014a. “Configuration Synthesis of Under-actuated Resilient Robotic Systems”, *Proceedings of the 2014 IEEE/ASME international Conference on Advanced Intelligent Mechatronics Conference*, July 7-11, pp. 721-726, Besançon, France.

101. Zhang, T., Backstrom, K., Prince, R., Liu, C.L., Qian, Z.Q., Zhang, D., and Zhang, W.J., 2014b, “Robotic Dynamic Sculpture”, *IEEE Robotics and Automation Magazine*, Vol. 21,

Issue 3, pp. 96-104.

102. Zhang, W.J., 1994. “An integrated environment for CADCAM of mechanical systems”, Ph.D dissertation, Technical University of Delft, The Netherlands.

103. Zhang, W.J., Lin, Y., and Sinha, N., 2005. “On the function-behavior-structure model for design”, *The 2nd Canadian Design Engineering Network Conference*, Kananaskis, Alberta, Canada.

104. Zhang, W. J., 2007. “Is resilience the destiny for safety management paradigm?” Presentation at the Northeastern University of China, Shenyang, China.

105. Zhang, W. J., and Lin, Y., 2010. “On the Principles of Design of Resilient Systems and its Application to Enterprise Information Systems”, *Enterprise Information Systems*, Vol. 4, No. 2, pp. 99-110.

106. Zhang W.J., Sun Z.H. and Zhang B., 2011. “On the concept of the resilient machine”, *Proceedings of the* 6th *IEEE International Conference on Industrial Electronics and Application (ICIEA)*, June 21-23, Beijing, China.

107. Zhang, W.J., and Van Luttervelt, C.A., 2011. “Toward a resilient manufacturing system”, *Annals of CIRP*, Vol. 60, pp. 469-472.

108. Zhang, W.J., 2013. “Advanced Engineering Design Methodology”, Lecture Notes, Mechanical Engineering Division, University of Saskatchewan, Saskatoon, Canada.

109. Zhang, W.J., 2014a. “Dynamics of a System-Mechanical System”, Lecture Notes, Mechanical Engineering, East China University, Shanghai, China.

110. Zhang, W.J., 2014b. “Dynamics of a System”, Lecture Notes, Mechanical Engineering, East China University, Shanghai, China.

111. Zykov, V., Mytilinaios, E., Adams, B., and Lipson, H., 2005. “Self-reproducing machines”, *Nature*, Vol. 435, pp. 163-164.

112. Zykov, V., Mytilinaios, E., Desnoyer, M., and Lipson, H., 2007. “Evolved and designed self-reproducing modular robotics”, *IEEE Transactions on Robotics*, Vol. 23, No. 2, pp. 308-319.

113. Zykov, V., 2008. “Morphological and behavioral resilience against physical damage for robotic systems”, Ph.D Thesis, Cornell University, New York, USA.

APPENDIX: COPYRIGHT PERMISSION

Permission to Reuse Figure 2.3A, Figure 2.3B, Figure 2.3C, Figure 2.3D, Figure 2.3E, Figure 2.4B, Figure 2.4C, Figure 2.4D, Figure 2.5, Figure 2.7A, Fig. 2.8A, Figure 2.8B, and Figure 3.7

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

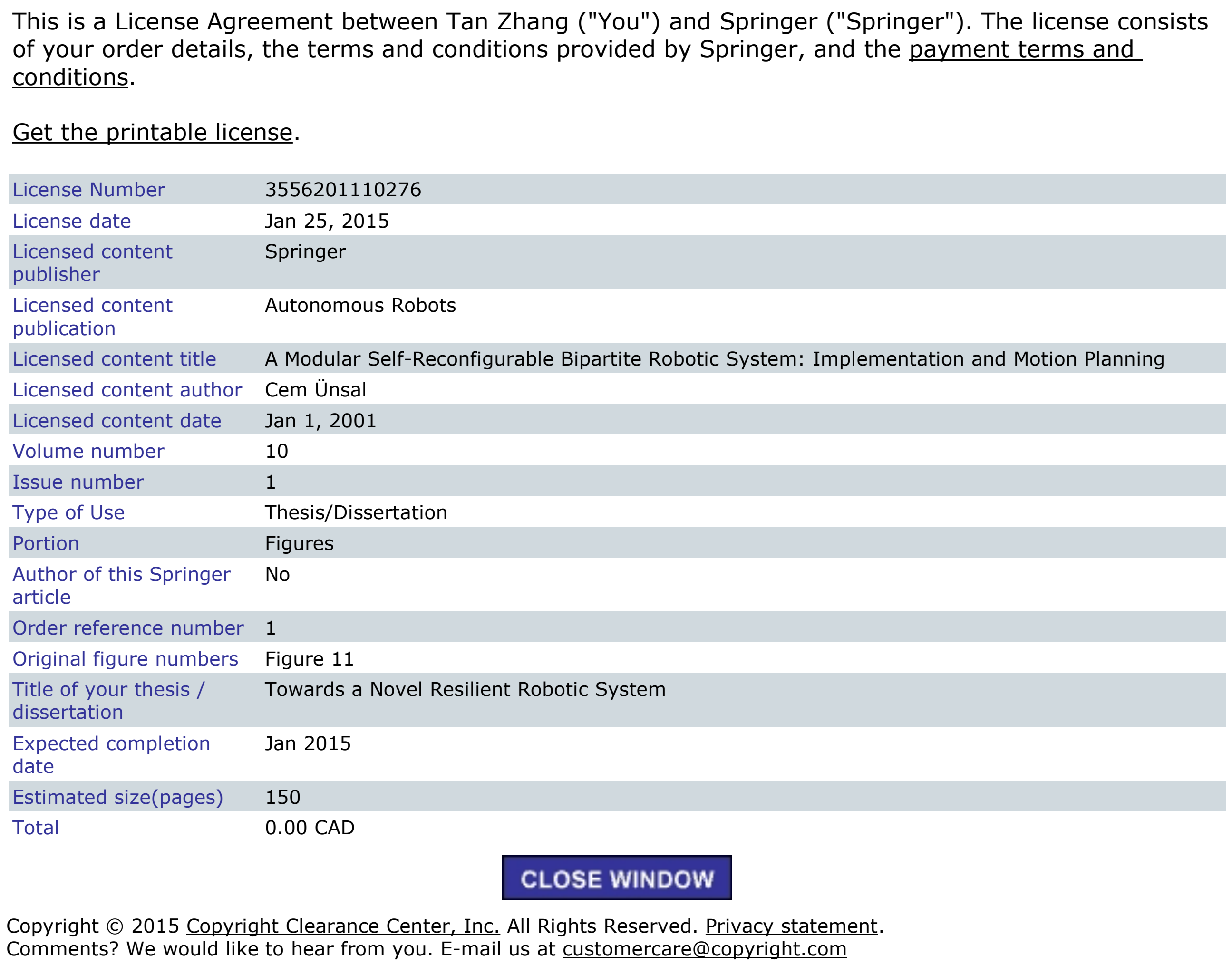
1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]

2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.

3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Permission to Reuse Figure 2.7B



This is a License Agreement between Tan Zhang ("You") and Springer ("Springer"). The license consists of your order details, the terms and conditions provided by Springer, and the payment terms and conditions.

Get the printable license.

License Number	3556201110276
License date	Jan 25, 2015
Licensed content publisher	Springer
Licensed content publication	Autonomous Robots
Licensed content title	A Modular Self-Reconfigurable Bipartite Robotic System: Implementation and Motion Planning
Licensed content author	Cem Ünsal
Licensed content date	Jan 1, 2001
Volume number	10
Issue number	1
Type of Use	Thesis/Dissertation
Portion	Figures
Author of this Springer article	No
Order reference number	1
Original figure numbers	Figure 11
Title of your thesis / dissertation	Towards a Novel Resilient Robotic System
Expected completion date	Jan 2015
Estimated size(pages)	150
Total	0.00 CAD

Permission to Reuse Part of the Work in Chapter 4

The copyright of the following papers:

Zhang, T., Liu, C.L., Qian, Z.Q., Zhang, D., Gupta, M. M., and Zhang, W.J., 2014a. "Configuration Synthesis of Under-actuated Resilient Robotic Systems", *Proceedings of the 2014 IEEE/ASME international Conference on Advanced Intelligent Mechatronics Conference*, July 7-11, pp. 721-726, Besançon, France.

Zhang, T., Backstrom, K., Prince, R., Liu, C.L., Qian, Z.Q., Zhang, D., and Zhang, W.J., 2014b, "Robotic Dynamic Sculpture", *IEEE Robotics and Automation Magazine*, Vol. 21, Issue 3, pp. 96-104.

are included in the following pages.

1/25/2015 Rightslink® by Copyright Clearance Center

Title: Configuration synthesis of underactuated resilient robotic systems

Conference Proceedings: Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on

Author: Tan Zhang; Changli Liu; Zhiqin Qian; Dan Zhang; Gupta, M.M.; Wenjun Zhang

Publisher: IEEE

Date: 8-11 July 2014

Copyright © 2014, IEEE

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

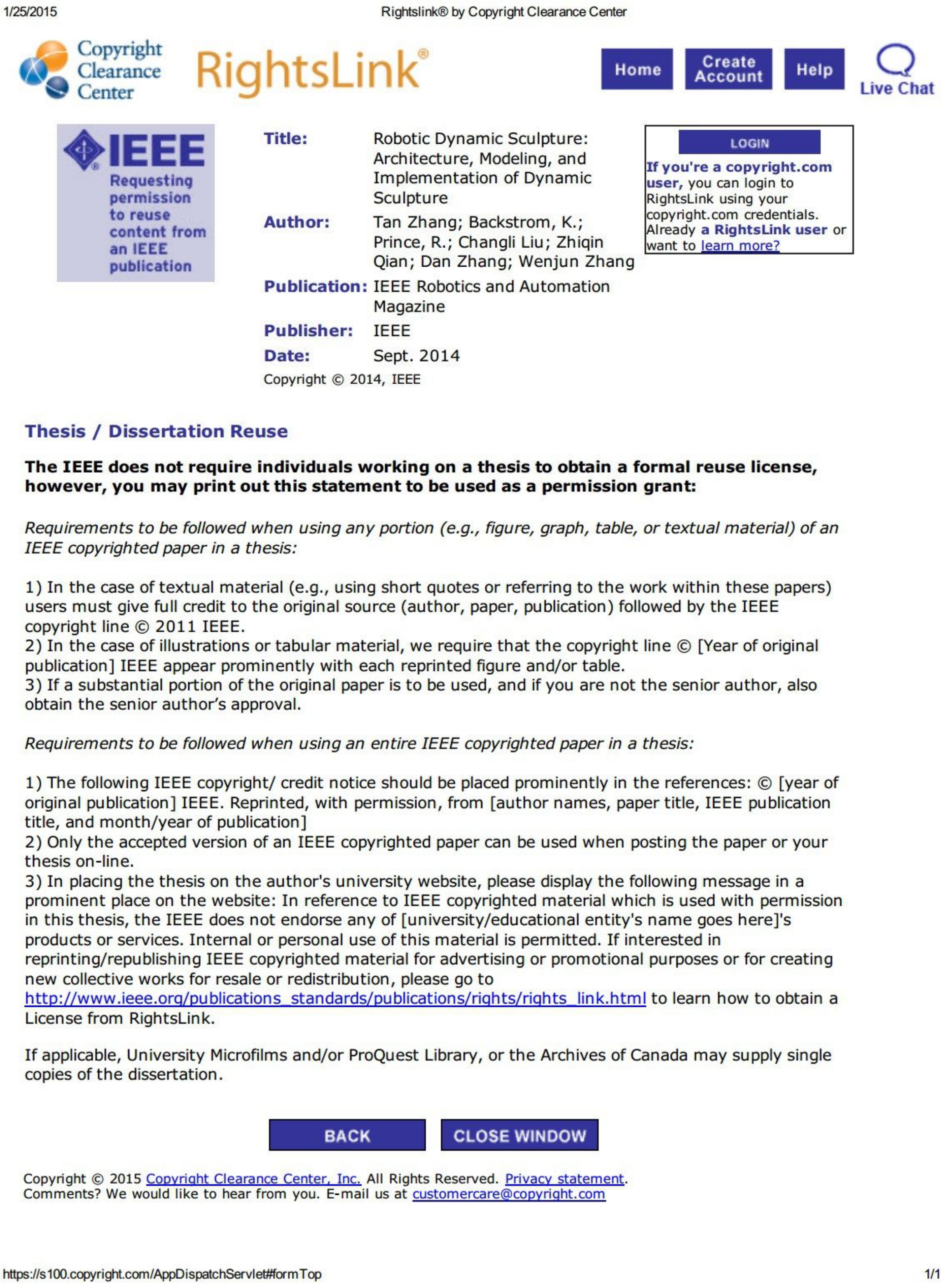
Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Copyright © 2015 Copyright Clearance Center, Inc. All Rights Reserved. Privacy statement.
Comments? We would like to hear from you. E-mail us at customercare@copyright.com

https://s100.copyright.com/AppDispatchServlet#formTop 1/1



1/25/2015 Rightslink® by Copyright Clearance Center

Title: Robotic Dynamic Sculpture: Architecture, Modeling, and Implementation of Dynamic Sculpture

Author: Tan Zhang; Backstrom, K.; Prince, R.; Changli Liu; Zhiqin Qian; Dan Zhang; Wenjun Zhang

Publication: IEEE Robotics and Automation Magazine

Publisher: IEEE

Date: Sept. 2014

Copyright © 2014, IEEE

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

https://s100.copyright.com/AppDispatchServlet#formTop 1/1