

# 國立交通大學

機械工程學系

博士論文

穩定與指向機構之研究

Design for Stabilizer and Pointing Mechanism



研究生：游武璋

指導教授：成維華教授

中華民國九十六年七月

# 穩定與指向機構之研究

研究生：游 武 璋

指導教授：成 維 華 教授

國立交通大學機械學系

## 摘 要

本論文由機構自由度的觀點設計機載感測裝置，進而發展出六自由度運動平台搭配空中載具，使置於其上之雙軸感應器環架能穩定完成其指向之位置，並利用工作空間分析規劃機載內外空間。

在飛行器上為了使感測器能穩定指向其視線並補償飛機飛行時所產生誤差，因此需要設計環架控制機構與六自由度的穩定器。環架機構可使感測器能完成指向功能，而六自由度穩定器則用於即時補償飛行器所產生的運動誤差，使得感測器能穩定的完成工作。

雙軸環架機構之討論包含精密之轉動區塊環架機構設計、機構最佳化設計、機械效能分析；而位置補償平台之分析包含平台工作空間重建與分析、靈巧度分析、工作空間最佳化。並利用邊界方塊技術來分析機構工作空間得到更詳盡三度空間圖形資料使得工作空間可被更精確估算。

# Design for Stabilizer and Pointing Mechanism

student: Wu-Jong Yu

Adviser: Dr. Wei-Hua Chieng

Department of Mechanical Engineering  
National Chiao Tung University

## Abstract

In this dissertation, we discuss the design of the airborne sensor hardware architecture based on the degree freedom perspective. The six D.O.F. motion base is therefore developed and placed beneath the two-axis sensor gimbal to compensate the air vehicle motion error, and stabilize the pointing direction. In addition we use the technique of the workspace analysis to layout the inside and outside space of the airborne vehicle.

In order to stably point the sensor into the commanded line-of-sight and to compensate the motion error derived from the aircraft flight, the gimbal control with the mechanism design and the six degree-of-freedom stabilizer are needed. The gimbal mechanism is used for the function of pointing, and the six degree-of-freedom motion platform is used for the motion compensation induced by the flight vehicle, therefore the sensor can be stably perform its task.

The two-axis gimbal design includes a precise mechanical gimbal design for the turning block, the mechanical optimization design, and the mechanical advantage analysis.

The platform analysis of the position compensation includes

workspace reconstruction technology, workspace analysis, and dexterity analysis with optimization. The marching cube technology is used for the workspace analysis and estimation with the detailed 3D graphic data for more accurate evaluation for the workspace.



## 誌 謝

感謝最支持我的指導老師成維華，有機會讓我從工業工程與技職體系的背景，拔擢接觸正統機械領域。老師包容我的無知和大膽，使得跨領域創造出許多非正規的結果，但老師卻諄諄善誘讓我有機會了解進而謹慎，數年來的督促與指導，使我獲益良多，並使得本論文得以完成。

感謝每位口試老師精闢的指教與見解，真的感恩，使得本論文更加完善。

感謝我的父母親與家人在我最沮喪的時候，給予我最大的支持。

感謝我未婚妻思育無怨無悔的支持，陪伴著我攻讀博士班的日子。

感謝至方學長的鞭策指導包容，讓我能完成許多難以克服的難關。

感謝學弟永成、家豐、仰宏、裕都分擔許多計畫和推導。

感謝領航賴董事長、學長建勳、俊旭、同事們兩年來的包容。

感謝過去教導我的陳福春、張敏德、施志雄、駱景堯……。

最後感謝所有在這期間曾經支持我的所有人，謝謝你們。

# Contents

摘要.....	ii
Abstract.....	iii
謝誌.....	v
Contents.....	vii
List of Figures.....	ix
List of Tables.....	xii
<b>CHAPTER 1 INTRODUCTION.....</b>	<b>1</b>
<b>CHAPTER 2 BUILDING BLOCK APPROACH TO PARALLEL MANIPULATOR DESIGN.....</b>	<b>5</b>
2.1 Reviews.....	5
2.2. Gruebler's criterion.....	6
2.2.1 Joint.....	7
2.2.2 Limb: Dyad, Triad, and Quad.....	8
2.2.3. Open chain.....	9
2.2.3. Parallel manipulator.....	10
2.3 Parallel manipulator design synthesis.....	11
2.3.1 Fully-symmetric, parallel manipulator (FSPM).....	12
2.3.2 Semi-symmetric, parallel manipulator (SSPM).....	13
2.3.3 Task-oriented, parallel manipulator (TOPM).....	14
2.3.4 Saturated limb.....	14
2.4 Design Synthesis by building blocks.....	15
2.4.1 FSPM/SSPM/TOPM.....	15
2.4.2. Ground vs. floating actuator.....	16
2.4.3. Selection of active limb.....	16

2.4.4. Selection of passive limb .....	17
2.4.5. Insertion of saturated limb .....	17
2.5 Design Examples of 3-dof TOPM .....	18
2.5.1. Parallel manipulated Cartesian machine .....	18
2.5.2 Parallel manipulated wobble machine .....	18
2.5.3 Parallel manipulated rotation machine.....	19
2.5.4 Parallel manipulated cobra-head machine .....	19
2.6 Design Examples of 6-dof TOPM .....	19
<b>CHAPTER 3 DESIGN OF THE SWINGING-BLOCK AND</b>	
<b>TURNING- BLOCK MECHANISM WITH</b>	
<b>SPECIAL REFERENCE TO THE MECHANICAL</b>	
<b>ADVANTAGE .....</b>	<b>21</b>
3.1 Reviews.....	21
3.2 Swinging-Block and Turning-Block Mechanisms .....	22
3.3 Maximum Average Mechanical Advantage.....	23
3.4 Optimal Design.....	26
3.5 Design Procedure.....	28
<b>CHAPTER 4 ROBOTIC SAFEGUARD SYSTEM AND</b>	
<b>WORKSPACE ANALYSIS.....</b>	<b>30</b>
4.1 Reviews .....	30
4.2 The Active Multi-Robotic Safeguard System Architecture.....	33
4.3 The Robot System Installation and the Kinematics Analysis.....	35
4.4 The Robotic Operation and Control Interface Program .....	37
4.5 The Robot Language Programming for the Safeguard System.....	38
4.6 Client-Server Networking Architecture.....	41
4.6.1 Client .....	42

4.6.2 Server.....	42
4.7 Robot Remote Supervision (Case One).....	45
4.8 Virtual Reality Remote Supervision for Safeguard of Robotics (Case Two) .....	46
<b>CHAPTER 5 WORKSPACE AND DEXTERITY ANALYSES OF THE STABILIZER (DELTA HEXAGLIDE PLATFORM) .....</b>	<b>48</b>
5.1 Reviews.....	48
5.2 Delta Hexaglide Platform .....	50
5.2.1 Inverse Kinematics.....	50
5.2.2 Dexterity Analysis.....	53
5.3 Marching Cube Method.....	57
5.3.1 Workspace Presentation.....	58
5.3.2 Workspace Examples.....	59
5.4 Analysis and Design.....	60
5.4.1 Combination of Workspace and Dexterity Analysis.....	61
5.4.2 Dimensional Design .....	62
<b>CHAPTER 6 CONCLUSION.....</b>	<b>64</b>
Reference.....	70
Figures.....	79
Tables.....	130



## List of Figures

Figure 1. Vehicle and Sensor relation .....	79
Figure 2. Sensor, Gimbal, stabilizer, air cushion .....	79
Figure 3. The limb.....	80
Figure 4. SS triad .....	80
Figure 5. Type of actuators .....	81
Figure 6. (a) Kinematic structure (b) corresponding graph .....	81
Figure 7. Reduced graph of parallel manipulation .....	82
Figure 8. (a) Tripod-based PKM (b) Stewart platform (Hexapod).....	82
Figure 9. Example of a 5-dof SSPM.....	83
Figure 10. DDB measurement as Saturated limb .....	83
Figure 11. Determination of parallel manipulator type .....	84
Figure 12. Limb Selection .....	85
Figure 13. Cartesian machine .....	86
Figure 14. Wobble machine .....	87
Figure 15. Rotation machine.....	88
Figure 16. Cobra-head machine.....	88
Figure 17. Different types of Hexaslide platform: (a) Hexaglide, (b) HexaM, and (c) Linapod .....	89
Figure 18. Delta Hexaglide platform .....	89
Figure 19. (a) Kinematic structure and (b) photograph of the Delta Hexaglide platform mechanism (Courtesy: IMON Inc.) .....	90
Figure 20. Oscillating-cylinder engine mechanism.....	91
Figure 21. RRRP kinematics inversion.....	91
Figure 22a. The real sensor pedestal picture .....	92
Figure 22b. Kinematic structure of turning-block mechanism.....	92



Figure 46. The network delay for the different size of the transmitted data among multiple robotic controllers .....	115
Figure 47. Top and front view of the Delta Hexaglide platform .....	117
Figure 48. Dexterity of the Delta Hexaglide platform.....	118
Figure 49. Graphical presentation of (a) the voxel with one center and (b) the cube with eight vertices .....	119
Figure 50. Surface representation in Marching Cube method.....	119
Figure 51. 15 unique cube configurations .....	120
Figure 52. The translational workspace of Delta Hexaglide at $\alpha = 0$ , $\beta = 0$ , $\gamma = 0$ ; 16416 faces are found on the boundary; 24775 cubes are found inside the workspace. ....	121
Figure 53. Workspace degeneracy: (a) workspace island (in dotted circle), and (b) workspace cavity (in dotted circle).....	123
Figure 54. The parking position.....	124
Figure 55. Combination of workspace and dexterity result of the Delta Hexaglide; shape complexity is 0.446 and workspace Volume is 5568. ....	125
Figure 56. Different designs of the Delta Hexaglide platform .....	126
Figure 57. (a) Workspace Volume and (b) Shape Complexity of the translational workspace via different L .....	127
Figure 58. (a) Workspace Volume and (b) Shape Complexity of the translational workspace via different $r_{upper}$ .....	128
Figure 59. (a) Workspace Volume and (b) Shape Complexity of the rotational workspace via different L .....	129

## List of Tables

Table 1. List of spatial triads .....	130
Table 2. List of spatial quads .....	130
Table 3. The classes of limb used in the spatial FSPM .....	131
Table 4. The classes of limb used in the spatial SSPM .....	131
Table 5. The classes of limb used in the spatial TOPM .....	132
Table 6. Dimensional parameters used in examples; L is varying from 3.15 to 3.40 in Figure 12.and 13. ....	132
Table 7. Thirty-seven normal directions (vectors) of different patches for surface representation in marching cubes .....	133
Table 7.1 Definition of patch-in-a-Cube based on the edge sequence number .....	134



## Chapter 1 Introduction

There are two types of motion errors that require compensation in air vehicles with stabilizers. The first type comprises orientation motion errors induced by atmospheric disturbance and maneuvering. Sensors, such as a TV camera, E-O/IR, radar, or even a gun, on an air vehicle must be kept stable for constant pointing orientation during flight. The ground stabilization command can be expressed as:

$$u_B = C_I^B u_I, \quad (1.1)$$

$$\text{where } C_I^B = [\varphi][\theta][\psi], \quad C_B^A = [\sigma_E][\sigma_A]$$

$[\sigma_E]$  and  $[\sigma_A]$  are antenna elevation and azimuth angles, respectively, installed on the aircraft body axis, and  $[\varphi]$ ,  $[\theta]$ , and  $[\psi]$  are roll, pitch, and roll Euler angles, respectively, where  $u_B$  is the unit vector of the aircraft body axis, and  $u_I$  is the unit vector of the line-of-sight for the sensor pointing direction in the inertial space.

Conversely, for image radar, such as synthetic aperture radar [1], the second motion error is line-of-sight range deviation between the antenna phase center of the air vehicle and the target map patch center. The deviation range decreases map resolution and quality, and requires compensation; this compensation is called motion compensation (MOCOMP) [2]. The along-track (line-of-sight (LOS)) range deviation-induced phase error (Fig. 1) can be calculated as follows:

$$\phi_{LOS} = -\frac{4\pi}{\lambda} \int V_{LOS} dt \quad (1.2)$$

where  $V_{LOS}$  is the LOS range deviation,  $\lambda$  is the wavelength of the radar transmission wave, and dt is the radar pulse repetition interval (PRI).

Conventionally, the first motion error type can be overcome with the sensor gimbal controller using inertial sensor data, and the second motion error type can be compensated by electronic phase adjustment, either in real time or non-real time.

This work introduces a novel six D.O.F.(degree of freedom) motion base [3] placed beneath the gimbal, which plays the role of a type-I (orientation) motion error stabilizer and a type-II (translation) [motion compensator OR MOCOMP]. The advantages of the new approach for motion stabilization and compensation are as follows.

1. The motion base provides high pointing accuracy and rapid frequency response due to the better stiffness of the inherent parallel mechanism characteristics.

2. A sensor gimbal typically only has two D.O.F.. The proposed stabilizer is a good solution for compensating for type-I errors.

3. The MOCOMP for image radar can be compensated for by the motion base directly, without need for any electronic phase adjustment, as long as the LOS range deviation is within the workspace of the motion base translational movement.

High frequency vibration is removed using an air cushion system [4] such as passive or active shock mounts. With the autopilot enabled, the remaining middle and low frequency errors, such as orientation error, flight-path error, or flight-speed error, can be compensated for by the six D.O.F. platform. Figure 1 presents the relationship between the sensor and the air vehicle.

In this dissertation, we presents a methodology for parallel manipulator design. The parallel manipulators are categorized into fully-symmetric, semi-symmetric, and task-oriented types. According to these types, different limbs may be chosen for building the desired parallel manipulator. Useful limbs including the dyad, triad, and quads are enumerated for individual selections in the design process. The task-oriented, parallel manipulator is found particularly useful for performing tasks applicable to the domain of high speed machining. Following this design methodology, the linkages and joints are decided as shown in Fig. 2 which is a new six D.O.F. motion base as the motion stabilizer and compensator.

The direction pointing equipment for airborne sensors is a two-axis gimbal. The two-axis design uses low power, is lightweight, has a small volume and high strength structure, has a minimal number of linkages, and generates maximal output force. With the proposed actuator, the turning block as the natural gearbox structure is used for the two-axis gimbal to acquire mechanical design limits. The turning block and swing block mechanical advantages, and the mechanical optimization of the gimbal [5] are analyzed.

The gimbal stabilizer is a Delta Hexaglide Platform. The platform consist of two plates, six linkages, and six sliders as shown in Fig. 19. The upper plate is a “moving plate” which is an end-effector with six D.O.F.. The other plate, which is fixed on ground, is called a “fixed base;” however, it is now fixed on the vehicle air cushion.

The stabilizer system is designed in light of vehicle space limitations and safety issues. This work considered robot safeguards with virtual boundary techniques [6], and other space adjustments,

using virtual reality (VR) programming to determine workspace boundaries. Therefore, the stabilizer must be rebuilt in a three-dimension Cartesian space using the VR technique. The dexterity problem can be verified and the platform design optimized.





## **Chapter 2 Building Block Approach to Parallel Manipulator Design**

### **2.1 Reviews**

The Stewart/Gough platform, the most well-known platform manipulator, is a 6-dof platform controlled by six active prismatic joints [7]. Six UPS limbs connect the fixed base to the moving platform. Applications of parallel manipulators are commonly found in the motion platform for pilot training simulators, positioning devices for high precision surgical tools, parallel-type multi-axis machining tools and precision assembly tools.

The design on trajectory planning and application developments of parallel manipulators are challenging due to the closed-loop nature. Analysis work include the generation of and forward pose solution for analytic parallel manipulators [8], parallel manipulator dynamics [9], singularity determination in spatial platform manipulators [10], and inertial measurement unit calibration [11].

Advanced manufacturing will involve application of new concepts, models, methodologies, and information technologies. Because of this recent trend towards high-speed machining, there is also a demand to develop machine tools with high dynamic performance, improved stiffness and reduced moving mass. Parallel manipulator has been adopted to develop this type of machine. Design and analysis work corresponding to a particular parallel manipulator called parallel kinematic machine (PKM) [12]-[14] has been

introduced.

Referring to the reconfigurable parallel robots, the kinematic design of modular reconfigurable in-parallel robots [15], the conceptual design of a modular robot [16], and the CMU reconfigurable modular manipulator system [17] have been presented. Toward the structural synthesis, the structure synthesis of a class of 4-D.O.F and 5-D.O.F parallel manipulators [18] and the qualitative synthesis method for 3-D.O.F and 4-D.O.F parallel manipulators [19] have been uncovered. As to the design synthesis of task-oriented modular parallel robots, the scientific methodology for matching tasks to modular robot [20] and the methodology for design of parallel mechanisms based on the application of graph theory and combinatorial analysis [21] have also been introduced.

Many methods have been developed for the analysis and design synthesis of parallel manipulators. However, the structural synthesis of parallel manipulators in general has not been attempted. This chapter introduces a general aspect of parallel manipulator design. Some innovated task-oriented parallel manipulators with 3-D.O.F. will also be introduced.

## **2.2. Gr uebler's criterion**

Gr uebler's criterion calculates the theoretical number of D.O.F within a mechanism. This is also known as the mechanism's  $F$  number. D.O.F is the number of independent joint variables which must be specified in order to define the position of all links within a mechanism. A body restricted to planar motion has at most three D.O.F. The link is a rigid body. The joint is a contact (or permanent

connection) between two links. The number of links may be denoted by  $l$ . The number of joints may be denoted by  $j$ . The type of joint (or connection) defines the relative motion of the two connected links. There are five categories of contacts in spatial motion, they allow for  $f_j$  D.O.F between the connected bodies, where  $1 \leq f_j \leq 5$ . A  $f_j = 6$  contact would be a non-contact. The theoretical D.O.F  $F$  within a mechanism calculated by the Gruebler's criterion is expressed as follows.

$$F = \lambda(l - j - 1) + \sum f_j \quad (2.1)$$

where  $\lambda$  is the mobility of the space in which the mechanism operates ( $\lambda = 3$  for general plane mechanism,  $\lambda = 6$  for spatial mechanism). The number of independent loops of the mechanism is denoted by  $L$ . According to Euler's formula it is obtained that  $L = j - l + 1$ . The above equation is then written into

$$F = \sum f_j - \lambda L \quad (2.2)$$

The above equation is useful in mechanism synthesis design when the number of independent loop usually indicates the complexity of the mechanism.

### 2.2.1 Joint

An artificial joint, joint with no D.O.F ( $f_j = 0$ ), is introduced in this chapter for the following derivations. The joints with  $1 \leq f_j \leq 5$  are then called the normal joints. The joints connect the links can be revolute ( $f_j = 1$ ), prismatic ( $f_j = 1$ ), cylindrical ( $f_j = 2$ ), universal ( $f_j = 2$ ) or spherical joints ( $f_j = 3$ ). Higher order joints with  $f_j > 3$  may be composed by the lower order joints  $f_j \leq 3$ . The high order joints may

not be generally useful in the parallel manipulators for their difficulty in fabrication. In this chapter, we will look only into the joints with  $f_j \leq 3$ . The normal joint attached with an actuator is called the active joint; on the other hand, it is called the passive joint.

### 2.2.2 Dyad, Triad, and Quad

The  $F$  number within a mechanism calculated by the Grüebler's criterion is provided with the existence of a ground link. The mobility ground link is subtracted from the overall  $F$  number. In case that the kinematic structure is floating with respect to the ground, the mobility  $M$  of the kinematic structure may be derived from Grüebler's criterion as

$$M = \lambda(l - j) + \sum f_j \quad (2.3)$$

A particular floating kinematic structure with  $L = j - l + 1 = 0$  is called the limb as shown in Fig. 3.

The mobility of the limb is derived from eq. (2.3) that

$$M_i = \lambda + \sum f_{j,i} \quad (2.4)$$

where  $\sum f_{j,i}$  denotes the total number joint D.O.F on the  $i$ th limb.

The limb with one joint connecting two links is called a dyad. The limb with two joints in series connection to three links is called a triad. The limb with three joints in series connection to four links is called a quad. For convenience, the total number of D.O.F of the joints is denoted by  $T$  that

$$T = \sum f_j . \quad (2.5)$$

The total number of joint D.O.F of the  $i$ th limb, according to eq. (2.4), is derived as

$$M = T + \lambda \quad (2.6)$$

In order to avoid the under-constraint condition of the Gruebler's criterion, the limb should have the total joint D.O.F  $T$  no more than  $\lambda$ . Otherwise, there will be either an uncontrollable rotation or an uncontrollable translation in the limb. The limb with  $T = \lambda$  is called the saturated limb. The ambient space of a saturated limb is the space in which the mechanism operates.

Table 1 lists some of the useful spatial ( $\lambda = 6$ ) triads using the aforementioned joints. The term “class” i.e.  $s^T$ , in Table 1 is used to denote the class of limbs with total joint D.O.F of the limbs equal to  $T$ .

In Table 1, the SS triad, as shown in Fig. 4, is a saturated limb, however, there is an uncontrollable rotation between two spherical joints. The uncontrollable rotation presents as an under-constraint of the Gruebler's criterion.

Table 2 lists some of the useful spatial ( $\lambda = 6$ ) quads using the aforementioned joints.

In order to classify limbs, we define the limb embodying an active joint the active limb; it is a passive limb if no actuator is attached. Different versions of actuators are shown in Fig. 5. The actuators being connected to the ground are called ground actuators; others are called the floating actuators.

### 2.2.3. Open chain

A limb is defined as the limb with one end grounded. A convenient way to represent an open-chain is the graph representation as shown in Fig.6

In the analysis point of view, the ambient space of the open-chain

is given by the Cartesian product of the joint spaces of all the joints that make up the open-chain [22].

### 2.2.3. Parallel manipulator

A parallel manipulator as shown in Fig. 7 is regarded as a set of limbs connected in parallel to a common rigid body, known as the end-effector. The reduced graph may be expanded into individual graphs representing different limbs.

Let an integer  $n$  denote the number of limbs, an integer  $l_i$  denote the number of links of the  $i$ th limb respectively, and an integer  $j_i$  denote the number of joints of the  $i$ th limb respectively. The total number of joints including the artificial joints and links in the parallel manipulators are

$$j = \sum j_i + 2n \quad (2.7)$$

$$l = \sum l_i + 2 \quad (2.8)$$

Knowing that the artificial joints are with zero joint D.O.F., the total joint D.O.F. is calculated as

$$\sum f_j = \sum \sum f_{j,i} \quad (2.9)$$

Since all limbs are grounded on one end and connected to the end-effector on the other end, the end-effector becomes a common link of all open-chains made of the limbs. According to previous statement that the ambient space of the open-chain, the ambient space of end-effector (the common link) is then the intersection of ambient spaces of all individual open-chains. For example, the intersection of PUP quad and PS triad open-chain will result maximally an ambient space of one axis of translation with two axes of rotation to the end effector.

### 2.3 Parallel manipulator design synthesis

The  $F$  number of a parallel manipulator composed of different limbs, may be calculated from eq. (2.1) according to eq. (2.7) to (2.9) that

$$\begin{aligned} F &= \lambda(2 + \sum l_i - \sum j_i - 2n - 1) + \sum f_j \\ &= \lambda(1 - 2n) + \sum (\lambda(l_i - j_i) + \sum f_{j,i}) \end{aligned} \quad (2.10)$$

Provided with the relation of  $l_i = j_i + 1$  for the limb and the relation in eq. (2.3), the above equation may be written into

$$F = \sum M_i + \lambda(1 - 2n) \quad (2.11)$$

Assuming that the parallel manipulator have two sets of limbs; one set is composed of  $p$  active limbs with identical mobility  $M_p = T_p + \lambda$ ; the other set is composed of  $q$  limbs with identical mobility  $M_q = T_q + \lambda$ . The above equation may then be written as

$$\begin{aligned} F &= pM_p + qM_q + \lambda(1 - 2(p + q)) \\ &= p(T_p - \lambda) + q(T_q - \lambda) + \lambda \end{aligned} \quad (2.12)$$

The difference of total joint D.O.F between these two distinct sets may be formulated as

$$T_q = T_p + \Delta T \quad (2.13)$$

Each active limb is attached with one single actuator, there will be  $n = p + q$  actuators when all  $p + q$  limbs are active limbs. Each actuator corresponds to one independent input parameter. There will be  $n$  independent input parameters. For the safety issue, the parallel manipulator is preferred to be stopped at any instance of time; the parallel manipulator should be controlled to change its velocity instantaneously in all available directions. A controllable parallel

manipulator having as many output D.O.F. as it has input D.O.F. is preferable. It is obtained that  $F = n = p + q$  and eq. (2.12) yields

$$\begin{aligned}\Delta T &= \frac{n(\lambda + 1 - T_p) - \lambda}{q} \\ &= \frac{F(\lambda + 1 - T_p) - \lambda}{q}\end{aligned}\quad (2.14)$$

The parallel manipulator corresponds to  $\Delta T = 0$  is referred to as the fully-symmetric, parallel manipulator (FSPM). All limbs in the FSPM have the same kinematic structure. On the other hand, the parallel manipulators with  $\Delta T \neq 0$  is referred to as the semi-symmetric, parallel manipulator (SSPM).

### 2.3.1 Fully-symmetric, parallel manipulator (FSPM)

FSPM has  $n$  limbs providing  $F = n$  and all limbs have identical mobility  $M_p = T_p + \lambda$ . Eq. (2.14) may be further derived from the relation  $\Delta T = 0$  that

$$\lambda = F(\lambda + 1 - T_p). \quad (2.15)$$

Each limb of the FSPM posses the relation that

$$T_p = \lambda + 1 - \frac{\lambda}{F} \quad (2.16)$$

Table 3 lists all combinatorial results according to the above equation for the spatial case. According to the results shown in Table 3, it is found that only three classes including  $S^4$ ,  $S^5$  and  $S^6$  of limbs are useful in the design FSPM. FSPM is capable of performing 2-, 3-, and 6-D.O.F. motion.

Well-known parallel manipulators such as the tripod-based parallel kinematic machines [14] employing  $S^5$  limbs and the Stewart



platform [7] employing  $S^6$  limbs are as shown in Fig.8 The advantage of FSPM is that the interchangeable limbs. Such advantage is significant in fabrication, assembly, and maintenance aspects.

Special cases for 4-dof or 5-dof parallel manipulators which connect a moving platform to a fixed base by four limbs of identical kinematic structure such that after assembly these four or five limbs provide only two or one linearly independent constraint to the end-effector are introduced by Fang and Tsai [18]. However, these special cases are presented as the over-constraint cases for general Grüebler's criterion, thus which may suffer from the fabrication and assembly inaccuracies.

### 2.3.2 Semi-symmetric, parallel manipulator (SSPM)

Other than the fully symmetric, parallel manipulators, the parallel manipulator have two distinct sets of identical limbs,  $T_q \neq T_p$ , is called the semi-symmetric, parallel manipulator (SSPM). Eq. (2.14) provided with  $\Delta T \neq 0$  may be written as

$$T_p = \lambda + 1 - \frac{q\Delta T + \lambda}{n} \quad (2.17)$$

Table 4 lists all combinatorial results according to the above equation for the spatial case. Compared to Table 3, the SSPM allows the parallel manipulator to have 2-, 3-, 4- and 5-dof.

The 4-dof and 5-dof parallel manipulator that cannot be found in FSPM will attract special focuses. An example of 5-dof SSPM is shown in Fig. 9. This example composes of 5 limbs, four of them are saturated limbs and one is with  $T_p = 5$ . The intersection among 6-space and 5-space is the 5-space, this result is agreed with the general

Grüebler's criterion.

### 2.3.3 Task-oriented, parallel manipulator (TOPM)

In the analysis point of view, the ambient space of the manipulator is given by the intersection of ambient spaces of the constituting limbs. The task-oriented, parallel manipulator (TOPM) is so arranged that the ambient space is preferable for the specified task.

When the  $q$  limbs out of total  $n = p + q$  limbs are passive limbs, it is obtained that  $F = p$  and eq. (2.12) may be rewritten into

$$T_q = \frac{F(\lambda + 1 - T_p) - \lambda}{q} + \lambda \quad (2.18)$$

Table 5 lists all combinatory results except for the ones that satisfy the condition  $\lambda = F(\lambda + 1 - T_p)$  for FSPM. Table 5 resembles the result in Table 4 except for the  $q$  limbs are passive limbs. It is found that all active limbs of TOPM are saturated limbs. The ambient space of the saturated limb is the 6-space. The intersection of 6-spaces of all active limbs is still the 6-space. Hence the ambient space of TOPM, an intersection of ambient spaces of all limbs, is then determined by the ambient space of the passive limb. For the case that  $F = 5$ , the passive limb must reduce the ambient space to 5-space, i.e.  $T_q$  must be 5. For the case at  $F = 4$ , there could be two limbs each has an ambient space of 5-dof providing that the minimal intersection of 5-spaces is 4-space. These results are agreed with Table 5.

### 2.3.4 Saturated limb

Substituting the relation that  $\lambda = F(\lambda + 1 - T_p)$  for FSPM from

eq. (2.15) into eq. (2.18), we obtain  $\Delta T = \lambda - T_p$  for all  $q$ . It is then obtained that  $T_q = \Delta T + T_p = \lambda$  which implies the passive limb must be a saturated limb. The insertion of saturated limbs, no matter how many of it, will not affect the total D.O.F.  $F$  of FSPM. The saturated limb may be attached to a suspension that damps the end-effector motion or a double ball bar (DBB) for measurement as shown in Fig. 10

## 2.4 Design Synthesis by building blocks

The conceptual design of parallel manipulator may be organized into the procedures as follows. Each individual steps are introduced in the following sections.

### 2.4.1 FSPM/SSPM/TOPM

All the fully-symmetric, parallel manipulator (FSPM), semi-symmetric, parallel manipulator (SSPM), and task-oriented, parallel manipulator (TOPM) are useful in domains such as the motion simulation and CNC machine tool. The choice may be made according to the following guidelines as shown in Fig.11

(1) The fully- and semi-symmetric, parallel manipulator require less number of limbs than the task-oriented, parallel manipulator.

(2) In the application that requires fabrication and inventory simplicity, the FSPM is sound.

(3) For task-oriented applications, the TOPM is chosen instead of all others.

(4) In the application that requires  $F = 4$  or  $5$ , the SSPM is chosen.

## 2.4.2. Ground vs. floating actuator

The hydraulic pump can offer a great power capacity and is useful for a large-scale motion platform. In the hydraulic applications, the prismatic joint associated with a hydraulic cylinder in extension type is preferred. In the hydraulic applications, the prismatic joints are usually floating. The electrical motors in rotational type are clean and easy to maintain. There is usually a need to convert the rotational motion into translational displacement when an electrical motor is used. The guide-way and ball-screw system is used for the kinematics conversion purpose. The prismatic joint may be equivalent to the guide-way and ball-screw system. In the electrical applications, the prismatic joints are usually fixed to the ground link.

## 2.4.3. Selection of active limb

The limbs being useful as active limbs include  $S^4$ ,  $S^5$ , and  $S^6$  according to Table 3, 4, and 5. Two  $S^4$  triads are found in Table 1, which are PS and RS. Only PS triad becomes useful in the electrical application. Eight  $S^4$  quads are found in Table 1. Six of them possessed a prismatic joint are PCR, PUR, PCP, CPR, UPR, and CPP. According to the application concern, it is found that PCR, PUR, PCP, and CPP quads are useful in the electrical application. CPR, UPR, and CPP quads are useful in the hydraulic applications.

Two  $S^5$  triads are found in Table 1, which are CS and US. In case that each triad must be connected to at least one actuator, neither spherical nor universal nor cylindrical joint with more than one joint D.O.F. is easy for actuator application. Fourteen  $S^5$  quads are found in

Table 1. Seven of them possessed a prismatic joint are PCU, PUC, PUU, PCC, CPC, CPU, and UPU. According to the application concern, it is found that PCU, PUC, PUU, and PCC quads are useful in the electrical application. CPC, CPU, and UPU quads are useful in the hydraulic applications.

None of  $S^6$  triads is found proper in Table 2. Twelve  $S^6$  quads are found in Table 2. Six of them possessed a prismatic joint are PCS, PUS, PSC, PSU, CPS, and UPS. According to the application concern, it is found that PCS, PUS, PSC, and PSU quads are useful in the electrical application. CPS and UPS quads are useful in the hydraulic applications.

#### 2.4.4. Selection of passive limb

According to Table 5, the useful limbs as passive limbs include  $S^2$ ,  $S^3$ ,  $S^4$ , and  $S^5$ . In Table 1, only one  $S^2$  triad, i.e. PR, is found. Four  $S^3$  triads are found in Table 1, which are PC, PU, RC and RU. Six  $S^3$  quads are found in Table 2, which are PPP, PPR, PRP, PRR, RPR, and RRR. Rest of the limbs suitable for passive limbs can also be found in Table 1 and 2.

#### 2.4.5. Insertion of saturated limb

The passive limb employing the mobility  $T_q$  same as the mobility of the space  $\lambda$  will yield no motion constraint to the parallel manipulator. Twelve  $S^6$  quads are found in Table 2. The purpose of introducing a passive limb could be increase the payload capacity, means of suspension, or varieties of means of measurement. The procedures for selecting limbs are summarized in Fig.12

## 2.5 Design Examples of 3-dof TOPM

A particular version of TOPM is to locate one end of the passive limb on the center of the end-effector. According to Table 5, all active limbs are saturated limbs. The ambient space of the end-effector is then determined by the ambient space of the passive limb. For instance, the PUS may be chosen as the  $S^6$  active limb for the specified electrical application. The  $S^3$  passive limb may be chosen from four  $S^3$  triads of Table 1 and six  $S^3$  quads of Table 2. The following examples are shown for different tasks.

### 2.5.1. Parallel manipulated Cartesian machine

The passive limb is selected to be PPP quad from Table 2. In case that all three prismatic joints are axially orthogonal to one another, the resulting parallel manipulator is a particular Cartesian machine as shown in Fig. 13. The ambient space of the passive limb, so does of the end-effector, is the space of three axes of translation.

### 2.5.2 Parallel manipulated wobble machine

The passive limb is selected to be PU triad from Table 1. In case that the prismatic joint is vertically aligned with the gravity direction, and the axially perpendicular revolute joints are located very near to the center of the end-effector, the ambient space of the end-effector is the vertical axis of translation with two axes of rotations. This parallel manipulator is called the wobble machine as shown in Fig. 14. The wobble machine may be used in the application of high speed machining since the rotation in the yaw-direction is eliminated by the

presence of the prismatic joint. The end-effector should be able to carry a spindle. Incorporated with a classical XY worktable, the wobble machine may form a 5-axis CNC structure.

### **2.5.3 Parallel manipulated rotation machine**

The passive limb is selected to be S dyad, i.e. two links connected by a spherical joint. In case that the spherical joint is near to the center of end effector, the ambient space of the end-effector is the three axes of rotations. The resulting parallel manipulator is the rotating machine as shown in Fig. 15. The rotating machine can be used in place of the robot gripper based on gear-train mechanism.

### **2.5.4 Parallel manipulated cobra-head machine**

The passive limb is selected to be PRP quad from Table 2. In case that the two prismatic joints are axially orthogonal to each other, the ambient space of the end-effector is the one axis of rotation with two axes of translations. The resulting parallel manipulator is the cobra-head machine as shown in Fig. 16. The direction of the translation may be determined by the axis of prismatic joint, so does the two axes of rotation determined by the axes of the revolute joints. The end-effector will perform like the head of cobra that does forward, downward and pitch motion. The cobra-head machine may be used as the motion simulator.

## **2.6 Design Examples of 6-dof FSPM**

A general Hexaslide-based machine tool comprises six distinct rails, as indicated in Fig. 17. The sliders move along their rails, while

the legs of constant lengths are connected to the sliders through universal joints. The other end of each leg is linked to the tool or moving platform through spherical joints. The actuation of the sliders on their respective rail drives the moving platform in space. These designs are all based on scissor drives and are different only in rail arrangements. The up-to-date Hexaslide-based machines may include the Hexaglide as illustrated in Fig. 17(a), consisting of coplanar and parallel rails; the HexaM depicted in Fig. 17(b), consisting slanted rail, and the Linapod as shown in Fig. 17(c), comprising vertically-arranged rails. The Delta Hexaglide (or so called Hexglider) discussed in this study was developed by IMON Inc. The Delta Hexaglide consists of coplanar and triangular rails as illustrated in Fig.18. Figure 19 (a) displays the kinematic structure, and Fig. 19(b) demonstrates a photograph of the Delta Hexaglide platform mechanism.





## **Chapter 3 Design of the Swinging-block and Turning-block Mechanism with special reference to the Mechanical Advantage**

### **3.1 Reviews**

The swinging-block and turning block mechanism are extensively applied in several mechanical fields. Examples of its use include the recent version of the freight truck loading mechanism, the camera variable zoom lens hood, and others [23]-[31]. Its primary advantage, the strong output torque, is generated by the conversion of a linear force into rotation, as often used in engine mechanisms, such as the oscillating-cylinder engine mechanism, depicted in Fig. 20 [32]. Furthermore, following the progress in motor technology in recent years, the high-torque and high-accuracy drive of the micro step-motor has been developed. The design that uses the swinging-block and turning-block mechanism with a confined output rotation angle, normally under  $\pi/2$ , depends on a high reduction ratio, such as 300:1 to support high-precision positioning. Nevertheless, a very small backlash is also required. The important limitation of mechanism is that the relationship between input and output angle is nonlinear. The transmission angle, which determines the mechanical advantage, may vary over a wide range so that the effective torque transmitted to the output link is variable. The mechanical advantage of a particular dimensional design must then be studied. The transmission angle

optimizations for the drag link, the crank-and-rocker, and the four-bar linkage are derived in [33]-[38]. However, none of these studies are directly applicable to the swinging-block or turning-block mechanisms.

### 3.2 Swinging-Block and Turning-Block Mechanisms

Figure 21 presents kinematic inversions associated with various selections of the ground link of the RRRP kinematic chain [32]. Both the swinging block and the turning block are much less well-known than the slider-and-crank from the same RRRP family, since the output cylinder must swing, which raises manufacturing difficulties. However, current technological advances of the linear motor and the helical motor have greatly simplified the swinging cylinder, such as in the sensor pedestal design, shown in Fig.22.

The typical design of such a rotational control member may involve a gear head with the servomotor, which suffers from excessive weight and backlash problems. The alternative design adopts a direct-drive motor, which however, requires a higher electric power than specified. There the design of the turning block mechanism is well conceived since it exhibits a low weight-to-power ratio.

Figure 23 illustrates the kinematic structures of the swinging-block and the turning-block mechanisms.  $L$  represents the length of the ground-link.  $R$  represents the length of the output-link. The slider-link, depending on the distance traveled by the motor, has a variable length  $S$ . With a single D.O.F., either the swinging block or the turning block mechanism is driven by the input variable  $S$ .

The simple trigonometric relation determines the input variable  $S$

as a function of the internal angles  $\theta$  and  $\phi$  as follows.

$$S = \frac{L \cdot \sin \theta}{\sin \phi} \quad (3.1)$$

$$S \cdot \cos \phi + L \cdot \cos \theta = R \quad (3.2)$$

The cosine law yields,

$$S^2 = R^2 + L^2 - 2RL \cdot \cos \theta \quad (3.3)$$

The internal angle  $\phi$  is known as the transmission angle of the swinging block or the turning block mechanism. In Fig.23,  $F_i$  represents the input force, exerted from the linear actuator.  $TK$  represents the output torque transmitted to the output link, and is a function of the transmission angle, as follows.

$$\begin{aligned} TK &= F_o \cdot R \\ &= F_i \cdot \sin \phi \cdot R \end{aligned} \quad (3.4)$$

The mechanical advantage is maximum only when  $\phi = \pm 90^\circ$ , discouraging the use of swinging block or turning block mechanism to beyond its positions of singularity,  $\phi = 180^\circ$ .

### 3.3 Maximum Average Mechanical Advantage

In practice, the swing angle of the output link is specified in the design of the swinging block or turning block mechanism; that is, on the range of  $\theta$  is specified. A set of dimensions of the swinging block or the turning block mechanism must be determined to optimize the mechanical advantage over the specified range  $\pm \varepsilon$ , about the middle-angle  $\theta_o$  of the swing angle  $\theta$ , as shown in Fig. 24. Its application must be limited to  $\varepsilon < 90^\circ$  to avoid the singularity. For example, the turning block mechanism may be required to function

over a range of swing angle of  $\varepsilon = 25^\circ$ , a typical value for radar applications. Nevertheless, the workspace of the swinging block mechanism must also be considered.

According to Eqs. (3.1) and (3.2), the transmission angle  $\phi$  relates to the swing angle  $\theta$  as follows.

$$\phi = \tan^{-1}\left(\frac{L \cdot \sin \theta}{R - L \cdot \cos \theta}\right) \quad (3.5)$$

The average mechanical-advantage  $E$  could be expressed as the integral form of the torque with respect to the swing angle  $\theta$ , over a specified range  $\pm \varepsilon$  with respect to the designing parameter, the middle angle  $\theta_o$ , as follows.

$$\begin{aligned} E &= \frac{1}{2\varepsilon} \int_{\theta_o - \varepsilon}^{\theta_o + \varepsilon} T d\theta \\ &= \frac{1}{2\varepsilon} \int_{\theta_o - \varepsilon}^{\theta_o + \varepsilon} F_i \cdot R \cdot \sin \phi d\theta \end{aligned} \quad (3.6)$$

Substituting Eq. (3.5) into the above equation yields,

$$E = \frac{F_i}{2\varepsilon} \cdot [\sqrt{R^2 + L^2 - 2RL \cos \theta}]_{\theta_o - \varepsilon}^{\theta_o + \varepsilon} \quad (3.7)$$

The maximum average mechanical-advantage with respect to the design parameter  $\theta_o$  is obtained by finding the stationary value of the average output torque  $E$ , as follows.

$$\frac{dE}{d\theta_o} = \frac{F_i}{2\varepsilon} \cdot L \cdot R \left\{ \frac{\sin(\theta_o + \varepsilon)}{S_{\max}} - \frac{\sin(\theta_o - \varepsilon)}{S_{\min}} \right\} = 0 \quad (3.8)$$

where,

$$S_{\max} = \sqrt{R^2 + L^2 - 2RL \cdot \cos(\theta_o + \varepsilon)} \quad (3.9a)$$

$$S_{\min} = \sqrt{R^2 + L^2 - 2RL \cdot \cos(\theta_o - \varepsilon)} \quad (3.9b)$$

According to Eq. (3.3),  $S_{\min}$  and  $S_{\max}$  are actually the minimum

and maximum length extended by the linear actuator, respectively.

In general cases, over a finite swing range  $\varepsilon$ , the input force  $F_i$ , the ground link length  $L$ , and the output link length  $R$ , must all be non-zero, (8), yielding,

$$\left(\cos\theta_o - \frac{R}{L}\cos\varepsilon\right)\left(\cos\theta_o - \frac{L}{R}\cos\varepsilon\right) = 0 \quad (3.10)$$

The optimal solution for  $\theta_o$  is obtained, yielding the maximum average mechanical-advantage as,

$$\cos\theta_o = \frac{R}{L}\cos\varepsilon \quad (3.11a)$$

or,

$$\cos\theta_o = \frac{L}{R}\cos\varepsilon \quad (3.11b)$$

The second derivative of Eq. (3.7), which equals the first derivative of Eq. (3.10) multiplied by a constant  $k$ , may be written as,

$$\frac{d^2E}{d\theta_o^2} = k \cdot H \cdot \sin\theta_o \quad (3.12)$$

where,

$$k = \frac{F_i}{2\varepsilon} \cdot L \cdot R > 0$$

and,

$$H = 8 \cdot \frac{R}{L} \cdot \cos\theta_o - 4 \cdot \cos\varepsilon \left(1 + \left(\frac{R}{L}\right)^2\right) \quad (3.13)$$

The middle-angle  $\theta_o$  is selected to be positive, as shown in Fig. 24; such that  $\sin\theta_o > 0$ . Hence the sign of Eq. (3.12) depends on the sign of  $H(\theta_o)$ . For a design free of any singularity,  $\varepsilon < 90^\circ$ , that is  $\cos$

$\varepsilon > 0$ , is required. Substituting Eq. (3.11a) into Eq. (3.13), yields,

$$H = 4 \left( \left( \frac{R}{L} \right)^2 - 1 \right) \cdot \cos \varepsilon \quad (3.14a)$$

However, substituting Eq. (3.11b) into Eq. (3.13), yields,

$$H = 4 \left( 1 - \left( \frac{R}{L} \right)^2 \right) \cdot \cos \varepsilon \quad (3.14b)$$

The maximum value is obtained from either Eq. (3.11a) or Eq. (3.11b) by setting  $H < 0$ , which parameter depends on the  $R/L$  ratio of the design. That is, the result of Eq. (3.11a) for  $L > R$  and that of Eq. (3.11b) for  $R > L$  are used to obtain the maximum value.

### 3.4 Optimal Design

Assuming no energy loss due to friction in the joints or any other viscous damping, the energy output to the output link equals the energy input from the linear actuator, since the total energy is conserved. That is,

$$\int_{S_{min}}^{S_{max}} F_i ds = \int_{\theta_o - \varepsilon}^{\theta_o + \varepsilon} TK d\theta \quad (3.15)$$

where  $S_{min}$  and  $S_{max}$  are defined in Eqs. (3.9a) and (3.9b), respectively. Given a constant input  $F_i$ , Eq. (3.15) may be reformulated as follows.

$$E = \frac{F_i}{2\varepsilon} D \quad (3.16)$$

where  $D$  denotes the required travel span of the linear actuator,

$$D = S_{max} - S_{min}$$

$E$  is the average mechanical advantage, which was previously defined in Eq. (3.6). Equation (3.16) shows that the average mechanical advantage  $E$  is proportional to the distance traveled by the

linear actuator  $D$ . Therefore, a linear actuator that can be extended farther is always preferred for its greater mechanical advantage.

According to Eq. (3.1), the transmission angles  $\phi_{min}$  and  $\phi_{max}$  are defined as follows.

$$\sin \phi_{min} = \frac{L \cdot \sin(\theta_o + \varepsilon)}{S_{max}} \quad (3.17a)$$

$$\sin \phi_{max} = \frac{L \cdot \sin(\theta_o - \varepsilon)}{S_{min}} \quad (3.17b)$$

Where  $\theta_o$  fulfills one of Eqs (3.11a) and (3.11b) to yield the maximum average mechanical advantage. According to Eqs. (3.8), (3.17a) and (3.17b),

$$\sin \phi_{min} = \sin \phi_{max} \quad (3.18)$$

Thus, the travel span of the linear actuator  $D$  can be related to the link length  $L$ , as follows.

$$D = \frac{L}{\eta} \cdot \{\sin(\theta_o + \varepsilon) - \sin(\theta_o - \varepsilon)\} \quad (3.19)$$

where  $\eta$  represents the minimum mechanical advantage and,

$$\eta = \sin \phi_{min} = \frac{TK}{F_i \cdot R}$$

The design problem concerns five design parameters,  $D$ ,  $L$ ,  $R$ ,  $\eta$  and  $\varepsilon$ . These five design parameters uniquely determine the four turning block mechanism link lengths and one the middle swing-angle. Of the five design parameters, the swing angle span  $\varepsilon$  is provided as a design specification. This set of design parameters can again be normalized by normalizing the link length. Consequently, only three normalized design parameters are to be determined; they are  $\eta$ ,  $D/R$  and  $L/R$ .

Optimal design problems may be separated into two categories.

The first is for  $L \geq R$ . Equation (3.11a) is applied to find the optimal solution. Manipulating and simplifying Eqs. (3.17a), (3.11a) and (3.9a) yield,

$$\eta = \cos \varepsilon \quad (3.20)$$

For  $L \geq R$ , the design parameters  $\eta$  and  $\varepsilon$  are not independent. In design procedure,  $\eta$  can not be freely specified for a given swing span  $\varepsilon$ . Substituting Eq. (3.11a) into Eq. (3.19) and combining with equation (3.20) yields,

$$\frac{D}{R} = 2 \cdot \sin \varepsilon \quad (3.21)$$

The second category of problem has  $L < R$ . Equation (11b) is applied to find the optimal solution. Manipulating and simplifying Eqs. (3.17b), (3.11b) and (3.9b) yields,

$$\text{for } R \cdot \sin \theta_0 > L \cdot \sin \varepsilon : \eta = \frac{L}{R} \cdot \cos \varepsilon \quad (3.22a)$$

$$\text{for } R \cdot \sin \theta_0 < L \cdot \sin \varepsilon : \eta = -\frac{L}{R} \cdot \cos \varepsilon \quad (3.22b)$$

Since Eq. (3.22a) (3.22b) contradicts the condition that  $L < R$ , no optimal solution exists. The second category is discarded because of the need to obtain a good mechanical advantage.

### 3.5 Design Procedure:

The design procedure is summarized as follows.

Step 1: Specify  $\varepsilon$

Step 2: Set the  $L/R$  ratio to no less than 1.

Step 3: Obtain  $\eta$  and the  $D/R$  ratio from Eqs. (3.20) and



(3.21), respectively.

For the configuration illustrated in Fig. 25,  $R = 1$ ,  $L = 2$  and  $\varepsilon = 30^\circ$  are set. The optimal value of the design parameter  $\theta_o$  is obtained from Eq. (3.11a), yielding  $\theta_o = 64.34^\circ$  and  $D/R = 1$ . Figure 26 presents more general cases subjected to different  $\varepsilon$  versus  $D/R$  and  $\varepsilon$  versus  $\eta$ . Figure 27 plots the curved surface of  $\varepsilon$  and  $R/L$  versus  $\theta_o$ .



## Chapter 4 Robotic Safeguard System and Workspace Analysis

### 4.1 Reviews

The conventional robotic safeguard systems have been developed from the mechanical hardware safeguard systems to the systems involving electrical hardware devices (such as emergency stop switch, dead-man switch, limit switch, etc.), and the safeguard systems inside and outside the robot working areas. The existing sensor warning systems for the robotic safeguard can be divided into the follows [39]: warning sign system, safety barrier device, pressure pad, inferred, capacitor, microwave, ultrasound, magnetic field, video image, etc. Sensors as defined in this context as the devices that detect if there anyone exists, based on the physical features. These methods have been used for a long time, and work well, however the robot application becomes wider, the conventional sensors warning method show its insufficiency of flexibility and impotence. The robot static positioning problem can be solved via the forward and backward kinematics [40], therefore the robotic movement can be displayed as the animation, and the collision detection can be performed by the computers.

Three levels [41, 42] of the robotic safety envelopes are defined. Level 1 is the maximum envelope, Level 2 is the restricted envelope, and Level 3 is the operating envelope. The robot's virtual boundary is defined as the Level 3 area, which is dynamically varied with the

operating conditions of the manipulator. The active robotic safeguard system involves not only the static protection areas (Level 1 and Level 2), but also the dynamic area (Level 3). The robot language is implemented for the robot movements, which involve the three levels of the robotic safety commands.

The International Standards Organization (ISO) introduced the Open Systems Interconnection (OSI) Reference Model [43], the layered network architecture, with the goal of international standardizing protocols governing the networking communication. However Transmission Control Protocol/Internet Protocol (TCP/IP) [44, 45, and 46] doesn't directly follow the OSI model. Although each network model has the goal of facilitating communication among different types and models of computers, and operating systems, the implementations of each network model present a variety of aspects. Whereas the OSI model was driven by a large standards organization, it took a long time to formulate a draft and adopt it as a standard. In a different situation, TCP/IP was driven by the immediate need of the United States government. The development of TCP/IP isn't burdened with the same stringent requirements as OSI. Local Area Network (LAN) [47] is a data communication network, typically a packet communication network, limited in geographic scope. A local-area network generally provides high-bandwidth communication over inexpensive transmission media. A local-area network is composed of hardware elements and software elements. Hardware elements belong to three basic categories: a transmission medium, a mechanism for control of transmission over the medium, and an interface between the network and devices that are connected to the network. The software

elements are the sets of protocols, implemented in the devices connected to the network, that control the transmission of information from one device to another via the hardware elements of the network. These protocols function at various levels from data-link layer protocols to application layer protocols. Also, LANs are characterized by a large and often variable number of devices requiring interconnection. LANs have a high bandwidth channel with short propagation delays, however shared by many independent users. A Web server [48] does a great deal of work in making Web pages and sites available to browsers. They are the linking mechanism between you and the Web, between people and pages. Web servers consist of special hardware and software that make it possible to carry out browser request. In this thesis, the client-server architecture is defined as the basis for communication between two robotic programs called the client and the server. A server is any application that provides a service to a network user. A client is any program that makes a request to a server. In general, a client and a server run on different computers. Client-server architecture contrasts with the classical centralized architecture popularized by typical mainframe installations. In a centralized environment, the “clients” are little more than dumb terminals that act as simple data entry / display devices. There’s a minimum of work done at the terminal. The user typically fills in the fields of a form before sending the field data to the central computer. All processing and screen formatting are done on the central computer, and dumb terminal simply displays the preformatted data. In a client-server environment, the client has much greater ability and more freedom with the final visual presentation of the data to the user.

Instead of the data being preformatted to match the way it will be viewed, they are transferred in its “raw” format to the application running on the client computer, which “decides” itself how to display that data. Thus, the “front end” that the user sees can be customized while the “back end” remains unchanged. The Centralized Monitoring and Control Computer is the “client” and the individual robot is the “server”. Any program can be opened to handle several connections simultaneously, to play both client and server at the same time.

## **4.2 The Active Multi-Robotic Safeguard System**

### **Architecture**

The traditional safeguard system is integrated with sensors, and then installed into the hazard areas statically. However, with the application of the robot increases, the traditional safeguard system shows its insufficiency of flexibility and efficiency. The intelligent safeguard system is generated, with some kinds of the high accuracy and performance devices integrated, and the software can be controlled for setting up the hazard areas dynamically, with the different operation types and locations of the robot. The omni-direction magnetic position trackers are integrated with the robots and the operator, with the client-server based networking system, software and hardware integrated, the complete real time data of the multi-robot kinematics and operator movement can be obtained, as shown in Figure 28.

The 3D space objects can be represented in the coordinate relative to the absolute coordinate system, and their position and orientation is in terms of the 4 by 4 homogeneous transformation

matrix which combines both rotation and translation process in a single matrix. During the time  $t$ , the point  $Loc$  in object A relative to the absolute coordinate can be represented as

$$Loc^{(0)}(t) = T_{A \rightarrow 0}(t)Loc^{(A)} \quad (4.1)$$

For calculating the collision detection between object A and object B, the point  $Loc$  on object A relative to the local coordinate of the local coordinate can be represented as

$$Loc^{(B)}(t) = T_{B \rightarrow 0}(t)Loc^{(0)} \quad (4.2)$$

Substitute equation (4.1) into (4.2), then

$$Loc^{(B)}(t) = [T_{A \rightarrow 0}(t)]^{-1} T_{B \rightarrow A}(t)Loc^{(0)} \quad (4.3)$$

Assume the end point  $Loc$  is inside the range of object B, then the object A and object B are in the collision condition.

As the depicted previously, the hazard areas determination for the intelligent safeguard system may vary from the robot operational conditions. In order to fortify the safeguard, the hazard areas may be enlarged. Figure 29 shows the coordinate system for the robotic collision detection. Object A refers to the operator, other people, or the robot's manipulator "wearing" the positioning sensor, and object B represents the main robotic system with positioning sensor also. Referring to this figure, the robot movement will be in slower speed when any objects are falling inside the safety envelope Level 2 (hazard zone), and the robot will be fully stopped when any objects are inside the safety envelope Level 3. The safety envelope Level 3 is defined as the robot's virtual boundary in this thesis.

With the 3D monitoring display implemented, the robot false movement can be detected, then the three-level robotic safeguard system is functional, and the robot speed is slowed down, or the brake

system works with the control system, to attain the emergency stop function. The real time robotic collision avoidance function can be used as the safeguard system eventually.

### **4.3 The Robot System Installation and the Kinematics**

#### **Analysis**

The robot systems have been installed with various safety-sensing devices for the complete safeguard system test. There are two kinds of robot systems used in this research. First, the U-type robot is a non-Cartesian coordinate kinematics mechanism, as shown in Figure 30. This type of the robot is widely used for the industrial field. The U-type robot mechanism entity has a three- D.O.F. manipulator with a two- D.O.F. robot wrist. This open-chain robot system is easy to be assembled, and is convenient for performing its manipulator control in the laboratory. The other type of the robot system, as shown in Figure 31, is the parallel-linked robotic system designed for the laboratory used, in order to verify the system theory. The kind of robot platform belongs to the six- D.O.F. Cartesian coordinate kinematics mechanism system, with feedback, high compliance and stability.

For obtaining the robot motion in the space, the kinematics is a very common problem to discuss the relation between robot joint space and Cartesian space. The kinematics analysis for the robot system is significant in this thesis, to integrate the robot virtual reality real time display with the real robot mechanism precisely. For instance, if we choose the U-type robot system as case of the motion mechanism for the robotic safeguard, its forward kinematics can be

described as follows:

$$\begin{matrix} \text{base} \\ \text{tracer} \end{matrix} \mathbf{T} = \begin{bmatrix} \mathbf{r}_{11} & \mathbf{r}_{12} & \mathbf{r}_{13} & \mathbf{p}_x \\ \mathbf{r}_{21} & \mathbf{r}_{22} & \mathbf{r}_{23} & \mathbf{p}_y \\ \mathbf{r}_{31} & \mathbf{r}_{32} & \mathbf{r}_{33} & \mathbf{p}_z \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (4.4)$$

where

$$r_{11} = \cos \theta_1 \cos(\theta_2 + \theta_3)$$

$$r_{21} = \sin \theta_1 \cos(\theta_2 + \theta_3)$$

$$r_{31} = -\sin(\theta_2 + \theta_3)$$

$$r_{12} = -\sin \theta_1$$

$$r_{22} = \cos \theta_1$$

$$r_{32} = 0$$

$$r_{13} = \cos \theta_1 \sin(\theta_2 + \theta_3)$$

$$r_{23} = \sin \theta_1 \sin(\theta_2 + \theta_3)$$

$$r_{33} = \cos(\theta_2 + \theta_3)$$

$$p_x = \cos \theta_1 \left( L_{\text{upper arm}} \cos \theta_2 + L_{\text{lower arm}} \cos(\theta_2 + \theta_3) \right)$$

$$p_y = \sin \theta_1 \left( L_{\text{upper arm}} \cos \theta_2 + L_{\text{lower arm}} \cos(\theta_2 + \theta_3) \right)$$

$$p_z = -L_{\text{upper arm}} \sin \theta_2 - L_{\text{lower arm}} \sin(\theta_2 + \theta_3)$$

Its inverse kinematics can be limited into a three D.O.F. mechanism, and if the input is position, then:

$$\theta_1 = \text{atan2}(-p_y, p_x)$$

$$\theta_3 = \cos^{-1} \left( \left( \frac{p_x}{\cos \theta_1} \right)^2 + p_z^2 - L_{\text{upper arm}}^2 - L_{\text{lower arm}}^2 \right)$$



$$\theta_2 = \cos^{-1} p_z - \operatorname{atan2}\left(L_{\text{upper arm}} + L_{\text{lower arm}} \cos\theta_3 - L_{\text{lower arm}} \sin\theta_3\right) \quad (4.5)$$

If the arm lengths, robot end effector movements are known, then the robot joint rotation angles can be derived from equation (4.5). Therefore, the robot central monitoring system can use of the robot kinematics to implement the 3D real time display, trackball control, position tracker control, and other sensor's control functions, interactively. The kinematics calculation for a parallel robot mechanism is in a similar way.

## 4.4 The Robotic Operation and Control Interface

### Program

The robotic operation and control interface program is written in C++ for Windows. This program contains four control panels: graphic display control, robot multi-axis control buttons, controllable parameter settings, and control status display. The position tracker driver program is added into this program, in order to process the tracker data for the robot upper arm. The interface program for the industrial robot is shown in Figure 32.

The Graphic Display Control window shows the robotic motion. With the command for each axis, the direct kinematics calculation, and the related graphics calculation, the accurate 3D robotic motion display can be obtained. The controller is in the “wait state” for the control system, and the high-level commands are issued from the client (Windows NT) VR control panel, and the motion program is sent out from the fiber network to control the server - the robot. Figure 33 shows the networking for the industrial robot controller interface.

The motion commands for both of the robotic 5-axis joint space and Cartesian space  $x, y, z$  can be issued via the control buttons in the lower position of the control panel. The left and right portions of each button control the positive and negative directions, respectively. The right half portion of the display controls the controllable parameter settings. With these settings, the PID parameters and speed override can be adjusted. The “Test” button in the middle panel can be used to control the mode and the status display switching. Input ports 1~6 show their values in the right portion of the display. Encoder 1~6 and Command 1~6 indicate the encoder position and command values, respectively. The robot upper arm position  $X, Y, Z$  can be obtained with the position tracker installed, then the data can be transferred to the Windows NT-based VR monitoring computer to perform the collision detect calculation and display function. Please refer to Figure 34, the industrial robot control interface display. The position loop input is based on the deviation between position command and position feedback from the encoder. The servo velocity command is then obtained based on the D/A conversion of the position loop output. All of the commands are issued for every constant time interval.

## **4.5 The Robot Language Programming for the Safeguard System**

Safety is the first thing that one should think about when working with robot. So, human safety must be built into a robotics system from the outset, even if humans are never expected to venture into the robot’s work area. The most dangerous situation is that human must work with a robot when repairing it. The next most dangerous

situation in which a human must work with a robot is during the training or programming of a robot. Once again, human may need to be in the robot work cell. The least dangerous situation where human must work with a robot is during the robot normal operation. However, the robot is still dangerous. Therefore, in order to avoid injuring human or equipment, The three-level safety rule is defined and shown in Figure 35. The definition is as follows:

1. Level 1: the maximum envelope. It includes the whole robotic static reachable area. The envelope boundary determines how close an operator is to the robot. The maximum manipulator movement radius is defined as  $r$ .

2. Level 2: the restricted envelope. The envelope describes the area in which the robot can move physically. Any object getting into this area can potentially cause injury. The sector angle is  $t$  and the manipulator movement radius is  $r_i$ .

3. Level 3: the operating envelope. This envelope describes the area in which the robot manipulator is currently positioned, or is moving toward. If the intruder gets into this area, immediate injury will occur. The sector angle is  $t$ , and the safe distance for the manipulator is  $d_i$ .

The syntax of *setting safe-range* instructions is used for the action determination. These rules describe three types of a grammatical construct. For example,

```
LEVEL3 BRAKE : THETA = 20 ,
CLOSE( ARM1, 20) ,
CLOSE( ARM3, 15) ;
```

(4.6)

This instruction defines the Level 3 of the robot safe range. That

action of Level 3 is “brake” when any other object or operator is inside the defined “operating envelope”.

Due to the increasing importance of the robotic safeguard, the robot language is discussed and implemented in the robot controller to perform the safety instruction, to reduce the harm of the operating people and the damage of the machine tools. To design a robotic safeguard system, we would emphasize the member safety should be the first factor to be considered, and then the related equipment. Therefore we can determine some safeguard parameters, based on these factors.

In order to obtain the above safeguard parametric definitions, the development tool, LEX & YACC (Mortice Kern System Inc.) [49], is used as the interpreter for the robot language. Figure 36 shows the whole interpretative process. As shown as this figure, the interpretative process can be divided into four steps:

Input the robot language program into YACC, then scan the input command and translate it.

Parse and build an expression tree.

Execute the expression tree already built by step two. Based on the expression tree, execute and obtain the parameters in sequence, and then save the results in the table.

We can take an example to realize the whole interpretative process:

```
LEVEL3 BRAKE: TEHTA = 30,  
CLOSE(ARM2, 25)  
CLOSE(ARM3, 30);
```

The above instructions means that we define the Level 3

safeguard envelope, and the associated parameters are:

Sector angle is 30°.

The safety distance from ARM2 is 25 cm.

The safety distance from ARM3 is 30 cm.

If any object is getting into this envelope, then the robot will send BRAKE command as its response.

The interpretative process is as shown in Figure 37.

## 4.6 Client-Server Networking Architecture

In the client-server architecture, the Network Class Library is the fundamental element of the program organization of this thesis. By utilizing the classes, the NETCOMM program can be constructed. As shown in Figure 38, it is very clear to show that there are some classes created in NETCOMM program. Where, “arrow” represents the inherited relationship between two classes. According to the inheritable feature of the object, a subclass will inherit all behaviors of parent class and own some private data or member functions different from the parent class. For example, the class RobotArmClient inherits the class TStreamSocket, such RobotArmClient owns the network communication capability provided by TStreamSocket.

Because the supervision is applied, it is necessary to monitor the returned information and give its command simultaneously. We adopt Full-Duplex Communication to let the information be sent back and the command be provided, therefore they won't interrupt each other.

Because the central supervised computer processes all messages and information, the burden will be heavy. As the workload is growing and can't be handled, the task of performing supervising can't be done

continuously. The controlled-end will produce some unexpected situations. Central-supervised hierarchical architecture is an ideal way to solve the above problem. The concept is “central monitoring and distributed control”. That means the local-end will still function well when the centrally controlled-end goes wrong. By achieving this, the efficiency of the system will be much higher. The client-server mode for the network is very similar to this kind of hierarchical architecture, so it is suitable for remote supervision application. As follows, some classes of this program, and their main functions will be introduced.

#### **4.6.1 Client**

RobotArmClient inherits TStreamSocket, so it also inherits all behaviors except private methods. RobotArmClient has the capabilities for giving commands, and transmitting and receiving data. The class segments are also shown in Figure 38. The most important member function of RobotArmClient is *ProcessRead()*. It is similar to an interpreter but it performs the tasks after receiving data. Another important issue is that RobotArmClient and RobotArmServer must be coordinated with each other, so they must have the same protocol. The content and rule of the protocol will be mentioned in next section. After both sides have same protocol, *ProcessRead()* can make the right explanation, execute right commands and transmit correct data.

The class will be used with other programs. So we use class RobotArmClient to do the integration to eliminate the difficulty and complexity.

#### **4.6.2 Server**

RobotArmServer inherits TStreamSocket as RobotArmClient does, and the most methods are the same as RobotClient. Some methods will be redefined in RobotArmServer, but will not be in RobotArmClient, such as the member function *BeforeCloseSocket()*. *BeforeCloseSocket()* doesn't only handle the normal disconnection, but also charges emergency handling abnormal disconnection, so it's a very important function.

The forming factor of service model is finding some restrictions and drawbacks in a two-layer architecture. For example, it is hard to share the same procedures or objects, the problems of security, and the main restriction from the browser. So service model is a good solution for it. Service model is divided into three layers, which are source service, agent service, and target service. The advantages of this model are easy to manage clients, powerful interface management, better security, and multipurpose, and so on. Two cases indicate some advantages of server model.

Although two-layer architecture for client-server application programs is simple and convenience to develop, it is not perfect. There are still some restrictions in it. First, it is hard to manage clients. For example, because client application programs could be anywhere on the Internet, if you change a little, you must to update all client application programs. The work isn't easy to do. Second, it is hard to share the same procedures. Third, there are some security problems. Fourth, if the client is an applet, it can't connect anywhere except the place which the applet is loaded. The reason is to prevent security problems. So in this case, two-layer architecture is not suitable but three-layer architecture is. Three-layer architecture is that adds a layer



between source-end and target-end. Figure 39 indicates three-layer architecture. Clearly to say, they have the relationship of control-end and controlled-end. Three-layer architecture is without the restrictions of two-layer architecture. First, the whole architecture is hierarchy. If it is designed properly, the change of some services provided by second layer changes doesn't affect the other services. If the change is internal operation rather than interface, no services are needed to change. Second, agent server indicating the server of second layer is designed for multipurpose, so its services can be shared with clients, which is applied for various purpose. Third, source-end doesn't directly connect to the target-end, so there is no back door that allows users to have a chance to destroy target-end. We can say that the three-layer architecture is securer than the two-layer architecture. Fourth is that the coding of clients become simpler.

Before the two cases are started, we must do some preparatory work. There is a test program used to test the three-layer architecture and protocol in Figure 40 (a). The purpose of the test program is to make sure the whole architecture is okay, and the server of second layer can parse the incoming messages, and dispatch the message correctly. The test program can be used not only as source-end but also as target-end, so the agent server can cooperate with test program to simulate the operation of three-layer architecture. In Figure 40 (b), there is a smaller window, and its functions provide all defined types. After the "send" button is depressed, it makes a frame, and sent it to target-end. Server design is not an easy work. Some factors must be considered. For instance, the security issue is an important part of network programming. We use some ways considered to achieve the



purpose of protection. Another issue is how to make the server robust, if the server is not robust, there may be a big trouble, because the agent server of second layer is the kernel of the whole architecture. Once the agent server is broken down, the whole works must be affected. This is why we use Java to write the agent server. We can utilize exception handling to prevent any condition that could happen to server. From the test program, the agent server's problems can be found out in advance. The following are two cases for the robotic safeguard networking study.

#### **4.7 Robot Remote Supervision (Case One)**

With the World Wide Web (WWW) and Java programming [50, 51, and 52] integrated into the robot controller, the robot supervision is implemented. Figure 41 shows the result of executing robot client. The right part of the figure is an applet of robot, and the left part is descriptions of what the robot is used for.

The entire architecture shown in Figure 42, is a three-layer architecture. The source-end is a browser, and the target-end is the robot controller. The user interface of robot is an applet embedded in a home page. It represents no matter where you are, as long as you are on Internet, with the browser, you can retrieve the home page, and then you can supervise remotely.

In this case, we can find out its advantages and convenience. First, as above-mentioned, there is only one browser, and then the applet does not need to exist in the computer. If you want to update the program, it doesn't need to care about source-end. It is very convenient to maintain the program, because the program is located on

the Web site. Second, the program is architecture-neutral, so it is portable. It means that once we write the program, it can be running in all platforms. The characteristic is very convenient and efficient for programmers. Third, syntax and style of Java are very similar to C++, so it isn't difficult to modify the programs. We should be noticed that the global variables must be included into a class, and Java doesn't support multiple inheritances and omits pointer. Forth, the program is response for updating the display of robot, giving commands and receiving new data, and displaying the data. In this case, using multithreading is the most convenient approach for those tasks of the program. The multithreading makes the program running smoothly.

#### **4.8 Virtual Reality Remote Supervision for Safeguard of Robotics (Case Two)**

This is the architecture of virtual reality for safeguard of robotics in Figure 43. In this case, the source-end is not an applet rather a C++ program and we will present the advantages of three-layer architecture. Figure 44 shows the photograph of the two-robot system.

If only one robot is under supervision, the two-layer architecture is better than three-layer. If we want to supervise more, there will be a problem. The reason is that when we want to add a robot under two-layer architecture, we must modify the network program. To modify the network program is inconvenient and inefficient. In case two, we only need to modify the configuration file under three-layer architecture. The network programs of source-end or controlled-end don't need to be modified. Another advantage is that when the IP address of one controlled robot changes, the source-end doesn't need

to know.

The second layer is like a black box. It charges the locations of controlled-end or other services. It makes the job of source-end for network easy to do. In the thesis, the three-layer architecture is excellent for remote supervision. The architecture has many advantages, like reusability, maintenance, and flexibility, and does many tasks, like security protection, asynchronous network operation, parsing incoming and outgoing data, dealing with errors, monitoring messages, and so on.



## **Chapter 5 Workspace and Dexterity Analyses of the Stabilizer (Delta Hexaglide Platform)**

### **5.1 Reviews**

The best-known extensible limb-based motion platform is the Stewart platform developed by Stewart [53] as a flight simulator in 1965. Many studies on parallel manufacturing with six D.O.F. have been published. The complete workspace of the “Stewart platform” is a six-dimensional space for which the complete graphical representation is extremely difficult to obtain. Ji [54] introduced the concept of “vertex space” of the workspace, evaluating the workspace using a search technique [55] based on inverse kinematics. Merlet [56][57] determined a six-dimensional workspace among different parallel manipulator platforms, which has been described through schemes based on a full discrimination of the Cartesian space. Gosselin [58] determined a three-dimensional translational workspace with a constant-orientation workspace.

Other motion platforms are called the slider-based platforms. These platforms include the Hexaglide [59], developed at ETH Zurich; the HexaM [60], developed by Toyota; the Linapod [61] developed at University of Stuttgart, and the ParaDex [62] built by Stoughton and a team from Sandia, NIST, and Case Western. Wang and et al [63] proposed a workspace analysis for ParaDex. Bonev and Ryu [64] also presented a workspace analysis and an extensive literature review for the general six D.O.F. PUS

(Prismatic-Universal-Spherical) parallel manipulators. Kim and Ryu [65] derived closed-form dynamics equations for the general six D.O.F. PUS parallel manipulators. Brian and Robert [66] demonstrated the workspace for modification 6-PSU (Prismatic-Spherical-Universal) platform. Rao and et al [67] presents workspace and dexterity analyses of a class of Hexaslides for a machine tool application, however no graphical presentation of workspace is introduced.

The platform path planning generally improves its dexterity. Stoughton, Klein, Pittens et al [68]-[70] modified the Stewart platform with improved dexterity, and also discussed dexterity measurement and optimization. Dean and Michael [71] demonstrated the NASA Ames Vertical Motion Simulator boundary dexterity.

Most literatures discuss the workspace analysis from the view of manipulator operators through defining the control area boundary in terms of “discrete points”. However, the performance indices such as the shape and complexity of the workspace which play important role of manipulator design are not included in the previous literature. The main purpose of this study is to find a set of performance indices which is good for justifying different 6 D.O.F. parallel mechanism designs. Some performance indices, such as the workspace shape complexity utilizing the Marching Cubes Algorithm (MCA) information, in this chapter are introduced the first time in the literature. This chapter starts from defining the shape and complexity of the workspace and proceeds to performing a parametric design of the Delta Hexaglide Platform.

Marching Cubes Algorithm (MCA) was first introduced by W. E. Lorensen and H. Cline [72] in 1987. MCA adopts the information at

the corners of a voxel to construct a surface that approximates the original surface. Nielson and Hamann [73] demonstrated an ambiguity in a cube's faces, when all four edges of the face are intersected. Matveyev [74] [75] discussed the interior ambiguity problem, resolving the ambiguous case by considering the behavior of the tri-linear function along the cell diagonals. Natarajan [76] independently recognized additional ambiguities in the representation of the tri-linear interpolate in the cube's interior. Chernyaev [77] definitively classified the ambiguities that can arise, discovering 33 different cases.

## 5.2 Delta Hexaglide Platform

The sliders move along their rails, while the legs of constant lengths are connected to the sliders through universal joints. The other end of each leg is linked to the tool or moving platform through spherical joints. The actuation of the sliders on their respective rail drives the moving platform in space. The Delta Hexaglide (or so called Hexglider) discussed in this study was developed by IMON Inc. The Delta Hexaglide consists of coplanar and triangular rails as illustrated in Fig. 18. Figure 19(a) displays the kinematic structure, and Fig. 19(b) demonstrates a photograph of the Delta Hexaglide platform mechanism.

### 5.2.1 Inverse Kinematics

Inverse kinematics attempts to identify the input sliding distance  $d_i$  of each of the six sliders,  $i = 1$  to 6, given the output position and orientation of the moving platform. Figure 47 illustrates the top and

front view of the Delta Hexaglide platform. A payload, such as the simulator cockpit, is fixed on top of the upper platform. The output parameters of the upper platform are represented in terms of the position  $\mathbf{U} = [U_x \ U_y \ U_z]^T$ , and the rotational transformation  $\mathbf{R}$  is a  $3 \times 3$  matrix

$$\mathbf{R} = \mathbf{R}(\alpha, \beta, \gamma) = \begin{bmatrix} c\alpha \cdot c\beta & c\alpha \cdot s\beta \cdot s\gamma - s\alpha \cdot c\gamma & c\alpha \cdot s\beta \cdot c\gamma + s\alpha \cdot s\gamma \\ s\alpha \cdot c\beta & s\alpha \cdot s\beta \cdot s\gamma + c\alpha \cdot c\gamma & s\alpha \cdot s\beta \cdot c\gamma - c\alpha \cdot s\gamma \\ -s\beta & c\beta \cdot s\gamma & c\alpha \cdot c\gamma \end{bmatrix}$$

Where, for example,  $c\beta = \cos\beta$  and  $s\alpha = \sin\alpha$ .

A  $3 \times 1$  column vector  $\mathbf{N}_i$  denotes the position of the  $i$ th ball joint assembled on the upper plate for  $i = 1$  to 6.

$$\mathbf{N}_i = \mathbf{U} + \mathbf{R}(\mathbf{N}_{i0} - \mathbf{U}_0)$$

Where the  $3 \times 1$  column vector  $\mathbf{U}$  represents the center position of the upper platform.

$$\mathbf{U} = \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix}$$



Here,  $\mathbf{N}_{i0}$  and  $\mathbf{U}_0$  are the initial positions of  $\mathbf{N}_i$  and  $\mathbf{U}$  respectively.

Another  $3 \times 1$  column vector  $\mathbf{M}_i$  is the position of the  $i$ th ball joint assembled on the slider for  $i = 1$  to 6. All supporting limbs are identical, and their link lengths are  $L$ . The following equation is obtained

$$(\mathbf{N}_i - \mathbf{M}_i)^T (\mathbf{N}_i - \mathbf{M}_i) = L^2 \quad \text{for } i = 1 \text{ to } 6 \quad (5.1)$$

where

$$\mathbf{N}_i = \mathbf{U} + r_{upper} \mathbf{R} \begin{bmatrix} \cos \phi_i \\ \sin \phi_i \\ 0 \end{bmatrix}$$

$$\mathbf{M}_i = r_{base} \begin{bmatrix} \cos \phi_i \\ \sin \phi_i \\ 0 \end{bmatrix} + d_i \begin{bmatrix} -\sin \phi_i \\ \cos \phi_i \\ 0 \end{bmatrix}$$

where  $\phi_i$  represents an angular constant, given by

$$\phi_i = 2\pi \left(1 - \frac{\text{floor}(i/2)}{3}\right)$$

The scalar value  $r_{base}$  is the radius of an inscribed circle of the base triangle, and  $\text{floor}()$  denotes a function which truncates a real value into an integer. Thus input sliding distances  $d_i$  may be obtained by solving the following second-order algebraic equations.

$$d_i^2 - 2d_i \left( \mathbf{U}^T + r_{upper} \begin{bmatrix} \cos \phi_i \\ \sin \phi_i \\ 0 \end{bmatrix}^T \mathbf{R}^T \right) \begin{bmatrix} -\sin \phi_i \\ \cos \phi_i \\ 0 \end{bmatrix} = L^2 - \mathbf{U}^T \mathbf{U} - r_{base}^2 - r_{upper}^2 -$$

$$2\mathbf{U}^T (r_{upper} \mathbf{R} - r_{base} \mathbf{I}_{3 \times 3}) \begin{bmatrix} \cos \phi_i \\ \sin \phi_i \\ 0 \end{bmatrix} + r_{upper} r_{base} \begin{bmatrix} \cos \phi_i \\ \sin \phi_i \\ 0 \end{bmatrix}^T (\mathbf{R}^T + \mathbf{R}) \begin{bmatrix} \cos \phi_i \\ \sin \phi_i \\ 0 \end{bmatrix}$$

(5.2)

for  $i = 1, 3, 5$ .

Two solutions, such as  $d_i^{(1)}$  and  $d_i^{(2)}$ , providing  $d_i^{(1)} > d_i^{(2)}$ , for each of the three equations in Eq.(5.2). Due to geometrical symmetry of the linkage mechanism, we let  $d_i = d_i^{(1)}$  and  $d_{i+1} = d_i^{(2)}$ . The inverse kinematics may be summarized as

$$\mathbf{d} = \mathbf{f}(\mathbf{P}) \tag{5.3}$$

where  $\mathbf{P}$  is a  $6 \times 1$  output vector, given by

$$\mathbf{P} = \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\theta} \end{bmatrix},$$

and  $\mathbf{d}$  represents a  $6 \times 1$  input vector and defined as



$$\mathbf{d} = [d_1 \quad d_2 \quad d_3 \quad d_4 \quad d_5 \quad d_6]^T$$

Significantly, all sliding distance must be less than half the rail length, i.e.  $|d_i| < L_{rail} / 2$  for  $i = 1$  to 6, else the output position and orientation P is unreachable, i.e. outside the workspace.

### 3.2.2 Dexterity Analysis

Considering the derivative of Eq.(5.1) with respect to time, the following is obtained:

$$(\mathbf{N}_i - \mathbf{M}_i)^T \frac{d(\mathbf{N}_i - \mathbf{M}_i)}{dt} = 0, \text{ for } i = 1 \text{ to } 6 \quad (5.4)$$

or

$$(\mathbf{N}_i - \mathbf{M}_i)^T \dot{\mathbf{M}}_i = (\mathbf{N}_i - \mathbf{M}_i)^T \dot{\mathbf{N}}_i, \text{ for } i = 1 \text{ to } 6$$

The position vectors of  $\mathbf{N}_i$  and  $\mathbf{M}_i$  in the  $S_i$  frame are given as  ${}^{S_i}\mathbf{N}_i$  and  ${}^{S_i}\mathbf{M}_i$  respectively.

$$({}^{S_i}\mathbf{N}_i - {}^{S_i}\mathbf{M}_i)^T {}^{S_i}\dot{\mathbf{M}}_i = ({}^{S_i}\mathbf{N}_i - {}^{S_i}\mathbf{M}_i)^T {}^{S_i}\dot{\mathbf{N}}_i, \text{ for } i = 1 \text{ to } 6 \quad (5.5)$$

Then,

$${}^{S_i}\mathbf{N}_i = \begin{bmatrix} {}^{S_i}N_{i,x} \\ {}^{S_i}N_{i,y} \\ {}^{S_i}N_{i,z} \end{bmatrix}$$

$${}^{S_i}\mathbf{M}_i = \begin{bmatrix} 0 \\ d_i \\ 0 \end{bmatrix}$$

and

$${}^{S_i}\mathbf{N}_i = \mathbf{A}_i \mathbf{N}_i - \begin{bmatrix} r_{base} \\ 0 \\ 0 \end{bmatrix} \quad (5.6)$$

where  $\mathbf{A}_i$  is a constant rotational transformation matrix given by

$$\mathbf{A}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_i & -\sin \phi_i \\ 0 & \sin \phi_i & \cos \phi_i \end{bmatrix}$$

where  $\phi_i$  denotes the angular constant as previously defined in Eq.(5.1).

Since

$$({}^{S_i} \mathbf{N}_i - {}^{S_i} \mathbf{M}_i)^T {}^{S_i} \dot{\mathbf{M}}_i = ({}^{S_i} N_{i,y} - d_i) \dot{d}_i$$

then Eq.(3.5) may be recast as follows:

$$\dot{d}_i = {}^{S_i} \mathbf{r}_i^T {}^{S_i} \dot{\mathbf{N}}_i \quad \text{for } i = 1 \text{ to } 6 \quad (5.7)$$

where

$${}^{S_i} \mathbf{r}_i = \frac{{}^{S_i} \mathbf{N}_i - {}^{S_i} \mathbf{M}_i}{{}^{S_i} N_{i,y} - d_i}$$

According to Eq.(5.6),  ${}^{S_i} \dot{\mathbf{N}}_i$  may be obtained as

$${}^{S_i} \dot{\mathbf{N}}_i = \mathbf{A}_i \dot{\mathbf{N}} = \mathbf{A}_i \left\{ \dot{\mathbf{U}} + \frac{d\mathbf{R}}{dt} (\mathbf{N}_{i_o} - \mathbf{U}_o) \right\}$$

Consequently, the sliding rate of the  $i$ th slider yields

$$\dot{d}_i = {}^{S_i} \mathbf{r}_i^T \mathbf{A}_i \left\{ \dot{\mathbf{U}} + \frac{d\mathbf{R}}{dt} (\mathbf{N}_{i_o} - \mathbf{U}_o) \right\} \quad (5.8)$$

where

$$\frac{d\mathbf{R}}{dt} = \frac{\partial \mathbf{R}}{\partial \alpha} \dot{\alpha} + \frac{\partial \mathbf{R}}{\partial \beta} \dot{\beta} + \frac{\partial \mathbf{R}}{\partial \gamma} \dot{\gamma}$$

and the partial derivatives of rotational transformation matrix  $\mathbf{R}$  appears in Table 6.

The Jacobian matrix,  $\mathbf{J}$ , transforms the sliding rates of the manipulator into the upper platform velocity states as

$$\dot{\mathbf{P}} = \mathbf{J}\dot{\mathbf{d}}$$

Equation (8) may be expressed as follows:

$$\dot{d}_i = {}^{S_i} \mathbf{r}_i^T \mathbf{A}_i [\mathbf{I}_{3 \times 3} \quad \mathbf{B}_i] \dot{\mathbf{P}} \quad (5.9)$$

where  $\mathbf{P}$  is the output velocity vector given by

$$\dot{\mathbf{P}} = \begin{bmatrix} \dot{\mathbf{U}} \\ \dot{\boldsymbol{\theta}} \end{bmatrix},$$

and  $\mathbf{B}_i$  is a  $3 \times 3$  matrix given by

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial \mathbf{R}}{\partial \alpha}(\mathbf{N}_{io} - \mathbf{U}_o) & \frac{\partial \mathbf{R}}{\partial \beta}(\mathbf{N}_{io} - \mathbf{U}_o) & \frac{\partial \mathbf{R}}{\partial \gamma}(\mathbf{N}_{io} - \mathbf{U}_o) \end{bmatrix}$$

Thus, given the sliding rates of the sliders, the upper platform velocities can be calculated directly. In a trajectory planning problem, the upper platform velocities are usually given along a desired path in the upper platform space (cockpit space), and must be converted into the slider rates in the slider space. This transformation requires the inverse transform to be computed as follows:

$$\dot{\mathbf{d}} = \mathbf{J}^{-1} \dot{\mathbf{P}}$$

The following is derived from Eq. (5.6)

$$\mathbf{J}^{-1} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_1 \mathbf{B}_1 \\ \mathbf{C}_2 & \mathbf{C}_2 \mathbf{B}_2 \\ \mathbf{C}_3 & \mathbf{C}_3 \mathbf{B}_3 \\ \mathbf{C}_4 & \mathbf{C}_4 \mathbf{B}_4 \\ \mathbf{C}_5 & \mathbf{C}_5 \mathbf{B}_5 \\ \mathbf{C}_6 & \mathbf{C}_6 \mathbf{B}_6 \end{bmatrix} \quad (5.10)$$

where

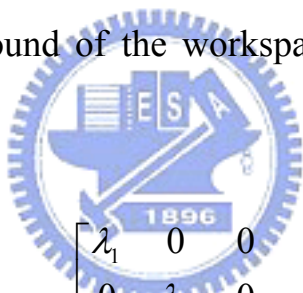
$$\mathbf{C}_i = {}^{S_i} \mathbf{r}_i^T \mathbf{A}_i$$

The singular values of the inverse of the Jacobian matrix may be obtained by a singular value decomposition approach [66]. The

dexterity is based on the inverse of Jacobian matrix calculation expressed as

$$J^{-1} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} \\ J_{31} & J_{32} & J_{33} & J_{34} & J_{35} & J_{36} \\ J_{41} & J_{42} & J_{43} & J_{44} & J_{45} & J_{46} \\ J_{51} & J_{52} & J_{53} & J_{54} & J_{55} & J_{56} \\ J_{61} & J_{62} & J_{63} & J_{64} & J_{65} & J_{66} \end{bmatrix}$$

If  $\det(J^{-1}) = 0$  or  $\text{rank}(J^{-1}) < 6$ , the system rank is decreased and causes linear dependent, then multiple or zero solution will be generated. From the geometric point of view, there will lose at least one D.O.F., and outboud of the workspace could exist in the joint space.



Else eigenvalue( $J^{-1}$ )=

$$\begin{bmatrix} \lambda_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_6 \end{bmatrix},$$

and  $\text{rank}(J^{-1})=6$ .

The ratio between the minimum singular value and the maximum singular value is defined as the parallel manipulator dexterity as the following expression:

$$DX = \frac{|\lambda_{Min}|}{|\lambda_{Max}|} \quad (5.11)$$

Figure 48 shows an example of the dexterity analysis of the Delta Hexaglide platform with dimensional parameters in Table 6.

### 5.3 Marching Cube Method

A voxel, as illustrated in Fig. 49(a), is defined as the center of the cube. A cube, as shown in Fig. 49(b), is defined as the vertices on the eight corners. Methods which simply use voxels to present the surface may cause some visual confusion since each voxel has up to three visible faces. Thus, only three different gray levels, representing three faces with normal directions 1-0-0, 0-1-0, and 0-0-1, are employed in the voxel presentation of the workspace.

The marching cube method was used to present the three-dimensional iso-surface of a user-defined iso-value of a given function. If at least one vertex of a cube has values below the user-specified iso-value, and at least one the other vertex has values above the iso-value, then the cube contributes some components of the three-dimensional iso-surface. By determining which edges of the cube are intersected by the iso-surface, triangular patches can be formed which divide the cube between zones within and outside the three-dimensional iso-surface. By connecting the patches from all cubes on the three-dimensional iso-surface boundary, a surface representation is obtained.

The marching cube employs triangular patches to represent the iso-surface. Each triangular patch consists of three vertices from the mid-points of the twelve edges of the cube, as illustrated in Fig. 50. The cube can have as many as 37 different normal directions for the visible patches, as shown in Table 7.

The vertex value may be binarized and represented by a single bit. If vertex value is higher than the iso-value (inside the workspace) then the corresponding bit is set to logic “1”, otherwise (outside the workspace) the corresponding bit is set to Logic “0”. After all 8 eight vertices of a cube has been examined, an index composed of eight bits for the cube is determined. According to the cube index, there are 256 possible polygonal configurations ( $2^8 = 256$ ) which can be obtained by reflections and symmetrical rotations of 15 unique cases, as shown in Fig.51. Each of the 15 unique cases consists of the different number of patches from Table 7.

### 5.3.1 Workspace Presentation

This research applies the marching cube algorithm for presentation the platform workspace. Marching cubes for workspace boundaries yield much better color resolution, or more gray levels, than the conventional voxel representation does. The scalar function for the workspace of the Delta Hexglide may be defined as follows:

$$g(d) = \prod_{i=1}^6 (1 - h(|d_i| - L_{rail} / 2)) \cdot \sum_{i=1}^6 (7 \cdot \text{Step}(|d_i| - L_{rail} / 2) - 1) \quad (5.12)$$

where  $d = f(v)$ , in which function  $f$  is the inverse kinematics function from Eq.(5.2); the step function is given by

$$\begin{aligned} \text{Step}(x) &= 0 & \text{if } x < 0 \\ &= 1 & \text{otherwise} \end{aligned}$$

and  $h(\bullet)$  represents an impulse function given by

$$\begin{aligned} h(x) &= 1 & \text{if } x = 0 \\ &= 0 & \text{otherwise} \end{aligned}$$

where  $d = f(v)$ , in which the function  $f$  denotes the inverse

kinematics function from Eq.(5.3).

In Eq.(5.12),  $v$  is a  $6 \times 1$  vector, which embodies only three varying entries for the workspace analysis. For example,  $v = \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\theta}^* \end{bmatrix}$  is applied to find the translational workspace with a fixed orientation  $\alpha = \alpha^*$ ,  $\beta = \beta^*$ ,  $\gamma = \gamma^*$ , and  $v = \begin{bmatrix} \mathbf{U}^* \\ \boldsymbol{\theta} \end{bmatrix}$  is used to determine the rotational workspace of a fixed position  $\mathbf{U}^*$ .

The iso-value may be set to 0. The corresponding iso-surface obtained from  $g(d) = 0$  is the workspace boundary. The region given by  $g(d) > 0$  lies outside the workspace. The region given by  $g(d) < 0$  lies inside the workspace.

### 5.3.2 Workspace Examples

The different workspace of Delta Hexaglide platforms may be obtained from assigning different dimensional parameters, i.e.  $r_{\text{upper}}$ ,  $r_{\text{base}}$ ,  $L$ , and  $L_{\text{rail}}$ . Examples of the translational workspace of Delta Hexaglide workspace set  $\alpha^* = \beta^* = \gamma^* = 0$  are given in Fig. 52 with the dimensional parameters specified in Table 6.

Figure 53 illustrates some interesting examples using parameters in Table 6. The workspace island in Fig. 53(a) is obtained when  $\alpha = 12^\circ$  and  $\beta^* = \gamma^* = 0$  and the workspace cavity which results from the degeneracy of the workspace in Fig. 53(b) can be found when  $\alpha^* = 14^\circ$  and  $\beta^* = \gamma^* = 0$ . These two forms of workspace degeneracy are often overlooked by conventional workspace analyses based on boundary tracing. Furthermore, the workspace degeneracy

may yield the workspace to be deformed, broken or vanished. Therefore, the control difficulty may be increased.

## 5.4 Analysis and Design

The workspaces corresponding to different dimensional parameters may be evaluated in terms of different indices. The first index of interest may be the workspace volume which equals the number of the cubes inside the workspace multiplied by the cube volume. The second index is referred to as the shape complexity (SC), which is defined as the ratio between the face area (FA) on the workspace boundary and the workspace volume (WV), as depicted in eq. (5.13).

$$SC = \frac{FA}{WV}$$



(5.13)

With the same volume of the workspace, greater the surface area corresponds to higher the shape complexity. The motion platform should ideally have a large workspace but a small shape complexity. Typically, the workspace volume must be normalized by the space required by the platform. The Required Platform Space (RPS) for the Delta Hexaglide is given by:

$$RPS = k \times r_{\text{upper}} \times L \times L_{\text{rail}}$$

(5.14)

where  $k$  is a geometric constant, ranged from 0 to 1 depending on the parking (homing) position of the upper platform as shown in Fig.



54.

The marching cube method can yield not only a workspace boundary but also the information stipulated by the workspace evaluation. For example, in Fig. 55, the cube volume (0.05) and the workspace volume is calculated as  $24775 \times (0.05)^3 = 3.097$ . The shape complexity of the design in Fig. 55 is 0.556. For a particular parking position as shown in Figure 54,  $k$  may be 0.3. The space required by the platform (RPS) is  $0.3 \times 3.47 \times 3.15 \times 7.7 = 25.25$ . The normalized workspace volume is thus  $3.097 / 25.25 = 0.127$ . Notably, eight times the workspace volume must be accommodated for the platform machine in such dimensional arrangement.

#### 5.4.1 Combination of Workspace and Dexteriy Analysis

The marching cube method generates the surface patches of the workspace boundary and no face in the workspace interior. The interior volume can be exposed by eliminating some portions of surface patches. The dexterity analysis can easily be performed by calculating the singular values of the inverse of the Jacobian matrix as in Eq. (5.10), or graphically through existing software packages, such as MATLAB [67]. The image of the dexterity analysis result can then be appended to the exposed interior of the workspace by known texture mapping techniques such as those given in OpenGL [68]. Figure 56 shows a dexterity analysis result using parameters in Table 6 with the orientation workspace at the home position  $\mathbf{U}^* = [0, 0, 0]^T$ , providing useful and complete information for the designer.

Since three-dimensional space is the most complex visual space possible, the parametric design of the Delta Hexaglride must be divided

into the translational workspace when  $\alpha^* = \beta^* = \gamma^* = 0$ , and the rotational workspace when  $U^* = [0, 0, 0]^T$ , to fit the individually three-dimensional visual spaces. The results of varying  $L$  of Table 6 between 3.15 and 3.40, the result are depicted and compared in Fig.56. The workspace volume and shape-complexity of translational workspace remained almost unchanged, but the workspace volume and shape-complexity of rotational workspace, and the dexterity of both translational and rotational workspace, all changed significantly. Figure 56 reveals that the dexterity improves as  $L$  increases.

## 5.4.2 Dimensional Design

$L_{rail}$  is preferred to be a value as large as possible to allow a maximum workspace. Due to that the rails are not be allowed to intersect one another, once the value  $r_{base}$  has been chosen, the maximum value of  $L_{rail} = 2\sqrt{3}r_{base}$  is determined. Thus only three parameters are left for the workspace design of Delta Hexaglide, which include the radius of upper plate  $r_{upper}$ , the radius of base plate  $r_{base}$ , and the length of the supporting link  $L$ . One may normalize the design by letting  $r_{base} = 1$ . The overall workspace includes a translational and a rotational workspace. Since the maximum vertical translation of the platform is the difference between the highest upper plate height  $L$  and the lowest height  $\sqrt{L^2 - (L_{rail}/2)^2}$ . It can be verified that the maximum vertical translation of the upper plate is monotonic decreasing function of  $L$ . As a result, smaller translational workspace corresponds to larger link length  $L$  as shown in Figure 57. We need only to  $r_{upper}$  in the optimization process of translational

workspace. The result is shown in Figure 59. It is fortunately found for both maximum workspace volume and minimum shape complexity that the optimal dimension is  $r_{\text{upper}} = r_{\text{base}}$ . Preceded from the optimal  $r_{\text{upper}}$  of the translational workspace design, the optimal link length  $L$  may be determined from the optimal rotational workspace design. The result is shown in Figure 59. It is found that the optimal design yields  $L = 1.057 r_{\text{base}}$ .



## Chapter 6 Conclusion

This dissertation has presented analyses methodology of the mechanism gimbal and stabilizer system for the radar image sensor, using the parallel mechanical structure, Delta Hexaglide, for the better performance of payload capability and control response.

Chapter 2 provides a methodology for parallel manipulator design. This methodology consists of procedures for building a parallel manipulator from the primitive limbs. By defining the active limb, we found that the fully-symmetric parallel manipulator employing structurally identical active limbs can only yield two, three and six D.O.F.. The semi-symmetric parallel manipulator which employs structurally different limbs can yield four and five D.O.F.. The task-oriented, parallel manipulators are suitable for specified tasks through introducing proper passive limbs. The saturated limb may be used as the suspension or measurement means through collaboration with dampers and transducers. This chapter has also demonstrated several examples on 3- D.O.F. task-oriented, parallel manipulators, namely the parallel manipulated Cartesian machine, wobble machine, rotation machine, and cobra-head machine. It is believed that many other useful parallel manipulators can be explored further according to this methodology.

Chapter 3 has shown a complete solution for the design optimization problem subjected to constant input force for the general turning-block as well as swinging-block mechanisms with a special reference to the average mechanical energy. The results were given for two types of applications,  $L > R$  and  $L < R$ . No optimal solution exists

in the  $L < R$  category. The rule-of-thumb design procedures allow the engineer to correlate the optimal mechanical advantage with the swing angle span  $\varepsilon$  of the output link. Nevertheless, the workspace associated with the  $L/R$  ratio is submitted to the designer in advance, and the optimal design procedure need not be further verified. The  $D/R$  ratio, which affects the assessed cost of the linear actuator, can be easily determined from the given swing span  $\varepsilon$ . Additional multi-objective optimization of the total cost-performance for different  $L/R$  ratios, and  $\varepsilon$  values, may be performed for different applications in the future. Therefore the optimized gimbal mechanism has been determined and presented with a prototyped model.

In Chapter 4, the transmitter is located on the center of the robotic working area. Receivers are installed on every robot's end effector, and the human body, then select any receiver to move toward one of the robot's virtual boundary. After the safety envelope Level 2 is entered, the robots are sensed and turned into the slower speed, 25 cm/sec. When any object falls inside the robot's virtual boundary, the stop command can be issued from the computer via the Ethernet. The individual robot controller uses the robot language to perform its safeguard functions. The robotic centralized monitoring and control computer maintains the client-server communication functions. Therefore the stop command can be sent to the data buffers. The delay time on network system will impact to the accuracy of the safeguard system. There is an internal counter activated when Windows system boots. It's a high-resolution counter that provides high-resolution elapsed times. The frequency of the counter depends on the hardware performance of the processor. The value of the frequency and the

present counter can be obtained from Win32 API easily. We can send the data of the net memory in one computer to the others and account the value of the counter. When someone receives this data, it sends back the data to the sender. The sender accounts the value of the counter again when it received this. The difference between two values is divided by the frequency, and this value is the network delay. The experimental result of the network delay for the different size of the transmitted data between two robotic controllers is shown in Figure 18. Figure 19 shows the network delay for the different size of the transmitted data among multiple robotic controllers.

The other eminent delay is derived from the update rate for the VR software in the central monitoring system simulation loop. In this thesis, the VR update rate is between 15 and 20 frames/second. Therefore the delay time generated from the VR software simulation loop is no greater than 67ms. The frame rate can be upgraded via either improving the 3D graphic engine or eliminating the graphic polygon number. For the two-robot system, the network delay time (including the agent server) is less than 5.4ms for the case of 256k-byte data transmission, according to Figure 18. For the six-robot system, the network delay (including the agent server) is less than 5.8ms, according to Figure 19. In the typical motor specification, the robot inertial delay for a speed of 25 cm/sec, with no greater than 10 N.m static friction motor torque, is less than 50 ms. Finally, the total delay for the entire multi-robotic safeguard system (six-robot system) is 67ms (VR induced delay) + 5.8ms (network delay) + 50ms (inertial delay), which is no greater than 0.15 second. At the robot safe speed of 25 cm/sec, the robot will advance less than  $0.15 \times 25 = 3.75$ cm,

which is acceptable for the 25cm of the safety envelope Level 3 protection of the safeguard system.

Remote supervision is becoming more and more important now. In the thesis, the approach combined Java, World Wide Web and the service model is generated accordingly. Java has some characteristics, such as portability, security, memory segmentation, and object orientation. Due to these advantages, remote supervision is suitable to be implemented with Java.

The case one of this thesis indicates that using applet of Java to remote supervision is convenient and efficient. Most portions of Java are similar to C++, so modifying the C++ to Java is not difficult. Because the applet is merged in home page, we can get some benefits from the home page. In a program, there must have help to describe some information about it. If the information is out of date, it must be updated. But the task in a program is more difficult than that is in a home page. A home page can dynamically display the proper information.

From the two cases, we can say that constructing a remote supervision architecture by service model is not hard, because we use PC-based to archive our purpose, rather than using special facilities, or special network card. What we emphasize is to use current network resources and facilities to make remote supervision. And the results of those two cases are in accordance with our expectations. With the increasing network bandwidth, the remote robot supervision system will become more practical in the future.

Chapter 5 has presented the analyses of the workspace and dexterity. The workspaces were analyzed by introducing the marching



cube method which permits workspace evaluation including the workspace volume as well as the shape complexity. The comparisons for the various Delta Hexaglide mechanisms due to different parametric designs were made. In addition to the analysis result, the degeneration, cavity, and island of workspace (refer to section 3.2 Workspace Examples) can also be presented in the form of 3D graphics for extensive studies.

The integrated analysis result has been shown using well-known software tools including MATLAB and OpenGL. The trade-off between the available workspace volume, the shape complexity, and the dexterity may be visualized. A multi-objective optimal design is in this study, which has also derived the inverse kinematics required by the marching cube method algorithm and the singular values required by the Delta Hexaglide platform dexterity analysis. Finally, we have presented a design of the Delta Hexaglide platform due to the optimal workspace. With the analyses of the workspace and dexterity, the radome design specification has been determined, the safeguard system using virtual reality (VR) programming to perform the workspace boundary determination for the area outside of the workspace has been designed, and the workspace optimization for the stabilizer has been implemented.

This stabilizer system from the degree freedom analysis is designed with considerations of vehicle space limitation and safety issues. We consider the robot safeguard with virtual boundary techniques, and the other space adjustments. Therefore the stabilizer needs to be rebuilt in the three-dimension Cartesian space by the VR technique, and the dexterity problem can be verified and optimize the



platform design.

By using a six D.O.F. motion base to compensate for the position error, the airborne sensor pointing direction can be stabilized, and the real-time motion compensation for the image radar can be performed accordingly.



## References

- [1] Walter G. Carrara, Ron S. Goodman, Ronald M. Mfajewski, "Spotlight Synthetic Aperture Radar, Signal Process Algorithms," Artech House, Boston/London, 1995.
- [2] C. John, Jr. Kirk, "Motion Compensation For Synthetic Aperture Rada," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-aa, No. 3, May 1975, pp. 338-348.
- [3] D. Stewart, "A Platform with Six Degrees of Freedom," Proc. Inst. Mech. Eng. London, Vol. 180, 1965, pp. 371-386.
- [4] Bo Li, and David Hullunder, "Self-Turning Controller for Nonlinear Inertial Stabilization System," IEEE Transactions On Control System Technology, Vol. 6, No. 3, May 1998.
- [5] Wu-Jong Yu, Chih-Fang HUNG and Wei-Hua CHIENG, "Design of Swinging-Block and Turning-Block Mechanism with Special Reference to the Mechanical Advantage," JSME. Int. J. Series C, Vol. 47, No. 1, 2004, pp. 363-368.
- [6] W-J Yu, C-F Huang, W-H Chieng, C-Y Gau , " The Integrated Application and Research of the Industrial Robotic Safeguard System with Ethernet," IOSH, Journal of Institute of Occupation Safety and Health, Vol. 10, No. 3, 2002, pp. 218-230.
- [7] D. Stewart, "A Platform with Six Degrees of Freedom," Proceedings of the Institute of Mechanical Engineers, London, Vol. 180, No. 5, 1966, pp. 371-386.
- [8] X. Kong and C.M. Gosselin, "Generation and Forward Displacement Analysis of Two New Classes of Analytic 6-SPS Parallel Manipulators," Journal of Robotic Systems, Vol. 18, No. 6, 2001, pp. 295-304.

- [9] N.D. Perreira, "Motions, Efforts, and Actuations in Constrained Dynamic Systems: A Multilink Closed Chain Example," *Journal of Robotic Systems*, Vol. 16, No. 7, 1999, pp. 363-385.
- [10] F. Hao, J.M. McCarthy, "Conditions for Line-Based Singularities in Spatial Platform Manipulators," *Journal of Robotic Systems*, Vol. 15, No. 1, 1998, pp. 43-55.
- [11] J.J. Hall, and R.L. Williams II, "Inertial Measurement Unit Calibration Platform," *Journal of Robotic Systems*, Vol. 17, No. 11, 2000, pp. 623-632.
- [12] D. Zhang, L. Wang, and S.Y.T. Lang, "Parallel Kinematic Machines: Design, Analysis and Simulation in an Integrated Virtual Environment," *ASME J. of Mechanical Eng.*, Vol. 12, No. 3, 2005, pp. 580-588.
- [13] J. Hollingum, "Features: Hexapods to Take Over?," *Ind. Robot*, Vol. 24, No. 6, 1997, pp. 428-431.
- [14] M. Valenti, "Machine Tools Get Smarter," *ASME J. of Mechanical Eng.*, Vol. 117, No. 11, 1995, pp. 70-75.
- [15] G. Yang, I.M. Chen, W.K. Lim, and S.H. Yeo, "Kinematic design of modular reconfigurable in-parallel robots," *Autonomous Robots*, Vol. 10, No. 1, 2001, pp. 83-89.
- [16] R. Cohen, M.G. Lipton, M.Q. Dai, and B. Benhabib, "Conceptual design of a modular robot," *ASME J. of Mechanical Design*, Vol. 114, No. 2, 1992, pp. 117-125.
- [17] D. Schmitz, P. Khosla, and T. Kanade, "The CMU reconfigurable modular manipulator system," Technical Report CMU-RI-TR-88-7, Carnegie Mellon University, 1988.
- [18] Y. Fang and L.W. Tsai, "Structure Synthesis of a Class of 4-DoF

- and 5-DoF Parallel Manipulators with Identical Limb Structures,” *Journal of Robotics Research*, Vol. 21, No. 9, 2002, pp. 799-810.
- [19] J. Angeles, “The Qualitative Synthesis of Parallel Manipulators, Proc. of Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulator,” Quebec, Canada, 2002, pp. 160-169.
- [20] R.O. Ambrose, “Design, Construction and Demonstration of Modular, Reconfigurable Robots,”. PhD thesis, University of Texas at Austin, U.S.A, 1991.
- [21] L.W. Tsai, “Mechanism design: Enumeration of kinematic structures according to function,” CRC Press, 2000.
- [22] G. Liu, Y. Lou, and Z. Li, “Singularities of Parallel Manipulators: A Geometric Treatment,” *IEEE Trans. On Robotics and Automation*, Vol. 19, No. 4, 2003, pp. 579-594.
- [23] Raymond A.Adee, Newton; Ellis E. Adee, Minneapolis, “Fold-back Implement Frame having Angle Adjustment,” US Patent 4,236,585, Dec.2, 1980.
- [24] Yasuo Shimizu; Junji Yuzuriha, “Variable Ratio Steering System,” US Patent 5,174,407, Dec.29, 1992.
- [25] Masaru Abe; Yoshimichi Kawamoto, “Front and Rear Wheel Steering System For a Vehicle,” US Patent 5,199,523, Apr.6, 1993.
- [26] Naoki Ito, “Variable Zoom Lens Hood,” US Patent 5,745,803, Apr.28, 1998.
- [27] Henry Arthur Hopgood, “Rotary Drive Mechanism,” US Patent 5,882,026, Mar.16, 1999.

- [28] Lloyd W. Rogers, Jr; John I. Moceanu, Sterling Hights, “Linkage for Vehical Door Latch” US Patent 5,253,906, Oct.19, 1993.
- [29] Patrick H.O’Brien; Willian J. Zabritski; “Multispeed Shift Linkage Control,”US Patent 4,018,099, Apr.19, 1977.
- [30] Hammer Bengt Olof (DE),“Freight Truck Loading Mechanism” Patent Number: DE19840151, Mac.03, 1999.
- [31] Ullrich Christian (DE), “Pivoting equipment carrier, especially to support desk lamps,” Patent Number: DE19600879, Jul.17, 1997.
- [32] Lung-Wen Tsai, “Mechanism Design: Enumeration of Kinematic Structures According to Function,” ISBN 0-8493-0901-8, TJ175. T78, 2000, pp. 16,119-120.
- [33] F. Freudenstein, and E. J. F. Primrose, “The Classical Transmission- Angle Problem,” Proc. Conf. Mechanism Synthesis, Inst. Mech. Engrs., London, 1973, pp. 105-110.
- [34] K.C. Gupta, “Design of Four-Bar Function Generators with Mini-Max Transmission Angle,” ASME Journal of Engineering for Industry, Vol. 99, No. 2, 1997, pp. 360-366.
- [35] F. Freudenstein, “Designing Crank and Rocker Links with Optimum Force Transmission,” Product Engineering, 1978, pp. 45-47.
- [36] T. E. Shoup, , and B. J. Pelan, “Design of Four-Bar Mechanisms for Optimum Transmission Angle and Optimum Structure Error,” Proceedings of the Second OSU Applied Mechanism Conference, Stillwater, Okla ,1971, pp.4.1-4.9.
- [37] L. W. Tsai, “Design of Drag-Link Mechanisms with Optimum Transmission Angle,” ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 105, No.2 , 1983,

pp. 254-259.

- [38] L. W. Tsai, "Design of Drag-Link Mechanisms with Minimax Transmission Angle Deviation," ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 105, No.2, 1983, pp. 686-691.
- [39] B. S. Dhillon, "Robot Reliability and Safety," Springer-Verlag, New- York, 1991.
- [40] J. J. Craig, "Introduction to Robotics Mechanics and Control," 2nd ed. Addison Wesley, 1989.
- [41] J. H. Graham, etc., "A Safety and Collision Avoidance System for Industrial Robots," IEEE Transactions on Industry Applications, Vol. IA-22, No.1, January/February, 1986, pp. 195-203.
- [42] J. H. Graham, "Safety, Reliability, and Human Factors in Robotic Systems," Van Nostrand Reinhold, 1991, pp. 116-131.
- [43] Arthur Dumas, "Programming WinSock," SAMS, 1995.
- [44] N. Tarek, Saadawi, et al., "Fundamentals of Telecommunication Networks," John Wiley & Sons, 1994.
- [45] D.E. Comer, and D.L. Stevens, "Internetworking with TCP/IP: Vol. III: Client-Server Programming and Applications," BSD Socket Version, Prentice-Hall, Englewood Cliffs, N.J. 1993.
- [46] D.E. Comer, and D.L. Stevens, "Internetworking with TCP/IP: Vol. II: Design, Implementation, and Internals, BSD Socket Version," Prentice-Hall, Englewood Cliffs, N.J. 1991.
- [47] A. Shay William, "Understanding Data Communications and Networks," PWS Publishing Company, 1995.
- [48] R. Bell Mark, and Terrell Rob, "The Mac Web server book: tools & techniques for building your Internet site," Ventana

Communications Group, 1996.

- [49] Gardner Jim, and Linseman Anne, "Scott Nicol, and Chris Retterath: MKS LEX & YACC," 1992.
- [50] D. Thomas Michael, et al., "Java Programming for the Internet: A Guide to Creating Dynamic," Interactive Internet Application, Ventana Communications Group, 1996.
- [51] Danielle Bird, et al, "Special Edition Using Java," Que Corporation, 1996.
- [52] Aaron E. Walsh, "Foundations of Java Programming for the World Wide Web," IDG Books Worldwide, 1996.
- [53] D. Stewart, "A platform with six degrees of freedom, Proceedings Institution of Mechanical Engineers (Part - I)," Vol. 180, No. 15, 1965, pp. 371-386.
- [54] Z. Ji, "Workspace Analysis of Stewart Platforms via Vertex Space, Journal of Robotic System," Vol. 11, No. 7, 1994, pp. 631 - 639.
- [55] O. Masory and J. Wang, "Workspace evaluation of Stewart platforms, Advanced Robotics," Vol. 9, No. 4, 1995, pp. 443-461.
- [56] J. P. Merlet, "Determination of 6D workspace of Gough-Type parallel manipulator and comparison between different geometries," The Internal Journal of Robotics & Research, Vol. 18, No. 9, 1999, pp. 902-916.
- [57] J. P. Merlet, "Guaranteed in-the-workspace improved trajectory/ surface/ volume verification for parallel robots," Proceedings of the IEEE International Conference on Robotics and Automation, PVP-Vol. 4, 2004, pp. 4103 - 4108.
- [58] C. Gosselin, "Determination of the Workspace of 6-DOF Parallel Manipulator," Journal of Mechanical Design, PVP-Vol. 112, 1990,



pp. 331-336.

- [59] M. Honegger, Codourey, and E. Burdet, "Adaptive control of the Hexaglide, a 6 DOF parallel manipulator," Proceedings of IEEE International Conference on Robotics and Automation, Albuquerque, 1997, pp. 543-548.
- [60] M. Suzuki, K. T. S. Watanabe, T. Tooyama, and K. Hattori, "Development of milling machine with parallel mechanism," Toyota Technical Review, Vol. 47, No. 1, 1997, pp. 125-130.
- [61] G. Pritschow, and K. H. Wurst, "Systematic design of Hexapods and parallel link systems," Annals of the CIRP, Vol. 46, No. 1, 1997, pp. 291-295.
- [62] B. R. Hopkins, and R. L. Williams II, , "Kinematics Design and Control of 6-PSU Platform," Industrial Robot: An International Journal, Vol. 29, No. 5, 2002, pp. 443-451.
- [63] Y. Wang, W. S. Newman, and R. Stoughton, "Workspace Analysis of the ParaDex Robot - A Novel, Close-Chain, Kinematically – Redundant Manipulator," IEEE International Conference on Rob. & Automation, 2000, pp. 2392-2397.
- [64] I. A. Bonev, and J. Ryu, "Workspace Analysis of 6-PRRS Parallel Manipulators Based on the Vertex Space Concept," ASME Design Technical Conferences, DETC99/DAC-8647, Las Vegas, NV, September 12-15, 1999.
- [65] J. P. Kim, and J. Ryu, "Closed-Form Dynamics Equations of 6-DOF PUS Type Parallel Manipulators," ASME Design Technical Conferences, 26<sup>th</sup> Biennial Mechanisms Conference, Baltimore, MD, September 10-13, 2000.
- [66] B. R. Hopkins, and R. L. Williams II, "MODIFIED 6-PSU



- PLATFORM,” ASME Design Engineering Technical Conferences, September 29 - October 2, Montreal, Canada, 2002.
- [67] K. A. B. Rao, P. V. M. Rao, and S. K. Saha, “Workspace and Dexterity Analysis of Machine Tools,” Proceeding of the IEEE International Conference on Robots & Automation, Taipei, Taiwan, September 14-19, 2003.
- [68] S. Stoughton Robert, and T. Tatsuo Arai, “A modified Stewart platform manipulator with improved dexterity,” Robotics and Automation, IEEE Transactions on, Vol. 9, No. 2, 1993, pp. 166-173.
- [69] C. A. Klein, and B. E. Blaho, “Dexterity measures for the design and control of kinematically redundant manipulators,” Int. J. Robot. Res., Vol. 6, No. 2, 1987, pp. 72-78.
- [70] H. Pittens, and R. P. Podhorodeski, “A family of Stewart platforms with optimal dexterity,” J. Robot. Syst., Vol. 10, No. 4, 1993, pp. 463-479.
- [71] D. Giovannetti, and M. Blum, “Design of a Hexapod Motion Cueing System for the NASA Ames Vertical Motion Simulator,” AIAA Modeling and Simulation Technologies Conference and Exhibit 5-8 August, Monterey, California, 2002.
- [72] W. E. Lorensen, and H. E. Cline, “Marching Cubes: A High Resolution 3D Surface Construction Algorithm,” Computer Graphics (Proceedings of SIGGRAPH '87), Vol. 21, No. 4, 1987, pp. 163-169.
- [73] G. M. Nielson, and B. Hamann, “The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes,” Proc. IEEE Visualization, 1992, pp. 83-91.

- [74] S.V. Matveyev, "Resolving the topological ambiguity in approximating the isosurface of scalar function," Visualization and Machine Vision, Proceedings, IEEE Workshop on, June, 1994, pp. 18-21.
- [75] S. Matveyev, "Approximation of Isosurface in the Marching Cube: Ambiguity Problem," Proc. IEEE Visualization, 1994, pp. 288-292.
- [76] B. Natarajan, "On Generating Topologically Consistent Isosurfaces from Uniform Samples," The Visual Computer, PVP-Vol. 11, 1994, pp. 52-62.
- [77] E. Chernyaev, "Marching Cubes 33: Construction of Topologically Correct Isosurfaces," Technical Report CN/ 95-17, CERN, 1995, <http://wwwinfo.cern.ch/asdoc/psdir/mc.ps.gz>.



**Figures**

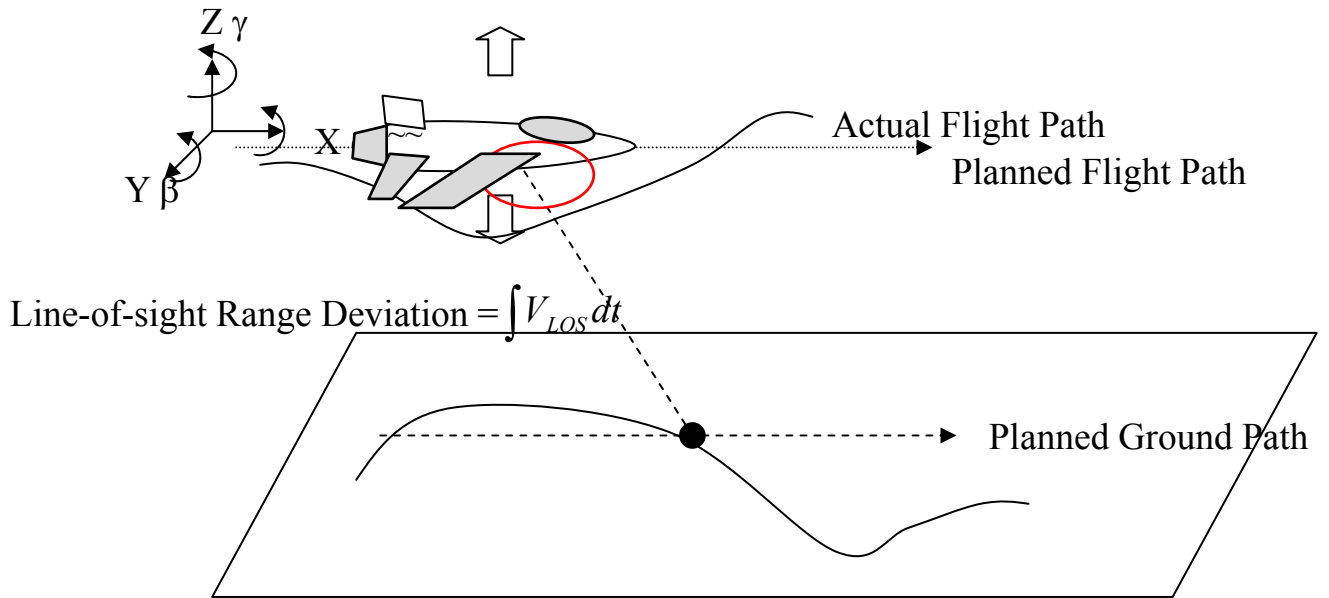


Figure 1. Vehicle and Sensor relation.

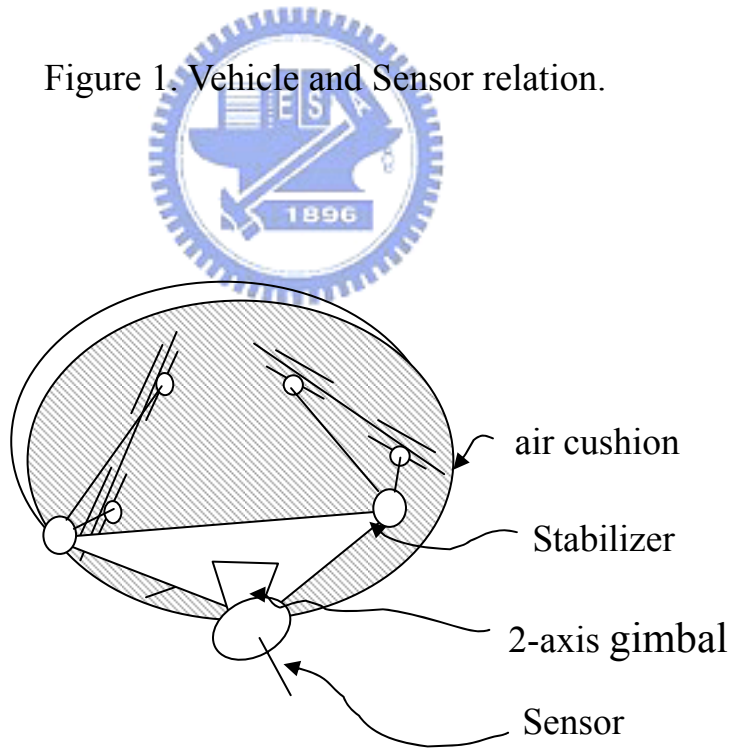


Figure 2. Sensor, Gimbal, stabilizer, air cushion.

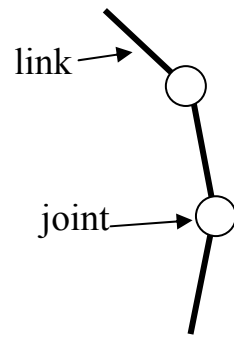


Figure 3. The limb

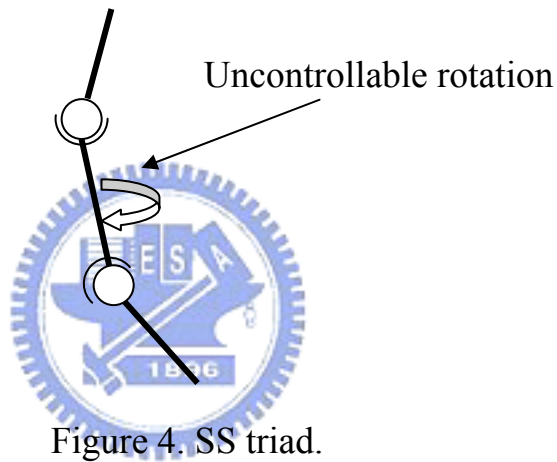


Figure 4. SS triad.

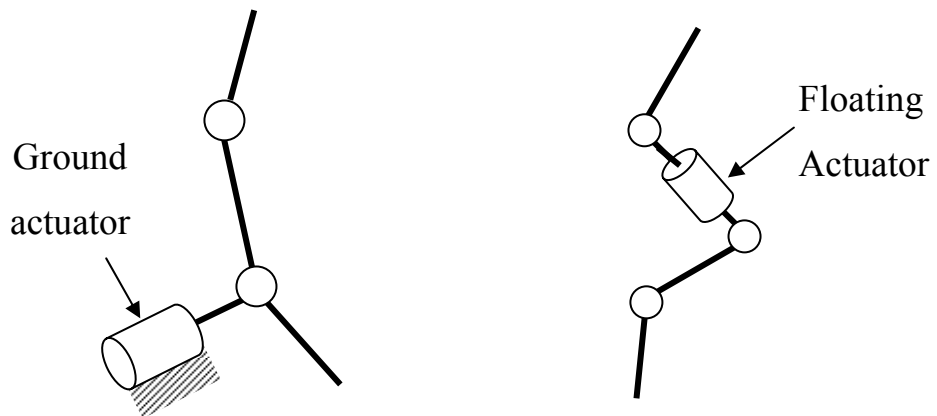


Figure 5. Type of actuators.

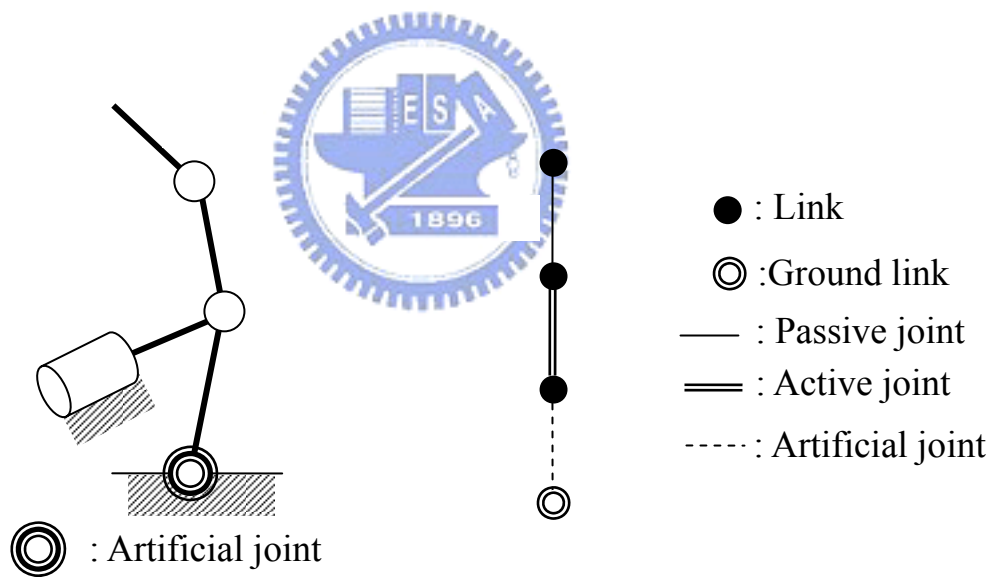


Figure 6. (a) Kinematic structure (b) corresponding graph.

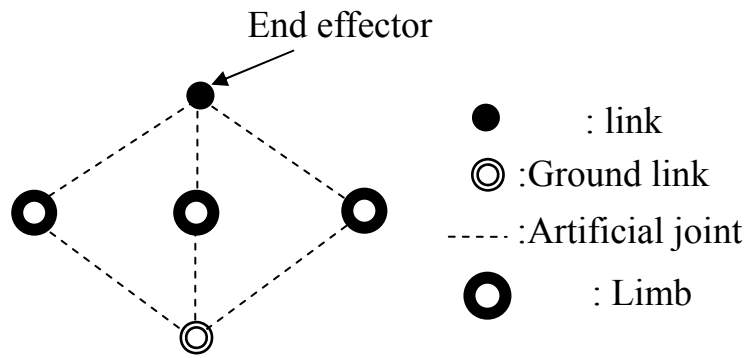


Figure 7. Reduced graph of parallel manipulation.

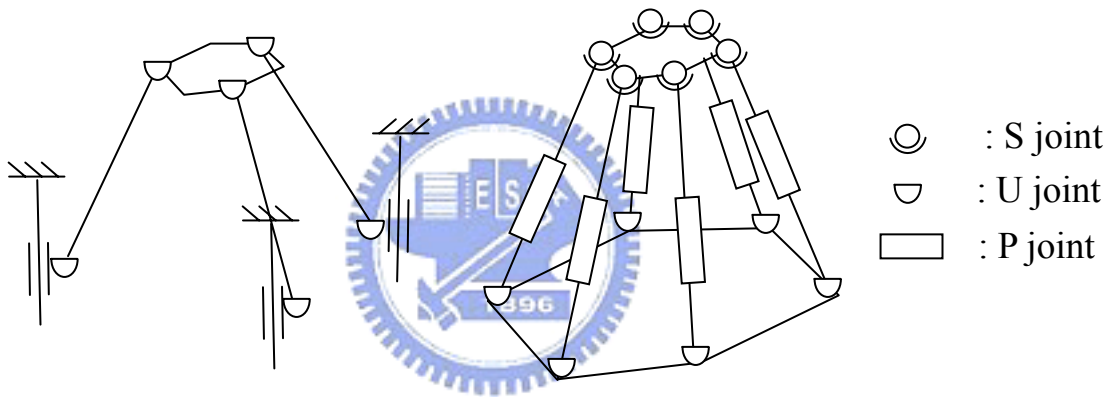


Figure 8. (a) Tripod-based PKM (b) Stewart platform (Hexapod).

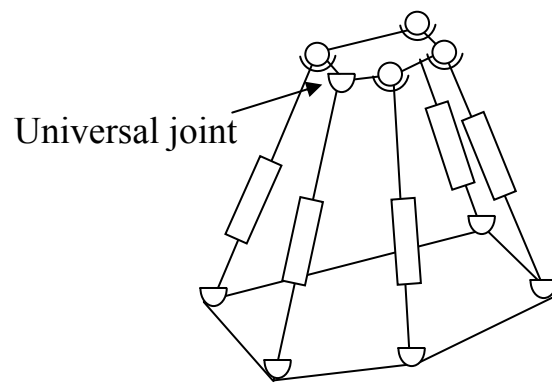


Figure 9. Example of a 5-dof SSPM.

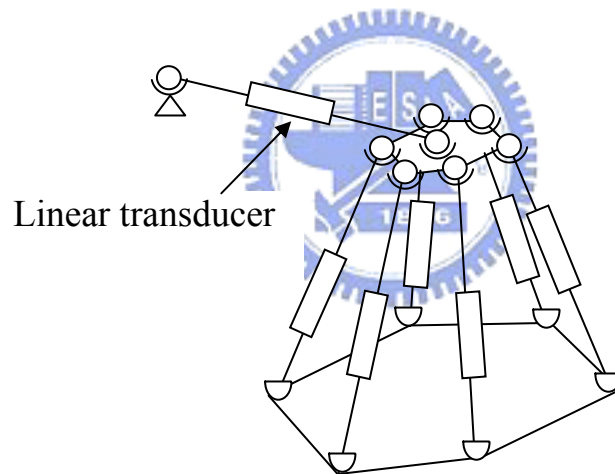


Figure 10. DDB measurement as Saturated limb.

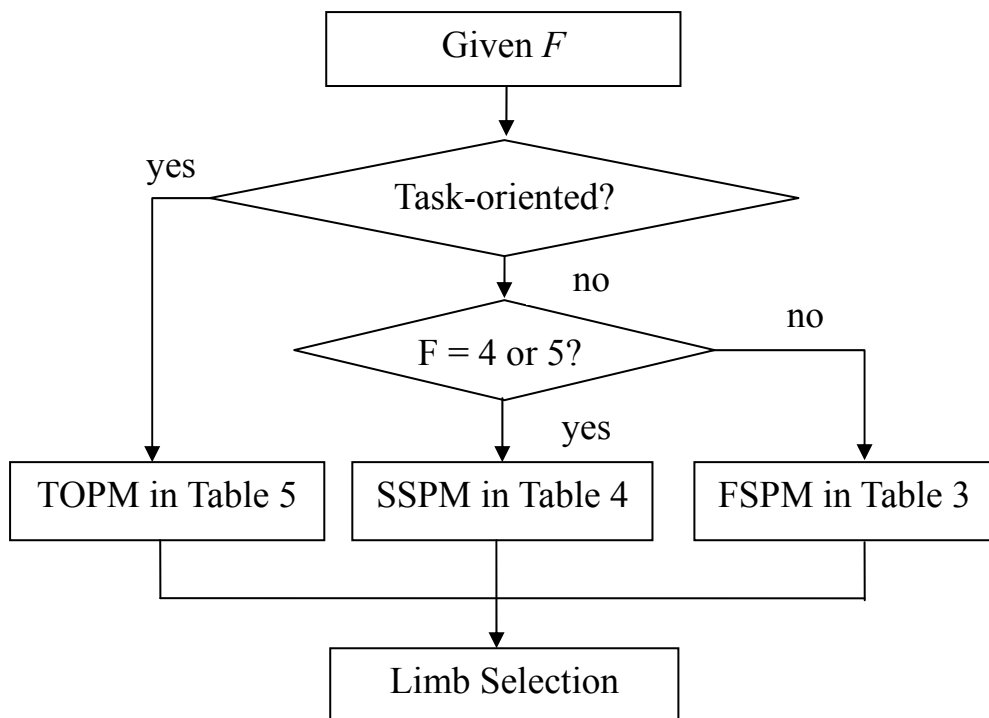


Figure 11. Determination of parallel manipulator type.





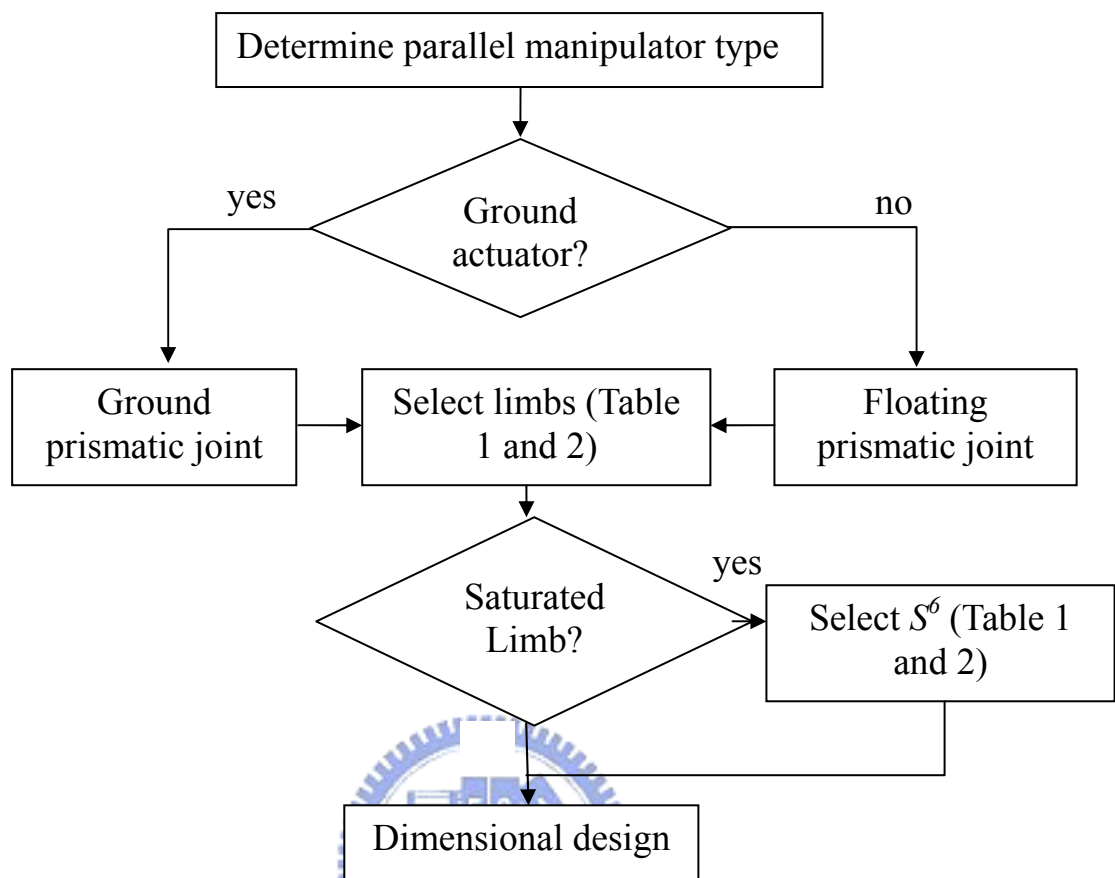


Figure 12. Limb Selection.

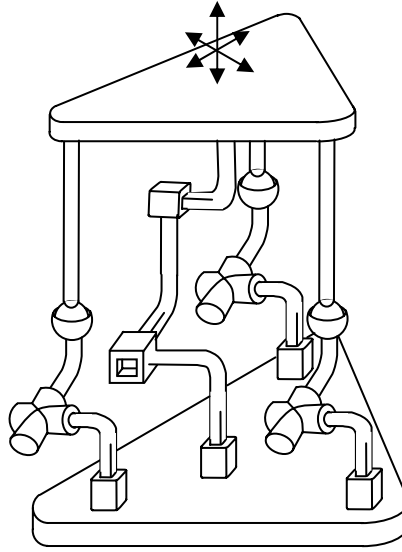
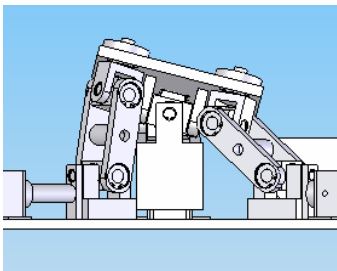
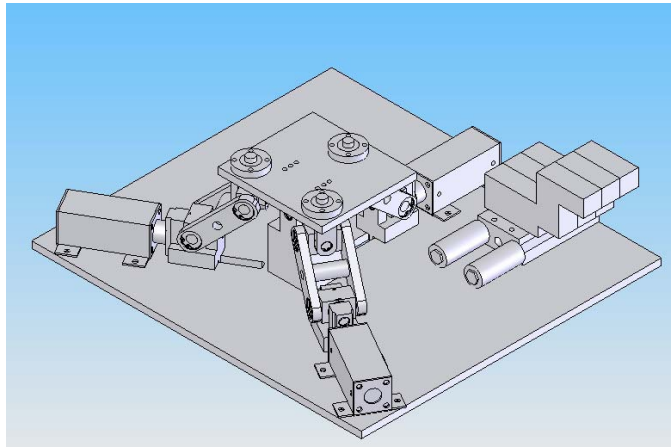
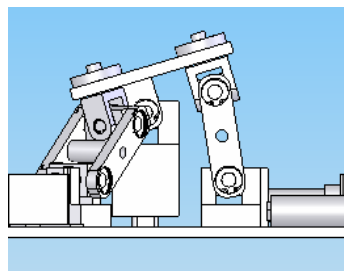


Figure 13. Cartesian machine.

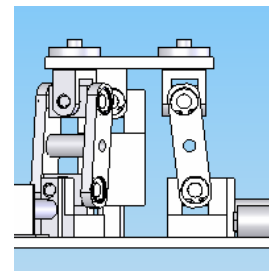




Rol



Pitch



Z

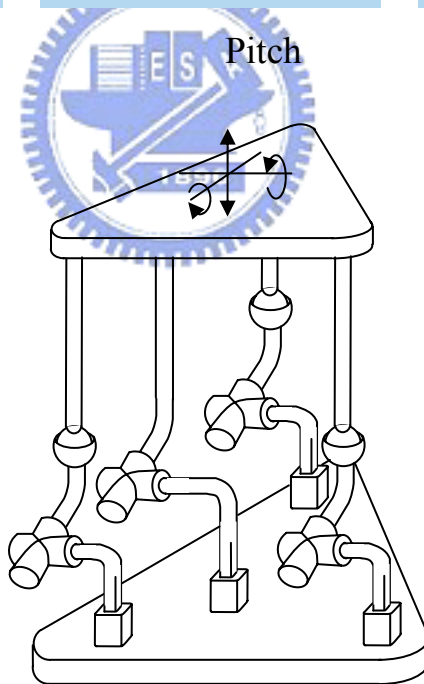


Figure 14. Wobble machine.

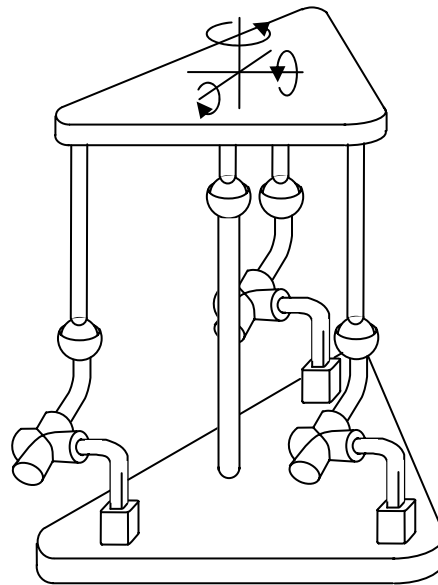


Figure 15. Rotation machine.

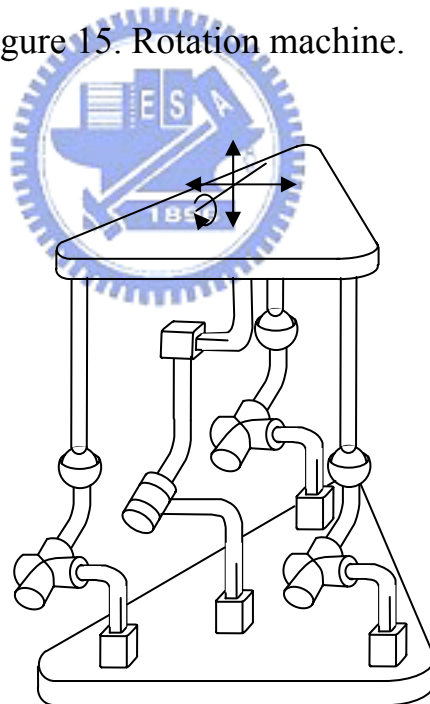


Figure 16. Cobra-head machine.

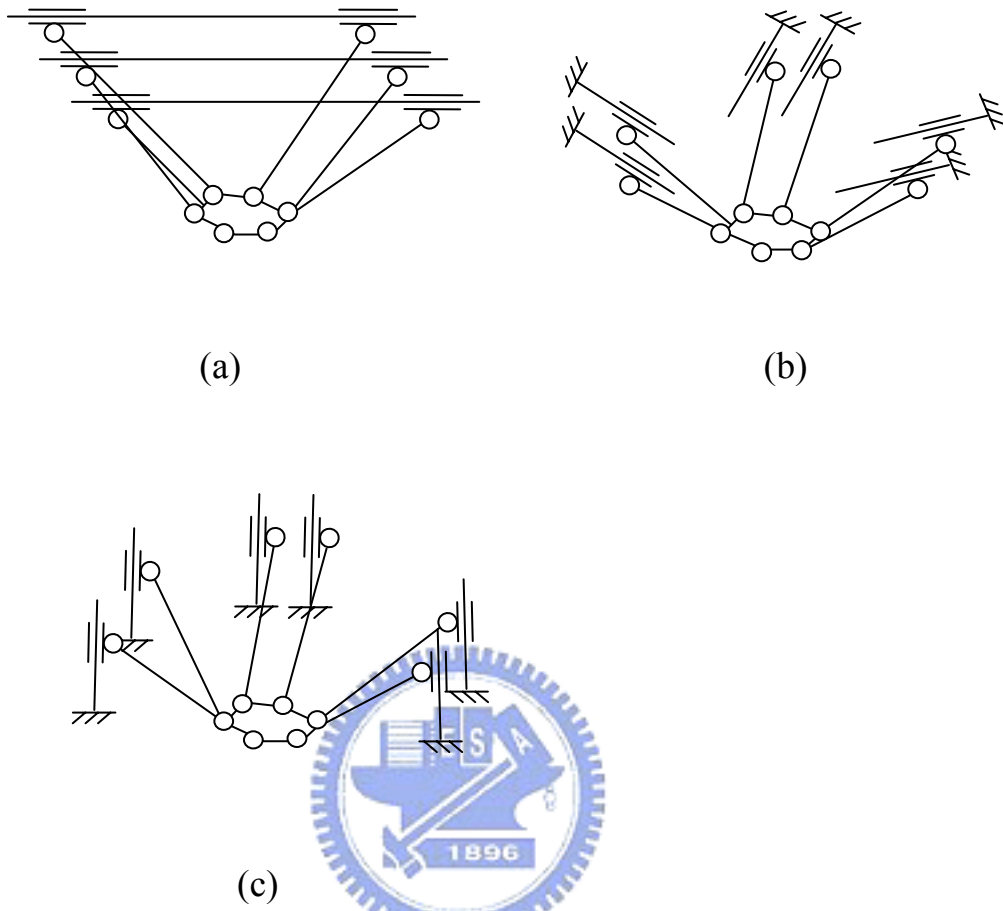


Figure 17 Different types of Hexaslide platform: (a) Hexaglide, (b) HexaM, and (c) Linapod.

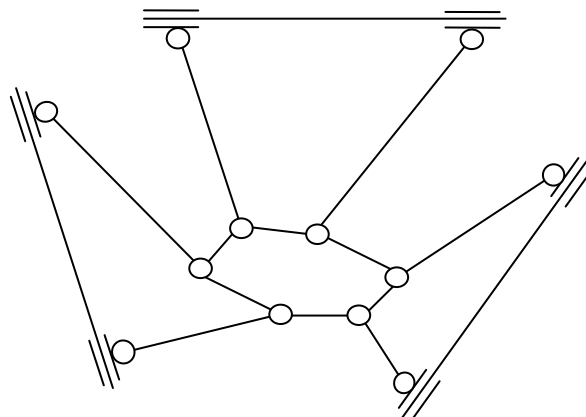
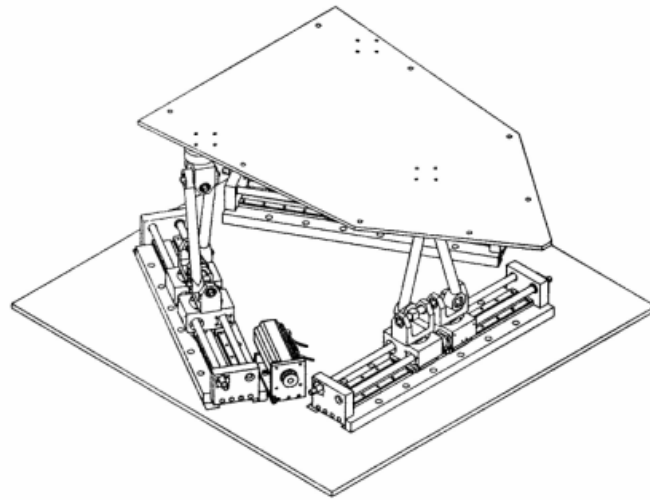


Figure 18 Delta Hexaglide platform.



(a)



(b)

Figure 19 (a) Kinematic structure and (b) photograph of the Delta Hexaglide platform mechanism (Courtesy: IMON Inc.).

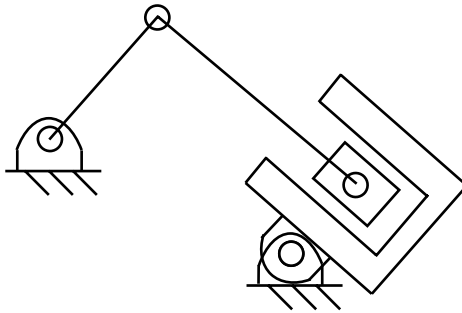


Figure 20. Oscillating-cylinder engine mechanism.

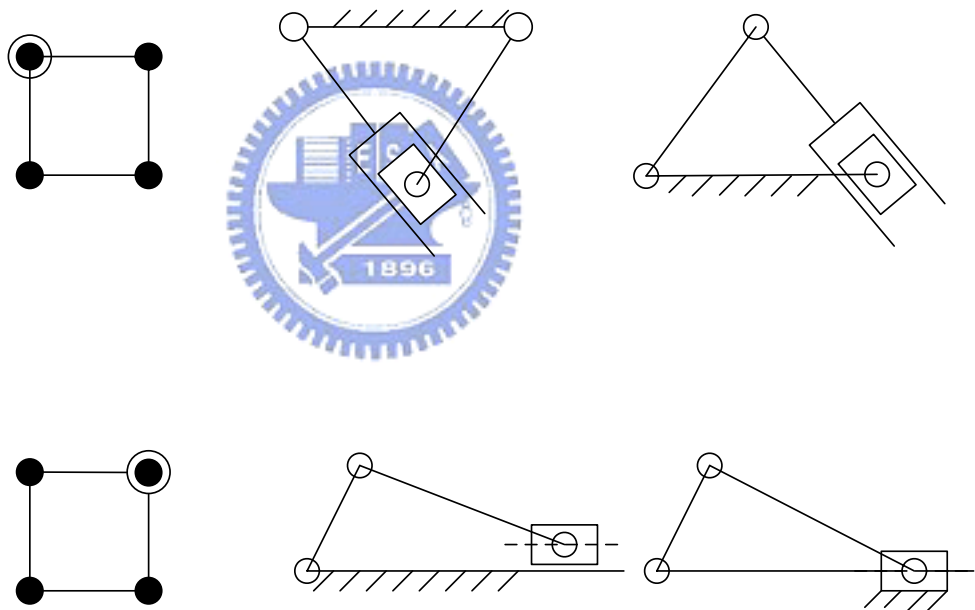


Figure 21. RRRP kinematics inversion.

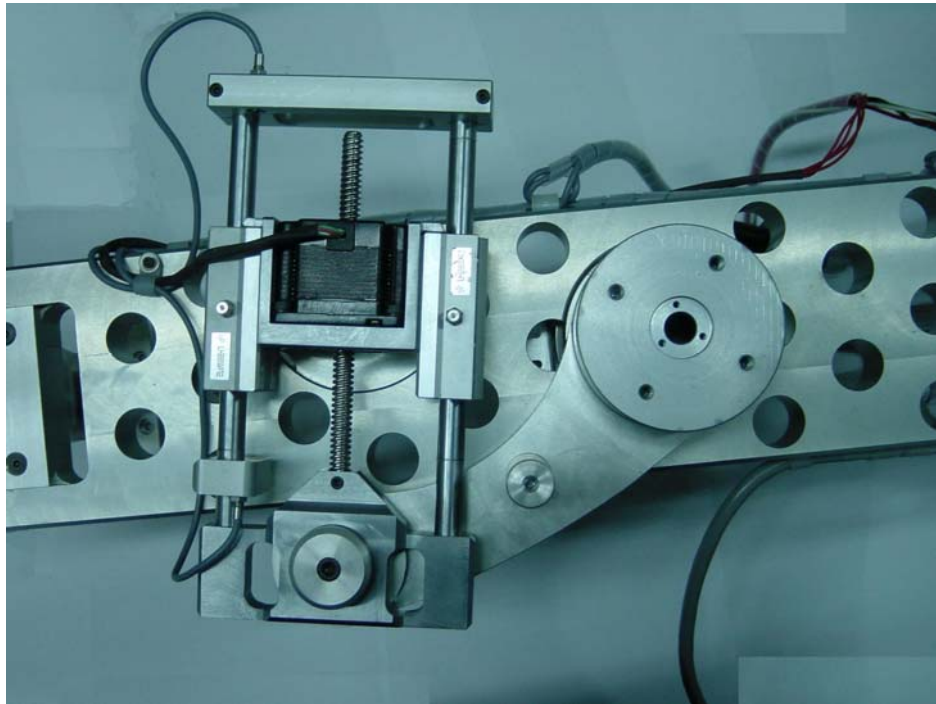


Figure 22a. The real sensor pedestal picture.

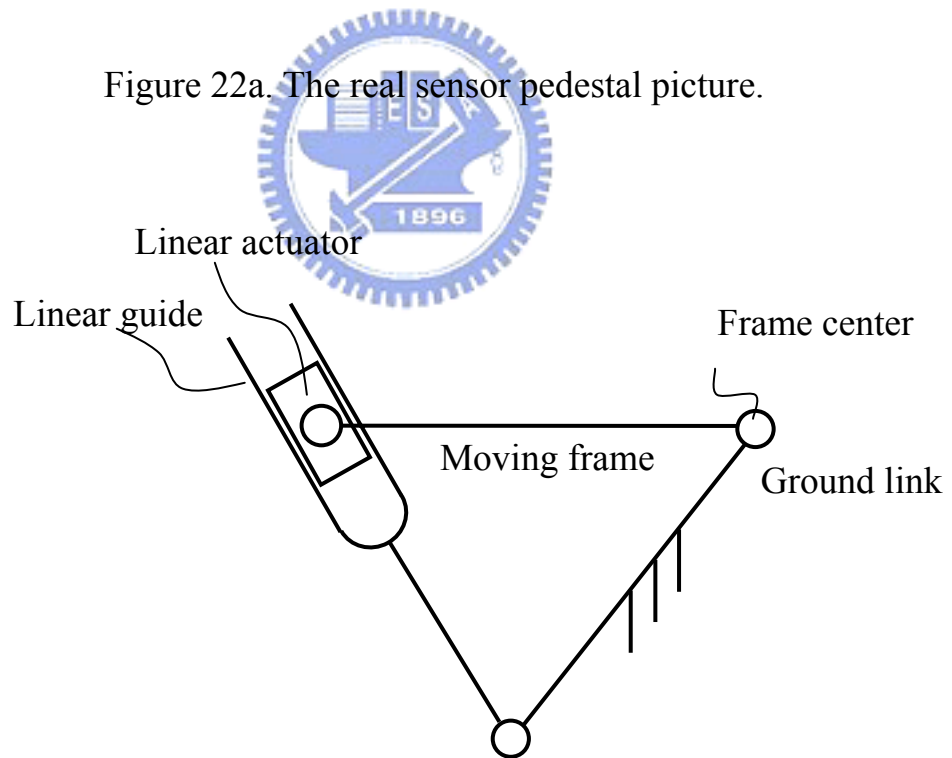


Figure 22b. Kinematic structure of turning-block mechanism.



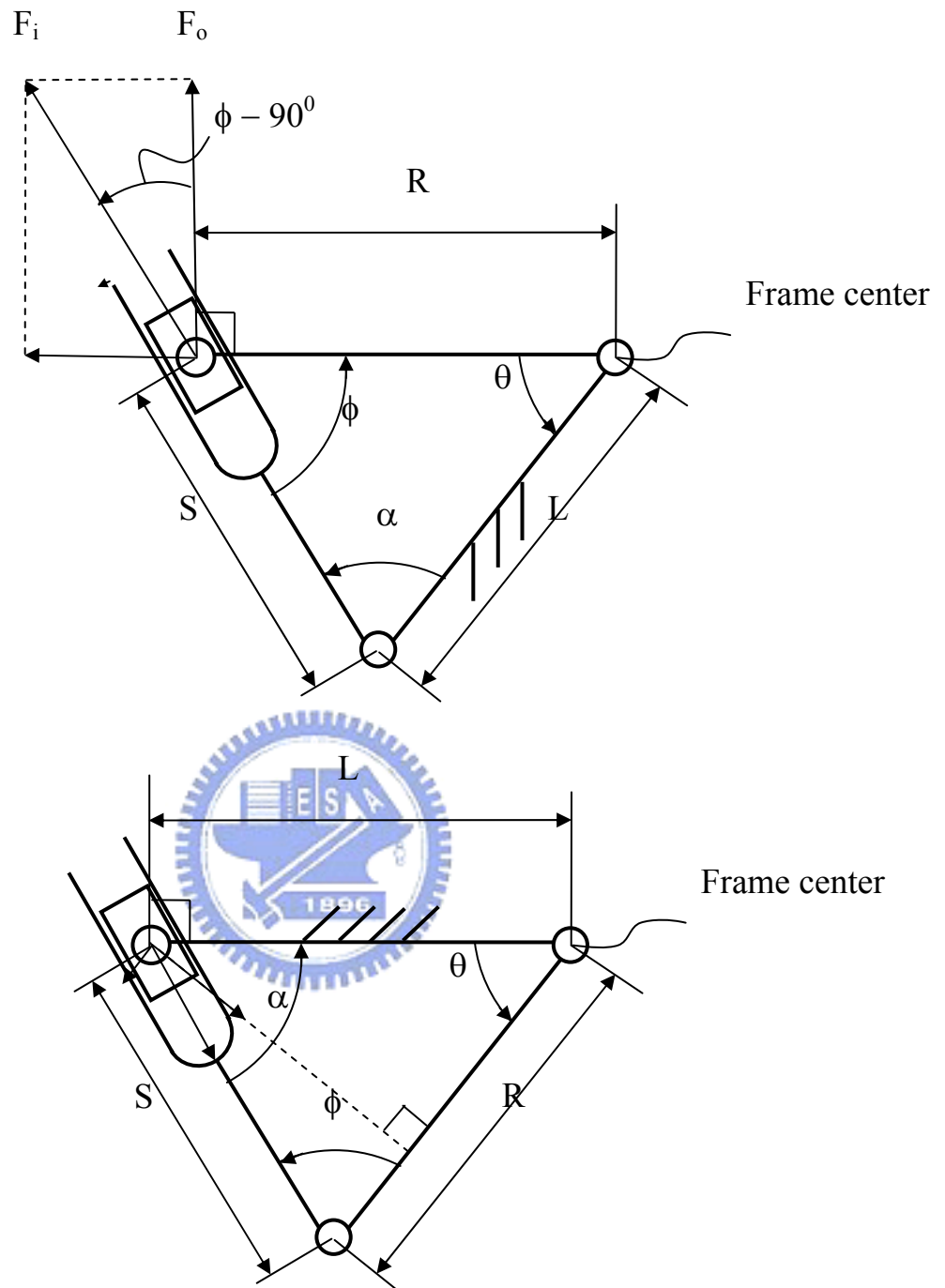


Figure 23. Force transmitted and kinematic structure of (a) Turning-block and (b) Swinging-block mechanism.

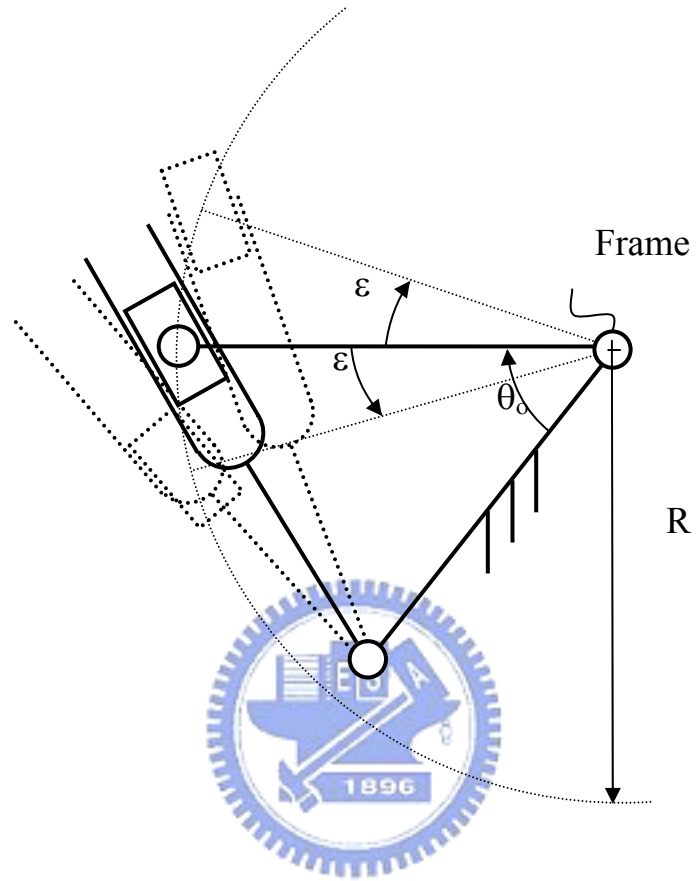


Figure 24. Dimensional design for turning block mechanism.

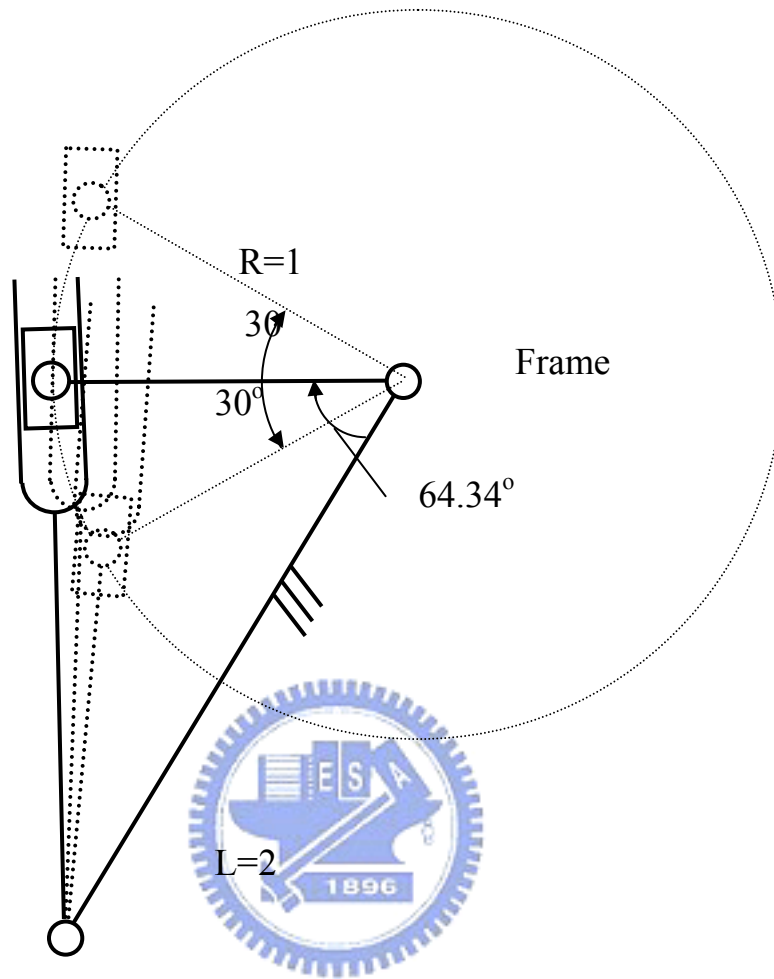


Figure 25. Example of turning-block illustrated configuration.

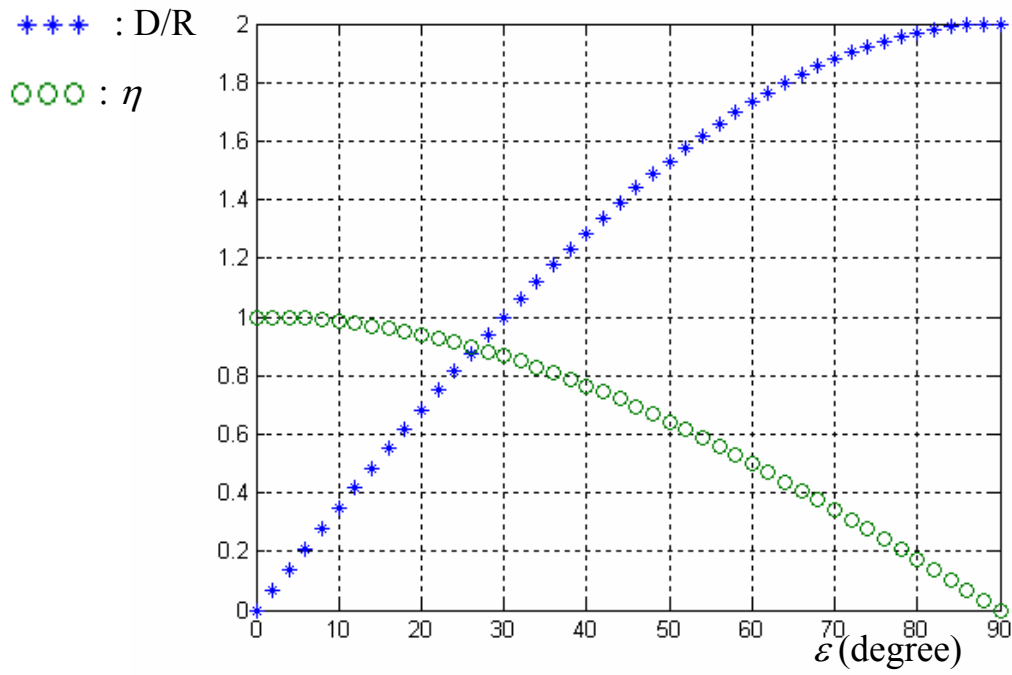


Figure 26.  $\varepsilon$  versus  $D/R$  and  $\varepsilon$  versus  $\eta$

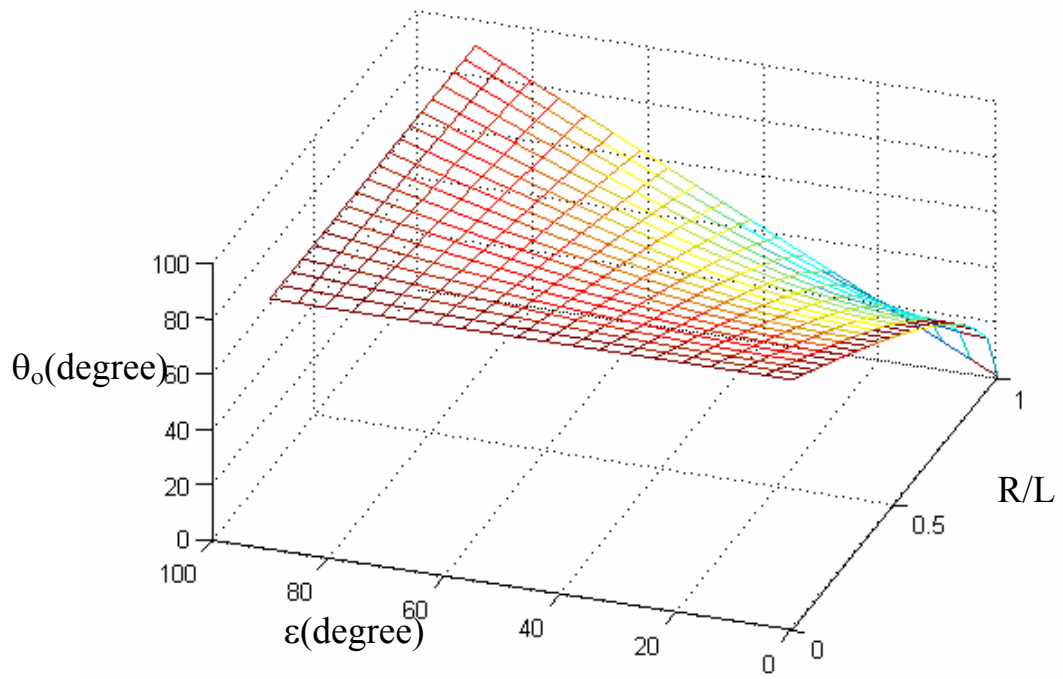


Figure 27.  $\varepsilon$  versus  $R/L$  versus  $\theta_o$ .

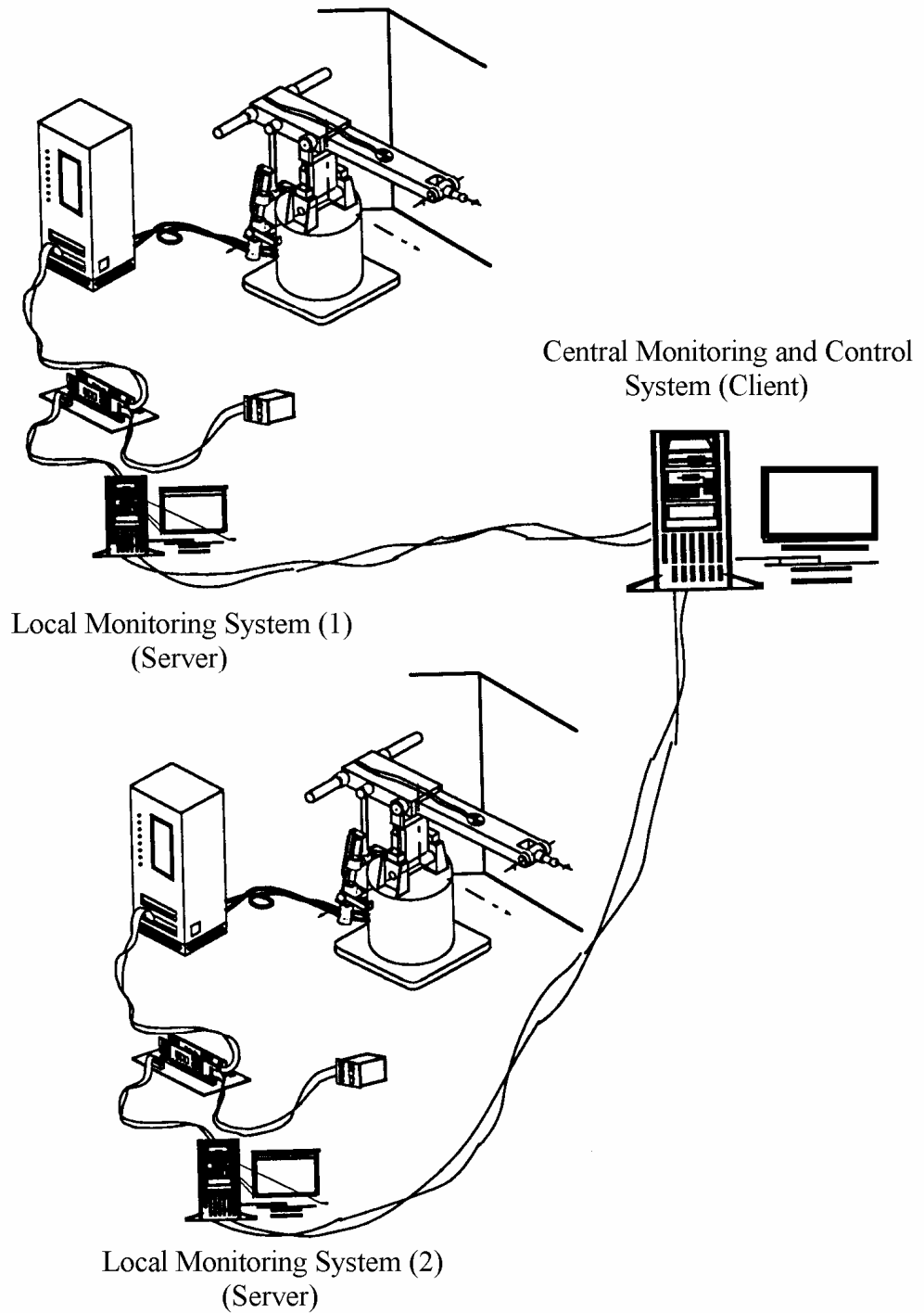


Figure 28. The active multi-robotic safeguard system architecture.

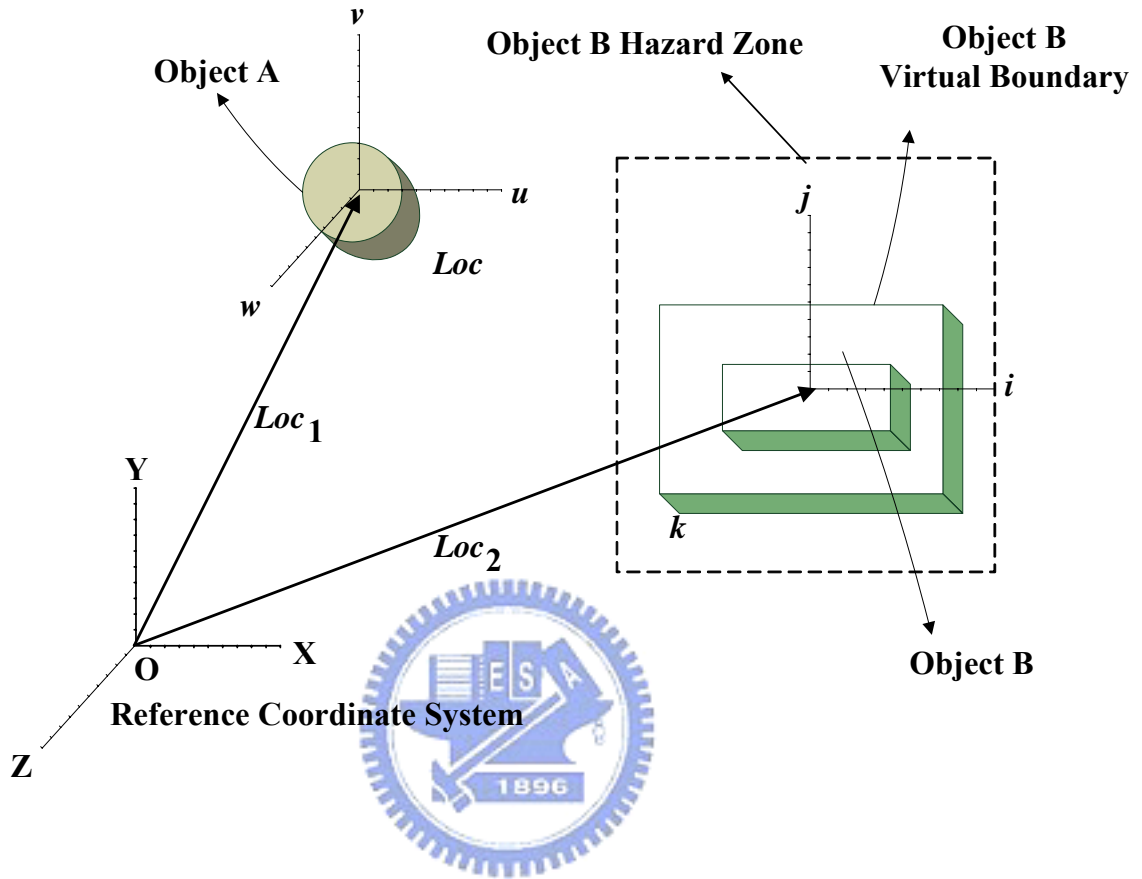


Figure 29. The coordinate system for the robotic collision detection.

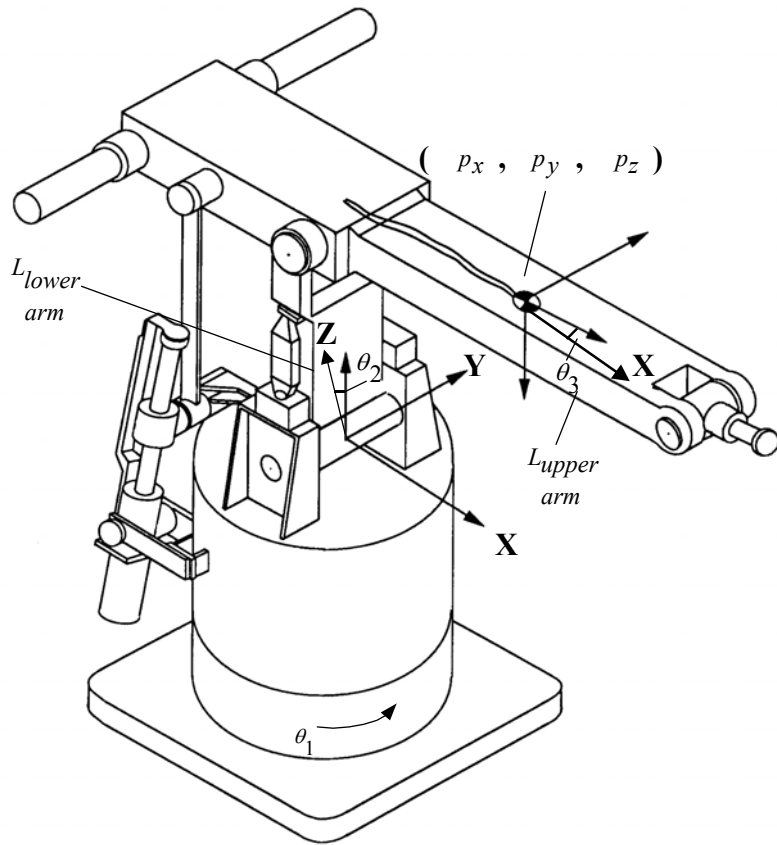


Figure 30. The industrial U-type robot.

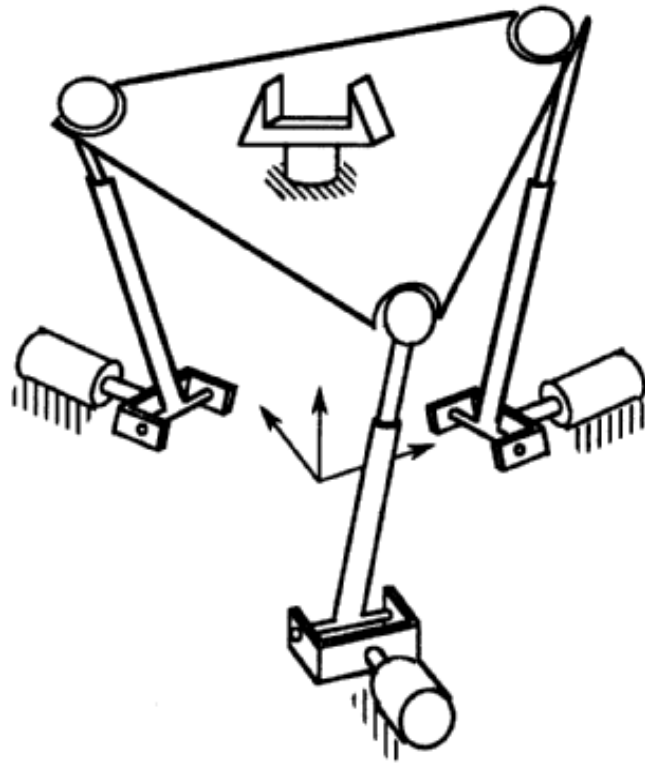


Figure 31. The Parallel-linked robot system.



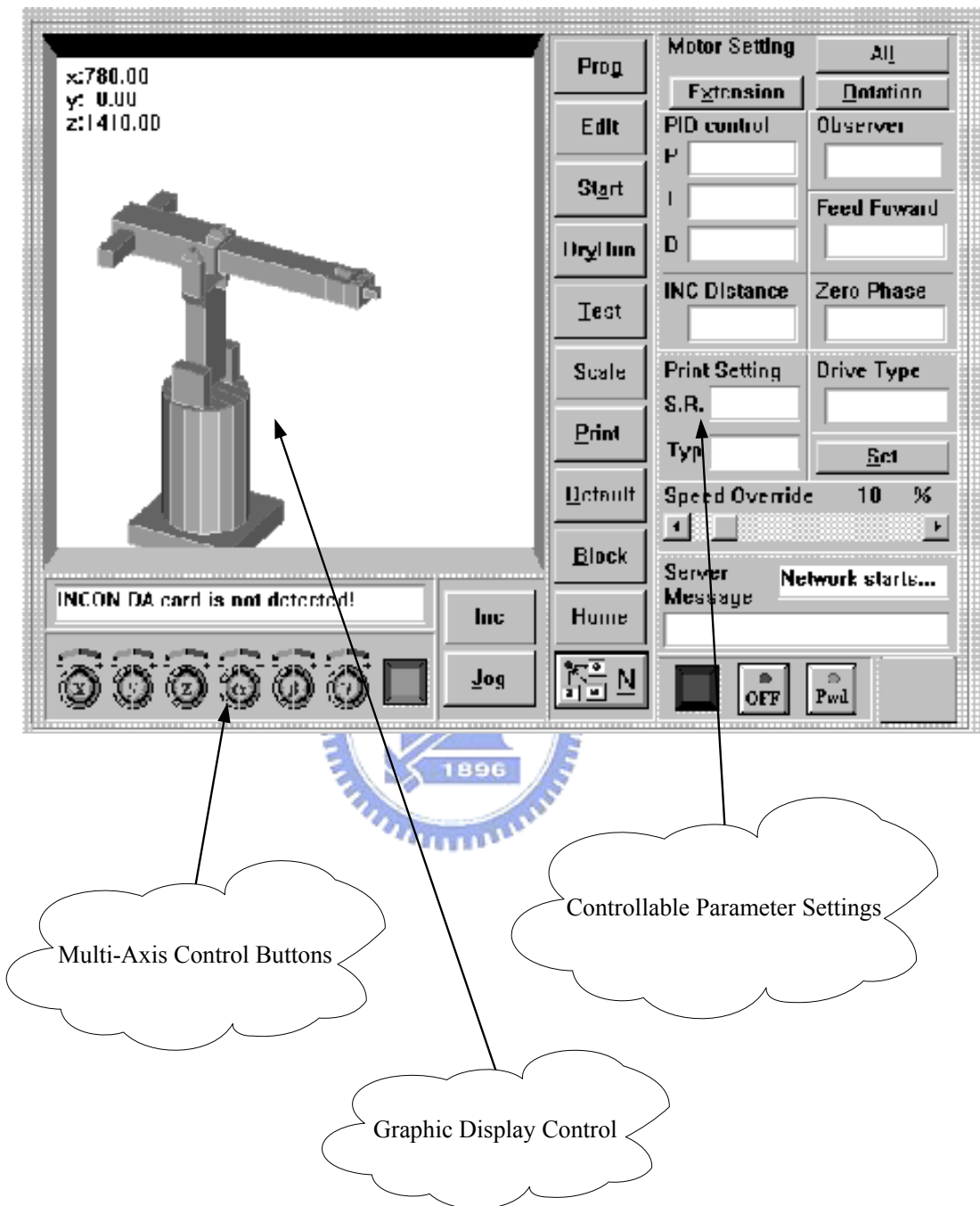


Figure 32. The interface program for the industrial robot.

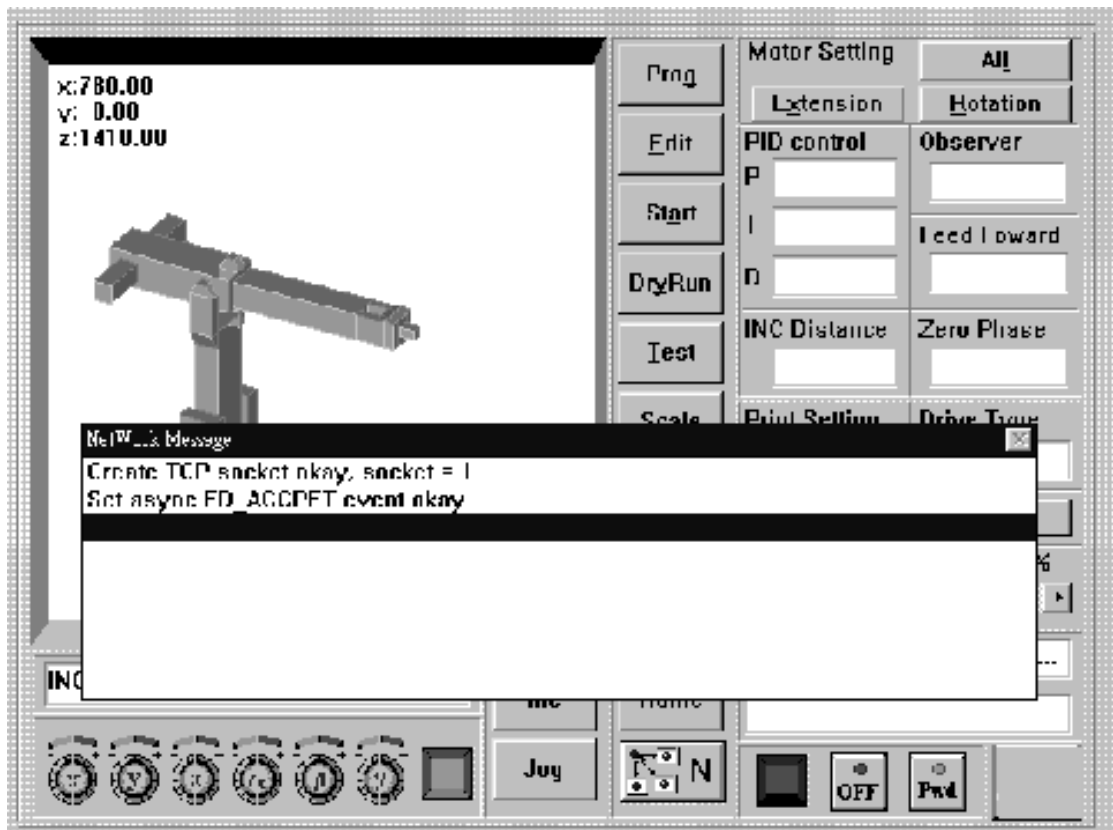
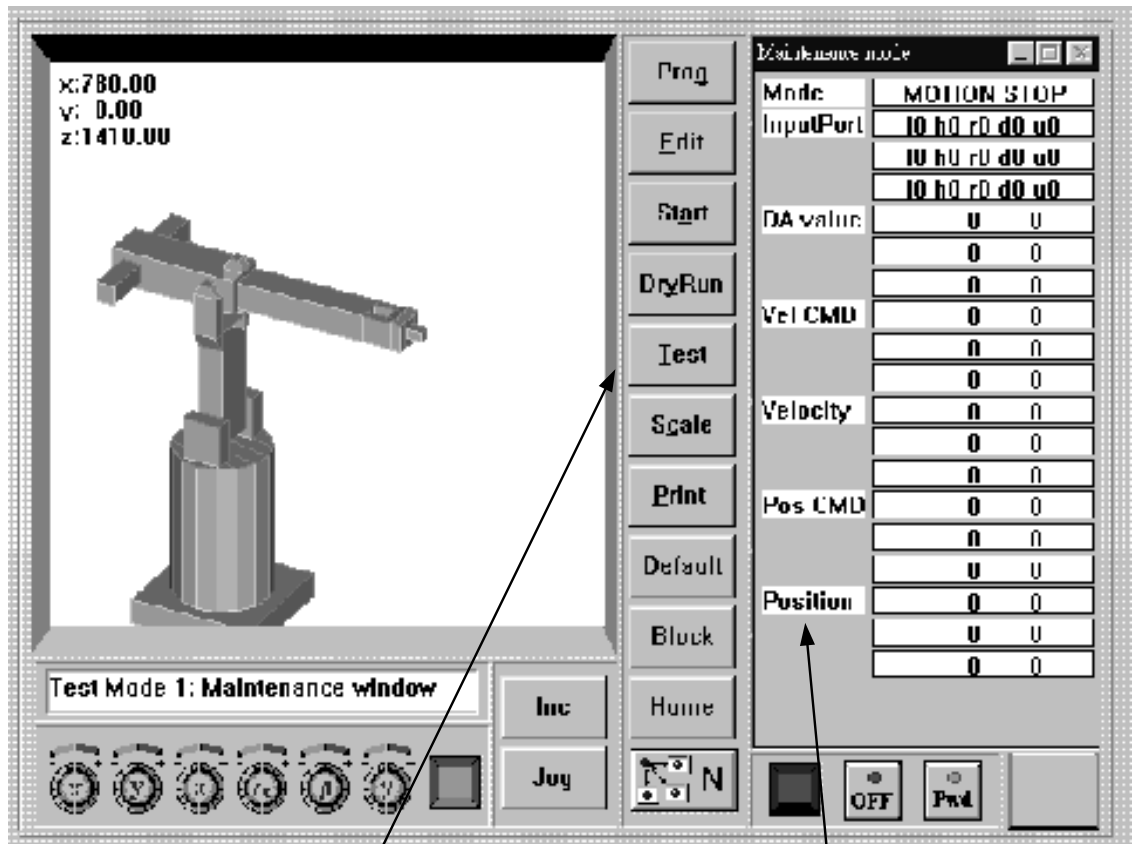


Figure 33. The networking for the industrial robot controller interface.



Test Button

Control Status Display

Figure 34. The industrial robot control interface display.

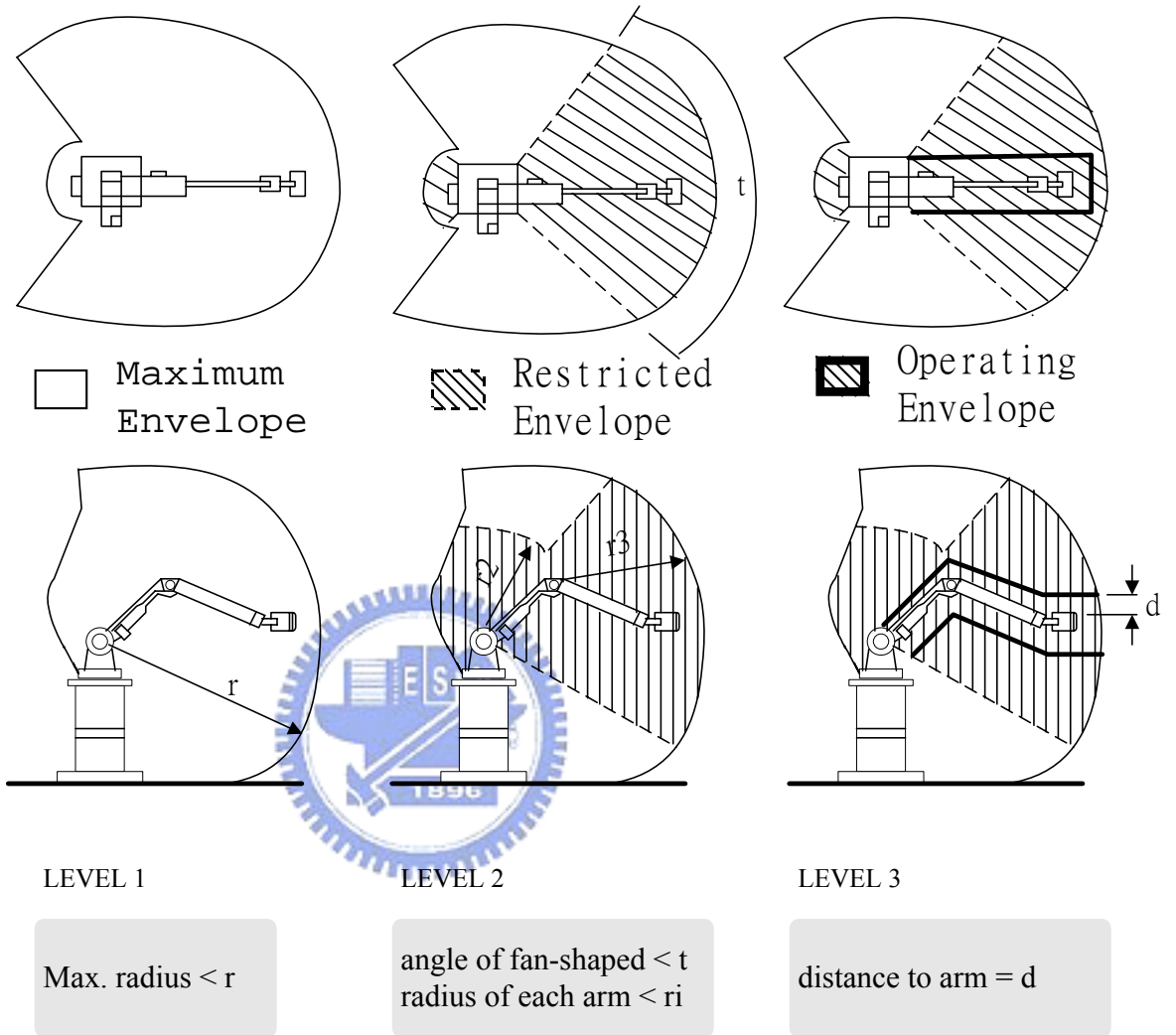


Figure 35. The levels of robot safe range.

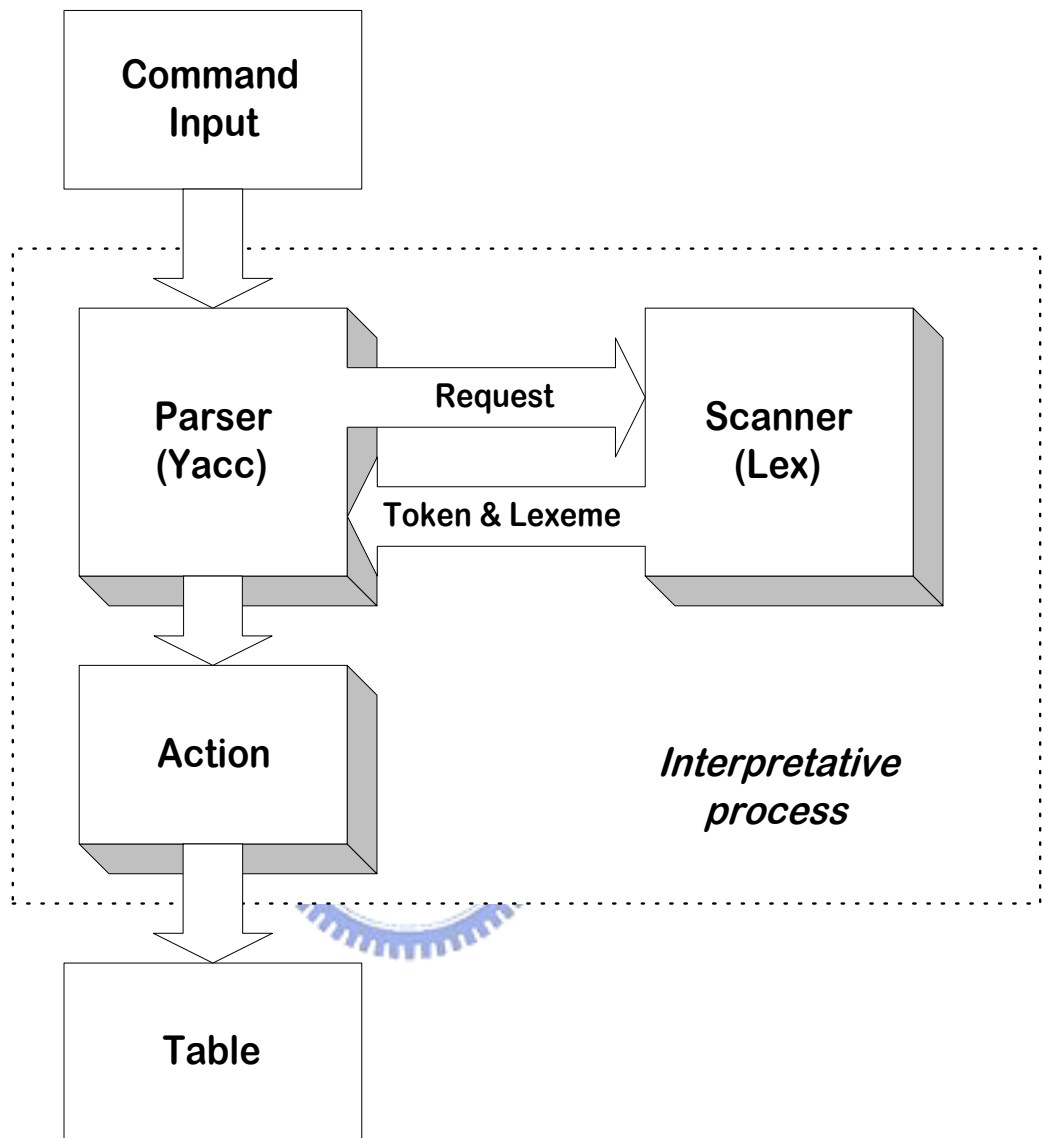
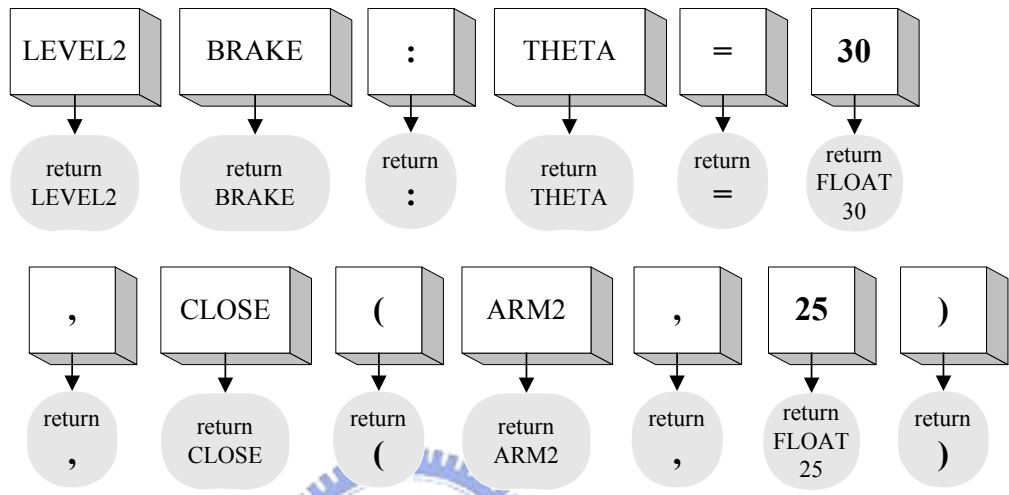


Figure 36. Interpretative process of LEX & YACC.

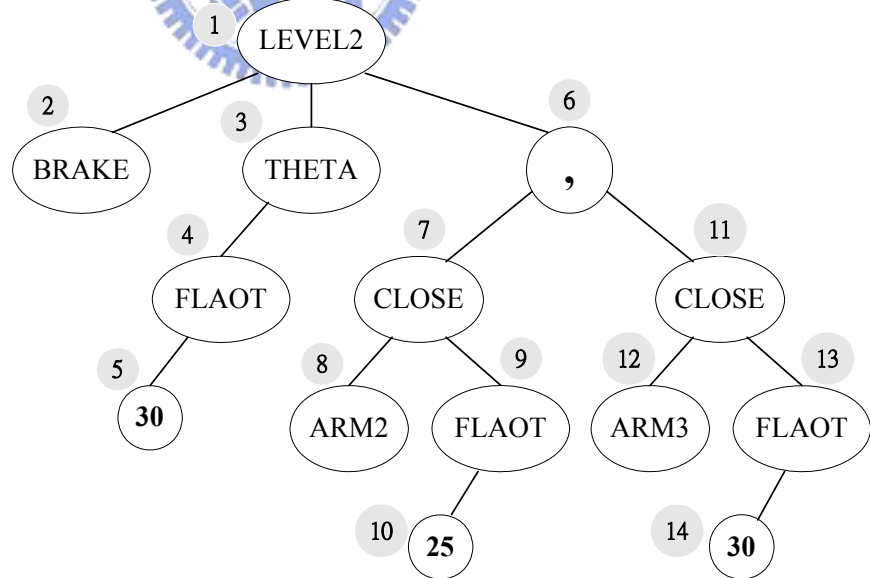
**Input command :**

**LEVLE2 BRAKE: THETA=30, CLOSE(ARM2, 25)  
,CLOSE(ARM3, 30) ;**

**Step1(Translation) :**



**Step2(Parser) :**



**Step3(Action) :**



Figure 37. The interpretative process of setting safe-range.

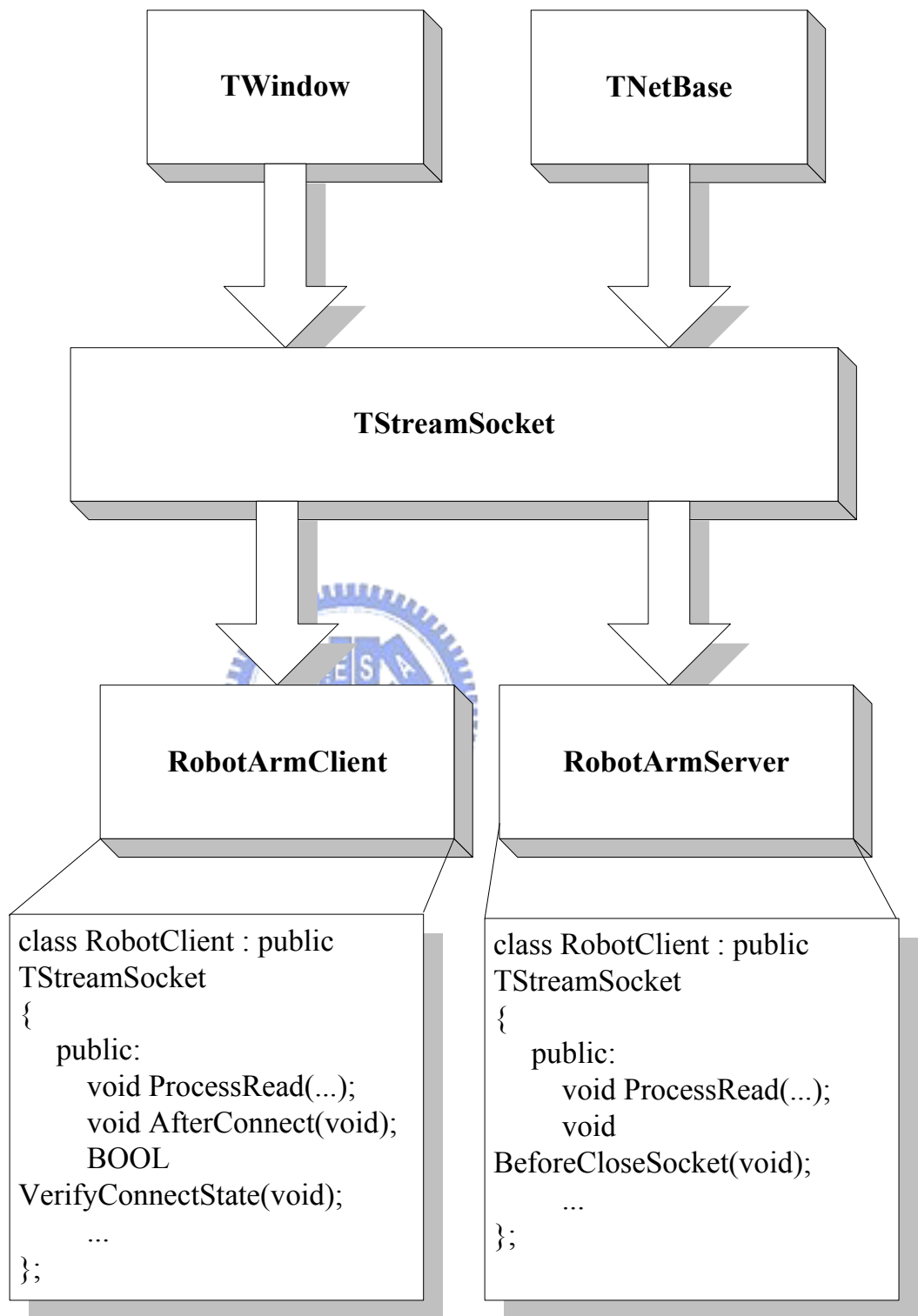


Figure 38. Programming construction of NETCOMM.

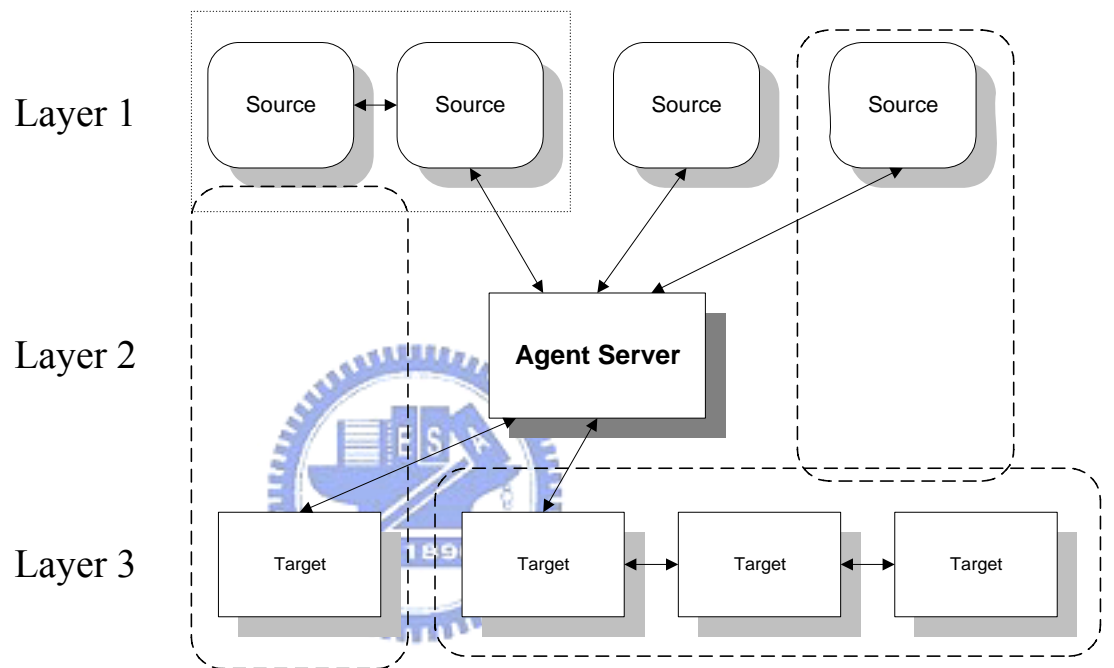
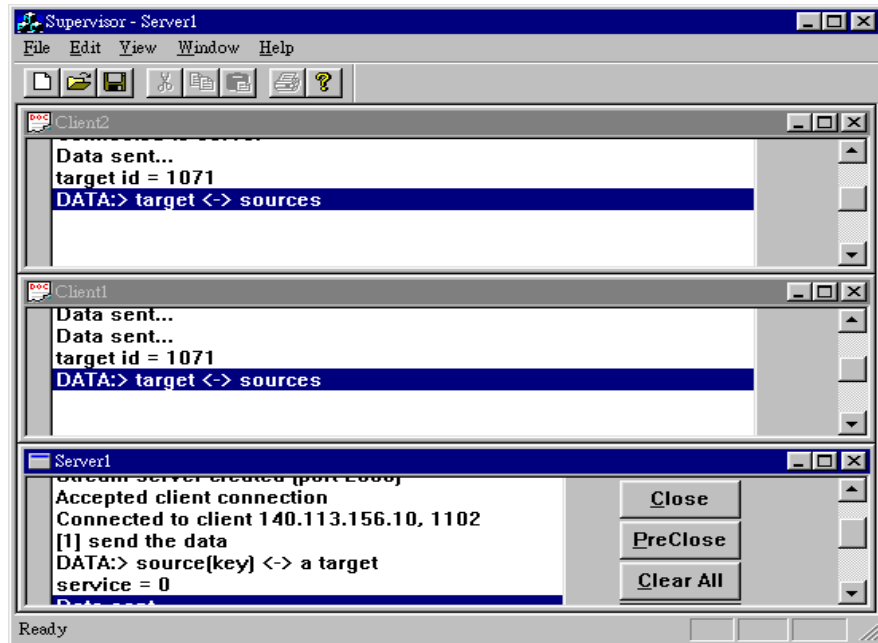
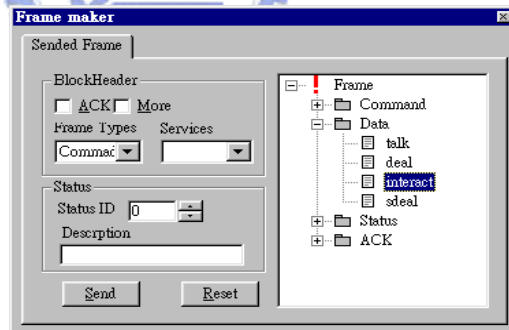


Figure 39. Three-Layer architecture (Service Model).





(a)



(b)

Figure 40. (a) The tester application program.  
 (b) The frame maker window of tester.

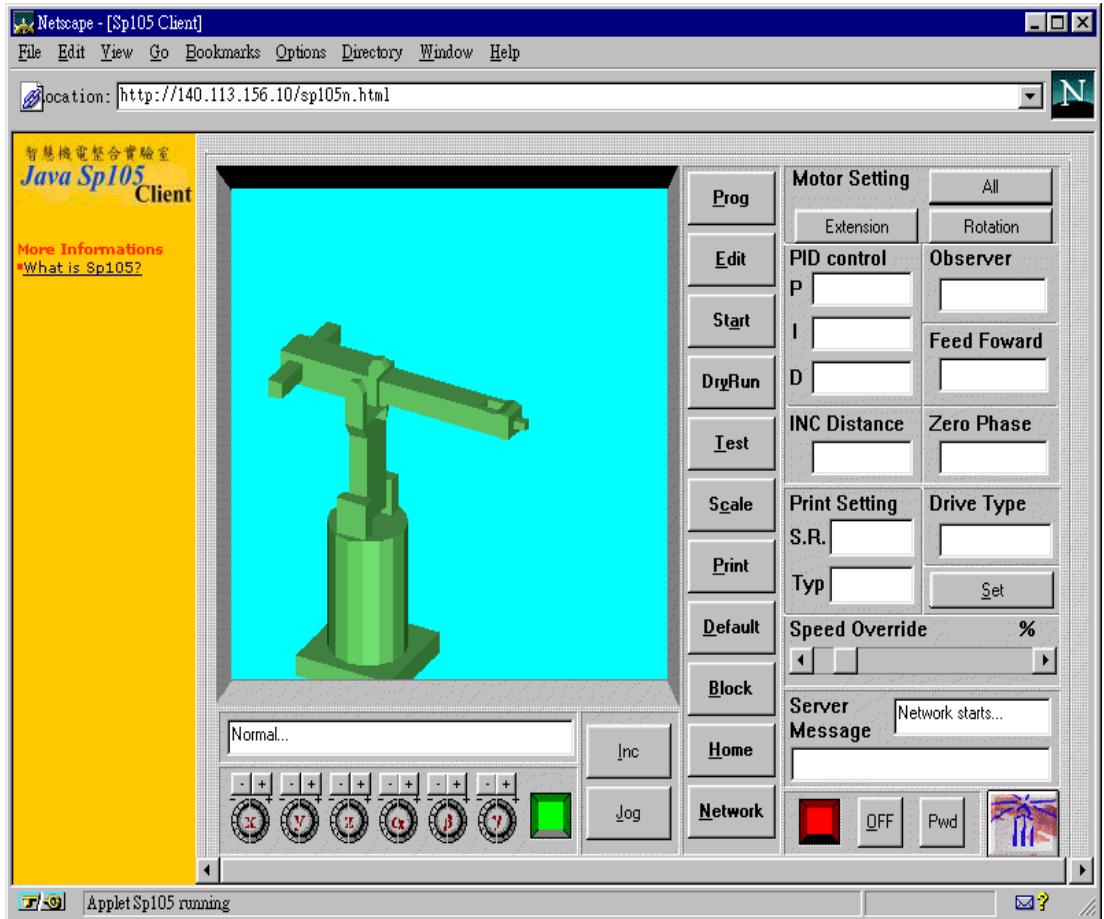


Figure 41. Java's applet of robot in Case One.

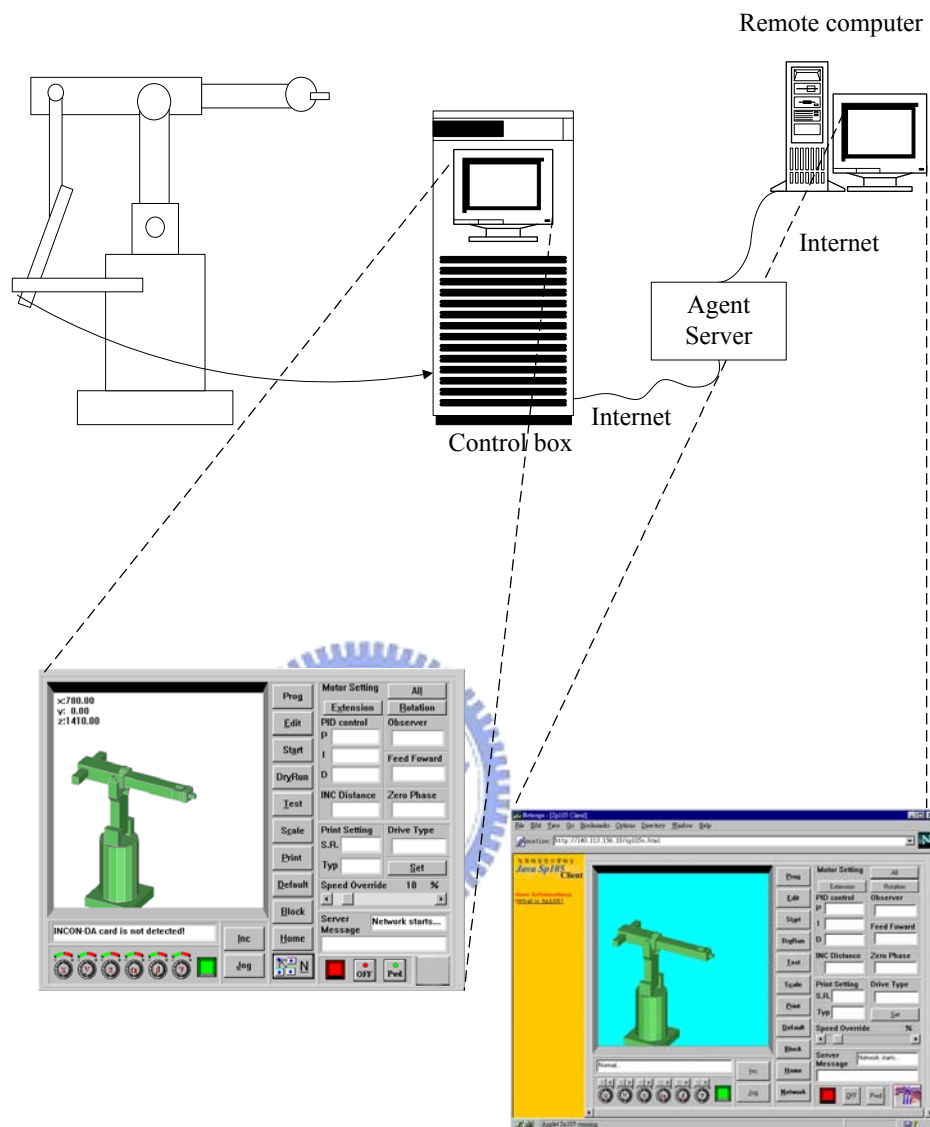


Figure 42. Architecture of robot supervision (Case One).

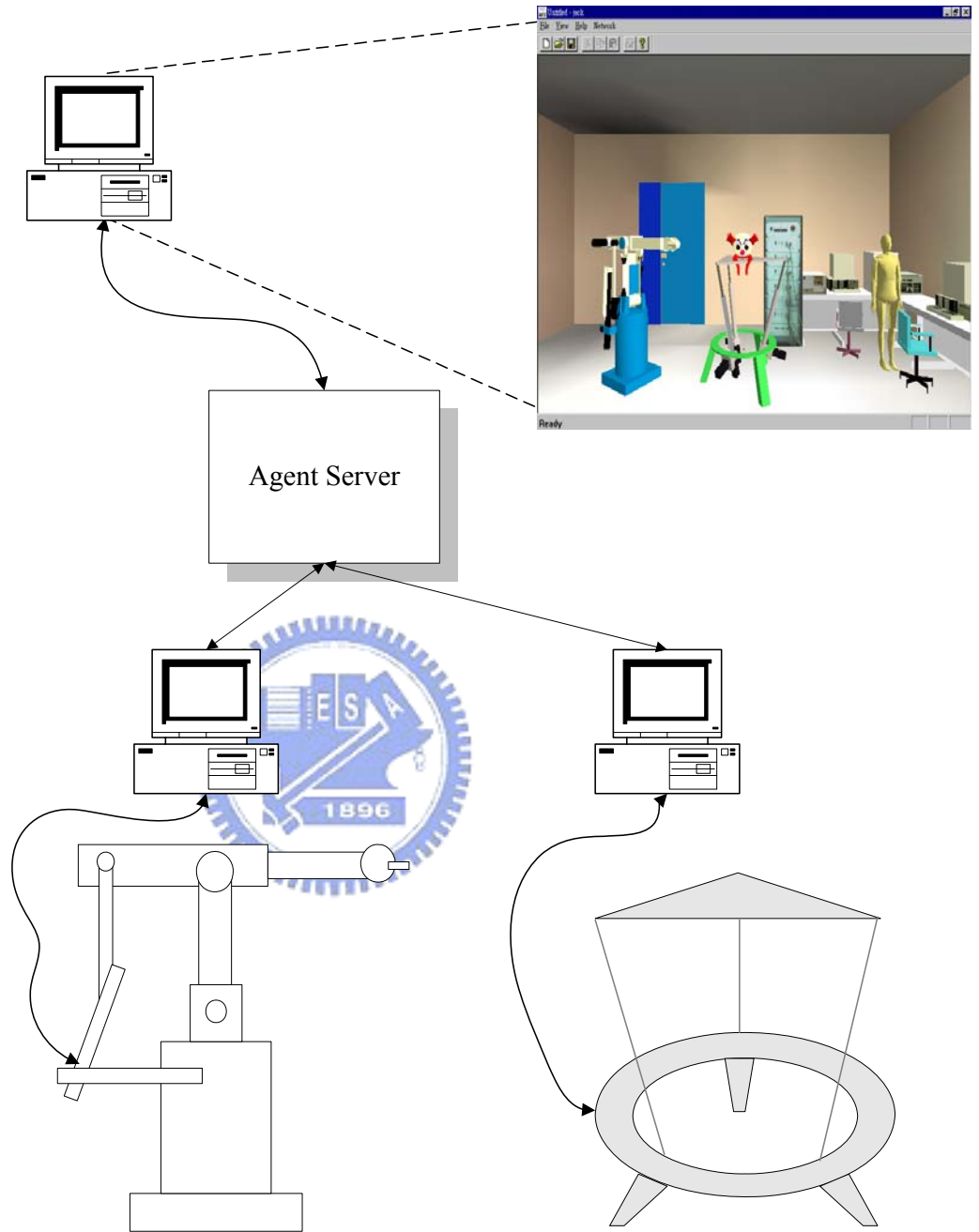


Figure 43. Architecture of virtual reality remote supervision for safeguard of robotics (Case Two).



Figure 44. The photograph of the two-robot system.

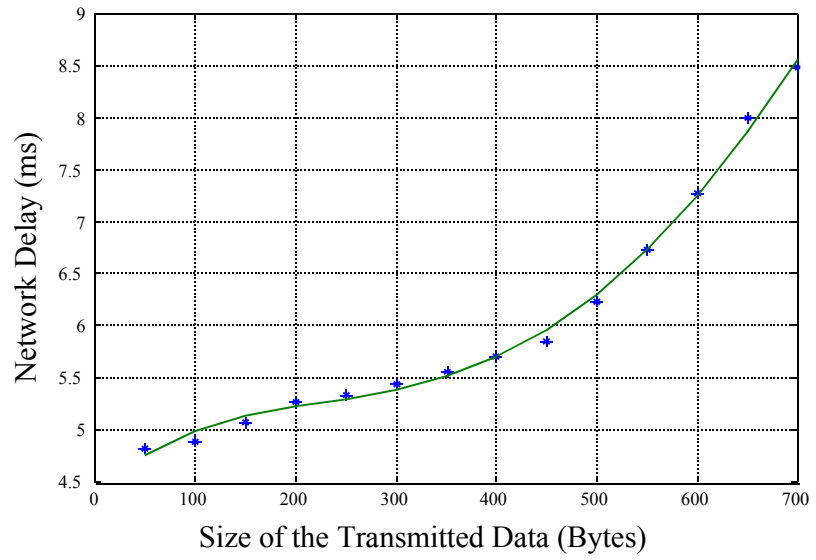


Figure 45. The network delay for the different size of the transmitted data between two robotic controllers.

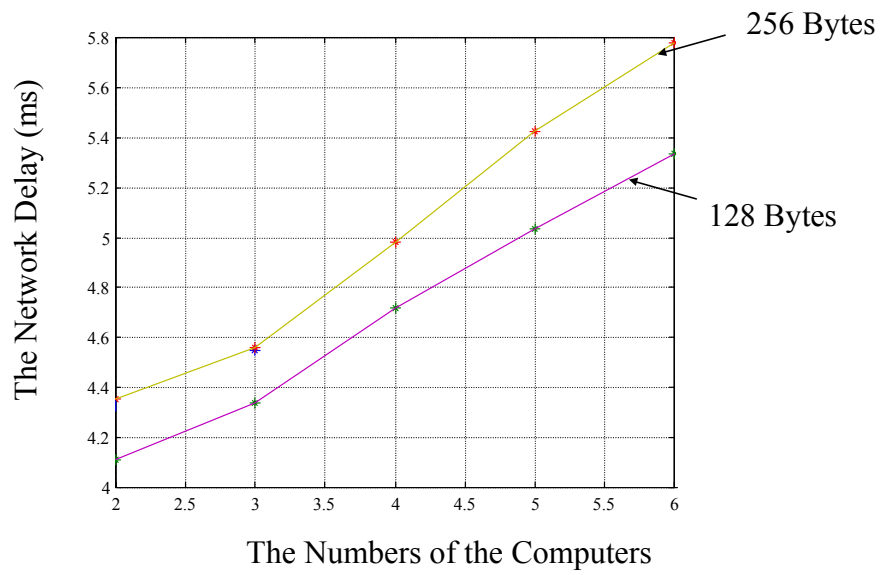
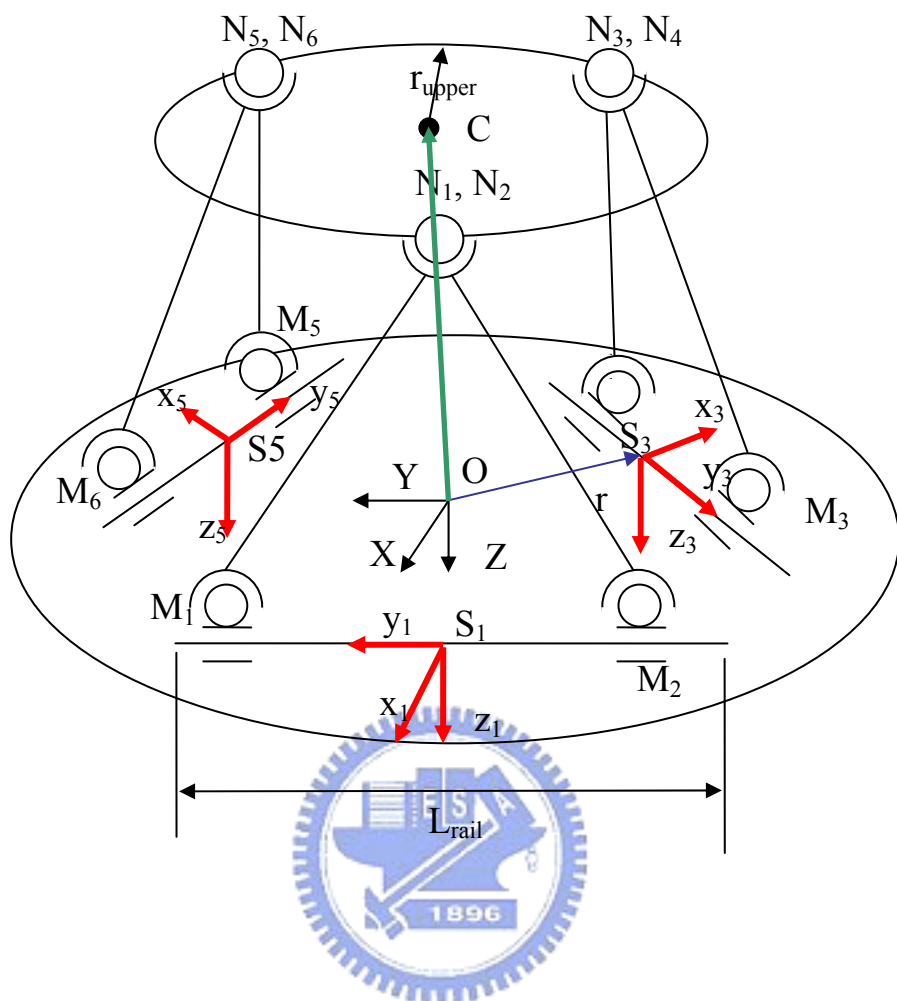


Figure 46. The network delay for the different size of the transmitted data among multiple robotic controllers.





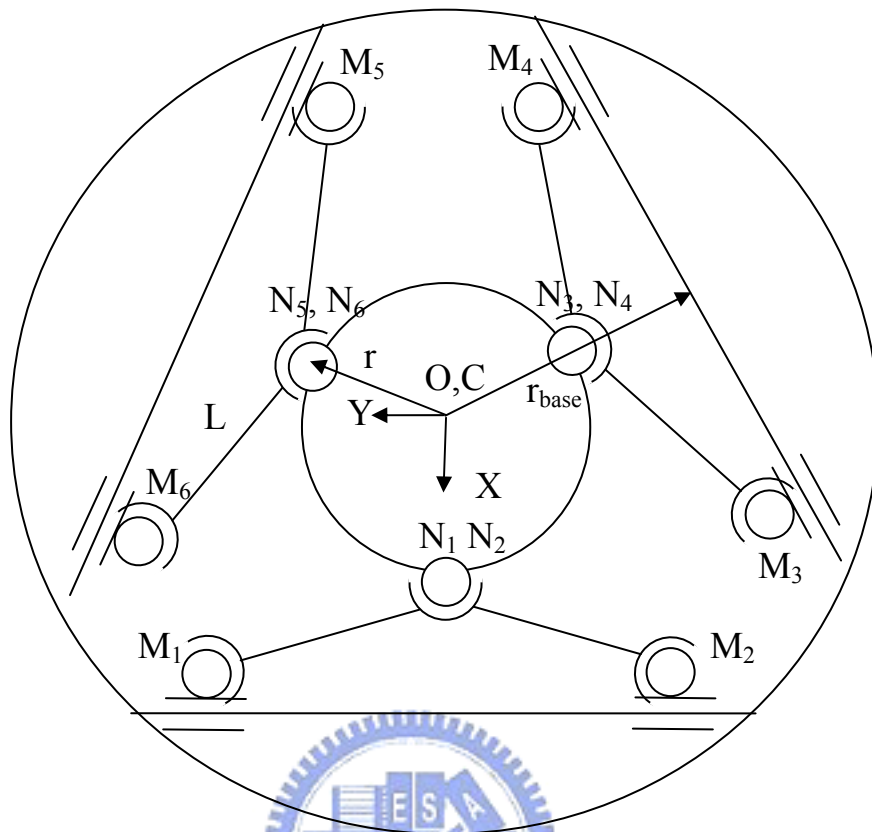


Figure 47. Top and front view of the Delta Hexaglide platform.

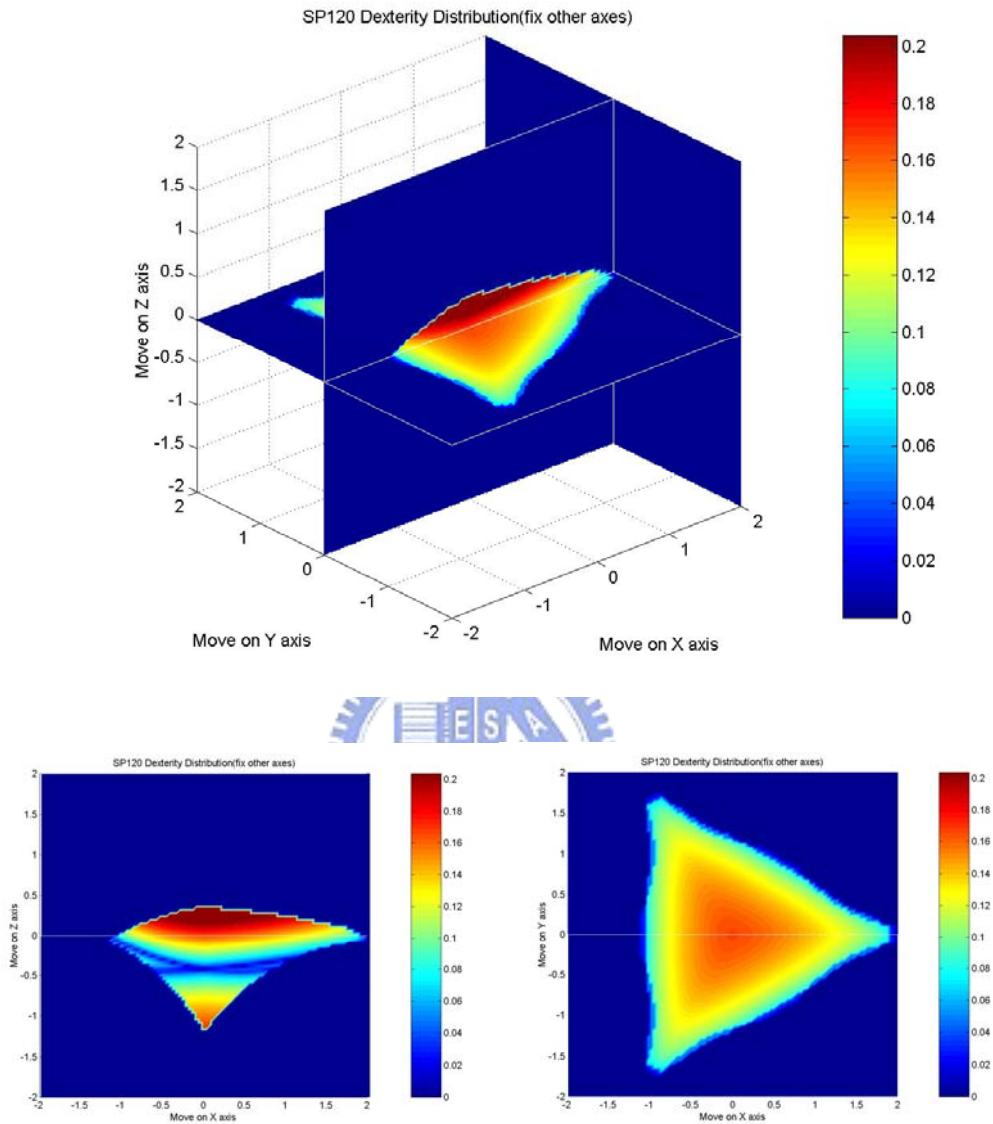


Figure 48. Dexterity of the Delta Hexaglide platform.

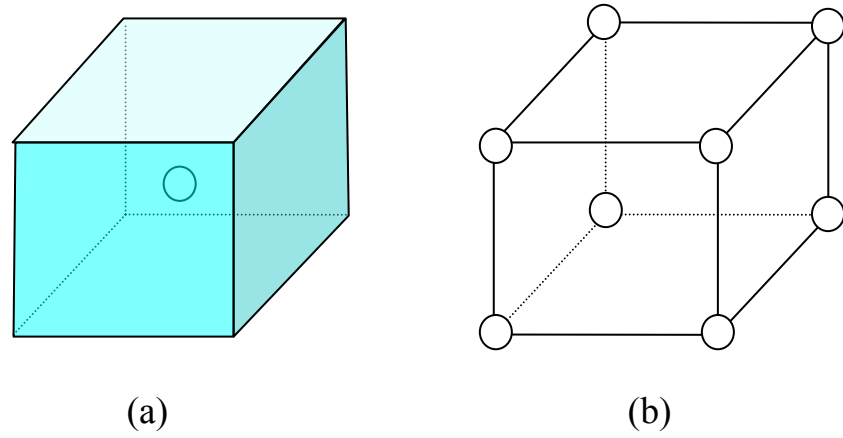


Figure 49. Graphical presentation of (a) the voxel with one center and (b) the cube with eight vertices.

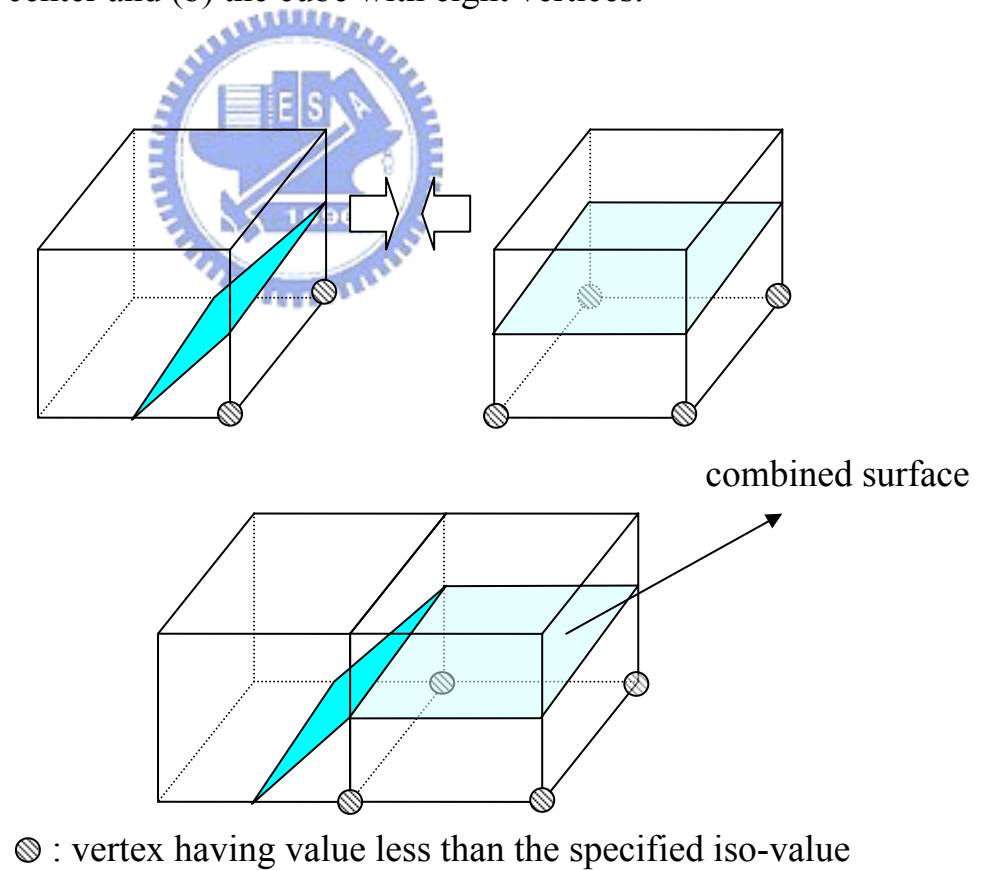
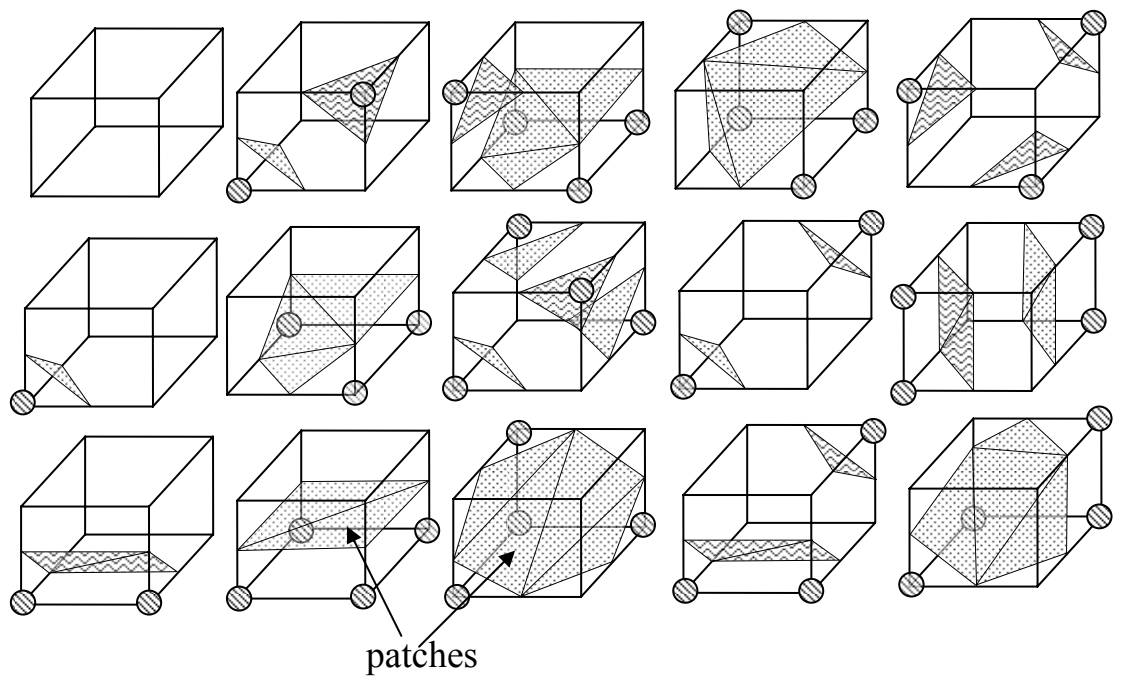


Figure 50. Surface representation in Marching Cube method.



⊙ : Vertex that is inside the workspace

△ : Front side

△ : Back side

Figure 51. 15 unique cube configurations.

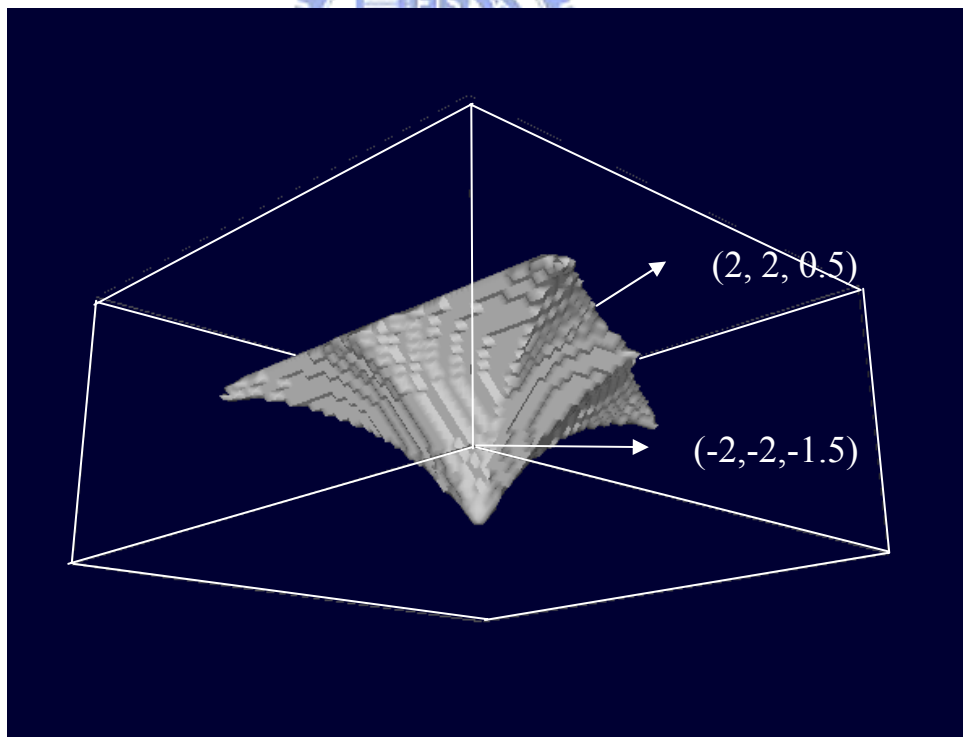
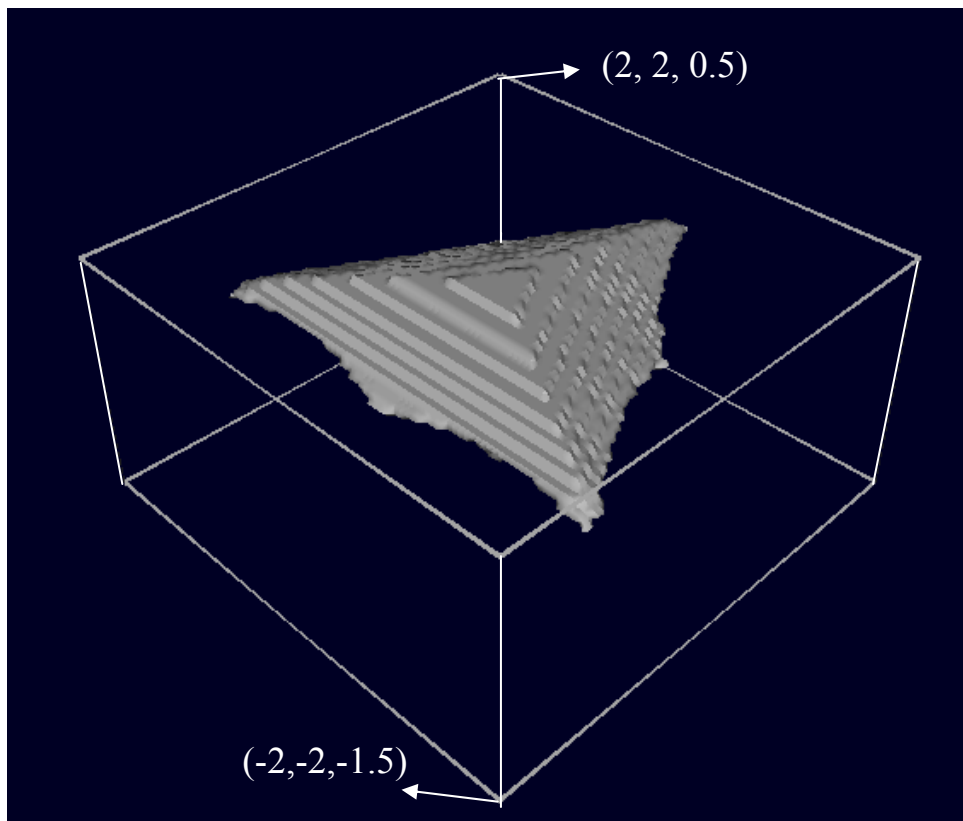
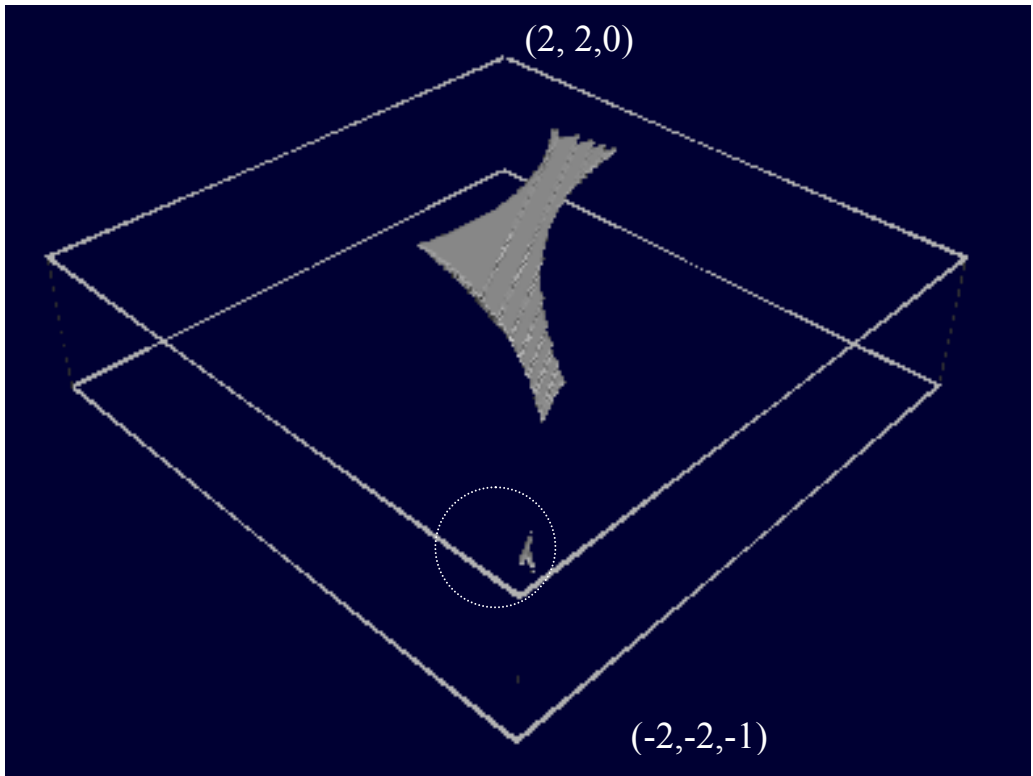
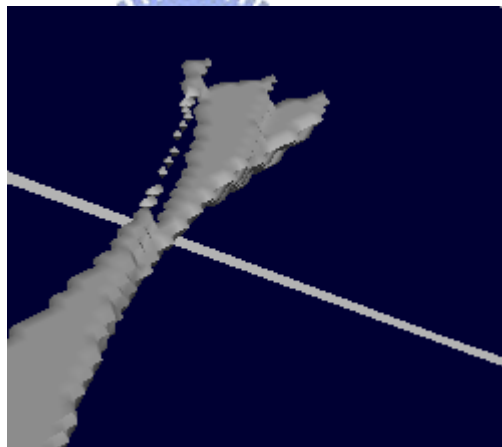
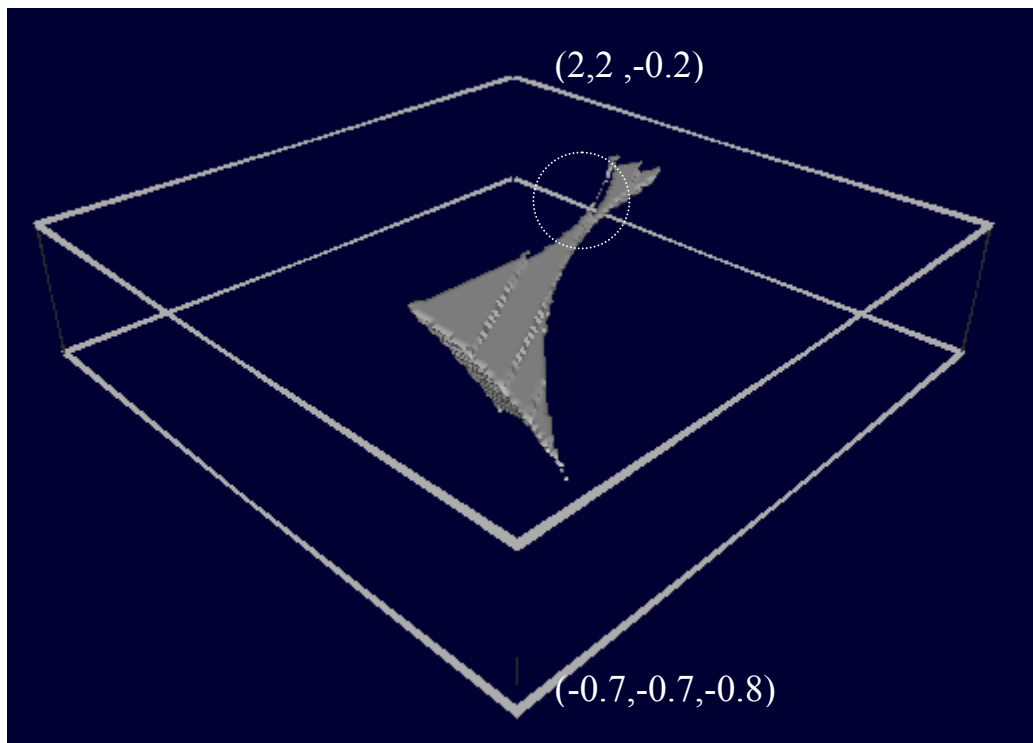


Figure 52. The translational workspace of Delta Hexaglide at  $\alpha = 0$ ,  $\beta = 0$ ,  $\gamma = 0$ ; 16416 faces are found on the boundary; 24775 cubes are found inside the workspace.



(a)



(b)

Figure 53. Workspace degeneracy: (a) workspace island (in dotted circle), and (b) workspace cavity (in dotted circle).

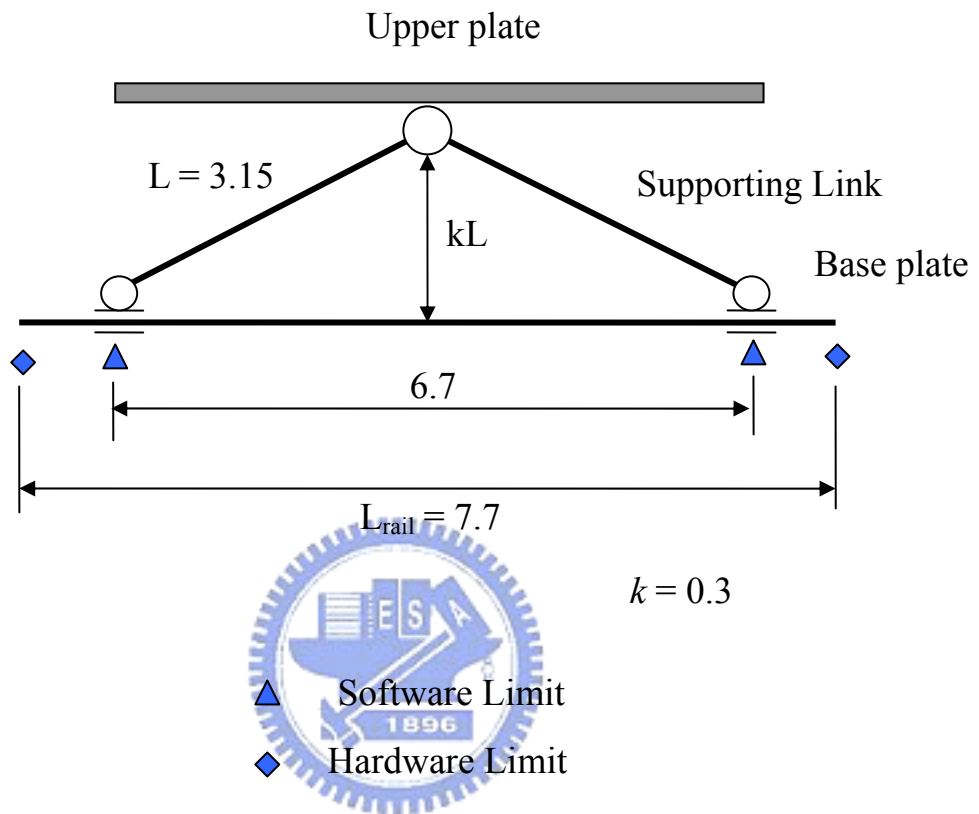


Figure 54. The parking position.



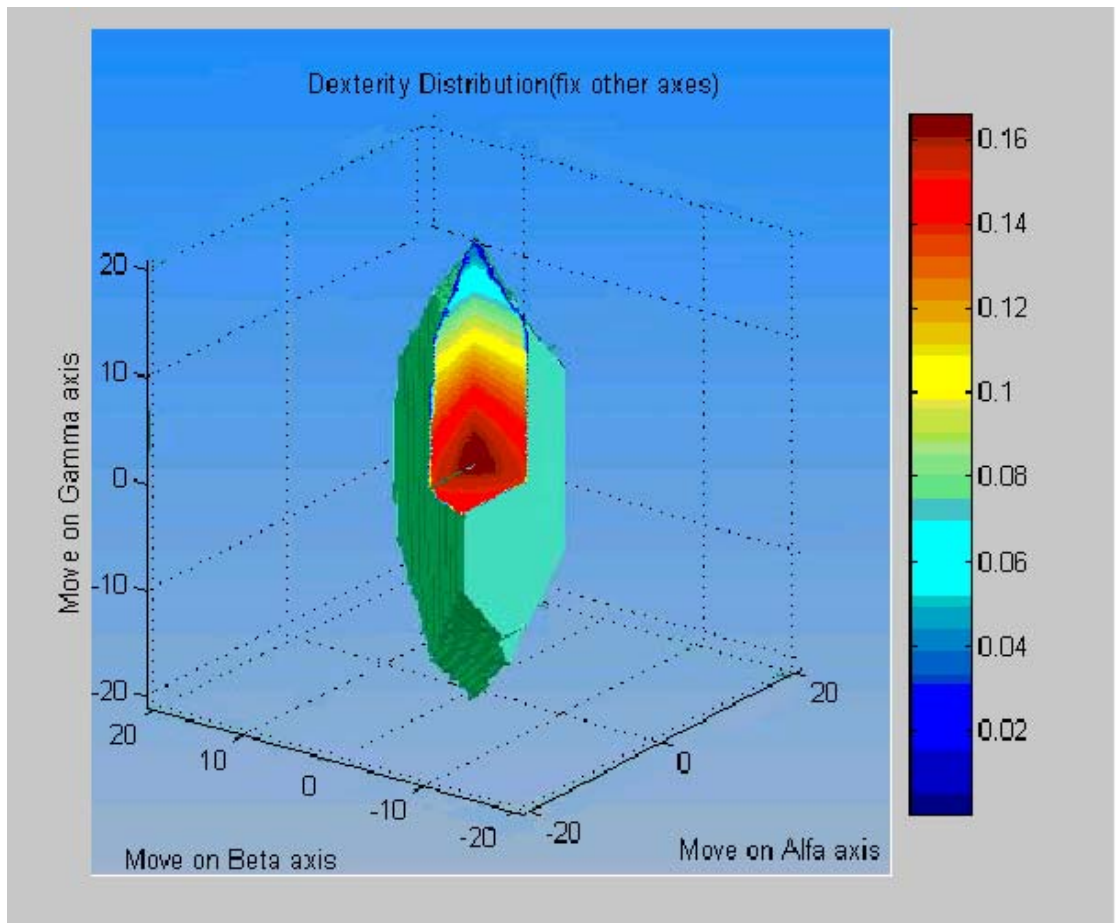


Figure 55. Combination of workspace and dexterity result of the Delta Hexaglide; shape complexity is 0.446 and workspace Volume is 5568.

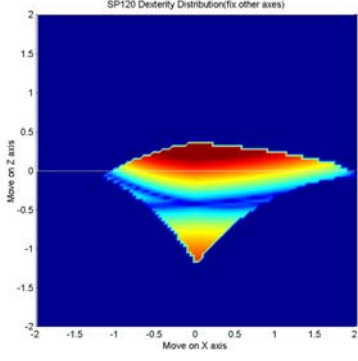
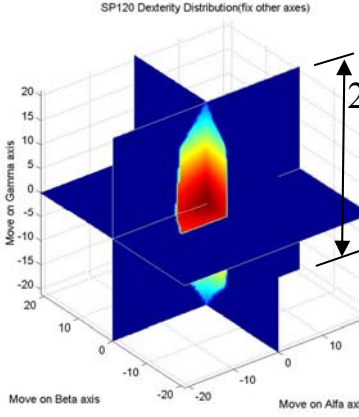
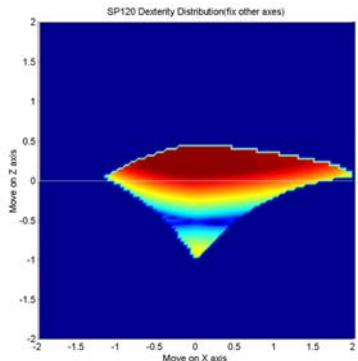
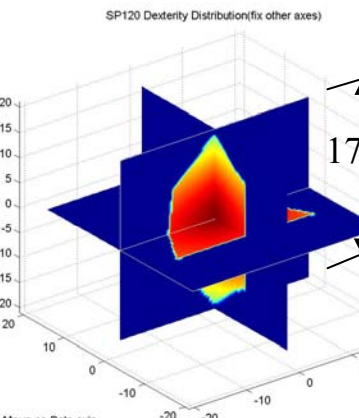
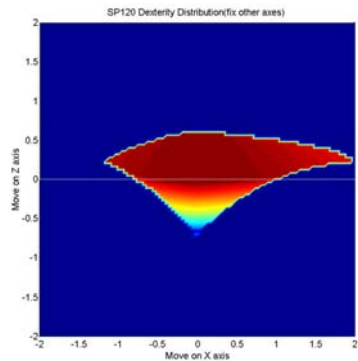
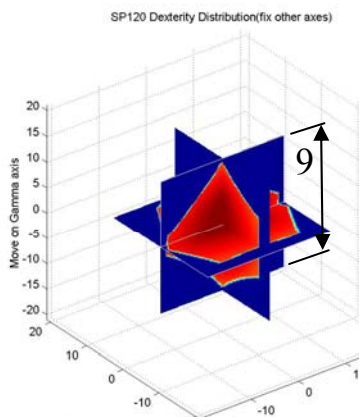
	Translational workspace	Rotational workspace
L=3.15	 <p>16416 faces, 24775 cubes</p>	 <p>20468 faces, 44544 cubes</p>
L=3.25	 <p>15868 faces, 23658 cubes</p>	 <p>20652 faces, 49116 cubes</p>
L=3.40	 <p>15340 faces, 22279 cubes</p>	 <p>17948 faces, 37946 cubes</p>

Figure 56. Different designs of the Delta Hexaglide platform.

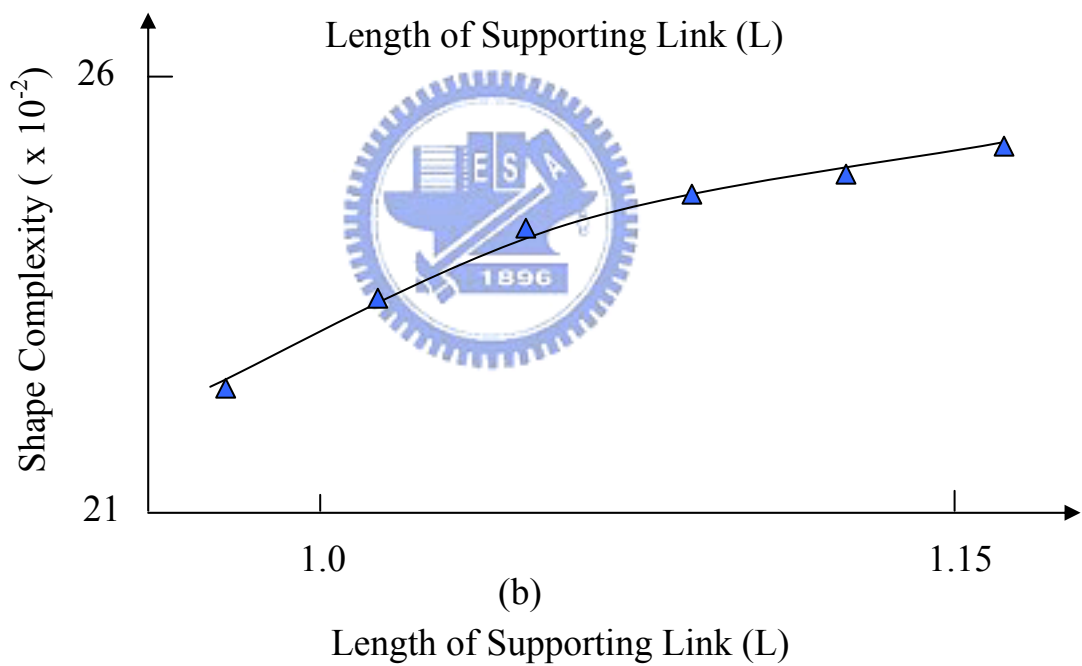
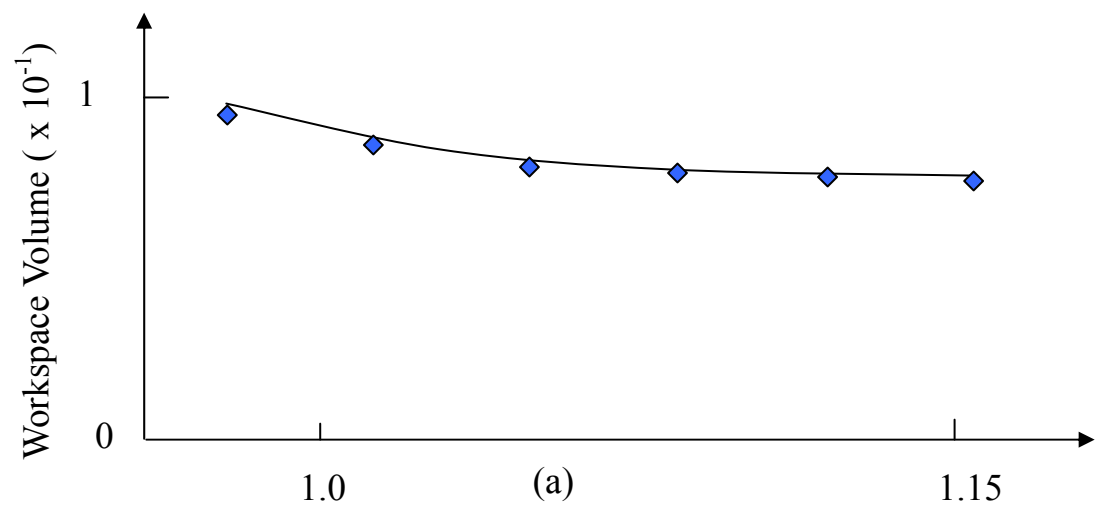


Figure 57. (a) Workspace Volume and (b) Shape Complexity of the translational workspace via different L.

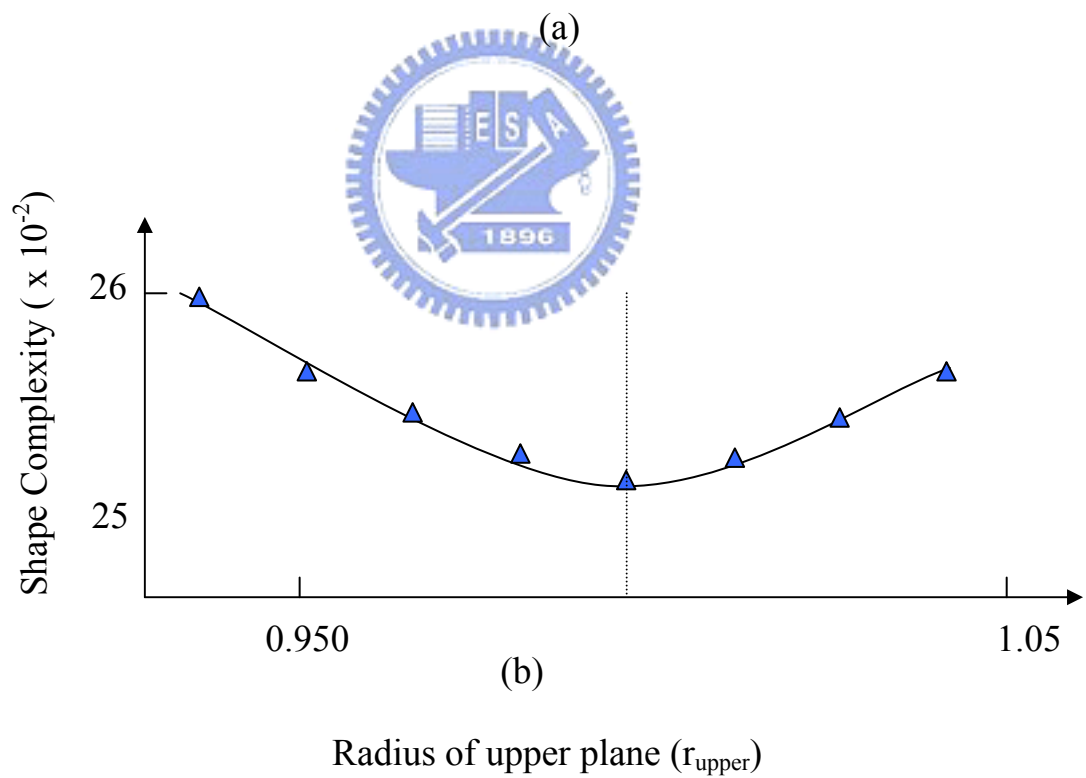
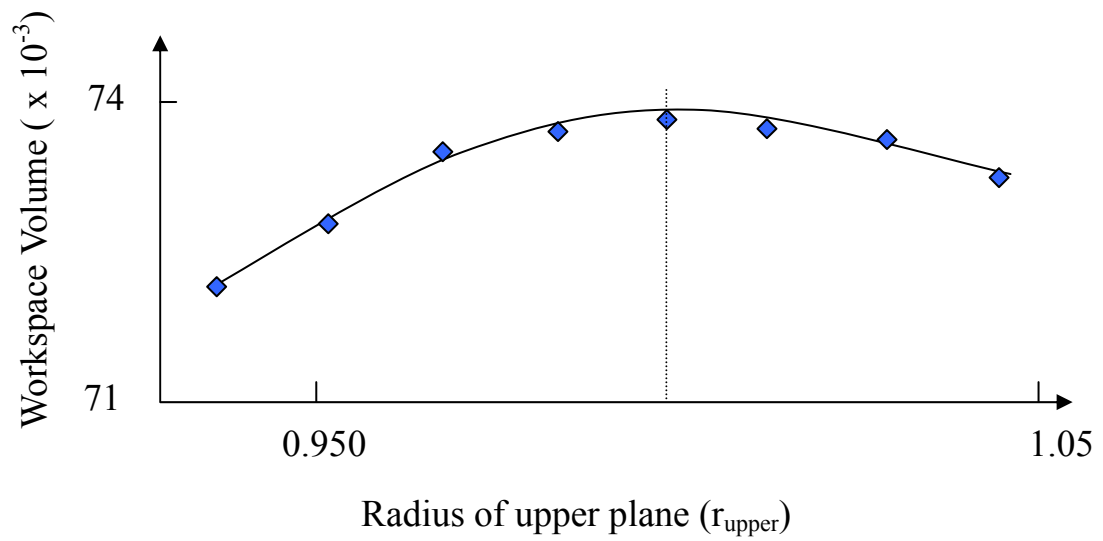
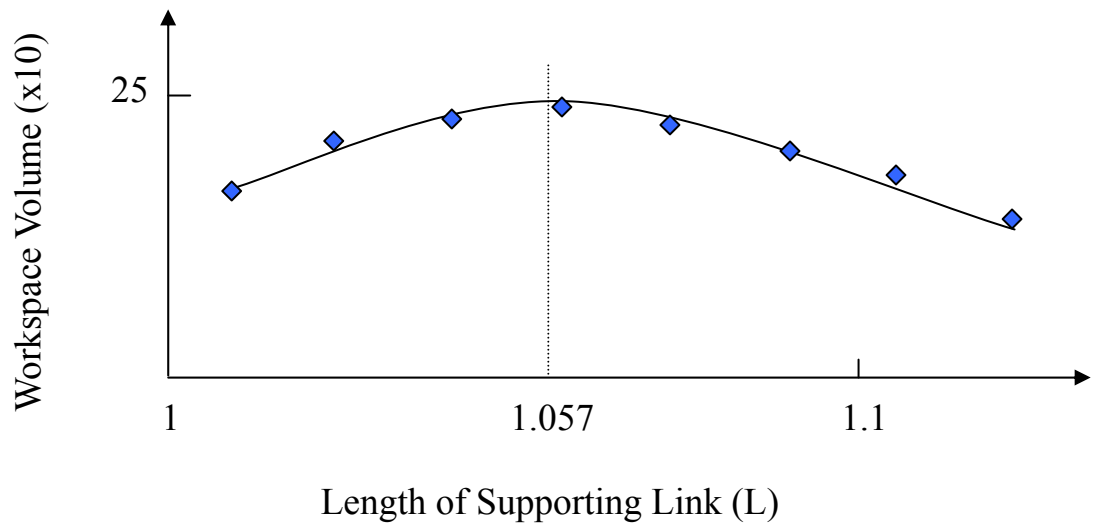
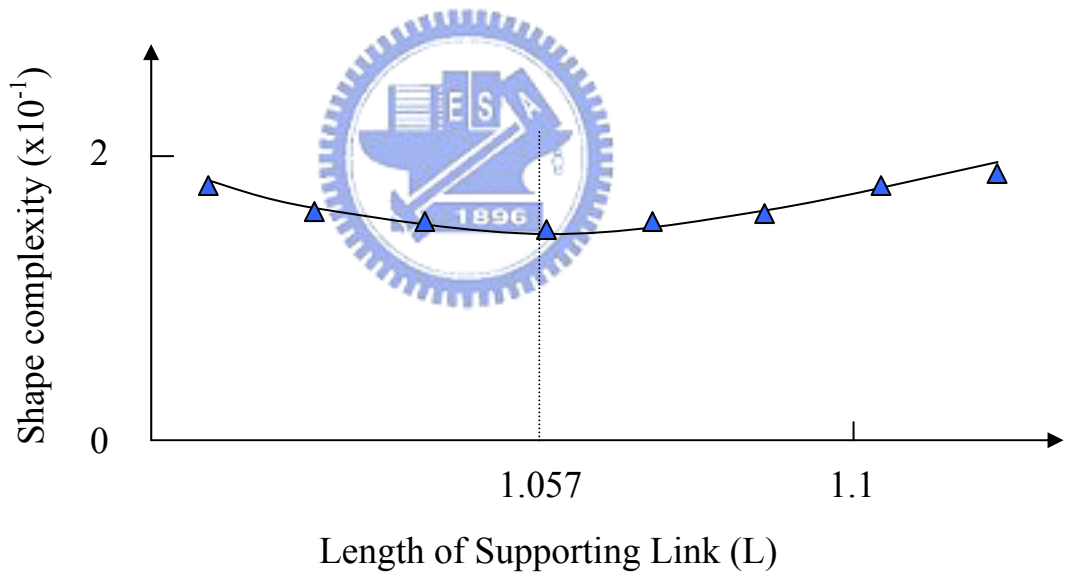


Figure 58. (a) Workspace Volume and (b) Shape Complexity of the translational workspace via different  $r_{upper}$ .



(a)



(b)

Figure 59 (a) Workspace Volume and (b) Shape Complexity of the rotational workspace via different L.

## Tables

M	T	class	Limb
8	2	$S^2$	PR
9	3	$S^3$	PC, PU, RC, RU
10	4	$S^4$	PS, RS
11	5	$S^5$	US, CS
12	6	$S^6$	SS*

*M*: mobility of the limb; *T*: joint D.O.F. of the limb;

\*: Improper kinematic arrangement

Table 1. List of spatial triads.

M	T	class	limb
9	3	$S^3$	PPP, PPR, PRP, PRR, RPR, RRR
10	4	$S^4$	PCR, PUR, PCP, RUR, CPR, UPR, CPP, URR
11	5	$S^5$	PCU, PUC, CPU, RCU, RUC, CRU, PCC, RCC, CPC, CRC, PUU, RUU, UPU, URU
12	6	$S^6$	PCS, PUS, RCS, RUS, PSC, PSU, RSC, RSU, CPS, UPS, CRS, URS

\*: Improper kinematic arrangement

Table 2. List of spatial quads.

F= n	T <sub>p</sub>	Class
6	6	S <sup>6</sup>
3	5	S <sup>5</sup>
2	4	S <sup>4</sup>

Table 3. The classes of limb used in the spatial FSPM.

F= n	qΔT	T <sub>p</sub>	q	T <sub>q</sub>	p= n - q	classes
5	-1	6	1	5	4	S <sup>6</sup> , S <sup>5</sup>
4	-2	6	1	4	3	S <sup>6</sup> , S <sup>4</sup>
			2	5	2	S <sup>6</sup> , S <sup>5</sup>
3	-3	6	1	3	2	S <sup>6</sup> , S <sup>3</sup>
2	-4	6	1	2	1	S <sup>6</sup> , S <sup>2</sup>

Table 4. The classes of limb used in the spatial SSPM.

$F=p$	$q(T_q - \lambda)$	$T_p$	$q$	$T_q$	$n$	Classes
5	-1	6	1	5	6	$S^6, S^5$
4	-2	6	1	4	5	$S^6, S^4$
			2	5	6	$S^6, S^5$
3	-3	6	1	3	4	$S^6, S^3$
2	-4	6	1	2	3	$S^6, S^2$

Table 5. The classes of limb used in the spatial TOPM.



$r_{upper}$	$r_{base}$	$L$	$L_{rail}$
3.47	3.9	3.15	6.7

Table 6. Dimensional parameters used in examples;  $L$  is varying from 3.15 to 3.40 in Figs. 12 and 13..



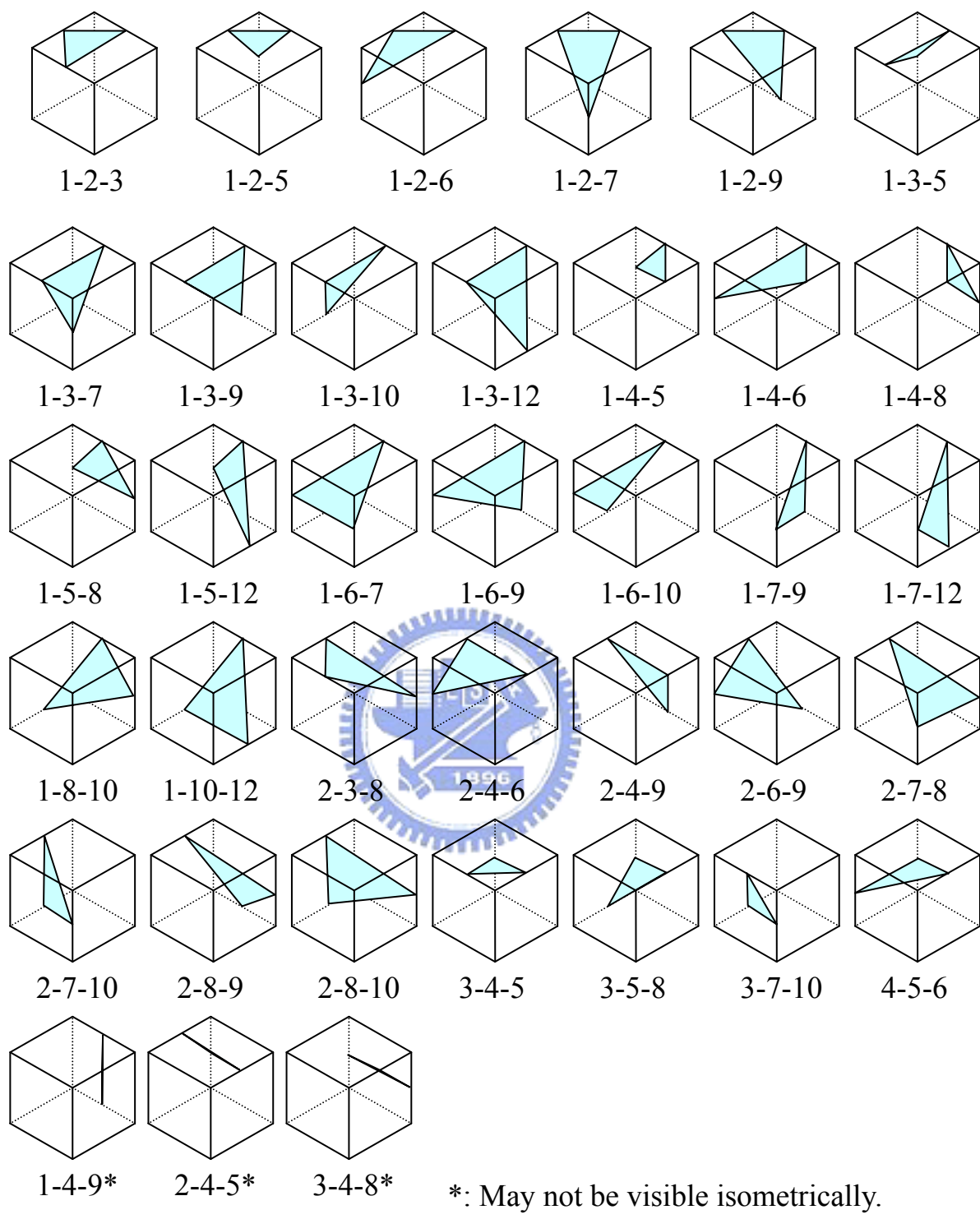


Table 7. Thirty-seven normal directions (vectors) of different patches for surface representation in marching cubes.

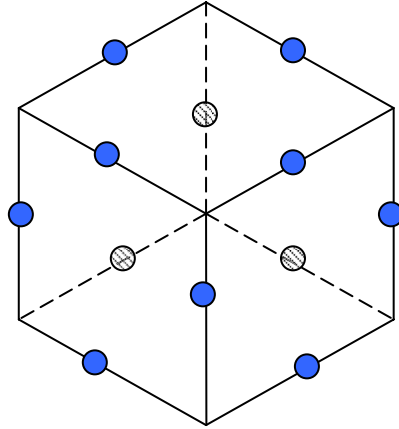
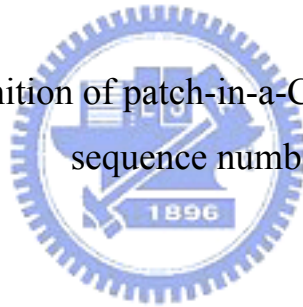


Table 7.1 Definition of patch-in-a-Cube based on the edge sequence number.



游武璋 (Wu-Jong Yu)

**List of Publication**

1. W-J Yu, C-F Huang, W-H Chieng, C-Y Gau, “The Integrated Application and Research of the Industrial Robotic Safeguard System with Ethernet”, IOSH, Journal of Institute of Occupation Safety and Health, Vol.10, No. 3, 2002, pp.218-230
2. Wu-Jong Yu, Chih-Fang HUNG and Wei-Hua CHIENG, “Design of Swinging-Block and Turning - Block Mechanism with Special Reference to the Mechanical Advantage”, JSME. Int. J. Series C, Vol.47, No.1, 2004, pp363-368
3. Wu-Jong Yu, Chih-Fang Huang and Wei-Hua Chieng, “Workspace and Dexterity Analyses of the Delta Hexaglide Platform,” Journal of Robotics and Mechatronics (JRM), Vol.20, No.1, 2007.5