

# **On Adaptive Robot Systems for Manufacturing Applications**

A Thesis Submitted to the College of  
Graduate Studies and Research  
in Partial Fulfilment of the Requirements  
for the Degree of Doctor of Philosophy  
in the Department of Mechanical Engineering  
University of Saskatchewan  
Saskatoon

By  
**Zhuming Bi**

Fall 2002

© Copyright Zhuming Bi, 2002. All rights reserved.

## **Permission to Use**

In presenting this thesis in partial fulfilment 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, 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 Department of Mechanical Engineering

University of Saskatchewan

Saskatoon, Saskatchewan (S7N 5A9)

# **Abstract**

System adaptability is very important to current manufacturing practices due to frequent changes in the customer needs. Two basic concepts that can be employed to achieve system adaptability are flexible systems and modular systems. Flexible systems are fixed integral systems with some adjustable components. Adjustable components have limited ranges of parameter changes that can be made, thus restricting the adaptability of systems. Modular systems are composed of a set of pre-existing modules. Usually, the parameters of modules in modular systems are fixed, and thus increased system adaptability is realized only by increasing the number of modules. Increasing the number of modules could result in higher costs, poor positioning accuracy, and low system stiffness in the context of manufacturing applications. In this thesis, a new idea was formulated: a combination of the flexible system and modular system concepts. Systems developed based on this new idea are called adaptive systems. This thesis is focused on adaptive robot systems.

An adaptive robot system is such that adaptive components or adjustable parameters are introduced upon the modular architecture of a robot system. This implies that there are two levels to achieve system adaptability: the level where a set of modules is appropriately assembled and the level where adjustable components or parameters are specified. Four main contributions were developed in this thesis study.

*First*, a General Architecture of Modular Robots (GAMR) was developed. The starting point was to define the architecture of adaptive robot systems to have as many configuration variations as possible. A novel application of the Axiomatic Design Theory (ADT) was applied to GAMR development. It was found that GAMR was the one with the most coverage, and with a judicious definition of adjustable parameters. *Second*, a system called Automatic Kinematic and Dynamic Analysis (AKDA) was developed. This system was a foundation for synthesis of adaptive robot configurations. In comparison with the existing approach, the proposed approach has achieved systemization, generality, flexibility, and completeness. *Third*, this thesis research has developed a finding that in modular system design, simultaneous consideration of both kinematic and dynamic behaviors is a necessary step, owing to a strong coupling between design variables and system behaviors. Based on this finding, a method for simultaneous consideration of type synthesis, number synthesis, and dimension synthesis was developed. *Fourth*, an adaptive modular Parallel Kinematic Machine (PKM) was developed to demonstrate the benefits of adaptive robot systems in parallel kinematic machines, which have found many applications in machine tool industries. In this architecture, actuators and limbs were modularized, while the platforms were adjustable in such a way that both the joint positions and orientations on the platforms can be changed.



## Acknowledgements

*First*, I would like to express my sincere gratitude to my supervisor, *Dr. C. Zhang*, for his enthusiasm, invaluable guidance, constant encouragement, and full support during this research. His vision and broad knowledge played a key role in the realization of this work. His support is not limited in academics; he acted as an excellent advisor as well as a big brother, and he also made a great effort in developing my future career.

*Second*, I would like to extend special thanks to my advisory committee members: *Prof. R. Besant*, *Prof. W. Szyszkowski*, and *Prof. R. L. Kushwaha*. Their valuable examinations and constructive suggestions have greatly improved my present work. Especially, I give thanks to *Prof. R. Besant* for giving his insightful advice and help in many aspects.

*Third*, I would like to acknowledge *Dr. I-Ming Chen* at Nanyang Technological University, Singapore. He provided an excellent environment and made it possible for me to do experimental validation for part of my theoretical work. I would like to thank *Dr. Guilin Yang*, *Dr. Weihua Chen*, *Dr. Shusong Xing*, and *Dr. Dash A. Kumar* for their kindness during my time in conducting experiments in Singapore.

*Fourth*, I would like to acknowledge all of the colleagues in the *Advanced Engineering Design Laboratory* (AEDL) for their friendship and kind support during my studies. In particular, *Mr. K. D. Backstrom*, *Mr. P. R. Ouyan*, *Dr. X. B. Chen*, *Mr. P. R. He*, and *Ms. Y.*

*Z. Lin* provided good suggestions to improve my thesis writing.

*Finally*, I would like to thank the financial support during this research from the *Natural Sciences and Engineering Research Council (NSERC)*, *Atomic Energy of Canada Limited (AECL)*, and *Scholarship of University of Saskatchewan*.

## **Dedication**

To my dear wife *Rongrong*, lovely son *Chenghao*, and angelic daughter *Chengyu*.

Without their love, inspiration, and devoted support, this work would never been possible.

# Acronyms

ADT	Axiomatic Design Theory
AIM	Assembly Incidence Matrix
AKDA	Automatic Kinematic and Dynamic Analysis
AMTEC	The name of a Germany company
ARCS	Adaptive Robot Configuration Synthesis
CODM	Concurrent Optimal Design Method
COM	Center of Mass
DC	Direct Current
D-H	Denavite and Hartenberg method
DOF	Degree of Freedom
DP	Design Parameter
FR	Functional Requirement
GA	Genetic Algorithm
GAMR	General Architecture of Modular Robots
MB	Modular Behavior
MDO	Multi-Disciplinary Optimization
MODRO	MODular RObots
MRS	Modular Robotic System

NEOS	A name of a Sweden company
NRC	National Research Council
OIM	Object Incidence Matrix
PMCG	Parametric Module Configuration Graph
PKM	Parallel Kinematic Machine
POE	Product-Of-Exponential method
RMMS	Reconfigurable Modular Manipulator System
SB	System Behavior
SCARA	Selective Compliance Assembly Robot Arm
SEMORS	Simulation Environment for Modular Robot System
TELBOT	TELe-roBOT system

# List of Contents

<b>Permission to Use .....</b>	<b>i</b>
<b>Abstract.....</b>	<b>ii</b>
<b>Acknowledgements .....</b>	<b>iv</b>
<b>Dedication.....</b>	<b>vi</b>
<b>Acronyms.....</b>	<b>vii</b>
<b>List of Contents .....</b>	<b>ix</b>
<b>List of Figures.....</b>	<b>xv</b>
<b>List of Tables.....</b>	<b>xviii</b>
<b>Chapter 1 Introduction.....</b>	<b>1</b>
1.1 Changes in Manufacturing Environment .....	1
1.2 Solutions to Coping with the Changes .....	4
1.3 Adaptive Systems: General Research Issues .....	7
1.4 Motivation .....	8
1.5 Objectives .....	9
1.6 Thesis Organization.....	12
<b>Chapter 2 Literature Review .....</b>	<b>14</b>
2.1 Modular Robot Architecture.....	14
2.1.1 The need for modular robot systems.....	14

2.1.2 Existing modular robot systems.....	16
2.1.3 Concluding remarks .....	23
2.2 Kinematic and Dynamic Modeling .....	25
2.2.1 Kinematic and dynamic model formulation methods .....	25
2.2.2 Solution techniques .....	27
2.2.3 Modeling of modular robot configuration .....	28
2.2.4 Further discussion and concluding remarks.....	29
2.3 Modular Robot Configuration Synthesis.....	31
2.3.1 Design variables.....	33
2.3.2 Design constraints .....	34
2.3.3 Design objectives .....	35
2.3.4 Formulation of a design problem model .....	36
2.3.5 Design synthesis.....	38
2.3.6 Concluding remarks .....	40
<b>Chapter 3 Modular Robot Architecture, Configuration and its Representation ..</b>	<b>41</b>
3.1 Robot Variations in Manufacturing Environment .....	42
3.1.1 Serial robots .....	44
3.1.2 Parallel robots .....	45
3.1.3 Hybrid robots .....	49
3.2 Functional Requirement for Modular Robot Architecture .....	51
3.3 General Architecture of Modular Robots (GAMR).....	53
3.3.1 Design parameters and functional requirements.....	53
3.3.2 Definition of GAMR.....	55

3.4 Justification of GAMR .....	56
3.5 Modular Robot Configuration and its Computer Representation.....	57
3.5.1 Configuration .....	57
3.5.2 Computer representation .....	58
3.5.3 Computer representation of modules .....	59
3.5.4 Computer representation of assemblies .....	61
3.5.5 Example .....	63
3.6 Summary.....	66
<b>Chapter 4 Automated Kinematic and Dynamic Modeling of Modular Robot</b>	
<b>Configurations.....</b>	<b>68</b>
4.1 Modeling Methodologies.....	69
4.2 From the GAMR to the D-H Architecture.....	70
4.2.1 Joints and links from a viewpoint of the D-H architecture .....	75
4.2.2 Computer-aided generation of the D-H architecture.....	75
4.3 Kinematic Conversion .....	75
4.3.1 Labeling for joint and link bodies within a kinematic chain .....	75
4.3.2 Kinematic properties of a joint and linkage body .....	75
4.3.3 Determination of the D-H parameters for joint body $i$ .....	76
4.4 Dynamic Conversion.....	78
4.5 Computer Program Implementation .....	82
4.6 Validation.....	84
4.6.1 Example system for validation.....	85
4.6.2 SEMORS system .....	87



4.6.3 Results and discussion .....	88
4.6.4 Remarks .....	90
4.7 Summary.....	90
<b>Chapter 5 Modular Robot Configuration Synthesis.....</b>	<b>91</b>
5.1 Problem Definition .....	92
5.2 An Example Revisited .....	93
5.3 An Observation: the Need for Concurrent Design .....	95
5.4 Optimization Model for Configuration Synthesis .....	97
5.4.1 The optimization model .....	97
5.4.2 General strategies for solving the optimization problem.....	103
5.5 Implementation by Genetic Algorithm (GA) .....	105
5.6 Case Study .....	107
5.7 Summary and Discussion .....	113
<b>Chapter 6 Design of an Adaptive Parallel Robot System — A Case Study.....</b>	<b>116</b>
6.1 The Adjustable Platform.....	117
6.2 Case Study: Adjustable Platform Structure I (Off-Line) .....	119
6.2.1 The Stewart platform robot with the off-line adjustable platform .....	120
6.2.2 Kinematics .....	121
6.2.3 Performance analysis .....	125
6.3 Case Study: Adjustable Platform Structure I (On-Line).....	126
6.3.1 Kinematic equation and translation workspace .....	129
6.3.2 Impact on system stiffness by the on-line adjustable platform.....	131
6.3.3 Effect of actuation platform on end-effector motion .....	133

6.4 Related Research .....	136
6.5 Summary and Conclusions .....	138
<b>Chapter 7 Summary, Conclusions, and Future Work .....</b>	<b>139</b>
7.1 Summary .....	139
7.2 Conclusions .....	140
7.3 Future Work and Discussion.....	142
<b>References.....</b>	<b>145</b>
<b>Appendix A AMTEC Modular Robotic System .....</b>	<b>158</b>
A.1 Introduction .....	158
A.2 Hardware System.....	160
A.2.1 Features of modules .....	160
A.2.2 Rotary joint modules .....	160
A.2.3 Linear joint modules .....	163
A.2.4 Wrist modules .....	165
A.2.5 Gripper modules.....	166
A.2.6 Link modules.....	167
<b>Appendix B Design Analysis and Synthesis Program .....</b>	<b>169</b>
B.1 AKDA .....	169
B.2 ARCS .....	170
B.3 Program Output .....	170
B.4 Program Execution .....	171
<b>Appendix C GA Implementation Issues .....</b>	<b>174</b>

C.1 Implementation Strategy .....	174
C.2 GA Codings of Design Variables .....	175
C.3 Initial and Final Solution Sets.....	176

# List of Figures

Figure 1.1 Evolution of manufacturing in terms of product complexity and time-to-market in the car industry (Prasad 1996) .....	2
Figure 1.2 Evolution of manufacturing environment (adopted from NRC 2000) .....	3
Figure 1.3 Income rate versus the time-to-market of a new product .....	3
Figure 1.4 Adaptability to system cost and profit (Stake 1999).....	6
Figure 2.1 The joint modules for serial robots (Pritschow and Wurst 1996).....	17
Figure 2.2 The joint modules for parallel robots (Wurst 1999) .....	17
Figure 2.3 Reconfigurable modular manipulator system (Paredis and Khosla 1993) .....	18
Figure 2.4 Joint modules in MODRO system.....	19
Figure 2.5 A conceptual modular robot system (Chen and Burdick 1998).....	20
Figure 2.6 A parallel system based on the ATMEC modules (Chen 2001).....	20
Figure 2.7 MRS joint modules (Behabib and Dai 1991) .....	21
Figure 2.8 TELBOT system.....	22
Figure 2.9 Re-configurable platform manipulator system .....	23
Figure 2.10 Two types of modular system architecture from a viewpoint of parameters coupling/uncoupling .....	32
Figure 3.1 Five most common types of industrial robots (Warnecke et al. 1999) .....	45
Figure 3.2 Variations of parallel robots by assembly of limbs (Pritschow 1999).....	47
Figure 3.3 Variations of parallel robots by connections between limbs and platforms ...	48
Figure 3.4 Example of parallel robots: Delta robot .....	49
Figure 3.5 Comparison of parallel and hybrid robots (Tonshoff and Grendel 1999) .....	50

Figure 3.6 Example of hybrid robots (NEOS 2001) .....	51
Figure 3.7 Functional requirements (FRs) for a modular robot architecture .....	53
Figure 3.8 Design parameters for robot variations .....	54
Figure 3.9 Isomorphic module assemblies.....	58
Figure 3.10 Examples of a graph and a labeled graph.....	61
Figure 3.11 Example of a 3 DOF configuration .....	64
Figure 4.1 Architecture for the D-H parameters .....	72
Figure 4.2 The concept of virtual linkage body: the wrist module .....	73
Figure 4.3 Procedure to derive the D-H architecture from OIM .....	74
Figure 4.4 Dynamic parametric conversion.....	79
Figure 4.5 The program flow chart of AKDA .....	83
Figure 4.6 Experimental modular robot.....	86
Figure 4.7 SEMORS developed by Nanyang Technological University .....	87
Figure 4.8 The comparison of joint displacements from SEMORS and AKDA .....	89
Figure 5.1 The construction of a 3 DOF configuration.....	93
Figure 5.2 Modular variables, and kinematic and dynamic parameters .....	95
Figure 5.3 Relationship between OIM and design variables .....	99
Figure 5.4 Decomposition of the design space .....	104
Figure 5.5 From design variables to GA strings .....	106
Figure 5.6 Feasible solution (1) consisting of 3 rotary modules (fitness= - 46.4797)...	108
Figure 5.7 Feasible solution (2) consisting of 3 rotary modules (fitness= - 11.0039)...	109
Figure 5.8 Best/average fitness (N·m) .....	110
Figure 5.9 Optimal design variables (normalized) .....	110
Figure 5.10 Feasible solution (3) consisting of one rotary module and two linear	

modules (fitness= - 43.89332) .....	111
Figure 5.11 Feasible solution (4) consisting of one rotary module and two linear modules (fitness= - 15.9761) .....	112
Figure 6.1 Methods to change platform dimensions .....	118
Figure 6.2 4 Structure types of the adaptive platform .....	119
Figure 6.3 Stewart platform structure .....	120
Figure 6.4 Indices to compare the translation workspace .....	125
Figure 6.5 Indices of translation workspace vs. $d_b$ .....	127
Figure 6.6 Cross-section shapes of translation workspaces .....	128
Figure 6.7 Cross-section shapes of translation workspace .....	130
Figure 6.8 System stiffness index of the Stewart robots .....	132
Figure 6.9 Stewart robot with an on-line adjustable platform .....	133
Figure 6.10 Block diagram of the Jacobian matrix .....	135
Figure 6.11 Gain vector measures along a trajectory .....	136
Figure A.1 Hardware architecture of the AMTEC system .....	159
Figure A.2 Software architecture of the AMTEC system .....	159
Figure A.3 Rotary joint module .....	163
Figure A.4 Linear joint module .....	164
Figure A.5 Link modules in the AMTEC system .....	167
Figure B.1 Path setting for executing the program .....	172
Figure B.2 Execution of the main program .....	172
Figure B.3 Window for the graphic simulation .....	173
Figure C.1 Coding scheme for design variables .....	176

## **List of Tables**

Table 2.1 Comparative overview of the existing studies .....	31
Table 3.1 Comparison of parallel robots with serial robots (Uchiyama 1994) .....	46
Table 3.2 The relationship between DPs and GAMR features .....	56
Table 3.3 A comparison of GAMR with others existing architectures .....	57
Table 3.4 Data objects for joint and link modules .....	64
Table 4.1 Prescribed pose for task specification .....	86
Table 5.1 Task specification .....	107
Table A.1 Features of rotary modules .....	161
Table A.2 Features of linear modules.....	163
Table A.3 Features of wrist modules.....	165
Table A.4 Features of gripper modules .....	166
Table A.5 Features of link modules .....	167
Table C.1 Design variables and fitness at the initial population (with three rotary modules).....	177
Table C.2 Design variables and fitness of the different robot configurations at the terminal population (with three rotary modules) .....	179
Table C.3 Design variables and fitness of the different robot configurations at the terminal population (with two linear modules and one rotary module).....	180

# **Chapter 1**

## **Introduction**

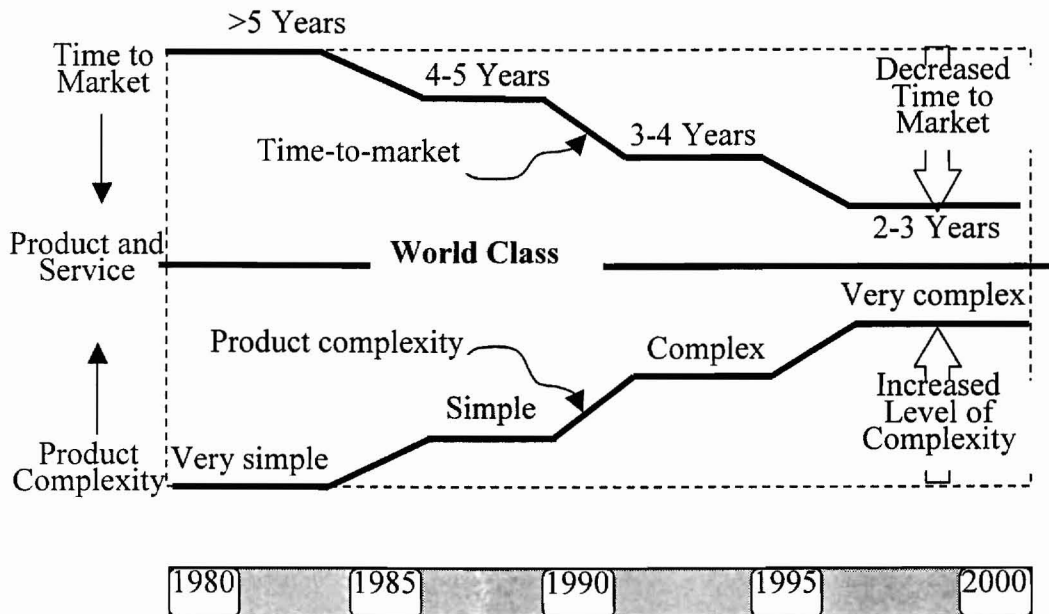
This chapter discusses the motivation and objective of this thesis study. It begins with the description of a bottleneck problem in manufacturing today, i.e., the need for frequent changes over a product life cycle. The solution to cope with these changes is the concept of an adaptive system. A definition of an adaptive system is given. The adaptive system concept leads to the concept of modular systems, which is the most important means to realize adaptive systems. This study is focused on modular systems with adjustable parameters, in particular, on the task-oriented synthesis of such systems. As a general research strategy, the study investigates modular robot systems as an example, because robots are dynamic systems and the core of many advanced manufacturing systems.

### **1.1 Changes in Manufacturing Environment**

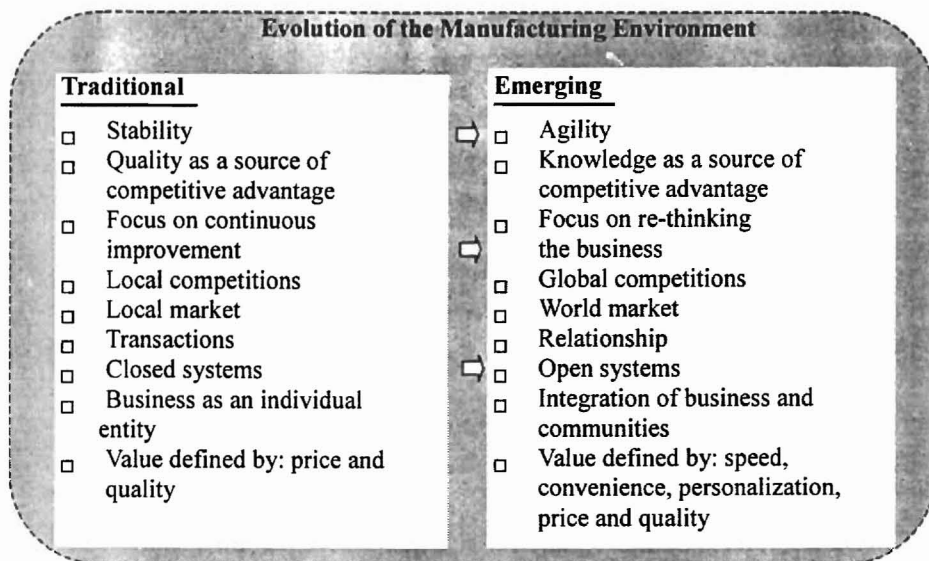
Manufacturing supply becomes more and more saturated demand with respect to the demands of the global market due to the rapid development of science and technology. It is possible for customers to demand products of higher quality, longer durability, personalized appearance, shorter delivery, and lower prices. Manufacturing companies



are pressured to deliver more variations of new products in an ever-increasing pace in order to be competitive. As a result, the lifetime of product becomes shorter and shorter, and the structure of a product becomes more and more complex. Considering the car industry as an example as shown in Figure 1.1 (Prasad 1996). Over the past several years, the variety and complexity of new cars has grown multi-fold from “very simple” to “very complex”; while at the same time, the time from conceptual design to market has shrunk. This evolution also happened in other areas, such as the computer and software industry. The evolution of manufacturing, in terms of the time to market and product complexity, was summarized in (NRC 2000); see Figure 1.2. The evolution shown in Figure 1.2 has implied that a manufacturing company must now be ready to change itself for uncertain and frequently changing environment.

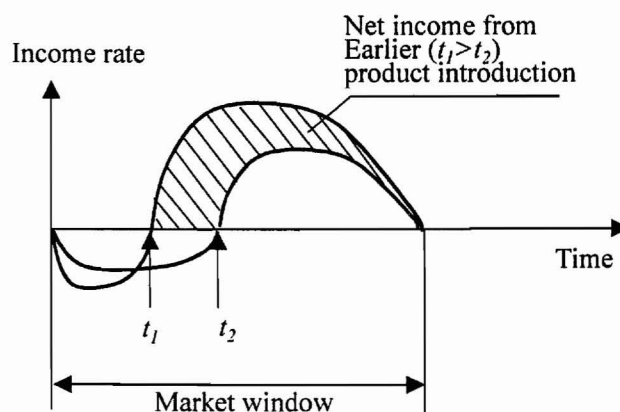


**Figure 1.1** Evolution of manufacturing in terms of product complexity and time-to market in the car industry (Prasad 1996)



**Figure 1.2** Evolution of manufacturing environment (adopted from NRC 2000)

The complexity of products implies that customers prefer products each with more functions. As shown in Figure 1.3, the time-to-market,  $t_1$  or  $t_2$ , influences both the product profitability and the potential market share as implied by the net area under these curves. The earlier a product is introduced into the market, the more profit and a greater market share can be rewarded to a company provided the product quality is the same for both (Smith and Reinertsen 1998).



**Figure 1.3** Income rate versus the time-to-market of a new product

The following can be concluded from the above discussions:

*Quick changes of customers' requirements are a theme of today's manufacturing environment. Most manufacturing companies must provide many product variations to the market with short delivery times.*

One may argue that the ultimate goal of a manufacturing enterprise is to make the maximum profit when providing products to the customers; therefore, product quality and manufacturing cost and final price are important factors for a product. This statement needs to be refined; as the customer now often regards higher quality as part of the product functions and manufacturing cost is only a fraction of the total cost. With regard to the product, a manufacturing company can say nothing about the price if it cannot provide competitive products to markets at a competitive price and meet the need for frequent changes of customer's requirements.

## **1.2 Solutions to Coping with the Changes**

There are several solutions for coping with these changes. One of the solutions is to improve quality of products. This may reduce the customers' desire for change because the high quality does correlate with the customer satisfaction. However, the alleviation of the demand on the changes, with the same high quality products, is not sufficient nowadays with the quality improvements of all manufactured products that were achieved in the 1980s; customers now require that products meet their ever-changing

desires. There is also a point where the changes are simply because of the uncertainty and inconsistency, which might occur due to the pursuit of short product development lead time through a concurrent and collaborative product development process. Reduction of the cost does not seem to be directly relevant to accommodating the changes.

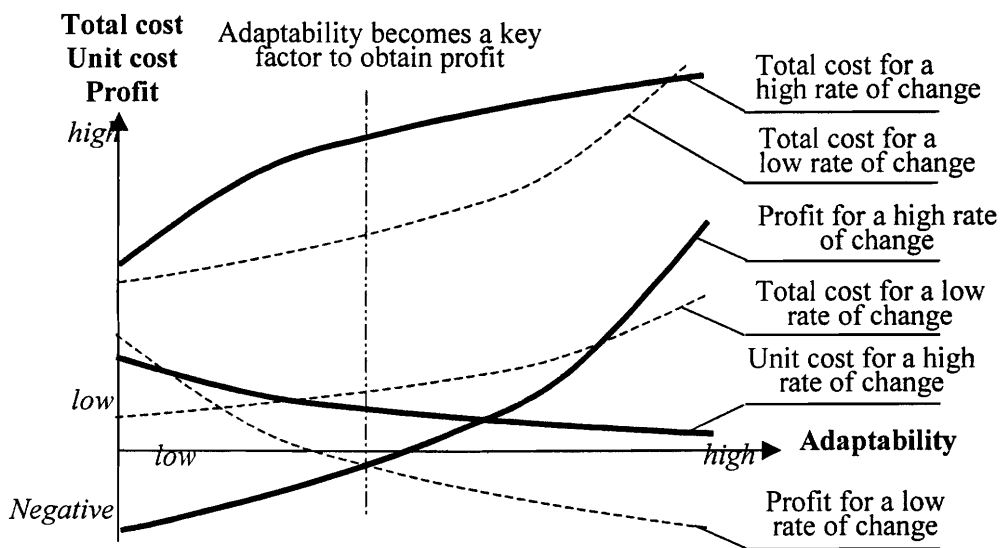
The most straightforward solution to the need for changes is to make a manufacturing system able to change with customer requirements. This concept is called an adaptive system. The adaptive system is a system that can change its functions with the change of tasks. It should be noted that the key concept in the adaptive system is system adaptability. A more formal definition of system adaptability, adapted from (Bordoloi et al. 1999), is given as follows.

**Definition 1.2** *Adaptability*. A measure of the system's ability to adapt to unanticipated internal or external changes accurately and rapidly in a manufacturing environment.

It should be further noted that the system adaptability could be achieved by making a system with a modular architecture and/or by making a system possess adjustable components/parameters. A further discussion on this issue will be given later in this thesis.

The example of an internal change is the change of available resources; the example of an external change is the change of the requirements on products being developed.

Generally, increasing system adaptability increases the total cost of manufacturing systems because the additional cost incurred to implement adjustments of the dynamic factors. However, the unit cost depends not only on the total cost, but also on the quantities of products sold. There are some studies on the relationships among system adaptability, manufacturing cost, and enterprise profits (Stake 1999). Their views are summarized in Figure 1.4, which shows the change trends of the unit cost, total cost, and profit versus the degree of system adaptability. In the case of a high rate of changes (shown with solid lines), the higher system adaptability, the higher the profit a manufacturing enterprise may achieve.



**Figure 1.4** Adaptability to system cost and profit (Stake 1999)

Two ways to make system more adaptive (Bi and Zhang 2001a) are the concept of flexible system and the modular system. The *flexible system concept* is based on increasing the *adjustable components*. These components are related to the manufacturing planning, scheduling, and control processes. The example of the system

developed following the flexible system concept is the flexible assembly system. The *modular system concept* requires developing an architecture of a system such that the system can be modified simply by assembling different modules. The parameters corresponding to the assembly are called *system dynamic parameters*. The modular system concept implies that the system topology can be changed by removing or adding modules. As such, the space of tasks that may be fulfilled increases nearly indefinitely (Ulrich 1995). A modular robot system is a typical example, which is developed on the basis of the modular system concept.

### **1.3 Adaptive Systems: General Research Issues**

There are two general research issues in the development of adaptive systems: (i) system architecture, and (ii) task-oriented determination of a configuration of the system based on a particular architecture.

The goal of studying issue (i) is to define primary system components to gain as much adaptability as possible subject to constraints derived from a particular application where the adaptive system is supposed to operate. The goal of studying issue (ii) is:

*Given a set of candidate modules and adjustable parameters, produce a design that is composed of a subset of the candidate modules and adjustable parameters that satisfies both a set of functional requirements and a set of constraints* (Grady and Liang 1998).

This goal clearly sets out a pre-condition. That is, the adaptive architecture of a particular system is specified, and types of modular components and types of their interactions are predefined. It is further noted that in this thesis an architecture of an adaptive system will take the modular architecture as a backbone with added adjustable parameters. Throughout the thesis, the term adaptive system is interchangeably used with the term modular system, unless otherwise specified. Hence, general research issue (ii) may simply be stated as the determination of modular system configurations.

Without loss of generality, while facilitating the development of theory and methodology for adaptive systems, this thesis uses robots as a vehicle. Robots are now indispensable and play an ever-increasing role in current manufacturing systems. It should be noted, however, that the robot systems discussed here, in a broad sense, are not restricted to conventional ‘serial’ robots, because they can be ‘parallel’ and ‘hybrid’ robots. A discussion of the definition of robots will be given in Chapter 2.

## **1.4 Motivation**

The motivation of this thesis research is based on the understanding that system adaptability is one of the most important factors to improve current manufacturing systems, and a systematic approach to achieve the best adaptability of robotic-based manufacturing systems is lacking. For example, no answer is available to the general questions of (i) why are the existing architectures of modular robots are defined as such,

and (ii) how can a task-oriented, optimal modular configuration be determined. It should be significant to the theory and methodology for manufacturing systems that a thorough investigation on the adaptive system concept is carried out.

The system adaptability could be achieved by both making certain components adjustable and designing systems to be of modular architecture. Furthermore, based on a preliminary analysis of the related work, the author was convinced that not modularity of a system but system adaptability is the more important ultimate goal.

## 1.5 Objectives

This thesis embarked a pilot investigation into a theory and methodology for adaptive systems in the context of manufacturing applications, where robotic systems are taken as an example throughout the thesis. Both issues discussed in Section 1.3 will be studied in this thesis, though issue (ii) will be more emphasized. The following research objectives are stated, where the ultimate goal is to achieve a high adaptability for robotic-based manufacturing systems.

**Objective 1:** *To develop a General conceptual Architecture of Modular Robot systems (GAMR) with adjustable parameters.* The generality of GAMR will be evaluated from two aspects (i) coverage of existing architectures of modular robot systems, and (ii) suitability for manufacturing applications. A conceptual architecture means that, for example, the selection of materials for modules and fabrication of modules are not the scope of this thesis.



A GAMR will be developed as a benchmark to evaluate the system adaptability of current modular robot systems, and GAMR-based task-oriented design methods will be applied to specific modular robot systems. The system architecture determines the possible system configurations, because the architecture specifies the primary building blocks and their connections. In order to make systems more adaptive, the configuration variations are expected to be as many as possible. However, the architecture of a modular robotic system must also consider the efficiency of forming a modular assembly, system stiffness, and the use of off-the-shelf components.

**Objective 2:** *To develop a general and systematic method for computer generation of kinematic and dynamic models for modular robot systems.* Systematization will be evaluated based on the need for human intervention during the formulation of the kinematic and dynamic models. The generality is associated with objective 1 in the sense that the method developed with objective 2 should be applicable to the GAMR developed with objective 1.

This work is essential because (i) a large number of feasible configurations should be evaluated in the procedure of configuration design, and (ii) kinematic and dynamic modeling of modular configurations is different from that of a configuration with a fixed structure in terms of design parameters and modeling procedure.

**Objective 3:** *To develop a general method for determining a task-oriented optimal configuration of modular robotic systems.* The generality of the method will be evaluated from two aspects (i) coverage of both topology synthesis and dimension

synthesis, and (ii) applicability to the GAMR.

There is a strong coupling among modular design variables with respect to design objectives and constraints. Consequently, any sequential design methods, which have been used in designing modular robotic configurations, are not adequate now. A new design method needs to be developed.

**Objective 4:** *To demonstrate the benefits of integrating the modular and flexible system concept to achieve more system adaptability using the example of modular parallel kinematic machine (PKM) with an adjustable platform.* The special emphasis is placed on the demonstration of increased system adaptability with adjustable components.

This case is regarded as a modular Parallel Kinematic Machine (PKM) with an adjustable platform. It should be noted that the PKM systems are now replacing or enhancing traditional machine tools in manufacturing industries. Therefore, the research on this objective also has significance to machine tool applications.

It should be noted that there are many other issues pertinent to adaptive systems, such as cost and accuracy. The cost of a modular robot system is in general higher than that of a non-modular robot systems, but the current technology for fabrication of modular components appears capable of producing cost-bearable modular robot systems. The reduction of the unit cost for modular robots is delegated to technologies for individual components, such as sensors and motors, and is not a concern of this thesis study.

Current modular robot systems cannot achieve as high accuracy as a non-modular robot system. The main reason for the reduced accuracy is not due to individual modules, e.g., sensors and motors, but rather due to the errors produced in assembling modules and in controlling the relative motion among modules. Reduction of the errors can be achieved by either (i) increasing the fabrication and assembly quality for each modular component, or (ii) compensating the accumulated modular errors in a new improved control system. Although these error-reducing methods may be feasible, they are not the scope of research in this thesis.

## **1.6 Thesis Organization**

The remainder of the thesis is organized as follows:

Chapter 2 provides a critical review of the literature on: modular robot architecture design, automatic kinematic and dynamic modeling, and task-oriented configuration design. Problems implied by the shortcomings of the existing literature, with reference to the objectives of this thesis, will be highlighted.

Chapter 3 presents the development of the general conceptual architecture of modular robot systems corresponding to objective 1. This new architecture incorporates the flexible system concept, adjustable parameters in this case, into the modular system architecture. This new architecture may also be called the architecture of adjustable modular robot systems. A new representational method for modular robot configurations based on the GAMR is developed.

Chapter 4 presents the development of a systematic procedure for automatic kinematic and dynamic modeling of adjustable modular robot configurations corresponding to objective 2. This includes a general discussion of the strategy for developing such a procedure, which leads to a so-called indirect modeling method. Based on this method, a complete program system for computer generation of the kinematic and dynamic equations is developed. There is also a discussion of the validation of the developed program through experiments.

Chapter 5 presents a new method for task-oriented modular robot configuration synthesis (synthesis for short). It starts with the definition of the scope of the synthesis, and then elaborates on a proposed method for the synthesis. The synthesis problem is modeled as an optimization problem. A discussion of the formulation of the optimization problem for type synthesis, number synthesis, and dimension synthesis is also presented. Finally, an implementation and a case study are given.

Chapter 6 presents a case study of the parallel kinematic machine (PKM) system to show the benefits of the adjustable modular robot system for manufacturing applications. The PKM system is modified into an adjustable modular PKM by following the proposed GAMR discussed in Chapter 3. A detailed discussion of the realization of this adjustable modular PKM system is presented.

Chapter 7 presents the conclusions of this work and recommends some new research.

## **Chapter 2**

### **Literature Review**

This chapter provides a critical review of (i) development of modular robot architecture, (ii) automatic modeling for modular robots, and (iii) modular robot configuration synthesis. The purpose of this review is to provide a further justification of the needs and the scope of research based on the objectives in Chapter 1. These needs and scope are explicitly elaborated at the end of a discussion of each subject.

#### **2.1 Modular Robot Architecture**

##### **2.1.1 The need for modular robot systems**

There are various demands on a robot system, such as the speed, accuracy, workspace, and the loading capacity. Manufacturers of industrial robots are forced to emphasize one compromise from all possible solutions. However, it is difficult to design a single robot that is simultaneously strong enough, nimble enough, and accurate enough to meet all the task requirements for an application (Chen and Burdick 1998). This difficulty can be observed easily by comparing the features of serial and parallel robots, or even two serial robots with the same Degree of Freedoms (DOFs). As an example, a Selective Compliance Assembly Robot Arm (SCARA) robot has a very fast cycle time, excellent repeatability, but its reachable range in the vertical direction is very limited, so it is only

suitable for a planar assembly task. An articulated robot has a large workspace with similar reachable ranges in all directions, and a good dexterity in avoiding spatial obstacles, but its accuracy is relatively poor; besides, it has complex kinematics and dynamics to make high-performance control difficult.

There are many definitions on modular robot systems (Tesar and Butler 1989, Benhabibi and Dai 1991, Paredis et al. 1996, Pritschow and Wurst 1996, Pamecha et al. 1997). Generally, a modular robot system consists of a stock of interchangeable modules (typically, link and joint modules) with various sizes and performance specifications. The combination of re-configurable modular hardware with modular software tools allows the user to rapidly create a most suitable robot structure, which is customer-tailored for a given task.

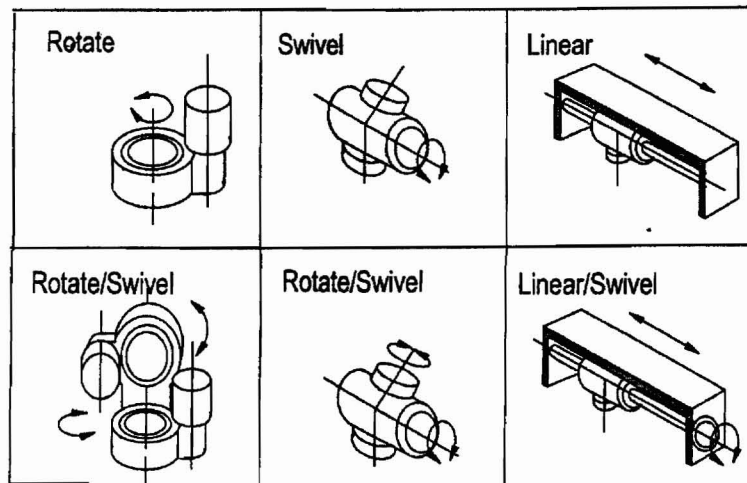
The concept of modular robot system provides a promising method for enhancing the adaptability of the present robots without incurring higher costs, while at the same time opening new market segments. Some merits of modular robot systems are listed as follows:

- (i) Flexibility: the customer-tailored configuration requirements can be met through the optimal reconfiguration of the modules.
- (ii) Low-cost: the standardization of the modules allows them to be manufactured with low unit costs.
- (iii) Interconnectability: modules are interchangeable and replaceable with each other.
- (iv) Reliability: redundancy that may be brought in with modular architecture is conducive to dealing with system malfunctions.

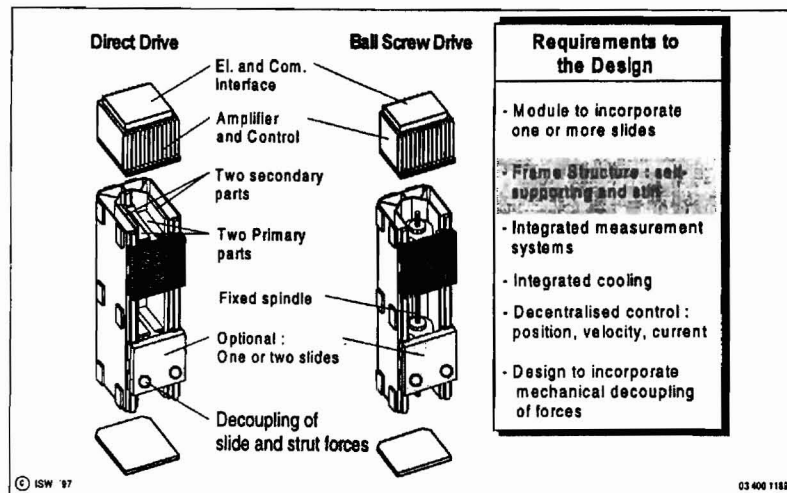
### **2.1.2 Existing modular robot systems**

The existing research efforts in modular robot systems include system hardware design, kinematics and dynamics, control, calibration, and modular robot configuration synthesis. This section is a literature survey on the development of robot modular components and modular robot systems.

Wurst (1986) began his studies in this area in the early 80's. As a robot construction could be divided into joint components and joint connections, he focused on the development of encapsulated modules. His survey showed that more than 80% of commercial industrial robots were assembled based on 8 different kinematic types with 4 or 6 DOFs, and these robots can be constructed using a set of drive and joint connection modules. As shown in Figure 2.1, he developed joint modules including rotary modules with parallel axes, rotary modules with vertical axes, linear modules and three other modules that combine the basic modules. His recent interest is the design of parallel modular robots (Wurst 1999). Figure 2.2 shows the joint modules used in his parallel robots.



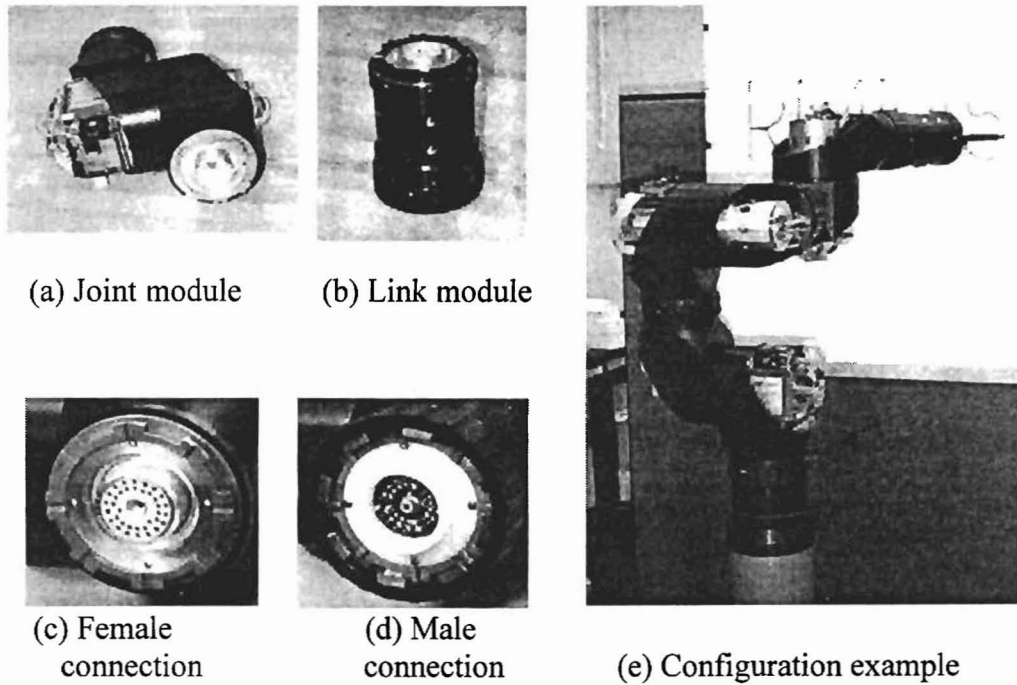
**Figure 2.1** The joint modules for serial robots (Pritschow and Wurst 1996)



**Figure 2.2** The joint modules for parallel robots (Wurst 1999)

Reconfigurable Modular Manipulator System (RMMS) was developed at Carnegie Mellon University (Paredis and Khosla 1993a). This system has the joint and link modules, see Figure 2.3a and b, and a quick-coupling mechanism designed for module connection, see Figure 2.3c and d. Each joint module is actuated by a DC motor in conjunction with a harmonic-drive mechanism. An example is shown in Figure 2.3e.





**Figure 2.3** Reconfigurable modular manipulator system (Paredis and Khosla 1993)

A MODular RObotic system (MODRO) was developed by the Institute of Robotics, Swiss Federal Institute of Technology (Nielsen and Huppi 1992). This system was constructed based on the building block principle. As shown in Figure 2.4, all MODRO joint modules were built with the mounts for standard profiles. Brushless DC or stepper motors were used to supply power. The linear modules were also built into the standard profiles and driven by the tooth belts or by the ball bearing screws, depending on the required speed and payload.



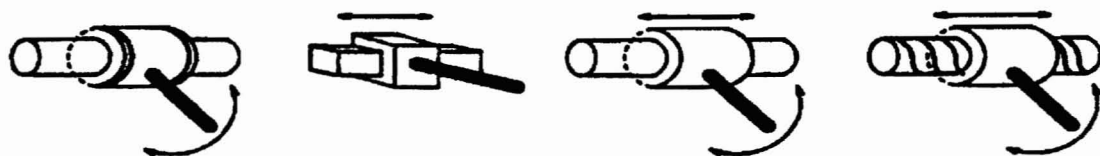
(a) Linear joint modules



(b) Rotary joint modules

**Figure 2.4** Joint modules in MODRO system

Chen and Burdick (1998) presented a conceptual modular robot system. Some of the unique features of their system are (i) types of joint modules include not only rotary joints (R) and linear joints (L), but also helical joints (H) and cylindrical joints (C), as shown in Figure 2.5a, and (ii) the link modules can have more than two assembly ports, which are symmetrically arranged. Figure 2.5b shows two types of joint modules: a cubic link with 6 assembly ports and a prism link with 10 assembly ports. Figure 2.6 shows a modular parallel robot that was built by extending the AMTEC system (Jaenisch et al. 2000).



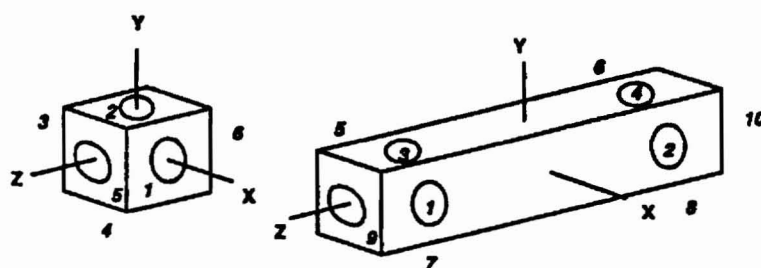
Joint (R)

Joint (L)

Joint (C)

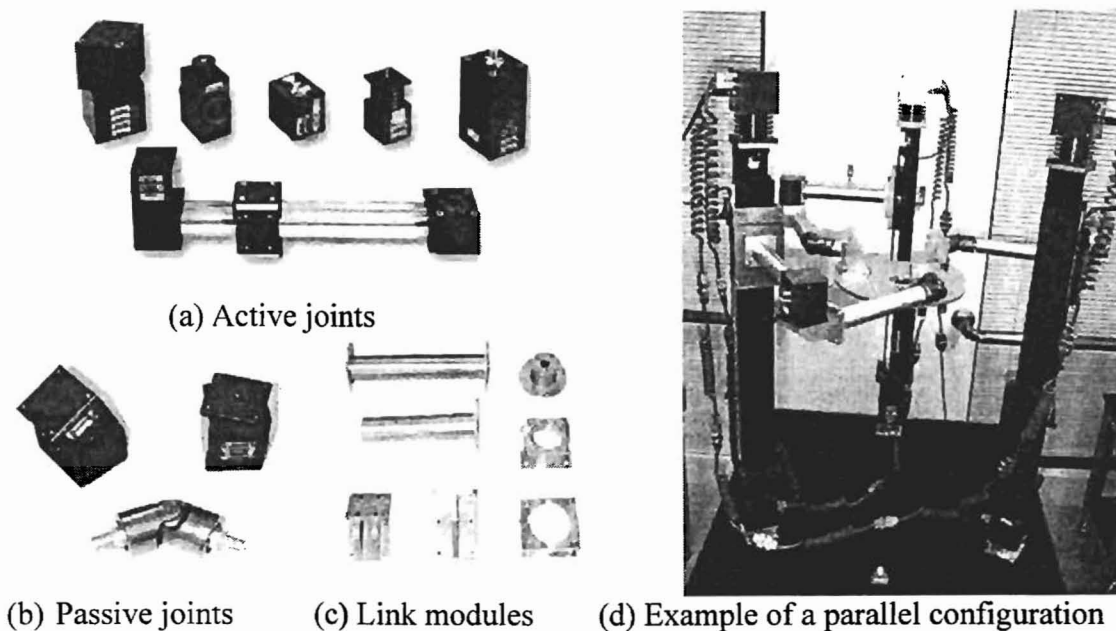
Joint (H)

(a) 4 types of join modules



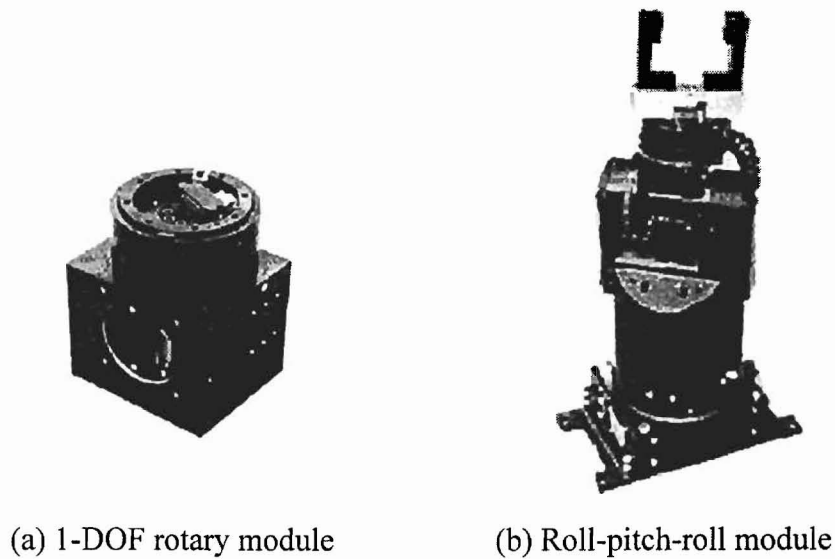
(b) Cube and prism link modules

**Figure 2.5** A conceptual modular robot system (Chen and Burdick 1998)



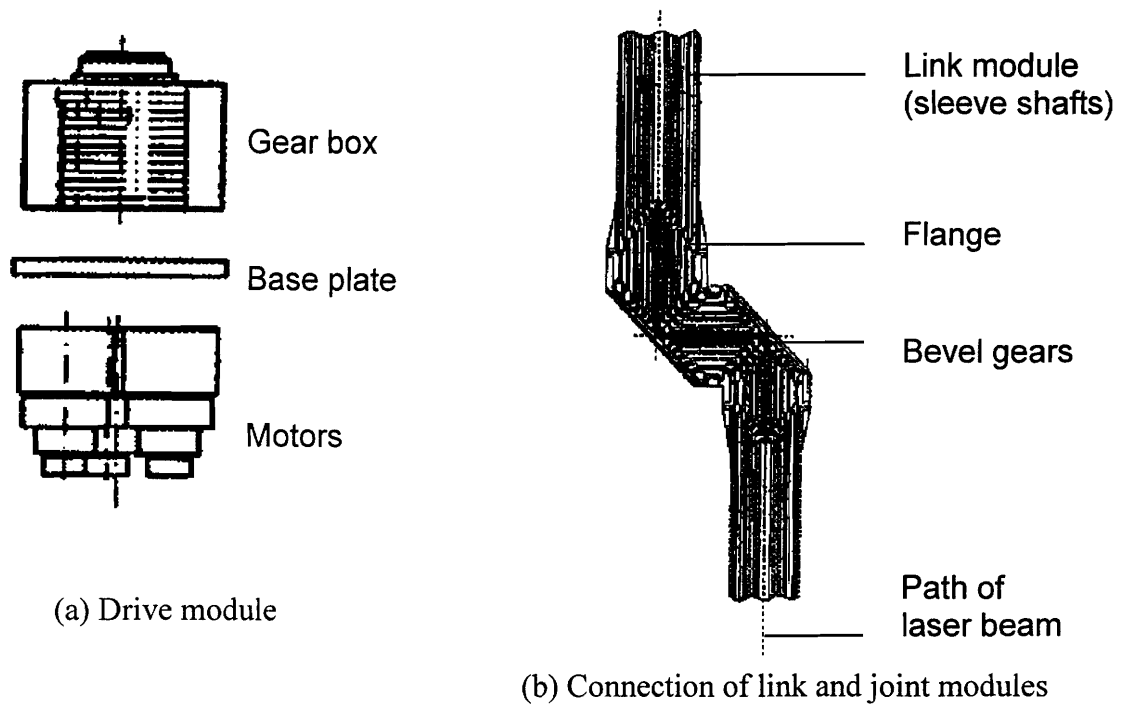
**Figure 2.6** A parallel system based on the ATMEC modules (Chen 2001)

The Modular Robotic System (MRS), which was developed at the University of Toronto, has a specific link with 45-degree connection surface to allow for both the straight and perpendicular connections (Behabib and Dai 1991, Benhabib et al. 1992). MRS is now commercially available, and it contains 1-DOF joints and joints with multi-DOFs, such as roll-pitch-roll module, linear-poll module, and roll-pitch module. Figure 2.7 shows a 1-DOF rotary joint module, and a 3-DOF roll-pitch-roll joint module.



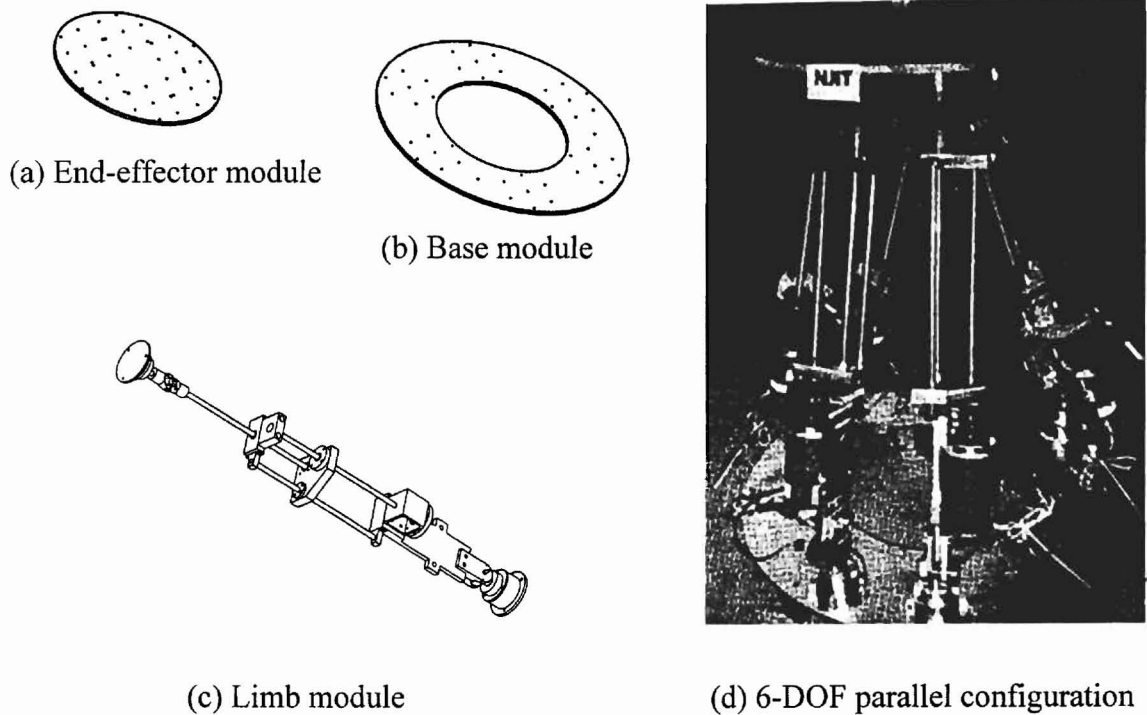
**Figure 2.7** MRS joint modules (Behabib and Dai 1991)

Figure 2.8 shows the Modular Tele-Robot System (TELBOT) that was developed by Walischmiller and Frager (1994). The distinctive point of TELBOT is that driving modules are all arranged on the base. The transmission of both the motion and force from driver to any link, including the gripper, is through a structure shown in Figure 2.8 b. The system has concentric sleeve shafts and 1:1 bevel wheels along the kinematic chain to implement the motion and force transfer.



**Figure 2.8** TELBOT system

Ji and Song (1998) developed a re-configurable platform manipulator system, as shown in Figure 2.9. This system contains a base module, an end-effector module, and limb modules. Both the base and end-effector modules have many assembly locations to connect the limbs as required. The limb module has three joints: (i) an actuated linear joint, (ii) a spherical joint, connected to the base module, and (iii) a hook joint, connected to the end-effector module.



**Figure 2.9** Re-configurable platform manipulator system

### 2.1.3 Concluding remarks

The architecture is the backbone of modular robot systems, as it inherently determines the modular robot configuration. The survey above has shown that there are variations of modular robot architectures proposed, often without any rational comparison and evaluation. The existing proposals of modular robot architecture were developed in an ad-hoc mode, as they were focused on the technical issues, such as integration of a sensor and a driver within a module and the design of modular connection. The lack of a rational evaluation approach to modular robot architecture occurs mainly because there is no objective criterion except for the general motivation that modular robot systems will provide another dimension to enhance manufacturing system adaptability.

The modular robot architecture based on the one-unit module concept (Nolfi and Floreano 2000, Bererton and Khosla 2001) is useful to the environment where a modular robot requires frequent self-reconfiguration operations. This architecture is generally unsuitable for manufacturing applications for the following reasons: (i) high loads are usually involved in machining manufacturing where each unit module can only sustain a small force and the connections between two unit modules are too slender to support large force transfers, (ii) each unit module is an independent mechatronic system, including the mechanical transmission, driver, sensor, and communication; a simple configuration, such as a bar with a certain length, needs many units, which incurs high costs, (iii) real-time controls require intensive calculations, as complex kinematics and dynamics are involved, (iv) it is difficult to achieve high accuracy, as there are many connections among the module units, and (v) it is impractical for an industrial environment to assemble such a robot configuration autonomously.

Another aspect missing in existing modular robot architecture is the lack of a methodology or a rational procedure that could be used to derive system architecture or the attributes of system architecture from the requirements of reconfigurable systems. At this point, it may be possible to apply those well-developed system design theories, such as Axiomatic Design Theory (Suh 1990), to the design of a modular robot architecture. It is further remarked that the focus of this thesis on the issue of modular robot architecture lies in the conceptual architecture (see also the previous discussion of objective 1).

## 2.2 Kinematic and Dynamic Modeling

Kinematic and dynamic modeling concerns the relationships among the robot kinematic and dynamic behavior and robotic structural parameters and joint variables. A *kinematic model* is a representation of the motion of robot manipulators without consideration of forces and the masses or moments of inertia, and a *dynamic model* is a representation of the relationship among the driving force or torque on the joint actuator, the external work load, and the mass and inertia. The dynamic model includes the kinematic model constraints. Both models are essential to design, simulation, and real-time control of robot manipulators (Goldenberg and Emami 1999). The kinematic model is a basis for the dynamic model. There are two tasks involved in kinematic and dynamic modeling: formulation of the governing equations and solutions techniques for these equations.

### 2.2.1 Kinematic and dynamic model formulation methods

The first step to represent the kinematic and dynamic model is to represent a manipulator system as an entity that can be mathematically manipulated. There are three ways of achieving this step.

A Lie group has a  $3 \times 3$  rotation matrix and a  $3 \times 1$  translation matrix for transferring the information from one component to other components, and this transformation is also called the *point coordinate transformation*. Brockett (1984) applied the theory of Lie groups to robot kinematics by introducing the product-of-exponential (POE) equations. The POE provides a uniform representation of different types of joint motions (Yang



1999).

Denavit and Hartenberg (1955) proposed a method, called the D-H method for short. In the *D-H method*, a  $4 \times 4$  homogeneous matrix was defined to describe the transfer of information from one component to others. The advantage of the D-H method is that a minimum number of parameters is obtained to describe the transformation, and thus the method will be computationally efficient (Asada and Slotine 1985). The D-H method has been widely used for different applications (Paul 1981, Halperin 1986, Benhabib et al. 1989, Kelmar and Khosla 1990, Corke 1996).

A method with a  $6 \times 6$  transformation matrix was proposed by Bottema and Roth (1979) and Duffy (1980). The method was built upon the screw theory and differential geometry.

There are two basic methods to derive the dynamic model: the recursive Newton-Euler algorithm and Lagrange's equations. In the recursive Newton-Euler approach, Newton's second law and Euler's equations are applied to each link, sequentially. The solution procedure consists of two recursions: (i) a forward recursion in which the velocities and accelerations of each link are propagated from the base to the end-effector, and (ii) in a backward recursion in which the forces and moments are propagated from the end-effector to the base. In the Lagrangian approach, the main task is to compute the Lagrangian,  $L=T-V$  (here  $T$  is the kinetic energy of the system and  $V$  is the potential energy of the system).

Automatic modeling refers to the computer-aided generation of kinematic and dynamic equations given a computer representation of the structure of a robot. Generally speaking, it is much easier to develop an automatic dynamic modeling system after the kinematic model is known than it is to develop an automatic kinematic modeling system in the first place.

### **2.2.2 Solution techniques**

Three methods are used to solve the kinematic and dynamic equations: symbolic, numerical, and symbolic-numerical.

The benefits of symbolic solution include: (i) symbolic solution can be compiled, (ii) it can identify the solution types existed in different manifolds, (iii) the solution could be executed repeatedly with different joint variables, (iv) many intermediate results can be reused, and (v) it can run significantly faster than the iterative solution techniques. However, symbolic solutions do not always exist. Even if symbolic solutions exist, the solution may be worked out manually.

A numerical solution generally pertains to the determination of design variables as the result of an iterative procedure. The most commonly used iterative methods are variations of either the Newton-Raphson, steepest descent, or conjugate gradient methods. Another iterative method known as the continuation method, it does not require a priori knowledge of an approximate solution, and it can find all possible solutions. However, the continuation method gives little or no information about how the

design parameters influence the solutions (Wampler et al. 1990). The most compelling advantage of iterative techniques is that any problem that can be represented as a finite set of equations can, in theory, be solved.

In a symbolic-numerical method, a part that is not possibly solved symbolically is solved numerically. The symbolic-numerical method has not been developed as universal as the numerical method, because the procedure of eliminating intermediate symbolic parameters depends largely on the configurations of robot manipulators.

### **2.2.3 Modeling of modular robot configuration**

Most studies were conducted on robot configurations with the fixed geometry. However, task-oriented modular configuration design requires evaluating a large number of feasible configurations. Therefore the requirement on computer-aided generation of kinematic and dynamic equations is more demanding.

The complexity of developing a system for the computer-aided generation of kinematic and dynamic equations depends on the generality of modular robot architecture. Chen and Yang (1998) considered a relatively general modular robot architecture, and they addressed the issue of the automatic kinematic and dynamic modeling for modular robot configurations based on their architecture using the POE method. Their approach falls into a so-called ‘direct modeling’. By the direct modeling, it is meant that the kinematic and dynamic equations are generated directly from a suitable description of configurations. As such, their approach may not be applicable to other modular robot

architectures.

The other strategy is called the ‘indirect modeling’, which consists of two steps. The first step is to convert a description of the configuration of a particular type of modular robot architecture into the Denavit-Hartenberg (D-H) notation. The second step is to derive the kinematic and dynamic equations from the description based on the D-H notation. Halperin (1986) applied the indirect modeling method in developing a robot CAD system. Some other related studies can be found (Schmitz et al. 1989, Benhabib et al. 1989, Kelmar and Khosla 1990, Mulders et al. 1993). Benhabib et al. (1989) and Kelmar and Khosla (1990) addressed the issue of how to generate the kinematic D-H parameters from their description of modular robot configuration; however, the modular robot architecture they considered is relatively specialized and simple, which restricts the use of their approach to similar cases. It is to be noted that no study has been found on the derivation of the dynamic parameters from the description of modular robot configuration using the indirect modeling method.

#### **2.2.4 Further discussion and concluding remarks**

The discussion above has overviewed the study on kinematic and dynamic modeling for both modular robots and non-modular robots. It may be helpful to have a comparative overview of different studies on kinematic and dynamic modeling for modular robots that are the focus of this thesis. There are three aspects in which a comparison can be made, and they are: (i) the modular architecture, (ii) the completeness of design

variables, and (iii) the coverage of analysis contents.

From the previous discussion in Section 2.1, it can be concluded that the features of the architecture can be viewed from the following attributes: (i) multiple module ports, (ii) module adjustable variables, and (iii) module types (more details are given in Chapter 3). The levels of design variables are: (i) type level, (ii) number level, and (iii) dimension level (more details are given in Chapter 5). The coverage of analysis means: (i) the availability of automatic kinematic modeling (kinematics for short), (ii) the availability of automatic dynamic modeling (dynamics for short), and (iii) the change of design variables with pre-storing of the kinematic and dynamic equations (simulation for short). Table 2.1 gives such an overview of the existing studies in terms of the three aspects. From this table, one can conclude that (i) none of these works has considered all the features of the modular architectures of robots, (ii) none of these works has considered the variables describing the types of configurations of a robot, and (iii) most of the studies are on kinematics only.

**Table 2.1** Comparative overview of the existing studies

Features \ Literature			Halperin 1986	Kinya 1993	Schmitz et al. 1989	Benhabib et al. 1989	Kelmar and Khosla 1990	Mulders et al. 1993	Hooper and Tesar 1994	Han et al. 1997	Yang and Chen 1997	Chen and Yang 1998	Leger 1999
Modular architecture	Fixed geometry		x										x
	Joint	1-DOF		x	x	x	x	x	x	x	x	x	x
		Multiple DOFs			x	x	x		x		x	x	x
		Adjustable initial position											
		Multiple assembly patterns		x	x	x		x			x	x	
	Link	Multiple assembly patterns				x	x						
		Adjustable Dim.								x			
		Platform types							x				x
Level of design variables	Type												
	Number								x				x
	Dimension			x	x	x	x	x	x	x	x	x	
Coverage	Kinematics		x	x	x	x	x	x			x		
	Dynamics		x									x	
	Simulation								x				x

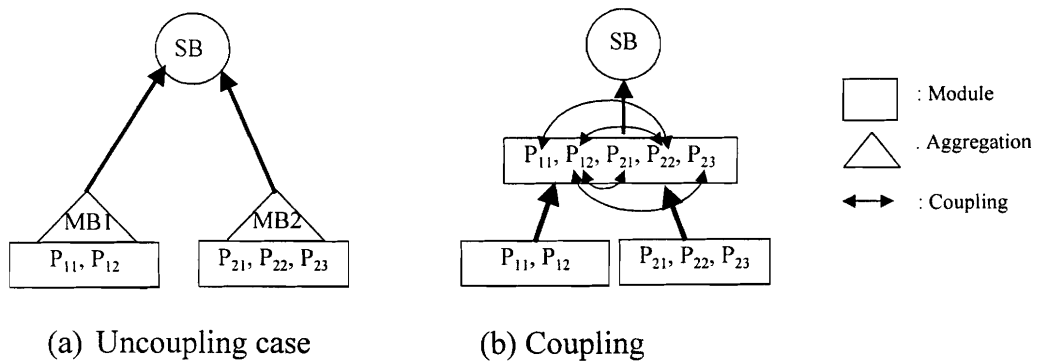
Note: x means availability

## 2.3 Modular Robot Configuration Synthesis

Modular robot configuration synthesis is a type of design activity in which a modular robot configuration is synthesized for a specified task from a set of pre-existed modules that can be combined only in certain ways.

Many methods have been developed for configuration synthesis of modular system, such as feature-based methods (Perremans 1996), modular-based methods (Tsai and Wang 1999), entity-based methods (Hong and Hong 1998), and case-based methods (Watson 1999). These methods are widely used in computer systems, modular fixtures, electronic

systems, and construction systems. The hidden assumption in these methods is that the modular system obeys the independence principle of Axiom Design Theory (Suh 1990, 1998); that is to say, each physical module satisfies only one functional requirement. The validity of this assumption lies in the definition of modular system architecture, i.e., the architecture that allows a meaningful uncoupling of modular parameters, see Figure 2.10a. In Figure 2.10a, MB1 (MB2) depends on  $P_{11}$ ,  $P_{12}$  ( $P_{21}$ ,  $P_{22}$ ,  $P_{23}$ ), and SB further depends on MB1, MB2. A close examination of modular robot architecture shows that the modular robot architecture exhibits a strong coupling of modular parameters, see Figure 2.10b. In Figure 2.10b, it can be seen that there is no layer of modular behavior, as exhibited in Figure 2.10a, and consequently system behavior (e.g., kinematic behavior) directly depend on modular parameters and their couplings. Therefore, the existing design methodologies for modular systems other than modular robot systems are not applicable to modular robot configuration synthesis.



P: parameter, MB: Modular Behavior, SB: System Behavior

**Figure 2.10** Two types of modular system architecture from a viewpoint of parameters coupling/uncoupling

Paredis (1996) classified modular robot configuration synthesis as an ‘innovative’ design problem, as the design space is pre-known, yet there is insufficient design

knowledge available. Two issues should be addressed to conduct modular robot configuration synthesis: (i) a design problem model, including the definition of design variables, design constraints, design objectives, and combination of these into a mathematical problem, and (ii) the strategy and method for solving the design problem model.

### **2.3.1 Design variables**

Design variables are defined from a given modular robot architecture. For example, if a given modular robot architecture provides adjustable parameters within a module, there is a need to define design variables to represent adjustable parameters. Another example is that if a designer wants to know whether a serial or a hybrid robot is better, in a design problem model, the model needs to include the variables to represent different structures. At this point, design variables in non-modular robot design are continuous and often divided to kinematic, dynamic, and control variables. The design procedure is organized in a sequential way, and each phase has its design sub-objectives and constraints, correspondingly (Nnaji 1986).

In modular robot configuration design, most studies were restricted to some dedicated modular prototype systems. Paredis and Khosla (1991) used the D-H parameters for configuration design of their reconfigurable modular manipulator system. It should be noted that the D-H parameters are not the same as the modular parameters, so their approach is not applicable to the design requirement where a set of modules needs to be determined. Chocron and Bidaud (1997) decomposed a robot configuration into several



segments (each with a motion axis), and treated joint orientation, joint type and link length as the design variables for each segment. The variables were defined at the module-level. His approach was only conceptually described, and the justification with implementation remains to be seen. Han et al. (1997) considered a design with fixed topology of modular robot configuration and with only the length of link module as the design variables. Chen and Burdick (1995) developed a very comprehensive model for modular robot configuration based on the definition of architecture. The design variables in their model include: (i) type of joints, (ii) type of links, and (iii) the assembly patterns of links. In their extended work, both the number of the link module and joint module were also defined as design variables. They did not consider any continuous variables, such as the length of link modules, and the global position and orientation of a module robot, due to the particular architecture of modular robot used (Yang 1999). Leger (1999) presented the Parametric Module Configuration Graph (PMCG) to represent a robot structure. PMCG was argued to be applicable to both modular and non-modular configuration designs. It is, however, noted that PMGC does not separate system architecture from system configuration, and consequently different types of configurations are not represented by any design variables. In fact, there is no type synthesis involved. The nature of the work by Leger (1999) is to vary the dimensions of a robot given a type of robot configuration, i.e., in other words, dimension synthesis.

### **2.3.2 Design constraints**

Design constraints are determined by two factors, namely, the domains for design variables and the task specification. In non-modular robot configuration design, the

domain of a design variable is assumed by designers, while in modular robot configuration design, the domain of a design variable, e.g., the type of link, etc., is determined by a modular robot architecture concerned. The task specification sets out the design requirements for a specific-purpose robot or a modular robot configuration. Design requirements include design constraints, such as the geometry of workspace, joint torque capabilities, obstacle avoidance, and operation time. Many studies on task-oriented robot design used a simplified task specification; i.e., a task is often defined as a sequence of points to be reached by the end-effector – without the prescription of time and behavior in-between points (Paredis and Khosla 1993b, Chen 1994, Han et al. 1997, and Yang 1999). Without prescription of time information, it is impossible to consider dynamic task requirements (Paredis 1996). Bi and Zhang (2001b) considered the dynamic task requirements by specifying desired velocities and accelerations at each of the given task points, but the time between any two points was not prescribed. Paredis (1996) tackled ‘fault tolerance’ as a kind of task specification, in addition to kinematic and dynamic constraints as previously discussed.

### **2.3.3 Design objectives**

For a general-purpose robot design, some overall performance measures, such as workspace and manipulability, were considered as design objectives (Angeles 1997, Bi et al. 1997). Freudenstein and Primrose (1984) designed a 3-DOF robot with the objective of obtaining the maximum workspace using some dedicated equations for calculating the workspace. The manipulability was used as a measure of the ability of

the end-effector to move in arbitrary directions by Yoshikawa (1985), and widely adopted as an objective in robot design.

When the task specification is known a priori, it is not important whether an arbitrary change in the end-effector position or orientation can be achieved except that the change in position/orientation required by the task. Therefore, manipulability and other dexterity measures are not very meaningful in case of task-oriented design of modular robot configurations.

Because the number of modules is directly related to the cost of a robot configuration, Yang and Chen (2000) introduced the concept of minimized degree-of-freedom and a weighted sum of the numbers of different types of modules was chosen as a design objective. Paredis (1996) used energy consumption as a design objective.

#### **2.3.4 Formulation of a design problem model**

Non-modular robot system design may involve a large number of design variables. For example, a serial manipulator with 6-DOF has 18 geometric variables, 60 mass variables, 42 rigidity variables and over 12 end-effector variables (Nnaji 1986). A general-purpose robot design problem has to be sequentially decomposed into kinematic, dynamic, and control problems (Nnaji 1986). However in some simple cases of specific-purpose robot designs, it is desirable to consider the kinematic and dynamic synthesis, as well as the optimal actuation and stress analysis in an integrated design process. Shakeri (1998) developed a knowledge-based method for such an integrated process to facilitate

designers from different disciplines in cooperating each other efficiently. Sims (1994) used an integrated method in design of mobile robots, which consist of block and rotational joints. For high-performance machines, Park and Asada (1994) used a concurrent design method in design of the mechanical structure and control of a two-link high-speed robot to achieve minimum settling time. Rasteger et al. (1999) proposed to simultaneously consider kinematic, dynamic, and control issues for design of 2-DOF robot based on the trajectory pattern method; the design objective was to minimize the higher harmonic portions of the actuating torques that are required to perform a class of motion patterns.

Previous efforts on modular robot configuration synthesis also followed the sequential design procedure; moreover, most of them considered the kinematic design only. Paredis and Khosla (1993b) described a general flow-chart of a selection program for modular robots, which covered three phases: kinematics, dynamics and sensor-based control. Furthermore, in their scheme, the kinematic design played a dominant role in the sense that the other two phases contribute to the modification of the result derived from the kinematic design. Fryer et al. (1997) proposed that object-oriented concepts could provide a useful tool for verifying the configuration of modular robot systems. They modelled robot resources and considered how the models could be adapted through the introduction of semantic annotations to accommodate the configuration process. Chen and Burdick (1995), Yang (1999) presented an approach for synthesis of task-oriented modular robot configuration with the minimum number of DOFs, which was essentially a sequential approach.

Paredis (1996) was the first to realize the importance of concurrent design in ‘fault tolerance’ modular robot design. He observed that if the sub-problems, such as kinematic design, dynamic design, trajectory planning, and control are tackled individually and sequentially, an optimal solution at one stage might not be optimal anymore at next stages. The globally optimal solution or even a feasible solution might never be found because of its possible sub-optimality at an intermediate stage. The limitations of his work include (i) a systematic analysis was not provided for the concurrent method, (ii) the design only considered a special task, to achieve ‘fault tolerance’ by increasing the number of joint modules and changing the assembly patterns, (iii) design variables were tackled simultaneously, which may lead to a large global design space and computationally expensive search of design solution. Leger (1999) also considered the kinematic and dynamic constraints, simultaneously in an automated synthesis approach to robot configuration design. In his approach, the kinematic and dynamic equations were pre-stored for a set of limited types of robots. The user selected a type of robots from the set, and the program (Leger 1999) calculated the optimal dimensions for the selected robot type. This method is not appropriate in modular robot configurations, as a large number of configuration candidates should be evaluated, and the storage of the equations for all these configurations is impractical.

### **2.3.5 Design synthesis**

The studies developed for complex systems, such as automobile and airplanes (Altus et al. 1996, Campbell et al. 1999) could be extended and used in modular robot configuration synthesis. Evolutionary algorithms or Genetic Algorithms (GA) were

widely used in the automatic synthesizing of modular configurations (Chen 1994, Han et al. 1997, Leger 1999, Campbell et al. 1999, Bi and Zhang 2001b). There are two reasons that GA are efficient. *First*, the problem with mixed discrete/continuous variables can be easily handled. *Second*, it is theoretically and empirically proven that GA can provide the robust search in complex spaces to find nearly global optimal solutions (Ramachandran and Chen 2000). The simulated annealing algorithm is another option for solving this kind of optimization problem.

The concurrent design approach will increase the problem dimension, and thus the computation is highly demanding. There were two approaches for coping with this problem: parallel computation and solution based on space reduction. In parallel computation technique aspect, Sims (1994) implemented a modified genetic algorithm to run in parallel based on a master/slave model. Paredis (1996) took a multi-agent paradigm to modify the genetic algorithm. The resulting algorithm can be executed in a distributed fashion on shared memory multi-processors. Ramachandran and Chen (2000) integrated an agent-based method with simulated annealing for modular configuration design. The principle of the space reduction technique is to reject poor candidate solutions as early as possible. Chen (1994) introduced structural evaluation besides task evaluation, that is, prior to evaluating how well a candidate manipulator design can fulfil the given task, evaluate the manipulator structure itself to exclude those structures with degenerated structures (e.g., structures with coincident rotation axes). Paredis (1996) separated the verification of design constraints into several phases: Cartesian paths, joint capabilities, and fault tolerant trajectories. The violation of solutions to the design constraints at a previous phase could be detected and thus removed from any further

computation.

### **2.3.6 Concluding remarks**

Relative to design modelling, the prior work has not provided a general model due to the limitation of the modular robot architecture employed. In most cases, a design model was only for dimension synthesis. The heterogeneous nature of modular robot configuration design, i.e., a mixture of type synthesis, number synthesis, and dimension synthesis, is yet to be addressed.

With regard to the formulation of a design model, the concurrent design approach was proposed, but the needs of this approach to modular robot configuration synthesis have yet to be elaborated. When a concurrent design problem model is formulated, the dimension of a design problem is likely very large, which calls for techniques to reduce the computational overhead.

## **Chapter 3**

### **Modular Robot Architecture, Configuration and its Representation**

The system architecture determines the system configuration variations, as the architecture specifies primary building blocks and the types of ways they are connected. In order to make systems more adaptive, configuration variations are expected to be as many as possible subject to the constraints of manufacturing applications (e.g. machining). Moreover, the architecture of a modular robot system should also consider manufacturing environments, in particular, such issues as the facilitation of assembling modules, the system stiffness, and the cost of a modular system. The goal of this chapter is to present a new architecture for modular robot systems with a balanced consideration of many configuration variations and their suitability of each for manufacturing applications. The cost of modular components is not a concern of this thesis, nor is the performance.

In Section 3.1, a review of the structures of industrial robots is presented to show what variations of robot configurations actually exist in the manufacturing industry. Section 3.2 proposes a functional requirement model for deriving the architecture of modular robot systems aimed at manufacturing applications. In Sections 3.3, a new architecture of modular robot systems is presented by applying Axiomatic Design Theory (ADT). In Section 3.4, a justification of the new modular architecture is provided. In Section 3.5,



the modular robot configuration and its representation are discussed, and a new representation method is presented. Section 3.6 provides a summary.

### 3.1 Robot Variations in Manufacturing Environment

The boundary between robots and other mechatronic systems appears to be disappearing. Many single-purpose machines, called hard automation, have features that resemble robots. Therefore, the definition of robots, which is used throughout this thesis, is given in the following:

**Definition 3.1 (Robot):** A robot is an automatically controllable, re-programmable, multipurpose, and manipulative machine with axes, which may be either fixed or mobile.

A robot is driven by electric-drive actuators, pneumatic devices, or hydraulic actuators. Actuators are applied to joints. A joint with an actuator is called *active* joint; otherwise, it is a *passive* joint. There are two motion types of joints: rotary (R) and linear (L). From the viewpoint of topology, robots are classified as serial robots, parallel robots and hybrid robots. Their definitions are as follows:

**Definition 3.2 (Serial Robot):** A serial robot is composed of a single open-loop kinematic chain connected from a base to an end-effector.

**Definition 3.3 (Parallel Robot):** A parallel robot is composed of two or more closed-loop kinematic chains in which the end-effector (mobile platform) is connected to the base platform by at least two independent kinematic chains. Between the base and end-effector platforms are serial chains (called *limbs*), which are symmetrically arranged.

Note that a *base platform* is a link fixed to the ground and connected to the limbs, and an *end-effector platform* is a link connected to limbs and the end-effector.

**Definition 3.3 (Hybrid Robot):** A hybrid robot is a combination of open-loop and closed-loop kinematic chains (Tonshoff and Grendel 1999).

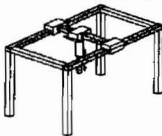
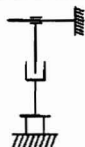
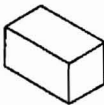
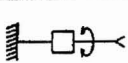

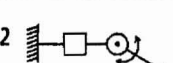
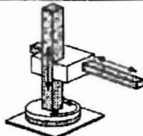


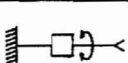

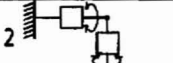

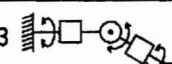
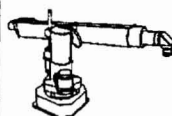


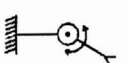
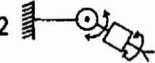
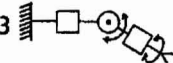



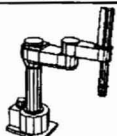
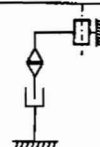

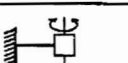

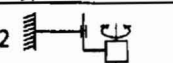
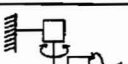




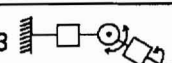

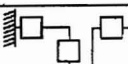

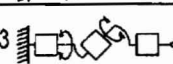
It should be noted that a serial kinematic chain is the basic sub-system in all types of robots. Theoretically, numerous types of serial kinematic chains could be formed by selecting a R or L actuator for each joint, by assembling two joints with a link body in a perpendicular or parallel mode, and by specifying dimensions of linkages between two joints. Taking a 6-DOF serial robot structure as an example, there are  $2^6=64$  possible combinations of joints types with either the R or L type and  $4^5=1024$  possible combinations of assemblies (perpendicular/parallel and intersection/non-intersection), an infinite number of variations of dimensions of the linkages among the joint axes, and several types of bases and the end-effectors.

### 3.1.1 Serial robots

In general, a serial robot has an actuator at each joint axis. Whilst having good operating characteristics, such as large workspace, high flexibility and manipulability, typical serial robots have the disadvantages of low precision, low stiffness, and low capacity. Additionally, they are generally operated at low speeds to avoid excessive vibration and deflection.

Although there are thousands of serial kinematic chains, the number of commercially available serial robots, which are designed optimally for specific industrial environments, is very limited. Figure 3.1 shows five of the most commonly used types of serial robots. The first column shows the positional structure which makes the robot move to any required position within the workspace. The second column is a diagram of their corresponding kinematic chains. The third column demonstrates the shapes of their workspaces. The remaining columns show the options of wrist structures to make an end-effector oriented to a specified direction. In summary, the following is a list of types of components in the industrial robots, shown in Figure 3.1.

- (i) active joints with 1-DOF: R and L,
- (ii) composite joints with 2-DOF or 3-DOF: RR, RRR,
- (iii) end-effectors: various types of grippers, and
- (iv) linkages with different lengths and orientations.

Robot	Axes		Wrist (DOF)		
Principle	Kinematic Chain	Workspace			
 cartesian robot			1 	1 	2 
 cylindrical robot			1 	1 	2 
			2 	3 	
 spherical robot			1 	2 	3 
			3 	3 	3 
 SCARA robot			1 	2 	2 
			2 		
 articulated robot			2 	3 	3 
			3 	3 	3 

**Figure 3.1** Five most common types of industrial robots (Warnecke et al. 1999)

### 3.1.2 Parallel robots

Parallel robots have many advantages, such as high precision, high loading capacity, high rigidity, and high speed, but they also have some disadvantages, such as small workspace, more singular configurations than serial robots, and complex kinematic and dynamic characteristics that make their control difficult. A comparison of parallel robots with serial robots is given in Table 3.1.

**Table 3.1** Comparison of parallel robots with serial robots (Uchiyama 1994)

Item to compare	Parallel robots	Serial robots
<i>Workspace</i>	Small	Large
<i>Forward kinematics</i>	Difficult	Easy
<i>Inverse Kinematics</i>	Easy	Difficult
<i>Forward statics</i>	Easy	Difficult
<i>Inverse statics</i>	Difficult	Easy
<i>Position error</i>	Averages	Accumulates
<i>Force error</i>	Accumulates	Averages
<i>Maximum force</i>	Summation of all actuator forces	Limited by minimum Actuator force
<i>Rigidity</i>	High	Low
<i>Dynamics</i>	Complex	Simple
<i>Inertia</i>	Small	Large

According to the definition of parallel robots, variations of parallel robots must be viewed from three aspects: the structure of limbs, the assembly of limbs, and how limbs connect the base and end-effector platforms.

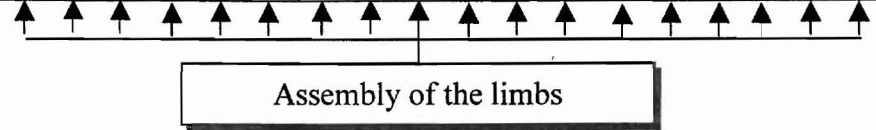
### **The structure of limbs**

A limb is also called a positional element in a parallel robot. The structure of a limb is the same as the structure of a serial kinematic chain, and therefore the variations of the structure are the same as those of a serial robot. Parallel robots differ from serial robots in that passive joints may be included in the parallel robots because the closed-loop chains in parallel robots introduce some kinematic constraints, and make a part of DOFs dependent on others.

## The assembly of limbs

The number of limbs required for a definite motion of the end-effector depends on the number of active joints in a limb. Figure 3.2 shows the relationship between the DOFs of the end-effector and the active joints in a limb.

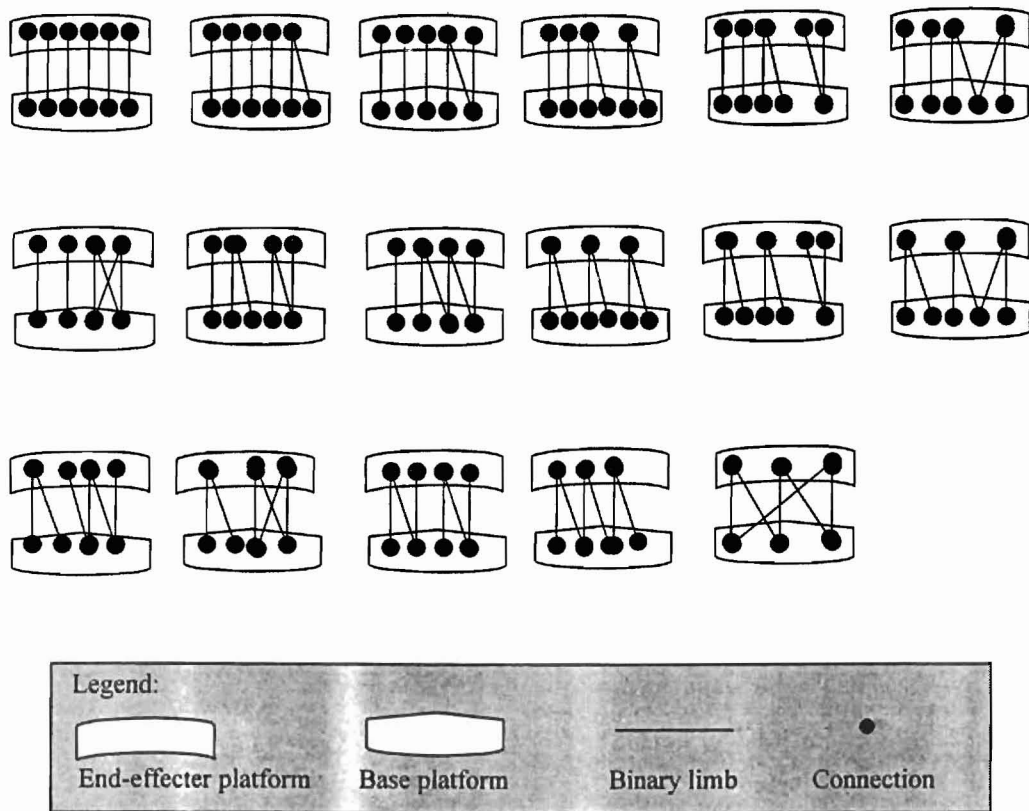
		Number of DOFs for parallel robots																	
		6						5					4				3		
Number of active DOFs for limbs	1	0	1	2	3	4	6	0	1	2	3	5	0	1	2	4	0	1	3
	2	3	1	2	0	1	0	1	2	0	1	0	2	0	1	0	0	1	0
	3	0	1	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0	0



**Figure 3.2** Variations of parallel robots by assembly of limbs (Pritschow 1999)

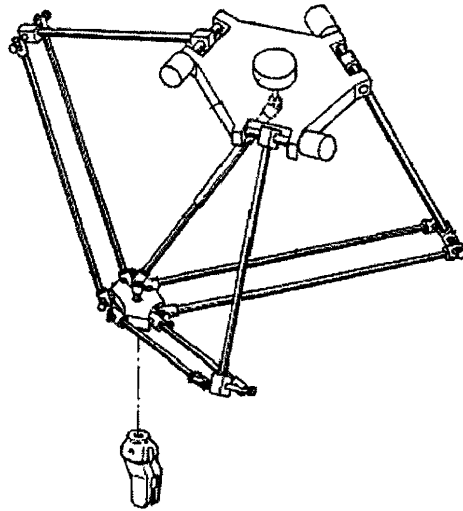
### *The connection the end-effector with base platforms*

Take a 6-DOF parallel robot as an example, where limbs are of the binary type, it has six limbs. Regarding the connection between the end-effector and the base platform, two limbs may share their connection parts at the base platform and/or the end-effector. A total of 17 variations of parallel robots could be produced by selecting locations of limbs with which to connect the end-effector and the base platform, see Figure 3.3 (Castelli 1999).



**Figure 3.3** Variations of parallel robots by connections between limbs and platforms

Most parallel robots are designed for applications in the machine tool industry. The Delta robot (Bonev 2000) is one of the most successful applications of parallel robots in machining; see Figure 3.4. The Delta system is designed to have the following characteristics: (i) the limbs are constructed as a parallelogram; (ii) the actuators are mounted on the base platform to make the end-effector achieve high acceleration.



**Figure 3.4** Example of parallel robots: Delta robot

It can be concluded that the following basic components, in addition to those for a serial robot, are needed for a parallel robot.

- (i) Passive joints with 1, 2, or 3 DOFs, such as R, L, RR, RL, LR, RRR, and
- (ii) Base platform and end-effector platform with different dimensions.

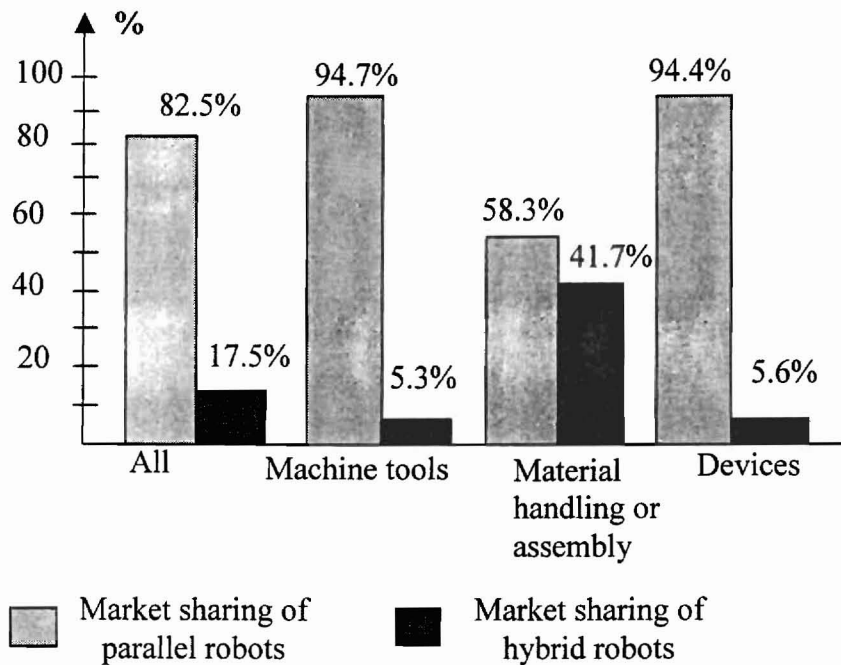
### 3.1.3 Hybrid robots

Hybrid robots are a compromise between serial and parallel robots as shown in Table 3.1.

Tonshoff and Grendel (1999) compared the applications of parallel robots and hybrid robots in three fields: machine tools, material handling and assembly machines, and devices. Figure 3.5 shows the industrial acceptance of the trade-off between the serial

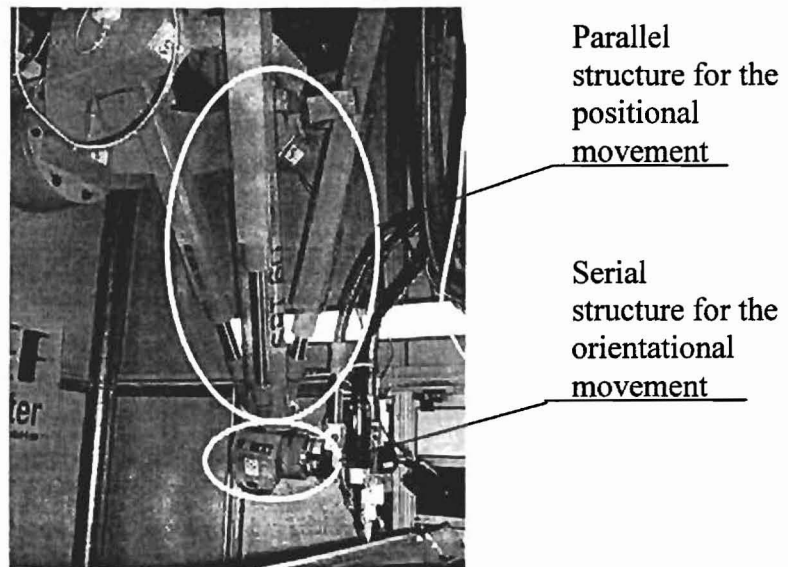


and parallel robots. It is interesting to note that hybrid robots are mostly used for the material handling applications that do not require high loading capacity or high speeds.



**Figure 3.5** Comparison of parallel and hybrid robots (Tonshoff and Grendel 1999)

Consider the Tricript hybrid robot (Figure 3.6), in which the parallel structure consists of 3 limbs functions as positioning. The serial structure consists of 3 rotary joints; it is built upon a parallel structure that makes orientational movement. It is designed for precision assembly and heavy material handling, and has proven to be outstanding for light milling and drilling applications. Its wrist has high dexterity; the ratio of the working space over the machine size could be 1/3 greater than parallel mechanisms, and its stiffness is in the range of 6-30 N/ $\mu$ m.



**Figure 3.6** Example of hybrid robots (NEOS 2001)

A hybrid robot involves a link that is connected with more than other two links; such a link is called a *hybrid link*.

### **3.2 Functional Requirement for Modular Robot Architecture**

To define the modular robot architecture, a general design theory called *Axiomatic Design Theory* (ADT) (Suh, 1990) is applied. According to ADT, the functional requirements should be defined, then a set of design parameters, which determine effects on functions, is defined, and finally the modular robot architecture is defined by a set of features.

The architecture of modular robot systems should support the creation of modular robot configurations to meet a set of functional requirements and possible manufacturing applications. *Functional Requirements* (FR) are those that produce configuration

variations. Based on the previous discussion on various robot configurations in manufacturing environments, the FRs for deriving the architecture of a modular robot system are presented as follows:

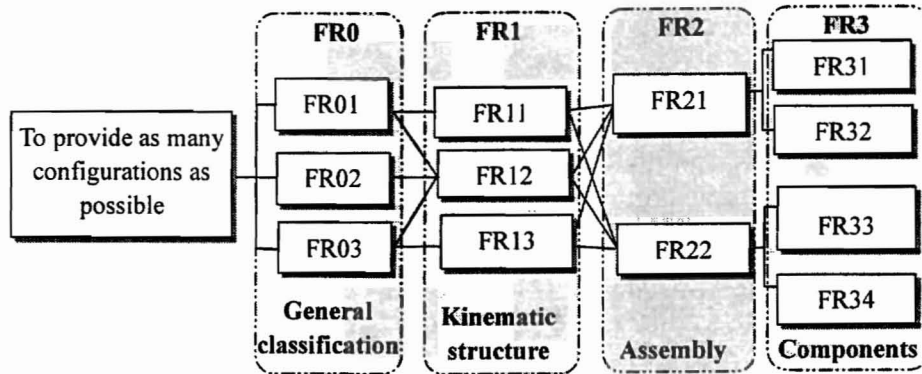
**FR0:** *To produce configurations of serial, hybrid, and parallel robots.*

**FR1:** *To enable variations in terms of kinematic chain structures, including serial robot structures, parallel robot structures, and hybrid robot structures.*

**FR2:** *To enable variations of assemblies for serial robots in terms of (i) the number of basic motion types (R or L), and (ii) permutation of the motion types; for parallel and hybrid robots, in terms of (i) permutation of multiple inputs and outputs, and (ii) different connections between parallel or hybrid structures and limbs.*

**FR3:** *To enable variations of basic components with specific physical properties, including (i) basic 1-DOF motion (R or L) with different properties, (ii) multiple DOFs joints; and (iii) various types of connections with special orientations and locations.*

It is noted that multi-DOF motions may not be assembled from a set of 1-DOF motion joints. This justifies the need for an integral multiple DOF joint module, i.e., FR32.



—— : means relationship (e.g., FR01 of FR0 has a FR11 and a FR 12 in FR1, etc.)

FR01: To provide hybrid robots	FR11: To provide hybrid structure	FR21: To provide different permutation of motions	FR31: To provide variations of basic motions (1-DOF)
FR02: To provide serial robots	FR12: To provide serial structure	FR22: To provide connection variations	FR32: To provide variations of combined motions (DOFs)
FR03: To provide parallel robots	FR13: To provide parallel structure		FR33: To provide variations of orientations
			FR34: To provide variations of spatial dimensions

**Figure 3.7** Functional requirements (FRs) for a modular robot architecture

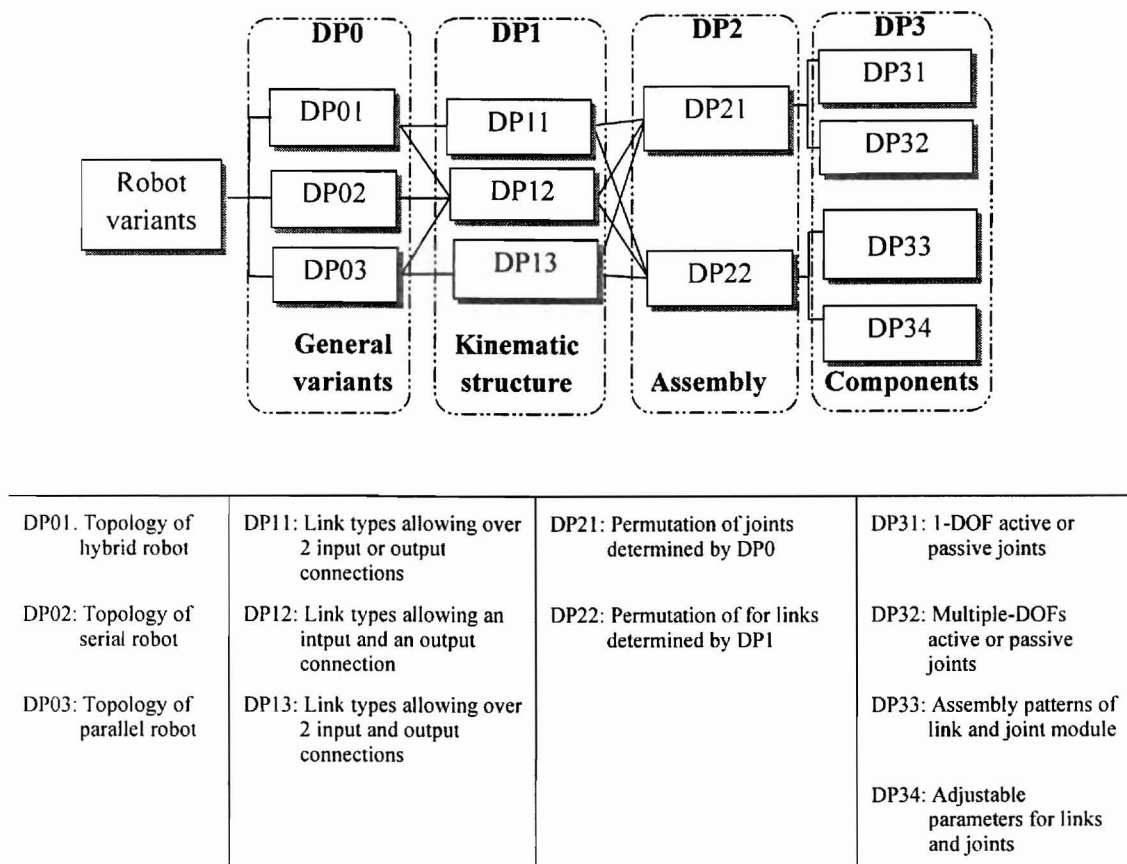
### 3.3 General Architecture of Modular Robots (GAMR)

#### 3.3.1 Design parameters and functional requirements

The general architecture of modular robots (GAMR) should be defined to meet all of the FRs. According to ADT, a set of design parameters (DPs) of a GAMR must be identified;

these design parameters must satisfy the FRs. To develop the GAMR, design parameters (DPs) are further used to specify the features.

Figure 3.8 shows the design parameters of the GAMR, presented hierarchically. DPs, shown in Figure 3.8, must correspond to the FRs (Figure 3.7). The meanings of these DPs are also illustrated in Figure 3.8. Taking DP11 as an example; DP11 stands for the types of links, which in particular has more than two connection ports; such types of links are needed in order to build a hybrid robot, which is FR1 (Figure 3.7).



**Figure 3.8** Design parameters for robot variations

### 3.3.2 Definition of GAMR

By summarizing all DPs, the following definition of a GAMR is given by a list of features.

**Feature 1:** A modular robot system consists of a set of joint modules and link modules. These modules have different types, which cover those in existing modular robot systems. Link modules also include the base and end-effector platforms, and joint modules include gripper modules and wrist modules.

**Feature 2:** Both joint and the link modules can have multiple ports through which a connection can be made, and the connection can take place between any two modules (joint/joint, link/joint, or link/link). Physically, ports have dimensions, and the ports of two different modules with identical dimensions can be connected.

**Feature 3:** There are two special types of link modules that are designed to connect with more than 2 modules: *hybrid link module* and *platform module*. A platform module is either a base (base platform) or an end-effector (end-effector platform) in a parallel robot, while a hybrid link module is a connection among the joint modules to form a hybrid robot.

**Feature 4:** Joint modules are further classified into the active and passive modules. An active joint module has actuator(s), while a passive module does not have any actuators.

**Feature 5:** Adjustable variables are included within some modules. These include (i) adjustable geometric dimensions of a link module, and (ii) the adjustable initial position in an active joint module.

Table 3.2 shows the correlation between the DPs and the features, where “x” means that a feature has a contribution to the corresponding DP.

**Table 3.2** The relationship between DPs and GAMR features

DP	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
DP01	x	x	x	x	
DP02	x		x		
DP03	x	x	x	x	
DP11	x		x		
DP12	x		x		
DP13	x		x		
DP21	x			x	
DP22	x		x	x	
DP31	x			x	
DP32	x			x	
DP33	x	x	x		x
DP34	x				x

### 3.4 Justification of GAMR

The generality of the GAMR can be demonstrated using Table 3.3. In Table 3.3, the left two columns show all the features of the GAMR. The remaining columns show other architectures, where the features covered by these architectures are indicted by “x”.

**Table 3.3** A comparison of GAMR with others existing architectures

Features \ Others		Tesar and Butler (1989)	RMMS (Schmitz et al. 1989)	MODRO (Nielsen and Huppi 1992)	TOMMS (Matsuma 1995)	(Chen 1994 )	TELBOT (Walischmiller and Frager 1994)	(Prischow and Wurst 1996)	(Cohen et al. 1992)	(Chen 2001)	AMTEC (Jaenisch, et al. 2000)
Joint	Active	x	x	x	x	x	x	x	x	x	x
	Passive	x								x	
	Multiple DOFs	x		x				x	x	x	x
	Adjustable position										x
Link	Fixed Dim.	x	x			x	x	x		x	x
	Changeable Dim.			x	x				x		
	Hybird types										
	Platform types										
Link/Joint connection	Single pattern	x		x			x	x			
	Multiple pattern		x		x	x			x	x	x
Link/Link Connection	Single pattern									x	x
	Multiple pattern					x					
Joint/Joint Connection	Single pattern										
	Multiple pattern					x					

It can be seen that the GAMR has the most coverage of features, which implies its generality. It is also demonstrated that very few architectures include adjustable components.

## 3.5 Modular Robot Configuration and its Computer

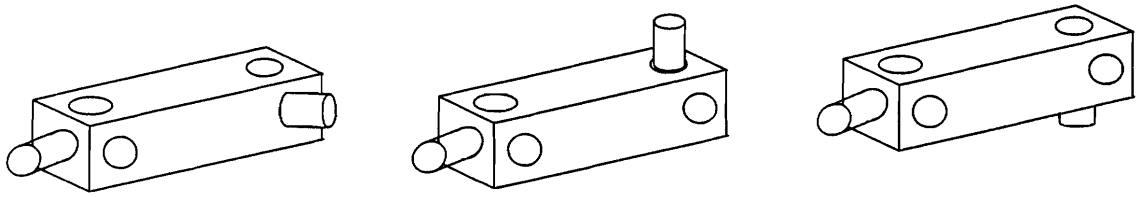
### Representation

#### 3.5.1 Configuration

A modular robot configuration is an assembly of a set of link and joint modules. Types of link modules and joint modules and their connectivity were defined in GAMR. Variations of configurations are attributed to both the module level and the assembly



level. At the module level, the different geometric and inertia properties, and capacities (e.g., torque and speed), contribute to different configurations. At the assembly level, variations of configurations result from the following factors: (i) assembly ports of each module, (ii) adjustable parameters of modules, (iii) number of link and joint modules, and (iv) connections of modules. It should be noted that the configurations may be isomorphic; in particular, at the assembly level, several assembly patterns of a link module (for example) are identical, see Figure 3.9. The isomorphic assembly patterns must be identified prior to configuration design, which considerably reduces the computational time. Chen (1994) developed a method to identify isomorphic assembly patterns; this method will be used in the present study.



**Figure 3.9** Isomorphic module assemblies

### 3.5.2 Computer representation

In order to create a mathematical model or computer model to perform modular robot configuration design, first of all the information about an individual configuration of modular robot needs to be structured into a representation that can be easily manipulated. A configuration can be viewed as a set of modules and a set of connections. Therefore, the representation of a configuration can be divided into the representation of modules and the representation of connections or assemblies. These two representations are

apparently related to each other; the representation of the connections will have references to the representation of modules, as modules form the connections.

Data modeling is a tool that can be used for developing the representations above, which consequently is a database model (Zhang 1994, Zhang and Liu 2000). The manipulation of a database model will need a database management system (DBMS). Although, from the viewpoint of computer integrated manufacturing, a database model is a core to communicate with other downstream manufacturing and assembly processes, the database model managed by a DBMS is not easily integrated with design process models, such as design analysis and design synthesis. An effective way is to create a data representation between the database and the design process. This data representation still needs to be divided into two parts: modules and assemblies. A basic idea for this representation is that the part for modules may follow a formalism called the object-based, while the part for assemblies follows the graph theory and its representation incident matrix in particular.

### **3.5.3 Computer representation of modules**

Object-based formalism represents the module information in the following form (taking the link and joint modules as examples):

**Entity Joint module**

Number of DOFs	: 1/2/3
Motion type of each DOF	: Linear/Rotary
Active attribute of each DOF	: Passive/active
Torque ranges	: Force/torque
Connectable module types	: Link and joint modules
Motion range for each DOF	: Displacement/Velocity/Acceleration
Adjustable parameter for each DOF	: Initial joint position
Isomorphic assembly pattern	: Number, input and output postures for each pattern
Dimensional parameters	: Length(s), width(s), height(s)
Dynamic parameters	: Mass(s), center of mass(s), inertial(s),

**End-Entity****Entity link module:**

Connectable module types	: Link and joint module
Isomorphic assembly pattern	: Number, input and output postures for each pattern
Fixed dimensions	: Displacement and orientation
Changeable dimensions	: Displacement or orientation
Dynamic parameters	: Mass(s), center of mass(s), inertial(s),

**End-Entity**

In the above, the left column contains attributes that define one aspect of the features of a particular module, while the right column contains the domains from which the attributes on the left column should take values.

3.5.4 Computer representation of assemblies

Several definitions from graph theory are given as follows:

**Definition 3.1** (Graph) a graph  $G=(V, E)$  consists of a vertex set,  $V(G)$ , and an edge set,  $E(G)$ , such that every edge in  $E(G)$  is associated with a pair of vertices in  $V(G)$ .

**Definition 3.2** (Labeled Graph) A labeled graph is a graph in which the vertices are labeled by  $v_1, v_2, v_3, \dots, v_m$  and the edges are labeled by  $e_1, e_2, e_3, \dots, e_n$ , such that  $V=\{ v_1, v_2, v_3, \dots, v_m \}$  and  $E= \{ e_1, e_2, e_3, \dots, e_n \}$ .

Examples of a graph and labeled graph are shown in Figure 3.10a, b, respectively. The graph has 6 vertices and 5 edges in Figure 3.10a, and the vertices and edges are labeled in Figure 3.10b.

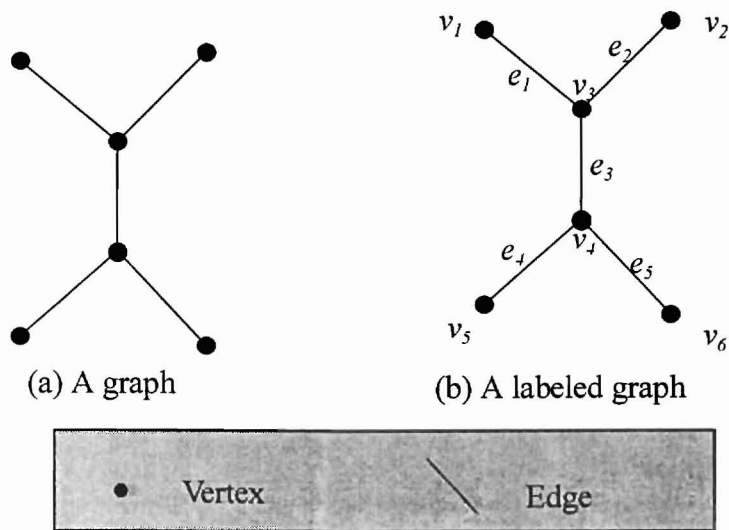


Figure 3.10 Examples of a graph and a labeled graph

**Definition 3.3** (Incidence matrix) Let  $G=(V, E)$  be a labeled graph, where  $V=\{ v_1, v_2, v_3, \dots, v_m \}$  and  $E= \{ e_1, e_2, e_3, \dots, e_n \}$ . The incidence matrix  $M(G)$  is an  $m \times n$  matrix in which the entry in row  $i$  and column  $j$  is 1 if edge  $e_j$  is incident on vertex  $v_i$ . Otherwise, it is 0 (Yang 1999).

Considering an example, the incidence matrix for Figure 3.10b could be obtained as follows:

$$M(G) = \begin{matrix} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

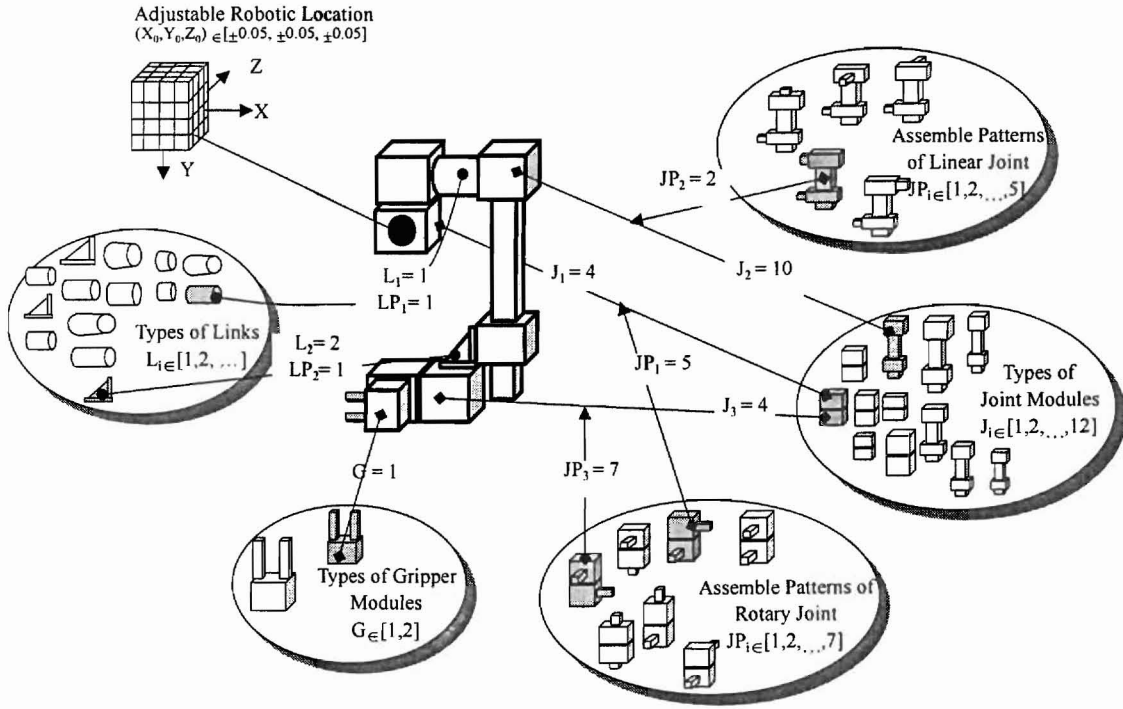
The incidence matrix cannot completely represent the physical properties of a vertex and an edge, nor the connectivity information between a vertex and an edge. For example, the incidence matrix cannot represent the following pieces of semantics: an edge (e.g., a joint) has 5 assembly ports (1, 2, 3, 4, 5), and port 1 is associated with a specific connection with a vertex (e.g., a link). An extended incidence matrix called the object incidence matrix (OIM), providing more detailed descriptions of vertices, edges and their connections and aiming to represent those pieces of semantics, is defined as follows:

**Definition 3.4.** (OIM) Object incidence matrix. Let  $G=(V, E)$  be a labeled graph, where

$S(V) = \{S(v_1), S(v_2), S(v_3), \dots, S(v_m)\}$ ,  $S(E) = \{S(e_1), S(e_2), S(e_3), \dots, S(e_n)\}$ , where  $S$  represents the structured object of links and joints. The incidence matrix  $M(S(G))$  is an  $m \times n$  matrix in which the  $i$ th column is  $S(v_i)$ , the  $j$ th row is  $S(e_j)$ , and the entry in row  $i$  (corresponding to  $S(v_i)$ ) and column  $j$  (corresponding to  $S(e_j)$ ) is  $\phi$  when there is no connection between  $S(v_i)$  and  $S(e_j)$  and a complex expression when  $S(v_i)$  and  $S(e_j)$  has a connection. This complex expression describes the detailed information of assembly between  $S(v_i)$  and  $S(e_j)$ . For example,  $S(v_1)$  refers to the detailed structure information of vertex 1, say link 1, which at the same time the subscript 1 means the first row of the incidence matrix. In the following discussion,  $S(v_i)$ ,  $S(e_j)$  and entries at row  $i$  and column  $j$  may be called *data objects*. The following example provides an illustration for OIM.

### 3.5.5 Example

Figure 3.11 shows an example of modular robot configuration based on the architecture of the AMTEC modular robot system (Jaenisch et al. 2000). The system consists of 12 types of joint modules and 14 types of link modules. A detailed description of the AMTEC system is provided in Appendix A.



- $J_i (i=1,2,3)$  : types of joints ( $J_1=4, J_2=10, J_3=4$ ), where  $i$  identifies a particular joint module in an assembly.
- $L_i (i=1,2)$  : types of links ( $L_1=1, L_2=2$ ), where  $i$  identifies a particular link module in an assembly.
- $G$  : type of gripper ( $G=1$ ).
- $JP_i (i=1,2,3)$  : assembly patterns for the joint modules ( $JP_1=5, JP_2=2, JP_3=7$ ).
- $LP (i=1,2)$  : the assembly patterns for the link modules ( $LP_1=1, LP_2=1$ ).

**Figure 3.11** Example of a 3 DOF configuration

For the configuration shown in Figure 3.11, the types of link and joint modules in this example modular robot (i.e., joint type 4, joint type 10, link type 1, link type 2, and gripper type 1) are data objects (see definition 3.4) and their detailed structural information is given in Table 3.4.

**Table 3.4** Data objects for joint and link modules

Joint Type 4	Joint type 10
One Degree of Freedom (active and rotary); The joint motion range is $[-0.95\pi, 0.95\pi]$ ; The joint torque: 18.7;	One Degree of Freedom (active and translational); The joint motion range is $[0, 0.35]$ ; The adjustable initial position $[0.0, 0.02]$ ;

<p>The connectable modules are the links with the size 70cm;  The number of isomorphic patterns is 7;  The input ports for the assembly patterns are  {[1,0,0,0;0,1,0,0;0,0,1,-0.070;0,0,0,1];  [1,0,0,0;0,1,0,0;0,0,1,-0.070;0,0,0,1];  [0,0,-1,0.035;1,0,0,0;0,-1,0,-0.035;0,0,0,1];  [1,0,0,0;0,1,0,0;0,0,1,-0.070;0,0,0,1];  [0,0,-1,0.035;1,0,0,0;0,-1,0,-0.035;0,0,0,1];  [0,0,-1,0.035;1,0,0,0;0,-1,0,-0.035;0,0,0,1];  [-1,0,0,0;0,0,-1,0.035;0,-1,0,-0.035;0,0,0,1]};  The output ports for the assembly patterns are  {[1,0,0,0;0,1,0,0;0,0,1,-0.070;0,0,0,1];  [0,0,1,0.035;1,0,0,0;0,1,0,0.035;0,0,0,1];  [1,0,0,0;0,1,0,0;0,0,1,-0.070;0,0,0,1];  [0,0,1,0.035;1,0,0,0;0,1,0,0.035;0,0,0,1];  [-1,0,0,0;0,0,1,0.035;0,1,0,0.035;0,0,0,1];  [0,0,-1,-0.035;-1,0,0,0;0,1,0,0.035;0,0,0,1];  [0,0,1,0.035;1,0,0,0;0,1,0,0.035;0,0,0,1] };  Two parts of masses are [0.9,0.9];  The centers of masses:  [0,0,-0.035]and [0,0,0.035];  The moments of inertia:  [0.05,0.05,0.05] and [0.05,0.05,0.05];  The joint size is '70cm';</p>	<p>The joint force: 500;  The connectable modules are the links with the size 70cm;  The number of isomorphic assembly patterns is 5;  The input ports for the assembly patterns are  {[1,0,0,0;0,1,0,0;0,0,1,-0.070;0,0,0,1];  [-1,0,0,0;0,0,-1,0.035;0,-1,0,-0.035;0,0,0,1];  [0,0,-1,0.035;1,0,0,0;0,-1,0,-0.035;0,0,0,1];  [1,0,0,0;0,0,1,-0.035;0,-1,0,-0.035;0,0,0,1];  0,0,1,-0.035;-1,0,0,0;0,-1,0,-0.035;0,0,0,1]};  The output ports for the assembly patterns are  {[-1,0,0,0;0,0,1,0.035;0,1,0,0.040;0,0,0,1];  [-1,0,0,0;0,0,1,0.035;0,1,0,0.040;0,0,0,1];  [-1,0,0,0;0,0,1,0.035;0,1,0,0.040;0,0,0,1];  [-1,0,0,0;0,0,1,0.035;0,1,0,0.040;0,0,0,1];  [-1,0,0,0;0,0,1,0.035;0,1,0,0.040;0,0,0,1]};  Two parts of masses are =[2.0,0.6];  The centers of masses:  [0,0,-0.035] and [0,0,0.040];  The moments of inertia:  [0.2,0.05,0.05] and [0.05,0.05,0.05];  The joint size is '70cm';</p>
<p><i>Link Type 1</i></p> <p>The connectable modules are the links or joints with the size 70cm;  The number of the isomorphic assembly patterns is 1;  The input port of the assembly pattern is  [1,0,0,0;0,1,0,0;0,0,1,0; 0,0,0,1];  The output port of the assembly pattern is  [1,0,0,0;0,1,0,0;0,0,1,0.090;0,0,0,1];  The link mass is 0.63;  The moments of inertia are [0.016, 0.016, 0.006];  The center of the mass is [0,0,0.045];  The link size is '70cm';</p>	<p><i>Link Type 2</i></p> <p>The connectable modules are the links or joints with the size 70cm;  The number of the isomorphic assembly patterns is 1;  The input port of the assembly pattern is  [1,0,0,0;0,1,0,0;0,0,1,0; 0,0,0,1]  The output port of the assembly pattern is  [1,0,0,0;0,0,1,0.035;0,-1,0,0.035;0,0,0,1];  The link mass is 0.54;  The moments of inertia are [0.015, 0.015, 0.01];  The center of the mass is [0.035,0,0.035];  The link size is '70cm';</p>
<p><i>Gripper Type 1</i></p> <p>The connectable modules are the links or joints with the size 70cm;  The number of the isomorphic assembly patterns is 1;  The input port of the assembly pattern is  [1,0,0,0;0,1,0,0;0,0,1,0; 0,0,0,1]  The output port of the assembly pattern is  [1,0,0,0;0,0,1,0,0,1,0,0.090;0,0,0,1];  The gripper mass is 1.1;  The moments of inertia are [0.2, 0.05, 0.05];  The center of the mass is [0, 0, -0.045];  The gripper size is '70cm';  The gripper stroke is 0.060;</p>	

In this configuration, there are 4 vertices (base, 2 links and gripper) and 3 edges (2 rotary joints and 1 linear joint). Vertices and edges are connected in a serial mode. The object incidence matrix from definition 3.4 is given as follows: (i)  $S(V)$  and  $S(E)$  are defined by two attributes  $\langle k, h \rangle$ , where  $k$  denotes the module class (e.g., base, link, gripper, etc.), and  $h$  denotes the module type in a module class (e.g., for the module class



of a link, the type identifiers are 1 and 2); (ii) the entry for row  $i$  and column  $j$  has the following format:  $\langle i_1, i_2, i_3, i_4 \rangle$ , where  $i_1$  denotes whether or not the corresponding vertex and edge is connected (1 for connected and otherwise 0),  $i_2$  denotes whether part A or B of a joint module (see Appendix A for details) is associated with the connection (0 for part A and 1 for part B),  $i_3$  denotes the assembly port on the vertex (link), and  $i_4$  denotes the assembly port on the edge (joint). Based on this discussion, the OIM for the modular robot shown in Figure 4.11 is given below:

$$\begin{array}{l}
 \begin{array}{l}
 \langle \text{base}, 1 \rangle \\
 \langle \text{link}, 1 \rangle \\
 \langle \text{link}, 2 \rangle \\
 \langle \text{gripper}, 1 \rangle
 \end{array}
 \begin{array}{l}
 \begin{array}{c}
 \langle \text{joint}, 4 \rangle \quad \langle \text{joint}, 10 \rangle \quad \langle \text{joint}, 4 \rangle \\
 \left( \begin{array}{ccc}
 \langle 1, 0, 1, 2 \rangle & \Phi & \Phi \\
 \langle 1, 1, 1, 8 \rangle & \langle 1, 0, 2, 2 \rangle & \Phi \\
 \Phi & \langle 1, 1, 1, 6 \rangle & \langle 1, 0, 2, 1 \rangle \\
 \Phi & \Phi & \langle 1, 1, 1, 7 \rangle
 \end{array} \right)
 \end{array}
 \end{array}
 \end{array}$$

In the above OIM, the entry in row  $i$  and column  $j$  has the following meanings: the first number “1” means that the first vertex (base) and the first edge (joint type 4) is connected, the second “0” means that part A of the first joint is connected to the base, the third “1” means the assembly port 1 on the base is used to connect the first joint, and the fourth “2” mean the assembly port 2 on the first joint is used to connect the base.

### 3.6 Summary

This chapter discussed the modular robot architecture, configuration and its representation. In particular, a general conceptual architecture of modular robot was proposed. Axiomatic Design Theory (ADT) moved a systematic process for developing

the GAMR. The GAMR was found to be the most general and provided a basis for subsequent development. The configurations of a modular robot were derived from its architecture. Separation of configuration and architecture is a key to achieve a systematic and complete method for computer-aided configuration design. Finally, an integrated representation of a modular robot configuration, the object incidence matrix (OIM), was developed.

## Chapter 4

### **Automated Kinematic and Dynamic Modeling of Modular Robot Configurations**

The design of a modular robot requires developing kinematic and dynamic models after the design variables are given. Computer-aided design and synthesis of modular robots requires automated generation of the kinematic and dynamic behavior. This chapter presents a new approach to achieve computer-aided generation of kinematic and dynamic behavior.

Many theories, methodologies and analysis tools are developed for modeling of non-modular robot structures. They cannot be adopted directly to modular robots for three reasons. *First*, descriptions of non-modular robots usually involve physical parameters, such as the Denavit-Hartenberg (D-H) parameters, screw parameters, and Product of Exponential (POE) parameters; while design variables for modular robot configurations include many discrete variables, such as the types of joint and link modules. *Second*, most previous research targeted non-modular robot configurations with fixed topology, but a large number of candidate robots with varying types of topology must be analyzed for modular robot configuration design. *Third*, for non-modular robot design, design variables were commonly decomposed into categories such as kinematic parameters, dynamic parameters, and control parameters. However,

for modular robot configurations, design variables simply represent a configuration based on a particular architecture. In fact, a change in one design variable may invoke the change of kinematic, dynamic, and control behaviors simultaneously.

This chapter is organized as follows. In Section 4.1, direct and indirect modeling methodologies are discussed. It is argued that indirect modeling is better and it is, therefore, adopted in our research. Section 4.2 presents the derivation of an architecture for the D-H notation from the OIM representation of a modular robot configuration. In Section 4.3 and Section 4.4, we present the kinematic and dynamic conversions from the OIM representation to the D-H representation of a modular robot configuration, respectively, a key step to implement indirect modeling methodology. Section 4.5 presents a computer program to implement an indirect modeling method. In Section 4.6 we validate the indirect modeling methodology through experiments and Section 4.7 provides a summary.

## 4.1 Modeling Methodologies

In modular robot configurations, design variables represent configurations for certain purposes, e.g., kinematic trajectory tracking, reduction of torques in actuators. Once the design variables are determined, two alternative ways can be employed to compute kinematic and dynamic behaviors, direct modeling and indirect modeling. *Direct modeling* establishes a dedicated procedure, directly from modular robot design variables, to compute the kinematic and dynamic behaviors. The disadvantages of the direct modeling are as follows. *First*, such a procedure only suits a particular definition

of modular robot architecture. *Second*, the cost of developing procedure is too large. The *indirect modeling* has two steps; the first step is to convert modular robot design variables to a description that views modular robots as a non-modular robot and follow a particular formalism for non-modular robots, such as the D-H notation; the second step is to compute the kinematic and dynamic behavior based on a dedicated procedure related to the formalism for non-modular robots. This modeling methodology has an advantage that the second step can be utilize computational methods for non-modular robots.

The indirect modeling method decomposes a complex problem into a set of relatively simpler problems, and a total solution to the problem is the aggregation of individual solutions to these simpler problems. This approach not only improves the flexibility of the method but also the robustness of the solution. In this research, the indirect modeling method is adopted.

A configuration is obtained from its underlying architecture. Therefore, design variables depend on a particular architecture. A modeling methodology developed for a more general architecture is applicable to a more specific architecture. In this thesis, the GAMR, described in Chapter 3, is chosen because of its generality.

## **4.2 From the GAMR to the D-H Architecture**

In Chapter 3, the GAMR architecture for modular robots was developed. Modular robot configurations are generated based on the GAMR. The indirect modeling method for

computing the kinematic and dynamic behaviors of a modular robot needs the conversion from the GAMR to the D-H architecture. It is noted that this thesis adopts the definition of the D-H architecture given by Paul (1981). This thesis uses the D-H architecture interchangeable with the D-H notation, though the former is with more emphasis on the definition of coordinate systems on the joint and link, while the latter on the notational convention.

#### **4.2.1 Joints and links from a viewpoint of the D-H architecture**

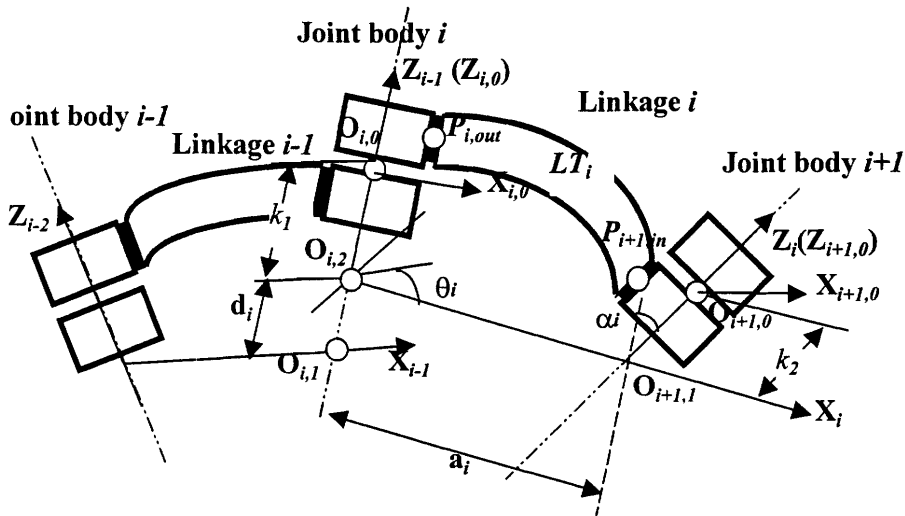
A robot configuration has one base linkage, which is connected to the ground or a part of the ground, and one or several grippers as an end-effector to hold the work piece. The chain, which relates the base to a gripper through a series of motion axes and intermediate linkages, is called the *main kinematic chain*; other chains which may start or end at the intermediate linkage of a main chain are called the *branch chains*. Both a main chain and a branch chain are formed in series and correspond to a group of D-H parameters. A serial robot has only one main kinematic chain; while a hybrid or parallel robot has a main chain and some branch chains.

As shown in Figure 4.1, D-H parameters are defined from the motion axes and the linkages, called the *D-H architecture*. To apply the indirect method, the robot modules defined in Chapter 3 must be viewed from the D-H architecture. For this purpose, joint modules are viewed as follows: (i) each motion axis corresponds to a joint body where *the joint body* consists of two components that could translate or rotate relative to each other; (ii) each connection between two motion axes corresponds to a linkage body

where *the linkage body* has ports at its ends through which to connect with joint bodies;

(iii) both a linkage body and a joint body are physical objects with physical properties as dimensions, assembly ports, masses, center of mass, and moment and product of inertias.

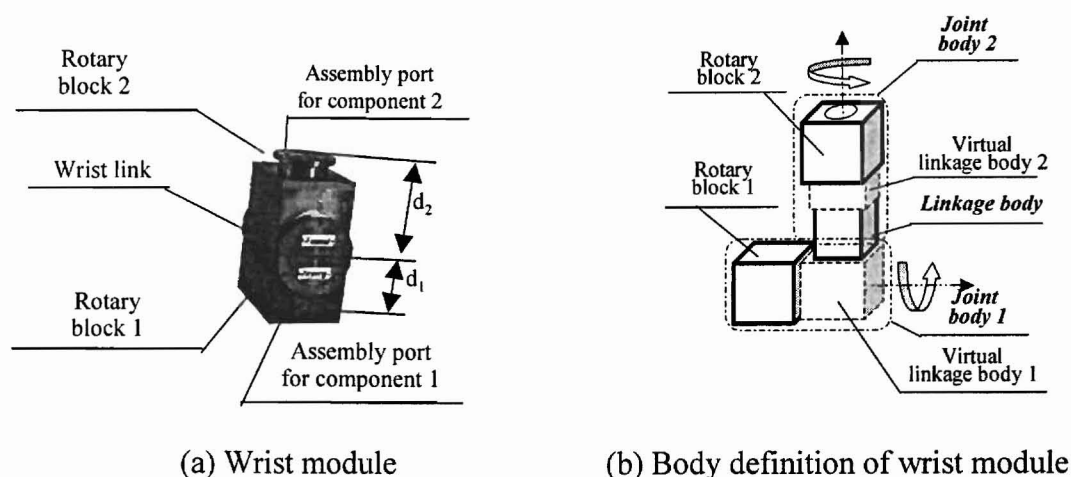
A coordinate system is associated with each linkage body and each joint body (see Figure 4.1), and such a coordinate system is called a local coordinate system. A local coordinate system on the base body, which is fixed on the ground, serves as a world coordinate system for describing motions of moving objects. In Figure 4.1, the local coordinate system for joint body  $i$  is shown to be  $\{O_{i,0}, X_{i,0}, Y_{i,0}, Z_{i,0}\}$ , and is denoted by  $F_{i,0}$  for short. Because objects are inter-connected, their local coordinate systems are related through a coordinate transformation matrix. A local coordinate system of any moving object is related to the world coordinate system through the coordinate transformation matrix. The definition of the local coordinate systems for different types of joint and linkage bodies for the AMTEC system is given in Appendix A.



$a_i$ : the length of the common normal between  $Z_{i-1}$  and  $Z_i$   
 $\alpha_i$ : the angle between  $Z_{i-1}$  and  $Z_i$  measured about  $X_i$   
 $d_i$ : the distance from  $X_{i-1}$  to  $X_i$  measured along  $Z_{i-1}$   
 $\theta_i$ : the angle between  $X_{i-1}$  and  $X_i$  measured about  $Z_{i-1}$

**Figure 4.1** Architecture for the D-H parameters

For a link module or a joint module with 1-DOF, a link module and a joint module correspond to a linkage body and a joint body, respectively. For a joint module with multi-DOFs, each motion axis corresponds to one joint body. Physically, there is no linkage body between two motion axes. However, the D-H notation requires a pattern, i.e., joint body-linkage body-joint body. The concept of a virtual linkage body is proposed to resolve this issue. The *virtual linkage body* has no mass, and is a geometric element. Taking a wrist module with 2-DOF, shown in Figure 4.2, as an example to explain the virtual linkage body concept, this module has three physical components (Figure 4.2a): rotary block 1, rotary block 2, and wrist link. The wrist link holds the two rotary motion axes. Figure 4.2b shows the representation of the wrist system in the D-H architecture, where two virtual linkage bodies are introduced.



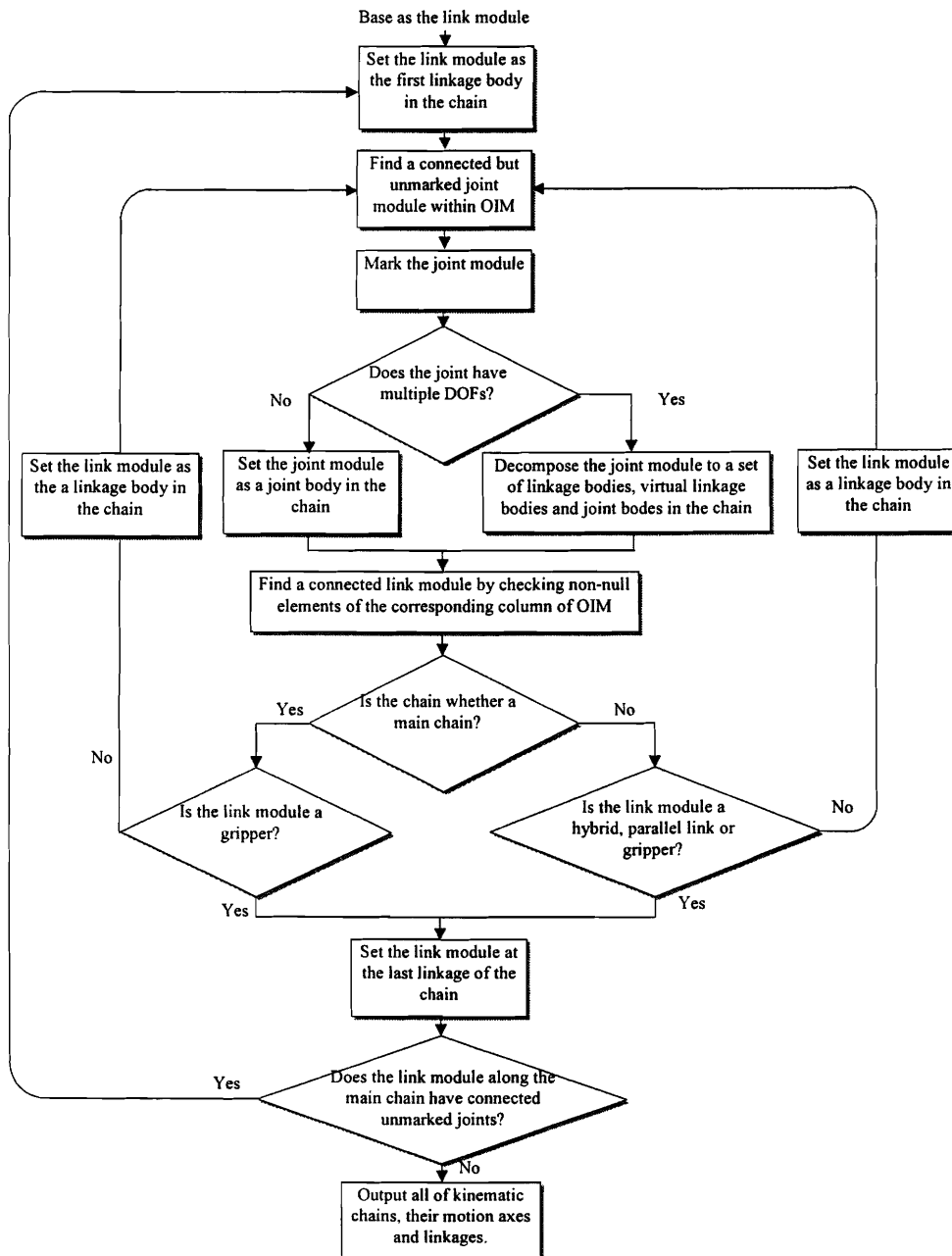
**Figure 4.2** The concept of virtual linkage body: the wrist module

### 4.2.2 Computer-aided generation of the D-H architectures

A procedure to obtain the D-H architecture (i.e., kinematic chains, their motion axes, and linkages) is shown in Figure 4.3. The main strategy in computer-aided generation of



D-H architecture from the OIM representation is as follows: (i) in the outer loop of the procedure, the process reads the OIM representation and determines on whether the chain is a main chain or a branch chain; (ii) if a link module is found to connect with three or more joint modules, there will be one or some branch chains, and (iii) the procedure will be finished when all joint modules are examined.



**Figure 4.3** Procedure to derive the D-H architecture from OIM

## **4.3 Kinematic Conversion**

Each kinematic chain contains a series of joint bodies and linkage bodies; therefore it has multiple sets of D-H parameters.

### **4.3.1 Labeling for joint and link bodies within a kinematic chain**

In a main kinematic chain or a branch kinematic chain, the joint bodies and linkage bodies are connected in a sequence from the initial link body to the final link body. Link bodies and joint bodies are labeled in terms of their positions within the kinematic chain, i.e., link body 0, link body 1, ..., link body  $n+1$  for link bodies and joint body 1, joint body 2, ..., joint body  $n$  for joint bodies, respectively, where  $n$  is the number of joint bodies. The number of sets of the D-H parameters equals the number of joint bodies. A set of the D-H parameters corresponding to each joint body is derived from the first to the last joint body.

### **4.3.2 Kinematic properties of a joint and linkage body**

The D-H parameters are determined by the spatial relationship of three joint bodies and two linkage bodies, as shown in Figure 4.1, and this set of the D-H parameters corresponds to the intermediate joint body. A particular set of D-H parameters, say D-H parameter set  $i$ , follows the identification of joint body  $i$ . For joint body  $i$ , the D-H parameters are determined by the following kinematic properties of individual joint

bodies:

- (i) Local coordinate system for joint body  $i$ ,  $F_{i,0} = \{ O_{i,0}, X_{i,0}, Y_{i,0}, Z_{i,0} \}$
- (ii) Type of joint body  $i$
- (iii) Two ports on joint body  $i$ : input port  $P_{i,in}$  and output port  $P_{i,out}$
- (iv) Input port on joint body  $i+1$ :  $P_{i+1,in}$
- (v) Transformation matrix  $LT_i$  from  $P_{i,out}$  and output port  $P_{i+1,in}$ .

Definitions of local coordinate systems for each joint and linkage body are given in Appendix A.

### 4.3.3 Determination of the D-H parameters for joint body $i$

Assume that the set of the D-H parameters for joint body  $i-1$  and its corresponding coordinate system  $i-1$  with respect to the world coordinate system have been determined in the preceding steps. Let:

- $X_i, Z_i$  : axes of the coordinate system attached to the motion axis of joint body  $i$   
with respect to the world coordinate system
- $O_{i,1}, O_{i,2}$  : intersecting points between  $X_{i-1}$  and  $Z_{i-1}$ , and between  $X_i$  and  $Z_{i-1}$ ,  
respectively; see Figure 4.1

The rules for assigning the D-H axes ( $X_i$  and  $Z_i$ ) are as follows:

*Rule 1:*  $Z_i$  is assigned to the motion axis of joint body  $i+1$ .

*Rule 2:*  $X_i$  is assigned along the common normal of  $Z_{i-1}$  and  $Z_i$ , and its direction follows the right-hand rule from  $Z_{i-1}$  to  $Z_i$ .

In the following, the positions of the D-H axes and the corresponding kinematic parameters are computed in the reference system: the local coordinate system of joint body  $i$ , i.e.,  $F_{i,0} = \{O_{i,0}, X_{i,0}, Y_{i,0}, Z_{i,0}\}$ . For the example of  $Z_i$ , it can be expressed in  $F_{i,0}$  by:

$${}_{i,0}Z_i = \left( P_{i,out} \cdot LT_i \cdot (P_{i+1,in})^{-1} \right) {}_{i+1,0}Z_i \quad (4.1)$$

where

${}_{i,0}Z_i$  :  $Z$ -axis of the coordinate system attached to the motion axis of joint body  $i+1$  with respect to the coordinate system  $F_{i,0}$ .

${}_{i+1,0}Z_i$  :  $Z$ -axis of the coordinate system attached to the motion axis of joint body  $i+1$  with respect to the local coordinate system of joint body  $i+1$ .

The location of  $X_i$  is decided by two scales  $k_1$  and  $k_2$ , see Figure 4.1. *First*, one obtains:

$$\left. \begin{aligned} O_{i,2} &= O_{i,0} + k_1 \cdot Z_{i-1} \\ O_{i+1,1} &= O_{i+1,0} + k_2 \cdot Z_i \end{aligned} \right\} \quad (4.2)$$

Because  $X_i$  is perpendicular to both  $Z_{i-1}$  and  $Z_i$ , the following equations can be derived:

$$\left. \begin{aligned} (O_{i,2} - O_{i+1,1}) \cdot Z_{i-1} &= 0 \\ (O_{i,2} - O_{i+1,1}) \cdot Z_i &= 0 \end{aligned} \right\} \quad (4.3)$$

From eq. (4.2) and (4.3),  $k_1$  and  $k_2$  can be obtained as:

$$\left. \begin{aligned} k_1 &= \mathbf{Z}_{i-1} \cdot (\mathbf{O}_{i,1} - \mathbf{O}_{i,0}) \\ k_2 &= -\mathbf{Z}_i \cdot (\mathbf{O}_{i+1,0} - \mathbf{O}_{i,1}) \end{aligned} \right\} \mathbf{Z}_{i-1} \cdot \mathbf{Z}_i = 0$$

$$\left. \begin{aligned} k_1 &= \frac{(\mathbf{Z}_{i-1} - \mathbf{Z}_i \cdot (\mathbf{Z}_{i-1} \cdot \mathbf{Z}_i)) \cdot (\mathbf{O}_{i+1,0} - \mathbf{O}_{i,0})}{(1 - (\mathbf{Z}_{i-1} \cdot \mathbf{Z}_i))^2} \\ k_2 &= \frac{(-\mathbf{Z}_i + \mathbf{Z}_{i-1} \cdot (\mathbf{Z}_{i-1} \cdot \mathbf{Z}_i)) \cdot (\mathbf{O}_{i+1,0} - \mathbf{O}_{i,0})}{(1 - (\mathbf{Z}_{i-1} \cdot \mathbf{Z}_i))^2} \end{aligned} \right\} \mathbf{Z}_{i-1} \cdot \mathbf{Z}_i \neq 0 \quad (4.4)$$

After  $\mathbf{X}_i$  is determined, the D-H kinematic parameters are calculated by:

$$\left. \begin{aligned} a_i &= |\mathbf{O}_{i+1,1} - \mathbf{O}_{i,2}| \\ \alpha_i &= \cos^{-1}(\mathbf{Z}_{i-1} \cdot \mathbf{Z}_i) \\ d_i &= (\mathbf{O}_{i,2} - \mathbf{O}_{i,1}) \cdot \mathbf{Z}_{i-1} \\ \varphi_i &= \cos^{-1}(\mathbf{X}_{i-1} \cdot \mathbf{X}_i) \end{aligned} \right\} \quad (4.5)$$

## 4.4 Dynamic Conversion

A joint is shared by two linkage bodies. Therefore, a joint is divided into two components from a viewpoint of the physical mass. Consequently, a linkage body, which has two end joints, should have three sources of contributions to its total masses: the linkage body, and two halves of the end joints.

Dynamic parameters must refer to a particular reference coordinate system through the center of mass. For linkage body  $i$ , this coordinate system is parallel to the coordinate system  $\mathbf{F}_{i+1,1} = \{ \mathbf{O}_{i+1,1}, \mathbf{X}_i, \mathbf{Y}_i, \mathbf{Z}_i \}$ , which is determined through kinematic conversion, and denoted by  $\mathbf{F}_{i+1,1}^C$ . Also,  $\mathbf{X}_i$ ,  $\mathbf{Y}_i$ , and  $\mathbf{Z}_i$  are assumed to be three principal inertia axes,

Figure 4.4 illustrates the dynamic parameters for linkage  $i$ , where  $m_i$  is the mass and vector  $r_i$  is the center of the mass, and they are computed, by:

$$\left. \begin{aligned} m_i &= m_{i,1} + m_{i,2} + m_{i,3} \\ r_i &= (m_{i,1}r_{i,1} + m_{i,2}r_{i,2} + m_{i,3}r_{i,3}) / m_i \end{aligned} \right\} \quad (4.6)$$

where

$m_i$ , and  $r_i$  are the mass and the center of mass with respect to  $F_{i+1,1}$ , respectively.

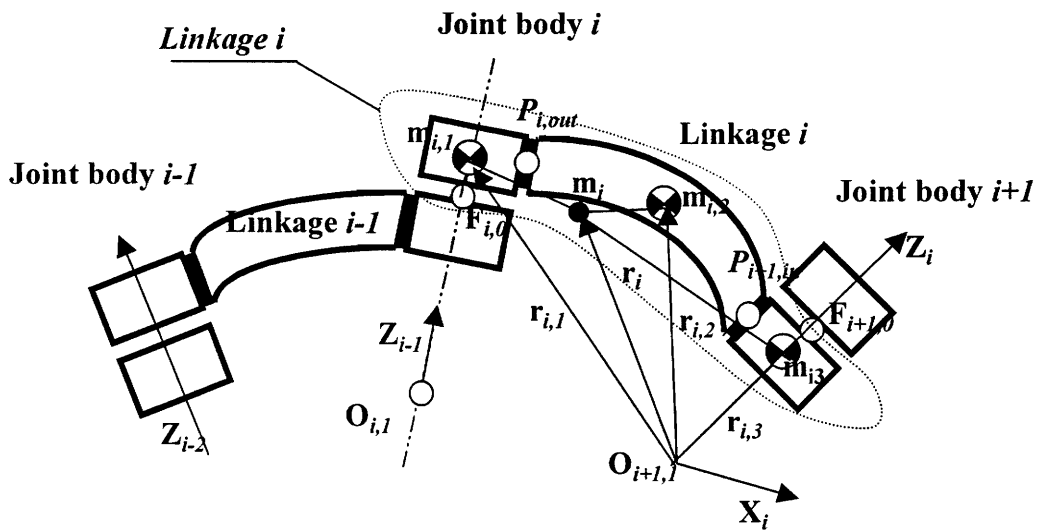
$m_{i,j}$ , and  $r_{i,j}$  ( $j=1,2,3$ ) are the mass and the center of the mass associated with extended

linkage body  $i$ , respectively, and in particular:

$j=1$ : half joint body  $i$  (left)

$j=2$ : linkage body  $i$  (middle)

$j=3$ : half joint body  $i+1$  (right)



**Figure 4.4** Dynamic parametric conversion

Three mass components for extended linkage body  $i$  have their own local coordinate

systems, which are denoted  $F_{i,o}$ ,  $P_{i,out}$  and  $F_{i+1,0}$ . Care must be taken that  $F_{i,0}$  and  $F_{i+1,0}$  are the frames which originate at the center of mass of half joint body  $i$  and half joint body  $i+1$  (Figure 4.4), respectively, and they are parallel to the local frames which are determined through kinematic conversion, as discussed above. The moments and the products of inertia of the three mass components are then calculated by the following two steps:

**Step 1:** Apply the parallel axis theorem to calculate the contribution of each mass component to the center of mass of the extended linkage body  $i$ . For example, the contribution of mass component of joint body  $i$  (left) is calculated by:

$$\left. \begin{aligned} \bar{I}_{x_{i,0}x_{i,0},1} &= I_{x_{i,0}x_{i,0},1} + m_{i,1} \left( (Y_{i,0} \cdot (r_i - r_{i,1}))^2 + (Z_{i,0} \cdot (r_i - r_{i,1}))^2 \right) \\ \bar{I}_{y_{i,0}y_{i,0},1} &= I_{y_{i,0}y_{i,0},1} + m_{i,1} \left( (X_{i,0} \cdot (r_i - r_{i,1}))^2 + (Z_{i,0} \cdot (r_i - r_{i,1}))^2 \right) \\ \bar{I}_{z_{i,0}z_{i,0},1} &= I_{z_{i,0}z_{i,0},1} + m_{i,1} \left( (X_{i,0} \cdot (r_i - r_{i,1}))^2 + (Y_{i,0} \cdot (r_i - r_{i,1}))^2 \right) \\ \bar{I}_{x_{i,0}y_{i,0},1} &= m_{i,1} (X_{i,0} \cdot (r_i - r_{i,1})) (Y_{i,0} \cdot (r_i - r_{i,1})) \\ \bar{I}_{x_{i,0}z_{i,0},1} &= m_{i,1} (X_{i,0} \cdot (r_i - r_{i,1})) (Z_{i,0} \cdot (r_i - r_{i,1})) \\ \bar{I}_{y_{i,0}z_{i,0},1} &= m_{i,1} (Y_{i,0} \cdot (r_i - r_{i,1})) (Z_{i,0} \cdot (r_i - r_{i,1})) \end{aligned} \right\} \quad (4.7)$$

where

$I_{ww,1}$  : moment of inertia of the mass component, joint body  $i$ , with respect to axis  $w$  in frame  $F_{i,o}$ .

$I_{wh,1}$  : product of inertia of the mass component, joint body  $i$ , with respect to axes  $w$  and  $h$  in frame  $F_{i,o}$ .

$\bar{I}_{ww,1}$  : moment of inertia of the mass component, joint body  $i$ , with respect to axis  $w$  in a frame with origin  $O_{i+1,1}$  and parallel to  $F_{i,o}$ .

$\bar{I}_{wh,1}$  : product of inertia of the mass component, joint body  $i$ , with respect to axes  $w$  and  $h$  in a frame with origin  $O_{i+1,1}$  and parallel to  $F_{i,0}$ .

By changing 1 to 2 (3) and  $F_{i,0}$  to  $F_{i+1,0}$  ( $P_{i,out}$ ), one can obtain the contribution of the moment of inertia and the product of inertia of half joint body  $i+1$  (right) and link body  $i$  to the center of mass of extended linkage body  $i$ .

**Step 2:** The moment of inertia and the product of inertia calculated from the first step occurs about the frames which are set up at the center of mass  $r_i$  of the extended linkage body  $i$ , and in parallel to the local frames. These local frames are different from the frame of extended linkage body  $i$ ,  $F_{i+1,1}$ . Therefore, a conversion must be performed, which calculates the contribution of each component mass with respect to  $F_{i+1,1}$  by applying the formula of rotation of axes (Jong and Rogers 1991). For joint body  $i$  (left), the rotation matrix,  $R_l$ , is formed by:

$$R_l = \begin{bmatrix} X_{i,0} \cdot X_i & Y_{i,0} \cdot X_i & Z_{i,0} \cdot X_i \\ X_{i,0} \cdot Y_i & Y_{i,0} \cdot Y_i & Z_{i,0} \cdot Y_i \\ X_{i,0} \cdot Z_i & Y_{i,0} \cdot Z_i & Z_{i,0} \cdot Z_i \end{bmatrix} \quad (4.8)$$

The inertia tensor of joint body  $i$  (left) with respect to  $F_{i+1,1}$  can be calculated by:

$$\begin{bmatrix} I_{x_i x_i, l} & -I_{x_i y_i, l} & -I_{x_i z_i, l} \\ -I_{x_i y_i, l} & I_{y_i y_i, l} & -I_{y_i z_i, l} \\ -I_{x_i z_i, l} & -I_{y_i z_i, l} & I_{z_i z_i, l} \end{bmatrix} = R \begin{bmatrix} \bar{I}_{x_{i,0} x_{i,0}, l} & -\bar{I}_{x_{i,0} y_{i,0}, l} & -\bar{I}_{x_{i,0} z_{i,0}, l} \\ -\bar{I}_{x_{i,0} y_{i,0}, l} & \bar{I}_{y_{i,0} y_{i,0}, l} & -\bar{I}_{y_{i,0} z_{i,0}, l} \\ -\bar{I}_{x_{i,0} z_{i,0}, l} & -\bar{I}_{y_{i,0} z_{i,0}, l} & \bar{I}_{z_{i,0} z_{i,0}, l} \end{bmatrix} R^T \quad (4.9)$$

where the left side of eq. (4.9) is the moment and product of inertia with respect to  $F_{i+1,1}$ , and the right side is calculated by eq. (4.8).



The contribution to the moments of inertia from linkage body  $i$  and joint body  $i+1$  can also be obtained through a similar procedure. As a result, the integrated moments of inertia of 'linkage'  $i$  can be obtained by:

$$\left. \begin{aligned} I_{x_i x_i} &= \sum_{j=1}^{j=3} I_{x_i x_i, j} & I_{y_i y_i} &= \sum_{j=1}^{j=3} I_{y_i y_i, j} & I_{z_i z_i} &= \sum_{j=1}^{j=3} I_{z_i z_i, j} \\ I_{x_i y_i} &= \sum_{j=1}^{j=3} I_{x_i y_i, j} & I_{x_i z_i} &= \sum_{j=1}^{j=3} I_{x_i z_i, j} & I_{y_i z_i} &= \sum_{j=1}^{j=3} I_{y_i z_i, j} \end{aligned} \right\} \quad (4.10)$$

where

$I_{ww}$ : moment of inertia of the extended linkage body  $i$  with respect to axis  $w$  in  $F_{i+1,1}$ .

$I_{wh}$ : product of inertia of the extended linkage body  $i$  with respect to axes  $w$  and  $h$  in  $F_{i+1,1}$ .

$j = 1$ : moments and products of inertia of the half joint body  $i$  (left).

$j = 2$ : moments and products of inertia of the linkage body  $i$  (middle).

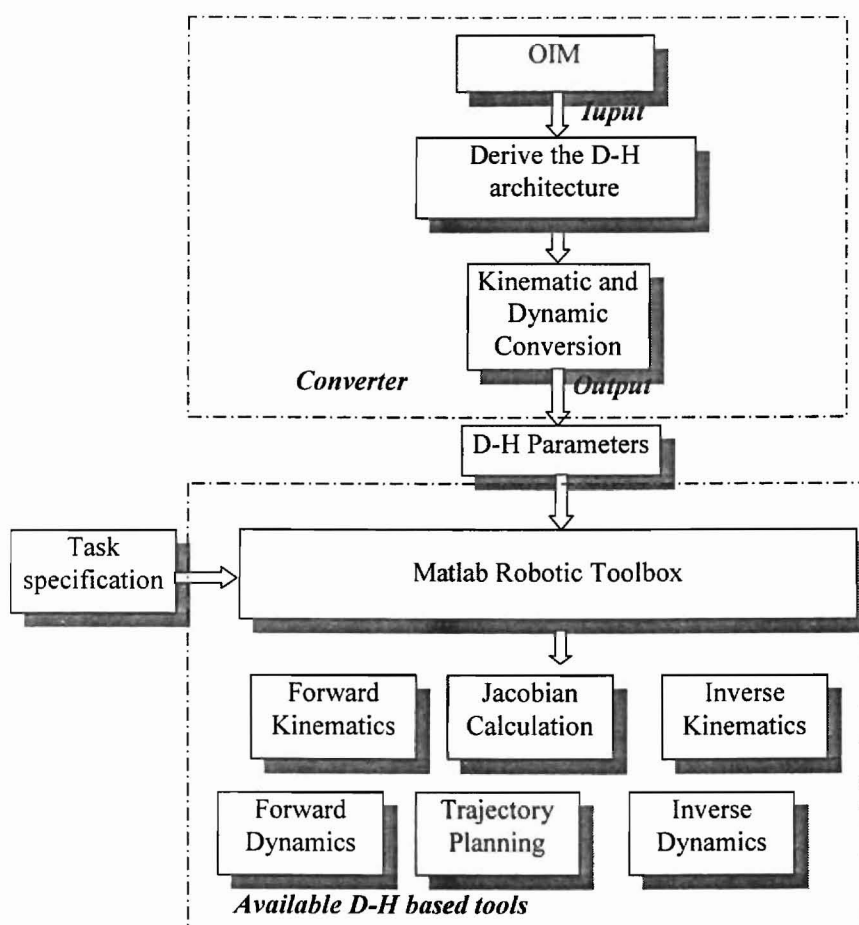
$j = 3$ : moments and products of inertia of the half joint body  $i+1$  (right).

## 4.5 Computer Program Implementation

A computer program for the kinematic and dynamic conversion was implemented. This program was coded using the Matlab language, and was called 'Converter'. The input to *Converter* is the OIM, and the output to Converter is the D-H parameters/variables, see Figure 4.5.

To further compute the kinematic and dynamic behaviors, such as the forward and

inverse kinematics and dynamics, Jacobian matrix, trajectory planning, a tool called the ‘Robotic Toolbox’, available in the Matlab environment, was used (Corke 1996). The complete program, which calculates the kinematic and dynamic behaviors for a given modular robot configuration, and contains the Converter and Robotic Toolbox, is called “automatic kinematic and dynamic analysis” (AKDA). The flowchart of AKDA is shown in Figure 4.5, and some details of AKDA are provided in Appendix B.



**Figure 4.5** The program flow chart of AKDA

## 4.6 Validation

The purpose of this validation is to verify whether our programs, Converter and AKDA, are correct. Only a validation of the kinematic model was conducted for the following reasons:

- (i) Commercial robotic systems typically use kinematic controls only; therefore the validation of kinematics has a practical significance.
- (ii) The kinematic model is a basis for the dynamic model. In other words, a reliable dynamic model largely depends on a reliable kinematic model.
- (iii) The determination of system dynamic parameters is a difficult task. Indeed, most robot manufacturers provide very limited dynamic information for their systems, which is not enough to build a dynamic model;
- (iv) Once a kinematic model is obtained, the D-H dynamic parameters are calculated from modular physical properties; thus the dynamic model could be derived using the Newton-Euler algorithm or Lagrange's method. There is no theoretical significance to validate dynamic modeling.

The validation was conducted with two methods: (i) in comparison with a validated theoretical model, and (ii) in comparison with a physical system. In this thesis, the first method is reported, because the second method is largely dependent on the quality of the control system for the physical system. The controller of the AMTEC system was inappropriate as a reference to validate a simulation model (Bi 2001d). The validated

theoretical model was developed by Nanyang University of Science and Technology, Singapore, which has been implemented in the Simulation Environment for Modular Robot System (SEMORS).

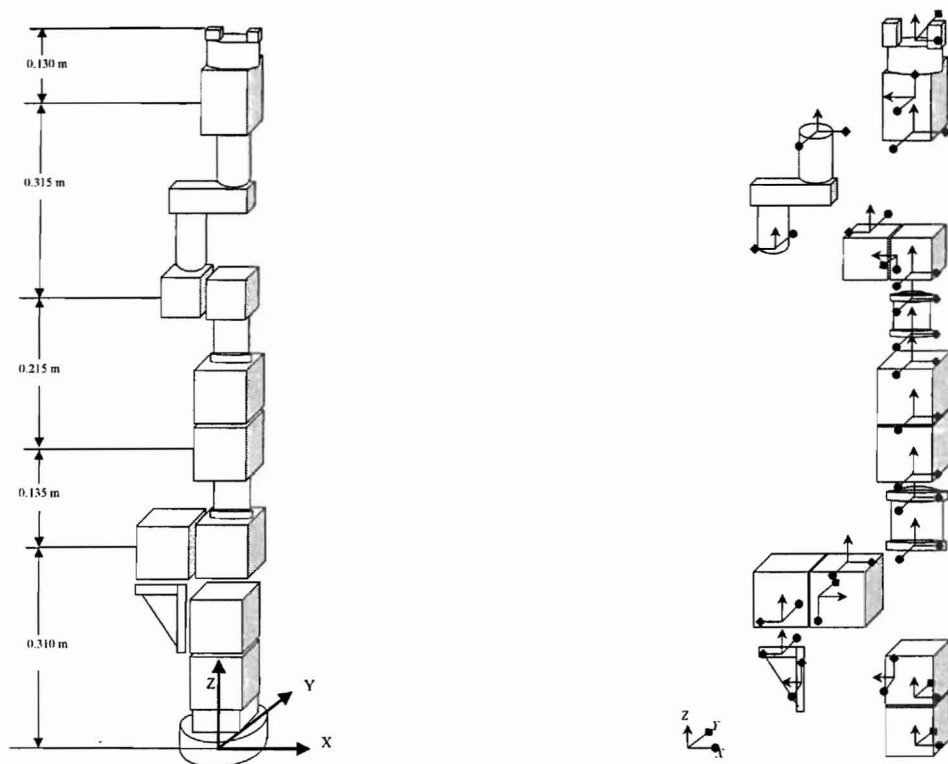
#### 4.6.1 Example system for validation

A modular robot system, manufactured by the AMTEC Company, Germany, was used for this validation. Information of the modules can be found in Appendix A. The AMTEC system was set up at the Robotic Research Center, Nanyang University of Technology, Singapore. A 6-DOF modular robot configuration, see Figure 4.6, was used in the validation. This robot system was intended for loading and unloading operations.

For this system, the OIM representation as follows:

$$\begin{array}{c}
 \begin{array}{c}
 <'base', 1> \\
 <'link', 19> \\
 <'link', 4> \\
 <'link', 3> \\
 <'link', 20> \\
 <'gripper', 2>
 \end{array}
 \begin{pmatrix}
 \begin{array}{c}
 <'joint', 2> \\
 <'joint', 2> \\
 <'joint', 2> \\
 <'joint', 1> \\
 <'wrist', 1>
 \end{array} \\
 \begin{array}{ccccc}
 <1,0,1,1> & \phi & \phi & \phi & \phi \\
 <1,1,1,7> & <1,0,2,2> & \phi & \phi & \phi \\
 \phi & <1,1,1,6> & <1,0,2,1> & \phi & \phi \\
 \phi & \phi & <1,1,1,6> & <1,0,2,2> & \phi \\
 \phi & \phi & \phi & <1,1,1,6> & <1,0,2,1> \\
 \phi & \phi & \phi & \phi & <1,1,1,2>
 \end{array}
 \end{pmatrix}
 \end{array}$$

In this example, the robot system consists of four rotary joint modules, one wrist module, five link modules, and one gripper module. Figure 4.6b shows assembly patterns of each module.



(a) Configuration at the initial position

(b) Modules and their sub-assemblies

**Figure 4.6** Experimental modular robot

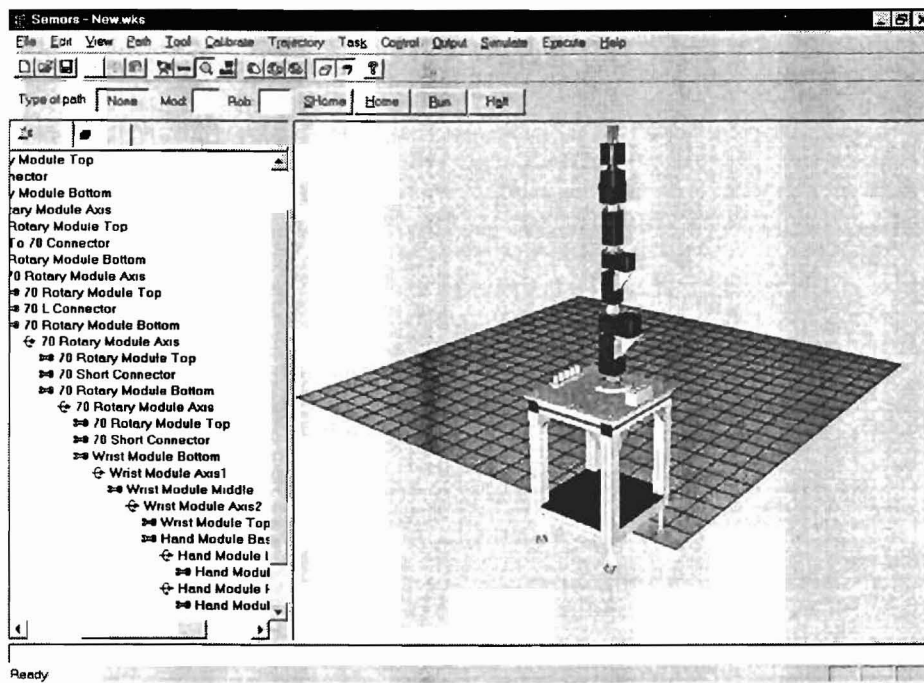
The information of task specification is given in Table 4.1, where a set of points is prescribed.

**Table 4.1** Prescribed pose for task specification

	$(x, y, z)$ (unit: m)	R
p0	(0.1775, 0.3350, 0.1900)	[1, 0, 0; 0, -1, 0; 0, 0, -1]
p1	(-0.2970, 0.1775, 0.2300)	[0, 1, 0; 1, 0, 0; 0, 0, -1]
p2	(-0.2970, 0.1775, 0.1000)	[0, 1, 0; 1, 0, 0; 0, 0, -1]
p3	(-0.2970, -0.0035, 0.2300)	[0, 1, 0; 1, 0, 0; 0, 0, -1]
p4	(-0.2970, -0.0035, 0.1000)	[0, 1, 0; 1, 0, 0; 0, 0, -1]
p5	(-0.2885, -0.1607, 0.2300)	[0, 1, 0; 1, 0, 0; 0, 0, -1]
p6	(-0.2885, -0.1607, 0.1000)	[0, 1, 0; 1, 0, 0; 0, 0, -1]
p7	(0.3443, 0.0011, 0.4300)	[1, 0, 0; 0, -1, 0; 0, 0, -1]
p8	(0.4150, 0.0011, 0.4300)	[1, 0, 0; 0, -1, 0; 0, 0, -1]
p9	(0.4150, 0.0011, 0.3600)	[1, 0, 0; 0, -1, 0; 0, 0, -1]

## 4.6.2 SEMORS system

The SEMORS system performs inverse kinematics and task level control. The methodology of SEMORS is based on the Product-of-Exponentials (POE) method (Chen 2001). For example, the graphical interface of SEMORS is shown in Figure 4.7.



**Figure 4.7** SEMORS developed by Nanyang Technological University

The modular configuration and the corresponding task specification were input to the SEMORS program. SEMORS generated motions and motion planning at the joint level through the inverse kinematics module. We assumed that SEMORS was correct and, therefore, was taken as a reference to validate the AKDA program.

A 6-DOF robot usually has multiple inverse solutions. Different solutions may be

reached using the same algorithm from different initial points. However, with the trajectory divided into 500 time segments, the same solution was obtained from a variety of initial points.

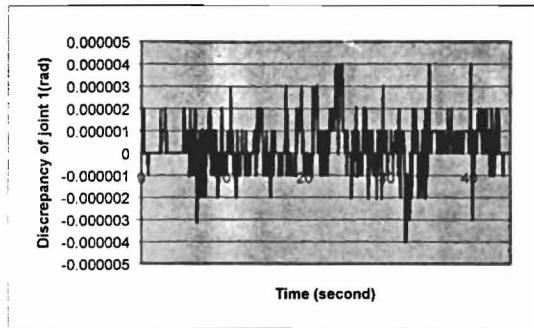
### 4.6.3 Results and discussion

Converter generated the following D-H parameters for the modular robot (Figure 4.6).

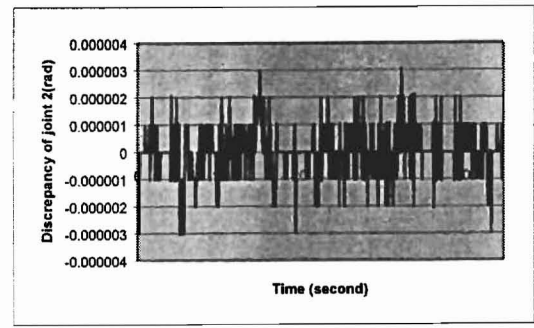
These parameters were verified by manual calculation.

Joint \ D-H	Motion type	$a_i(\text{m})$	$d_i(\text{m})$	$\alpha_i(\text{rad})$	$\theta_i(\text{rad})$
1	R	0.000000	0.310000	1.570796	1.570796
2	R	0.000000	0.000000	1.570796	3.141593
3	R	0.000000	0.350000	1.570796	0.000000
4	R	0.315000	0.000000	0.000000	1.570796
5	R	0.000000	0.000000	1.570796	1.570796
6	R	0.000000	0.130000	0.000000	-1.570796

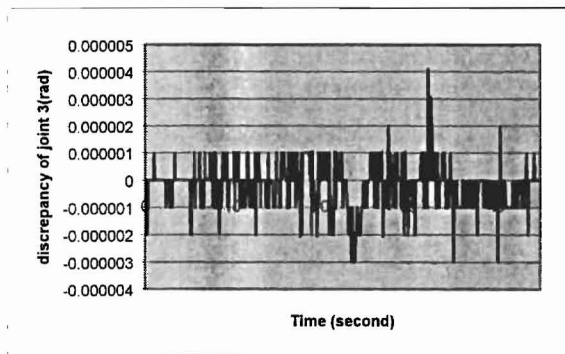
The joint displacements from the AKDA program were compared with those obtained from SEMORS. As shown in Figure 4.8 a-f, there is a good match between the results from the POE models and AKDA (maximum error  $\sim 0.00004$  rad). Corresponding to the given positions and orientations of working points, both SEMORS and AKDA reach the same inverse kinematic solutions, the joint displacements derived from the task specification.



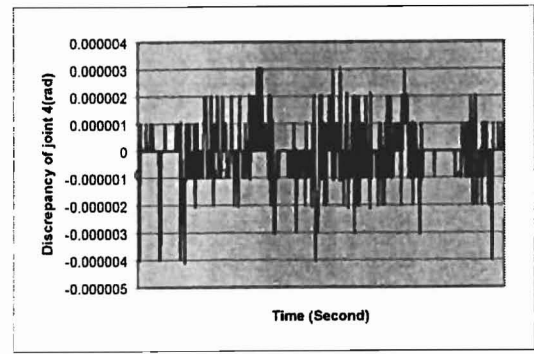
(a)



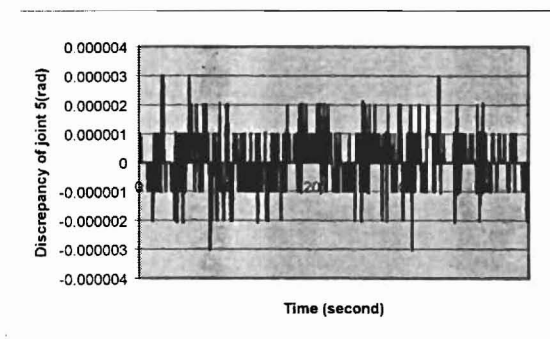
(b)



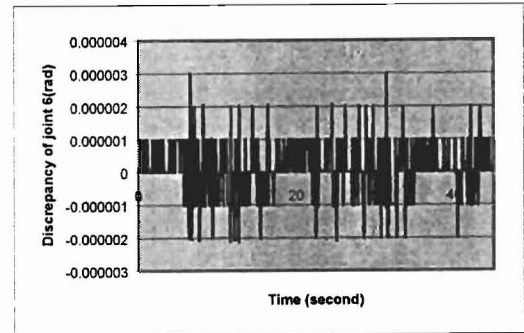
(c)



(d)



(e)



(f)

**Figure 4.8** The comparison of joint displacements from SEMORS and AKDA



#### **4.6.4 Remarks**

An experiment was conducted to acquire operational data from ATMEC robot. Our results showed some discrepancies between the joint displacements from theoretical model and those obtained from actual operation of the physical robot. A detailed explanation and some suggestions to improve real-time control of modular robotic systems are provided in the reference (Bi 2001d).

#### **4.7 Summary**

In this chapter, a new approach to the computer-aided generation of kinematic and dynamic equations was developed based on indirect modeling method. In indirect modeling, a kinematically independent representation of configurations of a modular robot was converted into a kinematically dependent representation by using D-H notation. Then kinematically dependent representations are employed to calculate the kinematic and dynamic behaviors.

In comparison with existing research work, our proposed approach has the following advantages: (i) it provides a systematic procedure for modeling and simulation of the kinematic and dynamic behaviors; (ii) it starts from the most primary level of a modular robot configuration, which is well-suited for modular robot configuration design; (iii) it is applicable to all types of modular robot systems that consist of link and joint modules; (iv) it can accommodate available programs for kinematic and dynamic analysis of non-modular robot systems.

## Chapter 5

### Modular Robot Configuration Synthesis

As discussed in Chapter 1, an important issue in adaptive or modular robot systems is to determine an optimal modular robot configuration, based on an underlying architecture, to meet a given task. The latter process is called *modular robot configuration synthesis*. This chapter presents the development of a methodology and its implementation for performing this process. The architecture used to generate various configurations is the GAMR, presented in Chapter 3.

This chapter is organized as follows. In Section 5.1, modular robot configuration synthesis is defined. In Section 5.2, we present the need for concurrent treatment of design objectives for effective modular system configuration synthesis. This leads to a new method for modular robot configuration synthesis, called the concurrent optimal design method. In Section 5.3, the configuration synthesis problem is expressed as an optimization problem. In Section 5.4, a computational method is described for implementing the theory and methodology developed in the preceding sections. In Section 5.5, a design case is illustrated. In Section 5.6, a summary with discussion is provided.

## 5.1 Problem Definition

Modular robot configuration synthesis can be viewed as a mapping from task space to design space. The *task space* is a set of task specifications. A task can be specified from the user-level to the system-level. At the user-level, we may view a task as the end-effector traveling along a circle with the diameter and center of the circle prescribed; at the system-level, that same task may be specified by a set of points that fit the circle. For the present study, the task specification at the system-level is only considered. For the purpose of demonstrating the theory and the methodology to be presented later, the following task is considered in the present study:

- (i) a set of working points in the world coordinate system,
- (ii) a set of payload requirements corresponding to the working points, and
- (iii) a set of time spans required to travel between any two working points.

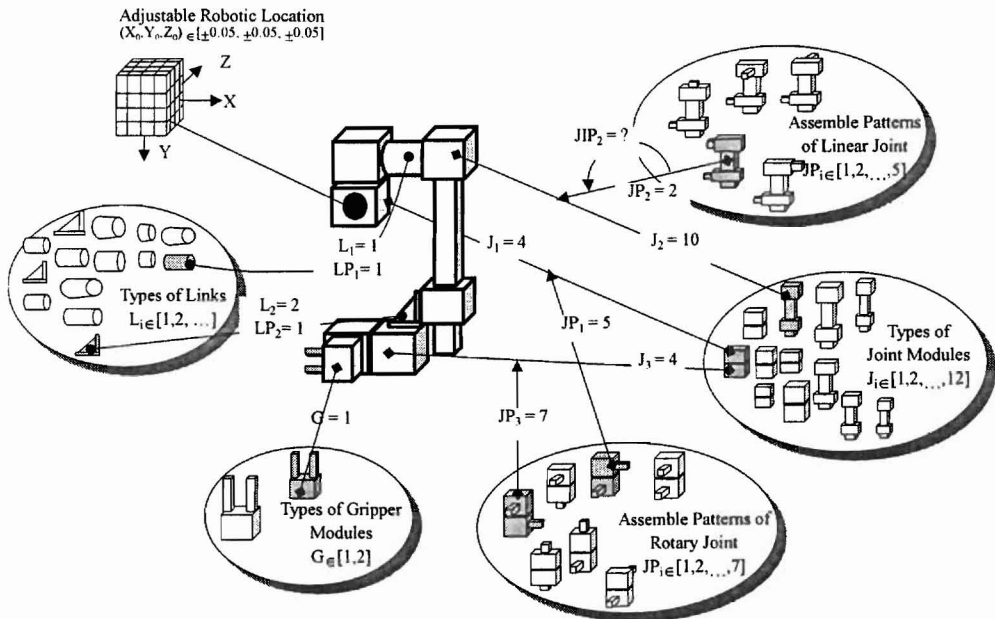
Note that the time requirement above makes it possible to take dynamic constraints into consideration for modular robot configuration synthesis.

The *design space* is a set of all feasible configuration variations. The size (the number of variations) of this set depends on the architecture of modular robot system. In this thesis, GAMR is chosen to be the architecture, and OIM is used for configurations based on the GAMR. Variations of modular robot configurations, based on GAMR, and design variables representing these variations, are discussed in Section 5.3.

## 5.2 An Example Revisited

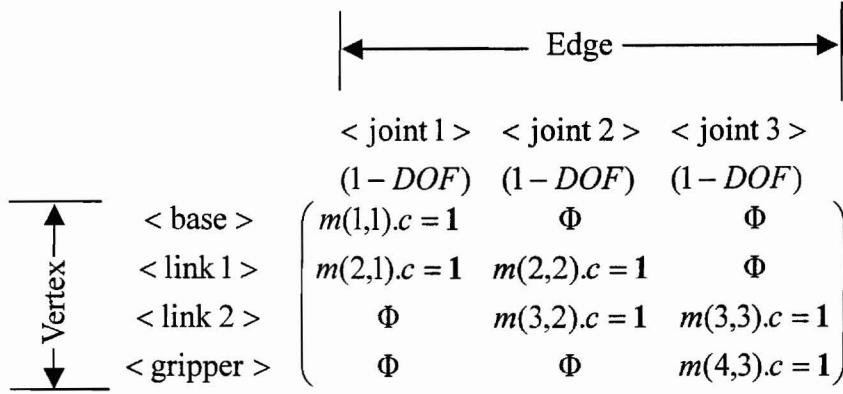
Section 3.5.6 provided an example of a 3-DOF configuration of a modular robot to show how to define the OIM of a modular robot configuration. This example is revisited for the purpose of illustrating the method for modular robot configuration synthesis.

As shown in Figure 5.1, the configuration is based on the architecture of the AMTEC modular robot system. The AMTEC system consists of 12 types of joint modules, 14 types of link modules, and 5 and 7 assembly patterns for the rotary and linear joint modules, respectively. The link module in the AMTEC system has a fixed kinematic dimension with a unique assembly pattern. A detailed description of the AMTEC system is provided in Appendix A.



**Figure 5.1** The construction of a 3 DOF configuration

The OIM for this example system is as follows:

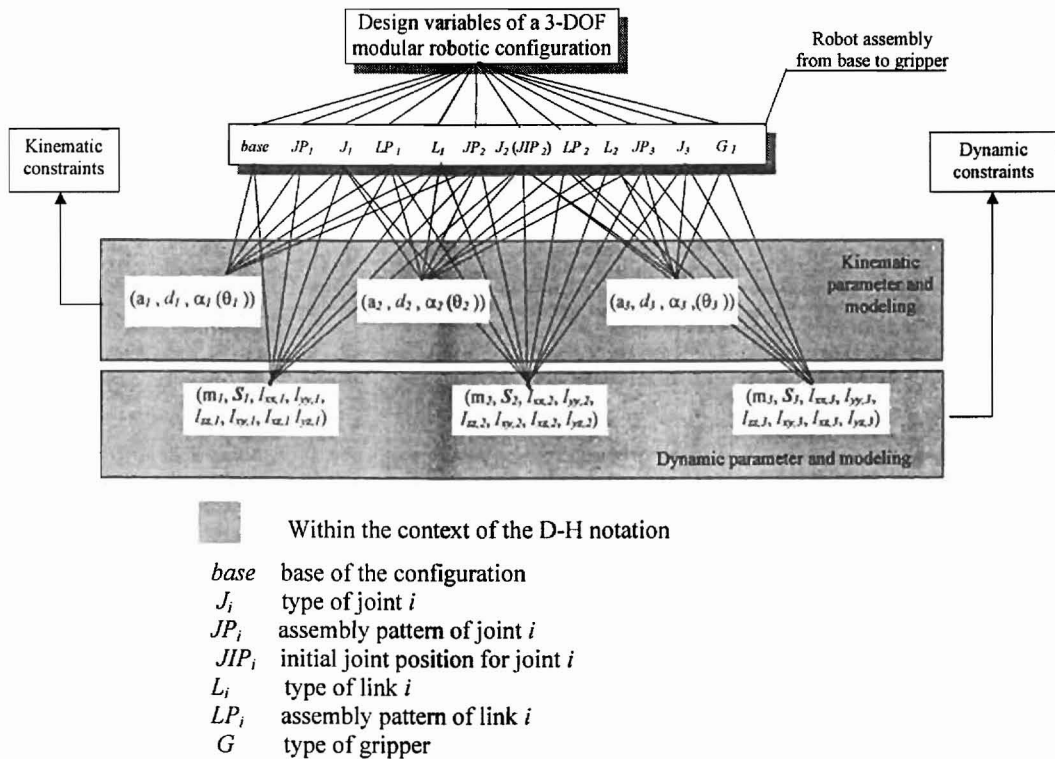


The OIM representing the system consists of 3 joint modules, 2 link modules and 1 gripper module. Furthermore, there are six non-null elements within the OIM, which implies six incidences between links and joints.

The following module variables are defined to determine this 3-DOF configuration:

- $J_i (i=1,2,3)$  : types of joints ( $J_1 = 4, J_2 = 10, J_3 = 4$ ), where  $i$  denotes a particular joint module in an assembly.
- $L_i (i=1,2)$  : types of links ( $L_1 = 1, L_2 = 2$ ), where  $i$  denotes a particular link module in an assembly.
- $G$  : type of gripper ( $G = 1$ ).
- $JP_i (i=1,2,3)$  : assembly patterns for the joint modules ( $JP_1 = 5, JP_2 = 2, JP_3 = 7$ ).
- $LP_i (i=1,2)$  : assembly patterns for the link modules ( $LP_1 = 1, LP_2 = 1$ ).
- $JIP_i (i=1,2,3)$  : initial joint position where the relative motion between two parts starts in a linear joint module.

The relationships between these module variables and the kinematic and dynamic variables or parameters of this configuration are shown in Figure 5.2. For example, module variable  $J_i$  is associated with the kinematic and dynamic parameters of link 1 and link 2, respectively. It should be noted that the kinematic and dynamic parameters are defined in terms of the D-H notation. The details of the conversions from these variables to the D-H parameters were discussed in Chapter 4.



**Figure 5.2** Modular variables, and kinematic and dynamic parameters

### 5.3 An Observation: the Need for Concurrent Design

The following general observations can be obtained from Figure 5.2 for the modular

robot configuration example:

- (i) In modular robot configuration synthesis, the module variables represent the physical construction of a robot configuration, and they are derived from the given modular robot architecture.
- (ii) The module variables determine the assembly of a robot configuration and its kinematic and dynamic behaviors. Both the kinematic and dynamic parameters are derived from the module variables; in other words, the kinematic and dynamic behavior are strongly coupled with the module variables. Once a module is selected, its effects on all aspects of robot behavior, the kinematic and dynamic behavior, are known.
- (iii) If the kinematic design and dynamic design are tackled sequentially, the solution in the kinematic design can 'fix' all module variables for the dynamic design, because the same set of the module variables is involved for both kinematic design and dynamic design. This makes it impossible to carry out further optimization for the dynamic design.
- (iv) The number of design variables (which are now module variables) is greatly reduced in comparison with the number of design variables in the case of non-modular robot system configuration synthesis. Taking the modular robot configuration shown in Figure 5.1 as an example, there are at most 14 module variables. The number of module variables changes with the selection of module types. For example, if a linear joint module is brought into the set of possible considerations, a new module variable for the initial joint position is created. However, for a 3-DOF non-modular robot configuration (which corresponds to the 3-DOF modular robot, but is represented in

the D-H notation), there are 39 design variables (3 kinematic and 10 dynamic D-H parameters for each DOF).

Based on the above observations, if modular robot configuration synthesis starts from the module variable/parameter level, a simultaneous consideration of all relevant design goals (kinematic and dynamic) is a necessity. The design model formulated based on these observations, is called the *concurrent optimal design method* (CODM) (Bi and Zhang 2001b). Furthermore, these observations also imply that any synthesis process that does not start from the module variable/parameter level may not have a solution.

## **5.4 Optimization Model for Configuration Synthesis**

As mentioned earlier, modular robot configuration synthesis is a mapping from the task space to design space. The key point here is to determine a representation for the mapping and a process to solve mapping. Optimization is widely recognized as a general tool for mechanism synthesis (Breteler 1997). An optimization model will be applied to develop a computational method for modular robot configuration synthesis. Two primary issues are: the formulation of an optimization model and its solution.

### **5.4.1 The optimization model**

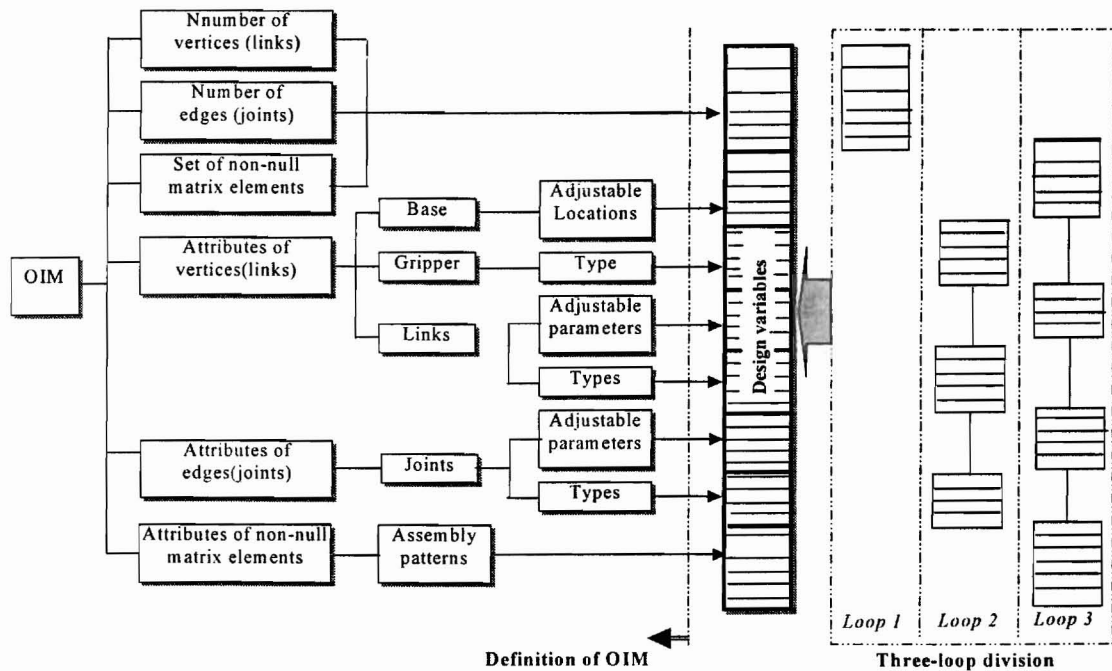
An optimization model consists of specifications of design variables, constraints, and objectives.



## Design variables

The *design variables* represent a modular robot configuration. In this thesis, the design variables are the *module variables*. The module variables are derived from the OIM. Indeed, the OIM includes all information regarding configuration variations: (i) at the robot type level, the number of the non-null matrix elements in the corresponding rows in the OIM imply the variations of serial, parallel, and hybrid robots; (ii) at the configuration level, the size of the OIM represents the variations caused due to the numbers of joint modules and link modules; (iii) at the module level, the assembly attributes of non-null elements in the OIM represent the variations of the module assembly patterns; and (iv) at the parameter level, the attributes of the OIM edges in the OIM represent the variations of local adjustable variables (for example, the initial position in a linear joint module of the example shown in Figure 5.1 is an adjustable variable/parameter).

Modular robot configuration synthesis needs a one-to-one mapping between an OIM and a set of design variables. Because the design synthesis deals with the design variables, while design analysis starts from the OIM, the set of design variables and the OIM are uniquely related. As shown in Figure 5.3, the OIM corresponds to a set of attributes; these attributes are described by design variables. There is a one-to-one correspondence between the OIM attributes and design variables. For example, in the system of Figure 5.1, the number of joints, which is a module variable, is, the number of columns in the OIM.



**Figure 5.3** Relationship between OIM and design variables

The domains of some design variables are determined by a given modular robot architecture. For the example system shown in Figure 5.1, there are 12 joint types for the AMTEC system. Other variables have domains determined by the size and morphology of the OIM. For example, the number of joints in a configuration has a domain which depends on the number of the columns in the OIM. Such domains are inherently large. Therefore, in the synthesis process, one needs to impose the limits to the domains of these design variables. For the optimization model, the domains of the design variables are represented as design constraints.

### Design constraints

Design constraints model the task specification. As mentioned earlier, there are three aspects. The modeling of these aspects with a representation of the constraints is

presented below.

#### *Constraints for the first task specification*

Given a point at task level, say  $P_i$ , through the inverse kinematics, one can obtain the corresponding displacements at the joint level,  $\theta_{ik}$  ( $k$  = index of a joint module).

The design task to make the robot end-effector achieve a desired joint can be modeled by the following constraint:

$$\theta_k^{\min} \leq \theta_{ik} \leq \theta_k^{\max} \quad (5.1)$$

where  $\theta_k^{\min}$  and  $\theta_k^{\max}$  are the joint limits.

#### *Constraints for the second task specification*

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 make the robot end-effector subject to the described payload can be modeled by the following constraints:

$$|f_{ik}| \leq f_k^{\max} \quad (5.2)$$

where  $f_k^{\max}$  is the maximum force/torque a joint module can generate.

#### *Constraints for the third task specification*

Given the time span between any two neighboring points ( $i, j$ ) at the task level  $t_{ij}$  (two points  $P_i$  and  $P_j$  or  $\theta_{ik}$  and  $\theta_{jk}$ ), a path planning method can be applied to obtain a curve along which the robot end-effector travels from  $P_i$  to  $P_j$  or joint  $k$  travels from  $\theta_{ik}$  to  $\theta_{jk}$ .

In this thesis, we selected a path planning method in the Matlab Toolbox for Robotics based on a fifth order polynomial. After the detailed curve is formed, the time span  $t_{ij}$  is divided into  $m$  time segments; for joint module  $k$ , a series of joint displacements, joint velocities, and joint accelerations can be obtained:

$$\theta_{ik}^1, \theta_{ik}^2, \dots, \theta_{ik}^m \text{ (displacements)}$$

$$\dot{\theta}_{ik}^1, \dot{\theta}_{ik}^2, \dots, \dot{\theta}_{ik}^m \text{ (velocities)}$$

$$\ddot{\theta}_{ik}^1, \ddot{\theta}_{ik}^2, \dots, \ddot{\theta}_{ik}^m \text{ (accelerations)}$$

In addition, one should apply the inverse dynamic analysis to obtain forces/torques on joint module  $k$ , corresponding to these interpolated points  $\theta_{ik}^1, \theta_{ik}^2, \dots, \theta_{ik}^m$ , i.e.,  $f_{ik}^1, f_{ik}^2, \dots, f_{ik}^m$ . Therefore, the design task to make the robot end-effector travel from  $P_i$  and  $P_j$  with time  $t_{ij}$  can then be modeled by the following constraints:

$$\theta_k^{\min} \leq \theta_{ik}^1, \theta_{ik}^2, \dots, \theta_{ik}^m \leq \theta_k^{\max} \quad (5.3)$$

$$\dot{\theta}_k^{\min} \leq \dot{\theta}_{ik}^1, \dot{\theta}_{ik}^2, \dots, \dot{\theta}_{ik}^m \leq \dot{\theta}_k^{\max} \quad (5.4)$$

$$|f_{ik}^j| \leq f_k^{\max} \quad j = 1, 2, \dots, m \quad (5.5)$$

where  $\dot{\theta}_k^{\min}$  and  $\dot{\theta}_k^{\max}$  are limits of the velocity in joint  $k$ .

### *Other constraints*

One of the most important constraints in robot configuration synthesis is absence of singularities for any design. This constraint can be modeled by examining the Jacobian matrices of the working points at the task-level. In particular, at the working points or any points between the working points (obtained through the path planning), their respective Jacobian matrices should be free of singularities:

$$\left| \Delta \left( J(\theta_{i1}, \theta_{i2}, \dots) \cdot J'(\theta_{i1}, \theta_{i2}, \dots) \right) \right| \geq \varepsilon \quad (5.6)$$

where

$J(\theta_{i1}, \theta_{i2}, \dots)$  is the Jacobian matrix corresponding to the inverse kinematics of  $P_i$ ,

$\Delta$  is the determinant of the matrix,

$\varepsilon$  is a small positive number.

### Design objectives

The main objective is to find a set of feasible solutions that meet all design constraints. To improve the design solution, an index of energy consumption is introduced, which is similar to the index of power consumption used by Paredis (1994). The motivation is that when a modular robot system is employed, the major cost associated with the configuration is the energy expended in operation. This index is calculated as follows:

$$E = \int_{Tra} \sum_{i=1}^n |\tau_i(\theta)| d\theta_i \quad (5.7)$$

where

$n$ : number of the joint axes of a robot configuration

$E$ : total energy consumption along the trajectory of the task

$Tra$ : space consisting of all of working points along the trajectory

$\tau_i(\theta)$ : torque executed on motion axis  $i$

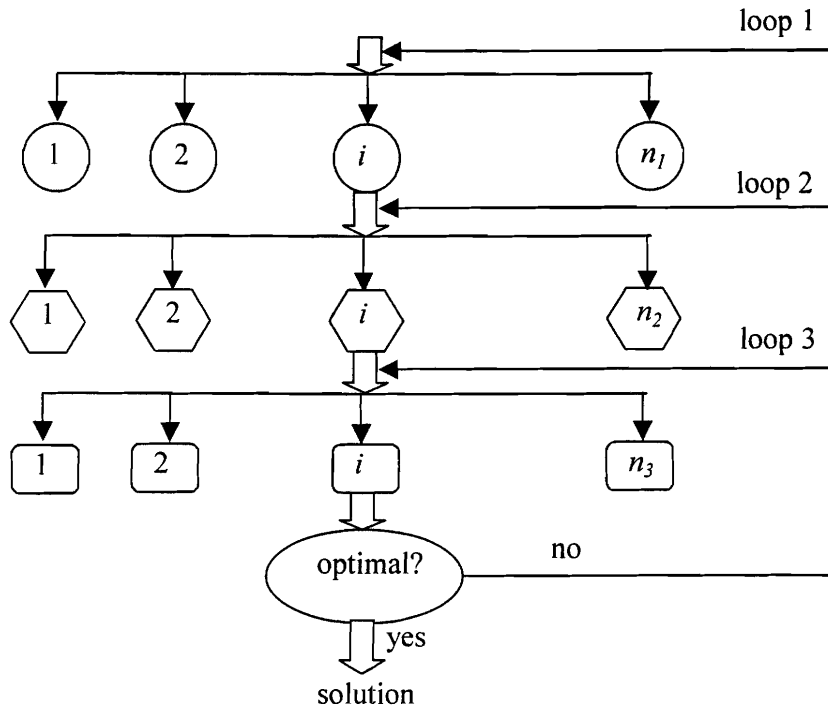
### 5.4.2 General strategies for solving the optimization problem




When the concurrent optimal design method (CODM) is applied for modular robot configuration synthesis, which was elaborated as a necessity, the dimension of the optimization problem could be large, thereby, challenging computational effectiveness. One solution to this problem is to decompose the design space.

There is another important reason for decomposing the design space. When the configuration synthesis is employed at the modular parameter level, the design variables are considerably heterogeneous in the sense that some variables represent types of a system (type-related variables), and other variables represent parameters of a single attribute (attribute-related variables). The updating of the type-related variables may require a change in variable topology (i.e., number, type, and domain of the variables). For example, introducing a linear joint module will require the introducing of one new variable for the initial position of that linear joint module.

This thesis proposes a three-loop decomposition of the design space, as shown in Figure 5.4. In loop 1, the design space is described in terms of the definition of the OIM. The design variables in this loop include: the number of joints, the number of links, and the connectivity between the joints and the links. In loop 2, the design variables include: joints types and link types, in conformity with loop 1. For example, the system shown in Figure 5.1 has three joints; each joint can be one of 12 types. In loop 3, the design variables include those adjustable parameters on each of the modules (attributes of the vertex in the OIM), and on each connection between modules (attributes of non-null

elements).



-  : Variations due to different sizes of the OIM
-  : Variations due to different types of vertices and edges of the OIM
-  : Variations due to different assembly patterns, adjustable parameters and location of the base platform

**Figure 5.4** Decomposition of the design space

In Figure 5.4, the three-loop decomposition is within the same synthesis process; it does not imply that sequential optimization has been employed. Because each loop determines only a part of design variables for an entire configuration candidate, the system kinematic and dynamic behaviors of the entire configuration could not be analyzed until the design variables of all three loops are obtained. Concurrent analysis of design variables, constraints, and objectives, is finally performed in loop 3.

## 5.5 Implementation by Genetic Algorithm (GA)

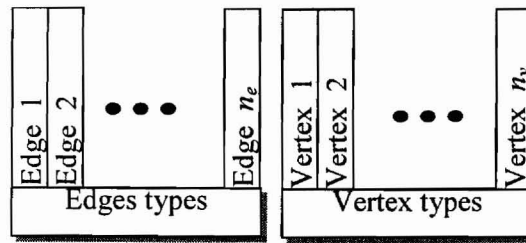
A computational method was developed in the Matlab environment for modular robot configuration synthesis, including a strategy for decomposing the design space. The Matlab Robotic Toolboxes (Corke 1996) and the GA Toolbox (Chipperfield et al. 1994), were used. The program was for the GAMR, though the AMTEC system was used as a case study. The resulting program is called the *Adaptive Robot Configuration Synthesis* (ARCS).

In ARCS, a Genetic Algorithm (GA) was used because it is capable of handling an optimization problem with continuous and discrete variables and it does not need the derivatives of the objective function.

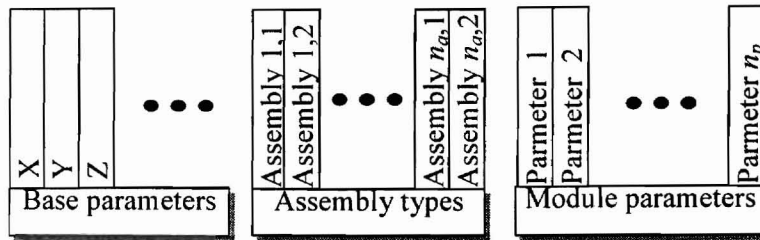
GA uses a fixed length binary string (chromosomes) to represent a design variable. A design solution (spanned by a set of design variables) is represented by a string that merges all variable strings. For the convenience, the variable string is called in the following discussion the *sub-string*. The arrangement of sub-strings within an entire string reflects the topology of a set of design variables. For example, the order of sub-strings could represent the sequential module connections. However, the means to represent a topology of variables based on the sequential ordering of sub-strings is not sufficient to ensure a solution, especially for complex problems. Therefore, rules are needed to arrange sub-strings into a solution string. Figure 5.5 shows how to arrange sub-strings within a solution in the ARCS program. In loop 2, the design variables represent options of the edges and vertices of the OIM, see Figure 5.5a. In particular, the



sub-strings for the variables corresponding to edge 1 to edge  $n_e$  (where  $n_e$  is the total number of the columns in the OIM) are first arranged within a solution string; the sub-strings for the variables corresponding vertex 1 to vertex  $n_v$  (where  $n_v$  is the total number of the rows in the OIM) follow. In loop 3, the design variables represent (i) the base location parameters, (ii) the assembly patterns, (iii) modular adjustable parameters. Their sub-strings are arranged, respectively, as shown in Figure 5.5 b.



(a) In loop 2



(b) In loop 3

**Figure 5.5** From design variables to GA strings

In ARCS, the objective function, eq. (5.7), was used as an evaluation function in the GA. Since the GA evaluates the solution that achieves the maximum function as the best solution, minus sign is put before the objective function described in eq. (5.7). For a candidate solution that does not meet the constraints, its evaluation function will be given a larger negative value to disqualify this candidate.

## 5.6 Case Study

The task specification is listed in Table 5.1. From Table 5.1, the end-effector is required to pass through the pre-defined points, and at each point, the robot has to stop and perform some operations. This means that the velocity and acceleration are required to be zero at each working point. The time elapsed between two neighboring working points is 1 second. The working load on each working point is the sum of the weights of the gripper module and the work piece.

**Table 5.1** Task specification

No.	Position (m)	Time between points (sec)
1	(0.12, 0.0, 0.0)	
2	(0.12, 0.0, 0.12)	1
3	(0.0649, 0.0649, 0.12)	1
4	(0, 0.12, 0.12)	1
5	(0, 0.12, 0)	1

Suppose that the user requires a 3-DOF robot for this task. This implies that the design synthesis in the first loop is completed. Therefore, only the synthesis for loop 2 and 3 are considered. The configuration of the system in loop 1 was shown in Figure 5.1, and its OIM was also given in Section 5.2.

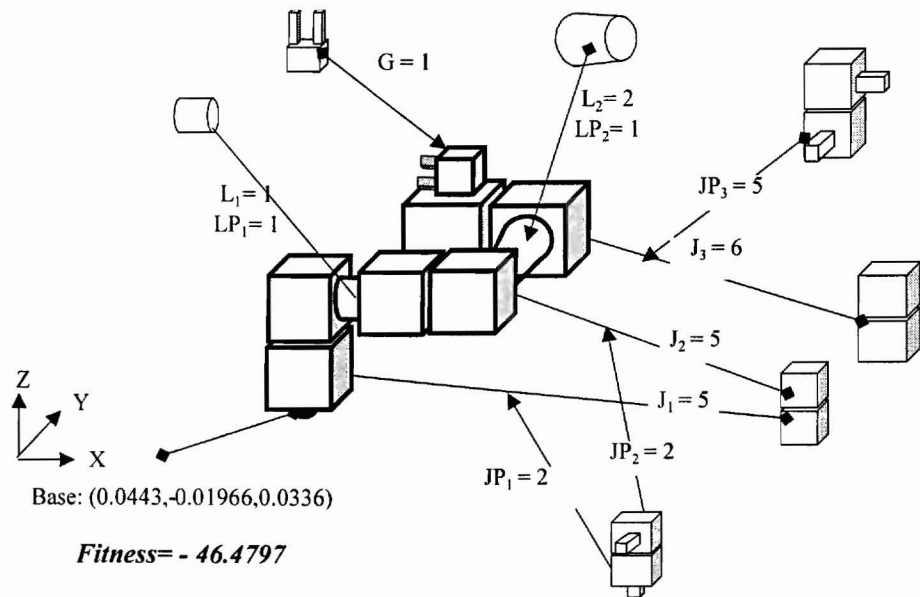
The design variables in loop 2 (6 in total) include: (i) 3 variables for joint types, (ii) 2 variables for link types, and (iii) 1 variable for gripper type. The design variables in loop 3 include: (i) 3 variables for the location of the base, (ii) 3 variables for assembly patterns, and (iii) 0 - 3 variables for adjustable parameters if there is any linear joint module type chosen by the variables for joint types. It is noted that the link modules of the AMTEC system have only two ports; therefore, there is only one assembly pattern of

link modules, which requires no variable.

For implementation,  $10^{-3}$  is the resolution of the design variables, i.e., the minimal change that can be made for the design variables. The initial population is 100, and the number of population in each generation is also 100. The termination condition for the GA program is 100 generations. More detailed information regarding the information for the case study can be found in Appendix C.

The first generation only created one feasible solution, see Appendix C, as shown in Figure 5.6. Figure 5.6 shows the configuration of a feasible solution (1). Detailed specifications about joint and link modules in the assembly appear in Appendix A.

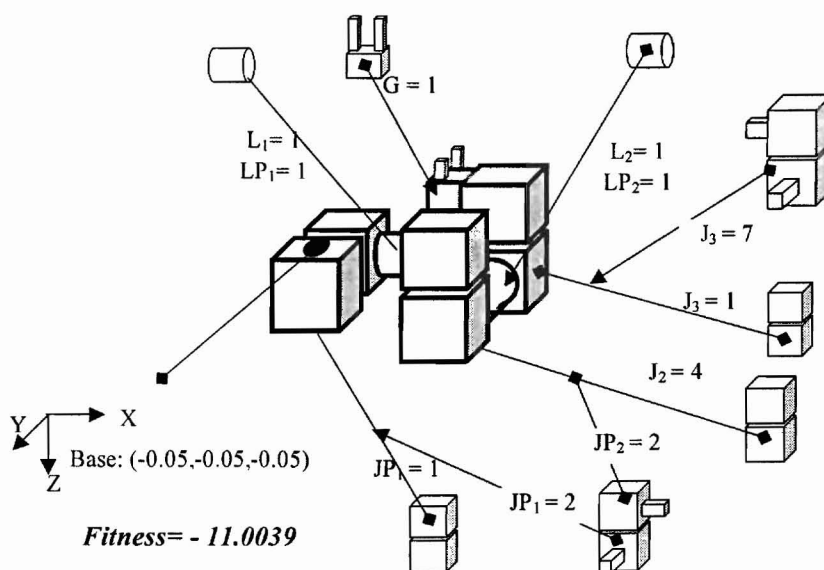
*Feasible solution (1)*



**Figure 5.6** Feasible solution (1) consisting of 3 rotary modules (fitness= - 46.4797)

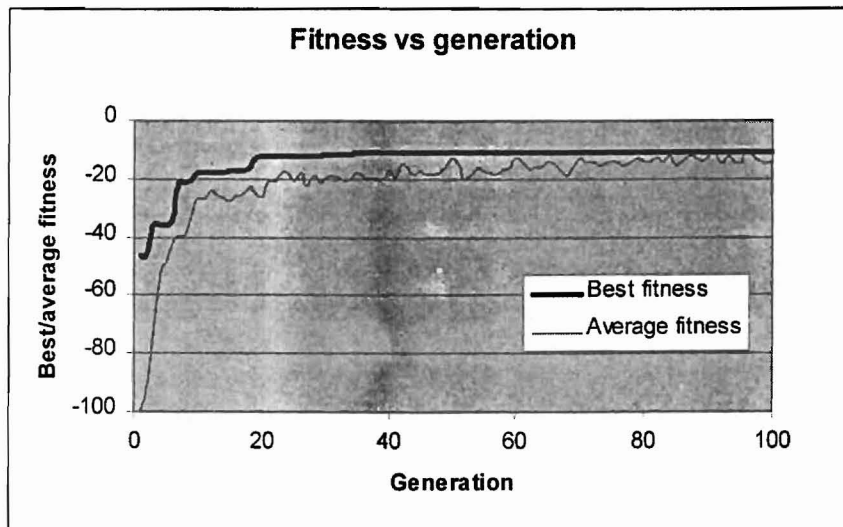
A feature of the system topology is that all three joints are rotary. With this topology, further GA search results in an optimal solution, which is shown in Figure 5.7:

*Feasible solution (2)*

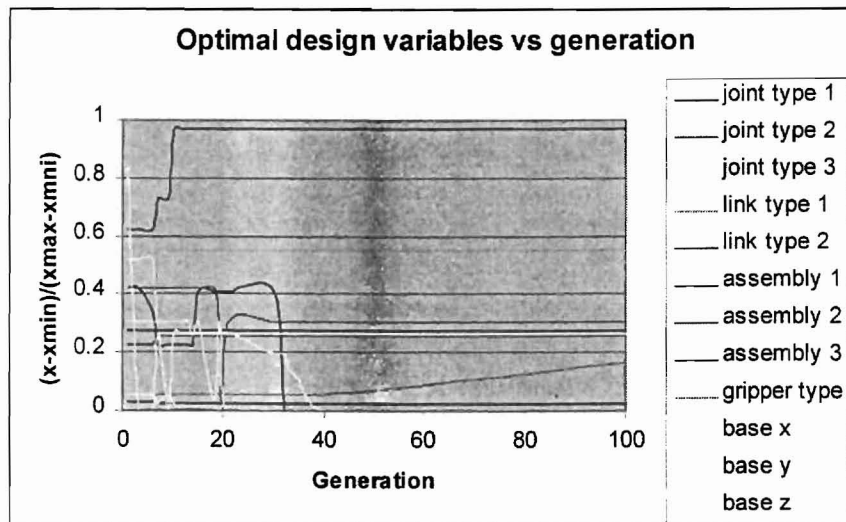


**Figure 5.7** Feasible solution (2) consisting of 3 rotary modules  
(fitness= - 11.0039)

Figure 5.8 shows the evolution process, where, after 30 generations, the solution has converged. The same phenomenon can be observed when the design variables are normalized to  $[0,1]$  using  $(x-x_{\min}) / (x_{\max}-x_{\min})$  as shown in Figure 5.9.



**Figure 5.8** Best/average fitness (N·m)

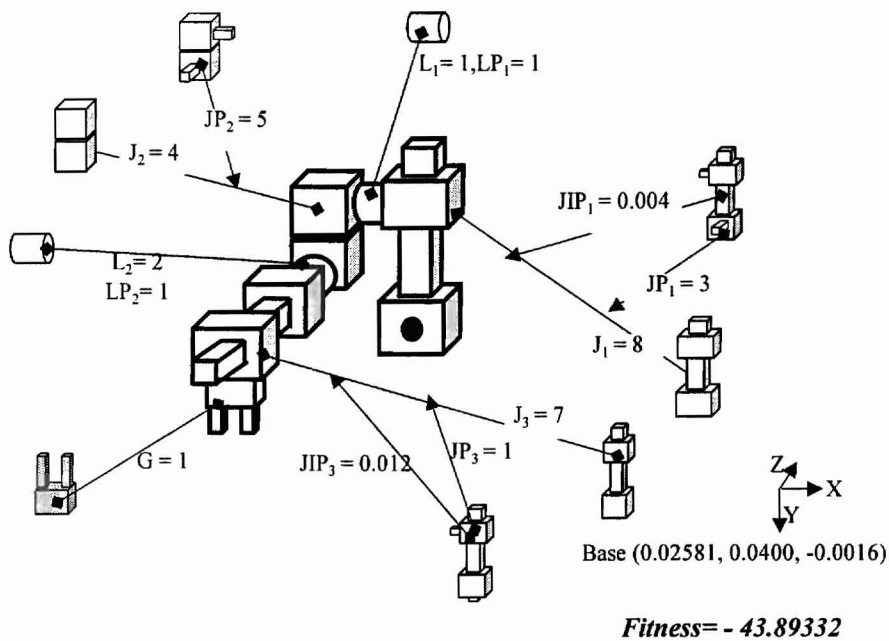


**Figure 5.9** Optimal design variables (normalized)

The ARCS program will further turn to the situation where in loop 2 there is one linear joint module among three joint modules. For this situation, the number of design

variables is changed from 12 to 13, where the additional variable represents the initial position of a linear joint module. There was no solution found for this situation. The ARCS program considers the situation where two joint modules are linear, and there are 14 variables (two new variables for the initial positions of two linear joint modules, respectively). A feasible solution was found for this situation, and as shown in Figure 5.10:

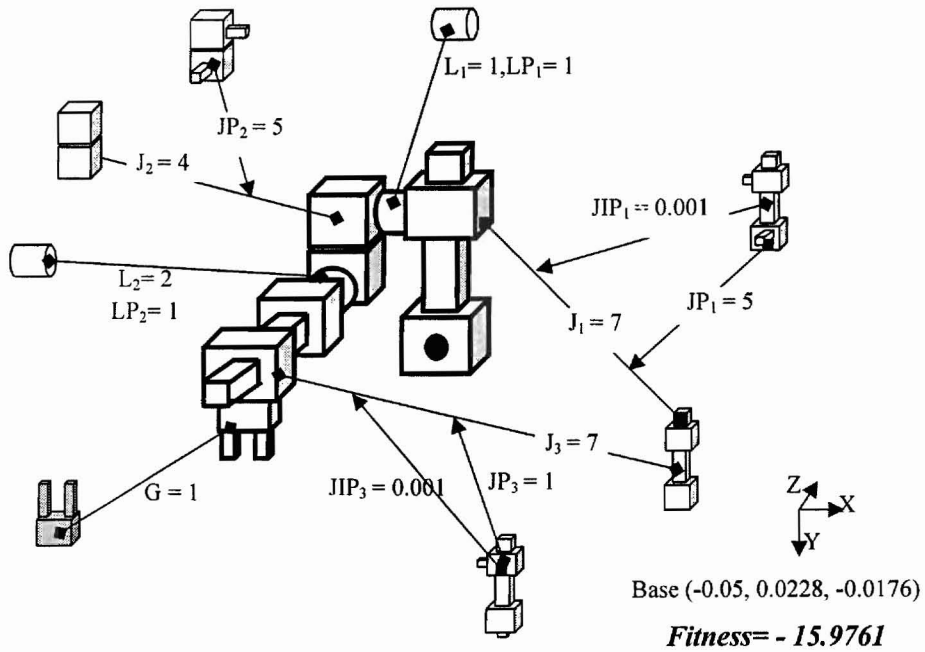
*Feasible solution (3)*



**Figure 5.10** Feasible solution (3) consisting of one rotary module and two linear modules (fitness= - 43.89332)

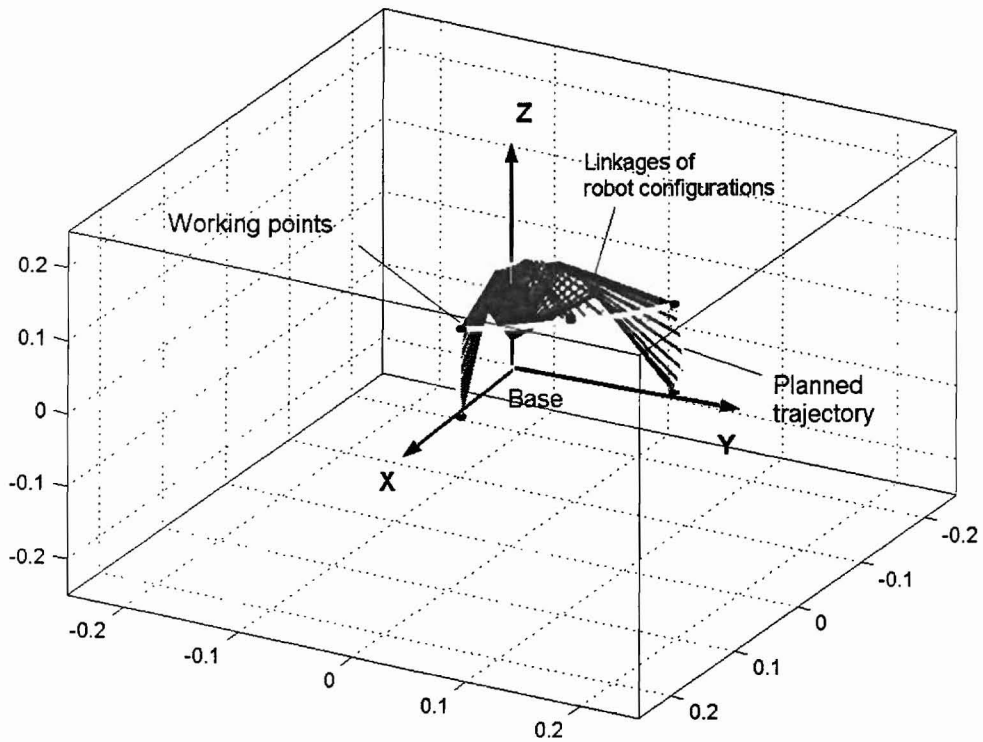
After finishing the iterations, an optimal solution for this situation was determined as shown in Figure 5.11.

*Feasible solution (4)*



**Figure 5.11** Feasible solution (4) consisting of one rotary module and two linear modules (fitness= - 15.9761)

Any further trials to update the design variables in loop 2 did not result in feasible solutions; i.e., no feasible solution can be found with one linear joint module or three linear joint modules for this system. Finally, the synthesis concluded that the optimal solution for this situation is a feasible solution (2). To give an example, Figure 5.12 shows the simulation of the final optimal solution for this task.



**Figure 5.11** Graphical simulation of an optimal configuration

## 5.7 Summary and Discussion

This chapter concerns the formulation of a model and computer program for the task-oriented design of modular robot configurations. The main goal is to develop a general approach to simultaneously perform both type synthesis and dimension synthesis. The generality has been achieved using the GAMR and the OIM. We conclude that a modular robot configuration synthesis should define the design variables at the module level, and a concurrent design with consideration of all task requirements is needed in order to achieve the global solution to the synthesis problem. Another advancement made here is that the proposed formulation of the synthesis problem



enables type synthesis, number synthesis and dimension synthesis simultaneously, which is a long-standing challenge in mechanism synthesis. There is another general observation regarding the optimization problem formulation. The problem topology, in particular the number and types of design variables in an optimization problem for modular robot configuration synthesis, may change with the iterations when type synthesis and dimension synthesis co-exist in one formulation. This research describes the classification of the design variables semantically in the case of modular robot configuration synthesis, which is an important step towards parallel computation processing of this complex optimization problem.

Closely related studies were published by Yang (1999) and Leger (1999). The simultaneous type synthesis and dimension synthesis were presented in Yang (1999), but he does not provide a general method at the problem formulation level. He considered defining virtual design variables that will accommodate new variables during the iterations, which has the limitations, due to the management of virtual variables when variables are produced and removed, and estimation of the dimension of the virtual space. He only considered the production (not removal) of new variables due to an increase of the number of joints and links, the size of the OIM.

The study presented by Leger (1999), with respect to configuration synthesis, can only be regarded as dimension synthesis. When the user selects a robot configuration type, the program determines a set of parameters to make the robot of a chosen type achieve the best performance against the desired tasks specified. When applied to modular robot configuration synthesis, he used a representation of configurations, called PMCG. There

are two difficulties associated with the content of PMCG. *First*, the parameters in the context of PMCG do not represent the information at the module level; therefore, when the solution is found at the PMCG level, a further process that tries to map the parameters at the PMCG level may not find a solution at the module level. *Second*, there is a possibility of losing optimal solutions at the module level, because the PMCG is an abstraction (to a certain degree) of a representation at the module level. Thus formulating a configuration synthesis problem at the PMCG level may impose constraints to the design space at the module level.

## **Chapter 6**

### **Design of an Adaptive Parallel Robot System — A Case Study**

The GAMR was developed in this thesis study (Chapter 3) to increase system adaptability. The main argument underlying the GAMR is that system adaptability can be achieved by adaptive system components (adaptive variables) and/or by modular system architecture (modular variables). One of the uses of the GAMR is that it can serve as a reference model for developing more specific architectures of adaptive systems in the context of specific applications, where other design criteria such as cost, accuracy, load, and mechatronic implementations, may need to be considered. In Chapter 5, an example was taken to demonstrate how the location of the base platform contributed to the fulfillment of the task. While this is indeed a good example to illustrate a contribution from both the modular system concept and the adjustable component concept to increasing system adaptability, the emphasis was more on modular systems. In this chapter, the integration of adaptive variables within a modular Parallel Kinematic Machine (PKM) architecture is addressed with a special emphasis on illustrating the effects of these adjustable parameters to system adaptability in the context of PKM systems.

This chapter is organized as follows. Section 6.1 introduces the PKM system and the

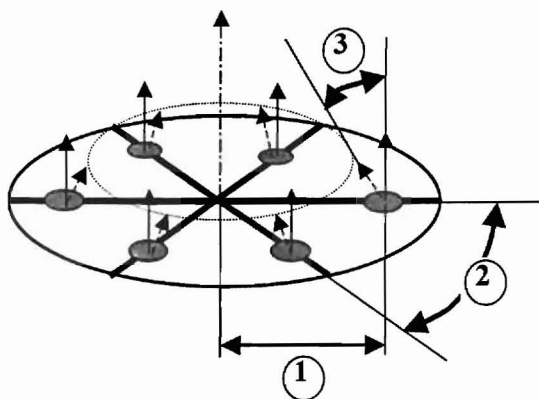
corresponding adjustable platform. Section 6.2 and Section 6.3 discuss the effect of the adjustable platform to system adaptability in terms of performance indices. Section 6.4 discusses related work in the PKM architecture and the PKM with redundancy to highlight the novelty of the adjustable platform concept. Section 6.5 gives a summary and conclusions.

## **6.1 The Adjustable Platform**

The main characteristics of the architecture of PKMs can be summarized as follows (Tsai 1999, Cerantes-Sanchez and Remdon-Sanchiea 1999):

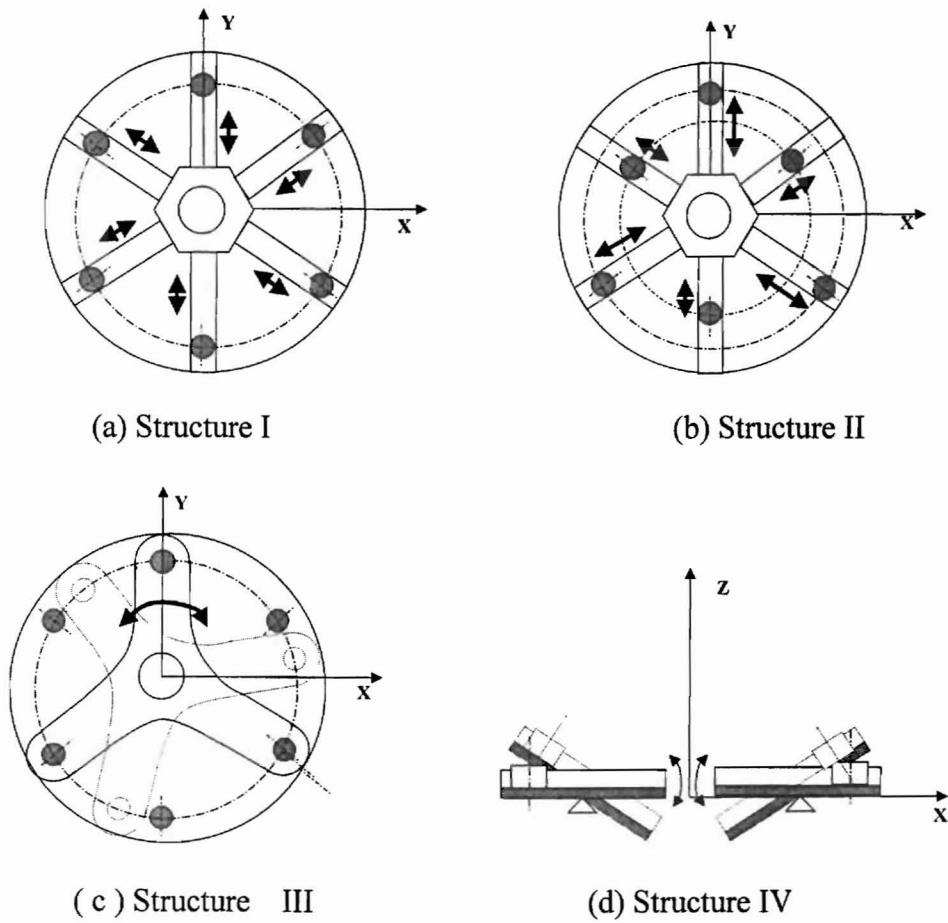
- (i) There is a moving platform that is connected to a fixed base by several symmetrical limbs. The moving platform is used as an end-effector.
- (ii) The number of limbs should be equal to the number of DOF in such a way that only one actuated joint is required for each limb; no actuator has to carry the weight of another, and the load on the moving platform can be shared by all actuators.
- (iii) The actuators are mounted on or near the fixed base. This implies that there is a base-connected rotary or linear joint in each limb, or linear joint, which is adjacent to a base-connected joint.
- (iv) Linear joints must be active because passive linear joints tend to introduce considerable friction, and thus degrade system performance in terms of accuracy.
- (v) Connection points between the limbs and the platform are symmetrically arranged.

Adjustable platforms must be designed to conform to the desired characteristic of the PKM system. The three methods are proposed to change the connection points on the platform: (i) change the longitude distributions of the connection ports on the platform, (ii) change the latitude distributions of the connection ports on the platform, and (iii) change the assembly directions of the connection ports on the platform, see Figure 6.1.



**Figure 6.1** Methods to change platform dimensions

Four structures of platforms can implement these methods, see Figure 6.2a, b, c and d. Both Structure I and Structure II implement the first method. Furthermore, Structure II also allows the changes of the locations of the connection ports in a uniform proportion in such a way that the characteristics of symmetric layout of PKM are maintained. Structure III implements the second method, and Structure IV implements the third method.



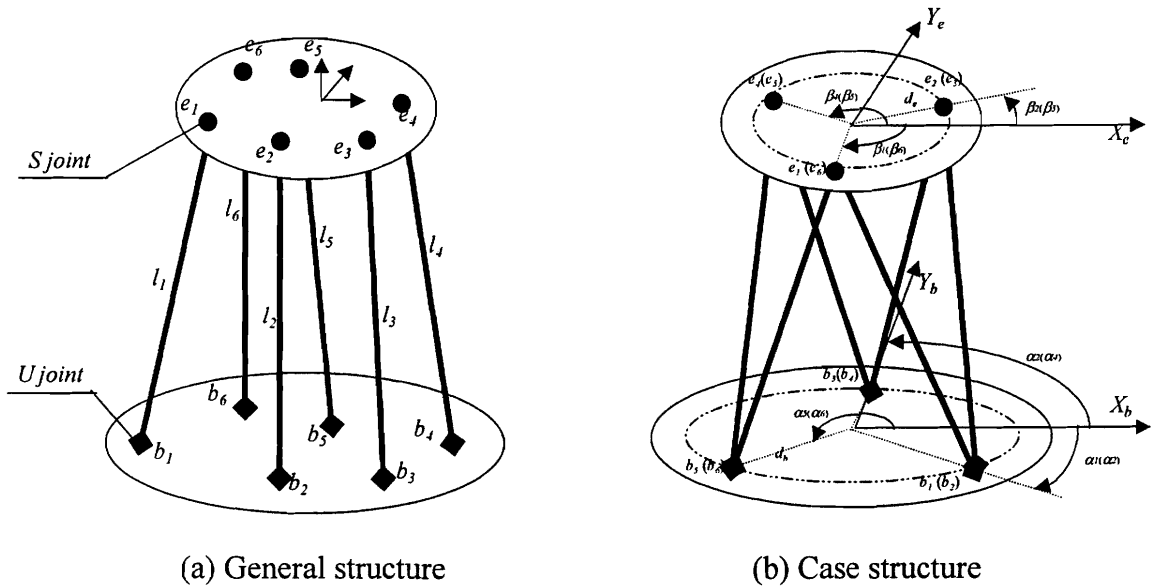
**Figure 6.2 4** Structure types of the adaptive platform

## 6.2 Case Study: Adjustable Platform Structure I (Off-Line)

To demonstrate the enhancement of system adaptability with an adjustable platform, an analysis is given for a Stewart platform robot with an adjustable platform of Structure I. In particular, the system adaptability to translation workspace and system stiffness will be discussed.

### 6.2.1 The Stewart platform robot with the off-line adjustable platform

A general structure of the Stewart platform robot is shown in Figure 6.3a. The robot consists of a fixed platform and an end-effector platform, with six limbs connected between the two platforms. The connection between the limbs and the base platform is a universal joint (U-joint), and that between the limb and the end-effector platform is a spherical joint (S-joint). The connection ports are denoted by  $b_i$  and  $e_i$  ( $i=1,2,\dots,6$ ), respectively.



**Figure 6.3** Stewart platform structure

The definition of geometric parameters of the case structure is shown in Figure 6.3b. The system has the following features. *First*, the connections between the limbs and the platforms are distributed on a circle with radius  $d_b$  on the base platform and a circle with radius  $d_e$  on the end-effector platform. *Second*, each pair of connections is shared at the same location, on the end-effector platform and on the base platform, see Figure 6.3b. These locations are evenly distributed at a longitude direction. As a result, one angular

parameter is sufficient to express each location ( $\alpha_i$  for the base platform and  $\beta_i$  for the end-effector platform). *Third*, all the limbs have an active linear actuator and connect with platforms with a U and S joint; they also share an actuator (on-line) at the adjustable platform to change the distribution in the latitudinal direction. The dimensions of the system are given as follows:

Radius of the end-effector platform for connections	$d_e = 1.0 \text{ inch}$
Radius of the base platform for connections	$d_b = 1.0 - 4.0 \text{ inch}$
Binary limbs	$l_i = 2.5 - 5.0 \text{ inch } (i=1, 2, \dots, 6)$
Connections between the limbs with the base platform in the longitudinal direction	$\alpha_1 = \alpha_2 = -\pi/6$ $\alpha_3 = \alpha_4 = \pi/2$ $\alpha_5 = \alpha_6 = 7\pi/6$
Connections between the limbs with the end-effector platform in the longitudinal direction	$\beta_1 = \beta_6 = -\pi/2$ $\beta_2 = \beta_3 = \pi/6$ $\beta_4 = \beta_5 = 5\pi/6$

### 6.2.2 Kinematics

Suppose that the world coordinate system is coincident with the local coordinate system on the base platform. The orientation of the end-effector is defined by  $(\theta_e, \psi_e, \phi_e)$  through the following procedure (Carretero et al., 2000):

- (i) rotation of angle  $\theta_e$  about the initial Y-direction,
- (ii) rotation of angle  $\psi_e$  about the initial X-direction, and
- (iii) rotation of angle  $\phi_e$  about the initial Z-direction.

The displacement of the end-effector is defined by  $(x_e, y_e, z_e)$ . The expression of the end-effector in the world coordinates is as follows (Carretero et al., 2000):



$$T_e = \begin{bmatrix} c\theta_e c\phi_e + s\psi_e s\theta_e s\phi_e & -c\theta_e s\phi_e + s\psi_e s\theta_e c\phi_e & c\psi_e s\theta_e & x_e \\ c\psi_e s\phi_e & c\psi_e c\phi_e & -s\psi_e & y_e \\ -s\theta_e c\phi_e + s\psi_e c\theta_e s\phi_e & s\theta_e s\phi_e + s\psi_e c\theta_e c\phi_e & c\psi_e c\theta_e & z_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.1)$$

The connections are expressed in the world coordinate system as:

$$\mathbf{b}_i = (d_b \cos \alpha_i, d_b \sin \alpha_i, 0) \quad i = 1, 2, \dots, 6 \quad (6.2)$$

The connections between the limb and the end-effector platform are expressed in the local coordinate system, which are attached to the end-effector:

$${}^e \mathbf{e}_i = (d_e \cos \beta_i, d_e \sin \beta_i, 0) \quad i = 1, 2, \dots, 6 \quad (6.3)$$

Using eqs. (6.1) and (6.2) gives:

$$\mathbf{e}_i^T = T_e {}^e \mathbf{e}_i = \begin{bmatrix} d_e c\beta_i (c\theta_e c\phi_e + s\psi_e s\theta_e s\phi_e) + d_e s\beta_i (-c\theta_e s\phi_e + s\psi_e s\theta_e c\phi_e) + x_e \\ d_e c\beta_i c\psi_e s\phi_e + d_e s\beta_i c\psi_e c\phi_e + y_e \\ d_e c\beta_i (-s\theta_e c\phi_e + s\psi_e c\theta_e s\phi_e) + d_e s\beta_i (s\theta_e s\phi_e + s\psi_e c\theta_e c\phi_e) + z_e \end{bmatrix} \quad i = 1, 2, \dots, 6 \quad (6.4)$$

For each limb,

$$l_i^2 = \|\mathbf{e}_i - \mathbf{b}_i\|^2 \quad i = 1, 2, \dots, 6 \quad (6.5)$$

which implies

$$\begin{aligned}
l_i^2 = & (d_e c \beta_i (c \theta_e c \phi_e + s \psi_e s \theta_e s \phi_e) + d_e s \beta_i (-c \theta_e s \phi_e + s \psi_e s \theta_e c \phi_e) + x_e - d_b \cos \alpha_i)^2 \\
& + (d_e c \beta_i c \psi_e s \phi_e + d_e s \beta_i c \psi_e c \phi_e + y_e - d_b \sin \alpha_i)^2 \\
& + (d_e c \beta_i (-s \theta_e c \phi_e + s \psi_e c \theta_e s \phi_e) + d_e s \beta_i (s \theta_e s \phi_e + s \psi_e c \theta_e c \phi_e) + z_e)^2 \quad (i=1,2,\dots,6)
\end{aligned} \tag{6.6}$$

Eq. (6.6) governs the kinematic behaviors, and also assists in determining the Jacobian matrix:

$$d(\mathbf{l}) = J_{6 \times 6} d(\mathbf{x}) \tag{6.7}$$

where

$$\mathbf{l} = (l_1, l_2, \dots, l_6)^T$$

$$\mathbf{x} = (x_e, y_e, z_e, \theta_e, \psi_e, \phi_e)$$

$$J(1, i) = \frac{\partial l_i}{\partial x_e} = (d_e c \beta_i (c \theta_e c \phi_e + s \psi_e s \theta_e s \phi_e) + d_e s \beta_i (-c \theta_e s \phi_e + s \psi_e s \theta_e c \phi_e) + x_e - d_b \cos \alpha_i) / l_i$$

$$J(2, i) = \frac{\partial l_i}{\partial y_e} = (d_e c \beta_i c \psi_e s \phi_e + d_e s \beta_i c \psi_e c \phi_e + y_e - d_b \sin \alpha_i) / l_i$$

$$J(3, i) = \frac{\partial l_i}{\partial z_e} = (d_e c \beta_i (-s \theta_e c \phi_e + s \psi_e c \theta_e s \phi_e) + d_e s \beta_i (s \theta_e s \phi_e + s \psi_e c \theta_e c \phi_e) + z_e) / l_i$$

$$J(4, i) = \frac{\partial l_i}{\partial \theta_e} = (d_e c \beta_i (c \theta_e c \phi_e + s \psi_e s \theta_e s \phi_e) + d_e s \beta_i (-c \theta_e s \phi_e + s \psi_e s \theta_e c \phi_e) + x_e - d_b \cos \alpha_i)$$

$$\begin{aligned}
& (d_e c \beta_i (-s \theta_e c \phi_e + s \psi_e c \theta_e s \phi_e) + d_e s \beta_i (s \theta_e s \phi_e + s \psi_e c \theta_e c \phi_e)) / l_i \\
& + (d_e c \beta_i (-s \theta_e c \phi_e + s \psi_e c \theta_e s \phi_e) + d_e s \beta_i (s \theta_e s \phi_e + s \psi_e c \theta_e c \phi_e) + z_e)
\end{aligned}$$

$$(d_e c \beta_i (-c \theta_e c \phi_e - s \psi_e s \theta_e s \phi_e) + d_e s \beta_i (c \theta_e s \phi_e - s \psi_e s \theta_e c \phi_e)) / l_i$$

$$J(5, i) = \frac{\partial l_i}{\partial \psi_e} = (d_e c \beta_i (c \theta_e c \phi_e + s \psi_e s \theta_e s \phi_e) + d_e s \beta_i (-c \theta_e s \phi_e + s \psi_e s \theta_e c \phi_e) + x_e - d_b \cos \alpha_i)$$

$$\begin{aligned}
& (d_e c \beta_i (c \theta_e c \phi_e + c \psi_e s \theta_e s \phi_e) + d_e s \beta_i (-c \theta_e s \phi_e + c \psi_e s \theta_e c \phi_e)) / l_i \\
& - (d_e c \beta_i c \psi_e s \phi_e + d_e s \beta_i c \psi_e c \phi_e + y_e - d_b \sin \alpha_i) (d_e c \beta_i s \psi_e s \phi_e + d_e s \beta_i s \psi_e c \phi_e) / l_i
\end{aligned}$$

$$+ (d_e c \beta_i (-s \theta_e c \phi_e + s \psi_e c \theta_e s \phi_e) + d_e s \beta_i (s \theta_e s \phi_e + s \psi_e c \theta_e c \phi_e) + z_e)$$

$$(d_e c \beta_i (-s \theta_e c \phi_e + c \psi_e c \theta_e s \phi_e) + d_e s \beta_i (s \theta_e s \phi_e + c \psi_e c \theta_e c \phi_e)) / l_i$$

$$\begin{aligned}
J(6,i) = \frac{\partial l_i}{\partial \phi_e} = & (d_e c \beta_i (c \theta_e c \phi_e + s \psi_e s \theta_e s \phi_e) + d_e s \beta_i (-c \theta_e s \phi_e + s \psi_e s \theta_e c \phi_e) + x_e - d_b \cos \alpha_i) \\
& (d_e c \beta_i (-c \theta_e s \phi_e + s \psi_e s \theta_e c \phi_e) - d_e s \beta_i (c \theta_e c \phi_e + s \psi_e s \theta_e s \phi_e)) / l_i \\
& + (d_e c \beta_i c \psi_e s \phi_e + d_e s \beta_i c \psi_e c \phi_e + y_e - d_b \sin \alpha_i) (d_e c \beta_i c \psi_e c \phi_e - d_e s \beta_i c \psi_e s \phi_e) / l_i \\
& + (d_e c \beta_i (-s \theta_e c \phi_e + s \psi_e c \theta_e s \phi_e) + d_e s \beta_i (s \theta_e s \phi_e + s \psi_e c \theta_e c \phi_e) + z_e) \\
& (d_e c \beta_i (s \theta_e s \phi_e + s \psi_e c \theta_e c \phi_e) + d_e s \beta_i (s \theta_e c \phi_e - s \psi_e c \theta_e s \phi_e)) / l_i
\end{aligned}$$

The system stiffness at a working point can be calculated from the Jacobian matrix (Gosselin 1990, Tsai 1999). Suppose that the limbs are considered as a flexible truss, and they all have the same cross sectional area. The stiffness matrix is (Gosselin 1990):

$$\mathbf{K} = (\mathbf{J})^T \text{diag}(\mathbf{C}) \mathbf{J} \quad (6.8)$$

where  $\mathbf{C}_i = k_i / l_i$ ,  $l_i$  is the length of the limbs, and  $k_i$  is the stiffness of the limbs.

Assume that all limbs have the same stiffness ( $k_1 = k_2 = k_3 = k_4 = k_5 = k_6 = k$ ); the stiffness matrix becomes:

$$\mathbf{K} = k \mathbf{J}^T \mathbf{J} \quad (6.9)$$

Without loss of generality, assume  $k = 1$ . The stiffness index at a working point is measured by the ratio of the smallest eigenvalue to the largest (Gosselin 1990),

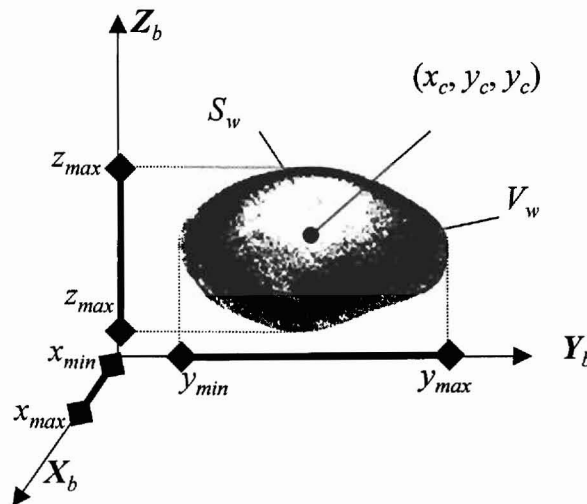
$$\text{stiffness index} = \frac{\lambda_{\min}}{\lambda_{\max}} \quad (6.10)$$

where  $\lambda_{min}$  and  $\lambda_{max}$  are the smallest and largest eigenvalues of the stiffness matrix, respectively.

### 6.2.3 Performance analysis

The translation workspace and the system stiffness were chosen for performance evaluation. The translation workspace is defined as a set of locations that a moving platform can reach when its orientation is fixed (Merlet 2000). For Structure I, the variation of the platform is made through  $d_b$ , see Figure 6.3b. The following indices are examined for the translation workspace, see Figure 6.4, i.e.,

- (i) Volume of the translation workspace:  $V_w$
- (ii) Average stiffness on the translation workspace:  $\bar{S}_w$
- (iii) Geometrical center of the translation workspace:  $(x_c, y_c, z_c)$
- (iv) Coordinate ranges of the translation workspace:  $(x_{min}, x_{max})$ ,  $(y_{min}, y_{max})$  and  $(z_{min}, z_{max})$



**Figure 6.4** Indices to compare the translation workspace

In this example, the fixed orientation is assumed to be:

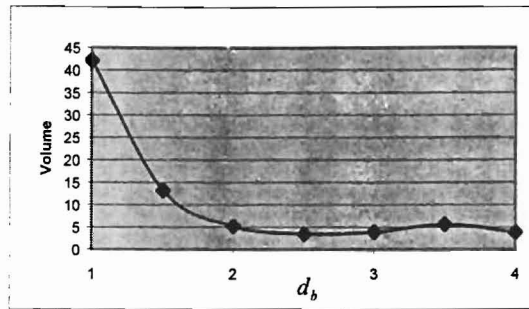
$$(\theta_e, \psi_e, \phi_e) = \left( \frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{4} \right)$$

The results of the calculation for the latter four indices versus  $d_b$  are shown in Figure 6.5a, b, c, d. The program developed by Bi et al. (1994) was employed for the calculation. Figure 6.6 also shows the cross-section shapes of the translation workspace when  $z_e = 2.5, 3.0$ , and  $3.5$ .

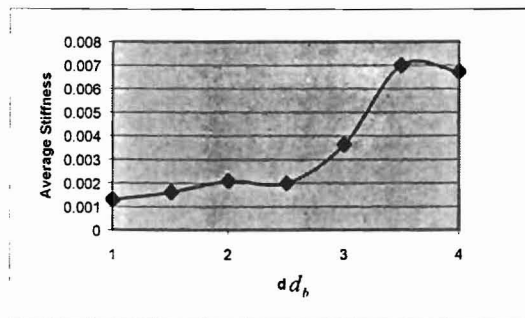
From Figure 6.5a and Figure 6.5b, the smaller platform (with smaller  $d_b$ ) will result in the larger volume of translation workspace, but smaller system stiffness. The change of the shape of the translation workspace due to the variation of the platform is also significant, as implied from Figure 6.5b, c, d, and Figure 6.6.

### 6.3 Case Study: Adjustable Platform Structure I (On-Line)

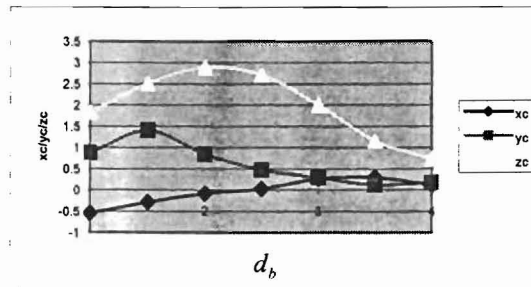
The platform dimension is changed on-line. In this case,  $d_b$  should be treated as a joint variable because it is shared by all limbs. The structure of the system leads to the kinematic redundancy.



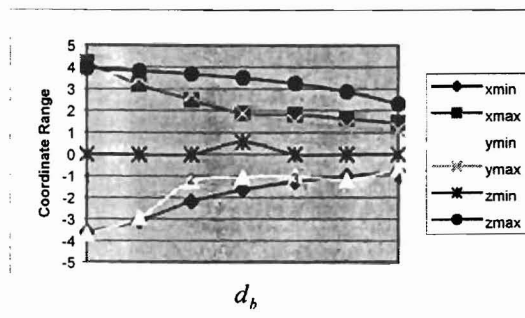
(a) Workspace volume ( $in^3$ )



(b) Average stiffness index

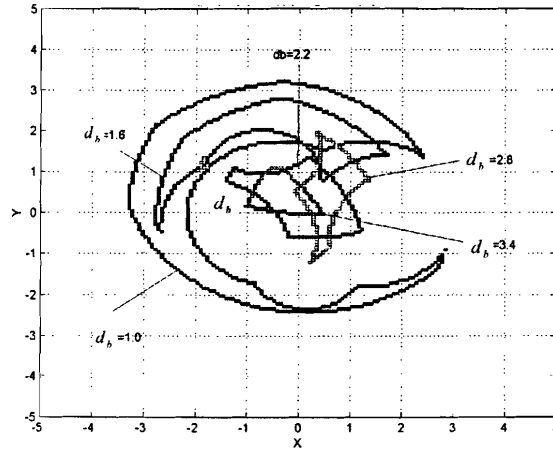


(c) Geometry center

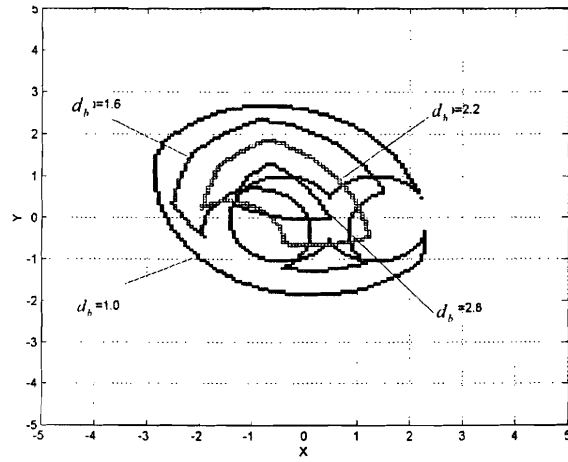


(d) Coordination ranges

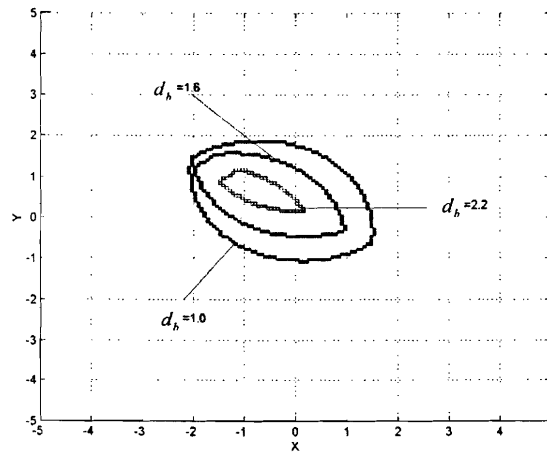
**Figure 6.5** Indices of translation workspace vs.  $d_b$



(a)  $z_e = 2.5$



(b)  $z_e = 3.0$



(c)  $z_e = 3.5$

**Figure 6.6** Cross-section shapes of translation workspaces

### 6.3.1 Kinematic equation and translation workspace

Eqs. (6.1) and (6.2) apply to this situation by observing that  $d_b$  is now a joint variable.

Derivation of eq. (6.1) with respect to time yields:

$$d \begin{pmatrix} l \\ d_b \end{pmatrix} = \tilde{J}_{7 \times 6} d(x) = \begin{pmatrix} J_{6 \times 6} \\ J_{d_b} \end{pmatrix} d(x) \quad (6.11)$$

where  $\tilde{J}_{7 \times 6}$  is the Jacobian matrix for the Stewart robot with an on-line adjustable platform.  $J_{d_b}$  is calculated by:

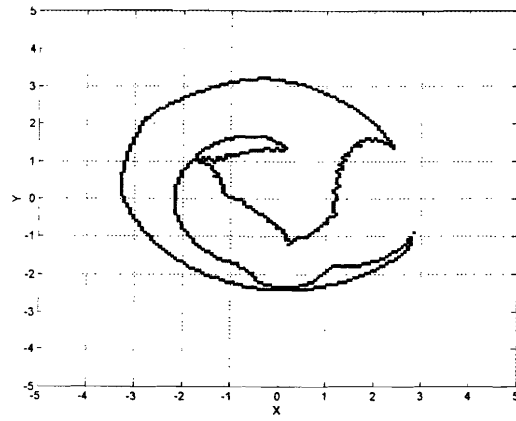
$$\begin{aligned} J_{d_b} &= \tilde{J}(7, i) = \frac{\partial l_i}{\partial d_b} \\ &= \cos \alpha_i (d_e c \beta_i (c \theta_e c \phi_e + s \psi_e s \theta_e s \phi_e) + d_e s \beta_i (-c \theta_e s \phi_e + s \psi_e s \theta_e c \phi_e) + x_e - d_b \cos \alpha_i) / l_i \\ &\quad - \sin \alpha_i (d_e c \beta_i c \psi_e s \phi_e + d_e s \beta_i c \psi_e c \phi_e + y_e - d_b \sin \alpha_i) / l_i \quad (i = 1, 2 \dots 6) \end{aligned} \quad (6.12)$$

For the purpose of comparison, we calculated the total volume of a workspace by specifying the same orientation of translation workspace as that for the off-line adjustable platform, resulting in:

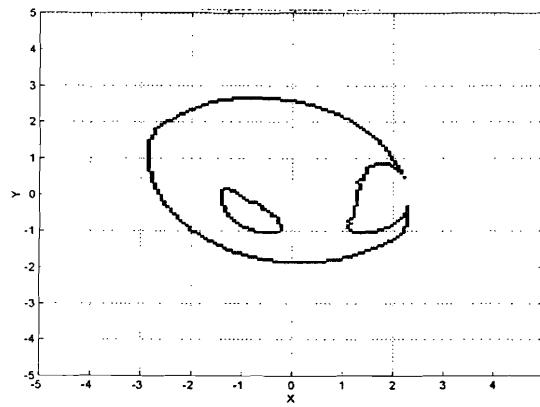
$$V_w \bigg|_{(\theta_e, \psi_e, \phi_e) = \left(\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{4}\right)} = 56.505 \text{ in}^3$$

From Figure 6.5a, we obtain the maximum volume by  $42.074 \text{ in}^3$  when  $d_b=1.0$ . Therefore, with an on-line adjustable platform, i.e., introducing a redundant actuator on the platform, the workspace volume is considerably increased. Furthermore, one can calculate the workspace shape for different  $z_e$ , see Figure 6.7.

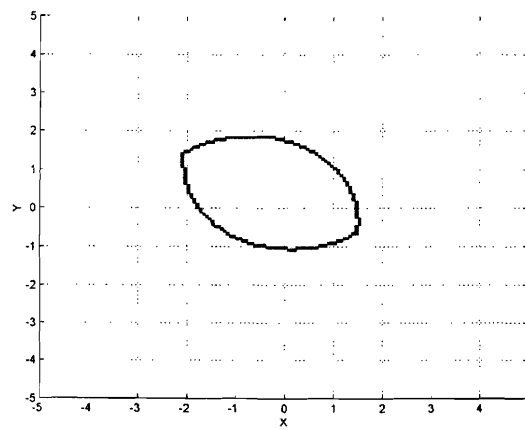




(a)  $z_e=2.5$



(b)  $z_e=3.0$



(c)  $z_e=3.5$

**Figure 6.7** Cross-section shapes of translation workspace

By comparing Figure 6.7 with Figure 6.6, there is a considerable difference in the translation workspaces.

### 6.3.2 Impact on system stiffness by the on-line adjustable platform

An example is shown that the on-line adjustable platform could improve the robot stiffness along a trajectory. The system stiffness at a working point depends largely on the robot configuration. This makes it possible for an adjustable platform to improve the robot system stiffness by changing the robot configuration. Eq. (6.9) is extended to evaluate the system stiffness at working points of the Stewart robot, i.e.,

$$\mathbf{K} = k(\tilde{\mathbf{J}})^T \tilde{\mathbf{J}} \quad (6.13)$$

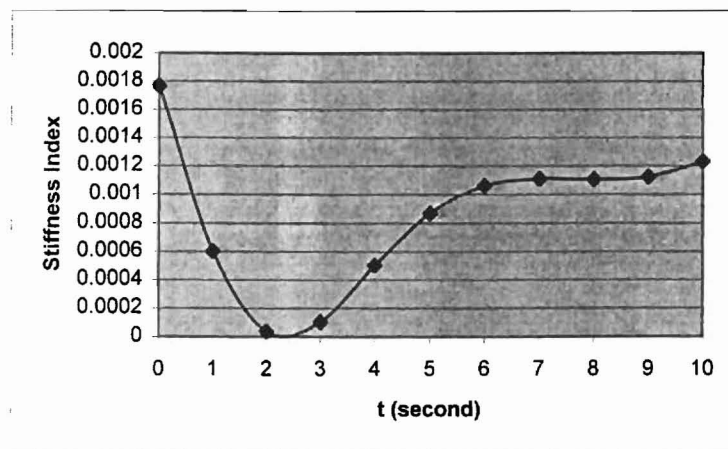
Although the linear joint at the adjustable platform could be made much stiffer than those at the limbs, we set the spring constant  $k_{db}=1$  in order to be comparable with the off-line adjustable platform. Consider a trajectory between the points:

$$x_0 = \left( x_e = -1.0, \quad y_e = 1.3, \quad z_e = 2.75, \quad \theta_e = 0, \quad \psi_e = \frac{\pi}{4}, \quad \phi_e = \frac{\pi}{4} \right)$$

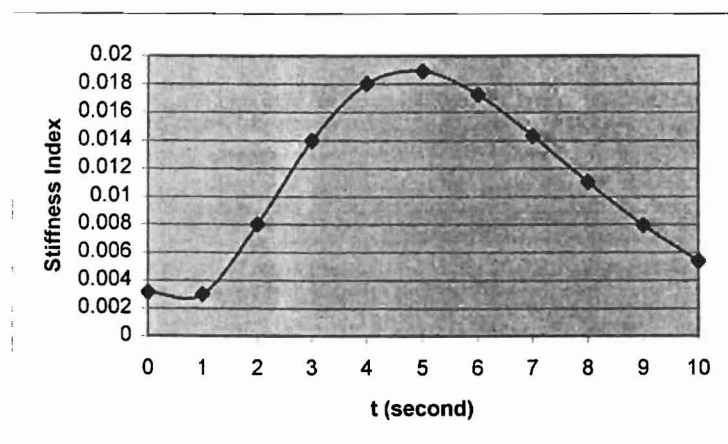
$$x_1 = \left( x_e = 0, \quad y_e = 1.3, \quad z_e = 2.75, \quad \theta_e = \frac{\pi}{2}, \quad \psi_e = \frac{\pi}{4}, \quad \phi_e = \frac{\pi}{4} \right)$$

The trajectory is planned by  $x_0 + \frac{t}{T}(x_1 - x_0)$ , where  $T$  is the total time moving from  $x_0$  to  $x_1$ , and  $t$  is a specific time along the trajectory. The optimizations are carried out to achieve the maximum stiffness along the trajectory for (i) the Stewart robot with an off-line adjustable platform, and (ii) the Stewart robot with an on-line adjustable platform.

Figure 6.8 shows the results of the system stiffness for both the off-line adjustable platform (Figure 6.8a) and the on-line adjustable platform (Figure 6.8b). The best stiffness index along the trajectory is obtained when  $d_b = 1.72$  for the Stewart robot with an off-line adjustable platform, and the average stiffness index along with the trajectory is 0.000866. The Stewart robot with the on-line adjustable platform could obtain much a better stiffness index along the trajectory, see Figure 6.8b. The average stiffness index is 0.011010.



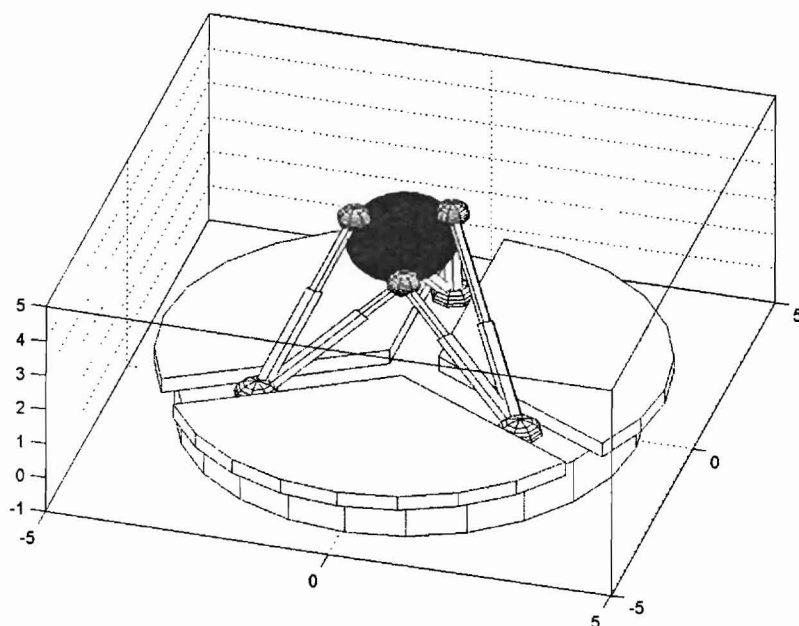
(a) off-line adjustable platform



(b) On-line adjustable platform

**Figure 6.8** System stiffness index of the Stewart robots

The Stewart robot with the on-line adjustable platform is illustrated in Figure 6.9.



**Figure 6.9** Stewart robot with an on-line adjustable platform

### 6.3.3 Effect of actuation platform on end-effector motion

The Stewart robot with an on-line adjustable platform has actuated joints on the limbs and an actuated joint on the platform. The effect of actuator on the platform to the motion of the end-effector is examined. To examine this effect: (i) the measurement of the effect is derived from the relationships between the velocity of the joints and the velocity of the end-effector, and (ii) the effects of the limb joints are regarded as a reference.

As shown in eq. (6.11), the Jacobian matrix completely determines the relationships

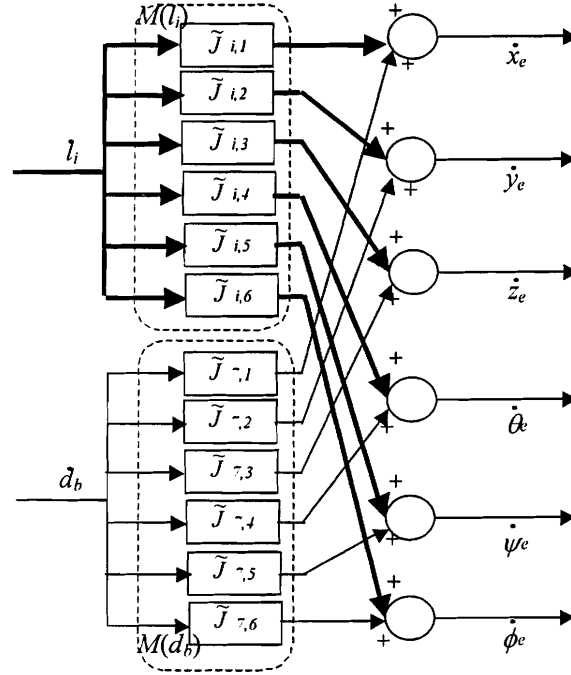
between the velocities of the joints and the velocities of the end-effector. Figure 6.10 shows the relationship between the joint motions and the end-effector motions and the corresponding items of the Jacobian matrix. From Figure 6.10, the elements in the Jacobian matrix are the gains of the joint velocities (input) contributing to the velocities of the end-effector (output). One can view the elements of the Jacobian matrix, which relate  $\dot{l}_i$  to the velocities of limb  $l_i$ , as the components of a gain vector  $M(l_i)$ , and the elements of the Jacobian matrix, which relate  $\dot{d}_b$  to the velocities of the platform motion, as the components of gain vector  $M(d_b)$ . The magnitudes of the two vectors describe the contributions from the actuations of the limbs ( $\dot{l}_i$ ) and the redundant actuation on the platform ( $\dot{d}_b$ ). The two measures based on the gain vectors are therefore:

$$\left. \begin{aligned} M(l_i) &= \sqrt{\sum_{j=1}^{j=6} (\tilde{J}(i, j))^2} \\ M(d_b) &= \sqrt{\sum_{j=1}^{j=6} (\tilde{J}(7, j))^2} \end{aligned} \right\} \quad (6.14)$$

where

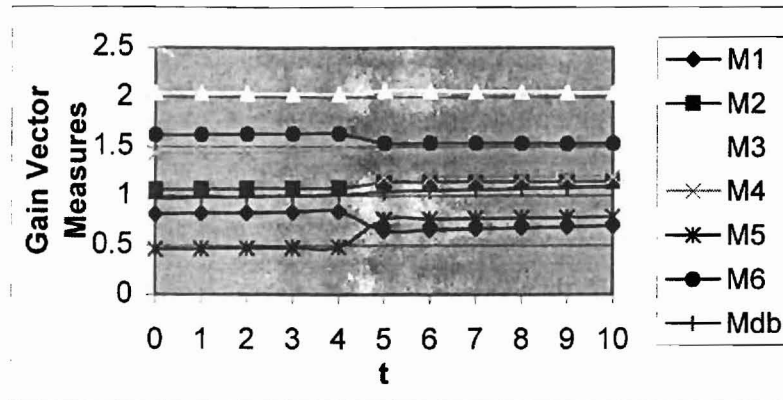
- $M(l_i)$  : Measure of the effect of the limb actuation  $i$  to the end-effector velocity,
- $M(d_b)$  : Measure of the effect of the platform redundant actuation to the end-effector velocity,
- $\tilde{J}(i, j)$  : Elements of the Jacobian matrix when the redundant  
 $(i = 1, \dots, 7; j = 1, \dots, 6)$  actuation on the platform is included into the joint actuation space, and determined by eqs. (6.7) and (6.12)

These effects are functions of the joint variables  $l_i$  and  $d_b$ , which are related to the position and orientation of the end-effector.



**Figure 6.10** Block diagram of the Jacobian matrix

Figure 6.11 shows the results of the calculation of  $M(l_i)$  and  $M(d_b)$ . The effect of the redundant actuator on the velocities of the end-effector is the same order of magnitude as the effects of the limb actuators. It should be noted that the relationships between the velocities of joints and that of the end-effector also reflect the effects of joint motion errors to the motion error of end-effector; the above result implies that the error of the platform motion will not be accumulated, although the platform joint is shared by all of limbs.



**Figure 6.11** Gain vector measures along a trajectory

## 6.4 Related Research

The PKM architectures consist of a set of symmetrical limbs, and two platforms where a moving platform is the end-effector and a fixed platform is the base. About 100 possible architectures have been proposed for PKM with 2-6 DOFs (Merlet 2000), while some architectural variations are distinguished by different connection locations between the limbs and platforms. The dimensions of the limbs and the connection locations between the limbs and the platforms play an important role in determining PKM performance. Ji and Song (1996 and 1998) studied the PKM performance with respect to the location between the limb and the base platform; the results encouraged him to develop a re-configurable parallel robot system. Similar studies were carried out by Merlet (1999), and concluded that a mechanical architecture which might seem to be more appropriate for a given task and whose dimensions have been chosen arbitrarily will perform more poorly than another mechanical architecture whose dimensions have been carefully selected. The conclusion developed by Merlet (1999), however, does not necessarily

lead to the concept of an adjustable platform.

By making the platform adjustable, in particular on-line adjustable, redundancy is added to the PKM. However, the redundant PKM here differs from previous studies on redundant robots both from their purpose and content. Merlet (1996) classified redundant parallel manipulators based on the purposes of the redundancies: (i) solving of forward kinematics, (ii) avoidance of singularity and obstacle, (iii) improvement of force control, and (iv) facilitating of kinematic calibration. Nair (1994) proposed a procedure to obtain a simple forward kinematics by introducing sensors to simplify control algorithm implementation. A study on avoiding singularities by extra actuators was conducted by Nakamura (1991). Liu et al. (2001) proposed the following three methods based on redundancy to eliminate singularities, (i) kinematic redundancy: substitute one of the serial limbs by a parallel limb with more DOFs, (ii) over-constrained method: increase the number of serial limbs, and (iii) over-actuation method: increase the number of actuators. The first method results in a redundant hybrid robot, which loses benefits of PKMs such as geometric symmetry. The other methods have no significant effects on system adaptability because they change neither the geometry of the original limbs nor the nature of their kinematics. Wang et al. (2000) presented a kinematically-redundant manipulator called ParaDex; it is a fully 6-DOF parallel structure with curved passive links integrating a redundant DOF at the end-effector platform. This was a specific design in which the redundancy of PKM could significantly increase the orientation workspace.



## 6.5 Summary and Conclusions

Based on the adaptive system concept, a novel design idea of an adjustable platform for PKM has been proposed. This idea has been realized on a Stewart parallel robot system. Four feasible structures were proposed to implement the adjustable platform, and, in particular, all the limbs symmetrically share the adjustable mechanism on the platform. This arrangement is preferable for static or dynamic balancing of the PKM. It was shown that the platform dimensions have a significant effect on the workspace and the system stiffness over the working trajectories. The adjustable platform can work either on-line or off-line. The on-line adjustable platform introduces redundancy into the system to provide greater system adaptability in terms of translation workspace and system stiffness, and the quality of motion transfer. A very important conclusion is that system adaptability could also be significantly enhanced by introducing adjustable components and parameters.

## **Chapter 7**

### **Summary, Conclusions, and Future Work**

#### **7.1 Summary**

The motivation of this thesis research is based on an emerging concern in manufacturing that system adaptability is the most important factor for the manufacturing systems; yet there is insufficient development in both theory and methodology for adaptive systems. Previous research was found to move along two separate directions to achieve system adaptability: (i) flexible system concept and (ii) modular system concept. Previous research on modular manufacturing systems, concerning mostly to modular robot systems, only considers item (ii). This thesis married these under the same umbrella, system adaptability enhancement. In particular, this thesis took a general approach of adding adjustable components/parameters to the system modular architecture.

The thesis addressed the following issues: (i) general architecture for modular robot systems incorporating adjustable parameters, (ii) automatic modeling of modular robot configurations, (iii) modular robot configuration synthesis, and (iv) integration of adjustable platforms into modular PKM architecture as a case study.

A literature review was conducted to show the significance of addressing these issues. This has led to the development of a general architecture of modular robots, which is a foundation for modular robot configurations. Modular robot configurations are a time varying motion system. Therefore, kinematic and dynamic analysis, in particular the computer-aided generation of kinematic and dynamic behavior, was studied. This study led to the implementation of a computational method to perform this task. Modular robot configuration synthesis was modeled as an optimization problem. This problem is very complex in terms of the dimension of the variables and the presence of both continuous and discrete variables. An optimization based on a Genetic Algorithm (GA) was applied. It was also observed that the optimization problem, in the context of modular robot configuration synthesis, is such that the problem topology (the number and types of variables) could change with the solution process. A strategy to cope with such a situation was developed. Finally, the benefit of the adaptive system concept was further shown by designing a modular PKM system with an adjustable platform. We demonstrated how the adjustable platform affects the translation workspace and the system stiffness.

## **7.2 Conclusions**

In general, the research documented in this thesis has demonstrated that the research objectives set out in chapter 1 can be achieved. A more detailed elaboration on this general statement is given as follows:

- (i) Regarding objective 1, a general conceptual architecture of modular robots (GAMR)

with added adjustable parameters is a foundation for adaptive systems. This architecture has the most extensive coverage of features among previously published architectures. The significance of the GAMR is that: (i) it provides a benchmark to evaluate system adaptability of modular robot systems, and (ii) GAMR-based modular robot configuration synthesis could be applied to any specific modular robot system. A new representation of modular robot configurations, called the Object Incidence Matrix (OIM), was developed. In comparison with a similar matrix developed by Chen (1994), the OIM contains a richer semantics of adaptive robot systems.

- (ii) Regarding objective 2, a systematic approach was developed for the automatic kinematic and dynamic modeling of modular robot configurations. The approach follows an indirect strategy whereby a kinematically independent representation of configurations of a modular robot is first converted into a kinematically dependent representation and then kinematically dependent representation is applied for calculating the kinematic and dynamic behaviors. In comparison with previous research, the approach developed in this thesis has included dynamic modeling. The approach is applicable to adaptive robot systems with all module types that were published in the literature, including modules with adjustable parameters.
- (iii) Regarding objective 3, a new observation was made for modular system configuration synthesis. There is a necessity to consider design goals (kinematic and dynamic) concurrently. We concluded that the optimization problem model for modular robot configuration synthesis could create a situation where the number and types of variables are changing during the iteration process for optimal solutions when both type synthesis and dimension synthesis are considered. A strategy was proposed to cope with this

problem by designing three-loop operations based on the three levels of information in the OIM representation. This strategy is more flexible and efficient in comparison with that reported by Yang (1999), in which the basic idea is to set ‘extra’ variables for possible new variables produced during the iteration process and these extra variables are given default values when they are not needed.

- (iv) Regarding objective 4, it can be concluded that the architecture that adds the adjustable parameters on the top of modular system concept can further improve system performance of modular parallel manipulator systems. This conclusion is drawn from a new concept of parallel manipulator system, i.e., a novel modular PKM with an adjustable platform. It was shown that the platform dimensions play a significant effect on increasing the workspace and the stiffness over the working points. Four feasible structures were proposed to fulfill this requirement. The actuator on the adjustable platform is shared by all the limbs symmetrically. This is preferable for static or dynamic balancing of the PKM. The adjustable platform can operate in either an on-line or off-line mode according to the design constraints. The redundancy will be introduced in when the adjustable platform operates in an on-line mode to further enhance system adaptability.

### **7.3 Future Work and Discussion**

The work described in this thesis has limitations. *First*, the dynamic control of the modular robot system has not been addressed. The issues involved in dynamic control include: (i) development of effective methods for task-level control to eliminate

positioning error, and (ii) incorporation of control behavior into configuration synthesis, which follows the design for control methodology for general mechatronic systems (Li et al. 2001). *Second*, implementation of the optimization model for modular robot configuration synthesis is ad-hoc and lacks generality, though computational overhead has been alleviated. Future studies should overcome these shortcomings and extend the present work as discussed below.

### **(i) Automatic kinematic and dynamic modeling**

The D-H based modeling methodology is preferable for robot configurations with a serial or tree-like structure, but it becomes inefficient when a robot configuration has parallel or closed-loop structures. This observation was also made by Tsai (1999). At this point, a method based on a special finite element formulation (Yang and Sadler 1990, Zhang 1994) for non-modular robot systems should be considered.

### **(ii) Integration of design and control**

The design variables may have a significant effect on control performance, as observed in non-modular robot systems (Zhang et al. 1999). Such an effect can likely take place in modular robot systems. Therefore, one should treat design and control simultaneously in modular robot system configuration synthesis. Also, other design issues, such as vibration control, and force/moment balancing, need to be considered concurrently with real-time control.

### **(iii) Task-level control**

The goal here is to eliminate positioning error at the end-effector of a modular robot system. The current strategy for operation at the joint module level needs to be improved to become more adaptive to the computational time needed for inverse kinematic analysis.

### **(iv) Computationally efficient method for modular robot configuration synthesis**

In Chapter 5, it was shown that the optimization problem model for modular robot configuration synthesis exhibits design variables that are highly heterogeneous in their semantics; i.e., the coexistence of type-related variables and parameter-related variables. Because type-related variables decide parameter-related variables, in the sense that a particular set of dimensions depends on module types, the topology of optimization problem model may change with the iteration process. That is to say, the updating of a type-related variable may lead to the production or removal of some parameter-related variables.

One of the basic steps to tackle this problem is to classify variables from the viewpoint of an application. Modular robot configuration synthesis has been developed in Chapter 5 using a three-loop procedure. However, there may be significant computational overhead, see the discussion in Appendix C. Further research is needed to reduce the computational overhead.

## References

- Altus, Stephen S., Kroo, Iian M., and Gago, Peter J., 1996. "A Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems," *Transactions of the ASME, Journal of Mechanical Design*, Vol. 118, pp. 486-489.
- Angeles, J., 1997. *Fundamentals of Robotic Mechanical Systems*, New York: Springer-Verlag.
- Asada, H. and Slotine, J.-J. E., 1985. *Robot Analysis and Control*, Wiley-Interscience, New York: John Wiley & Sons Inc.
- Benhabib, B., Zak, G., and Lipton, G., 1989. "A Generalized Kinematic Modeling Method for Modular Robots," *J. of Robotics System*, Vol. 6, No. 5, pp. 545-571.
- Benhabib, B. and Dai, M. Q., 1991. "Mechanical Design of a Modular Robot for Industrial Applications," *Journal of Manufacturing Systems*, Vol. 10, No. 4, pp. 297-304.
- Benhabib, B., Cohen, R., Lipton, M. G., and Dai, M. Q., 1992. "Conceptual Design of a Modular Robot," *ASME Journal of Mechanical Design*, Vol. 114, pp. 117-125.
- Bererton, Curt and Khosla, Pradeep, 2001. "Towards a Team of Robots with Reconfiguration and Repair Capabilities," *Proceedings of the 2001 IEEE international Conference on Robotics and Automation*, Seoul, Korea, May 21-26, pp. 2923-2928.
- Bi, Z. M., Wang, S. G., Zhao, C. X., and Cai, H. G., 1994. "Industrial Robot Kinematic Design System," *2nd Asian Conference on Robotics & Applications*



(ACRA), Beijing, October 13-15, pp. 200-204.

- Bi, Z. M., Li, Y. F., and Zhang, W. J., 1997. "A New Method for Dimensional Synthesis of Robot Manipulators," *Fifth Applied Mechanisms and Robotics Conference*, Cincinnati, Ohio, U.S.A., 1997, Oct. 12-15, Vol. 2, AMR97-041.
- Bi, Z. M. and Zhang, W. J., 2001a. "Modularity Technology in Manufacturing: Taxonomy and Issues," *International Journal of Advanced Manufacturing Technology*. Vol. 8, No. 5, pp. 381-390.
- Bi, Z. M. and Zhang, W. J., 2001b. "Concurrent Optimal Design of Modular Robotic Configuration," *Journal of Robotic Systems*, Vol. 18, No. 2, pp. 77-87.
- Bi, Z. M. and Zhang, W. J., 2001c. "Flexible Fixture Design and Automation: Review, Issues and Future Directions," *International Journal of Production Research*, Vol. 39, No. 13, pp. 2867-2894.
- Bi, Z. M., 2001d. "Experiments of Modular Robotic Systems," *Report for Ph.D. Committee Meeting (No.5)*. Unpublished report, Dept. of mechanical Engineering, University of Saskatchewan, Saskatoon, Canada.
- Bonev, Ilian, 2000. *Delta Parallel Robot - the Story of Success*, <http://www.parallemic.org/Reviews/Review002.html>.
- Bottema, O. and Roth, B., 1979. *Theoretical Kinematics*, North-Holland, Amsterdam: North Holland Publishing Company.
- Bordoloi, Sanjeev K., Cooper, William W., and Matsuo, Hirofumi, 1999. "Flexibility, Adaptability, and Efficiency in Manufacturing Systems," *Production and Operations Management*, Vol. 8, No. 2, pp. 133-150.

- Breteler, A. J. Klein, 1990. *Kinematics Optimization of Mechanisms, Finite Element Approach*, Doctoral Thesis, Delft Technical University, Delft, The Netherlands.
- Brockett, R. W., 1984. "Robotic Manipulators and the Product of Exponentials Formula," *Proc. Symposium on the Mathematical Theory of Networks and Systems*, Beer Shebar, Isreal, pp. 120 –129.
- Campbell, Matthew I., Cagan, Jonathan, and Kotovsky, Kenneth, 1999. "A-Design: an Agent-Based Approach to Conceptual Design in a Dynamic Environment", *Research in Engineering Design*, Vol .11, pp. 172-192.
- Carretero, J. A., Podhorodeski, R. P., Nahon, M. N., and Gosselin, C. M., 2000, "Kinematic Analysis and Optimization of a New Three Degree-of-Freedom Spatial Parallel Manipulator," *ASME Journal of Mechanical Design*, Vol. 122, pp. 17-24.
- Castelli, V. Parenti, 1999. "Classification and Kinematic Modeling of Full-Parallel Manipulators – a Review," *Parallel Kinematics Machines: Theoretical Aspects and Industrial Requirements* (ed. By Boer C. R., Molinari-Tosatti L. and Smith K. S.), pp. 51-67.
- 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, USA.
- Chen, I-Ming and Burdick, Joel W., 1995. "Determining Task Optimal Modular Robot Assembly Configurations," *IEEE International Conference on Robotics and Automation*, Nagoya, Aichi, Japan, May 21-27, pp. 132-137.
- 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.

- Chen, I.-M. and Yang Guilin, 1998. “Automatic Model Generation for Modular Reconfigurable Robot Dynamics,” *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol 120, No. 3, pp. 346-352.
- Chen, I-Ming, Yeo, Song Huat, Guang, Chen, and Guilin, Yang, 1999. “Kernel for Modular Robot Applications: Automatic Modeling Techniques,” *The International Journal of Robotics Research*, Vol. 18, No. 2, pp. 225-242.
- Chen, I-Ming, 2001. “Rapid Response Manufacturing through a Rapidly Reconfigurable Robotic Workcell,” *Robotics and Computer Integrated Manufacturing*, Vol. 17, No. 3, pp. 199-213.
- 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.
- Chocron, O. and Bidaud, P., 1997. “Genetic Design of 3D Modular Manipulators,” *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, pp. 223-228.
- Corke, P. I., 1996. “A Robotics Toolbox for MATLAB”, *IEEE Robotics and Automation Magazine*, Vol. 3, No.1, pp. 24-32.
- Denavit, J. and Hartenberg, R. S., 1955. “Kinematic Notation for Low-Pair Mechanisms Based on Matrices,” *ASME Journal of Applied Mechanics*, Vol. 2, pp. 215-211.

- Duffy, J., 1980. *Analysis of Mechanisms and Robot Manipulators*, John Wiley & Sons.
- Erixon, Gunnar, 1998. *Modular Function Deployment--a Method for Product Modularisation*, Doctoral Thesis, Royal Institute of Technology, Stockholm, Sweden.
- Freudenstein, F. and Primrose, E. J. F., 1984. "On the Analysis and Synthesis of the Workspace of a Three-link, Turning-pair Connected Robot Arm," *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 106, pp. 365-370.
- Fryer, A. James, Mckee, Gerard T., and Schenker, Paul S., 1997. "Configuring Robots from Modules: an Objected Oriented Approach," *8<sup>th</sup> International Conference on Advanced Robotics*, Monterey, California, U.S.A., pp. 907-912.
- Goldenberg, A. A. and Emami, M. R., 1999. "Kinematics and Dynamics of Robot Manipulators," *Handbooks of Industrial Robotics*, Second Edition, edited by Nof. Shimon Y., New York: John Wiley & Sons, Inc.
- Gosselin, C., 1990. "Stiffness Mapping for Parallel Manipulators," *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 3, pp. 377-382.
- Habibi, S. R. and Goldenberg, A. A, 1995. "Design and Control of a Reconfigurable Industrial Hydraulic Robot," *IEEE Int. Conf. On Robotics and Automation*, Nagoya, Aichi, Japan, May 21-27, pp. 2206-2211.
- Halperin, Dan 1986. *Kinematic Modeling of Robot Manipulators and Automatic Generation of Their Inverse Kineamtics Solutions*, M.Sc. Thesis, Tel-aviv University, Tel Aviv, Israel.

- Han, Jeongheon, Chung, W. K., Youm, Y., and Kim, S. H., 1997. "Task Based Design of Modular Robot Manipulator Using Efficient Genetic Algorithm," *1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico., Part 1 (of 4), April 20-25, IEEE, pp. 507-512.
- Hong, Namhee Kim and Hong, Sunggul, 1998. "Entity-Based Models for Computer-Aided Design Systems," *Journal of Computing in Civil Engineering*, Vol. 12, No. 1, pp. 30-41.
- Hooper, R. and Tesar, D., 1994. "Computer-Aided Configuration of Modular Robotics Systems," *Computing & Control Engineering J.*, June, pp. 137-142.
- Jaenisch, Uwe, Miertschink, Markus, Sperber, Wilfried, and Tschakarow, Roland, 2000. *Power&Cube Operating Manual*, Version. 1.10, AMTEC GmbH, Germany.
- Ji, Z., 1996. "Analysis of Design Parameters in Platform Manipulators," *ASME Journal of Mechanical Design*, Vol. 118, No. 4, pp. 526-531.
- Ji, Z. and Song, Philip, 1998. "Design of a Reconfigurable Platform Manipulator," *Journal of Robotic Systems*, Vol. 15, No. 6, pp. 341-346.
- Kelmar, Laura and Khosla, P. K., 1990. "Automatic Generation of Forward and Inverse Kinematics for Reconfigurable Modular Manipulator Systems," *J. Robotic Systems*, Vol .7, No. 4, pp. 599-619.
- Kinya, Tamarki, 1993. "Design Technique of Joint Configurations for Modularised Construction Robots," *24th International Symposium on Industrial Robots (ISIR)*, Tokyo, Nov. 1-5, pp. 551-554.
- Leger, Chris, 1999. *Automated Synthesis and Optimization of Robot Configurations*:

*An Evolutionary Approach*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

- Li, Q., Zhang, W. J., and Chen, L., 2001. "Design For Control (DFC): a Concurrent Engineering Approach for Mechatronic System Design," *IEEE/ASME Transaction on Mechatronics*, Vol. 6, No. 2, pp. 161-169.
- Liu, G. F., Wu, Y. L., Kuen, Y. Y., and Li, Z. X., 2001. "Analysis and Control of Redundant Parallel Manipulators," *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 21-26, pp.3748-3754.
- Matsumaru, T., 1995. "Design and Control of the Modular Robot Systems: TOMMS," *IEEE Int. Conf. On Robotics and Auto.*, Nagoya, Aichi, Japan, May 21-27, pp. 2125-2131.
- Mechrabi, M. G., Ulsoy, A. G., and Koren, Y., 2000. "Reconfigurable Manufacturing Systems," *Journal of Intelligent Manufacturing*, Vol. 11, pp. 403-419.
- Merlet, J. P., 1996, "Redundant Parallel Manipulators," *Laboratory Robotics and Automation*, Vol. 8, pp.17-24
- Merlet, J. P., 1999a. "Parallel Robot: Open Problems," *9th Int. Symp. of Robotics Research*, 9-12 October 1999, Snowbird.
- Merlet, J. P., 1999b. "Determination of 6D Workspaces of Gough-Type Parallel Manipulator and Comparison between Different Geometries," *Intentional Journal of Robotics Research*, Vol. 18, No. 9, pp.902-916.
- Merlet, J. P., 2000. *Parallel Robots*, Dordrecht, Hardbound: Kluwer Academic Publishers.

- Mulders, P. C., Kreffer, G. J., Bax, W. H. M., and Van Der Wolf, A. C. H., 1993. "Design and Modeling of a Modular Robot system," *Annals of the CIRP*, Vol. 42, No. 1, pp. 429-432.
- Nakamura, Y., 1991. *Advanced Robotics: Redundancy and Optimization*, Boston: Addison-Wesley Publishing Company.
- Nair, P., 1994, "On the Forward Kinematics of Parallel Manipulators," *International Journal of Robotics Research*, Vol. 13, No. 2, pp. 171-188.
- Neos Robotics, 2001. *Tricipt TR600*, [http:// www.neosrobotics.com](http://www.neosrobotics.com).
- Nielsen, E. and Huppi, R., 1992. "MODRO: A Modular Concept for an Application Adapted Robotics System," *Proceedings 23rd International Symposium on Industrial Robots*, October 6-9, Barcelona, Spain, pp. 495-500.
- Nnaji, Bartholomew O., 1986. *Computer-Aided Design, Selection, and Evaluation of Robots*, The Netherlands: Elsevier Science Pub. Co.
- NRC, 2000. *Manufacturing Innovations: An Impact Strategy for 2000-2005*, NRC Manufacturing Technologies Group, <http://web.mtg.nrc.ca>.
- Nolfi, Stefano and Floreano, Dario, 2000, *Evolutionary Robotics*, The MIT Press, Cambridge, Massachusetts: MIT Press/Bradford Books.
- Paredis, C. J. J. and Khosla, P. K. 1991. "An Approach for Mapping Kinematic Task Specifications into a Manipulator Design," *Proc. of 5th International Conference on Advanced Robotics*, Vol. 1, pp. 556-561, Pisa, Italy, June 1991.
- Paredis, C., J. J. and Khosla, P. K., 1993a. "Serial Link Manipulators from Task

Specifications,” *International of Robotics Research*, Vol. 12, No. 3, pp. 274-287.

- Paredis, C. J. J. and Khosla, P. K., 1993b. "Synthesis Methodology for Task Based Reconfiguration of Modular Manipulator Systems," *Proc. of International Symposium on Robotics Research*, Hidden Valley, PA, Oct. 2-5, 1993.
- Paredis, C., Brown H., and Khosla P., 1996. "A Rapidly Deployable Manipulator System," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Vol. 2, April, 1996, pp. 1434-1439.
- Paredis, C. J. J., 1996. *An Agent-Based Approach to the Design of Rapidly Deployable Fault Tolerant Manipulators*, Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, USA
- Park, F. C., 1995. "Optimal Robot Design and Different Geometry," *ASME Journal of Mechanical Design*, Vol. 117, pp. 87-91.
- Park, J. H. and Asada, H., 1994. "Concurrent Design Optimization of Mechanical Structure and Control for High Speed Robots," *ASME J. Dynamics System, Measurement, and Control*, Vol.116, pp. 344-356.
- Paul, R. P., 1981. *Robot Manipulators - Mathematics, Programming and Control*, MIT Press, Massachusetts: MIT Press.
- Perremans, P., 1996. "Feature-Based Description of Modular Fixturing Element: the Key to An Expert System for the Automatic Design of the Physical Fixture," *Advances in Engineering Software*, Vol. 25, pp.19-27.
- Prasad, Biren, 1996. *Concurrent Engineering Fundamentals: Integrated and Process Organization*, Upper Saddle River, New Jersey: Prentice Hall, Inc.



- Pritschow, G. and Wurst, K.-H., 1996. "Modular Robots for Flexible Assembly," *Proceedings of the 28th CIRP International Seminar on Manufacturing Systems "Advances in Manufacturing Technology - Focus on Assembly Systems"*, May 15-17, 1996, Johannesburg, South Africa, pp.153-158.
- Pritschow, G., 1999. "Research and Development in the Field of Parallel Kinematic Systems in Europe," *Parallel Kinematics Machines: Theoretical Aspects and Industrial Requirements* (ed. By Boer C. R., Molinari-Tosatti L. and Smith K. S.), pp.3-16.
- Ramachandran, S. and Chen, I.-M., 2000. "Distributed Agent Based Design of Modular Reconfigurable Robots," *5th International Conference on Computer Integrated Manufacturing*, Singapore, pp. 447-458.
- Rasteger, J. S., Liu, L., and Yin, D., 1999. "Task-Specific Optimal Simultaneous Kinematic, Dynamic, and Control Design of High-Performance Robotic Systems," *IEEE/ASME Transactions on Mechatronics*, Vol. 4, No.4, pp.387-395.
- Schmitz, D. E., Khosla, P. K., and Kanada, T., 1989. "The CMU Reconfigurable Modular Manipulator System," *Proc. 19<sup>th</sup> Int. Symp. Exp. Rob. (ISIR)*, Nov. 6-10, Sydney, New York, Springer-Verlag, pp. 473-388.
- Shakeri, Cirrus, 1998. *Discovery of Design Methodologies for the Integration of Multi-disciplinary Design Problems*, Ph.D. Thesis, Worcester Polytechnic Institute.
- Smith, Preston G. and Reinertsen, Donald G., 1998. *Developing Products in Half the Time: New Rules, New Tools*, New York: John Wiley & Sons.
- Sims, K. 1994. "Evolving Virtual Creatures," *Computer Graphics (Siggraph'94*

*Proceedings*), July 1994, pp.15-22.

- Stake, R. B., 1999. *A Hierarchical Classification of the Reasons for Dividing Products into Modules- A Theoretical Analysis*, Licentiate Thesis, Div. of Assembly Systems, Dept. of Manufacturing Systems, Royal Institute of Technology, Sweden.
- Suh, N. P., 1990. *The Principles of Design*, Oxford, New York: Oxford University Press.
- Suh, N. P., 1998. "Axiomatic Design Theory for System," *Research in Engineering Design*, Vol. 10, pp.189-209.
- Tesar, D. and Butler, M., 1989. "A Generalized Modular Architecture for Robot Structures," *Manufacturing Review*, Vol. 2, pp. 91-117.
- Tonshoff, H. K. and Grendel, H., 1999. "A System Comparison of Parallel Kinematics," *Parallel Kinematics Machines: Theoretical Aspects and Industrial Requirements* (ed. By Boer C. R., Molinari-Tosatti L. and Smith K. S.), pp.295-312.
- Tsai, L. W., 1999. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*, New York: John Wiley & Sons, Inc.
- Tsai, Yue-Tern and Wang, Kuo-Shong, 1999. "The Development of Modular-Based Design in Considering Technology Complexity," *European Journal of Operational Research*, Vol. 119. pp. 692-703.
- Uchiyama, Masaru, 1994. "Structures and Characteristics of Parallel Manipulators," *Advanced Robotics, The International Journal of the Robotics Society of Japan*, Vol. 8, No.6, pp. 545-557.

- Ulrich, K. 1995, "The Role of Product Architecture in the Manufacturing Firm," *Research Policy*, Vol. 24, pp. 419-440.
- Walischmiller, W. and Frager, O., 1994. "TELBOT- A Modular Tele-Robot System with Infinite Rotational Motions of Any Joint, Open Control System Architecture and Integrated 3-D On-Offline Simulation," *Proceedings of the 25th International Symposium on Industrial Robots*, April 25-27, Hannover, Germany, pp. 627-633.
- Wampler, C.W., Morgan, A. P., and Sommese, A. J., 1990. "Numerical Continuation Methods for Solving Polynomial Systems Arising in Kinematics," *ASME J. on Mechanical Design*, Vol.112, pp. 59-68.
- Wang, Yuesong, Newman, Wyatt S., and Stoughton, Robert S., 2000. "Workspace Analysis of the ParaDex Robot – a Novel, Closed-Chain, Kinematically-Redundant Manipulator," *Proceedings of the 2000 IEEE, International Conference on Robotics & Automation*, San Francisco, CA, April 2000, pp. 2392-2397.
- Watson I., 1999. "Case-Based Reasoning is a Methodology not a Technology," *Knowledge-Based Systems*, Vol. 12, pp.303-308.
- Wurst, K. H., 1986. "The Conception and Construction of a Modular Robot System," *Proc. 16<sup>th</sup> Int. Sym. Industrial Robotics (ISIR)*, Belgium, September, pp. 37-44.
- Wurst, K. H., 1999. "LINAPOSS – Machine Tools as Parallel Link Systems Based on Modular Design," *Parallel Kinematic Machines: Theoretical Aspects and Industrial Requirements*, (Boer, C. R. Molinari-Tosatti and K. S. Smith Eds.), Advanced Manufacturing Series, Springer-Verlag London Limited, pp. 377-394.
- Xi, Fengfeng, Ross, Andrew, and Lang, Sherman, 2001. "Exploring a Re-Configurable

Parallel Robot for Space Applications,” *Proceedings of the 6<sup>th</sup> International Symposium on Artificial Intelligence and Robotics & Automation in Space: I-SAIRAS*, Canadian Space Agency, Quebec, Canada, June 18-22, 2001.

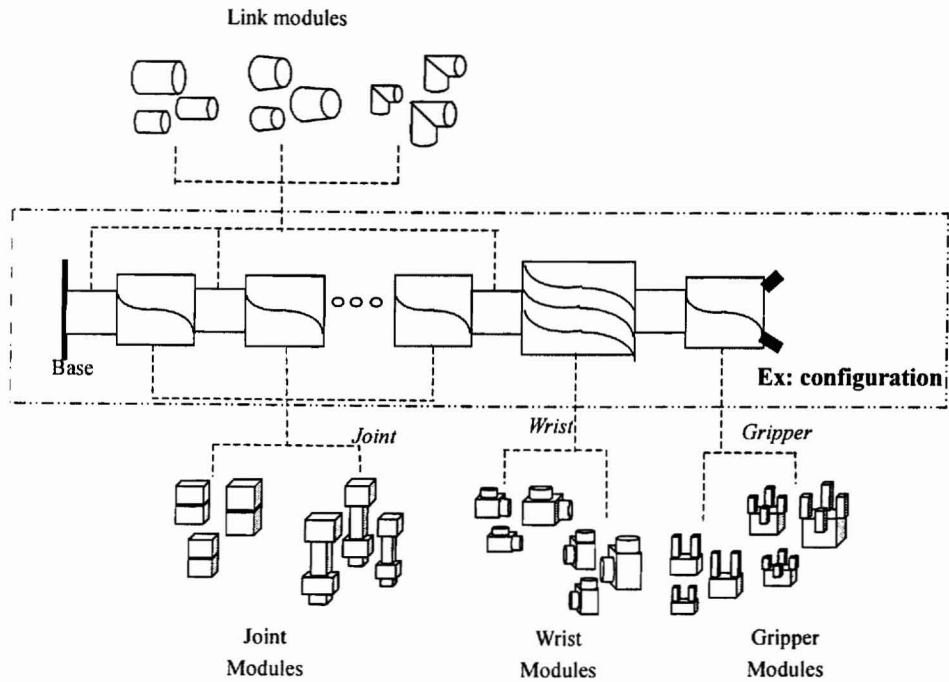
- Yang, Guilin, 1999. *Kinematics, Dynamics, Calibration, and Configuration Optimization of Modular Reconfigurable Robots*, Ph.D. Thesis, Nanyang Technology University, Singapore.
- Yang, Guilin and Chen, I-Ming, 2000. “Task-Based Optimization of Modular Robot Configurations - MDOF Approach,” *Mechanism and Machine Theory*, Vol. 35, No. 4, pp. 517-540.
- Yang, Z. and Sadler, J. P., 1990. “Finite Element Modeling of Spatial Robot Manipulators,” *Flexible Mechanisms, Dynamics and Robot Trajectories: Proc. of the 1990 ASME Design Technical Conference*, pp. 489-496
- Yoshikawa, T., 1985. “Manipulability of Robot Mechanisms,” *Int. J. of Robotics Research*, Vol.4, pp. 3-9.
- Zhang, W. J., 1994. *An Integrated Environment for CAD/CAM of Mechanical Systems*, Ph.D. Thesis, Delft University of Technology, Delft, Netherlands.
- Zhang, W. J., Li, Q., and Guo, S. L., 1999. “Integrated Design of Mechanical Structure and Control Algorithm for a Programmable Four-Bar Linkage,” *IEEE/ASME Trans. on Mechatronics*, Vol. 4 No. 4, pp. 354-362.
- Zhang, W. J. and Liu, S. N., 2000. “Data/Knowledge Representation of Modular Robot and its Working Environment,” *Robotics and Computer Integrated Manufacturing*, Vol. 16, No.3, pp. 143-159.

## **Appendix A**

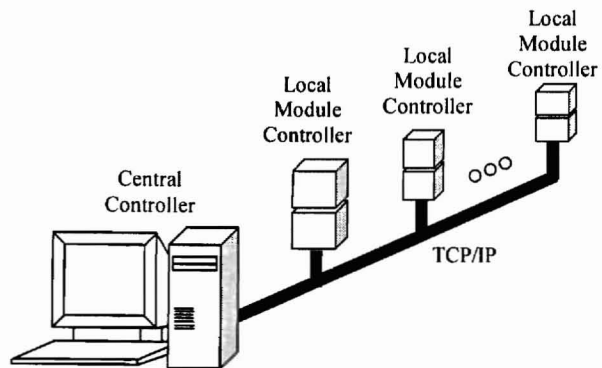
### **AMTEC Modular Robotic System**

#### **A.1 Introduction**

The AMTEC Modular robotic system was made in Germany (Jaenisch et al. 2000). As shown in Figure A.1, the system consists of joint modules, link modules, wrist modules and gripper modules of different types. Link modules have various shapes and dimensions for connecting joint modules and wrist modules. The joint, wrist and gripper modules contain one or more actuator(s). The control architecture is a centralized one, and the communication between actuators and the central controller is a bus model. Figure A.2 shows the software architecture of AMTEC systems, where each module with actuator(s) has its own local controller, the module control commands are issued from the central controller, which is responsible for decomposing a robot task into a series of the motion commands for all driving actuators.



**Figure A.1** Hardware architecture of the AMTEC system



**Figure A.2** Software architecture of the AMTEC system

## **A.2 Hardware System**

### **A.2.1 Features of modules**

As a physical body, a module has many features, such as the dimension, the mass, the center of mass, the moment of inertia and several ports. For the purpose of analysis, there is a local coordinate system frame attached to each module. Then all the features can be geometrically represented with respect to the local frame. An assembly pattern refers to a set of ports on a module through which other modules are connected. Given a module with a definite number of ports, the total number of assembly patterns can be calculated. However, among all these assembly patterns, some are isomorphic in the sense that they do not create any difference in terms of the features. A set of non-isomorphic assembly patterns should be identified prior to modular robot configuration synthesis, and this issue will be further discussed later.

### **A.2.2 Rotary joint modules**

There are 6 types of rotary joint modules in the AMTEC systems, and their features, including size, drive torque, precision, weight, limitation of joint velocity and displacement, are listed in Table A.1.

**Table A.1** Features of rotary modules

Size (a)	m	0.070	0.070	0.090	0.090	0.110	0.110
Drive Torque	Nm	13.1	18.7	34.6	49.2	65.6	93.2
Velocity	$^{\circ}/s$	3.768	2.356	3.768	2.356	3.768	2.356
Precision	rad	$\pm 0.00035$	$\pm 0.00035$	$\pm 0.00035$	$\pm 0.00035$	$\pm 0.00035$	$\pm 0.00035$
Range	rad	$\pm 2.97$	$\pm 2.97$	$\pm 2.97$	$\pm 2.97$	$\pm 2.97$	$\pm 2.97$
Weight	kg	2 $\times$ 0.9		2 $\times$ 1.9		2 $\times$ 3.3	
COM 1	m	(0,0,-0.035)		(0,0,-0.045)		(0,0,-0.055)	
COM 2	m	(0,0,0.035)		(0,0,-0.045)		(0,0,-0.055)	
Inertia	$kg \cdot m^2$	2 $\times$ (0.05,0.05,0.05)		2 $\times$ (0.06,0.06,0.06)		2 $\times$ (0.07,0.07,0.07)	

As shown in Figure A.3, a rotary joint module consists of two parts that rotate, called part A and B. Each part is cubic shaped. Part A has a port for cable communication. A rotary joint module has 9 ports for physical connections; see Figure A.3. Assembly ports are marked with numbers as the port identifiers. The total number of assembly patterns for a rotary joint module is 20, among which there are only 7 non-isomorphic patterns:

$$\{(1,6), (1,7), (2,6), (2,7), (2,8), (2,9), (3,7)\}$$

The local coordinate system is established at the center of a rotary joint module. The Z-axis is coincident with the joint axis. The X-axis is defined to meet the following constraints: (i) it lies in a plane which is perpendicular to the Z-axis and to which the module is symmetric, and (ii) it is a principal axis with regard to the moment of inertia and the product of inertia of the module, see Figure A.3.



The postures of assembly ports are defined in the module local coordinate system. To define the posture of each port, one needs to define a frame for each port. For a port on Part B, the center of the port is the origin of the posture, the direction of the Z-axis is defined to be toward the assembly face, and the direction of the X-axis is determined based on the right hand rule from the Z-axis of the assembly port to the Z-axis of the module. For a port on part A, the center of port is the origin of the posture, the direction of the Z-axis is defined to be opposite to the assembly face, and the direction of the X-axis is determined based on right hand rule from the Z-axis of the module to the Z-axis of the assembly port. In this way, the postures of the assembly ports are shown in Figure A.3, and they are expressed in a matrix form in the following (ports that will produce isomorphic assembly patterns are not included).

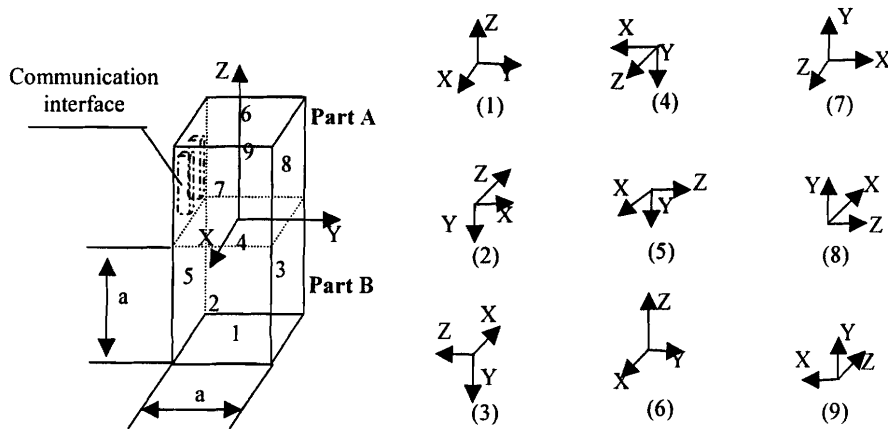
For the ports on part B:

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -a \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0 & 0 & -1 & a/2 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -a/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For the ports on part A,

$$A_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_7 = \begin{bmatrix} 0 & 0 & 1 & a/2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_8 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & a/2 \\ 0 & 1 & 0 & a/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_9 = \begin{bmatrix} 0 & 0 & -1 & -a/2 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



**Figure A.3** Rotary joint module

### A.2.3 Linear joint modules

There are 6 types of linear joint modules in the AMTEC system; their features are listed in Table A.2. It should be noted that a linear joint module has an adjustable parameter for the initial joint position.

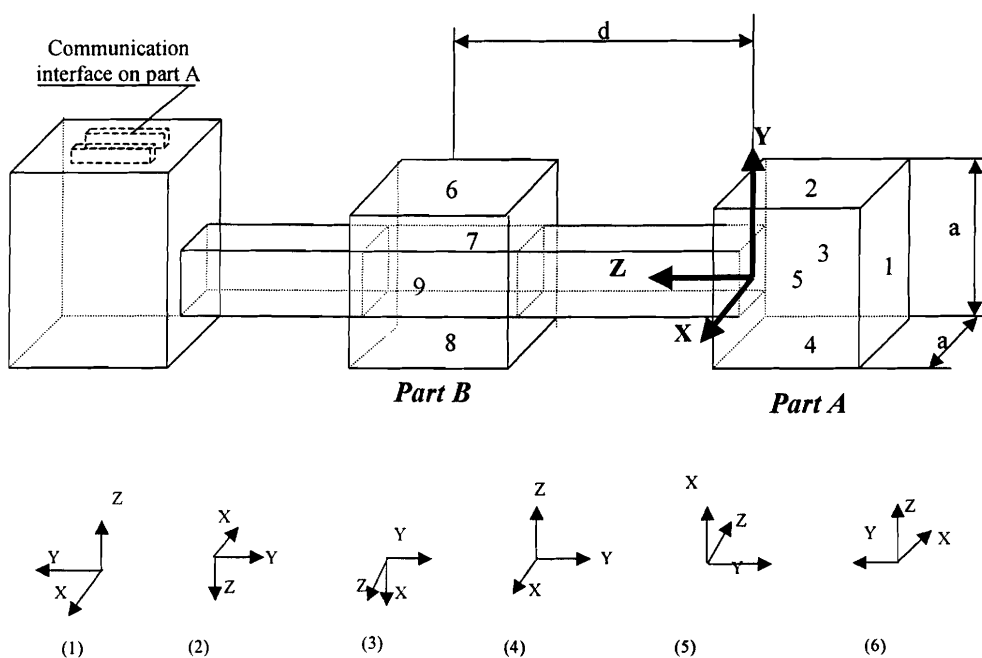
**Table A.2** Features of linear modules

Size (a)	m	0.070	0.070	0.090	0.090	0.110	0.110
Drive Force	N	280	500	520	800	740	1350
Velocity	m/s	0.250	0.125	0.300	0.150	0.400	0.200
Precision	m	$\pm 0.00005$	$\pm 0.00005$	$\pm 0.00005$	$\pm 0.00005$	$\pm 0.00005$	$\pm 0.00005$
Range (d)	m	0.35		0.45		0.58	
Initial position	m	(0.0, 0.02)		(0.0, 0.02)		(0.0, 0.02)	
Mass 1	kg	2.0		3.2		5.5	
Mass 2	kg	0.6		1.0		1.3	
COM 1	m	(0.0,-0.035)		(0.0,-0.045)		(0.0,-0.055)	

COM 2	m	(0,0,0.040)	(0,0,0.100)	(0,0,0.150)
Inertia 1	$\text{kg}\cdot\text{m}^2$	(0.2,0.05,0.05)	(0.25,0.06,0.06)	(0.28,0.08,0.08)
Inertia 2	$\text{kg}\cdot\text{m}^2$	(0.05,0.05,0.05)	(0.06,0.06,0.06)	(0.08,0.08,0.08)

A linear joint module is described as shown in Figure A.4. The set of the non-isomorphic assembly patterns is found in the following:

$$\{(1,6), (2,6), (3,6), (4,6), (5,6)\}$$



**Figure A.4** Linear joint module

The assembly ports are expressed as follows:

$$\begin{aligned}
A_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -a \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_2 &= \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & a/2 \\ 0 & -1 & 0 & -a/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_3 &= \begin{bmatrix} 0 & 0 & 1 & -a/2 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -a/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
A_4 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -a/2 \\ 0 & -1 & 0 & -a/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_5 &= \begin{bmatrix} 0 & 0 & -1 & a/2 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -a/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_6 &= \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & a/2 \\ 0 & 1 & 0 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

## A.2.4 Wrist modules

There are 4 types of wrist modules; their features are listed in Table A.3.

**Table A.3** Features of wrist modules

Size (a)	m	0.070	0.070	0.090	0.090
Drive Force	Nm	8.6	2.6	18.6	8.6
Velocity	rad/s	3.77	6.28	2.36	3.77
Precision	rad	±0.00035	±0.00035	±0.00035	±0.00035
Range	rad	±2.97		±2.97	
d <sub>1</sub>	m	0.07		0.09	
d <sub>2</sub>	m	0.09		0.11	
Mass 1	kg	0.6		1.2	
Mass 2	kg	1.05		2.1	
Mass 3	kg	0.15		0.3	
COM 1	m	(0,0,0.035)		(0,0,0.045)	
COM 2	m	(0,0,0)		(0,0,0)	
COM 3	m	(0,0,0.040)		(0,0,0.050)	
Inertia 1	kg·m <sup>2</sup>	(0.05,0.05,0.025)		(0.1,0.1,0.05)	
Inertia 2	kg·m <sup>2</sup>	(0.08,0.08,0.1)		(0.16,0.16,0.2)	
Inertia 3	kg·m <sup>2</sup>	(0.01,0.01,0.005)		(0.02,0.02,0.01)	

As shown in Figure 4.2, a wrist module has three physical components. Two joint bodies and a linkage body are formed when two virtual linkage bodies are introduced. Their assembly patterns are unique, i.e., (1,2). A wrist module has two assembly ports, i.e., port 1 and port 2. The local coordinate system is set to that of port 1.

Bodies	Joint 1	Link	Joint 2
Port 1		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
Port 2	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

### A.2.5 Gripper modules

Two types of grippers are considered, and their features are listed in Table A.4.

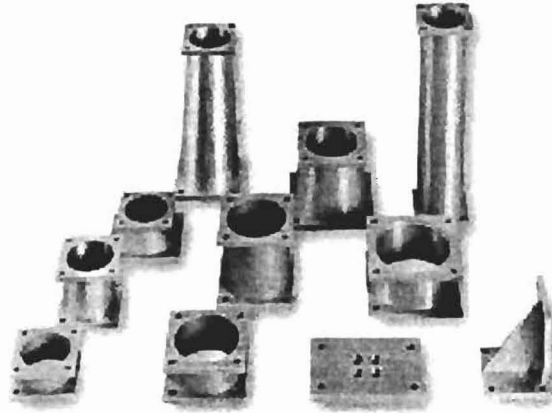
**Table A.4** Features of gripper modules

Size	m	0.070	0.090
Grip force	N	200	500
Grip stroke	m	2×0.030	2×0.030
Precision	m	±0.00005	±0.00005
Velocity	m/s	2×0.020	2×0.020
Mass	kg	1.1	1.9
COM	m	(0,0,-0.045)	(0,0,-0.056)
Inertia	kg·m <sup>2</sup>	(0.2,0.05,0.05)	(0.26,0.06,0.06)

A gripper module has one assembly port to a joint or wrist module. The reference coordinate system is set to that of the input port.

## A.2.6 Link modules

Link modules of the AMTEC system are shown in Figure A.5.



**Figure A.5** Link modules in the AMTEC system

The link module has the unique assembly pattern, and it has two assembly ports: port 1 and port 2. The local coordinate system is set to be port 1, as a result, port 2 could be described by a 4×4 matrix with respect to the frame of port 1.

**Table A.5** Features of link modules

		Size	Mass	COM	Inertia	Port 2
		m	kg	m	kg·m <sup>2</sup>	
Standardized link modules	1	0.07—0.07	0.33	(0,0,0.035)	(0.01,0.01,0.005)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.07 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	2		0.54	(0.035,0,0.035)	(0.015,0.015,0.01)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.035 \\ 0 & -1 & 0 & 0.035 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	3	0.09—0.09	0.43	(0,0,0.045)	(0.016,0.016,0.006)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.09 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	4		0.60	(0.045,0,0.045)	(0.020,0.020,0.015)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.045 \\ 0 & -1 & 0 & 0.045 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	5	0.11—0.11	0.63	(0,0,0.055)	(0.015,0.017,0.007)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.11 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Customerized link modules	6		0.93	(0.055,0.0.055)	(0.025,0.025,0.015)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.055 \\ 0 & -1 & 0 & 0.055 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	7	0.07—0.09	0.5	(0,0,0.030)	(0.015,0.015,0.005)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.07 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	8		0.62	(0,0,0.040)	(0.018,0.018,0.006)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.09 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	9	0.09—0.07	0.5	(0,0,0.040)	(0.015,0.015,0.005)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.07 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	10		0.62	(0,0,0.050)	(0.018,0.018,0.006)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.09 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	11	0.07—0.11	0.54	(0,0,0.030)	(0.020,0.020,0.0065)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.07 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	12		0.65	(0,0,0.050)	(0.025,0.025,0.0075)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.11 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	13	0.11—0.07	0.54	(0,0,0.040)	(0.020,0.020,0.0065)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.07 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	14		0.65	(0,0,0.060)	(0.025,0.025,0.0075)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.11 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	15	0.09—0.11	0.7	(0,0,0.040)	(0.026,0.026,0.007)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.09 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	16		0.85	(0,0,0.050)	(0.0280,0.028,0.009)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.11 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	17	0.11—0.09	0.7	(0,0,0.050)	(0.026,0.026,0.007)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.09 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	18		0.85	(0,0,0.060)	(0.0280,0.028,0.009)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.11 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	19	0.09—0.09	0.35	(0,0.045,0.0235)	(0.02,0.02,0.0015)	$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.065 \\ 0 & 1 & 0 & 0.045 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	20	0.09—0.07	0.75	(0,0.035,0.105)	(0.0315,0.3,0.02)	$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -0.07 \\ 0 & 0 & 1 & 0.21 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

COM: Center of the mass

## Appendix B

### Design Analysis and Synthesis Program

The program consists of two parts: Automatic Kinematic and Dynamic Analysis (AKDA) and Adaptive Robot Configuration Synthesis (ARCS). The inputs to the program are the definition of the modular robot system and given task specification, and the outputs are the optimal configuration and its graphic simulation. The source codes of this program are attached in a floppy disk.

#### B.1 AKDA

AKDA includes the following components:

<b>'init'</b>	To define the modular robotic system.
<b>'module_dh'</b>	To convert design variables to D-H kinematic and dynamic parameters.
<b>'fun'</b>	To solve inverse kinematics.
<b>Toolbox for robotics</b>	To perform trajectory planning, calculate Jacobian matrix, and solve inverse dynamics.



## B.2 ARCS

ARCS includes the following components:

<b>'main'</b>	the main program to organize all of functional modules and integrate with AKDA.
<b>'eva_case'</b>	To verify if a candidate meets design constraints and evaluate configuration performance.
<b>'Toolbox for Genetic Algorithm'</b>	To perform configuration synthesis.

## B.3 Program Output

The result of the program is stored in the following files:

<b>'new_par.out'</b>	Normalized values of design variables with respect to the generations.
<b>'new_obj.out'</b>	Evaluation fitness with respect to the generations.
<b>'new_org_par.out'</b>	Original values of design variables with respect to the generations.
<b>'conc_par3.out'</b>	Optimal design variables and corresponding D-H parameters.
<b>' new_end_pop.out'</b>	Terminal population.

## B.4 Program Execution

- (i) The '**main**' program and all subroutines of functional modules are stored in the same sub-directory.
- (ii) Run the Matlab R12 software.
- (iii) In the menu list '*File*' of **Command Window**, click '*Set Path*' and set the current directory as the directory where all of the programs are stored. All the user-defined routines would be searched in this directory. The result of steps (ii) and (iii) is showed in Figure B.1. Before a specific program is executed, The Matlab environment locates all routines needed, complies and links them directly and automatically to form an execution program.
- (iv) In command line of **Command Window**, type in the name of the main program '*main*' and press '*enter*' button as shown in Figure B.2. The program will start and finish once an optimal result is produced.

The window for the graphic simulation of the optimal configuration will be produced automatically after the process of the design synthesis is completed, as showed in Figure B.3.

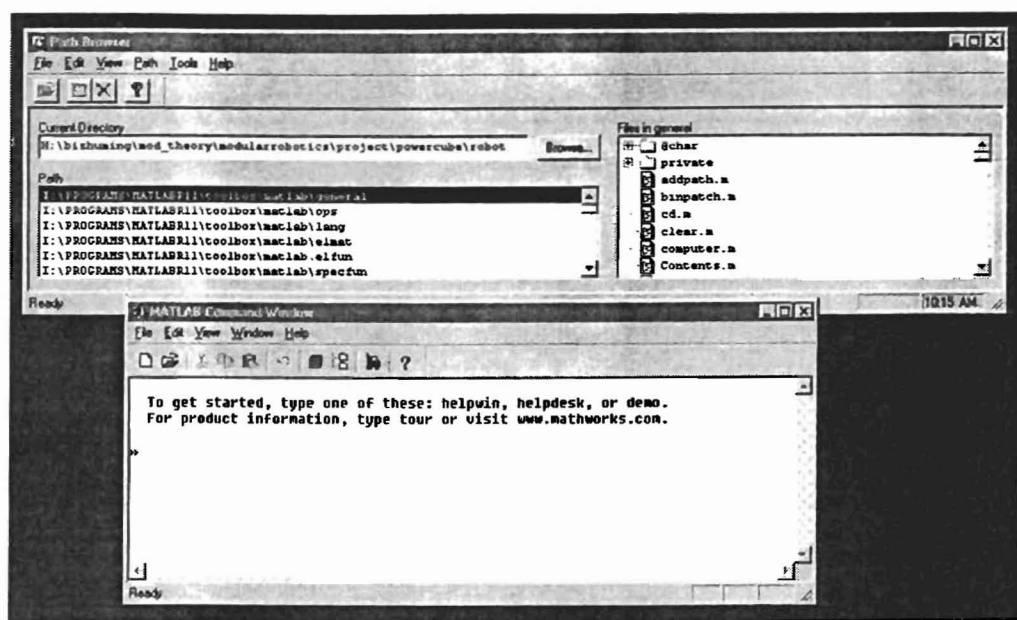


Figure B.1 Path setting for executing the program

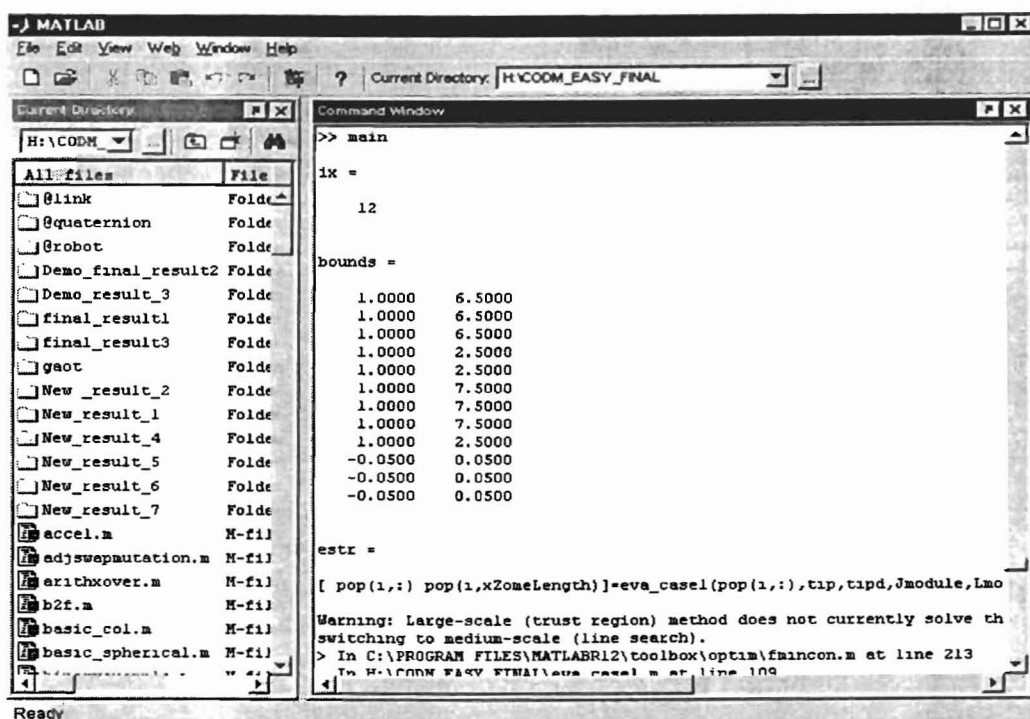


Figure B.2 Execution of the main program

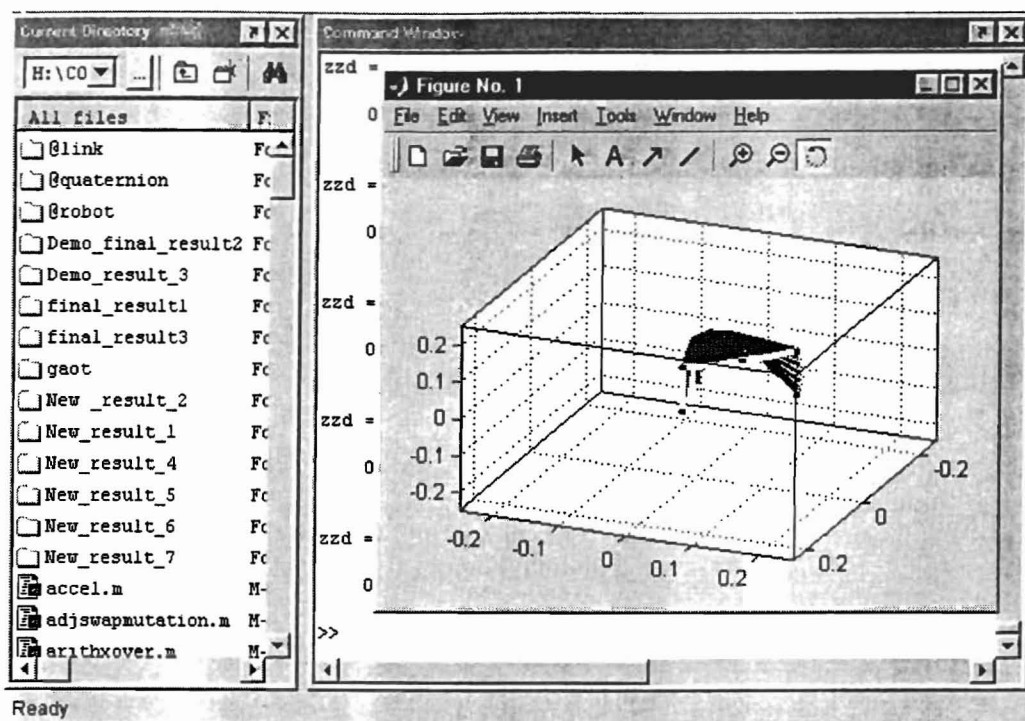


Figure B.3 Window for the graphic simulation

## Appendix C

### GA Implementation Issues

#### C.1 Implementation Strategy

After the design space is divided into three sub-spaces, GA implementation should be straightforward. The basic idea of the implementation is to have three loops of GA operations. However, this could introduce considerable computational overhead using a single machine because the three GA operations are coupled. Let  $i$  denote loop number,  $n_g^i$  the number of generations for terminating the GA operation in loop  $i$ , and  $n_p^i$  the number of individuals in each generation. The total number of times needed for evaluation of the objective function within loop  $i$  can be calculated by  $n_g^i \times n_p^i$ . It is further noted that these loops are coupled. If a problem requires two GA operations, GA loop 2 and GA loop 3 in the case study here, the total number of times needed for evaluation of the objective function will be  $(n_g^2 \times n_p^2) \times (n_g^3 \times n_p^3)$ .

In the present implementation, a simplification strategy is described. It is noted that in the case study here, only two loops (loop 2 and 3) are considered. For the example system discussed in this thesis, there are 12 variables if three joint modules are of a rotary type. When one of the rotary joint modules is replaced by a linear joint module, in loop 3 one more variable for representing the initial position of a linear joint module is created (case 1). When two of the rotary joint modules are replaced by two linear joint modules, two new variables, respectively, corresponding to the initial positions of two linear joint modules, will be created (case 2). Finally, when three rotary joint modules are all replaced by three linear joint modules, respectively, three variables are created (case 3).

To the above three cases, case 1 further corresponds to three situations because the linear joint module can be any one of the three rotary joint modules; likewise, case 2 to three situations; case 3 to one situation only. Therefore, the total number of the possibilities for loop 2 with respect to loop 3 is eight. To each of the eight situations in loop 2, the GA operation is performed. If  $n_g^3 = 100$  and  $n_p^3 = 100$ , the total number of times for evaluation of the objective function in this case is  $8 \times (100 \times 100)$ .

## C.2 GA Codings of Design Variables

Figure C.1 shows the coding scheme for design variables. Each variable corresponds to a sub-string. It should be noted that the number of design variables for adjustable joint parameters at the last part of the string depends on joint types in the three joints; only the

linear joint type has an adjustable parameter, i.e., the initial position.

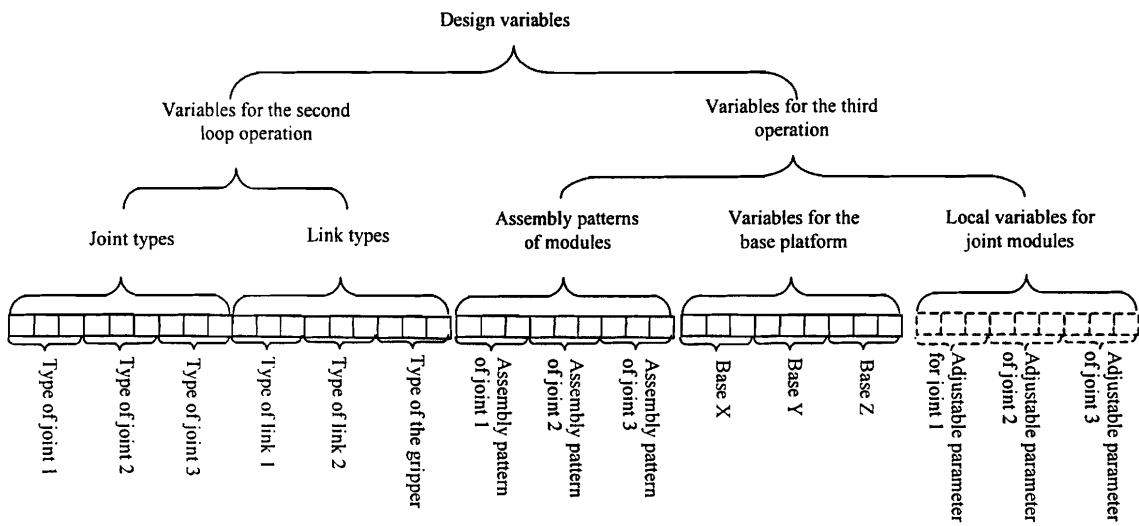


Figure C.1 Coding scheme for design variables

### C.3 Initial and Final Solution Sets

Tables C.1 shows the design variables and the fitness at the initial population, when all the three joints are of a rotary type. It should be noted that the fitness value for the candidate that does not meet the constraints is given  $-100$ .

**Table C.1** Design variables and fitness at the initial population (with three rotary joint modules)

Type of joint 1	Type of joint 2	Type of joint 3	Type of link 1	Type of link 2	Assembly type of joint 1	Assembly type of joint 2	Assembly type of joint 3	Type of gripper	Base X(m)	Base Y(m)	Base Z(m)	Fitness(N.m)
1	6	2	1	1	6	7	2	2	0.023781	-0.01715	-0.00054	-100
6	4	3	1	1	1	5	1	1	-0.02428	0.02695	-0.03789	-100
6	1	5	1	1	7	2	5	1	0.016559	-0.02855	0.028042	-100
1	1	6	1	1	2	3	4	1	0.021672	0.009968	-0.00365	-100
3	2	5	1	1	4	5	6	2	0.039505	0.0019	0.047169	-100
4	5	3	1	1	2	4	3	2	0.048712	0.024884	0.026679	-100
4	5	1	2	1	4	6	1	1	0.022676	-0.03476	0.042524	-100
2	3	3	2	2	4	3	3	2	-0.02311	0.012786	0.015855	-100
3	1	2	1	1	1	4	2	1	-0.04745	-0.02537	0.007256	-100
4	5	2	1	2	5	4	2	2	-0.02662	0.001297	0.039199	-100
4	5	1	2	1	4	5	5	2	0.045785	0.047199	-0.03238	-100
5	5	4	2	1	1	1	7	2	-0.01776	-0.00788	-0.02149	-100
5	4	1	1	2	5	6	2	2	-0.00493	0.004829	0.049026	-100
1	5	2	2	1	3	5	2	1	-0.03672	-0.0047	0.020146	-100
1	4	6	2	1	6	7	4	1	0.004514	0.037747	-0.02399	-100
3	1	6	1	1	2	5	3	2	0.049205	0.046885	0.04131	-100
6	6	2	2	2	5	4	7	1	-0.00435	0.032317	0.001178	-100
2	3	5	2	1	6	1	1	1	-0.00586	0.046246	0.041279	-100
2	1	1	2	1	4	5	5	1	-0.01967	-0.03527	0.019606	-100
2	4	4	2	2	7	5	2	1	-0.02258	-0.03628	0.036341	-100
4	4	1	2	2	6	1	1	1	-0.0066	-0.03891	-0.04142	-100
3	3	5	1	1	4	3	5	2	0.041369	-0.0205	-0.03693	-100
3	3	3	2	1	5	2	2	1	-0.03023	0.031725	0.046526	-100
5	4	1	1	2	5	7	4	1	-0.03103	-0.03617	-0.01776	-100
2	4	1	1	1	6	1	4	1	-0.0155	-0.03012	0.017976	-100
2	5	5	1	1	3	1	7	1	0.012369	-0.03066	-0.00234	-100
2	5	2	1	1	4	6	7	1	0.019969	-0.02137	-0.00419	-100
3	3	2	2	1	4	1	1	1	-0.02761	0.031215	-0.0023	-100
4	2	4	1	2	4	5	2	2	0.046376	0.032691	0.015517	-100
4	2	2	1	2	6	7	3	1	0.040333	0.029591	-0.04024	-100
2	1	5	1	2	4	4	1	1	0.035661	-0.03657	0.011044	-100
3	6	6	1	2	6	1	5	2	0.014678	0.000184	0.001954	-100
3	3	6	1	1	3	6	6	1	0.002702	-0.00381	0.013332	-100
4	3	3	1	1	5	5	5	1	-0.04592	-0.00534	-0.02375	-100
4	4	2	1	1	6	7	6	2	0.018593	-0.02518	-0.03007	-100



5	5	6	1	1	2	2	5	1	0.0443	-0.01966	0.033658	-46.4797
2	5	6	1	1	3	4	6	2	-0.03537	0.04922	-0.04488	-100
5	5	4	2	2	6	1	1	2	-0.04795	-0.01801	-0.03411	-100
5	4	5	1	1	4	6	5	1	-0.02685	0.012021	-0.03742	-100
6	6	3	1	1	1	1	6	1	0.017196	0.012083	0.025098	-100
4	4	6	2	2	3	1	2	2	0.028595	0.011817	0.046413	-100
2	2	2	1	1	4	1	4	2	0.011005	0.030653	0.002449	-100
4	4	5	1	1	3	5	6	1	0.024149	-0.03738	0.044094	-100
1	6	4	1	2	2	3	7	1	0.032538	0.032522	-0.03582	-100
1	2	1	1	2	4	3	6	2	0.036573	-0.02183	0.047199	-100
4	3	3	2	1	6	6	3	1	0.037701	-0.03024	0.028521	-100
1	1	5	1	1	1	1	4	1	-0.00151	-0.04815	-0.0052	-100
1	6	3	2	2	7	2	5	1	0.034828	0.048852	0.019605	-100
5	4	1	2	2	5	1	5	1	-0.02408	-0.04412	0.048729	-100
4	3	3	1	1	4	5	7	1	0.028748	-0.01943	-0.03697	-100
1	3	4	2	1	5	5	4	1	-0.0489	0.008945	0.029103	-100
3	6	2	1	2	2	1	4	2	0.041332	0.032645	0.046723	-100
5	1	5	2	1	1	1	3	1	0.017044	-0.00404	0.031441	-100
1	3	5	1	1	2	2	5	1	0.043338	0.002046	-0.00427	-100
3	3	3	1	1	6	4	7	1	0.046149	0.033745	-0.02129	-100
1	1	4	2	1	6	5	3	2	0.046681	0.024406	0.047897	-100
2	2	4	2	1	1	6	3	1	0.006949	0.049666	0.031772	-100
3	2	5	2	1	1	2	3	2	0.044806	0.000715	-0.04005	-100
1	4	4	1	1	2	5	7	1	-0.00069	0.047583	-0.00287	-100
4	4	4	2	1	7	6	7	1	0.01682	0.035237	-0.02628	-100
5	3	4	1	2	4	4	5	2	-0.00412	0.03326	-0.0214	-100
3	6	2	1	1	4	6	1	1	-0.04171	-0.02842	0.045018	-100
5	5	6	1	2	4	1	1	1	0.04939	0.030658	0.02142	-100
1	1	3	1	2	4	3	2	2	-0.02368	-0.03189	0.006842	-100
1	2	4	2	1	2	6	3	1	0.011813	-0.04955	0.047484	-100
2	3	1	2	2	6	6	7	1	0.045233	0.002752	-0.03857	-100
3	4	1	1	2	4	6	1	1	0.005052	0.01459	0.010856	-100
1	2	5	1	1	3	3	2	1	0.049609	-0.00135	0.007324	-100
1	5	5	1	1	3	3	1	1	-0.02267	-0.04783	0.017828	-100
5	3	5	1	2	7	3	5	1	0.044271	0.046289	-0.0188	-100
1	5	2	1	1	4	3	5	1	0.034371	0.042408	-0.03832	-100
5	4	6	1	2	6	4	7	2	-0.03124	-0.03779	-0.00565	-100
3	2	2	1	2	2	6	3	1	0.014749	0.004103	0.035605	-100
1	4	6	2	1	5	6	7	1	0.03008	-0.04356	0.025711	-100
4	4	2	1	1	6	6	6	1	-0.02786	-0.04929	-0.01141	-100
1	2	2	2	1	3	2	6	1	0.02948	-0.03756	0.034306	-100
2	5	3	1	2	6	1	3	1	-0.04163	0.039809	-0.01398	-100
3	2	2	2	1	5	2	7	1	0.006295	0.041991	-0.0153	-100
5	6	5	2	1	7	6	4	2	-0.00585	0.004698	-0.03223	-100
3	2	1	1	1	3	3	4	1	0.040665	0.044069	0.027531	-100
6	2	6	2	2	2	5	4	1	-0.01649	-0.01573	0.006528	-100

1	1	5	1	1	4	5	5	1	-0.04787	-0.03703	0.03174	-100
1	6	4	2	1	3	5	4	1	-0.04183	0.017727	0.036176	-100
3	3	5	2	1	4	4	5	1	-0.01669	-0.03186	0.025925	-100
5	4	2	2	1	1	2	6	1	-0.04067	0.004665	0.048083	-100
1	1	4	1	1	5	6	4	1	-0.02073	0.018619	-0.023	-100
5	1	5	2	1	7	4	7	1	-0.02193	-0.00701	-0.03835	-100
3	5	6	1	2	2	6	7	2	-0.00712	-0.01256	-0.02155	-100
6	5	5	1	1	5	2	3	1	-0.04967	0.005117	-0.01888	-100
2	3	3	1	1	2	1	7	1	0.00343	-0.039	-0.0213	-100
3	5	4	1	2	6	1	1	1	-0.04827	0.045891	0.030952	-100
2	1	1	1	1	4	4	4	1	-0.04748	0.043873	-0.02189	-100
4	5	5	2	1	4	1	6	1	-0.01975	0.002197	-0.01266	-100
3	3	3	1	1	2	4	5	2	-0.02638	0.011926	-0.03625	-100
5	2	5	1	2	1	6	1	1	-0.03585	0.047167	0.019621	-100
3	5	3	1	2	6	2	1	2	0.043142	-0.02596	0.039998	-100
3	3	4	1	1	3	5	4	1	-0.02331	0.005528	-0.02822	-100
4	1	6	1	2	5	4	1	2	0.044099	0.041874	-0.03449	-100
1	2	4	2	1	4	5	1	1	-0.02815	-0.02346	0.043358	-100
6	3	3	2	1	2	2	4	1	0.025988	0.032861	0.013682	-100

Table C.2 shows the design variables and the fitness of the different robot configurations at the terminal population, when all the joints are of a rotary type.

**Table C.2** Design variables and fitness of the different robot configurations at the terminal population (with three rotary joint modules)

Type of joint 1	Type of joint 2	Type of joint 3	Type of link 1	Type of link 2	Assembly type of joint 1	Assembly type of joint 2	Assembly type of joint 3	Type for gripper	Base X(m)	Base Y(m)	Base Z(m)	Fitness(N.m)
1	4	4	1	1	2	2	7	1	-0.05	-0.05	-0.05	-11.0954
1	4	4	1	1	2	2	1	1	-0.05	-0.05	-0.05	-100
1	4	6	1	1	2	2	7	1	-0.05	-0.05	-0.05	-100
1	4	1	1	1	2	2	1	1	-0.05	-0.05	-0.05	-100
1	4	1	1	1	2	2	7	1	-0.05	0.014126	-0.05	-100
1	4	1	1	1	2	2	7	1	-0.0219	-0.05	-0.05	-11.973
1	4	1	1	1	2	2	7	1	-0.05	-0.05	-0.05	-11.0039
1	4	1	2	1	2	2	7	1	-0.05	-0.05	-0.05	-100

Table C.3 shows the variables and fitness of the different robot configurations at the terminal population if two of joint modules are of a linear type, where the two more variables representing the initial positions of the linear joints are added.

**Table C.3** Design variables and fitness of the different robot configurations at the terminal population (with two linear joint modules and one rotary joint module)

Type of joint 1	Type of joint 2	Type of joint 3	Type of link 1	Type of link 2	Assembly type of joint 1	Assembly type of joint 2	Assembly type of joint 3	Type of gripper	Base X(m)	Base Y(m)	Base Z(m)	Initial position of linear joint module at the first joint	Initial position of linear joint module at the third joint	Fitness(N.m)
7	4	7	1	1	3	5	1	1	-0.05	0.022834	-0.01757	0.001	0.001	-15.9761
7	3	7	1	1	3	5	1	1	-0.05	0.022834	-0.01757	0.001	0.001	-100
7	4	7	1	1	5	5	1	1	-0.05	0.022834	-0.01757	0.001	0.001	-100
7	4	7	1	1	3	5	1	1	-0.05	0.022834	0.05	0.001	0.001	-100
7	3	7	1	1	2	4	1	1	-0.05	0.022834	-0.01757	0.001	0.001	-100
7	4	7	1	1	1	5	1	1	-0.05	0.022834	-0.01757	0.001	0.001	-100
7	4	7	1	1	3	7	1	1	-0.05	0.022834	-0.01757	0.001	0.001	-100
7	3	8	1	1	3	4	1	1	-0.05	0.022834	-0.01757	0.001	0.001	-100
7	4	7	1	1	3	5	1	1	-0.05	0.022834	-0.01757	0.001	0.001	-100
7	4	7	1	1	3	2	1	1	-0.05	0.022834	-0.01757	0.001	0.001	-100
7	4	7	1	1	2	4	1	1	-0.05	0.022834	-0.01757	0.001	0.001	-100